BREAST CANCER PREDICTION

PROJECT REPORT

Submitted By

VEDHAVYAS ASOKAN

Reg. No. CCAVBCA026

For the award of the Degree of

Bachelor of Computer Application

(BCA)

(University of Calicut)

under the guidance of

Mr. Joju Sebastian

Assistant Professor

BACHELOR OF COMPUTER APPLICATION
DEPARTMENT OF COMPUTER SCIENCE
CHRIST COLLEGE
IRINJALAKUDA, KERALA

2021-2024

DEPARTMENT OF COMPUTER SCIENCE

CHRIST COLLEGE (AUTONOMOUS)

IRINJALAKUDA

CERTIFICATE

This is to certify that the project report entitled ***"BREAST CANCER PREDICTION"*** *is a bonafide record of the project work done by* ***VEDHAVYAS ASOKAN*** *in partial fulfillment of the requirement for the sixth semester of* ***Bachelor of Computer Application*** *in Department of Computer Science of* ***CHRIST COLLEGE (AUTONOMOUS) IRINJALAKUDA***

.

Mr.Joju Sebastian
Assistant Professor,CS
Internal Guide

Ms. Sini Thomas
Head of the Department
Computer Science

EXTERNAL EXAMINER **INTERNAL EXAMINER**

DECLARATION

We here by declare that this project work **"BREAST CANCER PREDICTION"** submitted by Christ College (Autonomous)Irinjalakuda,affliated to Calicut University in partial fullfillment of the requirement for the award of the Bachelor of Computer Application, is a record of original work done by us,under the guidance of Mr.JOJU SEBASTIAN,Department of Computer Science.

Place: Irinjalakuda VEDHAVYAS ASOKAN

ACKNOWLEDGEMENT

First and foremost we like to thank Lord almighty for his providence and for being the guiding light throughout the project. We wish to express my sincere gratitude to our beloved Department head for giving me all the facilities for our project. We take this opportunity to express my gratitude to the class teacher Ms. SOWMYA PS and head of the department Ms. SINI THOMAS who has been supported us throughout the course of this project. We are thankful for her aspiring guidance and valuable advice during the project work. We express my sincere thanks to my project guide Mr.JOJU SEBASTIAN for supporting and guiding throughout the project.We would take this opportunity to specially thank all other faculty members for their constant and continuous motivation. Finally we would like to thank my family and friends for giving valuable advice and moral support throughout our project.

ABSTRACT

BREAST CANCER PREDICTION is a desktop application that uses CNN algorithm to identify cancer and non cancerous cells in medica lorganization. This system automates the process of predicting cancer by inputting histopathological images and analyzing them to identify cancer and non cancer cells.

Contents

Chapter 1

1 Introduction

"Breast Cancer Prediction" introduces a revolutionary solution for predicting breast cancer, replacing traditional methods. This system incorporates deep learning algorithms to enhance the entire prediction process. The advantages of this project includes heightened accuracy, time efficiency, improved security, and reduced costs. As medical organizations embrace this user-friendly and innovative system, they are propelled into a more streamlined and technologically advanced future of breast cancer prediction.

.

1.1 Overview

The objective of the "Breast Cancer Prediction" project is to develop and implement a cutting-edge solution that automates breast cancer prediction using CNN algorithm. The goal is to replace traditional prediction methods with a more accurate, efficient, and secure solution, ultimately improving outcomes for patients and medical professionals.

Chapter 2

2 System Analysis

2.1 Purpose

Breast Cancer Prediction is a desktop applications that uses CNN algorithm to identify cancer and non-cancer cells in medical organizations. The system automates the process of predicting cancer by inputing histopathological images and analysing them to identify cancer and non-cancer cells. The system is typically composed of a software. The histopathological image of individuals browsed from the system are taken and send to the software, upon clicking predict button the image is analysed and compared with the trained dataset. The dataset contains information about each person's multiple histopathological images cells. The weights of the images are recorded and based on the weights of the new images the result is predicted.

2.1.1 Existing System

The existing system of the breast cancer prediction primarily relies on traditional prediction methods, such as using machine learning algorithms which predicts optimally the cancer and non-cancer cells from the trained histopathological dataset. Here, when a new histopathological image is taken the result accuracy is poor.

2.1.2 Proposed System

Breast Cancer Prediction is a desktop application that is in need for breast cancer identification on lab. Here, Project wise there is a GUI created that takes an image as input and classifies that it's a cancerous or non-cancerous. In this system when a new histopathological image is given here the result is predicted optimally and accuracy is good due to the use of CNN algorithm. Medically speaking Biopsy is taken place here .The histopathological images (Histology – Study of tissues , Pathology – Study of diseases) are taken from the monitor using advanced electron microscope by selecting a sample tissue and then processed and cut into thin sections then stained with Hematoxylin – Eosin (H&E).

2.2 Problem definition

The proposed project aims to tackle the limitation and inefficiency of traditional prediction methods by developing "Breast Cancer Prediction". The existing system predicts the cancer using a machine learning algorithm which only predicts the result based on the image given as input is the trained histopathological images which often lead to inaccuracies. To address these issues, this project will leverage deep learning method to provide a more accurate, secure, and

user-friendly approach to breast cancer prediction. By integrating seamlessly with existing systems and adapting to varying environmental conditions, this system promises to revolutionize breast cancer prediction, delivering enhanced accuracy, efficiency, and security for medical organizations.

2.3 FEASIBILITY STUDY

After the problem is clearly understood and solutions proposed, the next step is to conduct the feasibility study. Feasibility study is defined as evaluation or analysis of the potential impact of the proposed project or program. The objective is to determine whether the proposed system is feasible. There are three aspects of feasibility study to which the proposed system is subjected as discussed below.

2.3.1 Technical Feasibility

The proposed breast cancer prediction system using CNN algorithms is technically feasible with all required tools readily available. The system, implemented in Python, leverages CNN for its speed, efficiency, and low resource requirements. However, the accuracy of these algorithms depends on the number of trained images, necessitating thorough testing and optimization. The system requires a well-curated database of histopathological images, with regular updates and management crucial for accurate prediction. Robust measures for data protection and compliance with privacy regulations are essential due to the sensitive nature of the data involved. In conclusion, the technical feasibility of the system is evident, and with proper planning, testing, and attention to privacy and security, it can be a viable solution for various settings, enhancing operational efficiency.

2.3.2 Economical Feasibility

In this proposed system, all the tools used are free and open source. So that development cost is minimum. The hardware is used to run the system. It is built in our laptop. And also hardware requirement is feasible and not much breast cancer prediction system maintenance is required. The development of the system will not need a huge amount of money. It will be economically feasible. And the money spend for the application will be worth. Hence, the proposed system is economically feasible.

2.3.3 Operational Feasibility

The feasibility of a breast cancer prediction system using CNN algorithms is crucial.CNN's ease of implementation reduces development time and costs, and its low resource requirements allow deployment on various hardware configurations. The system requires a well-maintained dataset of pre-registered faces, regular updates, and management to accommodate user base changes. Addressing privacy and security concerns is critical due to the sensitive nature of

biometric data. User acceptance is a significant factor for successful implementation, requiring effective communication of benefits, addressing concerns, and providing adequate training. With proper planning, thorough testing, and compliance with privacy regulations, a breast cancer prediction system using CNN can predict optimal breast cancer results and enhance operational efficiency.

Chapter 3

3 Software Requirement Specification

3.1 Purpose

The breast cancer diagnosis system aims to leverage deep learning algorithms to accurately analyze histopathological images and automatically predict whether cancer is present. Key goals are achieving real-time cancer screening with minimal errors, extracting insightful image features, training robust models CNN, and obtaining analytics to uncover patterns in cancer data - ultimately enhancing efficiency, reliability and research capabilities compared to manual diagnosis processes. The system requires histopathology image datasets to train models to classify images as normal or malignant based on texture, shape and cell characteristics. It will provide a practical tool set to augment and automate parts of the breast cancer screening process.

3.2 Scope

The scope of this project encompasses the development of a software-based solution for automated breast cancer prediction from histopathological images. It leverages deep learning techniques to analyze cell structure and identify malignant indicators.

3.3 Overall Description

The desktop application caters to medical professionals, offering streamlined analysis of digitized histopathology slide images. Convolutional Neural Networks, it assesses cell morphology, texture, shape, and spatial characteristics to classify images as malignant or benign with confidence scores. The software guides users through preprocessing, feature extraction, model execution, and post-processing steps. Results, metrics, and visual heatmaps highlighting malignant regions are easily accessible. The back-end architecture includes a database, modeling module, processing pipelines, integration logic, and access control layers. Through rigorous training and testing with labeled datasets, the system ensures accuracy, aiding medical decision-making and enhancing patient care.

3.3.1 Product Perspective

The breast cancer prediction system aims to integrate smoothly with current medical systems, empowering healthcare professionals with accurate diagnostics. It aligns with organizational goals of enhancing patient care and operational efficiency, offering a user-friendly interface and scalability to adapt to diverse healthcare settings and technological advances.

3.3.2 Product Functionality

The breast cancer prediction system functions by analyzing histopathological images using CNN algorithm to classify cancerous and non-cancerous cells. Users input images into the system, which then processes them, compares them with a trained dataset, and generates predictions with high accuracy. The system offers a user-friendly interface for seamless interaction, contributes to faster and more accurate cancer detection, and integrates with existing medical infrastructure to enhance overall diagnostic capabilities.

3.3.3 Users and Characteristics

The breast cancer prediction system caters primarily to healthcare professionals involved in cancer diagnosis, such as doctors, radiologists, and pathologists. These users typically possess medical expertise and are familiar with diagnostic processes. They require a system that is intuitive, reliable, and capable of providing accurate predictions to support their clinical decision-making. Additionally, administrators responsible for implementing and maintaining the system within healthcare organizations may also interact with it, requiring a broader understanding of its functionality and integration capabilities.

3.4 Specific Requirements

3.4.1 Hardware Requirements

- System: IBM-Compatible PC
- Processor: Intel Core i3 or above
- Speed: Above 1GHz
- RAM capacity: 4 GB
- Hard Dsk drive: 500 GB
- Keyboard: Standard
- Mouse: Standard
- Monitor: SVGA Color

3.4.2 Software Requirements

- Front End:Python
- Back End: Python
- IDE: Anaconda

3.5 Functional Requirements

It contains three main modules.

- 1.Automated Cancer Prediction
- 2.Real-Time Prediction
- 3.User Interface

Automated Cancer Prediction

The system should automate the process of predicting breast cancer using CNN algorithms based on input histopathological images.

Real-Time Prediction

The system should offer real-time breast cancer prediction capabilities, enabling medical professionals to obtain results promptly.

User Interface

The user interface should be user-friendly and intuitive, allowing medical professionals to input histopathological images easily and view predicted results.

3.6 Non Functional Requirements

Non-functional requirements define the overall qualities or attributes of the resulting system. Non-functional system place restrictions on the product being developed, the developed process, and specify external constraints that the product must meet. Examples of non-functional requirements include safety, security, usability, reliability and performance requirements. Project management issues (costs, time and schedule) are often considered as non-functional requirements. The principal non - functional constraints which are relevant to critical systems :

- performance
- security
- safety
- usability

Performance

Performance requirements concern the speed of operation of a system. Types of performance requirements :

- Response requirements (how quickly the system reacts to a user input).

- Throughput requirements (how much the system can accomplish within a specified amount of time).
- Availability requirements (is the system available for service when requested by end users). The speed of operation of this system is adequate for the requirements.

Reliability

- Reliability is the ability of a system to perform its required function under stated conditions for a specified period of time.
- constraints on the runtime behavior of the system. This system is reliable because its functionalities can be done on the required conditions.

Safety

Safety requirements are not required which exclude unsafe situation from the possible solution of the system.

Usability

Usability is the ease with which a user can learn to operate, prepare inputs for, and interpret outputs of system or components. Usability requirements include :

- information error messages.
- well-formed user interfaces.

3.7 Interface Requirements

3.7.1 Hardware interfaces

The system must run over the internet, all the hardware shall require to connect internet will be hardware interface for the system. As for example modem, WAN-LAN.

3.7.2 Software interfaces

Software interface required for the working of the project is the appropriate operating system.

3.7.3 Communication interfaces

The user of the site communicate using the network connectivity and the data set is accessed

3.8 Security Requirements

- User accesses only their account.
- Validation of input is handled.
- This application containing the computer systems is physically secured against arms or surreptitious entry by intruders.
- Users must be authorized carefully to reduce changes of any such user giving access to an intruder in exchange for a bribe or other favour.

3.9 Platform Used

Windows 11 is the operating system chosen for the "Breast Cancer Prediction" project. Windows 10 provides a user-friendly interface and robust support for various hardware components and software applications. It is widely used in both personal and professional settings, making it a suitable choice for medical organizations implementing the breast cancer prediction system. Additionally, Windows 11 offers security features and regular updates to ensure system stability and reliability.

3.10 Technologies Used

PYTHON

Python is a versatile, high-level programming language known for its simplicity and readability. It offers dynamic typing and automatic memory management, making it suitable for various domains such as web development, data analysis, artificial intelligence, and automation. Python's extensive standard library provides pre-built modules and packages for tasks ranging from file manipulation to network programming. Its interpreted nature allows for rapid development and easy debugging. Python supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Its popularity is fueled by a vibrant community, extensive documentation, and numerous third-party libraries. Overall, Python's simplicity, flexibility, and robust ecosystem make it an ideal choice for beginners and seasoned developers alike

CNN

Convolutional Neural Networks (CNNs) are a class of deep neural networks primarily used for image recognition and computer vision tasks. They consist of multiple layers, including convolutional layers, pooling layers, and fully connected layers. CNNs excel at learning hierarchical representations of features from input data. Convolutional layers apply filters to input images, extracting features such as edges and textures. Pooling layers reduce spatial dimensions while preserving important features. Fully connected layers combine extracted features for classification or regression tasks. CNNs leverage parameter sharing and local connectivity to efficiently learn patterns from large datasets. Their success in image recognition tasks has made them a cornerstone in various fields, including autonomous driving, medical imaging, and facial recognition

Chapter 4

4 Design Document

4.1 Purpose

The purpose of the Design Document for the "Breast Cancer Prediction" project is to provide a concise yet comprehensive overview of the system's architecture, components, functionalities, and interactions. It serves as a roadmap for development, ensuring alignment with project goals and requirements. Additionally, the document facilitates communication and collaboration among team members and stakeholders, guiding informed decision-making and ensuring the successful development and implementation of the breast cancer prediction system.

4.2 Scope

The scope of the "Breast Cancer Prediction" project involves developing an automated breast cancer prediction system using deep learning algorithms. This includes designing a user-friendly interface for inputting histopathological images and implementing backend logic for image processing and prediction generation. Additionally, the project encompasses addressing data security and privacy concerns to ensure compliance with medical regulations and standards. Ultimately, the goal is to deliver a reliable and efficient system that aids medical professionals in accurately predicting breast cancer outcomes.

4.3 Overview

The Design Document for the "Breast Cancer Prediction" project provides an overview of the system architecture, functionalities, and key design considerations. It outlines the scope of the project, which involves developing an automated breast cancer prediction system using and deep learning algorithms. The document addresses the purpose of the project, which is to create a reliable and efficient system that aids medical professionals in accurately predicting breast cancer outcomes. Additionally, it highlights the importance of data security and privacy, as well as adherence to ethical and legal standards. Overall, the Design Document serves as a comprehensive blueprint for the development and implementation of the breast cancer prediction system.

4.4 Data Design

Database are the storehouses of data used in the software systems.The data is stored in tables inside the database.Several tables are created for the manipulation of the data for the system.Two essential settings for a database are

- Primary Key- The field that is unique for all the record occurrence.

- Foreign Key- The field used to set relation between tables.

Normalization

- Normalization is the process of organizing the data in the database.
- Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate the undesirable characteristics like Insertion, Update and Deletion Anomalies.
- Normalization divides the larger table into the smaller table and links them using relationship.
- The normal form is used to reduce redundancy from the database table.

Chapter 5

5 Development of the System

The "Breast Cancer prediction" system was developed using an agile approach, incorporating deep learning algorithms to provide an accurate and user-friendly solution for breast cancer prediction in medical organizations, replacing traditional prediction methods.

Chapter 6

6 System Testing

Testing is the penultimate step of software development.An elaborate testing of the data is prepared and the system is using the test data,while doing testing,errors are noted and correction is made.The users are trained to operate the developed system.Both hardware and software securities are made to run the developed system successfully.System testing is aimed at ensuring the system works accurately before the live operation commences.

6.1 Test Plan

6.1.1 Scope

This test plan will cover the following testing activities as identified in the testing strategy.

- White Box Testing

White box testing is a testing technique, that examines the program structure and derives test data from the program logic/code. In white box testing, the UI is bypassed. Inputs and outputs of ratings are tested directly at the code level and the results are compared against specifications. It Reveals errors in "hidden" code.

- Black Box Testing

Black - box testing is a method of software testing that examines the functionality of an application based on the specifications. Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end - user would.

- Unit Testing

The module includes four main components to undergo unit testing. It checks the syntax error, logic error and validity of the program correctness. This test will be performed by the developers.

- Integration Testing

After all components pass the unit testing, they get to be tested whether they work correctly when they are running concurrently and communicating to each other. Integration tests exercise an entire subsystem and ensure that a set of components play nicely together.

- Internal Data Testing

We will test the validity of the data before it enters the database to avoid any problems that may face us in the database. We will test the encryption of the personal information of all the users along with the user names and passwords to ensure maximum security of the users privacy.

6.1.2 Software risk issues

The "Breast Cancer Prediction" system leverages complex deep learning algorithms to predict breast cancer from histopathological images. While this provides enhanced accuracy compared to traditional methods, it also introduces certain software risks that need to be proactively managed. Key risks include model overfitting on limited training data, algorithmic limitations in capturing real-world variations, integration challenges between system components, performance bottlenecks when dealing with large images, lack of explainability of predictions, and potential security vulnerabilities given the sensitive nature of medical data. Careful software design, extensive testing and validation, performance tuning, security hardening, and adoption of explainable AI techniques are crucial to mitigate these risks. Additionally, bias in training data, software complexity, and obscure failures related to machine learning models require rigorous monitoring and maintenance processes. Overall, the advanced techniques used in Breast Cancer Prediction make software risk management an integral part of developing a robust and reliable solution.

6.1.3 Features to be tested

- Image Upload: Test uploading of histopathology images in different formats, sizes, and resolutions. Verify proper validation and preprocessing.
- User Interface: Test all UI flows and elements like buttons, forms, navigation, and data displays. Check for responsiveness across devices.
- Cancer Prediction: Test prediction accuracy with diverse sample images. Check for correct classification of cancer vs non-cancer.
- Model Training: Validate training process completed without errors. Confirm models are saved properly for prediction.
- Performance: Test system response times and resource usage under different normal and peak load conditions.
- Security: Validate only authorized access to prediction results. Check for data encryption and other security measures.
- Error Handling: Verify proper response for invalid inputs or missing images. Check for fail-safe behavior.
- Integration: Test seamless data flow between UI, preprocessing, and prediction components. Confirm no errors during integration.
- Compatibility: Validate working on different OS versions and hardware configurations per compatibility matrix.
- Functional Flows: Test end-to-end flows like user login, image upload, prediction, and result display for expected behavior.

6.2 Test consolidation

6.2.1 Test item

The items or features to be tested in the test cases are included in the document. Each and every user input is tested. The present condition of the system is tested. It is checked to make sure that environment is ready for the application to work.

6.2.2 Input specifications

These are the inputs required to execute the test case. All required inputs including data elements and values are given below.

Data	Values
Image	histopathological images

Chapter 7

7 System Implementation and Maintenance

7.1 Implementation

System implementation is the conversion of new system into an operating one which involves creating compatible files, training clients and installing hardware. User training is crucial for minimizing resistance to change and giving chance to prove its worth. Training aids user friendly manuals and healthy screens provide the user with a good start. Software maintenance follows conversion to the extent that changes are necessary to maintain satisfactory operations relative to changes in the user's environment. Maintenance often includes minor enhancements or corrections to the problem that surface late in the systems operations. In the implementation phase, the team builds the components either from scratch or by composition. Given the architecture document meant from the design phase and the requirement document from the analysis phase, the team should build exactly what has been requested, though there is still room for innovation and flexibility. For example, a component may be narrowly designed for this particular system, or the component may be made more general to satisfy a reusability

- Careful planning
- Investigation of system and constraints
- Design the methods to achieve changeover.
- Training the staff in the changed phase.
- Evaluation of change over method.
- The method of implementation and time scale to be adopted are found out initially.

7.2 Maintenance

This phase occurs as a result of deploying the whole system at the end users organization. They will perform the beta testing at the end users and inform to the developers about any needed modification to the application. The customer records all the problems that are encountered during the beta testing and reports these to the developer at regular intervals.

7.2.1 Corrective Maintenance

Even with the best quality assurance activities, it is likely that the customer will uncover defects in the software. Corrective maintenance changes the software to correct the defects. Corrective Maintenance activity may consist of repair,

restoration or replacement of equipment. This activity will be result of a regular inspection, which identifies the failure in time for corrective maintenance to be planned and scheduled, then performed during a routine maintenance shutdown.

7.2.2 Adaptive Maintenance

Over time, the original environment(CPU, operating system, business rules, external product characteristics) for which the software was developed is likely to change. Adaptive maintenance results in modification to the software to accommodate changes to its external environment.

7.2.3 Enhanced Maintenance

As software is used, the customer/user will recognize additional functions that will provide the benefit. Perfect maintenance extends the software beyond its original functional requirements.

7.2.4 Preventive Maintenance

Computer software deteriorates due to change, and because of this preventive maintenance often called software re-engineering, must be conducted to enable the software to serve the needs of its end users. Preventive maintenance makes changes to computer programs so that they can be more easily corrected, adapted and enhanced.

Chapter 8

8 Conclusion and Future Scope

8.1 Conclusion

Breast Cancer prediction systems have seen remarkable advancements, providing accurate and efficient solutions for cancer prediction. Looking ahead, the scope for further development is vast, with potential enhancements to enhance accuracy, security, and user experience. Strengthening privacy and security measures are essential aspects to consider. Here, any histopathological images can be given as inputs and optimal result is predicted.vThe integration of AI-based adaptive learning can enhance system accuracy, particularly in challenging environments. Breast cancer prediction systems will continue to be a vital and transformative tool for medical organizations and institutions in the future

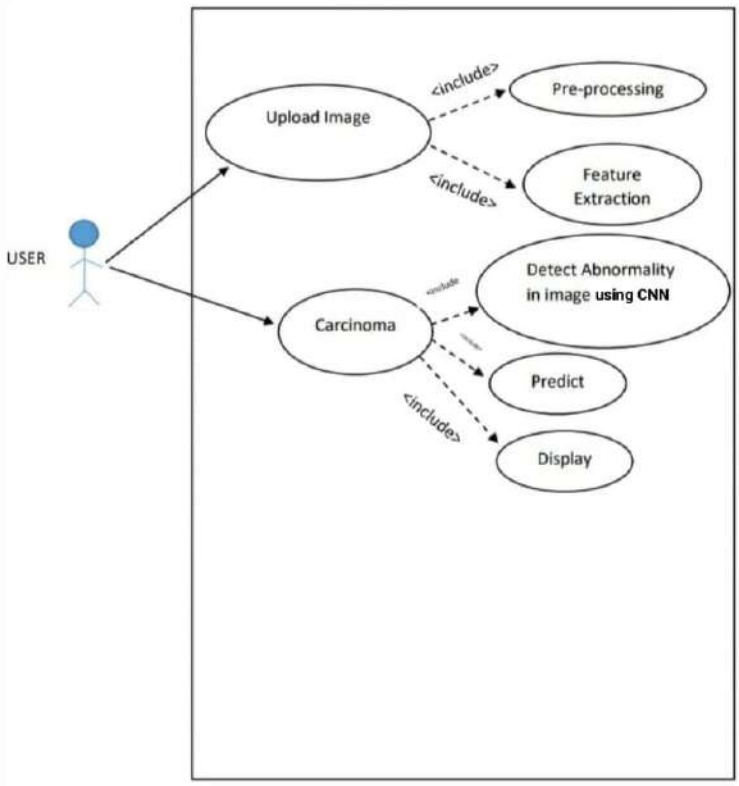
8.2 Future Scope

- Web based platform: Accessible to all the users where they interact with the doctor and results are available online.
- Privacy and Data Protection: Enhancing privacy features and adopting privacy-preserving algorithms will be crucial to safeguarding user data and complying with evolving data protection regulations.
- User Interface and Experience: Investing in user-friendly interfaces and seamless integration with existing systems will encourage greater user acceptance and adoption of the system.
- Real-time Analytics: Developing real-time attendance analytics can provide valuable insights into cancer prediction trends, enabling medical organizations to make data-driven decisions. More image should be trained or the accuracy may vary. Therefore, Several new models should be trained and the more f1 – score giving model should be taken in the future.
- Customization Options: Offering customization options to tailor the system to specific organizational needs and requirements will increase its versatility and adaptability.
- Cross-Platform Compatibility: Ensuring compatibility with various devices and operating systems will make the system accessible and widely applicable in different settings.

Appendix

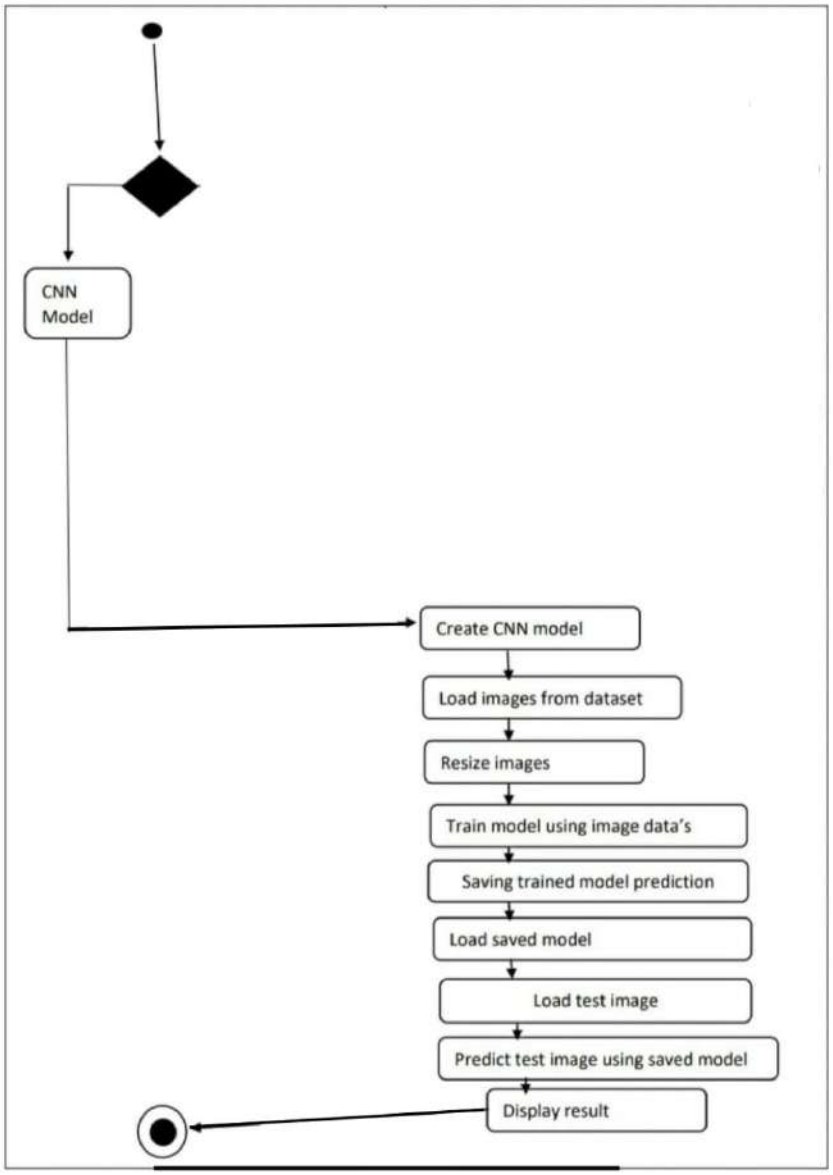
A Usecase Diagram

A.1



B Activity Diagram

B.1



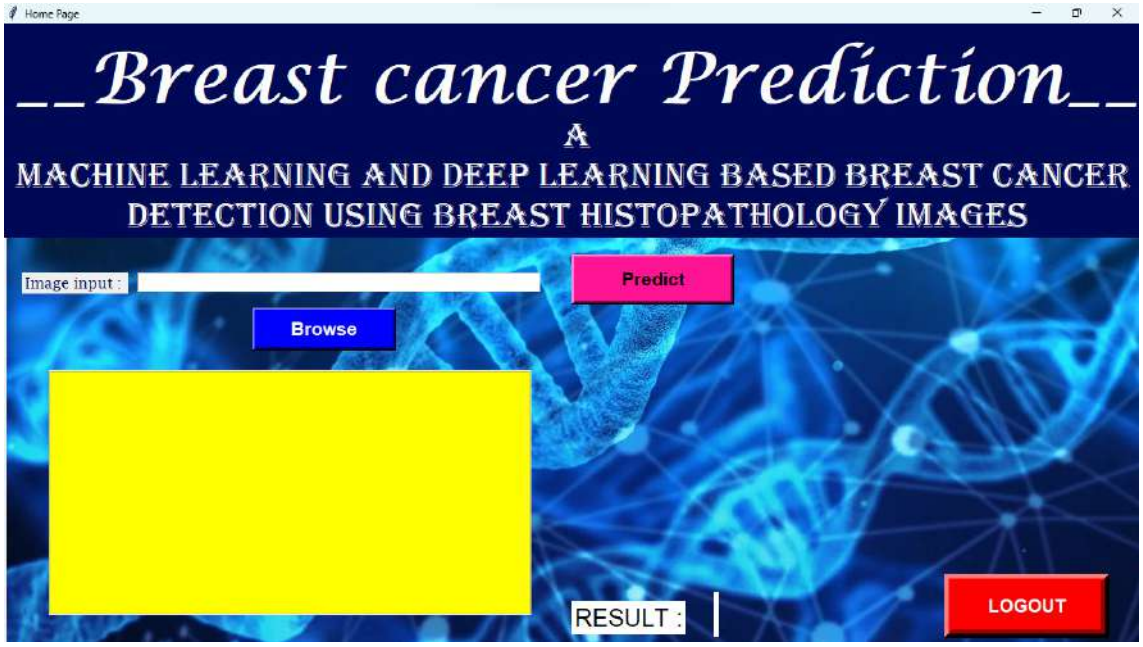
B.2

2. **Breast Cancer**

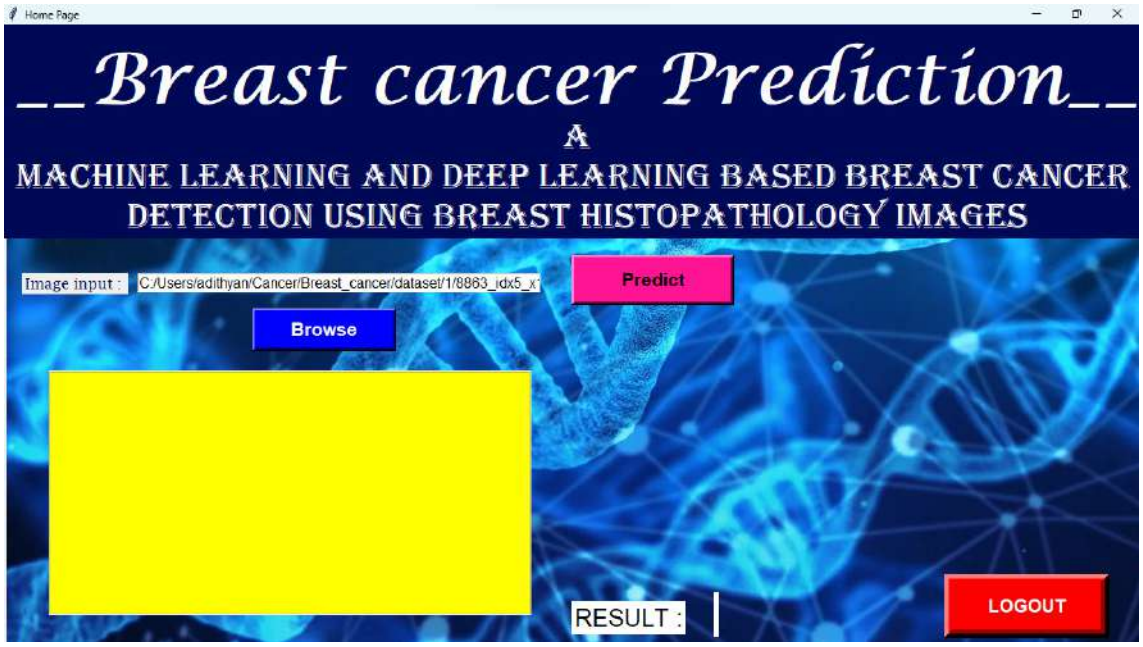
System	**Breast cancer**

C USER INTERFACES

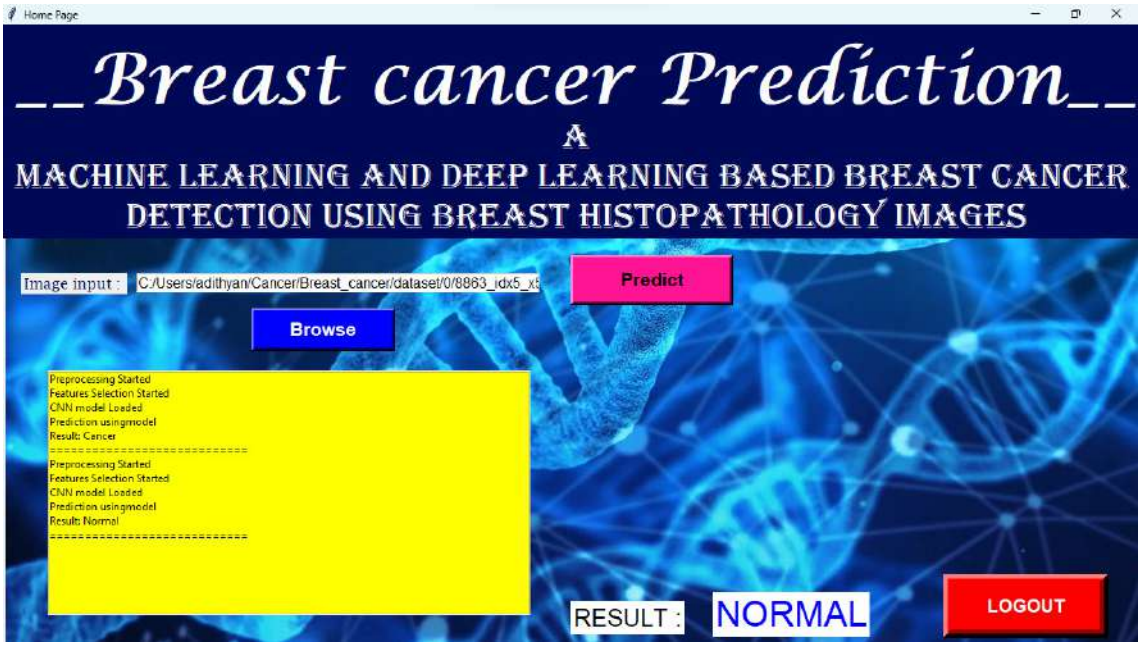
C.1 HOME



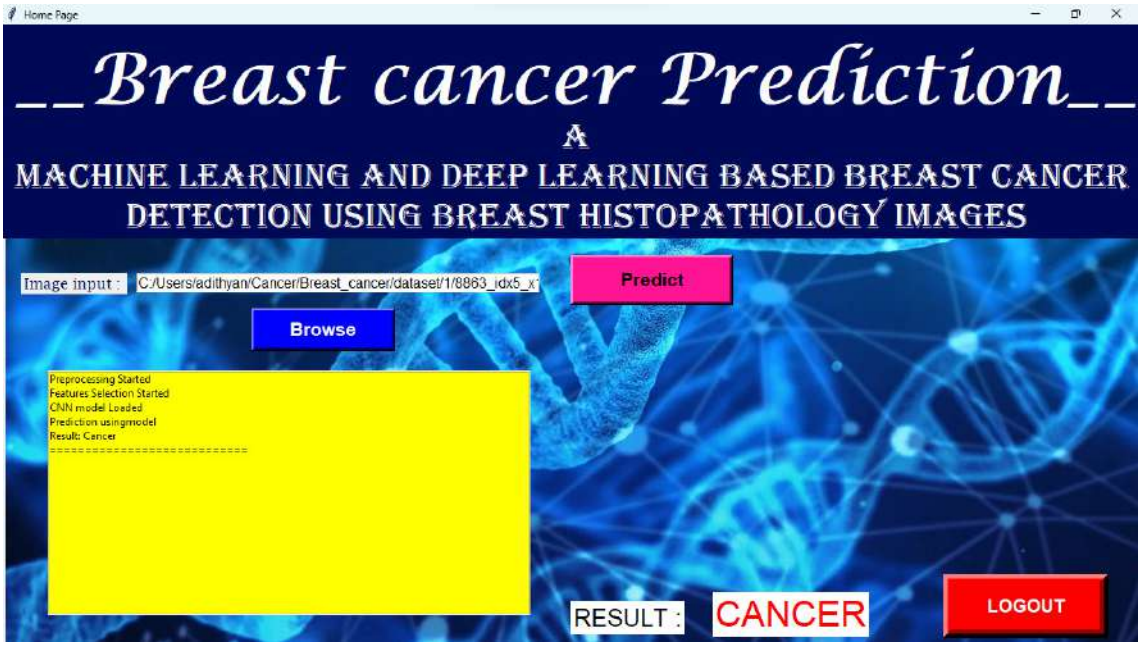
C.2 IMAGE LOCATION



C.3 NORMAL RESULT



C.4 CANCEROUS RESULT



D CODE

preprocess.py

```
import cv2

import numpy as np
import random
from imutils import paths
import os
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
import json
from tensorflow.keras.preprocessing.image import
    img_to_array
from tensorflow.keras.utils import to_categorical
from sklearn import metrics
import pickle

import mahotas as mt
from skimage.filters import gabor

def extract_features(gray):
    textures = mt.features.haralick(gray)
    # cv2.imshow("textures",textures)
    # cv2.waitKey(0)
        # take the mean of it and return it
    ht_mean  = textures.mean(axis=0)
    # print(ht_mean)
    return ht_mean


def build_filters():
        filters = []
        ksize = 31
        for theta in np.arange(0, np.pi, np.pi / 16):
            kern = cv2.getGaborKernel((ksize, ksize),
                4.0, theta, 10.0, 0.5, 0, ktype=cv2.CV_32F
                )
            kern /= 1.5*kern.sum()
            filters.append(kern)
            return filters

def process(img, filters):
        accum = np.zeros_like(img)
```

```
        for kern in filters:
            fimg = cv2.filter2D(img, cv2.CV_8UC3, kern)
            np.maximum(accum, fimg, accum)
            return accum


filters=build_filters()



data = []
label = []
print("[INFO] loading images...")
img_dir=sorted(list(paths.list_images("dataset")))
random.shuffle(img_dir)
print("[INFO]   Preprocessing...")
count=0
for i in img_dir:
    img = cv2.imread(i)
    img=cv2.resize(img, (64,64))
    img1=process(img,filters)
    gray = cv2.cvtColor(img1, cv2.COLOR_BGR2GRAY)
    res2=extract_features(gray)
    # print(res2)
    # print(type(res2))
    # imgdata=img_to_array(res2)
    data.append(res2)
    lb=i.split(os.path.sep)[-2]
    if(lb=="1"):
        label.append(1)
    else:
        label.append(0)
print(len(data))
print(len(label))
```

[breaklines=true]

SvmTrain.py

```
import cv2

import numpy as np
import random
from imutils import paths
import os
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
```

```
import json
from tensorflow.keras.preprocessing.image import
    img_to_array
from tensorflow.keras.utils import to_categorical
from sklearn import metrics
import pickle

import mahotas as mt
from skimage.filters import gabor

def extract_features(gray):
    textures = mt.features.haralick(gray)
    # cv2.imshow("textures",textures)
    # cv2.waitKey(0)
        # take the mean of it and return it
    ht_mean  = textures.mean(axis=0)
    # print(ht_mean)
    return ht_mean


def build_filters():
        filters = []
        ksize = 31
        for theta in np.arange(0, np.pi, np.pi / 16):
            kern = cv2.getGaborKernel((ksize, ksize),
                4.0, theta, 10.0, 0.5, 0, ktype=cv2.CV_32F
                )
            kern /= 1.5*kern.sum()
            filters.append(kern)
            return filters

def process(img, filters):
        accum = np.zeros_like(img)
        for kern in filters:
            fimg = cv2.filter2D(img, cv2.CV_8UC3, kern)
            np.maximum(accum, fimg, accum)
            return accum


filters=build_filters()



data = []
label = []
print("[INFO] loading images...")
```

```
img_dir=sorted(list(paths.list_images("dataset")))
random.shuffle(img_dir)
print("[INFO]   Preprocessing...")
count=0
for i in img_dir:
    img = cv2.imread(i)
    img=cv2.resize(img, (64,64))
    img1=process(img,filters)
    gray = cv2.cvtColor(img1, cv2.COLOR_BGR2GRAY)
    res2=extract_features(gray)
    # print(res2)
    # print(type(res2))
    # imgdata=img_to_array(res2)
    data.append(res2)
    lb=i.split(os.path.sep)[-2]
    if(lb=="1"):
        label.append(1)
    else:
        label.append(0)
print(len(data))
print(len(label))
lenofimage=len(data)
traindata=np.array(data).reshape(lenofimage,-1)
trainlabel=np.array(label)
print(traindata.shape)
print(trainlabel.shape)
X_train, X_test, y_train, y_test = train_test_split(
    traindata, trainlabel, test_size=0.25, random_state
    =42)
print(X_train.shape)
print(y_train.shape)
SVMclassifier = SVC(kernel='linear', random_state=0)
SVMclassifier.fit(X_train, y_train)
# Saving Trained Model
# dbfile=open("SVMmodel","wb")
# pickle.dump(SVMclassifier,dbfile)
# dbfile.close()
y_pred= SVMclassifier.predict(X_test)
print(y_pred)
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))

# 62.38
```

[breaklines=true] **Svmtest.py**

```
import cv2
```

```
import numpy as np
import random
from imutils import paths
import os
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
import json
from tensorflow.keras.preprocessing.image import
    img_to_array
from tensorflow.keras.utils import to_categorical
from sklearn import metrics
import pickle

import mahotas as mt
from skimage.filters import gabor



def extract_features(gray):
    textures = mt.features.haralick(gray)
    # cv2.imshow("textures",textures)
    # cv2.waitKey(0)
        # take the mean of it and return it
    ht_mean  = textures.mean(axis=0)
    # print(ht_mean)
    return ht_mean


def build_filters():
        filters = []
        ksize = 31
        for theta in np.arange(0, np.pi, np.pi / 16):
            kern = cv2.getGaborKernel((ksize, ksize),
                4.0, theta, 10.0, 0.5, 0, ktype=cv2.CV_32F
                )
            kern /= 1.5*kern.sum()
            filters.append(kern)
            return filters

def process(img, filters):
        accum = np.zeros_like(img)
        for kern in filters:
            fimg = cv2.filter2D(img, cv2.CV_8UC3, kern)
            np.maximum(accum, fimg, accum)
```

```
            return accum


# img=cv2.imread("im1.JPG")


filters=build_filters()


img = cv2.imread("31.png")
img=cv2.resize(img, (64,64))
img1=process(img,filters)
gray = cv2.cvtColor(img1, cv2.COLOR_BGR2GRAY)
res2=extract_features(gray)

dbfile=open("SVMmodel","rb")
adamodel=pickle.load(dbfile)
dbfile.close()
# Prediction based on selected model
y_pred = adamodel.predict(res2.reshape(1,-1))
print(y_pred)
```

[breaklines=true] **CNNTrain.py**

```
import cv2

import numpy as np
import random
from imutils import paths
import os
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
import json
from tensorflow.keras.preprocessing.image import
    img_to_array
from tensorflow.keras.utils import to_categorical
from sklearn import metrics
import pickle
import pandas as pd
import numpy as np
import os
from glob import glob
import random
import matplotlib.pylab as plt
from tensorflow.keras.models import model_from_json
```

```
from sklearn.model_selection import train_test_split
import tensorflow as tf
import keras
from keras.utils.np_utils import to_categorical
from keras.models import Sequential
from keras.preprocessing.image import ImageDataGenerator

from tensorflow.keras.optimizers import SGD, RMSprop,
    Adam, Adagrad, Adadelta
from keras.layers import Dense, Dropout, Activation,
    Flatten, BatchNormalization, Conv2D, MaxPool2D,
    MaxPooling2D
from tensorflow.keras.callbacks import EarlyStopping


data = []
label = []
print("[INFO] loading images...")
img_dir=sorted(list(paths.list_images("dataset")))
random.shuffle(img_dir)
print("[INFO]  Preprocessing...")
count=0
for i in img_dir:
    img = cv2.imread(i)
    img=cv2.resize(img, (64,64))
    imgdata=img_to_array(img)
    data.append(imgdata)
    lb=i.split(os.path.sep)[-2]
    if(lb=="1"):
        label.append(1)
    else:
        label.append(0)

traindata=np.array(data)
trainlabel=np.array(label)

X_train, X_test, y_train, y_test = train_test_split(
    traindata, trainlabel, test_size=0.25, random_state
    =42)
print(X_train.shape)
print(y_train.shape)

model = Sequential()

# Convolutional layer and maxpool layer 1
model.add(Conv2D(32,(3,3),activation='relu',input_shape
```

```
    =(64,64,3)))
model.add(MaxPool2D(2,2))

# Convolutional layer and maxpool layer 2
model.add(Conv2D(64,(3,3),activation='relu'))
model.add(MaxPool2D(2,2))

# Convolutional layer and maxpool layer 3
model.add(Conv2D(128,(3,3),activation='relu'))
model.add(MaxPool2D(2,2))

# Convolutional layer and maxpool layer 4
model.add(Conv2D(128,(3,3),activation='relu'))
model.add(MaxPool2D(2,2))

# This layer flattens the resulting image array to 1D
    array
model.add(Flatten())

# Hidden layer with 512 neurons and Rectified Linear Unit
     activation function
model.add(Dense(512,activation='relu'))

# Output layer with single neuron which gives 0 for Cat
    or 1 for Dog
#Here we use sigmoid activation function which makes our
    model output to lie between 0 and 1
model.add(Dense(1,activation='sigmoid'))

# model = Sequential()
# model.add(Conv2D(64, kernel_size=3, activation='relu',
    input_shape=(50,50,3)))
# model.add(Conv2D(32, kernel_size=3, activation='relu'))
# model.add(Flatten())
# model.add(Dense(2, activation="softmax"))
# adam = Adam(learning_rate=0.0001)
model.compile(loss="binary_crossentropy", optimizer="adam
    ", metrics=['accuracy'])

history = model.fit(
    X_train, y_train,
    validation_data=(X_test, y_test),
    epochs= 10,
    batch_size=10
    )
Y_pred = model.predict(X_test)
```

```
print(Y_pred)
#serialize model to JSON
model_json = model.to_json()
with open("CNNmodel.json", "w") as json_file:
        json_file.write(model_json)
# serialize weights to HDF5
model.save_weights("CNNmodelw.h5")
print("Saved model to disk")
#0.9133
```

[breaklines=true]
CNNTest.py

```
from tensorflow.keras.preprocessing.image import
    img_to_array
from tensorflow.keras.models import load_model
import numpy as np
import argparse
import imutils
import cv2
from tensorflow.keras.models import model_from_json


img = cv2.imread("10.png")
img=cv2.resize(img, (64,64))
img = img_to_array(img)
##img = img.reshape(1, 100, 36, 1)

json_file = open('CNNmodel.json', 'r')
loaded_model_json = json_file.read()
json_file.close()
model = model_from_json(loaded_model_json)
# load weights into new model
model.load_weights("CNNmodelw.h5")
print("Loaded model from disk")
img = np.expand_dims(img, axis = 0)
result = model.predict(img)
res1=result[0][0]
if(res1>0.32):
    print("Cancer")
else:
    print("Normal")
print("res=>",result[0][0])
result=np.argmax(result[0][0])
print(result)
```

[breaklines=true] **Resnet_train.py**

```
import numpy as np
import random
import cv2
from glob import glob
import matplotlib.pyplot as plt
from imutils import paths
import os
from PIL import Image
import keras
from tensorflow.keras.preprocessing.image import
    ImageDataGenerator
from tensorflow.keras.applications import ResNet50
from tensorflow.keras.applications.resnet50 import
    preprocess_input
from tensorflow.keras import Model, layers
from tensorflow.keras.models import load_model,
    model_from_json
from tensorflow.keras.preprocessing.image import
    img_to_array
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.models import model_from_json

from sklearn.model_selection import train_test_split



conv_base = ResNet50(
    include_top=False,
    weights='imagenet')

for layer in conv_base.layers:
    layer.trainable = False
x = conv_base.output
x = layers.GlobalAveragePooling2D()(x)
x = layers.Dense(128, activation='relu')(x)
predictions = layers.Dense(2, activation='softmax')(x)
model = Model(conv_base.input, predictions)

model.compile(loss='sparse_categorical_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])


data = []
label = []
```

```
print("[INFO] loading images...")
img_dir=sorted(list(paths.list_images("dataset")))
random.shuffle(img_dir)
print("[INFO]   Preprocessing...")
count=0
for i in img_dir:
    img = cv2.imread(i)
    img=cv2.resize(img,(256,256))
    imgdata=img_to_array(img)
    data.append(imgdata)
    lb=i.split(os.path.sep)[-2]
    if(lb=="0"):
        label.append(0)
    else:
        label.append(1)
print(len(data))
print(len(label))
lenofimage=len(data)
traindata=np.array(data)
trainlabel=np.array(label)
train_labels = to_categorical(trainlabel, num_classes=2)
print(traindata.shape)
print(train_labels.shape)
X_train, X_test, y_train, y_test = train_test_split(
    traindata, trainlabel, test_size=0.25, random_state
    =42)
print(X_train.shape)
print(y_train.shape)
print(X_test.shape)
print(y_test.shape)
```

```
model.fit(X_train, y_train, epochs=3, validation_split
    =0.2, batch_size=5)

model_json = model.to_json()
with open("resnet_model.json", "w") as json_file:
        json_file.write(model_json)
# serialize weights to HDF5
model.save_weights("resnetw_model.h5")
print("[INFO] Saved model to disk")
y=model.predict(X_train)
print(y)
```

[breaklines=true] **Densenettrain.py**

```
import numpy as np
import random
import cv2
from glob import glob
import matplotlib.pyplot as plt
from imutils import paths
import os
from PIL import Image

import keras
from tensorflow.keras.preprocessing.image import
    ImageDataGenerator
from tensorflow.keras.applications.densenet import
    DenseNet121
from keras import Model, layers
from keras.models import load_model, model_from_json
from tensorflow.keras.preprocessing.image import
    img_to_array
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.models import model_from_json
from tensorflow.keras.layers import Dense,
    GlobalAveragePooling2D,Convolution2D,
    BatchNormalization
from tensorflow.keras.layers import Flatten,MaxPooling2D,
    Dropout
from sklearn.model_selection import train_test_split




from tensorflow.keras.layers import Dense, Dropout, Input
    ,Flatten, Conv2D, MaxPool2D, BatchNormalization,
    AveragePooling2D, GlobalAveragePooling2D
from tensorflow.keras.models import Model,Sequential,
    load_model
from tensorflow.keras.optimizers import Adam
def build_densenet():
    densenet = DenseNet121(weights='imagenet',
        include_top=False)

    input = Input(shape=(256, 256, 3))
```

```
    x = Conv2D(3, (3, 3), padding='same')(input)

    x = densenet(x)

    x = GlobalAveragePooling2D()(x)
    x = BatchNormalization()(x)
    x = Dropout(0.5)(x)
    x = Dense(256, activation='relu')(x)
    x = BatchNormalization()(x)
    x = Dropout(0.5)(x)

    # multi output
    output = Dense(15,activation = 'softmax', name='root
        ')(x)


    # model
    model = Model(input,output)

    optimizer = Adam(lr=0.002, beta_1=0.9, beta_2=0.999,
        epsilon=0.1, decay=0.0)
    model.compile(loss='sparse_categorical_crossentropy',
         optimizer=optimizer, metrics=['accuracy'])
    model.summary()

    return model

model = build_densenet()



data = []
label = []
print("[INFO] loading images...")
img_dir=sorted(list(paths.list_images("dataset")))
random.shuffle(img_dir)
print("[INFO]   Preprocessing...")
count=0
for i in img_dir:
    img = cv2.imread(i)
    img=cv2.resize(img,(256,256))
    imgdata=img_to_array(img)
    data.append(imgdata)
    lb=i.split(os.path.sep)[-2]
    if(lb=="0"):
        label.append(0)
```

```
    else:
        label.append(1)
print(len(data))
print(len(label))
lenofimage=len(data)
traindata=np.array(data)
trainlabel=np.array(label)
train_labels = to_categorical(trainlabel, num_classes=2)
print(traindata.shape)
print(train_labels.shape)
X_train, X_test, y_train, y_test = train_test_split(
    traindata, trainlabel, test_size=0.25, random_state
    =42)
print(X_train.shape)
print(y_train.shape)
print(X_test.shape)
print(y_test.shape)


model.fit(X_train, y_train, epochs=10, validation_split
    =0.2, batch_size=5,verbose=1)

# model_json = model.to_json()
# with open("densenet_model.json", "w") as json_file:
#         json_file.write(model_json)
# # serialize weights to HDF5
# model.save_weights("densenetw_model.h5")
# print("[INFO] Saved model to disk")
```

[breaklines=true] **GUI_new.py**

```
from tkinter import *
import time
import re
#Import scikit-learn metrics module for accuracy
    calculation
import pickle
from PIL import Image, ImageTk
import cv2
from tkinter.filedialog import askopenfile
from tensorflow.keras.preprocessing.image import
    img_to_array
from tensorflow.keras.models import load_model
import numpy as np
import argparse
import imutils
```

```
import cv2
from tensorflow.keras.models import model_from_json
from tensorflow.keras.preprocessing.image import
    img_to_array
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.applications.vgg16 import
    preprocess_input
from tensorflow.keras.models import model_from_json

json_file = open('resnet_model.json', 'r')
loaded_model_json = json_file.read()
json_file.close()
modelres = model_from_json(loaded_model_json)
# load weights into new model
modelres.load_weights("resnetw_model.h5")
print("Loaded Resnet model from disk")

json_file = open('densenet_model.json', 'r')
loaded_model_json = json_file.read()
json_file.close()
modeldense = model_from_json(loaded_model_json)
# load weights into new model
modeldense.load_weights("densenet.h5")

print("Loaded Sensenet model from disk")

json_file = open('CNNmodel.json', 'r')
loaded_model_json = json_file.read()
json_file.close()
cnnmodel = model_from_json(loaded_model_json)
# load weights into new model
cnnmodel.load_weights("CNNmodelw.h5")
print("Loaded model from disk")

def pp(a):
    global mylist
    mylist.insert(END, a)



def predict(val):
    print(val)
    img = cv2.imread(val)
    imgr=cv2.resize(img, (64,64))
    imgarr = img_to_array(imgr)
    ##img = img.reshape(1, 100, 36, 1)
```

```
imgarr = np.expand_dims(imgarr, axis = 0)
result = cnnmodel.predict(imgarr)
res1=result[0][0]
print(res1)

if(res1>0.32):
    cnnres=1
else:
    cnnres=0
print("cnnpred==>",cnnres)
imgres=cv2.resize(img,(256,256))
imgdata=img_to_array(imgres)
print(imgdata)
# print(type(imgdata))
# print(imgdata.shape)

train_ds= np.expand_dims(imgdata, axis=0)
# print(train_ds.shape.)

yres=modelres.predict(train_ds)
print(yres)
resres=np.argmax(yres[0])
print("respred==>",resres)

ydense=modeldense.predict(train_ds)
print(ydense)
denseres=np.argmax(ydense[0])
print("densepred==>",denseres)

reslist=[cnnres,resres,denseres]

print("result list==>",reslist)
finalres=max(reslist)
if(finalres==1):
    resc="Cancer"
    print("Cancer")
    root.after(3100, lambda :shrslt.config(text="
        CANCER",fg="red"))

else:
    resc="Normal"
    print("Normal")
    root.after(3100, lambda :shrslt.config(text="
        NORMAL",fg="blue"))
```

```
    root.after(500, lambda : pp("Preprocessing Started ")
        )
    root.after(2000, lambda : pp("Features Selection
        Started "))
    root.after(2200, lambda : pp("CNN model Loaded"))
    root.after(2400, lambda : pp("Prediction usingmodel
        "))
    root.after(2600, lambda : pp("Result: "+resc))
    root.after(2800, lambda : pp
        ("=========================================="))


def browseim():
    global cimg,shrslt,E1
    path = askopenfile()
    n=path.name
    print(path)
    E1.delete(0,"end")
    E1.insert(0, n)

#def upload_file():
 #    global cimg,shrslt,E1
 #    root=Tk()
 #    f_types=[('Png Files','*.png')]
 #    filename=filedialog.askopenfilename(filetypes=
     f_types)
 #   cimg=ImageTk.PhotoImage(file=filename)

def userHome():
    global root, mylist,shrslt,E1
    root = Tk()
    root.geometry("1400x625+0+0")
    root.title("Home Page")

    image = Image.open("dna.png")
    image = image.resize((1500, 725), Image.ANTIALIAS)
    pic = ImageTk.PhotoImage(image)
    lbl_reg=Label(root,image=pic,anchor=CENTER)
    lbl_reg.place(x=0,y=0)

    #----------------------INFO TOP----------------
    lblinfo = Label(root, font=( 'lucida calligraphy'
        ,61, 'bold' ),text="__Breast cancer Prediction__",
        fg="white",bg="#000955",bd=10,anchor='w')
```

```
lblinfo.place(x=0,y=0)

lblinfo2 = Label(root, font=( 'Algerian' ,31 ),text="
    A\n Machine learning and Deep learning based
    Breast cancer \n   detection using Breast
    Histopathology Images   ",fg="white",bg="#000955",
    anchor='w')
lblinfo2.place(x=0,y=100)
lblinfo3 = Label(root, font=( 'Lucida fax' ,0 ),text
    ="Image input : ",fg="#000955",anchor='w')
lblinfo3.place(x=20,y=280)
E1 = Entry(root,width=50,font="Will&Grace")
E1.place(x=150,y=280)
mylist = Listbox(root,width=90, height=17,bg="yellow
    ")

mylist.place( x = 50, y = 390 )
btntrn=Button(root,padx=10,pady=2, bd=4 ,fg="white",
    font=('ariel' ,16,'bold'),width=10, text="Browse",
     bg="blue",command=lambda:browseim())
btntrn.place(x=280, y=320)
# btntrn=Button(root,padx=10,pady=2, bd=4 ,fg="white",
   font=('ariel' ,16,'bold'),width=10, text="image",
   bg="red",command=lambda:upload_file())
#btntrn.place(x=700, y=400)
btnhlp=Button(root,padx=40,pady=6, bd=4 ,fg="black",
    font=('ariel' ,16,'bold'),width=7, text="Predict",
     bg="deep pink",command=lambda:predict(E1.get()))
btnhlp.place(x=640, y=260)



rslt = Label(root, font=( 'aria' ,20, ),text="RESULT
    :",fg="black",bg="white",anchor=W)
rslt.place(x=640,y=650)
shrslt = Label(root, font=( 'aria' ,30, ),text="",fg
    ="blue",bg="white",anchor=W)
shrslt.place(x=800,y=640)

def qexit():
    root.destroy()

btnexit=Button(root,padx=16,pady=8, bd=10 ,fg="white
    ",font=('ariel' ,16,'bold'),width=10, text="LOGOUT
    ", bg="red",command=qexit)
btnexit.place(x=1060, y=620)
```

```
    root.mainloop()

userHome()
```

[breaklines=true]

AIR QUALITY INDEX

PROJECT REPORT

Submitted By

Savitha Sebastian
Reg. No. CCAVBCA012

For the award of the Degree of

Bachelor of Computer Application

(BCA)

(University of Calicut)

under the guidance of

Ms. Viji Viswanathan

Assistant Professor

BACHELOR OF COMPUTER APPLICATION
DEPARTMENT OF COMPUTER SCIENCE
CHRIST COLLEGE
IRINJALAKUDA, KERALA
2021-2024

DEPARTMENT OF COMPUTER SCIENCE

CHRIST COLLEGE (AUTONOMOUS)

IRINJALAKUDA

CERTIFICATE

This is to certify that the project report entitled ***"Air Quality Index"*** *is a bonfied record of the project work done by* ***Savitha Sebastian*** *in partial fulfillment of the requirement for the sixth semester of* ***Bachelor of Computer Application*** *in Department of Computer Science of* **CHRIST COLLEGE (AUTONOMOUS) IRINJALAKUDA**

Ms. Viji Viswanathan
Assistant Professor,CS
Internal Guide

Ms. Sini Thomas
Head of the Department
Computer Science

EXTERNAL EXAMINER

INTERNAL EXAMINER

DECLARATION

We hereby declare that this project work **"AIR QUALITY INDEX"** submitted to Christ College (Autonomous) Irinjalakuda, affiliated to Calicut University in partial fulfillment of the requirement for the award of the Bachelor of Computer Application, is a record of original work done by us, under the guidance of Ms. VIJI VISWANATHAN, Department of computer Science.

Place: Irinjalakuda SAVITHA SEBASTIAN

ACKNOWLEDGEMENT

First and foremost we like to thank Lord almighty for his providence and for being the guiding light throughout the project. We wish to express my sincere gratitude to our beloved Department head for giving me all the facilities for our project. We take this opportunity to express my gratitude to the class teacher Ms. SOUMYA P.S and head of the department Ms. SINI THOMAS who has been supported us throughout the course of this project. We are thankful for her aspiring guidance and valuable advice during the project work. We express my sincere thanks to my project guide Ms. VIJI VISWANATHAN for supporting and guiding throughout the project.We would take this opportunity to specially thank all other faculty members for their constant and continuous motivation. Finally we would like to thank my family and friends for giving valuable advice and moral support throughout our project.

ABSTRACT

Air Quality Index is a innovative web application that takes on paramount importance as it addresses the need for timely and accurate information about air quality. The web application is enriched with a login form ,a registratin form and a prediction window where we enter the prediction attributes.This application gives accurate air qualty and a graphical visualization for the same. All these features make this website more adaptable and user-friendly.

Contents

Chapter 1

1 Introduction

Air quality is a critical component of environmental health management and its significance cannot be overstated.Air pollution has become a most pressing concern and it has affected not only the environment but also the health of billions of people. Air quality indexing takes on importance as it addresses the timely and accurate information about the air quality.The purpose of the system is to create a robust and reliable system that not only monitors air quality but also predicts its future trends.This step represents a critial step forward in environmental management and how we safeguard the air we breathe.

.

1.1 Overview

The core objective of the project is to develop an Air quality indexing and prediction system that harness the power of data and predictive analytics to provide real-time insights to air quality. The development of this system can lead to a healthier and more sustainable future.

Chapter 2

2 System Analysis

2.1 Purpose

The purpose of this project is to develop an Air quality indexing and prediction system that harness the power of data and predictive analytics to provide real-time insights to air quality.

2.1.1 Existing System

Presently all the processes gives very simple outlook. It is provided with login and registeration pages. The web application is developed using HTML,CSS and Bootstrap in the front end. Backend is developed using Python and the framework being django. Limitations of existing system :No Graph provided,No login and registeration pages provided,

2.1.2 Proposed System

As a software developer, we are expected to design and develop any program that works efficiently without any delays.For Convinence we provider a login and registeration pages to make it a web application type project . And with the addition of a Graph to make the project more understandable to the users.

2.2 Problem definition

To know what the problem is and what the needs are before developing it. Here we design a web application for detecting the air quality

2.3 FEASIBILITY STUDY

After the problem is clearly understood and solutions proposed, the next step is to conduct the feasibility study. Feasibility study is defined as evaluation or analysis of the potential impact of the proposed project or program. The objective is to determine whether the proposed system is feasible. There are three aspects of feasibility study to which the proposed system is subjected as discussed below.

2.3.1 Technical Feasibility

Technical feasibility assess whether the current technical resources are sufficient for the new system. We can upgrade the level of technology for supporting our web application. We check whether the proposed system can be implemented in the present system without supporting the existing hardware.

2.3.2 Economical Feasibility

Economic feasibility determines whether the time and money are available to develop the system. There is no additional hardware used to develop the site, it is inexpensive to build.

2.3.3 Operational Feasibility

Operational feasibility determines if the human resources are available to operate the system once it has been installed. No extra training is needed to use this system. Anyone who has knowledge in internet in english language can easily use the system. The resources that are required to implement or install are already available with the organization.

Chapter 3

3 Software Requirement Specification

3.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the Project Air quality detection system. It illustrate the purpose and complete description for the development of the system. It explain system constraints, interface and interactions with other external applications. This document is primarily intended to harness the power of data and predictive analytics to provide real-time insights to air quality

3.2 Scope

Our project has made it easier to detect the air quality in real-time . We can get all information about the quality of air wheather it is good or moderate or poor etc with a suitable graph included .

3.3 Overall Description

This section give an overview of our web application is to create a robust and reliable system that not only monitors air quality but also predicts its future trends.This step represents a critial step forward in environmental management and how we safeguard the air we breathe.

3.3.1 Product Perspective

The product perspective of this Air Quality Detection System involves designing and integrating these components to provide accurate,timely and actionable information to the users abut the air they breathe.

3.3.2 Product Functionality

This system can empower users to make informed decesion about their health and well-being in relation to air pollution .

3.3.3 Users and Characteristics

Each user group may have different needs,preferences and levels of technical expertise,so the Air Quality Detection System should be designed with user-centric features and interfaces to accomodate the diverse characterstics effectively.

3.4 Specific Requirements

3.4.1 Hardware Requirements

- System: IBM-Compatible PC
- Processor: Pentium IV or above
- Speed: Above 1GHz
- RAM capacity: 512 MB
- Hard Dsk drive: 40 GB
- Keyboard: Standard
- Mouse: Standard
- Monitor: SVGA Color

3.4.2 Software Requirements

- Operating System:Windows or ubuntu
- Languages used: Python Django
- Data Collection : CSV
- Technologies used: HTML,CSS,Bootsrap

3.5 Functional Requirements

It contains three main modules.

- 1.Login Form
- 2.Registration form
- 3. Prediction window

login Form

The login page is for the logging in the necessary credentials into the project for accessing into the main index page of the project.

Registration Form

In case if the user doesn't have an account he/she might not be able to log in.Therefore a registration form is created to enter the provided details into the form and after registering the user cn go back to the login page to log in the necessary credentials.

Prediction Window

Once logging in a prediction window is displayed with the necessary pollutant information in the air and by typing can give the accurate air quality information.

3.6 Non Functional Requirements

Non-functional requirements define the overall qualities or attributes of the resulting system. Non-functional system place restrictions on the product being developed, the developed process, and specify external constraints that the product must meet. Examples of non-functional requirements include safety, security, usability, reliability and performance requirements. Project management issues (costs, time and schedule) are often considered as non-functional requirements. The principal non - functional constraints which are relevant to critical systems :

- performance
- security
- safety
- usability

Performance

Performance requirements concern the speed of operation of a system. Types of performance requirements :

- Response requirements (how quickly the system reacts to a user input).
- Throughput requirements (how much the system can accomplish within a specified amount of time).
- Availability requirements (is the system available for service when requested by end users). The speed of operation of this system is adequate for the requirements.

Reliability

- Reliability is the ability of a system to perform its required function under stated conditions for a specified period of time.
- Constraints on the runtime behavior of the system. This system is reliable because its functionalities can be done on the required conditions.

Safety

Safety requirements are not required which exclude unsafe situation from the possible solution of the system.

Usability

Usability is the ease with which a user can learn to operate, prepare inputs for, and interpret outputs of system or components. Usability requirements include :

- information error messages.
- well-formed user interfaces.

3.7 Interface Requirements

3.7.1 Hardware interfaces

The system must run over the internet, all the hardware shall require to connect internet will be hardware interface for the system. As for example modem, WAN-LAN.

3.7.2 Software interfaces

Software interface required for the working of the project is the appropriate operating system.

3.7.3 Communication interfaces

The user of the site communicate using the network connectivity and the data set is accessed

3.8 Security Requirements

- User accesses only their account.
- Validation of input is handled.
- This application containing the computer systems is physically secured against arms or surreptitious entry by intruders.
- Users must be authorized carefully to reduce changes of any such user giving access to an intruder in exchange for a bribe or other favour.

3.9 Platform Used

Windows 10 is a major version of the Microsoft Windows operating system that was released on July 29, 2015.It is built on the Windows NT kernel and follows windows 8.Part of the reason Microsoft decided to name the 2015 release "Windows 10"(and skipped "windows 9")is because the operating system is designed to be a new direction for Microsoft. One of the primary aims of windows 10 is to Unify the windows experience across multiple devices, such desktop computers, tablets, and smartphones. As part of this effort, Microsoft developed Windows10 Mobile alongside Windows 10 to replaces Windows Phone

- Microsoft's previous mobile OS. Windows 10 also integrates other Microsoft services, such as Xbox Live and the Cortana voice recognition assistant.

3.10 Technologies Used

Python Django

Django is a high level Python web framework that enables rapid developement of secure,scalable web applications,emphasizing clean pragmatic design and the principle of "don't repeat yourself"(DRY).It includes built-in features for authentication,database modeling,URL routing,templating and more facilitating efficient development and maintenance of complex web projects.

CSV

Comma-separated values (CSV) is a text file format that uses commas to separate values. A CSV file stores tabular data (numbers and text) in plain text, where each line of the file typically represents one data record. Each record consists of the same number of fields, and these are separated by commas in the CSV file.

HTML

HTML, or Hypertext Markup Language, is a markup language for the web that defines the structure of web pages.It is one of the most basic building blocks of every website, so it's crucial to learn if you want to have a career in web development.This structure alone is not enough to make a web page look good and interactive. So you'll use assisted technologies such as CSS and JavaScript etc to make your HTML beautiful and add interactivity, respectively.

CSS

CSS stands for Cascading Style Sheets.It describes how HTML elements are to be displayed on screen, paper, or in other media. CSS saves a lot of work. It can control the layout of multiple web pages all at once. External stylesheets are stored in CSS files.

Bootstrap

Bootstrap is a free, open source front-end development framework for the creation of websites and web apps. Designed to enable responsive development of mobile-first websites, Bootstrap provides a collection of syntax for template designs.

Chapter 4

4 Design Document

4.1 Purpose

The Air Quality Index is a web application. The main purpose of this web application is to develop an Air quality indexing and prediction system that harness the power of data and predictive analytics to provide real-time insights to air quality.The benefit of this web application is very high because the air pollution has become a most pressing concern and it has affected not only the environment but also the health of billions of people.This web application is easy to use and aesthetic in appearance.The user can register and view the details of the air quality.The web application is easy to handle registration and predicting the accurate air quality. Purpose of this document is to give the detailed description of the architecture and system design for the software.The system design document shows the software will be structured to satisfy the requirement identified in the software requirement specification.It is a translation of requirements into a description of the software structure,software component,interfaces and data necessary for the implementation phase.In a complete software design document,each requirements must be traceable to one or more design entities.

4.2 Scope

The Air Quality Index is a web application which helps in predicting the accurate air quality an .Which also helps in leading to a more healthier and more sustainable future .

4.3 Overview

The purpose of this document is to help the reader to visualize the solution to the project presented.This document verifies how the design meet the requirement stimulated in the SRS document through design viewpoints.The design viewpoints will cover all design elements presented before. This document will provide a direct approach to the development of this project hence reducing feature creep and ponitedly determine the quality of the design.

4.4 Data Design

Database are the storehouses of data used in the software systems.The data is stored in tables inside the database.Several tables are created for the manipulation of the data for the system.Two essential settings for a database are

- Primary Key- The field that is unique for all the record occurrence.
- Foreign Key- The field used to set relation between tables.

Normalization

- Normalization is the process of organizing the data in the database.
- Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate the undesirable characteristics like Insertion, Update and Deletion Anomalies.
- Normalization divides the larger table into the smaller table and links them using relationship.
- The normal form is used to reduce redundancy from the database table.

Tables

Login Form

Name	DataType	Constraints	Description
Username	varchar(45)	Primarykey	to enter name of user
Password	text	Notnull	to set password

Registration Form

Name	DataType	Constraints	Description
FirstName	varchar(50)	Primarykey	To enter first name of the person
LastName	varchar(45)	Notnull	To enter last name of the person
Username	varchar(50)	Notnull	Name of the users
Email	varchar(100)	Notnull	Email of the users
password	text	Notnul	Password of users
Confirm Password	text	Notnull	Confirm the Password of users

Chapter 5

5 Development of the System

This website can be decomposed into a number of submodules. The submodules of web application are Login Form, Registration Form and Prediction Window. Each sub module have specific objectives,to avoid unwanted coupling between modules and to increase cohesion, modules are again decomposed into submodules.

Chapter 6

6 System Testing

Testing is the penultimate step of software development.An elaborate testing of the data is prepared and the system is using the test data,while doing testing,errors are noted and correction is made.The users are trained to operate the developed system.Both hardware and software securities are made to run the developed system successfully.System testing is aimed at ensuring the system works accurately before the live operation commences.

6.1 Test Plan

6.1.1 Scope

This test plan will cover the following testing activities as identified in the testing strategy.

- White Box Testing

White box testing is a testing technique, that examines the program structure and derives test data from the program logic/code. In white box testing, the UI is bypassed. Inputs and outputs of ratings are tested directly at the code level and the results are compared against specifications. It Reveals errors in "hidden" code.

- Black Box Testing

Black - box testing is a method of software testing that examines the functionality of an application based on the specifications. Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end - user would.

- Unit Testing

The module includes four main components to undergo unit testing. It checks the syntax error, logic error and validity of the program correctness. This test will be performed by the developers.

- Integration Testing

After all components pass the unit testing, they get to be tested whether they work correctly when they are running concurrently and communicating to each other. Integration tests exercise an entire subsystem and ensure that a set of components play nicely together.

- Internal Data Testing

We will test the validity of the data before it enters the database to avoid any problems that may face us in the database. We will test the encryption of the personal information of all the users along with the user names and passwords to ensure maximum security of the users privacy.

6.1.2 Software risk issues

In this section, the plan is to test the risk involved in critical issues such as: Difficult to run on visual studio due to large loading time. Some inherent software risks such as complexity exist; also these issues need to be identified. Proper network connection Working of Mysql database

6.1.3 Features to be tested

- Test whether correct user name and password allows you to login.
- Test whether invalid user name and password prevents you from login.
- Test whether there is any connection problem in the Server.
- Test whether the student and admin details are entered correctly.
- Test whether invalid data entry allows saving data successfully.
- Test whether all pages are loaded correctly.
- Test whether watermark is embedded and extracted properly.
- Test whether the provided security works correctly.

6.2 Test consolidation

6.2.1 Test item

The items or features to be tested in the test cases are included in the document. Each and every user input is tested. The present condition of the system is tested. It is checked to make sure that environment is ready for the application to work.

6.2.2 Input specifications

These are the inputs required to execute the test case. All required inputs including data elements and values are given below.

Data	Values
Email	A valid email ID
Password	Combination of characters and numbers
Confirm Password	Combination of characters and numbers

Chapter 7

7 System Implementation and Maintenance

7.1 Implementation

System implementation is the conversion of new system into an operating one which involves creating compatible files, training clients and installing hardware. User training is crucial for minimizing resistance to change and giving chance to prove its worth. Training aids user friendly manuals and healthy screens provide the user with a good start. Software maintenance follows conversion to the extent that changes are necessary to maintain satisfactory operations relative to changes in the user's environment. Maintenance often includes minor enhancements or corrections to the problem that surface late in the systems operations. In the implementation phase, the team builds the components either from scratch or by composition. Given the architecture document meant from the design phase and the requirement document from the analysis phase, the team should build exactly what has been requested, though there is still room for innovation and flexibility. For example, a component may be narrowly designed for this particular system, or the component may be made more general to satisfy a reusability

- Careful planning
- Investigation of system and constraints
- Design the methods to achieve changeover.
- Training the staff in the changed phase.
- Evaluation of change over method.
- The method of implementation and time scale to be adopted are found out initially.

7.2 Maintenance

This phase occurs as a result of deploying the whole system at the end users organization. They will perform the beta testing at the end users and inform to the developers about any needed modification to the application. The customer records all the problems that are encountered during the beta testing and reports these to the developer at regular intervals.

7.2.1 Corrective Maintenance

Even with the best quality assurance activities, it is likely that the customer will uncover defects in the software. Corrective maintenance changes the software to correct the defects. Corrective Maintenance activity may consist of repair,

restoration or replacement of equipment. This activity will be result of a regular inspection, which identifies the failure in time for corrective maintenance to be planned and scheduled, then performed during a routine maintenance shutdown.

7.2.2 Adaptive Maintenance

Over time, the original environment(CPU, operating system, business rules, external product characteristics) for which the software was developed is likely to change. Adaptive maintenance results in modification to the software to accommodate changes to its external environment.

7.2.3 Enhanced Maintenance

As software is used, the customer/user will recognize additional functions that will provide the benefit. Perfect maintenance extends the software beyond its original functional requirements.

7.2.4 Preventive Maintenance

Computer software deteriorates due to change, and because of this preventive maintenance often called software re-engineering, must be conducted to enable the software to serve the needs of its end users. Preventive maintenance makes changes to computer programs so that they can be more easily corrected, adapted and enhanced.

Chapter 8

8 Conclusion and Future Scope

8.1 Conclusion

In conclusion, our "Air Quality Indexing and Prediction" project represents a significant advancement in the field of environmental health and public well-being. This project's primary objective was to develop a comprehensive system that enables accurate air quality prediction, real-time assessment, and informed decision-making.

8.2 Future Scope

- Our "Air Quality Indexing and Prediction" project has laid a strong foundation for future enhancements and expansions. As technology continues to evolve and environmental challenges persist, there are several exciting possibilities for further development and improvement.

Appendix

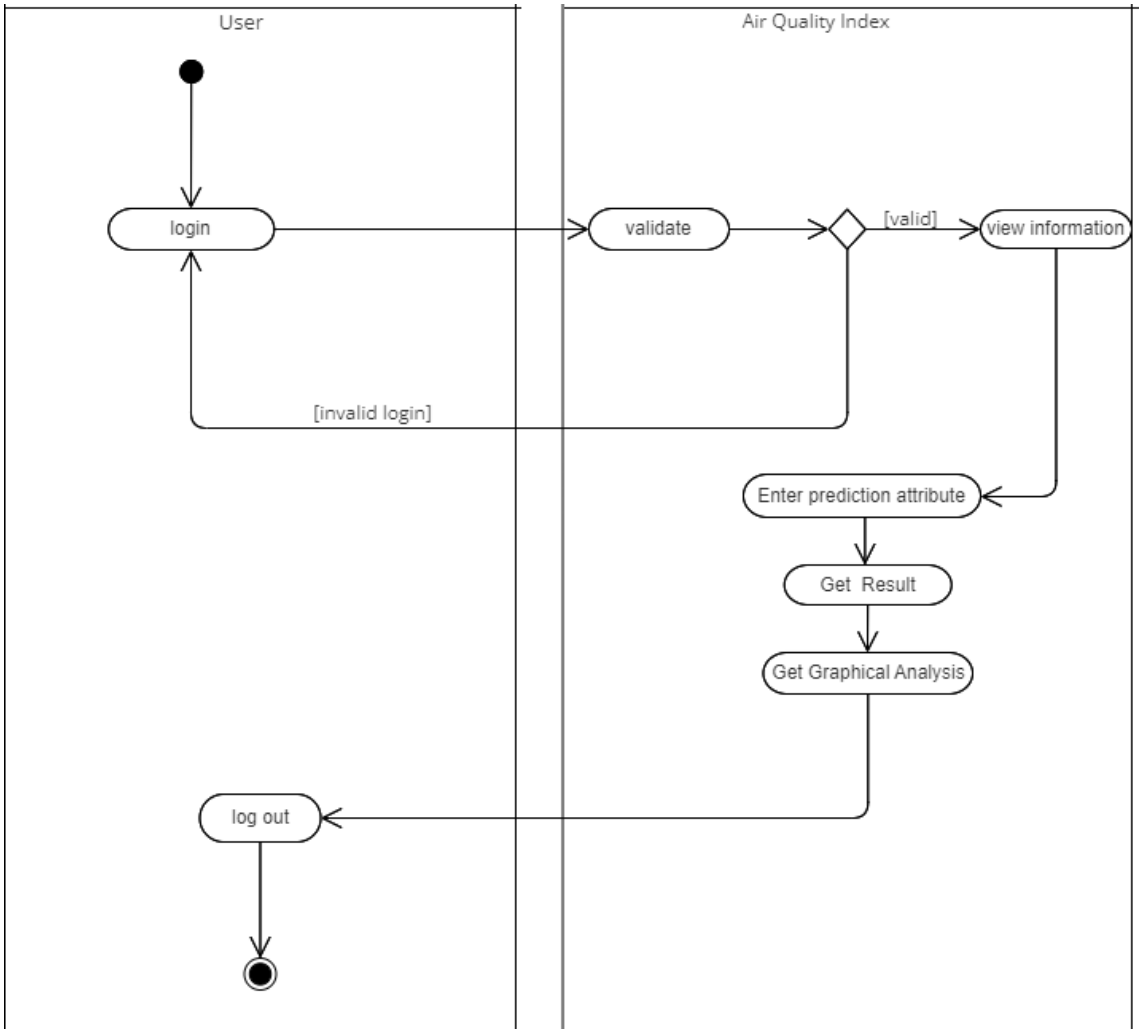
A Activity Diagram

Activity diagrams show the flow of one activity to another within a system or process. Even complex systems can be visualized using activity diagrams. They are used within organizations to model customer journeys, to show the process of receiving an order through shipping to the customer, and to model sales processes.

The Uses of Activity Diagram :

- We describe what causes a particular event using an activity diagram.
- An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed.
- They are used in business and process modeling where their primary use is to depict the dynamic aspects of a system.
- Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques.
- It does not show any message flow from one activity to another. Activity diagram is sometimes considered as the flowchart.

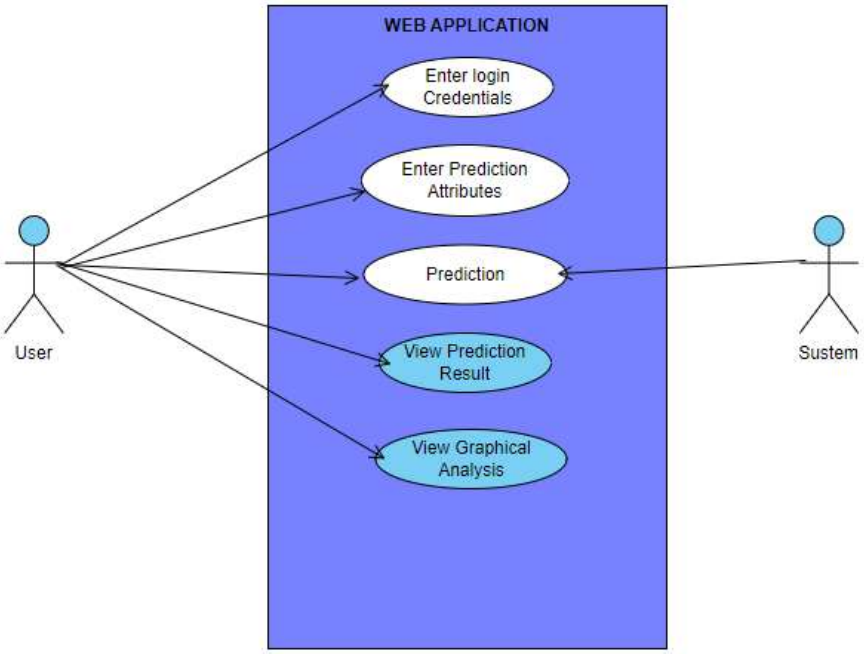
Activity Diagram for Air Quality Index Using Machine Learning



USE CASE DIAGRAM

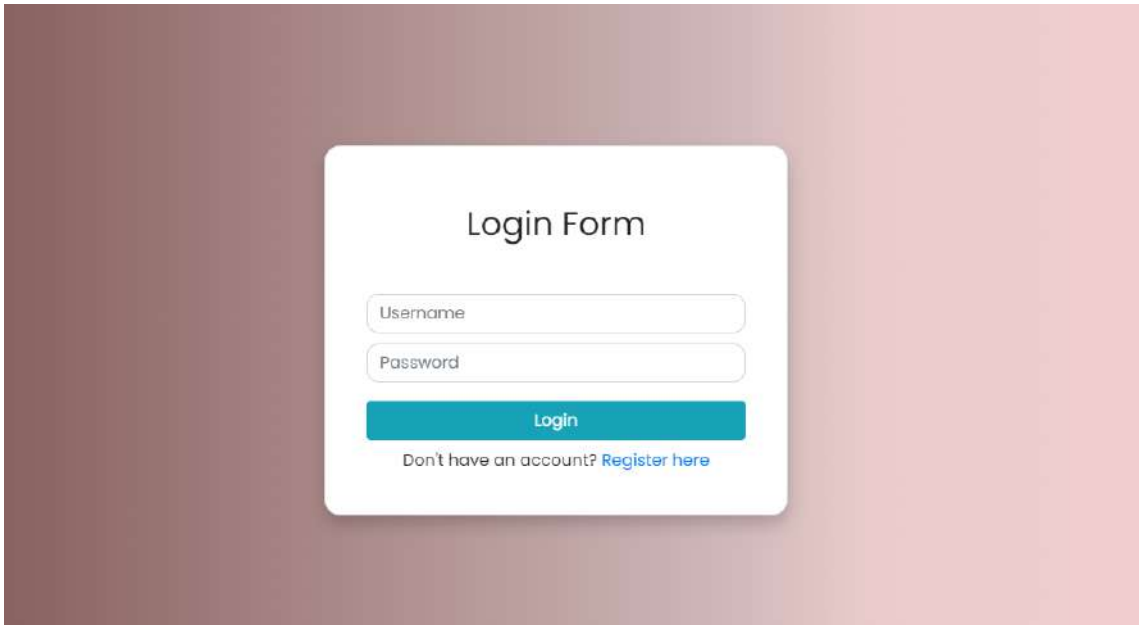
A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.

USE CASE DIAGRAM FOR AIR QUALITY INDEX

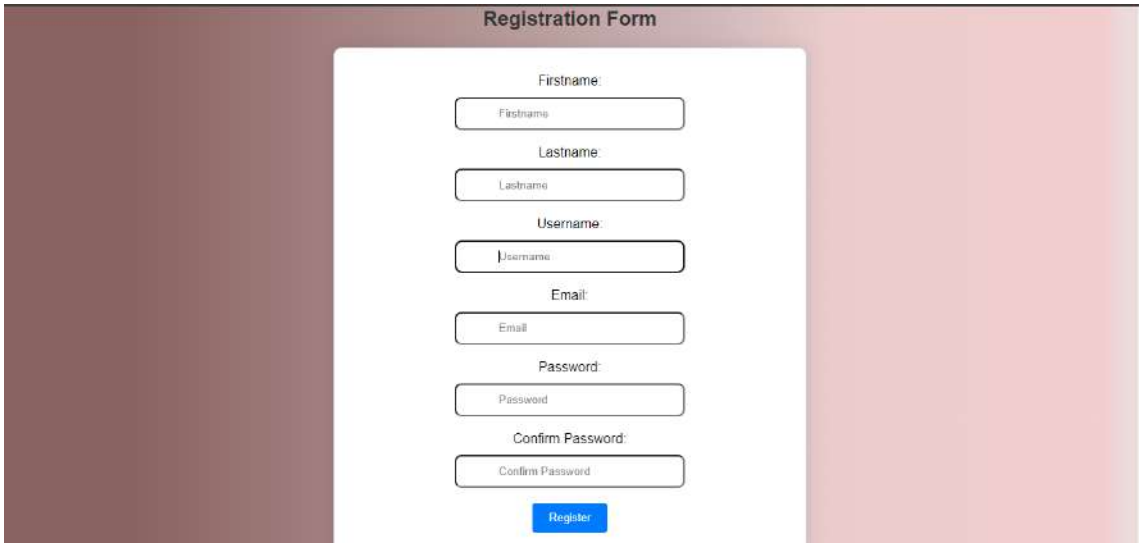


B USER INTERFACES

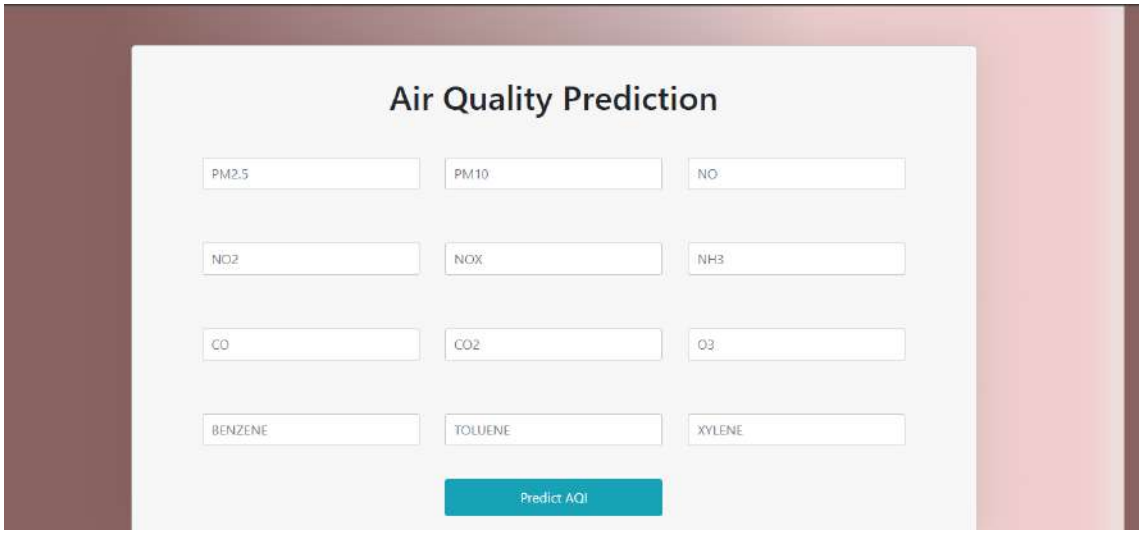
B.1 LOGIN FORM



B.2 REGISTRATON FORM



B.3 PREDICTION WINDOW



C CODE

Login page

```
[breaklines=true]
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Air Quality Detection System - Login</title>
  <link rel="stylesheet"
      href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
  <style>
    @import url
        ('https://fonts.googleapis.com/css2?family=Agdasima&family=Poppins&display=swap');
body {
    font-family: 'Poppins', sans-serif;
    margin: 0;
    padding: 0;
    background: rgb(137,98,97);
    background: linear-gradient(90deg, rgba(137,98,97,1) 11
%, rgba(155,120,119,1) 24%, rgba(169,137,136,1) 34
%, rgba(180,150,150,1) 42%, rgba(208,184,184,1) 62
%, rgb(235, 204, 204) 73%, rgb(242, 206, 206) 97%, rgb(237, 220, 220) 99%);
    display: flex;
    justify-content: center;
    align-items: center;
    min-height: 100vh;
  }

  .container {
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100%;
  }

  .card {
    width: 450px;
    background-color: white;
    border-radius: 15px;
    padding: 40px;
    backdrop-filter: blur(10px);
    box-shadow: 0 8px 16px rgba(0, 0, 0, 0.2);
  }
```

```
h2 {

  text-align: center;
  margin-bottom: 20px;
}

form {
  display: flex;
  flex-direction: column;
}

label {

  margin-bottom: 5px;
}

input {
  padding: 10px;
  margin-bottom: 10px;
  border: none;
  border-radius: 5px;
  background-color: rgba(255, 255, 255, 0.8);
}

button {
  padding: 10px;
  background-color: #fff;
  color: #498ffc;
  border: none;
  border-radius: 5px;
  cursor: pointer;
}

button:hover {
  background-color: #70c1ff;
}

@media (max-width: 580px) {
  .card {
    width: 100%;
    max-width: 450px;
  }
}

  </style>
```

```
</head>
<body>
    <div class="container">
        <div class="card">
          <h2 class="mt-3 mb-5">Login Form</h2>
          <form method="post" action="{% url 'log' %}">
            {% csrf_token %}
            <!-- <label for="username">Username</label> -->
            {{form.username.error}}
                                {{form.username}}
            <!-- <label for="password">Password</label> -->
            {{form.password.error}}
            {{form.password}}

            <button type="submit" class="mt-2 btn btn-info">Login</button>
          </form>
          <div class="register-link text-center mt-2">
            Don't have an account? <a href="{% url 'reg' %}">Register here</a>
        </div>
        </div>
      </div>
    <!-- <div class="container mt-5">

        <div class="row justify-content-center">
          <div class="col-md-6">
              <div class="card mt-5">
                  <div class="card-header">
                      <h2 class="text-center">Login</h2>
                   </div>
                   <div class="card-body">
                      <form method="post" action="{% url 'log' %}">
                          {% csrf_token %}
                         <div class="form-group">
                            <label for="username">Username:</label>
                             {{form.username.error}}
                             {{form.username}}
                          </div>
                           <div class="form-group">
                              <label for="password">Password:</label>
                               {{form.password.error}}
                               {{form.password}}
                          </div>
                         <button
     type="submit" class="btn btn-primary btn-block">Login</button>
                      </form>
                  </div>
```

```
              </div>
           </div>
       </div>
       <div class="register-link text-center">
         Don't have an account? <a href="{% url 'reg' %}">Register here</a>
       </div>
  </div> -->
</body>
</html>
```

Registration Form

```
<!DOCTYPE html>
<html>
<head>
    <title>Registration Form</title>
    <style>
        body {
            font-family: 'Poppins', sans-serif;
            background: rgb(137,98,97);
             background: linear-gradient(90deg, rgba(137,98,97,1) 11
%, rgba(155,120,119,1) 24%, rgba(169,137,136,1) 34
%, rgba(180,150,150,1) 42%, rgba(208,184,184,1) 62
%, rgb(235, 204, 204) 73%, rgb(242, 206, 206) 97%, rgb(237, 220, 220) 99%);

            text-align: center;
        }
        h2 {
            color: #333;
        }
        form {
            width: 500px;
            margin: 0 auto;
            background-color: #fff;
            padding: 30px;
            border: 1px solid #ccc;
            border-radius: 10px;
            box-shadow: 0 0 20px rgba(0, 0, 0, 0.1);
        }
        label {
            display: block;
            margin-bottom: 10px;
        }
```

```
        input[type="submit"] {
            background-color: #007bff;
            color: #fff;
            padding: 10px 20px;
            border: none;
            border-radius: 3px;
            cursor: pointer;
        }
        input[type="submit"]:hover {
            background-color: #0056b3;
        }
    </style>
</head>
<body>
    <h2>Registration Form</h2>
    <form action="" method="post">
        {% csrf_token %}
        <label for="first_name">Firstname:</label>
        {{form.first_name.error}}
        {{form.first_name}}
        <br><br>
        <label for="last_name">Lastname:</label>
        {{form.last_name.error}}
        {{form.last_name}}
        <br><br>
        <label for="username">Username:</label>
        {{form.username.error}}
        {{form.username}}
        <br><br>

        <label for="email">Email:</label>
        {{form.email.error}}
        {{form.email}}
        <br><br>

        <label for="password">Password:</label>
        {{form.password1.error}}
        {{form.password1}}
        <br><br>

        <label for="confirm_password">Confirm Password:</label>
        {{form.password2.error}}
        {{form.password2}}
        <br><br>
```

```
      <input  type="submit" value="Register">
      <div class=
        "text-center" style=
        "font-family: Cambria,Cochin,Georgia,Times,'Times New Roman',serif;padding:10px 0px
          Back to<a href="{% url 'log' %}"> Login</a>
        </div>
    </form>
</body>
</html>
```

lndex.html

```
<!doctype html>
<html lang="en">
  <head>
    <title>AQI</title>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!-- Optional JavaScript -->
    <!-- jQuery first, then Popper.js, then Bootstrap JS -->
    <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
 integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>
    <script src=
"https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"
 integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
 crossorigin="anonymous"></script>
    <script src=
"https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
 integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
 crossorigin="anonymous"></script>
    <!-- Bootstrap CSS -->
    <link rel=
"stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css'
 integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
 crossorigin="anonymous">
    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
    <link rel="stylesheet"
 href=
"https://fonts.googleapis.com/css2
?family=Material+Symbols+Outlined:opsz,wght,FILL,GRAD@40,700,0,200" />
    {% load static %}
    <style>
        form {
```

```
            background-color: #f7f7f7;
            padding: 20px;
            border: 1px solid #ccc;
            border-radius: 5px;
            box-shadow: 0 0 50px rgba(0, 0, 0, 0.1);
        }
        body {
            background: rgb(137,98,97);
background: linear-gradient(90deg, rgba(137,98,97,1) 11
%, rgba(155,120,119,1) 24
%, rgba(169,137,136,1) 34
%, rgba(180,150,150,1) 42%, rgba(208,184,184,1) 62
%, rgb(235, 204, 204) 73%, rgb(242, 206, 206) 97%, rgb(237, 220, 220) 99%);
            background-size: cover;
            background-color: #c6c2c2;
        }
        form {
            margin: 0 auto;
            max-width: 1000px;
            margin-top: 3%;
        }
        .col {
            padding: 10x;
        }
        input[type="text"],
        input[type="number"] {
            width: 100%;
            padding: 10px;
            margin-bottom: 15px;
            border: 1px solid #ccc;
            border-radius: 3px;
        }

        label {
            display: block;
            margin-bottom: 10px;
            font-weight: bold;
            color: #333;
        }
        br {
            margin: 10px 0;
        }
        button.btn {
            color: #fff;
            padding: 10px 20px;
            border: none;
```

```
            border-radius: 3px;
            cursor: pointer;
        }

        button.btn:hover {
            background-color: #0056b3;
        }

        h2 {
            font-size: 1.5rem;
            color: #333;
            text-align: center;
            margin-top: 20px;
        }
        h6 {
            padding: 10px;

            margin-left: 95%;
            border-radius: o.75rem;
            text-decoration: none;
        }
    </style>

    </head>
  <body>
    <h6 class="mt-2"><a href="{% url 'log' %}">
<span class="material-symbols-outlined" style="color: white;">
        exit_to_app
        </span></a></h6>
    <form method="post" class="" autocomplete="on" id="myform">
        {% csrf_token %}
        <h1 class="text-center mt-3 mb-5 ">Air Quality Prediction</h1>

        <div class="row ml-5 mr-5 mt-5" >
            <div class="col" >
                {{form.pm25.error}}
                {{form.pm25}}
            </div>
            <div class="col">
                {{form.pm10.error}}
                {{form.pm10}}
            </div>
            <div class="col">
                {{form.no.error}}
                {{form.no}}
            </div>
```

```
</div><br><br>
<div class="row ml-5 mr-5">
    <div class="col">
        {{form.no2.error}}
         {{form.no2}}
    </div>
    <div class="col">
        {{form.nox.error}}
        {{form.nox}}
    </div>
    <div class="col">
        {{form.nh3.error}}
        {{form.nh3}}
    </div>
</div><br><br>
<div class="row ml-5 mr-5">
    <div class="col">
        {{form.co.error}}
        {{form.co}}
    </div>
    <div class="col">
        {{form.so2.error}}
        {{form.so2}}
    </div>
    <div class="col">
        {{form.o3.error}}
        {{form.o3}}
    </div>
</div><br><br>
<div class="row ml-5 mr-5">
    <div class="col">
        {{form.benzene.error}}
        {{form.benzene}}
    </div>
    <div class="col">
        {{form.toluene.error}}
        {{form.toluene}}
    </div>
    <div class="col">
        {{form.xylene.error}}
        {{form.xylene}}
    </div>
</div><br>
<div class="row ml-5 mr-5">
    <div class="col"></div>
    <div class="col ">
```

```
                <button type="submit" class="btn btn-info btn-block ">Predict AQI</button>
            </div>
            <div class="col"></div>
        </div>

    </form>
    <div class="row" style="width: 100%;">
          <div class="col">
            {% if prediction %}
            <h2 id="output">Predicted AQI: {{ prediction }}</h2>
            <div class="text-center mb-3">
            <img c; src="{{ plot_url }}" alt="Pie Chart" >
        {% endif %}
          </div>
        </div>
    </div>


    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
    <!-- <script>
        document.addEventListener("DOMContentLoaded", function() {
            setTimeout(function() {
                document.getElementById('output').style.display = 'none';
            }, 5000);
        });

        </script> -->

</body>
</html>
```

data.py

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
df=pd.read_csv('D:\Abhay_project\Air Quality\city_day.csv')
df

df.dtypes

df.columns
```

```
df.shape

df.info()

df.nunique()

df.describe()

df.dropna(inplace=True)
df=df.reset_index(drop=True)
df

# Select only numeric columns for correlation calculation
numeric_columns = df.select_dtypes(include=[np.number])

# Calculate the correlation matrix
correlation_matrix = numeric_columns.corr()

# Create a heatmap to visualize the correlations
plt.figure(figsize=(12, 10))
sns.heatmap(correlation_matrix, cmap="coolwarm", annot=True)

df_city_day = df.copy()
df_city_day.columns

pollutants = ['PM2.5', 'PM10', 'NO',
 'NO2', 'NOx', 'NH3','CO', 'SO2', 'O3',
 'Benzene', 'Toluene', 'Xylene']
df_city_day = df_city_day[pollutants]
print('Distribution of different pollutants in last 5 years')
df_city_day.plot(kind='line',figsize=(18,18),cmap='coolwarm',subplots=True,fontsize=9)

x=df.drop(['City','Date','AQI_Bucket'],axis=1).values
y=df['AQI_Bucket'].values
x

y

from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=.3,random_state=42)
x_train

x_test

y_train
```

```
y_test

from sklearn.neighbors import KNeighborsClassifier
model=KNeighborsClassifier()
model.fit(x_train,y_train)
y_pred=model.predict(x_test)
y_pred

df1=pd.DataFrame({'y_test':y_test,'y_pred':y_pred})
df1

from sklearn.metrics import accuracy_score,r2_score,confusion_matrix,classification_report
print('confusion_matrix','\n',confusion_matrix(y_test,y_pred))
print('*'*100)
print('accuracy_score is:',accuracy_score(y_test,y_pred))
print('*'*100)
print('classification report:',classification_report(y_test,y_pred))
```

views.py

```
from django.shortcuts import render,redirect
from .forms import AirQualityCheckForm
import pandas as pd
from .models import *
from .forms import *
from django.views.generic import FormView,CreateView,TemplateView
from django.contrib.auth.models import User
from django.contrib.auth import authenticate,login
from django.urls import reverse_lazy



class LoginView(FormView):
    template_name="login.html"
    form_class=LogForm
    def post(self,request,*args,**kwargs):
        log_form=LogForm(data=request.POST)
        if log_form.is_valid():
            us=log_form.cleaned_data.get('username')
            ps=log_form.cleaned_data.get('password')
            user=authenticate(request,username=us,password=ps)
            if user:
                login(request,user)
```

```
                return redirect('h')
            else:
                return render(request,'login.html',{"form":log_form})
        else:
            return render(request,'login.html',{"form":log_form})


class RegView(CreateView):
     form_class=Reg
     template_name="reg.html"
     model=User
     success_url=reverse_lazy("log")



# Load the CSV data
df = pd.read_csv('C:/Users/User/Downloads/Air Quality (2)/Air Quality/city_day.csv')

# Define the prediction function
def predict_aqi(pm25, pm10, no, no2, nox, nh3, co, so2, o3, benzene, toluene, xylene):
    # Use the input values to filter the DataFrame and calculate AQI
    filtered_data = df[
        (df['PM2.5'] == pm25) &
        (df['PM10'] == pm10) &
        (df['NO'] == no) &
        (df['NO2'] == no2) &
        (df['NOx'] == nox) &
        (df['NH3'] == nh3) &
        (df['CO'] == co) &
        (df['SO2'] == so2) &
        (df['O3'] == o3) &
        (df['Benzene'] == benzene) &
        (df['Toluene'] == toluene) &
        (df['Xylene'] == xylene)
    ]

    if not filtered_data.empty:
        prediction = filtered_data['AQI_Bucket'].values[0]
    else:
        prediction = "No matching data found"

    return prediction

def predict_air_quality(request):
    if request.method == 'POST':
        form = AirQualityCheckForm(request.POST)
```

```
        if form.is_valid():
            user_input = form.cleaned_data
            pm25 = user_input['pm25']
            pm10 = user_input['pm10']
            no = user_input['no']
            no2 = user_input['no2']
            nox = user_input['nox']
            nh3 = user_input['nh3']
            co = user_input['co']
            so2 = user_input['so2']
            o3 = user_input['o3']
            benzene = user_input['benzene']
            toluene = user_input['toluene']
            xylene = user_input['xylene']

            # Use the prediction function to predict AQI
            prediction = predict_aqi(pm25, pm10,
                  no, no2, nox, nh3, co, so2, o3, benzene, toluene, xylene)
            plot_url = create_pie_chart(pm25, pm10,
                 no,no2,nox,nh3,co,so2,o3,benzene,toluene,xylene)
            context = {
                'form': form,
                'prediction': prediction,
                'plot_url':plot_url
            }
            return render(request, 'index.html', context)
    else:
        form = AirQualityCheckForm()

    context = {
        'form': form,
    }
    return render(request, 'index.html', context)



from django.shortcuts import render
import matplotlib.pyplot as plt
from io import BytesIO
import base64



def create_pie_chart(pm25, pm10,no,no2,nox,nh3,co,so2,o3,benzene,toluene,x):
    plt.figure()
    labels = ['PM2.5', 'PM10', 'NO' ,'NO2','NOX','NH3','CO','SO2','O3','B','T','X']
```

```
    data = [pm25, pm10, no,no2,nox,nh3,co,so2,o3,benzene,toluene,x]
    colors = ['red', 'yellow', 'green','blue', 'gray', 'orange','cyan',
      'lightcoral', 'violet','black','greenyellow','brown']
    explode=(0.1,0.1,0.6,0.1,0.6,0.1,0.6,0.1,0.1,0.5,0,0.5)


    plt.pie(data, labels=labels, explode=explode, colors=colors,
        autopct='%1.1f%%', startangle=90)
    plt.legend()
    plt.axis('equal')

    # Save the plot to a BytesIO object
    image_stream = BytesIO()
    plt.savefig(image_stream, format='png')
    image_stream.seek(0)

    # Encode the image to base64 for embedding in HTML
    plot_url = base64.b64encode(image_stream.read()).decode('utf-8')

    return f'{plot_url}'
```

urls.py

```
[breaklines=true]
from django.urls import path
from .views import *

urlpatterns = [
    path('reg/',RegView.as_view(),name='reg'),
    path('predict/',predict_air_quality,name='h'),
]
```

Settings.py

```
"""
Django settings for quality project.

Generated by 'django-admin startproject' using Django 4.2.6.

For more information on this file, see
https://docs.djangoproject.com/en/4.2/topics/settings/

For the full list of settings and their values, see
```

```
https://docs.djangoproject.com/en/4.2/ref/settings/
"""

from pathlib import Path

# Build paths inside the project like this: BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent


# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/4.2/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'django-insecure-@f_-vc23$@iwqs-s%1d84j&^c+v6hl@v5bc$u@rf36l^vd3jn6'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = []


# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'setup'
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'quality.urls'

TEMPLATES = [
```

```
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]

WSGI_APPLICATION = 'quality.wsgi.application'


# Database
# https://docs.djangoproject.com/en/4.2/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
    }
}


# Password validation
# https://docs.djangoproject.com/en/4.2/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]
```

```
# Internationalization
# https://docs.djangoproject.com/en/4.2/topics/i18n/

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_TZ = True


# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/4.2/howto/static-files/

STATIC_URL = 'static/'

# Default primary key field type
# https://docs.djangoproject.com/en/4.2/ref/settings/#default-auto-field

DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
```

Forms.py

```
from django import forms
from django.contrib.auth.forms import UserCreationForm
from django.contrib.auth.models import User

class LogForm(forms.Form):
    username=forms.CharField
      (widget=forms.TextInput
      (attrs={"placeholder":"Username",
       "class":"form-control","style":"border-radius: 0.75rem; "}))
    password=forms.CharField
      (widget=forms.PasswordInput
      (attrs={"placeholder":"Password",
      "class":"form-control","style":"border-radius: 0.75rem; "}))


class Reg(UserCreationForm):
    class Meta:
        model=User
        fields=['first_name','last_name','email','username','password1','password2']
```

```
    first_name=forms.CharField
     (widget=forms.TextInput
    (attrs=
  {"placeholder":"Firstname",
     "class":"form-control","style":"border-radius: 0.5rem;padding:10px 50px;"}))
    last_name=forms.CharField
   (widget=forms.TextInput
    (attrs=
   {"placeholder":"Lastname",
    "class":"form-control","style":"border-radius: 0.5rem;padding:10px 50px; "}))
     username=forms.CharField
   (widget=forms.TextInput
   (attrs=
     {"placeholder":"Username",
     "class":"form-control","style":"border-radius: 0.5rem;padding:10px 50px; "}))
      email=forms.CharField
     (widget=forms.TextInput
      (attrs=
        {"placeholder":"Email",
        "class":"form-control","style":"border-radius: 0.5rem;padding:10px 50px; "}))
    password1=forms.CharField
     (widget=forms.PasswordInput
     (attrs=
      {"placeholder":"Password",
       "class":"form-control","style":"border-radius: 0.5rem;padding:10px 50px; "}))
    password2=forms.CharField
     (widget=forms.PasswordInput
      (attrs=
       {"placeholder":"Confirm Password",
        "class":"form-control","style":"border-radius: 0.5rem;padding:10px 50px; "}))
```

```
class AirQualityCheckForm(forms.Form):
    pm25 = forms.FloatField
                   (label='PM2.5',widget=forms.TextInput
                   (attrs={"placeholder":"PM2.5","class":"form-control","id":"pm25"}))
    pm10 = forms.FloatField
                  (label='PM10',widget=forms.TextInput
                  (attrs={"placeholder":"PM10","class":"form-control","id":"pm10"}))
    no = forms.FloatField
                  (label='NO',widget=forms.TextInput
                  (attrs={"placeholder":"NO","class":"form-control"}))
    no2 = forms.FloatField
                  (label='NO2',widget=forms.TextInput
```

```
                (attrs={"placeholder":"NO2","class":"form-control"}))
    nox = forms.FloatField
                (label='NOx',widget=forms.TextInput
                (attrs={"placeholder":"NOX","class":"form-control"}))
    nh3 = forms.FloatField
                (label='NH3',widget=forms.TextInput
                (attrs={"placeholder":"NH3","class":"form-control"}))
    co = forms.FloatField
                (label='CO',widget=forms.TextInput
                (attrs={"placeholder":"CO","class":"form-control"}))
    so2 = forms.FloatField
                (label='SO2',widget=forms.TextInput
                (attrs={"placeholder":"CO2","class":"form-control"}))
    o3 = forms.FloatField
                (label='O3',widget=forms.TextInput
                (attrs={"placeholder":"O3","class":"form-control"}))
    benzene = forms.FloatField
                (label='Benzene',widget=forms.TextInput
                (attrs={"placeholder":"BENZENE","class":"form-control"}))
    toluene = forms.FloatField
                (label='Toluene',widget=forms.TextInput
                (attrs={"placeholder":"TOLUENE","class":"form-control"}))
    xylene = forms.FloatField
                (label='Xylene',widget=forms.TextInput
                (attrs={"placeholder":"XYLENE","class":"form-control"}))
```

Model.py

```
[breaklines=true]

from django.db import models

class AirQualityData(models.Model):
    city = models.CharField(max_length=100)
    date = models.DateField(auto_now_add=True)
    aqi_bucket = models.CharField(max_length=20)
    pm25 = models.FloatField()
    pm10 = models.FloatField()
    no = models.FloatField()
    no2 = models.FloatField()
    nox = models.FloatField()
    nh3 = models.FloatField()
```

```
    co = models.FloatField()
    so2 = models.FloatField()
    o3 = models.FloatField()
    benzene = models.FloatField()
    toluene = models.FloatField()
    xylene = models.FloatField()
    aqi = models.CharField(max_length=255)

    def __str__(self):
        return f"{self.city} - {self.date}"
```

VENTURE VISTA

PROJECT REPORT

Submitted By
SHAMNAS A.P
Reg. No. CCAVBCA013

for the award of the Degree of

Bachelor of Computer Application (B.C.A.)

(University of Calicut)

under the guidance of

Ms. Soumya P.S
Assistant Professor

Bachelor of Computer Application
DEPARTMENT OF COMPUTER SCIENCE
CHRIST COLLEGE(Autonomous)
IRINJALAKUDA, KERALA
INDIA

2021-24

DEPARTMENT OF COMPUTER SCIENCE

CHRIST COLLEGE (AUTONOMOUS)

IRINJALAKUDA

CERTIFICATE

This is to certify that the project report entitled ***"VENTURE VISTA"*** *is a bonafide record of the project work done by* ***SHAMNAS AP*** *in partial fulfillment of the requirement for the sixth semester of* ***Bachelor of Computer Application*** *in Department of Computer Science of* ***CHRIST COLLEGE (AUTONOMOUS) IRINJALAKUDA***

.

Ms. SOUMYA P.S
Assistant Professor,CS
Internal Guide

Ms. SINI THOMAS
Head of the Department
Computer Science

EXTERNAL EXAMINER

INTERNAL EXAMINER

DECLARATION

We hereby declare that this project work **"VENTURE VISTA"** submitted to Christ College (Autonomous) Irinjalakuda, affiliated to Calicut University in partial fulfillment of the requirement for the award of the Bachelor of Computer Application, is a record of original work done by us, under the guidance of Ms. SOUMYA P.S, Department of computer Science.

Place: Irinjalakuda SHAMNAS AP

ACKNOWLEDGEMENT

First and foremost I like to thank Lord almighty for his providence and for being the guiding light throughout the project. I wish to express my sincere gratitude to my beloved Department head for giving me all the facilities for my project. I take this opportunity to express my gratitude to the class teacher Ms. SOUMYA P.S and head of the department Ms. SINI THOMAS who has been supported me throughout the course of this project. I are thankful for her aspiring guidance and valuable advice during the project work. I express my sincere thanks to my project guide Ms. SOUMYA P.S for supporting and guiding throughout the project.I would take this opportunity to specially thank all other faculty members for their constant and continuous motivation. Finally I would like to thank my family and friends for giving valuable advice and moral support throughout my project.

ABSTRACT

VENTURE VISTA Our travel guide website aims to inspire and assist travelers in exploring new destinations worldwide. From detailed city guides to off-the-beaten-path adventures, we provide a wealth of information to help users plan unforgettable trips. With curated recommendations for accommodations, dining, activities, and transportation, our platform caters to all types of travelers, security scores,live money exchange rates. By combining expert insights with user reviews and ratings, we strive to be a one-stop resource for discovering and planning your next adventure.

Contents

Chapter 1

1 Introduction

This is an web application that helps the users to know more about the major attractions near to them. The application provides all the details about tourist places or major attractions like their opening time, closing time, history and other details. The app also provides all the details of the hotels available for the users. The app can also provide the best route to the places you have selected. The app will come in handy for those who travel a lot, or like to travel to new places.

1.1 Overview

The objective of the TRAVEL GUIDE APPLICATION is to design an simple and adaptable application that helps the users to know more about the major attractions near to them. The web application includes many interesting features such as providing place suggestions to visit which best suites for the preferences you have given.It is a User friendly application.

Chapter 2

2 System Analysis

2.1 Purpose

A travel guide application serves as a comprehensive resource for travelers, offering a wide range of information and features to enhance their travel experiences. Its primary purpose is to provide users with detailed insights into various destinations, including popular attractions, historical sites, cultural experiences, dining options, accommodations, and local services.

By leveraging the app, users can efficiently plan their trips, exploring different destinations based on their interests and preferences. They can access detailed descriptions, reviews, and ratings for various attractions and services, helping them make informed decisions about their travel itinerary.

Furthermore, a travel guide application often includes features such as offline maps, itinerary planners, currency converters, and language translators, which can be invaluable for travelers navigating unfamiliar environments. These features can help users navigate their destinations more effectively, communicate with locals, and manage their travel budgets.

Overall, a travel guide application aims to simplify the travel planning process, inspire users to explore new destinations, and enhance their overall travel experiences by providing them with relevant and reliable information.
.

2.1.1 Existing System

The existing system are apps that provides data about tour packages and so on. Apps like this, which provide all details of the places are not available. Limitations of existing system :We cannot get all the details about the places we are looking for from a single application.

2.1.2 Proposed System

The proposed system is an web application which provides all the details related to the places you are looking for your next trip. We also provide a feature that gives you suggestions of the places near you to visit, based on the preferences you have provided. Advantages of proposed system :The application gives all the details about the places you are looking for.It also

provide place suggestions to visit which best suites for the preferences you have given.

2.2 Problem definition

To know what the problem is and what the needs are before developing it. Here we design an application for travel guiding.

2.3 FEASIBILITY STUDY

After the problem is clearly understood and solutions proposed, the next step is to conduct the feasibility study. Feasibility study is defined as evaluation or analysis of the potential impact of the proposed project or program. The objective is to determine whether the proposed system is feasible. There are three aspects of feasibility study to which the proposed system is subjected as discussed below.

2.3.1 Technical Feasibility

Technical feasibility assess whether the current technical resources are sufficient for the new system. We can upgrade the level of technology for supporting our website. We check whether the proposed system can be implemented in the present system without supporting the existing hardware.

2.3.2 Economical Feasibility

Economic feasibility determines whether the time and money are available to develop the system. There is no additional hardware used to develop the site, it is inexpensive to build.

2.3.3 Operational Feasibility

Operational feasibility determines if the human resources are available to operate the system once it has been installed. No extra training is needed to use this system. Anyone who has knowledge in internet in English language can easily use the system. The resources that are required to implement or install are already available with the organization.

Chapter 3

3 Software Requirement Specification

3.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the TRAVEL GUIDE APPLICATION. It illustrate the purpose and complete description for the development of the system. It explain system constraints, interface and interactions with other external applications. This document is primarily intended to be proposed to provide all the details related to the places you are looking for your next trip.

3.2 Scope

The application gives all the details about the places you are looking for.It also provide place suggestions to visit which best suites for the preferences you have given.

3.3 Overall Description

This section give an overview of our application, TRAVEL GUIDE APPLICATION. This is an web application that helps the users to know more about the major attractions near to them. The application provides all the details about tourist places or major attractions like their opening time, closing time, history and other details. The app also provides all the details of the hotels available for the users. The app can also provide the best route to the places you have selected. The app will come in handy for those who travel a lot, or like to travel to new places.

3.3.1 Product Perspective

TRAVEL GUIDE APPLICATION is mainly used to know the details about your next trip. It also provide features that gives you suggestions to visit according to the preferences you have given.

3.3.2 Product Functionality

Through this system admin can upload various data. User can view the major attractions and their history and other details available in an area.

Users can also view all the information about the hotels available in a specific area.

3.3.3 Users and Characteristics

There are two types of users that interact with the application, people who wish to travel and the guide. Each of these have different tasks which is performed. Admin is able to upload details of places and can view the feedback.

3.4 Specific Requirements

3.4.1 Hardware Requirements

- System: IBM-Compatible PC
- Processor: Intel Core i3
- Speed: Above 1GHz
- RAM capacity: 4 GB
- Hard Dsk drive: 500 GB

3.4.2 Software Requirements

- Front End : React
- Back End : Djanko Python
- Database : MySql

3.5 Functional Requirements

It contains three main modules.

- 1.Admin
- 2.User

Admin

The admin will upload the details of the places. All the details that are provided in the app is uploaded by the admin. The admin can view user's feedbacks and can take suitable decisions based on the user's decisions.

User

The users can view the major attractions and their history and other details available in an area. Users can also view all the information about the hotels available in a specific area. The users can get the best route available to the selected place from their current location, which is calculated by the app and shown in the map. Users can also get notifications on the places that are suggested by the app based on the preferences they have given.

3.6 Non Functional Requirements

Non-functional requirements define the overall qualities or attributes of the resulting system. Non-functional system place restrictions on the product being developed, the developed process, and specify external constraints that the product must meet. Examples of non-functional requirements include safety, security, usability, reliability and performance requirements. Project management issues (costs, time and schedule) are often considered as non-functional requirements. The principal non - functional constraints which are relevant to critical systems :

- performance
- security
- safety
- usability

Performance

Performance requirements concern the speed of operation of a system. Types of performance requirements :

- Response requirements (how quickly the system reacts to a user input).
- Throughput requirements (how much the system can accomplish within a specified amount of time).
- Availability requirements (is the system available for service when requested by end users). The speed of operation of this system is adequate for the requirements.

Reliability

- Reliability is the ability of a system to perform its required function under stated conditions for a specified period of time.

- constraints on the runtime behavior of the system. This system is reliable because its functionalities can be done on the required conditions.

Safety

Safety requirements are not required which exclude unsafe situation from the possible solution of the system.

Usability

Usability is the ease with which a user can learn to operate, prepare inputs for, and interpret outputs of system or components. Usability requirements include :

- information error messages.
- well-formed user interfaces.

3.7 Interface Requirements

3.7.1 Hardware interfaces

The system must run over the internet, all the hardware shall require to connect internet will be hardware interface for the system. As for example modem, WAN-LAN.

3.7.2 Software interfaces

Software interface required for the working of the project is the appropriate operating system.

3.7.3 Communication interfaces

The user of the site communicate using the network connectivity and the data set is accessed

3.8 Security Requirements

- User accesses only their account.
- Validation of input is handled.
- This application containing the computer systems is physically secured against arms or surreptitious entry by intruders.
- Users must be authorized carefully to reduce changes of any such user giving access to an intruder in exchange for a bribe or other favour.

3.9 Platform Used

Windows 10 is a major version of the Microsoft Windows operating system that was released on July 29, 2015.It is built on the Windows NT kernel and follows windows 8.Part of the reason Microsoft decided to name the 2015 release "Windows 10"(and skipped "windows 9")is because the operating system is designed to be a new direction for Microsoft. One of the primary aims of windows 10 is to Unify the windows experience across multiple devices, such desktop computers, tablets, and smartphones. As part of this effort, Microsoft developed Windows10 Mobile alongside Windows 10 to replaces Windows Phone - Microsoft's previous mobile OS. Windows 10 also integrates other Microsoft services, such as Xbox Live and the Cortana voice recognition assistant.

3.10 Technologies Used

Djanko

Django is a free and open-source web framework, written in Python, that follows the model-template-views (MTV) architectural pattern. It is used to build database-driven websites. Django takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel.

React

React.js is an open-source JavaScript library used to build user interfaces (UIs) for single-page applications. React manages the view layer and is used for the development of both web and mobile applications.React is a component-based library, which means that UIs are built by combining reusable components. Components are self-contained pieces of code that can be used to render HTML, CSS, and JavaScript.React uses a virtual DOM, which is a lightweight representation of the real DOM. The virtual DOM is used to efficiently update the UI when data changes.React is a popular choice for building UIs because it is fast, scalable, and easy to learn.

MySQL

MySQL, pronounced either "My S-Q-L" or "My Squel, " is an open source relational database management system. It is based on the structure query

language (SQL), which is used for adding, removing and modifying information in the database. Standard SQL commands, such as ADD, DROP, INSERT,and UPDATE can be used with MySQL.

What distinguishes PHP from something like client - side Javascript is that the code is executed on the server, generating HTML which is then sent to the client. The client would receive the results of running that script, but would not know what the underlying code was. You can even configure your web server to process all your HTML files with PHP, and then there's really no way that users can tell what you have up your sleeve. The best things in using PHP are that it is extremely simple for a newcomer, but offers many advanced features for a professional programmer.

Chapter 4

4 Design Document

4.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the TRAVEL GUIDE APPLICATION.. It illustrate the purpose and complete description for the development of the system. It explain system constraints, interface and interactions with other external applications. This document is primarily intended help the users to know more about the major attractions near to them.This section give an overview of our system, "9 to 5" APPLICATION. The application provides all the details about tourist places or major attractions like their opening time, closing time, history and other details. The app also provides all the details of the hotels available for the users. The app can also provide the best route to the places you have selected. The app will come in handy for those who travel a lot, or like to travel to new places.

4.2 Scope

The application gives all the details about the places you are looking for.It also provide place suggestions to visit which best suites for the preferences you have given.

4.3 Overview

The purpose of this document is to help the reader to visualize the solution to the project presented.This document verifies how the design meet the requirement stimulated in the SRS document through design viewpoints.The design viewpoints will cover all design elements presented before. This document will provide a direct approach to the development of this project hence reducing feature creep and ponitedly determine the quality of the design.

4.4 Data Design

Database are the storehouses of data used in the software systems.The data is stored in tables inside the database.Several tables are created for the manipulation of the data for the system.Two essential settings for a database are

- Primary Key- The field that is unique for all the record occurrence.
- Foreign Key- The field used to set relation between tables.

Normalization

- Normalization is the process of organizing the data in the database.
- Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate the undesirable characteristics like Insertion, Update and Deletion Anomalies.
- Normalization divides the larger table into the smaller table and links them using relationship.
- The normal form is used to reduce redundancy from the database table.

Chapter 5

5 Development of the System

The system is an web application which provides all the details related to the places you are looking for your next trip. We also provide a feature that gives you suggestions of the places near you to visit, based on the preferences you have provided.

Chapter 6

6 System Testing

Testing is the penultimate step of software development.An elaborate testing of the data is prepared and the system is using the test data,while doing testing,errors are noted and correction is made.The users are trained to operate the developed system.Both hardware and software securities are made to run the developed system successfully.System testing is aimed at ensuring the system works accurately before the live operation commences.

6.1 Test Plan

6.1.1 Scope

This test plan will cover the following testing activities as identified in the testing strategy.

- White Box Testing

White box testing is a testing technique, that examines the program structure and derives test data from the program logic/code. In white box testing, the UI is bypassed. Inputs and outputs of ratings are tested directly at the code level and the results are compared against specifications. It Reveals errors in "hidden" code.

- Black Box Testing

Black - box testing is a method of software testing that examines the functionality of an application based on the specifications. Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end - user would.

- Unit Testing

The module includes four main components to undergo unit testing. It checks the syntax error, logic error and validity of the program correctness. This test will be performed by the developers.

- Integration Testing

After all components pass the unit testing, they get to be tested whether they work correctly when they are running concurrently and communicating to each other. Integration tests exercise an entire subsystem and ensure that a set of components play nicely together.

- Internal Data Testing

We will test the validity of the data before it enters the database to avoid any problems that may face us in the database. We will test the encryption of the personal information of all the users along with the user names and passwords to ensure maximum security of the users privacy.

6.1.2 Software risk issues

In this section, the plan is to test the risk involved in critical issues such as: Difficult to run on visual studio due to large loading time. Some inherent software risks such as complexity exist; also these issues need to be identified. Proper network connection Working of Mysql database

6.1.3 Features to be tested

- Test whether correct user name and password allows you to login.
- Test whether invalid user name and password prevents you from login.
- Test whether there is any connection problem in the Server.
- Test whether the student and admin details are entered correctly.
- Test whether invalid data entry allows saving data successfully.
- Test whether all pages are loaded correctly.
- Test whether watermark is embedded and extracted properly.
- Test whether the provided security works correctly.

6.2 Test consolidation

6.2.1 Test item

The items or features to be tested in the test cases are included in the document. Each and every user input is tested. The present condition of the system is tested. It is checked to make sure that environment is ready for the application to work.

6.2.2 Input specifications

These are the inputs required to execute the test case. All required inputs including data elements and values are given below.

Data	Values
Email	A valid email ID
Password	Combination of characters and numbers
Phone number	10 Digit number

Chapter 7

7 System Implementation and Maintenance

7.1 Implementation

System implementation is the conversion of new system into an operating one which involves creating compatible files, training clients and installing hardware. User training is crucial for minimizing resistance to change and giving chance to prove its worth. Training aids user friendly manuals and healthy screens provide the user with a good start. Software maintenance follows conversion to the extent that changes are necessary to maintain satisfactory operations relative to changes in the user's environment. Maintenance often includes minor enhancements or corrections to the problem that surface late in the systems operations. In the implementation phase, the team builds the components either from scratch or by composition. Given the architecture document meant from the design phase and the requirement document from the analysis phase, the team should build exactly what has been requested, though there is still room for innovation and flexibility. For example, a component may be narrowly designed for this particular system, or the component may be made more general to satisfy a reusability

- Careful planning
- Investigation of system and constraints
- Design the methods to achieve changeover.
- Training the staff in the changed phase.
- Evaluation of change over method.
- The method of implementation and time scale to be adopted are found out initially.

7.2 Maintenance

This phase occurs as a result of deploying the whole system at the end users organization. They will perform the beta testing at the end users and inform to the developers about any needed modification to the application. The customer records all the problems that are encountered during the beta testing and reports these to the developer at regular intervals.

7.2.1 Corrective Maintenance

Even with the best quality assurance activities, it is likely that the customer will uncover defects in the software. Corrective maintenance changes the software to correct the defects. Corrective Maintenance activity may consist of repair, restoration or replacement of equipment. This activity will be result of a regular inspection, which identifies the failure in time for corrective maintenance to be planned and scheduled, then performed during a routine maintenance shutdown.

7.2.2 Adaptive Maintenance

Over time, the original environment(CPU, operating system, business rules, external product characteristics) for which the software was developed is likely to change. Adaptive maintenance results in modification to the software to accommodate changes to its external environment.

7.2.3 Enhanced Maintenance

As software is used, the customer/user will recognize additional functions that will provide the benefit. Perfect maintenance extends the software beyond its original functional requirements.

7.2.4 Preventive Maintenance

Computer software deteriorates due to change, and because of this preventive maintenance often called software re-engineering, must be conducted to enable the software to serve the needs of its end users. Preventive maintenance makes changes to computer programs so that they can be more easily corrected, adapted and enhanced.

Chapter 8

8 Conclusion and Future Scope

8.1 Conclusion

There are two types of users that interact with the system Admin, and the traveler.Each of these two types has different uses of the system so each of them has their own panel.Admin can manage all the features of application dynamically by login on admin panel. The users can view the major attractions and their history and other details available in an area. Users can also view all the information about the hotels available in a specific area.

8.2 Future Scope

- Gives all the details about the places you are looking for.
- Provide place suggestions to visit which best suites for the preferences you have given.

Appendix

A Data Flow Diagram

Data flow is the one of the best way of documenting the entire functionality of the system.For the system ,which will have data flows in and have some processing inside and then some data flow out from the system can be documented or represented effectively by means of data flow out from the system can be documented or represented effectively by means of data flow diagrams. The data flow diagram are a diagrammatic representation of the system,which has input,process and output.Once any system is represented using a data flow diagram we can identify the following things easily:

- Various entities interacting with the system are identified
- Flow of data from one entity to another is identified
- The various processes involved in between the interaction of two or more entities in the system are clearly pointed out
- The various data stores which hold the data in between the process,are clearly identified

Some Data Flow Diagram charting forms:

A.1 External source or receiver

A source or sink is a person or part of organization ,which enters or receives information from the system,but is considered to be outlining the contest of data flow model.

A.2 Transform process

A process represents transformation where incoming data flowsare changed into outgoing data flow

A.3 Data Store

A data store is repository of data that is to be stored for use by one or more process may be as simple as buffer or queue or sophisticated as relational database. They should have clear names.If a process merely uses the contest of store and does not alter it,the arrowhead goes only form the store to the

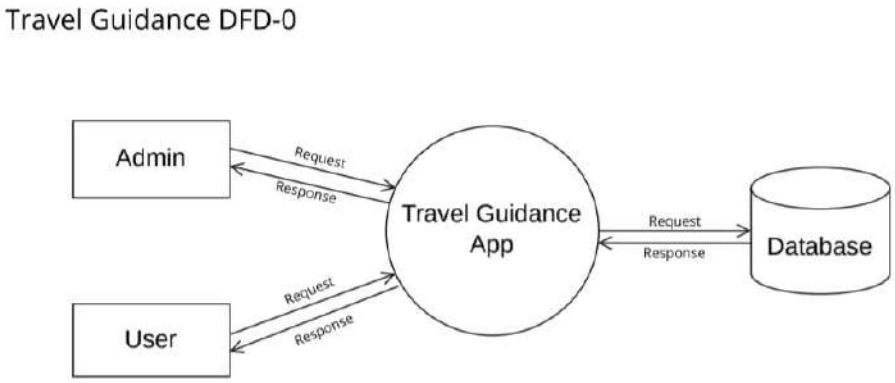
process. If a process alters the details in the store then double-headed arrow is used.

A.4 Data flow

A data flow is a route, which enable packets of data to travel from one point to another.Data may flow,with arrowhead pointing in the direction of the flow

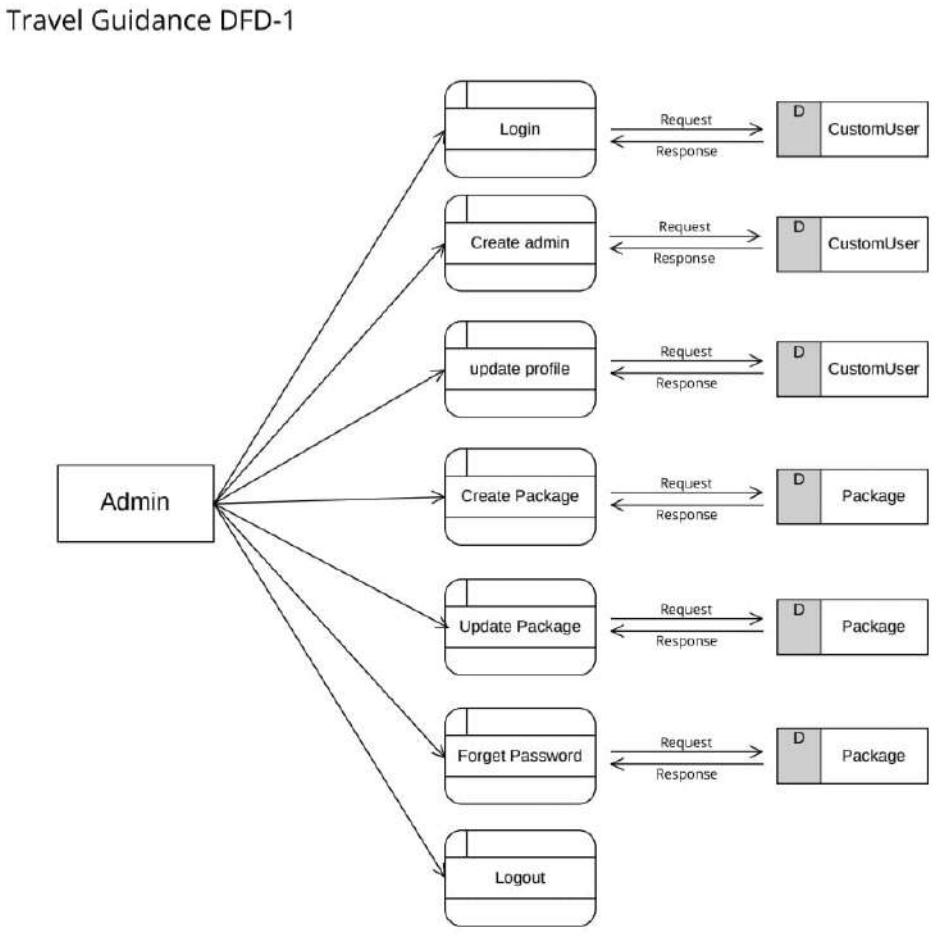
B dfd0

B.1 Level 0



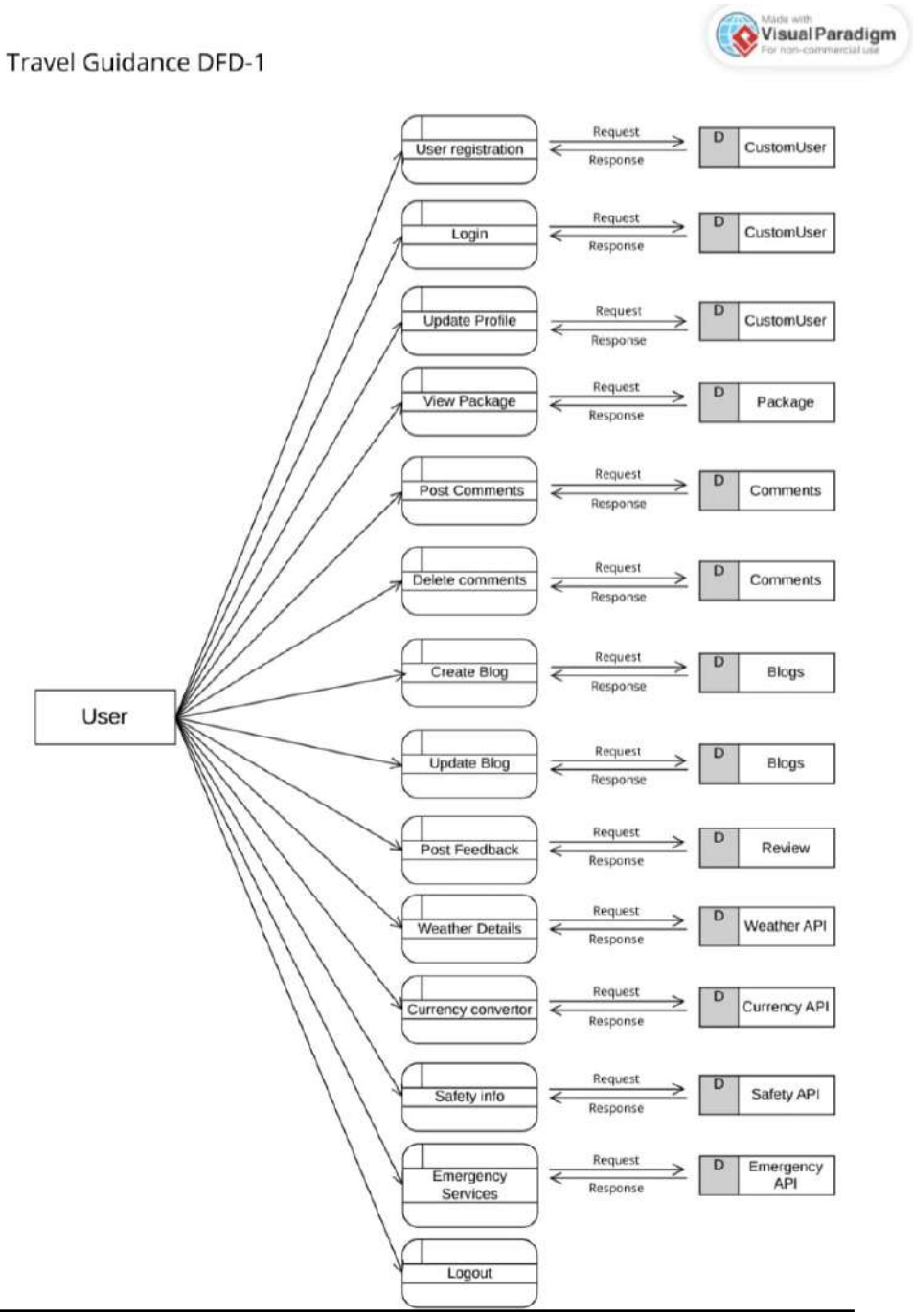
C Data Flow Diagram of Admin

C.1 Level 0

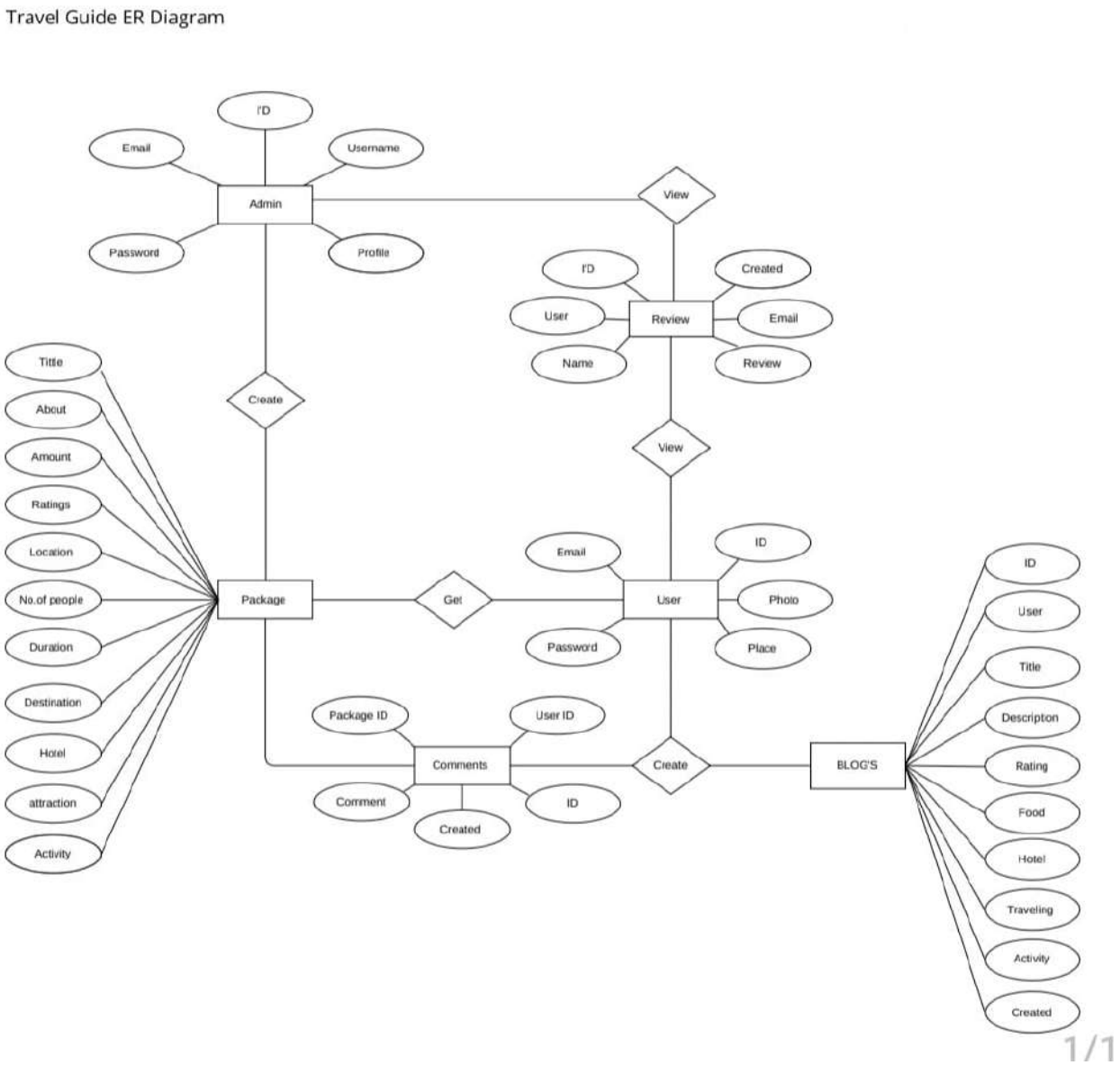


D Data Flow Diagram of User

D.1 Level 0

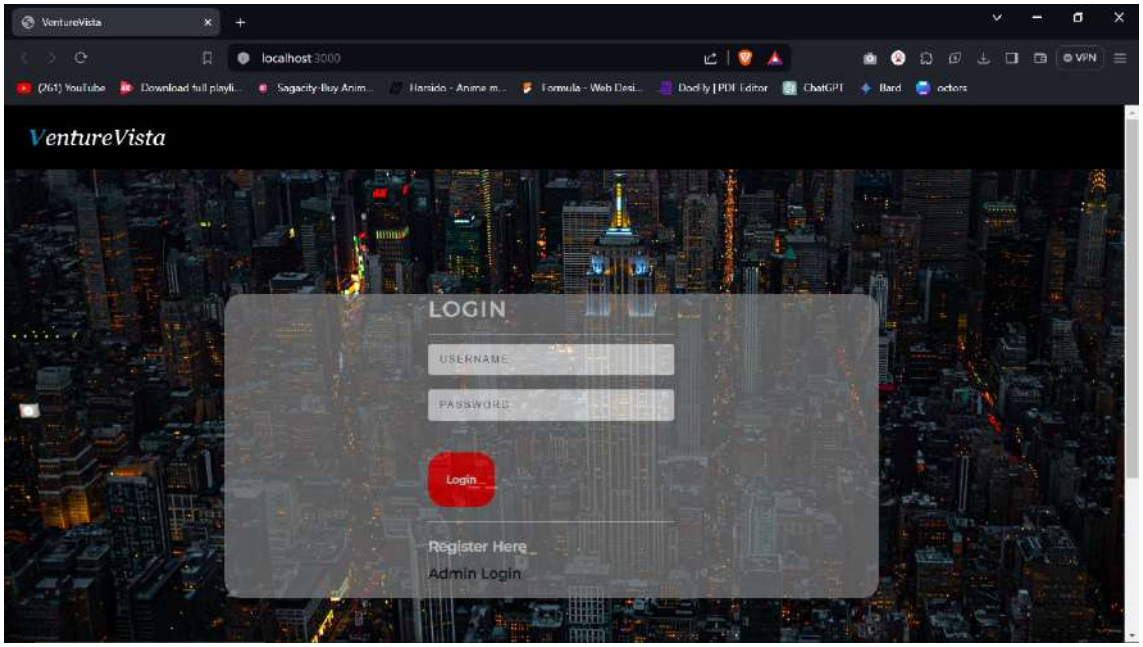


E E-R Diagram

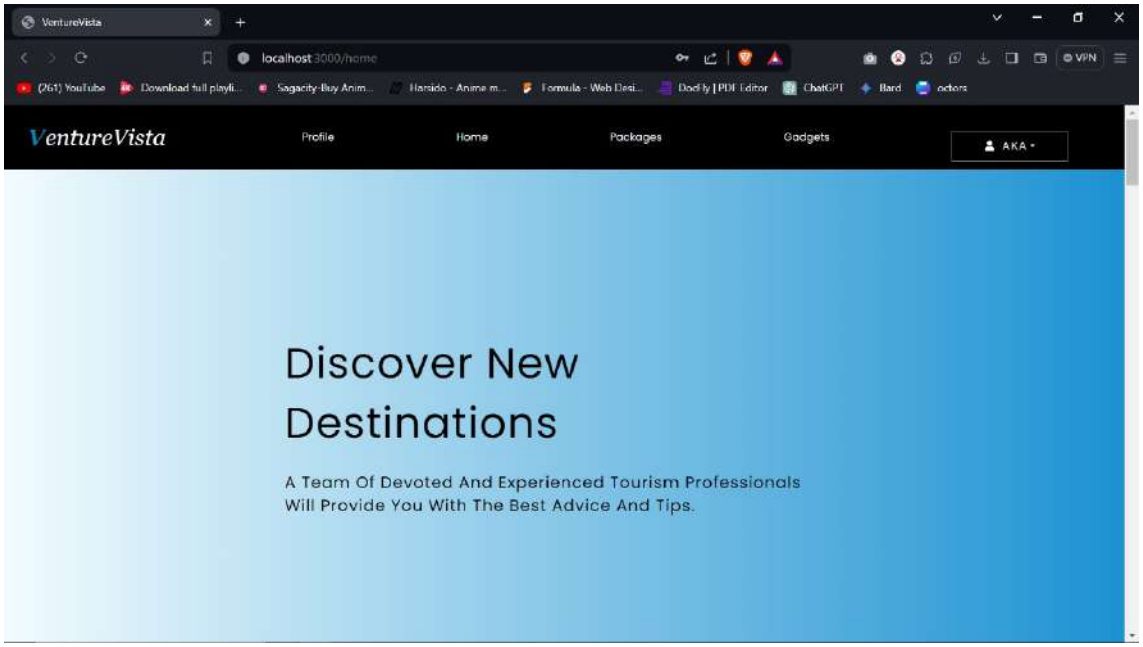

Travel Guide ER Diagram
I'D
Email
Username
Admin
Password
Profile
View
I'D
Created
User
Review
Email
Name
Review
Tittle
About
Amount
Ratings
Location
No.of people
Duration
Destination
Hotel
attraction
Activity
Create
View
Email
ID
Package
Get
User
Photo
Password
Place
Package ID
User ID
Comments
Create
BLOG'S
Comment
ID
Created
ID
User
Title
Description
Rating
Food
Hotel
Traveling
Activity
Created
1/1

F USER INTERFACES

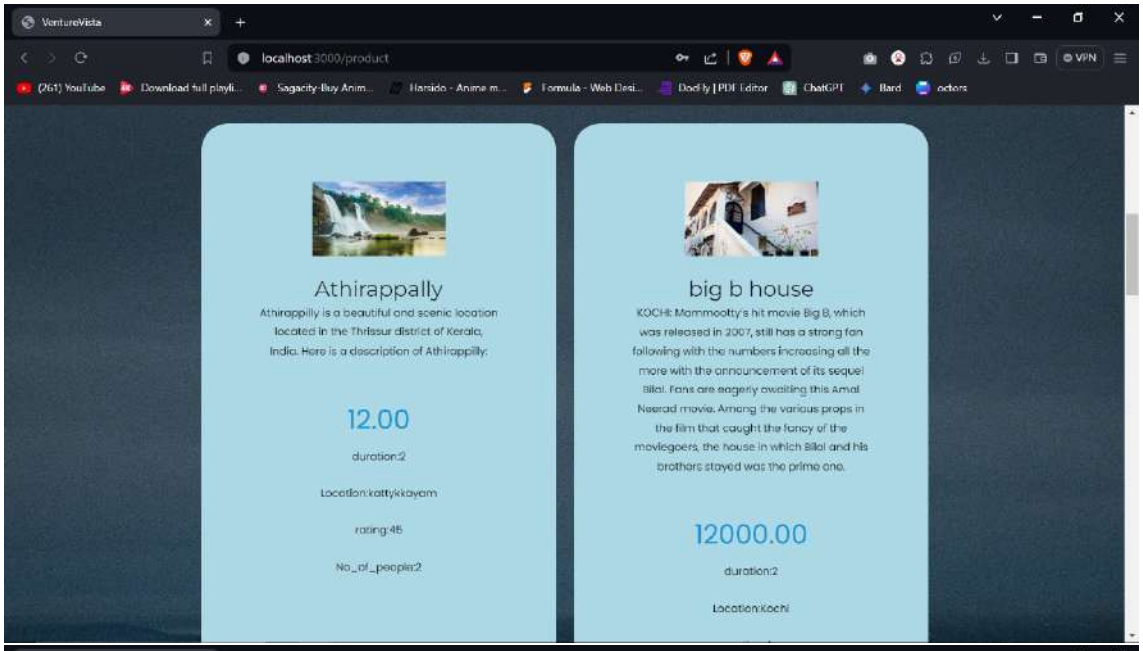
F.1 LOGIN

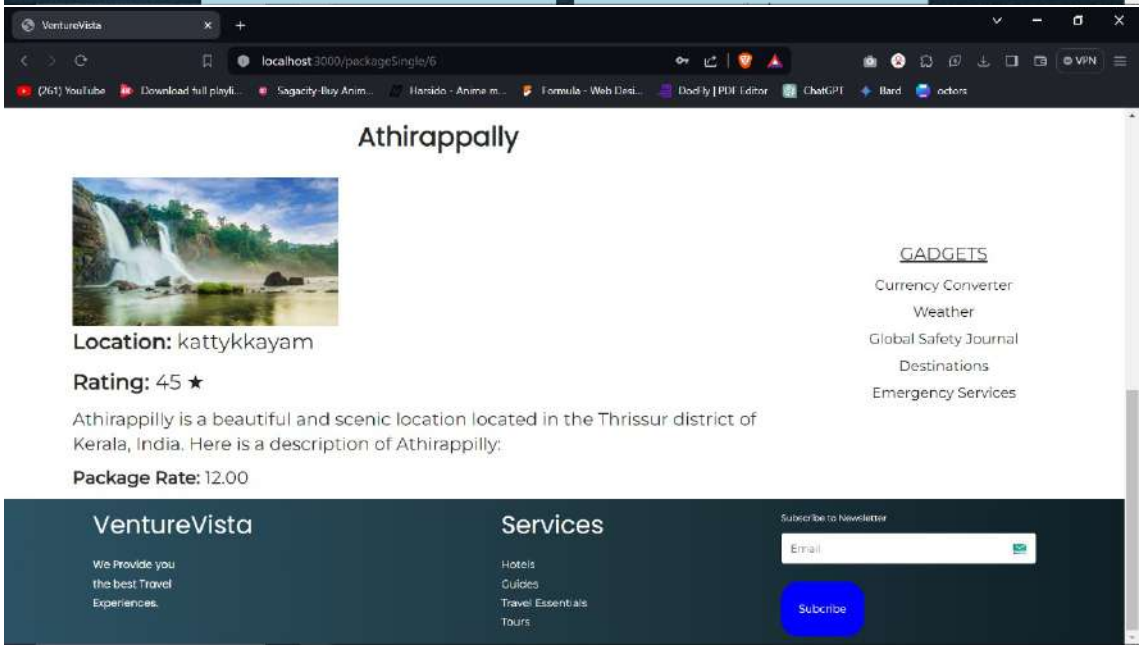


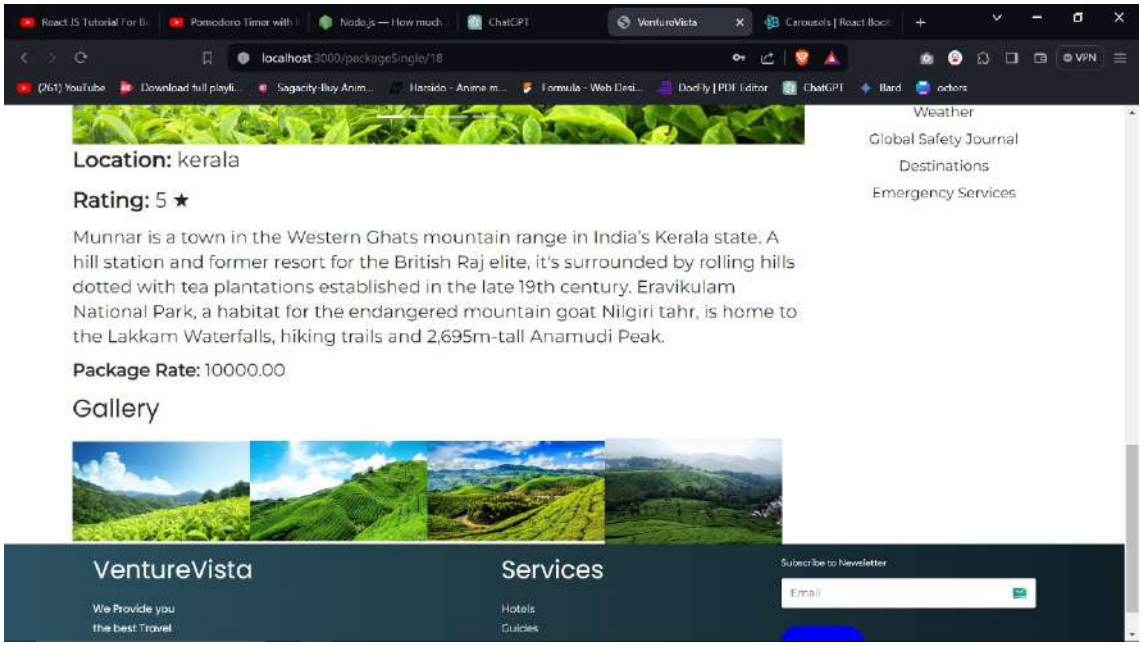
F.2 HOME


VentureVista
localhost:3000/home
VentureVista
Profile
Home
Packages
Gadgets
AKA
Discover New Destinations
A Team Of Devoted And Experienced Tourism Professionals
Will Provide You With The Best Advice And Tips.

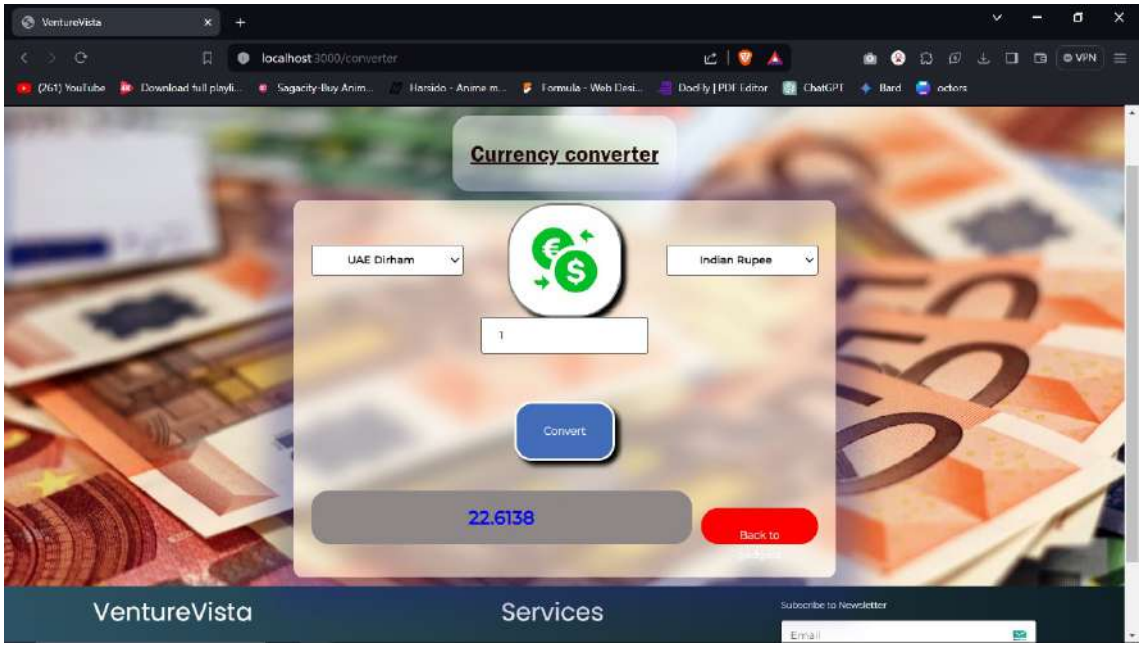
F.3 PACKAGES


VentureVista
localhost:3000/product
Athirappally
Athirappilly is a beautiful and scenic location located in the Thrissur district of Kerala, India. Here is a description of Athirappilly:
12.00
duration:2
Location:kattykkayam
rating:45
No_of_people:2
big b house
KOCHI: Mammootty's hit movie Big B, which was released in 2007, still has a strong fan following with the numbers increasing all the more with the announcement of its sequel Bilal. Fans are eagerly awaiting this Amal Neerad movie. Among the various props in the film that caught the fancy of the moviegoers, the house in which Bilal and his brothers stayed was the prime one.
12000.00
duration:2
Location:Kochi

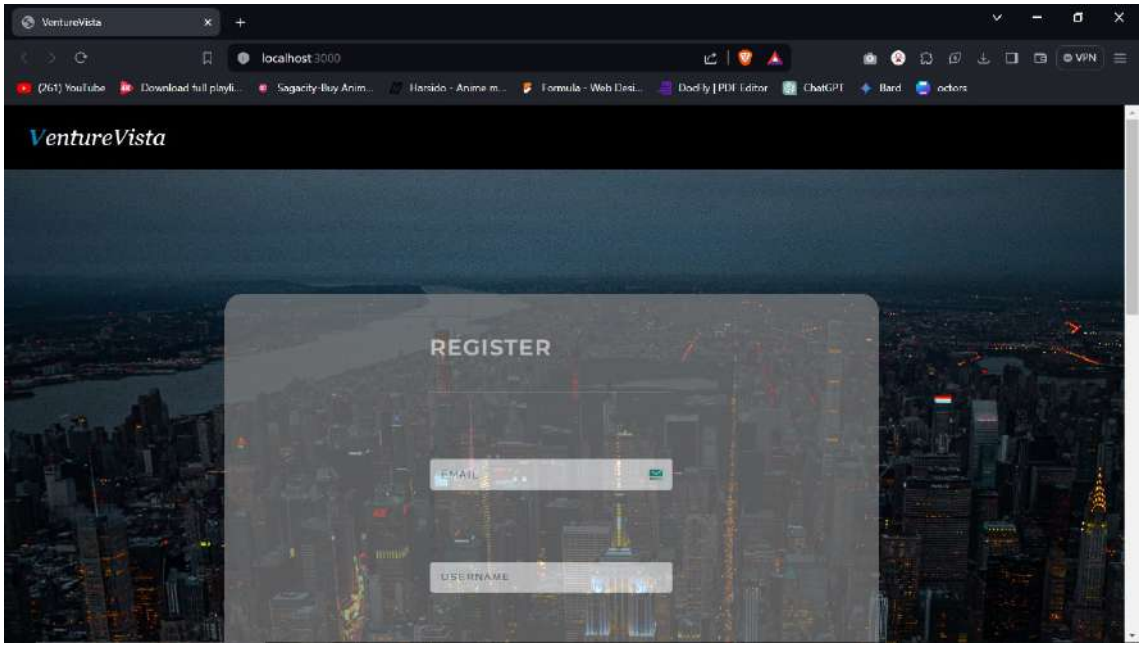

VentureVista
localhost:3000/packageSingle/6
Athirappally
Location: kattykkayam
Rating: 45 ★
Athirappilly is a beautiful and scenic location located in the Thrissur district of Kerala, India. Here is a description of Athirappilly:
Package Rate: 12.00
GADGETS
Currency Converter
Weather
Global Safety Journal
Destinations
Emergency Services
VentureVista
We Provide you the best Travel Experiences.
Services
Hotels
Guides
Travel Essentials
Tours
Subscribe to Newsletter
Email
Subcribe


localhost:3000/packageSingle/18
Weather
Global Safety Journal
Destinations
Emergency Services
Location: kerala
Rating: 5 ★
Munnar is a town in the Western Ghats mountain range in India's Kerala state. A hill station and former resort for the British Raj elite, it's surrounded by rolling hills dotted with tea plantations established in the late 19th century. Eravikulam National Park, a habitat for the endangered mountain goat Nilgiri tahr, is home to the Lakkam Waterfalls, hiking trails and 2,695m-tall Anamudi Peak.
Package Rate: 10000.00
Gallery
VentureVista
We Provide you the best Travel
Services
Hotels
Guides
Subscribe to Newsletter
Email

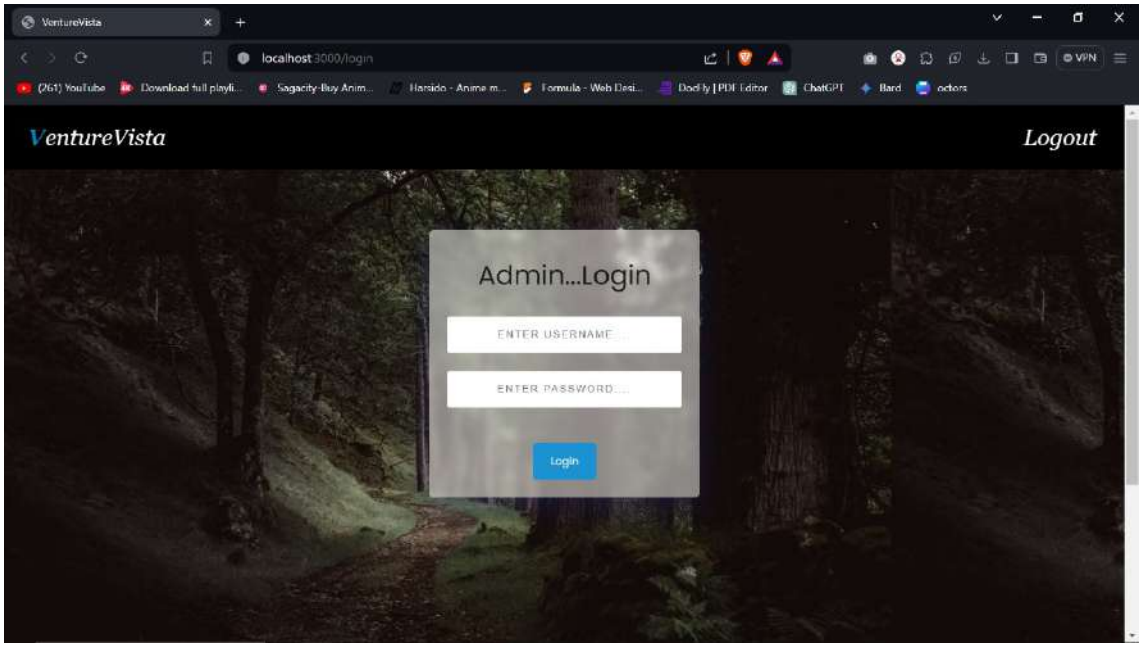
F.4 CURRENCY CONVERTER



F.5 REGISTER


VentureVista
localhost:3000
VentureVista
REGISTER
EMAIL
USERNAME

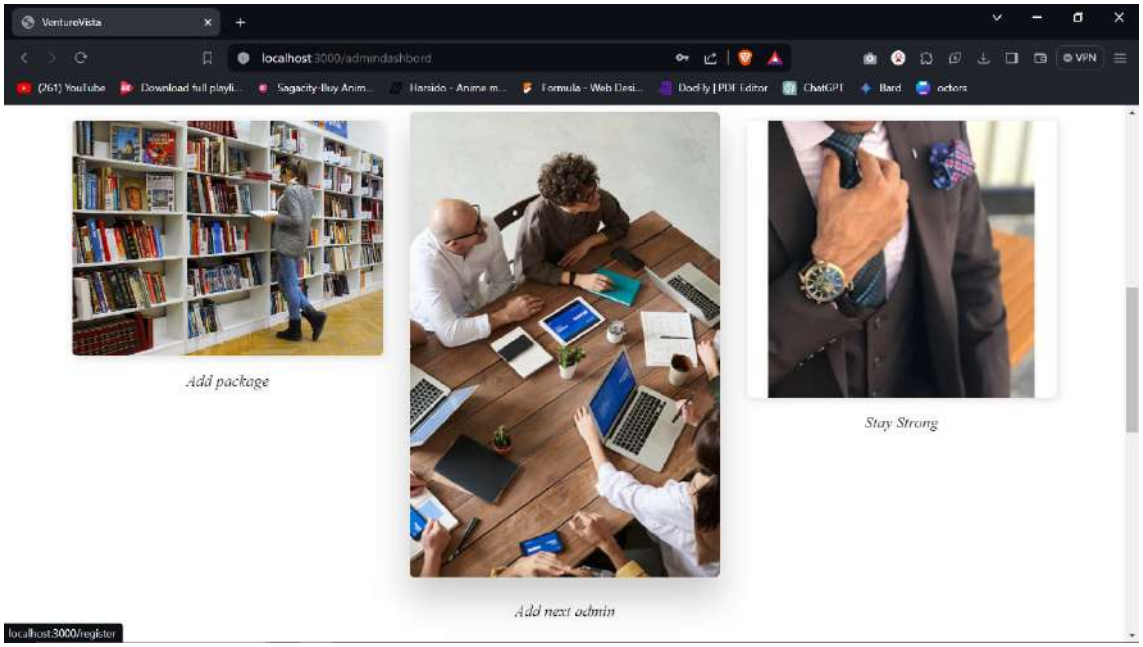
F.6 ADMIN LOGIN



F.7 ADMIN


VentureVista
localhost:3000/admindashboard
VentureVista
Logout
WELCOME MR.ADMIN
THE MAGICAL POWERES OF
YOU ARE....

F.8 ADMIN OPTIONS



G CODE

FRONTEND

```
//
// Source code recreated from a .class
file by
IntelliJ IDEA
// (powered by Fernflower decompiler)
//

import React, { useState, useEffect }
from
"react";
import { Modal, Button, Carousel } from
"react-bootstrap";
import { BrowserRouter, Route, Link }
from
"react-router-dom";
import Image from
"../images/athirappalli.jpeg";
import "./product.css";
function Product({ product }) {
const [show, setShow] = useState(false);
const handleClose = () =>
setShow(false);
const handleShow = () => setShow(true);  console.log(product)

  return (
    <>
      <div>
        <div className="product--card">
<img src={product.attraction}
alt="Product
Shoes" className="product--img" />
          <h4>{product.title}</h4>
          <p>
            {product.about}
          </p>
          <h2>{product.amount}</h2>
```

```
<p>duration:{product.duration}</p><p>Location:{product.location}
</p><p>rating:{product.rating}</p><p>
No_of_people:{product.no_of_people}</p>
        </div>
      </div>
    </>
  );
}

export default Product;
import React from 'react'
import './gadgets.css'
function AdminPowers() {
  return (
<body data-spy="scroll"
data-target=".onpage-navigation"
data-offset="60">
      <main>

<section class="bg-dark-30
showcase-page-header module parallax-bg"id='re' data-background="">
          <div class="titan-caption">
<div class="caption-content"><div class="font-alt mb-30
titan-title-size-1">Our best gadgets to
assits
you</div>
<div class="font-alt mb-40
titan-title-size-4">feel free to rush
trough
them</div>
            </div>
          </div>
        </section>
<div class="main showcase-page"><section class="module-medium"
id="demos"> <div class="container">
<div class="row multi-columns-row"><div
class="col-md-4 col-sm-6 col-xs-12"><a
class="content-box" href="/converter">
<div class="content-box-image"><img
src="https://blog.cdn.cmarix.com/blog/wp-content
```

```
/uploads/2023/07/develop-a-currency-converter-mobile-app.webp"
alt="Currency Converter" /></div>
<h3 class="content-box-title
font-serif">Currency
Converter</h3></a></div>
<div class="col-md-4 col-sm-6
col-xs-12"><a
class="content-box" href="Wheather">
<div class="content-box-image"><img
src="https://play-lh.googleusercontent.com/znc7FD1B
qyFR92b8n6hyROZrAR3FsTyV_ThO79hyuQG-Nb45z2qHFDTnf1HXH50DYg"
alt="Agency" /></div>
<h3 class="content-box-title
font-serif">wheather app</h3></a></div>
<div class="col-md-4 col-sm-6
col-xs-12"><a
class="content-box" href="/journal">
<div class="content-box-image"><img
src="https://s.yimg.com/ny/api/res/1.2/rx7Bjb7IghNsT35t13jufQ--
/YXBwaWQ9aGlnaGxhbmRlcjt3PTY0MDtoPTQ4MA--
/https://s.yimg.com/os/creatr-images/2020-05/45815870-9238-11ea-8fff-
1180915abec1"
alt="Portfolio" /></div>
<h3 class="content-box-title
font-serif">Global Safety
Journal</h3></a></div>
<div class="col-md-4 col-sm-6
col-xs-12"><a
class="content-box"
href="/destinations">
<div class="content-box-image"><img
src="https://images.pexels.com/photos/7235910/pexels-photo-
7235910.jpeg?auto=compress&cs=tinysrgb&w=300"
alt="Restaurant" /></div>
<h3 class="content-box-title
font-serif">Destinations</h3></a></div>
<div class="col-md-4 col-sm-6
col-xs-12"><a
class="content-box" href="/emergency">
<div class="content-box-image"><img
```

```
src="https://cdn.dribbble.com/userupload/7319839/
file/original-86f171450ede48437f8b1cfd8ac352f1.png?
crop=0x0-2880x2160&resize=400x300&vertical=center"
alt="Finance" /></div>
<h3 class="content-box-title
font-serif">Emergency
services</h3></a></div>
{/* <div class="col-md-4 col-sm-6
col-xs-12"><a class="content-box"
href="/product">
<div class="content-box-image"><img
src="https://cdn.dribbble.com/userupload/10920096/file/

original-4b166c456ae22ae068a4c961d6053a74.png?resize=400x0"
alt="Landing"/></div>
<h3 class="content-box-title
font-serif">Travel
packages</h3></a></div> */}
              </div>
            </div>
          </section>
        </div>
<div class="scroll-up"><a
href="#totop"><i
class="fa
fa-angle-double-up"></i></a></div>
      </main>
<script
src="assets/lib/jquery/dist/jquery.js"></script>
<script
src="assets/lib/bootstrap/dist/js/bootstrap.min.js"></script>
<script
src="assets/lib/wow/dist/wow.js"></script>
<script
src="assets/lib/jquery.mb.ytplayer/dist/jquery.mb.YTPlayer.js"></script>
<script
src="assets/lib/isotope/dist/isotope.pkgd.js"></script>
<script
src="assets/lib/imagesloaded/imagesloaded.pkgd.js"></script>
<script
```

```
src="assets/lib/flexslider/jquery.flexslider.js"></script>
<script
src="assets/lib/owl.carousel/dist/owl.carousel.min.js"></script>
<script
src="assets/lib/smoothscroll.js"></script>
<script
src="assets/lib/magnific-popup/dist/jquery.magnific-popup.js"></script>
<script
src="assets/lib/simple-text-rotator/jquery.simple-text-rotator.min.js"></script>
<script
src="assets/js/plugins.js"></script><script
src="assets/js/main.js"></script>
</body>

  )
}

export default AdminPowers
import React, { useState, useEffect }
from
"react";
import { useNavigate } from
"react-router-dom";
import axios from "axios";
import Loader from
"../components/Loader";
import Error from "../components/Error";import Success from
"../components/Success";
import "./login.css";
import { Link } from "react-router-dom";// import Loginimg from
"../images/Login.svg";function
LoginScreen() {

const [FormData, setFormData] =
useState({

  });
const navigate=useNavigate()
const [loading, setLoading] =
useState(false); const [error, setError]
```

```
= useState("");
const [success, setSuccess] =
useState("");
const handlechange=(e)=>{
  setFormData({
    ...FormData,
    [e.target.name]:e.target.value
  })
}
const Login=async(e)=>{
const {username,password}=FormData

await axios.post('admin-login/',{
  username,
  password
}).then((res)=>{
  if(res.status===200){
    console.log(res.status)
    if(res.status==200){
localStorage.setItem('adminkey',res.data.access)
localStorage.setItem('isadmin',res.data.is_superuser)
alert("admin login
successfull..welcome..."+res.data.username)
navigate('/admindashbord')
    }
    else{
alert("detected unautherized entry...")    }
    setLoading(false);
  }

}).catch(()=>{
alert("detected unauthorized entry..")  })
}

  return (
    <div className="login--screen">
{/* <img src={Loginimg} alt="login img"
className="login--img" /> */}
      {loading && <Loader></Loader>}
```

```
<div className="row
justify-content-center"> <div
className="col-md-5 mt-5">
{error.length > 0 && <Error
msg={error}></Error>}
          <div className="bs">
            <h2>Admin...Login</h2>

            <input
              type="text"
              className="form-control"
              name="username"
placeholder="Enter username...."
onChange={handlechange}
              style={{
                textAlign: "center",
                padding: "20px",
                marginBottom: "20px",
              }}
            />
            <input
              type="password"
              className="form-control"
placeholder="enter Password...."
name="password"
              onChange={handlechange}
              style={{
                textAlign: "center",
                padding: "20px",
                marginBottom: "20px",
              }}
            />
            {loading ? (
<div>Login...Please Wait...</div> ) : (
<button className="login--btn"
onClick={Login}>
                Login
              </button>
            )}
          </div>
```

```
        </div>
      </div>
    </div>
  );
}

export default LoginScreen;
```

BACKEND

```
//
// Source code recreated from a .class
file by
IntelliJ IDEA
// (powered by Fernflower decompiler)
//
from django.db import models

# Create your models here.


from django.contrib.auth.models import
AbstractUser
from django.db import models

class CustomUser(AbstractUser):

phone = models.CharField(max_length=15,
blank=True)
place = models.CharField(max_length=255,blank=True)
nickname =
models.CharField(max_length=30,
blank=True)
color = models.CharField(max_length=30,
blank=True)
image =
models.ImageField(upload_to='user_images/',
blank=True, null=True)
```

```
    def _str_(self):
        return self.username



class Package(models.Model):
admin = models.ForeignKey(CustomUser,
on_delete=models.CASCADE)
title = models.CharField(max_length=255)    about = models.TextField()
amount =
models.DecimalField(max_digits=10,
decimal_places=2)
    rating = models.IntegerField()
location =
models.CharField(max_length=255)duration
=
models.CharField(max_length=50)no_of_people
= models.CharField(max_length=50)hotel =
models.ImageField(upload_to='package_photos/',
blank=True, null=True)
destination =
models.ImageField(upload_to='package_photos/',
blank=True, null=True)
activity =
models.ImageField(upload_to='package_photos/',
blank=True, null=True)
attraction =
models.ImageField(upload_to='package_photos/',
blank=True, null=True)

    def _str_(self):
        return self.title


class Comments(models.Model):
user = models.ForeignKey(CustomUser,
on_delete=models.CASCADE)
package = models.ForeignKey(Package,
on_delete=models.CASCADE)
```

```
    comment = models.TextField()
created =
models.DateField(auto_now_add=True)
    def _str_(self):
        return self.comment


class Blogs(models.Model):
user = models.ForeignKey(CustomUser,
on_delete=models.CASCADE)
title = models.CharField(max_length=255)    description = models.TextField()
    rating = models.IntegerField()
food =
models.ImageField(upload_to='blog_photos/',
blank=True, null=True)
hotel =
models.ImageField(upload_to='blog_photos/',
blank=True, null=True)
travelling =
models.ImageField(upload_to='blog_photos/',
blank=True, null=True)
activity =
models.ImageField(upload_to='blog_photos/',
blank=True, null=True)
created =
models.DateField(auto_now_add=True)
    def _str_(self):
        return self.title

class Review(models.Model):

name =
models.CharField(max_length=255,blank=True,null=True)
email=models.CharField(max_length=255,blank=True,null=True)
    review = models.TextField()
created =
models.DateField(auto_now_add=True)
    def _str_(self):
        return self.review
"""
```

```
URL configuration for travel_guide
project.

The urlpatterns list routes URLs to
views. For
more information please see:
https://docs.djangoproject.com/en/5.0/topics/http/urls/
Examples:
Function views
1. Add an import: from my_app import
views2. Add a URL to urlpatterns:
path('',
views.home, name='home')
Class-based views
1. Add an import: from other_app.views
import
Home
2. Add a URL to urlpatterns: path('',
Home.as_view(), name='home')
Including another URLconf
1. Import the include() function: from
django.urls import include, path
2. Add a URL to urlpatterns:
path('blog/',
include('blog.urls'))
"""
from django.contrib import admin
from django.urls import path
from . import views
from rest_framework_simplejwt import
views as
jwt_views

urlpatterns = [
path('',views.UserRegistrationView.as_view(),
name='user-registration'),
path('admin-register/',
views.SuperuserRegistrationView.as_view(),
name='superuser-registration'),
path('user-login/',
```

```
views.UserloginView.as_view(),
name='user-token_obtain_pair'),
path('admin-login/',
views.AdminloginView.as_view(),
name='admin-token_obtain_pair'),
path('api/token/refresh/',
jwt_views.TokenRefreshView.as_view(),
name='token_refresh'),
path('user-details/',
views.UserDetailsView.as_view(),
name='user-details'),
path('update-user/',
views.UpdateUserDetailsView.as_view(),
name='update-user-details'),

path('send-otp/',
views.PasswordResetOTPSendView.as_view(),
name='update-user-details'),
path('otp-validation/',
views.OTPValidationView.as_view(),
name='otp-validation'),
path('change-password/',
views.ChangePasswordView.as_view(),
name='new-password'),

path('package-crud/',
views.PackageCRUDView.as_view()),
path('package-crud/<int:pk>/',
views.PackageCRUDView.as_view(),),

path('list-packages/',
views.ListPackagesView.as_view(),
name='list-packages'),
path('package-detail/<int:pk>/',
views.PackageDetailView.as_view(),
name='package-detail-view'),

path('create-comment/<int:pk>/',
views.CreateCommentView.as_view(),
name='create-comment'),
```

```
path('list-comments/<int:pk>/',
views.ListCommentsView.as_view(),
name='list-comments'),
path('delete-comment/<int:pk>/',
views.DeleteCommentView.as_view(),
name='delete-comment'),


path('create-blog/',
views.CreateBlogView.as_view(),
name='create-blog'),
path('list-blogs/',
views.ListBlogsView.as_view(),
name='list-blogs'),
path('blog-detail/<int:pk>/',
views.BlogDetailView.as_view(),
name='blog-detail'),

path('list-user-blogs/',
views.ListUserBlogsView.as_view(),
name='list-user-blogs'),
path('update-blog/<int:pk>/',
views.UpdateBlogView.as_view(),
name='update-blog'),
path('delete-blog/<int:pk>/',
views.DeleteBlogView.as_view(),
name='delete-blog'),


path('create-review/',
views.CreatReviewView.as_view(),),
path('list-reviews/',
views.ListReviewView.as_view(),),
path('review-detail/<int:pk>/',
views.ReviewDetailView.as_view(),),


]
from rest_framework import serializers
from .models import
```

```
CustomUser,Package,Comments,Blogs,Reviewfrom
rest_framework_simplejwt.serializers
import TokenObtainPairSerializer
from django.core.exceptions import
ObjectDoesNotExist


class
UserTokenObtainPairSerializer(TokenObtainPairSerializer):
    @classmethod
    def get_token(cls, user):
        token = super().get_token(user)
        # adding custom claims
token['username'] = user.usernametoken['is_superuser'] =
user.is_superuser return token

    def validate(self, attrs):
        data = super().validate(attrs)
        user = self.user
        if user.is_superuser==False:
data["is_superuser"] = user.is_superuser
data["username"] = user.username
            return data
        else:
raise serializers.ValidationError("Only
common
users are allowed to log in here.")


class
AdminTokenObtainPairSerializer(TokenObtainPairSerializer):
    @classmethod
    def get_token(cls, user):
        token = super().get_token(user)
        # adding custom claims
token['username'] = user.usernametoken['is_superuser'] =
user.is_superuser return token

    def validate(self, attrs):
        data = super().validate(attrs)
```

```
        user = self.user
        if user.is_superuser==True:
data["is_superuser"] =
user.is_superuserdata["username"] =
user.username return data
        else:
raise serializers.ValidationError(" Onlyadministrators are allowed to log in
here.")



class
CustomUserSerializer(serializers.ModelSerializer):
password_confirmation =
serializers.CharField(write_only=True)
email =
serializers.EmailField(required=True)
image=serializers.ImageField(max_length=None,use_url=True,required=False)
    class Meta:
        model = CustomUser
fields = ('id', 'username', 'email',
'phone',
'place', 'image',
'password','password_confirmation')
extra_kwargs = {'password':
{'write_only':
True}}

    def validate(self, data):
        try:
CustomUser.objects.get(email=data['email'])raise
serializers.ValidationError({'email':'Email
already exists.'})
        except ObjectDoesNotExist:
            pass

if data['password'] !=
data['password_confirmation']:
raise
serializers.ValidationError({'password_confirmation':"Passwords
```

```
do not match."})
        return data

    def create(self, validated_data):
validated_data.pop('password_confirmation',
None)
user =
CustomUser.objects.create_user(**validated_data)
        return user

class
CustomUserupdateSerializer(serializers.ModelSerializer):
image=serializers.ImageField(max_length=None,use_url=True,required=False)
    class Meta:
        model = CustomUser
fields =
['username','image','phone','place','email']

    def validate_email(self, value):
if
CustomUser.objects.filter(email__iexact=value)
.exclude(pk=self.instance.pk).exists():
raise serializers.ValidationError("This
email
address is already in use.")
        return value


class
PackageSerializer(serializers.ModelSerializer):
    class Meta:
        model = Package
        exclude = ['admin']

class
PackagelistSerializer(serializers.ModelSerializer):
    class Meta:
        model = Package
        fields = "_all_"
```

```
class
CommentSerializer(serializers.ModelSerializer):
    class Meta:
        model = Comments
        fields = ['comment']

class
CommentlistSerializer(serializers.ModelSerializer):
    class Meta:
        model = Comments
        fields = "_all_"

class
BlogSerializer(serializers.ModelSerializer):
    class Meta:
        model = Blogs
        exclude = ['user']

class
BloglistSerializer(serializers.ModelSerializer):
    class Meta:
        model = Blogs
        fields = "_all_"


class
ReviewSerializer(serializers.ModelSerializer):
    class Meta:
        model = Review
        fields = "_all_"

class
ReviewlistSerializer(serializers.ModelSerializer):
    class Meta:
        model = Review
        fields = "_all_"
from django.contrib import admin

# Register your models here
from .models import
```

```
CustomUser,Package,Comments,Blogs,Review# admin.site.register(CustomUser)
class
CustomUseryadmin(admin.ModelAdmin):
list_display =
['username','is_superuser']admin.site.register(CustomUser,CustomUseryadmin)

class Packageadmin(admin.ModelAdmin):
list_display =
['admin','title','amount','location','duration']
admin.site.register(Package,Packageadmin)

class Commentsadmin(admin.ModelAdmin):
list_display =
['user','comment','package']admin.site.register(Comments,Commentsadmin)

class Blogsadmin(admin.ModelAdmin):
    list_display = ['user','title']
admin.site.register(Blogs,Blogsadmin)



class Reviewadmin(admin.ModelAdmin):
    list_display = ['name','review']
admin.site.register(Review,Reviewadmin)
```

BREAST CANCER PREDICTION

PROJECT REPORT

Submitted By

STUART ROLINES

Reg. No. CCAVBCA047

For the award of the Degree of

Bachelor of Computer Application

(BCA)

(University of Calicut)

under the guidance of

Mr. Joju Sebastian

Assistant Professor

BACHELOR OF COMPUTER APPLICATION
DEPARTMENT OF COMPUTER SCIENCE
CHRIST COLLEGE
IRINJALAKUDA, KERALA

2021-2024

DEPARTMENT OF COMPUTER SCIENCE

CHRIST COLLEGE (AUTONOMOUS)

IRINJALAKUDA

CERTIFICATE

This is to certify that the project report entitled ***"BREAST CANCER PREDICTION"*** *is a bonafide record of the project work done by the* ***STUART ROLINES*** *in partial fulfillment of the requirement for the sixth semester of* ***Bachelor of Computer Application*** *in Department of Computer Science of* ***CHRIST COLLEGE (AUTONOMOUS) IRINJALAKUDA***

.

Mr.Joju Sebastian
Assistant Professor,CS
Internal Guide

Ms. Sini Thomas
Head of the Department
Computer Science

EXTERNAL EXAMINER

INTERNAL EXAMINER

DECLARATION

We here by declare that this project work **"BREAST CANCER PREDICTION"** submitted by Christ College (Autonomous)Irinjalakuda,affliated to Calicut University in partial fullfillment of the requirement for the award of the Bachelor of Computer Application, is a record of original work done by us,under the guidance of Mr.JOJU SEBASTIAN,Department of Computer Science.

Place: Irinjalakuda STUART ROLINES

ACKNOWLEDGEMENT

First and foremost we like to thank Lord almighty for his providence and for being the guiding light throughout the project. We wish to express my sincere gratitude to our beloved Department head for giving me all the facilities for our project. We take this opportunity to express my gratitude to the class teacher Ms. SOWMYA PS and head of the department Ms. SINI THOMAS who has been supported us throughout the course of this project. We are thankful for her aspiring guidance and valuable advice during the project work. We express my sincere thanks to my project guide Mr.JOJU SEBASTIAN for supporting and guiding throughout the project.We would take this opportunity to specially thank all other faculty members for their constant and continuous motivation. Finally we would like to thank my family and friends for giving valuable advice and moral support throughout our project.

ABSTRACT

BREAST CANCER PREDICTION is a desktop application that uses CNN algorithm to identify cancer and non cancerous cells in medica lorganization. This system automates the process of predicting cancer by inputting histopathological images and analyzing them to identify cancer and non cancer cells.

Contents

Chapter 1

1 Introduction

"Breast Cancer Prediction" introduces a revolutionary solution for predicting breast cancer, replacing traditional methods. This system incorporates deep learning algorithms to enhance the entire prediction process. The advantages of this project includes heightened accuracy, time efficiency, improved security, and reduced costs. As medical organizations embrace this user-friendly and innovative system, they are propelled into a more streamlined and technologically advanced future of breast cancer prediction.

.

1.1 Overview

The objective of the "Breast Cancer Prediction" project is to develop and implement a cutting-edge solution that automates breast cancer prediction using CNN algorithm. The goal is to replace traditional prediction methods with a more accurate, efficient, and secure solution, ultimately improving outcomes for patients and medical professionals.

Chapter 2

2 System Analysis

2.1 Purpose

Breast Cancer Prediction is a desktop applications that uses CNN algorithm to identify cancer and non-cancer cells in medical organizations. The system automates the process of predicting cancer by inputing histopathological images and analysing them to identify cancer and non-cancer cells. The system is typically composed of a software. The histopathological image of individuals browsed from the system are taken and send to the software, upon clicking predict button the image is analysed and compared with the trained dataset. The dataset contains information about each person's multiple histopathological images cells. The weights of the images are recorded and based on the weights of the new images the result is predicted.

2.1.1 Existing System

The existing system of the breast cancer prediction primarily relies on traditional prediction methods, such as using machine learning algorithms which predicts optimally the cancer and non-cancer cells from the trained histopathological dataset. Here, when a new histopathological image is taken the result accuracy is poor.

2.1.2 Proposed System

Breast Cancer Prediction is a desktop application that is in need for breast cancer identification on lab. Here, Project wise there is a GUI created that takes an image as input and classifies that it's a cancerous or non-cancerous. In this system when a new histopathological image is given here the result is predicted optimally and accuracy is good due to the use of CNN algorithm. Medically speaking Biopsy is taken place here .The histopathological images (Histology – Study of tissues , Pathology – Study of diseases) are taken from the monitor using advanced electron microscope by selecting a sample tissue and then processed and cut into thin sections then stained with Hematoxylin – Eosin (H&E).

2.2 Problem definition

The proposed project aims to tackle the limitation and inefficiency of traditional prediction methods by developing "Breast Cancer Prediction". The existing system predicts the cancer using a machine learning algorithm which only predicts the result based on the image given as input is the trained histopathological images which often lead to inaccuracies. To address these issues, this project will leverage deep learning method to provide a more accurate, secure, and

user-friendly approach to breast cancer prediction. By integrating seamlessly with existing systems and adapting to varying environmental conditions, this system promises to revolutionize breast cancer prediction, delivering enhanced accuracy, efficiency, and security for medical organizations.

2.3 FEASIBILITY STUDY

After the problem is clearly understood and solutions proposed, the next step is to conduct the feasibility study. Feasibility study is defined as evaluation or analysis of the potential impact of the proposed project or program. The objective is to determine whether the proposed system is feasible. There are three aspects of feasibility study to which the proposed system is subjected as discussed below.

2.3.1 Technical Feasibility

The proposed breast cancer prediction system using CNN algorithms is technically feasible with all required tools readily available. The system, implemented in Python, leverages CNN for its speed, efficiency, and low resource requirements. However, the accuracy of these algorithms depends on the number of trained images, necessitating thorough testing and optimization. The system requires a well-curated database of histopathological images, with regular updates and management crucial for accurate prediction. Robust measures for data protection and compliance with privacy regulations are essential due to the sensitive nature of the data involved. In conclusion, the technical feasibility of the system is evident, and with proper planning, testing, and attention to privacy and security, it can be a viable solution for various settings, enhancing operational efficiency.

2.3.2 Economical Feasibility

In this proposed system, all the tools used are free and open source. So that development cost is minimum. The hardware is used to run the system. It is built in our laptop. And also hardware requirement is feasible and not much breast cancer prediction system maintenance is required. The development of the system will not need a huge amount of money. It will be economically feasible. And the money spend for the application will be worth. Hence, the proposed system is economically feasible.

2.3.3 Operational Feasibility

The feasibility of a breast cancer prediction system using CNN algorithms is crucial.CNN's ease of implementation reduces development time and costs, and its low resource requirements allow deployment on various hardware configurations. The system requires a well-maintained dataset of pre-registered faces, regular updates, and management to accommodate user base changes. Addressing privacy and security concerns is critical due to the sensitive nature of

biometric data. User acceptance is a significant factor for successful implementation, requiring effective communication of benefits, addressing concerns, and providing adequate training. With proper planning, thorough testing, and compliance with privacy regulations, a breast cancer prediction system using CNN can predict optimal breast cancer results and enhance operational efficiency.

Chapter 3

3 Software Requirement Specification

3.1 Purpose

The breast cancer diagnosis system aims to leverage deep learning algorithms to accurately analyze histopathological images and automatically predict whether cancer is present. Key goals are achieving real-time cancer screening with minimal errors, extracting insightful image features, training robust models CNN, and obtaining analytics to uncover patterns in cancer data - ultimately enhancing efficiency, reliability and research capabilities compared to manual diagnosis processes. The system requires histopathology image datasets to train models to classify images as normal or malignant based on texture, shape and cell characteristics. It will provide a practical tool set to augment and automate parts of the breast cancer screening process.

3.2 Scope

The scope of this project encompasses the development of a software-based solution for automated breast cancer prediction from histopathological images. It leverages deep learning techniques to analyze cell structure and identify malignant indicators.

3.3 Overall Description

The desktop application caters to medical professionals, offering streamlined analysis of digitized histopathology slide images. Convolutional Neural Networks, it assesses cell morphology, texture, shape, and spatial characteristics to classify images as malignant or benign with confidence scores. The software guides users through preprocessing, feature extraction, model execution, and post-processing steps. Results, metrics, and visual heatmaps highlighting malignant regions are easily accessible. The back-end architecture includes a database, modeling module, processing pipelines, integration logic, and access control layers. Through rigorous training and testing with labeled datasets, the system ensures accuracy, aiding medical decision-making and enhancing patient care.

3.3.1 Product Perspective

The breast cancer prediction system aims to integrate smoothly with current medical systems, empowering healthcare professionals with accurate diagnostics. It aligns with organizational goals of enhancing patient care and operational efficiency, offering a user-friendly interface and scalability to adapt to diverse healthcare settings and technological advances.

3.3.2 Product Functionality

The breast cancer prediction system functions by analyzing histopathological images using CNN algorithm to classify cancerous and non-cancerous cells. Users input images into the system, which then processes them, compares them with a trained dataset, and generates predictions with high accuracy. The system offers a user-friendly interface for seamless interaction, contributes to faster and more accurate cancer detection, and integrates with existing medical infrastructure to enhance overall diagnostic capabilities.

3.3.3 Users and Characteristics

The breast cancer prediction system caters primarily to healthcare professionals involved in cancer diagnosis, such as doctors, radiologists, and pathologists. These users typically possess medical expertise and are familiar with diagnostic processes. They require a system that is intuitive, reliable, and capable of providing accurate predictions to support their clinical decision-making. Additionally, administrators responsible for implementing and maintaining the system within healthcare organizations may also interact with it, requiring a broader understanding of its functionality and integration capabilities.

3.4 Specific Requirements

3.4.1 Hardware Requirements

- System: IBM-Compatible PC
- Processor: Intel Core i3 or above
- Speed: Above 1GHz
- RAM capacity: 4 GB
- Hard Dsk drive: 500 GB
- Keyboard: Standard
- Mouse: Standard
- Monitor: SVGA Color

3.4.2 Software Requirements

- Front End:Python
- Back End: Python
- IDE: Anaconda

3.5 Functional Requirements

It contains three main modules.

- 1.Automated Cancer Prediction
- 2.Real-Time Prediction
- 3.User Interface

Automated Cancer Prediction

The system should automate the process of predicting breast cancer using CNN algorithms based on input histopathological images.

Real-Time Prediction

The system should offer real-time breast cancer prediction capabilities, enabling medical professionals to obtain results promptly.

User Interface

The user interface should be user-friendly and intuitive, allowing medical professionals to input histopathological images easily and view predicted results.

3.6 Non Functional Requirements

Non-functional requirements define the overall qualities or attributes of the resulting system. Non-functional system place restrictions on the product being developed, the developed process, and specify external constraints that the product must meet. Examples of non-functional requirements include safety, security, usability, reliability and performance requirements. Project management issues (costs, time and schedule) are often considered as non-functional requirements. The principal non - functional constraints which are relevant to critical systems :

- performance
- security
- safety
- usability

Performance

Performance requirements concern the speed of operation of a system. Types of performance requirements :

- Response requirements (how quickly the system reacts to a user input).

- Throughput requirements (how much the system can accomplish within a specified amount of time).
- Availability requirements (is the system available for service when requested by end users). The speed of operation of this system is adequate for the requirements.

Reliability

- Reliability is the ability of a system to perform its required function under stated conditions for a specified period of time.
- constraints on the runtime behavior of the system. This system is reliable because its functionalities can be done on the required conditions.

Safety

Safety requirements are not required which exclude unsafe situation from the possible solution of the system.

Usability

Usability is the ease with which a user can learn to operate, prepare inputs for, and interpret outputs of system or components. Usability requirements include :

- information error messages.
- well-formed user interfaces.

3.7 Interface Requirements

3.7.1 Hardware interfaces

The system must run over the internet, all the hardware shall require to connect internet will be hardware interface for the system. As for example modem, WAN-LAN.

3.7.2 Software interfaces

Software interface required for the working of the project is the appropriate operating system.

3.7.3 Communication interfaces

The user of the site communicate using the network connectivity and the data set is accessed

3.8 Security Requirements

- User accesses only their account.
- Validation of input is handled.
- This application containing the computer systems is physically secured against arms or surreptitious entry by intruders.
- Users must be authorized carefully to reduce changes of any such user giving access to an intruder in exchange for a bribe or other favour.

3.9 Platform Used

Windows 11 is the operating system chosen for the "Breast Cancer Prediction" project. Windows 10 provides a user-friendly interface and robust support for various hardware components and software applications. It is widely used in both personal and professional settings, making it a suitable choice for medical organizations implementing the breast cancer prediction system. Additionally, Windows 11 offers security features and regular updates to ensure system stability and reliability.

3.10 Technologies Used

PYTHON

Python is a versatile, high-level programming language known for its simplicity and readability. It offers dynamic typing and automatic memory management, making it suitable for various domains such as web development, data analysis, artificial intelligence, and automation. Python's extensive standard library provides pre-built modules and packages for tasks ranging from file manipulation to network programming. Its interpreted nature allows for rapid development and easy debugging. Python supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Its popularity is fueled by a vibrant community, extensive documentation, and numerous third-party libraries. Overall, Python's simplicity, flexibility, and robust ecosystem make it an ideal choice for beginners and seasoned developers alike

CNN

Convolutional Neural Networks (CNNs) are a class of deep neural networks primarily used for image recognition and computer vision tasks. They consist of multiple layers, including convolutional layers, pooling layers, and fully connected layers. CNNs excel at learning hierarchical representations of features from input data. Convolutional layers apply filters to input images, extracting features such as edges and textures. Pooling layers reduce spatial dimensions while preserving important features. Fully connected layers combine extracted features for classification or regression tasks. CNNs leverage parameter sharing and local connectivity to efficiently learn patterns from large datasets. Their success in image recognition tasks has made them a cornerstone in various fields, including autonomous driving, medical imaging, and facial recognition

Chapter 4

4 Design Document

4.1 Purpose

The purpose of the Design Document for the "Breast Cancer Prediction" project is to provide a concise yet comprehensive overview of the system's architecture, components, functionalities, and interactions. It serves as a roadmap for development, ensuring alignment with project goals and requirements. Additionally, the document facilitates communication and collaboration among team members and stakeholders, guiding informed decision-making and ensuring the successful development and implementation of the breast cancer prediction system.

4.2 Scope

The scope of the "Breast Cancer Prediction" project involves developing an automated breast cancer prediction system using deep learning algorithms. This includes designing a user-friendly interface for inputting histopathological images and implementing backend logic for image processing and prediction generation. Additionally, the project encompasses addressing data security and privacy concerns to ensure compliance with medical regulations and standards. Ultimately, the goal is to deliver a reliable and efficient system that aids medical professionals in accurately predicting breast cancer outcomes.

4.3 Overview

The Design Document for the "Breast Cancer Prediction" project provides an overview of the system architecture, functionalities, and key design considerations. It outlines the scope of the project, which involves developing an automated breast cancer prediction system using and deep learning algorithms. The document addresses the purpose of the project, which is to create a reliable and efficient system that aids medical professionals in accurately predicting breast cancer outcomes. Additionally, it highlights the importance of data security and privacy, as well as adherence to ethical and legal standards. Overall, the Design Document serves as a comprehensive blueprint for the development and implementation of the breast cancer prediction system.

4.4 Data Design

Database are the storehouses of data used in the software systems.The data is stored in tables inside the database.Several tables are created for the manipulation of the data for the system.Two essential settings for a database are

- Primary Key- The field that is unique for all the record occurrence.

- Foreign Key- The field used to set relation between tables.

Normalization

- Normalization is the process of organizing the data in the database.
- Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate the undesirable characteristics like Insertion, Update and Deletion Anomalies.
- Normalization divides the larger table into the smaller table and links them using relationship.
- The normal form is used to reduce redundancy from the database table.

Chapter 5

5 Development of the System

The "Breast Cancer prediction" system was developed using an agile approach, incorporating deep learning algorithms to provide an accurate and user-friendly solution for breast cancer prediction in medical organizations, replacing traditional prediction methods.

Chapter 6

6 System Testing

Testing is the penultimate step of software development.An elaborate testing of the data is prepared and the system is using the test data,while doing testing,errors are noted and correction is made.The users are trained to operate the developed system.Both hardware and software securities are made to run the developed system successfully.System testing is aimed at ensuring the system works accurately before the live operation commences.

6.1 Test Plan

6.1.1 Scope

This test plan will cover the following testing activities as identified in the testing strategy.

- White Box Testing

White box testing is a testing technique, that examines the program structure and derives test data from the program logic/code. In white box testing, the UI is bypassed. Inputs and outputs of ratings are tested directly at the code level and the results are compared against specifications. It Reveals errors in "hidden" code.

- Black Box Testing

Black - box testing is a method of software testing that examines the functionality of an application based on the specifications. Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end - user would.

- Unit Testing

The module includes four main components to undergo unit testing. It checks the syntax error, logic error and validity of the program correctness. This test will be performed by the developers.

- Integration Testing

After all components pass the unit testing, they get to be tested whether they work correctly when they are running concurrently and communicating to each other. Integration tests exercise an entire subsystem and ensure that a set of components play nicely together.

- Internal Data Testing

We will test the validity of the data before it enters the database to avoid any problems that may face us in the database. We will test the encryption of the personal information of all the users along with the user names and passwords to ensure maximum security of the users privacy.

6.1.2 Software risk issues

The "Breast Cancer Prediction" system leverages complex deep learning algorithms to predict breast cancer from histopathological images. While this provides enhanced accuracy compared to traditional methods, it also introduces certain software risks that need to be proactively managed. Key risks include model overfitting on limited training data, algorithmic limitations in capturing real-world variations, integration challenges between system components, performance bottlenecks when dealing with large images, lack of explainability of predictions, and potential security vulnerabilities given the sensitive nature of medical data. Careful software design, extensive testing and validation, performance tuning, security hardening, and adoption of explainable AI techniques are crucial to mitigate these risks. Additionally, bias in training data, software complexity, and obscure failures related to machine learning models require rigorous monitoring and maintenance processes. Overall, the advanced techniques used in Breast Cancer Prediction make software risk management an integral part of developing a robust and reliable solution.

6.1.3 Features to be tested

- Image Upload: Test uploading of histopathology images in different formats, sizes, and resolutions. Verify proper validation and preprocessing.
- User Interface: Test all UI flows and elements like buttons, forms, navigation, and data displays. Check for responsiveness across devices.
- Cancer Prediction: Test prediction accuracy with diverse sample images. Check for correct classification of cancer vs non-cancer.
- Model Training: Validate training process completed without errors. Confirm models are saved properly for prediction.
- Performance: Test system response times and resource usage under different normal and peak load conditions.
- Security: Validate only authorized access to prediction results. Check for data encryption and other security measures.
- Error Handling: Verify proper response for invalid inputs or missing images. Check for fail-safe behavior.
- Integration: Test seamless data flow between UI, preprocessing, and prediction components. Confirm no errors during integration.
- Compatibility: Validate working on different OS versions and hardware configurations per compatibility matrix.
- Functional Flows: Test end-to-end flows like user login, image upload, prediction, and result display for expected behavior.

6.2 Test consolidation

6.2.1 Test item

The items or features to be tested in the test cases are included in the document. Each and every user input is tested. The present condition of the system is tested. It is checked to make sure that environment is ready for the application to work.

6.2.2 Input specifications

These are the inputs required to execute the test case. All required inputs including data elements and values are given below.

Data	Values
Image	histopathological images

Chapter 7

7 System Implementation and Maintenance

7.1 Implementation

System implementation is the conversion of new system into an operating one which involves creating compatible files, training clients and installing hardware. User training is crucial for minimizing resistance to change and giving chance to prove its worth. Training aids user friendly manuals and healthy screens provide the user with a good start. Software maintenance follows conversion to the extent that changes are necessary to maintain satisfactory operations relative to changes in the user's environment. Maintenance often includes minor enhancements or corrections to the problem that surface late in the systems operations. In the implementation phase, the team builds the components either from scratch or by composition. Given the architecture document meant from the design phase and the requirement document from the analysis phase, the team should build exactly what has been requested, though there is still room for innovation and flexibility. For example, a component may be narrowly designed for this particular system, or the component may be made more general to satisfy a reusability

- Careful planning
- Investigation of system and constraints
- Design the methods to achieve changeover.
- Training the staff in the changed phase.
- Evaluation of change over method.
- The method of implementation and time scale to be adopted are found out initially.

7.2 Maintenance

This phase occurs as a result of deploying the whole system at the end users organization. They will perform the beta testing at the end users and inform to the developers about any needed modification to the application. The customer records all the problems that are encountered during the beta testing and reports these to the developer at regular intervals.

7.2.1 Corrective Maintenance

Even with the best quality assurance activities, it is likely that the customer will uncover defects in the software. Corrective maintenance changes the software to correct the defects. Corrective Maintenance activity may consist of repair,

restoration or replacement of equipment. This activity will be result of a regular inspection, which identifies the failure in time for corrective maintenance to be planned and scheduled, then performed during a routine maintenance shutdown.

7.2.2 Adaptive Maintenance

Over time, the original environment(CPU, operating system, business rules, external product characteristics) for which the software was developed is likely to change. Adaptive maintenance results in modification to the software to accommodate changes to its external environment.

7.2.3 Enhanced Maintenance

As software is used, the customer/user will recognize additional functions that will provide the benefit. Perfect maintenance extends the software beyond its original functional requirements.

7.2.4 Preventive Maintenance

Computer software deteriorates due to change, and because of this preventive maintenance often called software re-engineering, must be conducted to enable the software to serve the needs of its end users. Preventive maintenance makes changes to computer programs so that they can be more easily corrected, adapted and enhanced.

Chapter 8

8 Conclusion and Future Scope

8.1 Conclusion

Breast Cancer prediction systems have seen remarkable advancements, providing accurate and efficient solutions for cancer prediction. Looking ahead, the scope for further development is vast, with potential enhancements to enhance accuracy, security, and user experience. Strengthening privacy and security measures are essential aspects to consider. Here, any histopathological images can be given as inputs and optimal result is predicted.vThe integration of AI-based adaptive learning can enhance system accuracy, particularly in challenging environments. Breast cancer prediction systems will continue to be a vital and transformative tool for medical organizations and institutions in the future

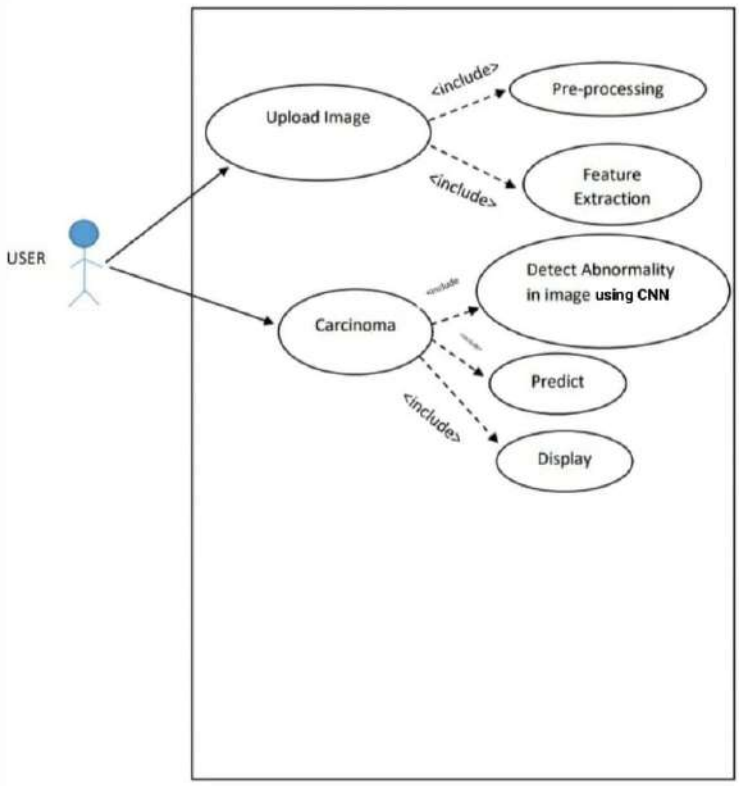
8.2 Future Scope

- Web based platform: Accessible to all the users where they interact with the doctor and results are available online.
- Privacy and Data Protection: Enhancing privacy features and adopting privacy-preserving algorithms will be crucial to safeguarding user data and complying with evolving data protection regulations.
- User Interface and Experience: Investing in user-friendly interfaces and seamless integration with existing systems will encourage greater user acceptance and adoption of the system.
- Real-time Analytics: Developing real-time attendance analytics can provide valuable insights into cancer prediction trends, enabling medical organizations to make data-driven decisions. More image should be trained or the accuracy may vary. Therefore, Several new models should be trained and the more f1 – score giving model should be taken in the future.
- Customization Options: Offering customization options to tailor the system to specific organizational needs and requirements will increase its versatility and adaptability.
- Cross-Platform Compatibility: Ensuring compatibility with various devices and operating systems will make the system accessible and widely applicable in different settings.

Appendix

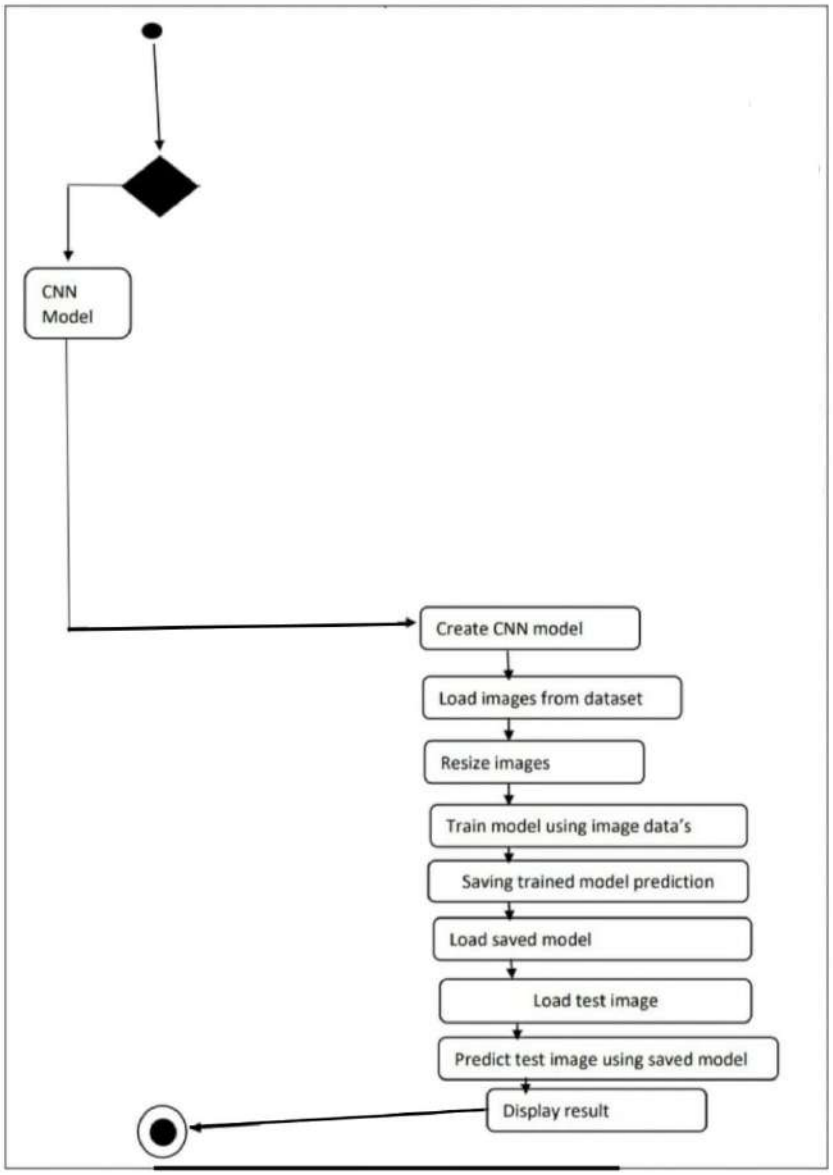
A Usecase Diagram

A.1



B Activity Diagram

B.1



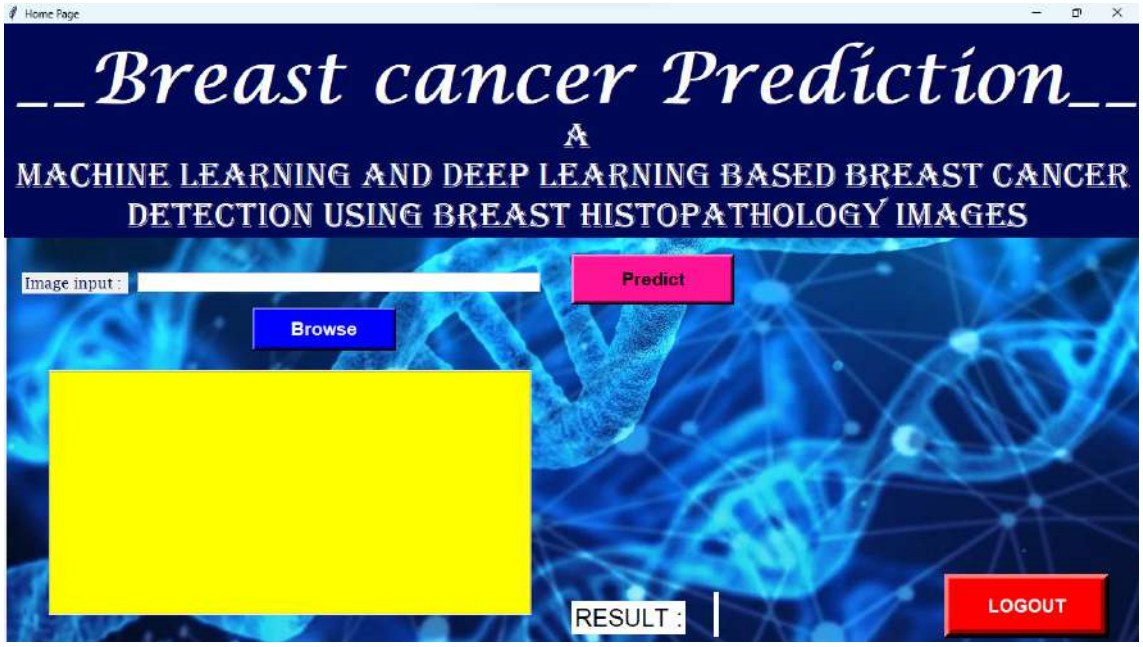
B.2

2. **Breast Cancer**

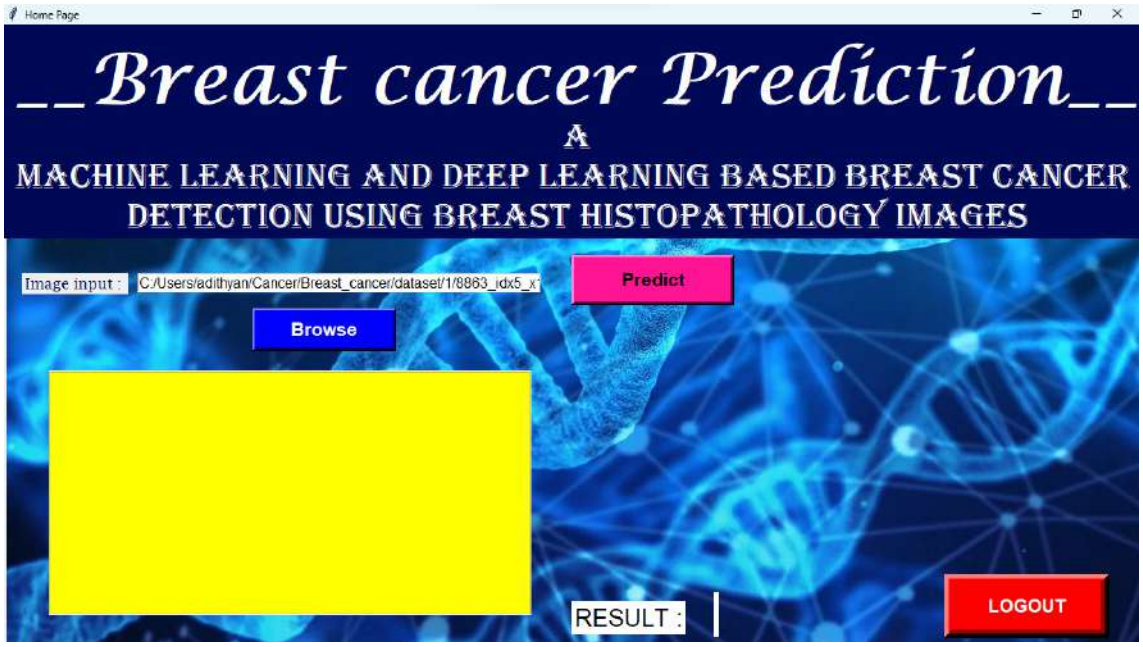
System	**Breast cancer**

C USER INTERFACES

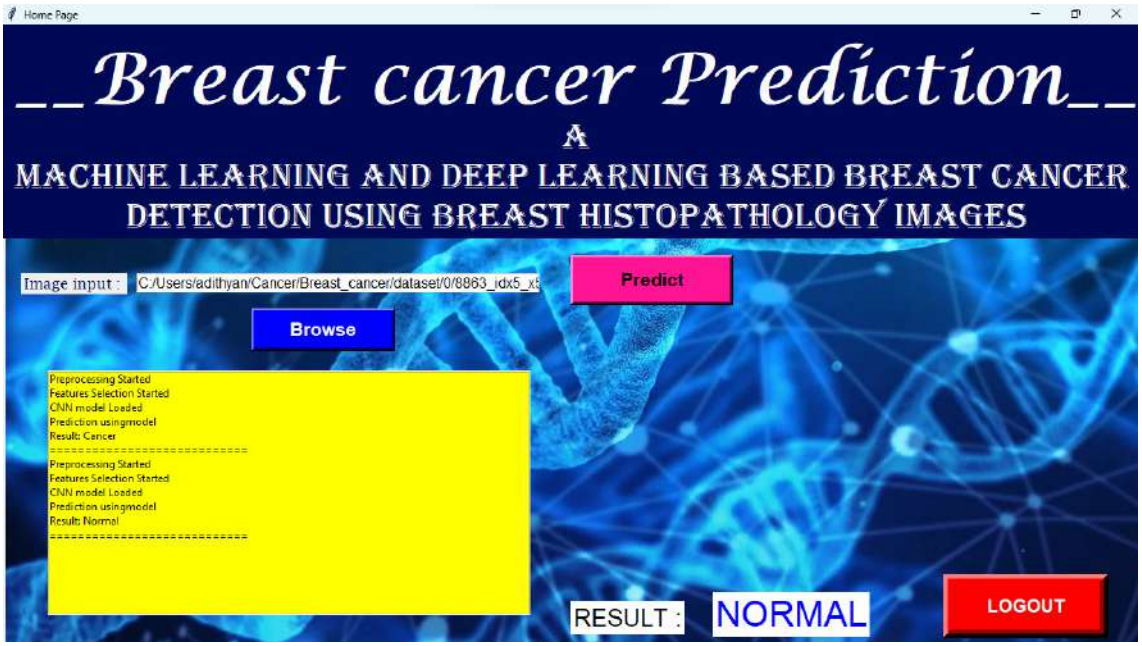
C.1 HOME



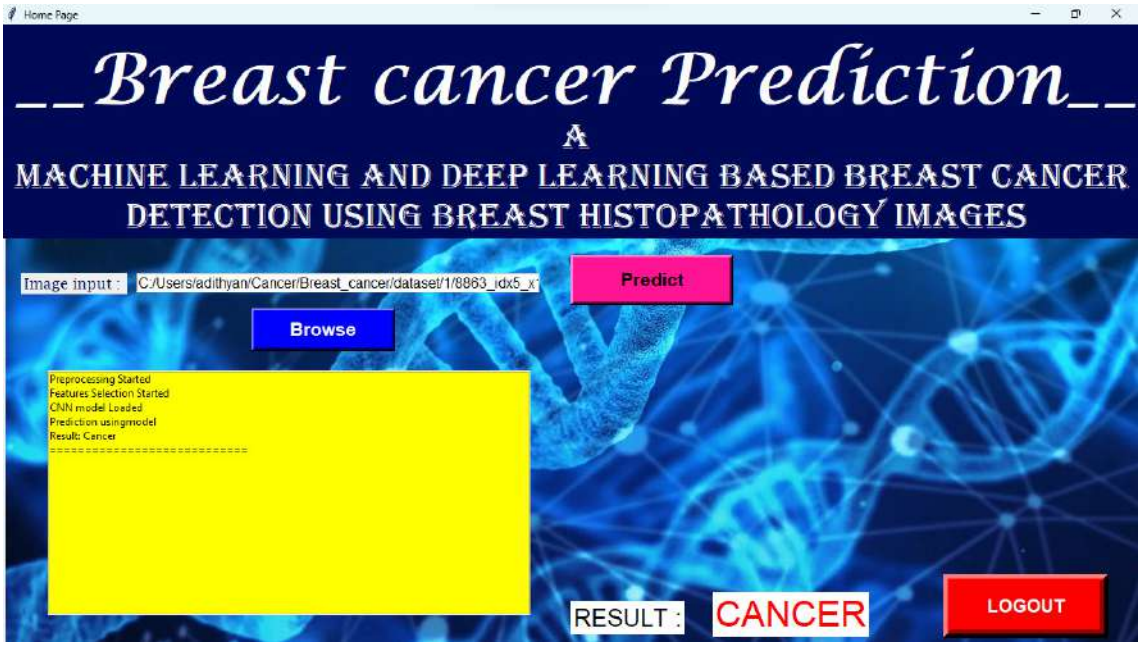
C.2 IMAGE LOCATION



C.3 NORMAL RESULT



C.4 CANCEROUS RESULT



D CODE

preprocess.py

```
import cv2

import numpy as np
import random
from imutils import paths
import os
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
import json
from tensorflow.keras.preprocessing.image import
    img_to_array
from tensorflow.keras.utils import to_categorical
from sklearn import metrics
import pickle

import mahotas as mt
from skimage.filters import gabor

def extract_features(gray):
    textures = mt.features.haralick(gray)
    # cv2.imshow("textures",textures)
    # cv2.waitKey(0)
        # take the mean of it and return it
    ht_mean  = textures.mean(axis=0)
    # print(ht_mean)
    return ht_mean


def build_filters():
        filters = []
        ksize = 31
        for theta in np.arange(0, np.pi, np.pi / 16):
            kern = cv2.getGaborKernel((ksize, ksize),
                4.0, theta, 10.0, 0.5, 0, ktype=cv2.CV_32F
                )
            kern /= 1.5*kern.sum()
            filters.append(kern)
            return filters

def process(img, filters):
        accum = np.zeros_like(img)
```

```
        for kern in filters:
            fimg = cv2.filter2D(img, cv2.CV_8UC3, kern)
            np.maximum(accum, fimg, accum)
            return accum


filters=build_filters()



data = []
label = []
print("[INFO] loading images...")
img_dir=sorted(list(paths.list_images("dataset")))
random.shuffle(img_dir)
print("[INFO]   Preprocessing...")
count=0
for i in img_dir:
    img = cv2.imread(i)
    img=cv2.resize(img, (64,64))
    img1=process(img,filters)
    gray = cv2.cvtColor(img1, cv2.COLOR_BGR2GRAY)
    res2=extract_features(gray)
    # print(res2)
    # print(type(res2))
    # imgdata=img_to_array(res2)
    data.append(res2)
    lb=i.split(os.path.sep)[-2]
    if(lb=="1"):
        label.append(1)
    else:
        label.append(0)
print(len(data))
print(len(label))
```

[breaklines=true]

SvmTrain.py

```
import cv2

import numpy as np
import random
from imutils import paths
import os
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
```

```
import json
from tensorflow.keras.preprocessing.image import
    img_to_array
from tensorflow.keras.utils import to_categorical
from sklearn import metrics
import pickle

import mahotas as mt
from skimage.filters import gabor

def extract_features(gray):
    textures = mt.features.haralick(gray)
    # cv2.imshow("textures",textures)
    # cv2.waitKey(0)
        # take the mean of it and return it
    ht_mean  = textures.mean(axis=0)
    # print(ht_mean)
    return ht_mean


def build_filters():
        filters = []
        ksize = 31
        for theta in np.arange(0, np.pi, np.pi / 16):
            kern = cv2.getGaborKernel((ksize, ksize),
                4.0, theta, 10.0, 0.5, 0, ktype=cv2.CV_32F
                )
            kern /= 1.5*kern.sum()
            filters.append(kern)
            return filters

def process(img, filters):
        accum = np.zeros_like(img)
        for kern in filters:
            fimg = cv2.filter2D(img, cv2.CV_8UC3, kern)
            np.maximum(accum, fimg, accum)
            return accum


filters=build_filters()



data = []
label = []
print("[INFO] loading images...")
```

```
img_dir=sorted(list(paths.list_images("dataset")))
random.shuffle(img_dir)
print("[INFO]   Preprocessing...")
count=0
for i in img_dir:
    img = cv2.imread(i)
    img=cv2.resize(img, (64,64))
    img1=process(img,filters)
    gray = cv2.cvtColor(img1, cv2.COLOR_BGR2GRAY)
    res2=extract_features(gray)
    # print(res2)
    # print(type(res2))
    # imgdata=img_to_array(res2)
    data.append(res2)
    lb=i.split(os.path.sep)[-2]
    if(lb=="1"):
        label.append(1)
    else:
        label.append(0)
print(len(data))
print(len(label))
lenofimage=len(data)
traindata=np.array(data).reshape(lenofimage,-1)
trainlabel=np.array(label)
print(traindata.shape)
print(trainlabel.shape)
X_train, X_test, y_train, y_test = train_test_split(
    traindata, trainlabel, test_size=0.25, random_state
    =42)
print(X_train.shape)
print(y_train.shape)
SVMclassifier = SVC(kernel='linear', random_state=0)
SVMclassifier.fit(X_train, y_train)
# Saving Trained Model
# dbfile=open("SVMmodel","wb")
# pickle.dump(SVMclassifier,dbfile)
# dbfile.close()
y_pred= SVMclassifier.predict(X_test)
print(y_pred)
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))

# 62.38
```

[breaklines=true] **Svmtest.py**

```
import cv2
```

```
import numpy as np
import random
from imutils import paths
import os
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
import json
from tensorflow.keras.preprocessing.image import
    img_to_array
from tensorflow.keras.utils import to_categorical
from sklearn import metrics
import pickle

import mahotas as mt
from skimage.filters import gabor



def extract_features(gray):
    textures = mt.features.haralick(gray)
    # cv2.imshow("textures",textures)
    # cv2.waitKey(0)
        # take the mean of it and return it
    ht_mean  = textures.mean(axis=0)
    # print(ht_mean)
    return ht_mean


def build_filters():
        filters = []
        ksize = 31
        for theta in np.arange(0, np.pi, np.pi / 16):
            kern = cv2.getGaborKernel((ksize, ksize),
                4.0, theta, 10.0, 0.5, 0, ktype=cv2.CV_32F
                )
            kern /= 1.5*kern.sum()
            filters.append(kern)
            return filters

def process(img, filters):
        accum = np.zeros_like(img)
        for kern in filters:
            fimg = cv2.filter2D(img, cv2.CV_8UC3, kern)
            np.maximum(accum, fimg, accum)
```

```
            return accum


# img=cv2.imread("im1.JPG")


filters=build_filters()


img = cv2.imread("31.png")
img=cv2.resize(img, (64,64))
img1=process(img,filters)
gray = cv2.cvtColor(img1, cv2.COLOR_BGR2GRAY)
res2=extract_features(gray)

dbfile=open("SVMmodel","rb")
adamodel=pickle.load(dbfile)
dbfile.close()
# Prediction based on selected model
y_pred = adamodel.predict(res2.reshape(1,-1))
print(y_pred)
```

[breaklines=true] **CNNTrain.py**

```
import cv2

import numpy as np
import random
from imutils import paths
import os
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
import json
from tensorflow.keras.preprocessing.image import
    img_to_array
from tensorflow.keras.utils import to_categorical
from sklearn import metrics
import pickle
import pandas as pd
import numpy as np
import os
from glob import glob
import random
import matplotlib.pylab as plt
from tensorflow.keras.models import model_from_json
```

```
from sklearn.model_selection import train_test_split
import tensorflow as tf
import keras
from keras.utils.np_utils import to_categorical
from keras.models import Sequential
from keras.preprocessing.image import ImageDataGenerator

from tensorflow.keras.optimizers import SGD, RMSprop,
    Adam, Adagrad, Adadelta
from keras.layers import Dense, Dropout, Activation,
    Flatten, BatchNormalization, Conv2D, MaxPool2D,
    MaxPooling2D
from tensorflow.keras.callbacks import EarlyStopping


data = []
label = []
print("[INFO] loading images...")
img_dir=sorted(list(paths.list_images("dataset")))
random.shuffle(img_dir)
print("[INFO]  Preprocessing...")
count=0
for i in img_dir:
    img = cv2.imread(i)
    img=cv2.resize(img, (64,64))
    imgdata=img_to_array(img)
    data.append(imgdata)
    lb=i.split(os.path.sep)[-2]
    if(lb=="1"):
        label.append(1)
    else:
        label.append(0)

traindata=np.array(data)
trainlabel=np.array(label)

X_train, X_test, y_train, y_test = train_test_split(
    traindata, trainlabel, test_size=0.25, random_state
    =42)
print(X_train.shape)
print(y_train.shape)

model = Sequential()

# Convolutional layer and maxpool layer 1
model.add(Conv2D(32,(3,3),activation='relu',input_shape
```

```
    =(64,64,3)))
model.add(MaxPool2D(2,2))

# Convolutional layer and maxpool layer 2
model.add(Conv2D(64,(3,3),activation='relu'))
model.add(MaxPool2D(2,2))

# Convolutional layer and maxpool layer 3
model.add(Conv2D(128,(3,3),activation='relu'))
model.add(MaxPool2D(2,2))

# Convolutional layer and maxpool layer 4
model.add(Conv2D(128,(3,3),activation='relu'))
model.add(MaxPool2D(2,2))

# This layer flattens the resulting image array to 1D
    array
model.add(Flatten())

# Hidden layer with 512 neurons and Rectified Linear Unit
     activation function
model.add(Dense(512,activation='relu'))

# Output layer with single neuron which gives 0 for Cat
    or 1 for Dog
#Here we use sigmoid activation function which makes our
    model output to lie between 0 and 1
model.add(Dense(1,activation='sigmoid'))

# model = Sequential()
# model.add(Conv2D(64, kernel_size=3, activation='relu',
    input_shape=(50,50,3)))
# model.add(Conv2D(32, kernel_size=3, activation='relu'))
# model.add(Flatten())
# model.add(Dense(2, activation="softmax"))
# adam = Adam(learning_rate=0.0001)
model.compile(loss="binary_crossentropy", optimizer="adam
    ", metrics=['accuracy'])

history = model.fit(
    X_train, y_train,
    validation_data=(X_test, y_test),
    epochs= 10,
    batch_size=10
    )
Y_pred = model.predict(X_test)
```

```
print(Y_pred)
#serialize model to JSON
model_json = model.to_json()
with open("CNNmodel.json", "w") as json_file:
        json_file.write(model_json)
# serialize weights to HDF5
model.save_weights("CNNmodelw.h5")
print("Saved model to disk")
#0.9133
```

[breaklines=true]
CNNTest.py

```
from tensorflow.keras.preprocessing.image import
    img_to_array
from tensorflow.keras.models import load_model
import numpy as np
import argparse
import imutils
import cv2
from tensorflow.keras.models import model_from_json


img = cv2.imread("10.png")
img=cv2.resize(img, (64,64))
img = img_to_array(img)
##img = img.reshape(1, 100, 36, 1)

json_file = open('CNNmodel.json', 'r')
loaded_model_json = json_file.read()
json_file.close()
model = model_from_json(loaded_model_json)
# load weights into new model
model.load_weights("CNNmodelw.h5")
print("Loaded model from disk")
img = np.expand_dims(img, axis = 0)
result = model.predict(img)
res1=result[0][0]
if(res1>0.32):
    print("Cancer")
else:
    print("Normal")
print("res=>",result[0][0])
result=np.argmax(result[0][0])
print(result)
```

[breaklines=true] **Resnet_train.py**

```
import numpy as np
import random
import cv2
from glob import glob
import matplotlib.pyplot as plt
from imutils import paths
import os
from PIL import Image
import keras
from tensorflow.keras.preprocessing.image import
    ImageDataGenerator
from tensorflow.keras.applications import ResNet50
from tensorflow.keras.applications.resnet50 import
    preprocess_input
from tensorflow.keras import Model, layers
from tensorflow.keras.models import load_model,
    model_from_json
from tensorflow.keras.preprocessing.image import
    img_to_array
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.models import model_from_json

from sklearn.model_selection import train_test_split



conv_base = ResNet50(
    include_top=False,
    weights='imagenet')

for layer in conv_base.layers:
    layer.trainable = False
x = conv_base.output
x = layers.GlobalAveragePooling2D()(x)
x = layers.Dense(128, activation='relu')(x)
predictions = layers.Dense(2, activation='softmax')(x)
model = Model(conv_base.input, predictions)

model.compile(loss='sparse_categorical_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])


data = []
label = []
```

```
print("[INFO] loading images...")
img_dir=sorted(list(paths.list_images("dataset")))
random.shuffle(img_dir)
print("[INFO]   Preprocessing...")
count=0
for i in img_dir:
    img = cv2.imread(i)
    img=cv2.resize(img,(256,256))
    imgdata=img_to_array(img)
    data.append(imgdata)
    lb=i.split(os.path.sep)[-2]
    if(lb=="0"):
        label.append(0)
    else:
        label.append(1)
print(len(data))
print(len(label))
lenofimage=len(data)
traindata=np.array(data)
trainlabel=np.array(label)
train_labels = to_categorical(trainlabel, num_classes=2)
print(traindata.shape)
print(train_labels.shape)
X_train, X_test, y_train, y_test = train_test_split(
    traindata, trainlabel, test_size=0.25, random_state
    =42)
print(X_train.shape)
print(y_train.shape)
print(X_test.shape)
print(y_test.shape)
```

```
model.fit(X_train, y_train, epochs=3, validation_split
    =0.2, batch_size=5)

model_json = model.to_json()
with open("resnet_model.json", "w") as json_file:
        json_file.write(model_json)
# serialize weights to HDF5
model.save_weights("resnetw_model.h5")
print("[INFO] Saved model to disk")
y=model.predict(X_train)
print(y)
```

[breaklines=true] **Densenettrain.py**

```
import numpy as np
import random
import cv2
from glob import glob
import matplotlib.pyplot as plt
from imutils import paths
import os
from PIL import Image

import keras
from tensorflow.keras.preprocessing.image import
    ImageDataGenerator
from tensorflow.keras.applications.densenet import
    DenseNet121
from keras import Model, layers
from keras.models import load_model, model_from_json
from tensorflow.keras.preprocessing.image import
    img_to_array
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.models import model_from_json
from tensorflow.keras.layers import Dense,
    GlobalAveragePooling2D,Convolution2D,
    BatchNormalization
from tensorflow.keras.layers import Flatten,MaxPooling2D,
    Dropout
from sklearn.model_selection import train_test_split




from tensorflow.keras.layers import Dense, Dropout, Input
    ,Flatten, Conv2D, MaxPool2D, BatchNormalization,
    AveragePooling2D, GlobalAveragePooling2D
from tensorflow.keras.models import Model,Sequential,
    load_model
from tensorflow.keras.optimizers import Adam
def build_densenet():
    densenet = DenseNet121(weights='imagenet',
        include_top=False)

    input = Input(shape=(256, 256, 3))
```

```
    x = Conv2D(3, (3, 3), padding='same')(input)

    x = densenet(x)

    x = GlobalAveragePooling2D()(x)
    x = BatchNormalization()(x)
    x = Dropout(0.5)(x)
    x = Dense(256, activation='relu')(x)
    x = BatchNormalization()(x)
    x = Dropout(0.5)(x)

    # multi output
    output = Dense(15,activation = 'softmax', name='root
        ')(x)


    # model
    model = Model(input,output)

    optimizer = Adam(lr=0.002, beta_1=0.9, beta_2=0.999,
        epsilon=0.1, decay=0.0)
    model.compile(loss='sparse_categorical_crossentropy',
         optimizer=optimizer, metrics=['accuracy'])
    model.summary()

    return model

model = build_densenet()



data = []
label = []
print("[INFO] loading images...")
img_dir=sorted(list(paths.list_images("dataset")))
random.shuffle(img_dir)
print("[INFO]   Preprocessing...")
count=0
for i in img_dir:
    img = cv2.imread(i)
    img=cv2.resize(img,(256,256))
    imgdata=img_to_array(img)
    data.append(imgdata)
    lb=i.split(os.path.sep)[-2]
    if(lb=="0"):
        label.append(0)
```

```
    else:
        label.append(1)
print(len(data))
print(len(label))
lenofimage=len(data)
traindata=np.array(data)
trainlabel=np.array(label)
train_labels = to_categorical(trainlabel, num_classes=2)
print(traindata.shape)
print(train_labels.shape)
X_train, X_test, y_train, y_test = train_test_split(
    traindata, trainlabel, test_size=0.25, random_state
    =42)
print(X_train.shape)
print(y_train.shape)
print(X_test.shape)
print(y_test.shape)


model.fit(X_train, y_train, epochs=10, validation_split
    =0.2, batch_size=5,verbose=1)

# model_json = model.to_json()
# with open("densenet_model.json", "w") as json_file:
#         json_file.write(model_json)
# # serialize weights to HDF5
# model.save_weights("densenetw_model.h5")
# print("[INFO] Saved model to disk")
```

[breaklines=true] **GUI_new.py**

```
from tkinter import *
import time
import re
#Import scikit-learn metrics module for accuracy
    calculation
import pickle
from PIL import Image, ImageTk
import cv2
from tkinter.filedialog import askopenfile
from tensorflow.keras.preprocessing.image import
    img_to_array
from tensorflow.keras.models import load_model
import numpy as np
import argparse
import imutils
```

```
import cv2
from tensorflow.keras.models import model_from_json
from tensorflow.keras.preprocessing.image import
    img_to_array
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.applications.vgg16 import
    preprocess_input
from tensorflow.keras.models import model_from_json

json_file = open('resnet_model.json', 'r')
loaded_model_json = json_file.read()
json_file.close()
modelres = model_from_json(loaded_model_json)
# load weights into new model
modelres.load_weights("resnetw_model.h5")
print("Loaded Resnet model from disk")

json_file = open('densenet_model.json', 'r')
loaded_model_json = json_file.read()
json_file.close()
modeldense = model_from_json(loaded_model_json)
# load weights into new model
modeldense.load_weights("densenet.h5")

print("Loaded Sensenet model from disk")

json_file = open('CNNmodel.json', 'r')
loaded_model_json = json_file.read()
json_file.close()
cnnmodel = model_from_json(loaded_model_json)
# load weights into new model
cnnmodel.load_weights("CNNmodelw.h5")
print("Loaded model from disk")

def pp(a):
    global mylist
    mylist.insert(END, a)



def predict(val):
    print(val)
    img = cv2.imread(val)
    imgr=cv2.resize(img, (64,64))
    imgarr = img_to_array(imgr)
    ##img = img.reshape(1, 100, 36, 1)
```

```
imgarr = np.expand_dims(imgarr, axis = 0)
result = cnnmodel.predict(imgarr)
res1=result[0][0]
print(res1)

if(res1>0.32):
    cnnres=1
else:
    cnnres=0
print("cnnpred==>",cnnres)
imgres=cv2.resize(img,(256,256))
imgdata=img_to_array(imgres)
print(imgdata)
# print(type(imgdata))
# print(imgdata.shape)

train_ds= np.expand_dims(imgdata, axis=0)
# print(train_ds.shape.)

yres=modelres.predict(train_ds)
print(yres)
resres=np.argmax(yres[0])
print("respred==>",resres)

ydense=modeldense.predict(train_ds)
print(ydense)
denseres=np.argmax(ydense[0])
print("densepred==>",denseres)

reslist=[cnnres,resres,denseres]

print("result list==>",reslist)
finalres=max(reslist)
if(finalres==1):
    resc="Cancer"
    print("Cancer")
    root.after(3100, lambda :shrslt.config(text="
        CANCER",fg="red"))

else:
    resc="Normal"
    print("Normal")
    root.after(3100, lambda :shrslt.config(text="
        NORMAL",fg="blue"))
```

```
    root.after(500, lambda : pp("Preprocessing Started ")
        )
    root.after(2000, lambda : pp("Features Selection
        Started "))
    root.after(2200, lambda : pp("CNN model Loaded"))
    root.after(2400, lambda : pp("Prediction usingmodel
        "))
    root.after(2600, lambda : pp("Result: "+resc))
    root.after(2800, lambda : pp
        ("=========================================="))


def browseim():
    global cimg,shrslt,E1
    path = askopenfile()
    n=path.name
    print(path)
    E1.delete(0,"end")
    E1.insert(0, n)

#def upload_file():
 #    global cimg,shrslt,E1
 #    root=Tk()
 #    f_types=[('Png Files','*.png')]
 #    filename=filedialog.askopenfilename(filetypes=
     f_types)
 #   cimg=ImageTk.PhotoImage(file=filename)

def userHome():
    global root, mylist,shrslt,E1
    root = Tk()
    root.geometry("1400x625+0+0")
    root.title("Home Page")

    image = Image.open("dna.png")
    image = image.resize((1500, 725), Image.ANTIALIAS)
    pic = ImageTk.PhotoImage(image)
    lbl_reg=Label(root,image=pic,anchor=CENTER)
    lbl_reg.place(x=0,y=0)

    #----------------------INFO TOP----------------
    lblinfo = Label(root, font=( 'lucida calligraphy'
        ,61, 'bold' ),text="__Breast cancer Prediction__",
        fg="white",bg="#000955",bd=10,anchor='w')
```

```
lblinfo.place(x=0,y=0)

lblinfo2 = Label(root, font=( 'Algerian' ,31 ),text="
    A\n Machine learning and Deep learning based
    Breast cancer \n   detection using Breast
    Histopathology Images   ",fg="white",bg="#000955",
    anchor='w')
lblinfo2.place(x=0,y=100)
lblinfo3 = Label(root, font=( 'Lucida fax' ,0 ),text
    ="Image input : ",fg="#000955",anchor='w')
lblinfo3.place(x=20,y=280)
E1 = Entry(root,width=50,font="Will&Grace")
E1.place(x=150,y=280)
mylist = Listbox(root,width=90, height=17,bg="yellow
    ")

mylist.place( x = 50, y = 390 )
btntrn=Button(root,padx=10,pady=2, bd=4 ,fg="white",
    font=('ariel' ,16,'bold'),width=10, text="Browse",
     bg="blue",command=lambda:browseim())
btntrn.place(x=280, y=320)
# btntrn=Button(root,padx=10,pady=2, bd=4 ,fg="white",
   font=('ariel' ,16,'bold'),width=10, text="image",
   bg="red",command=lambda:upload_file())
#btntrn.place(x=700, y=400)
btnhlp=Button(root,padx=40,pady=6, bd=4 ,fg="black",
    font=('ariel' ,16,'bold'),width=7, text="Predict",
     bg="deep pink",command=lambda:predict(E1.get()))
btnhlp.place(x=640, y=260)



rslt = Label(root, font=( 'aria' ,20, ),text="RESULT
    :",fg="black",bg="white",anchor=W)
rslt.place(x=640,y=650)
shrslt = Label(root, font=( 'aria' ,30, ),text="",fg
    ="blue",bg="white",anchor=W)
shrslt.place(x=800,y=640)

def qexit():
    root.destroy()

btnexit=Button(root,padx=16,pady=8, bd=10 ,fg="white
    ",font=('ariel' ,16,'bold'),width=10, text="LOGOUT
    ", bg="red",command=qexit)
btnexit.place(x=1060, y=620)
```

```
    root.mainloop()

userHome()
```

[breaklines=true]

WEBSITE OF ABILITY CONNECT

PROJECT REPORT

Submitted By

ADHITHYAN T J

Reg. No. CCAVBCA017

For the award of the Degree of

Bachelor of COMPUTER APPLICATION(BCA)

in Computer Science

(University of Calicut)

under the guidance of

Ms. Rasmi P M

Assistant Professor

BACHELOR OF COMPUTER APPLICATION
DEPARTMENT OF COMPUTER SCIENCE
CHRIST COLLEGE
IRINJALAKUDA, KERALA

2021-2024

DEPARTMENT OF COMPUTER SCIENCE

CHRIST COLLEGE (AUTONOMOUS)

IRINJALAKUDA

CERTIFICATE

This is to certify that the project report entitled ***"Website of Ability Connect"*** *is a bonfied record of the project work done by* ***ADHITHYAN T J*** *in partial fulfillment of the requirement for the sixth semester of* ***Bachelor of Computer Science*** *in Department of Computer Science of* **CHRIST COLLEGE (AUTONOMOUS) IRINJALAKUDA**

Ms. Rasmi P M
Assistant Professor,CS
Internal Guide

Ms. Sini Thomas
Head of the Department
Computer Science

EXTERNAL EXAMINER

INTERNAL EXAMINER

DECLARATION

We here by declare that this project work **"WEBSITE OF ABILITY CONNECT"** submitted by Christ College (Autonomous)Irinjalakuda,affliated to Calicut University in partial fullfillment of the requirement for the award of the Bachelor of Computer Application, is a record of original work done by us,under the guidance of Ms.RASMI P M,Department of Computer Science.

Place: Irinjalakuda ADHITHYAN T J

ACKNOWLEDGEMENT

First and foremost we like to thank Lord almighty for his providence and for being the guiding light throughout the project. We wish to express my sincere gratitude to our beloved Department head for giving me all the facilities for our project. We take this opportunity to express my gratitude to the class teacher Ms.SOWMYA P.S and head of the department Ms.SINI THOMAS who has been supported us throughout the course of this project. We are thankful for her aspiring guidance and valuable advice during the project work. We express my sincere thanks to my project guide Ms.RASMI P M for supporting and guiding throughout the project.We would take this opportunity to specially thank all other faculty members for their constant and continuous motivation. Finally we would like to thank my family and friends for giving valuable advice and moral support throughout our project.

ABSTRACT

WEBSITE OF ABILITY CONNECT is a innovative website introduced as a learning platform for differently abled students in assistance of Computer Science Department of Christ College(Autonomous) Irinjalakuda. The website is enriched with two kind of login facilities - student login, teachers login. It is a learning platform and the main features are- student registration, learning and exam dashboards and so on. All these features make this website more adaptable and user-friendly.

Contents

Chapter 1

1 Introduction

Education is the cornerstone of empowerment and growth, yet education systems often fail to accommodate the diverse needs of students with various disabilities .Our website endeavors to help the learning process by crafting inclusive assessments specifically tailored to each student's abilities, while also introducing a spectrum of learning methods—audio, text, video, and interactive activities that can help foster a learning environment.

.

1.1 Overview

The objective of the Ability Connect Website is to design an simple and adaptable website that helps the users in their learning process by tailoring different modes through which they can access education . Different modes of examinations, in-website voice navigation and other features helps website to be user friendly and create a learning environment for differently abled students .

Chapter 2

2 System Analysis

2.1 Purpose

The main purpose of the website owned by Don Davis, Adhithyan T.J, Leyon T.John ,Mithra prothasis is to make a user friendly website as a inclusive learning platform for diffrently abled students.

2.1.1 Existing System

In the existing system, there may be limited opportunities for disabled children to learn or develop their skills independently, especially at a young age. While schools and special education programs may offer support from teachers and other aides, there may be gaps in providing opportunities for self-directed learning and skill development outside of structured classroom settings. Additionally, parents of disabled children may face challenges in providing supplemental learning experiences or resources at home due to lack of time, knowledge, or access to suitable materials. Limitations of existing system : Self-directed learning and skill development outside of structured classroom settings and difficulties in accesing diffrent modes of learning according to disabilities.

2.1.2 Proposed System

The materials will be designed to be accessible to children with various disabilities. Users can specify any disabilities or special needs they have to ensure that the platform can customize them.The website will incorporate learning technologies to adjust the difficulty level and pacing of activities based on the student's performance and progress. This ensures that each student is appropriately challenged and supported throughout their learning journey. Proposed system aims to create an inclusive and empowering learning environment for disabled children. The website will provide feedback to students on their performance and progress, including scores, achievements, and areas for improvement. Parents or teachers may also have access to progress reports to monitor the student's development and provide additional support as needed. Other Features : A message system that sends notification to user mail about the new contents or exams assigned.

2.2 Problem definition

To know what the problem is and what the needs are before developing it. Here we design a website as a learning platform for differently abled students.

2.3 Feasibility Study

After the problem is clearly understood and solutions proposed, the next step is to conduct the feasibility study. Feasibility study is defined as evaluation or analysis of the potential impact of the proposed project or program. The objective is to determine whether the proposed system is feasible. There are three aspects of feasibility study to which the proposed system is subjected as discussed below.

2.3.1 Technical Feasibility

Technical feasibility assess whether the current technical resources are sufficient for the new system. We can upgrade the level of technology for supporting our website. We check whether the proposed system can be implemented in the present system without supporting the existing hardware.

2.3.2 Economical Feasibility

Economic feasibility determines whether the time and money are available to develop the system. There is no additional hardware used to develop the site, it is inexpensive to build.

2.3.3 Operational Feasibility

Operational feasibility determines if the human resources are available to operate the system once it has been installed. No extra training is needed to use this system. Anyone who has knowledge in internet in english language can easily use the system. The resources that are required to implement or install are already available with the organization.

Chapter 3

3 Software Requirement Specification

3.1 Purpose

The purpose of creating an educational website for differently-abled students is multifaceted and revolves around addressing the unique needs and challenges that these students may face. The educational website for differently-abled students is to create an inclusive, accessible, and supportive online learning environment that empowers individuals with diverse abilities to thrive academically and personally. The project aims to bridge gaps, break down barriers, and contribute to a more inclusive educational landscape.

3.2 Scope

The scope of the "Educational Website for Differently-Abled Students" project is a comprehensive initiative aimed at developing an inclusive and accessible online learning platform. The project will focus on catering to the unique needs of differently-abled students, educators, and administrators, providing a supportive and engaging environment for learning. It also aims to create a holistic and inclusive educational platform that not only meets the immediate needs of its users but also lays the groundwork for ongoing growth and enhancement.

3.3 Overall Description

The project for developing an "Educational Website for Differently-Abled Students" is a visionary initiative with the primary goal of creating an inclusive and accessible online learning platform. This comprehensive website is designed to cater to the diverse educational needs of students with varying abilities, ensuring they have equal access to quality learning resources and a supportive community. At the heart of the project is the creation of an adaptive learning environment. The website will feature a dynamic system that tailors educational content to the individual needs, preferences, and learning styles of differently-abled students.

3.3.1 Product Perspective

The "Educational Website for Differently Abled Students" involves considering how the website fits into the broader context of educational technology and the needs of its users.

3.3.2 Product Functionality

Through this website teachers can upload study materials and quizzes as a part of exam and teacher can see the result of each student. The students get notified

if new material or new test been assigned. And the main highlight is that this website can be voice controlled specially for blind students.

3.3.3 Users and Characteristics

There are two types of users admin or teacher and student. In admin page the teacher can upload new study materials and tests, teacher can also see the growth of each student. And teachers can give suggestions for each type of student. In the student page they can see all the study materials and tests. There will be an entertainment section where there will be small stories, contents and games.

3.4 Specific Requirements

3.4.1 Hardware Requirements

- System: IBM-Compatible PC
- Processor: Intel Core i3 Or Above
- Speed: Above 1GHz
- RAM capacity: 4 GB Or Above
- Hard Disk Drive: 256 GB Or Above

3.4.2 Software Requirements

- Front End : HTML, CSS, Javascript
- Back End : Python
- Database : Sqlite3
- IDE : Visual Studio Code

3.5 Functional Requirements

It contains two main modules.

- 1.Teacher/Admin
- 2.Student

Admin

An admin account is used for editing or managing the website dynamically by admin panel. The admin can add new study materials, new tests and they can also review student performance and assign necessary suggestions according to their performance. The admin will be acting as teacher role.

Student

The user or the student can login the website, and can access the study materials assignd by the admin or teacher. The will be tests, suggestions, and small entertainment section for leisure time.

3.6 Non Functional Requirements

Non-functional requirements define the overall qualities or attributes of the resulting system. Non-functional system place restrictions on the product being developed, the developed process, and specify external constraints that the product must meet. Examples of non-functional requirements include safety, security, usability, reliability and performance requirements. Project management issues (costs, time and schedule) are often considered as non-functional requirements. The principal non - functional constraints which are relevant to critical systems :

- performance
- security
- safety
- usability

Performance

Performance requirements concern the speed of operation of a system. Types of performance requirements :

- Response requirements (how quickly the system reacts to a user input).
- Throughput requirements (how much the system can accomplish within a specified amount of time).
- Availability requirements (is the system available for service when requested by end users). The speed of operation of this system is adequate for the requirements.

Reliability

- Reliability is the ability of a system to perform its required function under stated conditions for a specified period of time.
- constraints on the runtime behavior of the system. This system is reliable because its functionalities can be done on the required conditions.

Safety

Safety requirements are not required which exclude unsafe situation from the possible solution of the system.

Usability

Usability is the ease with which a user can learn to operate, prepare inputs for, and interpret outputs of system or components. Usability requirements include :

- information error messages.
- well-formed user interfaces.

3.7 Interface Requirements

3.7.1 Hardware interfaces

The system must run over the internet, all the hardware shall require to connect internet will be hardware interface for the system. As for example modem, WAN-LAN.

3.7.2 Software interfaces

Software interface required for the working of the project is the appropriate operating system.

3.7.3 Communication interfaces

The user of the site communicate using the network connectivity and the data set is accessed

3.8 Security Requirements

- User accesses only their account.
- Validation of input is handled.
- This application containing the computer systems is physically secured against arms or surreptitious entry by intruders.
- Users must be authorized carefully to reduce changes of any such user giving access to an intruder in exchange for a bribe or other favour.

3.9 Platform Used

Windows 10 is a major version of the Microsoft Windows operating system that was released on July 29, 2015.It is built on the Windows NT kernel and follows windows 8.Part of the reason Microsoft decided to name the 2015 release "Windows 10"(and skipped "windows 9")is because the operating system is designed to be a new direction for Microsoft. One of the primary aims of windows 10 is to Unify the windows experience across multiple devices, such desktop computers, tablets, and smartphones. As part of this effort, Microsoft developed Windows10 Mobile alongside Windows 10 to replaces Windows Phone

- Microsoft's previous mobile OS. Windows 10 also integrates other Microsoft services, such as Xbox Live and the Cortana voice recognition assistant.

3.10 Technologies Used

JavaScript

JavaScript is a programming language that enables interactivity and dynamic behavior on web pages. It can be used to manipulate the HTML and CSS of a webpage, handle user interactions, fetch data from servers, and much more.HTML and CSS can be embedded with JavaScript to create a working efficient website. HTML is the standard markup language for creating web pages. It provides the structure of a webpage by using elements or tags to define different parts of the content. CSS is used to style the HTML elements and define their appearance on the webpage. It allows you to control the layout, colors, fonts, and other visual aspects of your website.

Sqlite3

SQLite is a lightweight, serverless, self-contained, and embedded SQL database engine. It's widely used in various applications due to its simplicity, efficiency, and ease of integration. In the context of a website, SQLite can be utilized to store and manage data on the server-side, providing persistence for web applications. The Sqlite libraries provide APIs for performing CRUD (Create, Read, Update, Delete) operations, executing SQL queries, and managing database connections. By integrating SQLite into your website, you can efficiently store and manage data, enabling features like user authentication, data persistence, and dynamic content generation.

Chapter 4

4 Design Document

4.1 Purpose

The purpose of the website is to provide accessible and inclusive educational resources for disabled students with visual and auditory impairments. The website aims to support these students in their studies by offering:

- Accessible Study Materials: The website provides notes and educational materials in various accessible formats, such as voice notes for blind students and video/photo classes for students with hearing impairments.
- Interactive Learning Tools: Students can engage with interactive learning tools tailored to their specific needs, including audio descriptions for visual content and subtitles or sign language interpretation for video content.
- Assessments and Tests: The website offers assessments and tests designed to accommodate the needs of disabled students, ensuring fair evaluation of their understanding and progress.
- Teacher Support and Suggestions: Teachers can provide personalized support and suggestions to disabled students through the platform, offering guidance and assistance to help them succeed academically.
- Voice Command Input: For blind students, the website incorporates voice command input functionality, allowing them to navigate the platform and interact with content using voice commands, enhancing accessibility and usability.

4.2 Scope

The website aims to provide accessible study materials, including voice notes for blind students and video/photo classes for those with hearing impairments. It facilitates interactive learning through tests, offers teacher suggestions, and incorporates voice command input for blind users, fostering an inclusive educational environment for disabled students.

4.3 Overview

The purpose of this document is to help the reader to visualize the solution to the project presented.This document verifies how the design meet the requirement stimulated in the SRS document through design viewpoints.The design viewpoints will cover all design elements presented before. This document will provide a direct approach to the development of this project hence reducing feature creep and ponitedly determine the quality of the design.

4.4 Data Design

Database are the storehouses of data used in the software systems.The data is stored in tables inside the database.Several tables are created for the manipulation of the data for the system.Two essential settings for a database are

- Primary Key- The field that is unique for all the record occurrence.
- Foreign Key- The field used to set relation between tables.

Normalization

- Normalization is the process of organizing the data in the database.
- Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate the undesirable characteristics like Insertion, Update and Deletion Anomalies.
- Normalization divides the larger table into the smaller table and links them using relationship.
- The normal form is used to reduce redundancy from the database table.

Tables

Login

Name	DataType	Constraints	Description
username	varchar(100)	Notnull	Username of student
password	varchar(100)	Notnull	Password of student

Student

Name	DataType	Constraints	Description
studentid	Charfield(50)	Primarykey	ID of users
firstname	Charfield(100)	Notnull	First name of users
lastname	Charfield(100)	Notnul	Last name of users
email	Emailfield)	Notnull	Email of the users
gender	Charfield(100)	Notnull	Gender of the user
age	PositiveIntegerfield	Notnull	Age of user
disability	Charfield(100)	Notnull	Disability of user
access technology	Charfield(200)	Notnul	Users access technology

Question Table

Name	DataType	Constraints	Description
type	Charfield	Not Null	Question Type
text_id	Charfield	Foreignkey	Text Question
image	Filefield	Foreignkey	Image Question
audio	Filefield	Foreignkey	Audio Question

Scoremodel Table

Name	DataType	Constraints	Description
student	Charfield	Foreignkey	Student Name
score	Integerfield	Foreignkey	Exam Score
category	Charfield	Foreignkey	Category
suggestion	Textfield	Not Null	Suggestions

Suggestions

Name	DataType	Constraints	Description
suggestion	Textfield	Notnull	Suggestions
category	Charfield	Foreignkey	Category
video	Filefield	Foreignkey	Suggested Video
audio	Filefield)	Foreignkey	Suggested audio

Chapter 5

5 Development of the System

The website will feature a user-friendly interface with options for visually impaired students to access voice notes and hearing-impaired students to engage in video and photo classes. It will include an integrated testing platform, teacher feedback system, and voice command functionality for blind students, aiming to enhance accessibility and support disabled students in their studies.

Chapter 6

6 System Testing

Testing is the penultimate step of software development.An elaborate testing of the data is prepared and the system is using the test data,while doing testing,errors are noted and correction is made.The users are trained to operate the developed system.Both hardware and software securities are made to run the developed system successfully.System testing is aimed at ensuring the system works accurately before the live operation commences.

6.1 Test Plan

6.1.1 Scope

This test plan will cover the following testing activities as identified in the testing strategy.

- White Box Testing

White box testing is a testing technique, that examines the program structure and derives test data from the program logic/code. In white box testing, the UI is bypassed. Inputs and outputs of ratings are tested directly at the code level and the results are compared against specifications. It Reveals errors in "hidden" code.

- Black Box Testing

Black - box testing is a method of software testing that examines the functionality of an application based on the specifications. Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end - user would.

- Unit Testing

The module includes four main components to undergo unit testing. It checks the syntax error, logic error and validity of the program correctness. This test will be performed by the developers.

- Integration Testing

After all components pass the unit testing, they get to be tested whether they work correctly when they are running concurrently and communicating to each other. Integration tests exercise an entire subsystem and ensure that a set of components play nicely together.

- Internal Data Testing

We will test the validity of the data before it enters the database to avoid any problems that may face us in the database. We will test the encryption of the personal information of all the users along with the user names and passwords to ensure maximum security of the users privacy.

6.1.2 Software risk issues

In this section, the plan is to test the risk involved in critical issues such as: Difficult to run on visual studio due to large loading time. Some inherent software risks such as complexity exist; also these issues need to be identified. Proper network connection Working of Mysql database

6.1.3 Features to be tested

- Test whether correct user name and password allows you to login.
- Test whether invalid user name and password prevents you from login.
- Test whether there is any connection problem in the Server.
- Test whether the student and admin details are entered correctly.
- Test whether invalid data entry allows saving data successfully.
- Test whether all pages are loaded correctly.
- Test whether watermark is embedded and extracted properly.
- Test whether the provided security works correctly.

6.2 Test consolidation

6.2.1 Test item

The items or features to be tested in the test cases are included in the document. Each and every user input is tested. The present condition of the system is tested. It is checked to make sure that environment is ready for the application to work.

6.2.2 Input specifications

These are the inputs required to execute the test case. All required inputs including data elements and values are given below.

Data	Values
Email	A valid email ID
Password	Combination of characters and numbers
Phone number	10 Digit number

Chapter 7

7 System Implementation and Maintenance

7.1 Implementation

System implementation is the conversion of new system into an operating one which involves creating compatible files, training clients and installing hardware. User training is crucial for minimizing resistance to change and giving chance to prove its worth. Training aids user friendly manuals and healthy screens provide the user with a good start. Software maintenance follows conversion to the extent that changes are necessary to maintain satisfactory operations relative to changes in the user's environment. Maintenance often includes minor enhancements or corrections to the problem that surface late in the systems operations. In the implementation phase, the team builds the components either from scratch or by composition. Given the architecture document meant from the design phase and the requirement document from the analysis phase, the team should build exactly what has been requested, though there is still room for innovation and flexibility. For example, a component may be narrowly designed for this particular system, or the component may be made more general to satisfy a reusability

- Careful planning
- Investigation of system and constraints
- Design the methods to achieve changeover.
- Training the staff in the changed phase.
- Evaluation of change over method.
- The method of implementation and time scale to be adopted are found out initially.

7.2 Maintenance

This phase occurs as a result of deploying the whole system at the end users organization. They will perform the beta testing at the end users and inform to the developers about any needed modification to the application. The customer records all the problems that are encountered during the beta testing and reports these to the developer at regular intervals.

7.2.1 Corrective Maintenance

Even with the best quality assurance activities, it is likely that the customer will uncover defects in the software. Corrective maintenance changes the software to correct the defects. Corrective Maintenance activity may consist of repair,

restoration or replacement of equipment. This activity will be result of a regular inspection, which identifies the failure in time for corrective maintenance to be planned and scheduled, then performed during a routine maintenance shutdown.

7.2.2 Adaptive Maintenance

Over time, the original environment(CPU, operating system, business rules, external product characteristics) for which the software was developed is likely to change. Adaptive maintenance results in modification to the software to accommodate changes to its external environment.

7.2.3 Enhanced Maintenance

As software is used, the customer/user will recognize additional functions that will provide the benefit. Perfect maintenance extends the software beyond its original functional requirements.

7.2.4 Preventive Maintenance

Computer software deteriorates due to change, and because of this preventive maintenance often called software re-engineering, must be conducted to enable the software to serve the needs of its end users. Preventive maintenance makes changes to computer programs so that they can be more easily corrected, adapted and enhanced.

Chapter 8

8 Conclusion and Future Scope

8.1 Conclusion

There are three types of users that interact with the system Admin,Teacher and Student.Each of these three types has different uses of the system so each of them has their own panel.Admin can manage all the features of website dynamically by login on admin panel.Teacher can view the details about their event assigned by the admin and publish the result of the event.And the student can register for events and view the results of the events.

8.2 Future Scope

- Advanced adaptive learning technologies can be implemented.
- Advanced AI Chatbots

Appendix

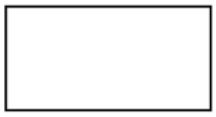
A Data Flow Diagram

Data flow is the one of the best way of documenting the entire functionality of the system.For the system ,which will have data flows in and have some processing inside and then some data flow out from the system can be documented or represented effectively by means of data flow out from the system can be documented or represented effectively by means of data flow diagrams. The data flow diagram are a diagrammatic representation of the system,which has input,process and output.Once any system is represented using a data flow diagram we can identify the following things easily:

- Various entities interacting with the system are identified
- Flow of data from one entity to another is identified
- The various processes involved in between the interaction of two or more entities in the system are clearly pointed out
- The various data stores which hold the data in between the process,are clearly identified

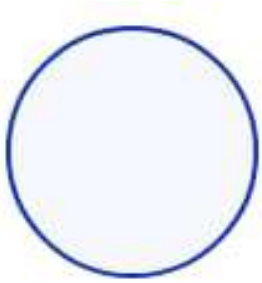
Some Data Flow Diagram charting forms:

A.1 External source or receiver



A source or sink is a person or part of organization ,which enters or receives information from the system,but is considered to be outlining the contest of data flow model.

A.2 Transform process

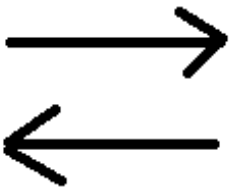


A process represents transformation where incoming data flowsare changed into outgoing data flow

A.3 Data Store

A data store is repository of data that is to be stored for use by one or more process may be as simple as buffer or queue or sophisticated as relational database. They should have clear names.If a process merely uses the contest of store and does not alter it,the arrowhead goes only form the store to the process. If a process alters the details in the store then double-headed arrow is used.

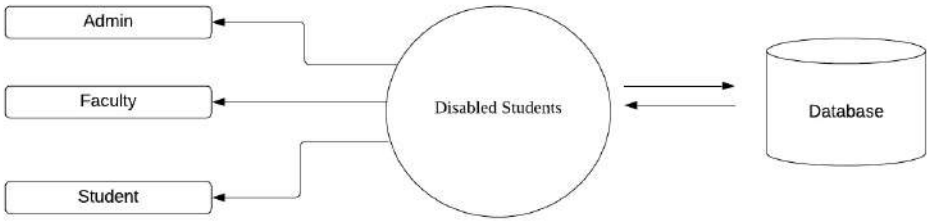
A.4 Data flow



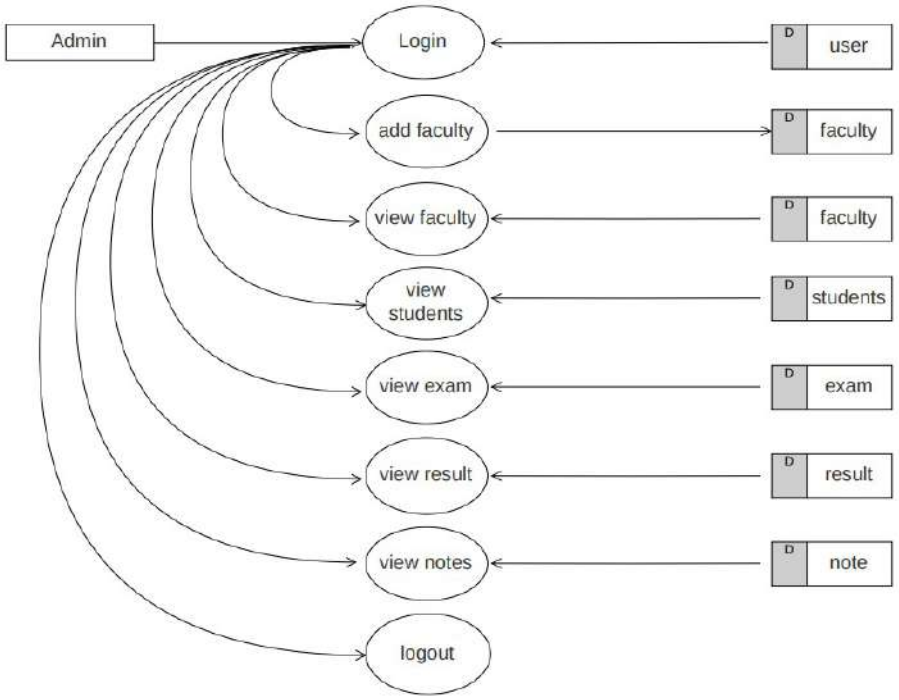
A data flow is a route, which enable packets of data to travel from one point to another.Data may flow,with arrowhead pointing in the direction of the flow

B Data Flow Diagrams

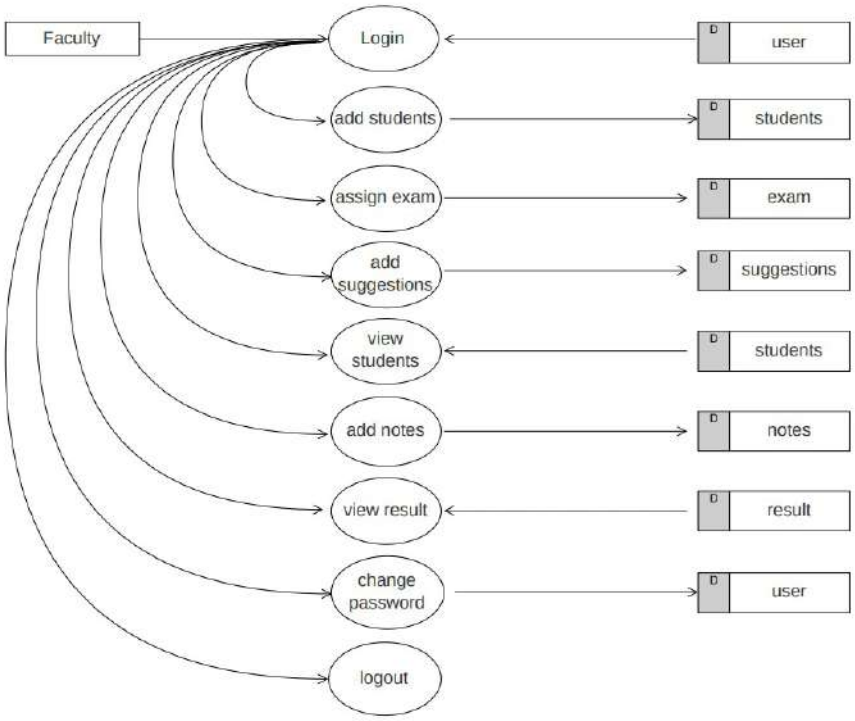
B.1 Level 0



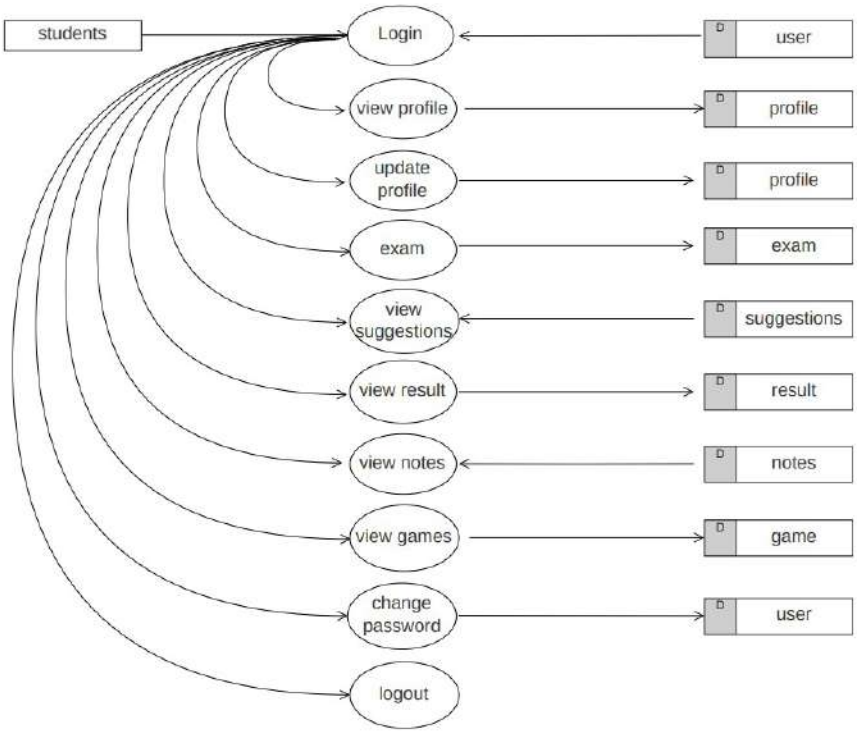
B.2 Level 1.1 - Admin



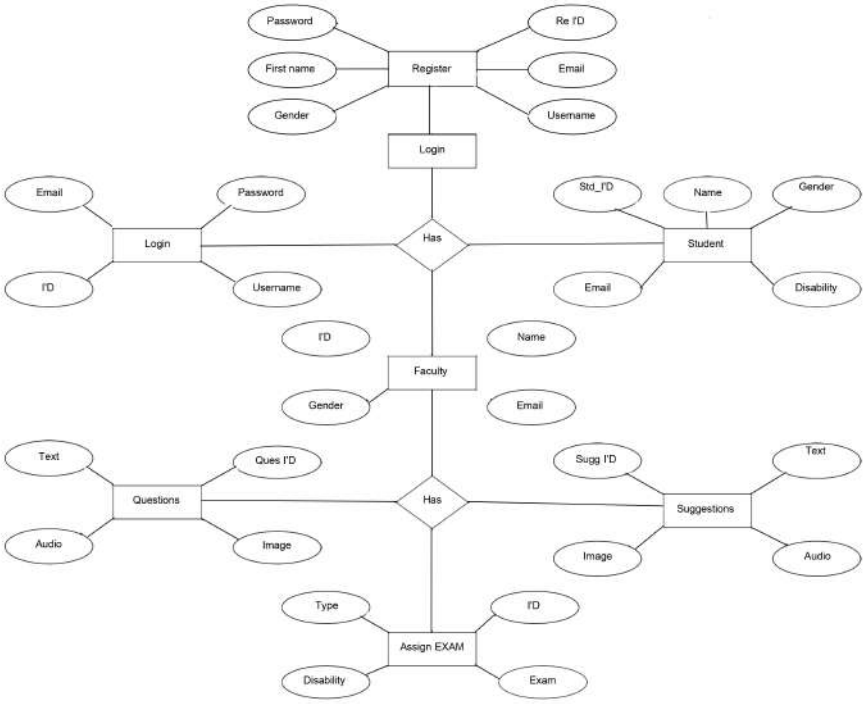
B.3 Level 1.2 - Faculty



B.4 Level 1.3 - Students

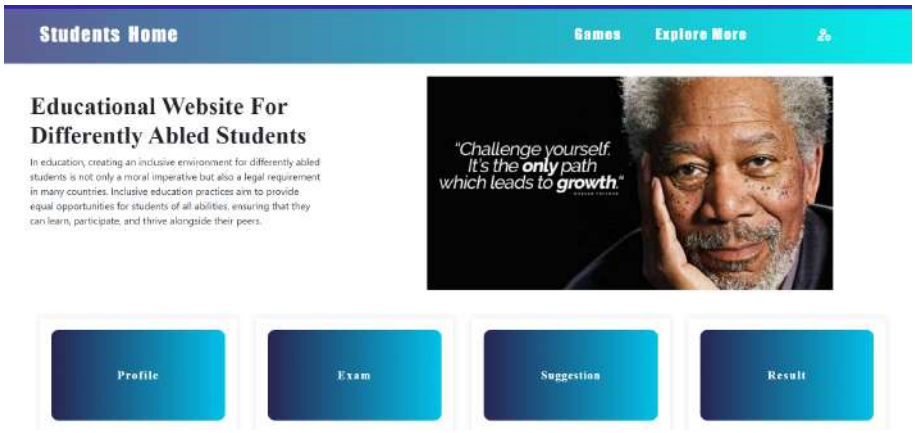


B.5 ER DIAGRAM

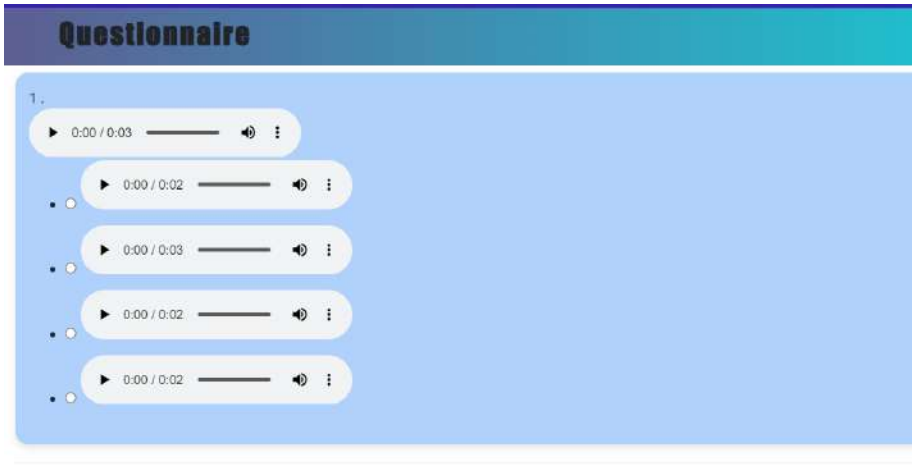


C USER INTERFACES

C.1 HOME



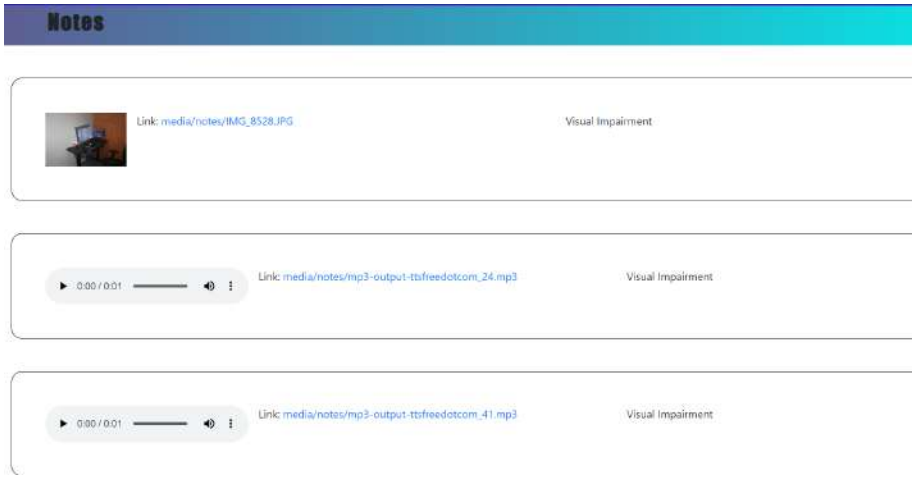
C.2 EXAM



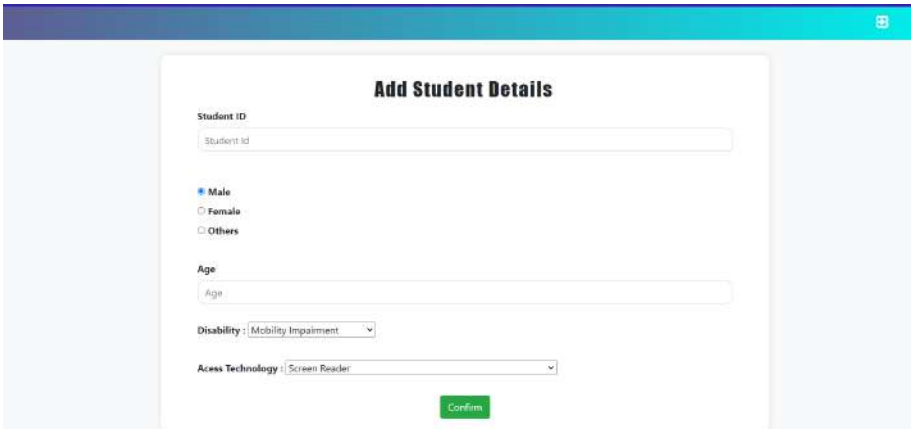
C.3 RESULT


Result
TestScore
1
Category
Very Poor
Result is here Check it....
No.
Question
Result
1
0:00 / 0:03
False
2
0:00 / 0:03
False
3
0:00 / 0:02
False
4
0:00 / 0:06
True

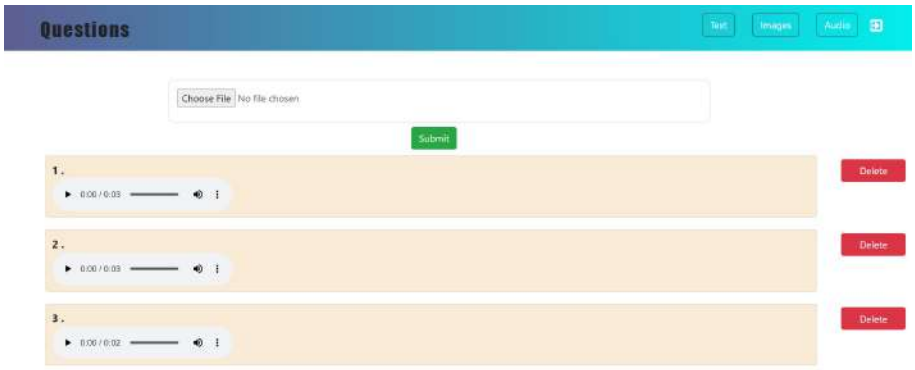
C.4 NOTES


Notes
Link: media/notes/IMG_8528.JPG
Visual Impairment
0:00 / 0:01
Link: media/notes/mp3-output-ttsfreedotcom_24.mp3
Visual Impairment
0:00 / 0:01
Link: media/notes/mp3-output-ttsfreedotcom_41.mp3
Visual Impairment

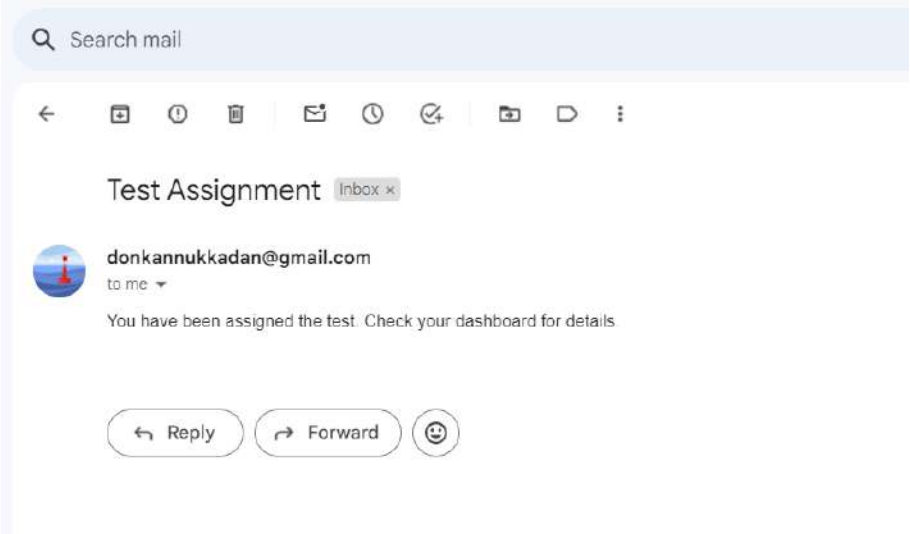
C.5 REGISTRATION



C.6 QUESTIONS WINDOW



C.7 NOTIFICATION



D CODE

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width,
      initial-scale=1.0">
  <!-- <link rel="stylesheet" href="styles.css"> -->
  <script src="toggle-sideNav.js" defer></script>
  <script src="featured-games.js" defer></script>

  <title>Game for mentally challenged people</title>
  <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,
        initial-scale=1, shrink-to-fit=no">
    <!-- Optional JavaScript -->
    <!-- jQuery first, then Popper.js, then Bootstrap JS
        -->
    <script src="https://code.jquery.com/jquery-3.3.1.
        slim.min.js" integrity="sha384-q8i/X+965
        DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8
        abtTE1Pi6jizo" crossorigin="anonymous"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/
        popper.js/1.14.7/umd/popper.min.js" integrity="
        sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9W
      O1clHTMGa3JDZwrnQq4sF86dIHNDz0W1" crossorigin="
          anonymous"></script>
    <script src="https://stackpath.bootstrapcdn.com/
        bootstrap/4.3.1/js/bootstrap.min.js" integrity="
        sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/
        nJGzIxFDsf4x0xIM+B07jRM" crossorigin="anonymous
        "></script>
    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://stackpath.
        bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min
        .css" integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH
        /1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
        crossorigin="anonymous">
    <link rel="stylesheet" href="https://fonts.googleapis
        .com/css2?family=Material+Symbols+Sharp:opsz,wght,
        FILL,GRAD@48,700,0,200" />
```

```
    <link rel="stylesheet" href="https://fonts.googleapis
        .com/css2?family=Material+Symbols+Outlined:opsz,
        wght,FILL,GRAD@24,700,0,200" />
   <link rel="stylesheet" href="https://fonts.googleapis.
       com/css2?family=Material+Symbols+Rounded:opsz,wght,
       FILL,GRAD@40,600,0,200" />
  </head>
<style>
  #play-now{
    text-decoration: none;
  }
</style>
<body>
{% load static %}
<div style="margin-left:95%;text-decoration: none;padding
    : 4px;border-radius: 0.5rem;" ><a href="{% url 'sh'
    %}" class="text-black ml-3" style="color: black;"><
    span class="material-symbols-outlined">
   exit_to_app
  </span></a></div>
    <section class="hero-wrapper">
        <div class="container">
          <p class="greeting " style="color: green;">
            Brain Games For Disabled Students!!!
          </p>
        </div>
      </section>
    <section id="play-now" class="program-wrapper">
      <div class="program-container container">
        <h3 class="title">
            Featured Games
        </h3>
        <div class="program-content">
          <div class="row">
            <div class="col">
                  <a href="{% url 'animal' %}" ><div
                      class="program-detail">
                    <img src="{% static 'images/first.png
                        ' %}" alt="reloading" width="350px
                        " height="450px">
                    <h5 class="mt-1">who am i ?</h5>
                  <p>A great game for the brain! It
                      improves visual scanning, planning,
                      and spatial memory!! </p>
                </div></a>
            </div>
```

```
            <div class="col">
              <a href="{% url 'math' %}"><div class="
                  program-detail">
                <img src="{% static 'images/third.jpg'
                    %}" alt="icon of a speaker" width="550
                    px" height="450px">
                <h5 class="mt-1">Fun with numbers</h5>
              <p>Helps to quickly solve the elementary
                  math problems!! </p>
            </div></a>
            </div>
        </div>
        <div class="row mt-5">
         <div class="col">
           <a href="{% url 'memory' %}"><div class="
               program-detail">
           <img src="{% static 'images/forth.jpg' %}"
              alt="icon of a speaker" width="500px">
           <h5>Behind the scenes</h5>
         <p>Helps to improve Visual Scanning and memory
             while remembering the cards!! </p>
         </div></a>
         </div>
        </div>
         </div>
    </section>
  <script>
   function playWelcomeMessage() {
         const welcomeMessage = new
            SpeechSynthesisUtterance('welcome games');
         window.speechSynthesis.speak(welcomeMessage);
         }
   playWelcomeMessage();
  var recognition = new webkitSpeechRecognition();
  recognition.continuous = true;
  recognition.lang = 'en-US';
  recognition.onresult = function(event) {
    var result = event.results[event.results.length -
        1][0].transcript.toLowerCase();
    console.log("result",result);
    var currentQuestionIndex=0;
     if (result.includes('back to home')) {
        // Redirect to the home page
        window.location.href = '{% url "sh" %}';
    }
};
```

```
 recognition.onerror = function(event) {
      console.error('Speech recognition error:', event.
          error);
  };
  recognition.onend = function() {
      // Restart recognition when it ends
      recognition.start();
  };
  // Start speech recognition
  recognition.start();
  </script>
</body>
</html>
```

views.py

```
from typing import Any
from django.forms.models import BaseModelForm
from django.http import HttpResponse
from django.shortcuts import render,redirect,
    HttpResponseRedirect
from .models import *
from .models import Question
from .forms import *
from django.views.generic import FormView,CreateView,
    UpdateView,TemplateView,View
from django.contrib.auth import authenticate,login,logout
from django.urls import reverse_lazy
from django.contrib.auth.hashers import make_password
from django.http import FileResponse
from django.shortcuts import get_object_or_404
from .models import Suggestion
from student_app.forms import ChangePasswordForm
from django.forms import formset_factory
# Create your views here.

import pandas as pd
import random
from django.db.models.signals import post_save
from django.dispatch import receiver
from django.contrib.auth import get_user_model
from django.http import JsonResponse

class MainHome(TemplateView):
    template_name="mainhome.html"
```

```
class ClearDataView(View):
    def post(self, request):
        StudentAnswer.objects.all().delete()
        StudentAnswerImage.objects.all().delete()
        StudentAnswerAudio.objects.all().delete()
        ScoreModel.objects.all().delete()
        return JsonResponse({'message': 'Data cleared
            successfully'})
@receiver(post_save, sender=Student)
def create_user_from_student(sender, instance, created,
    **kwargs):
    if created:
        # User = get_user_model()
        CustUser= get_user_model()
        student_id_id=instance.id
        username = instance.std_id
        password = 'admin@123'  # Set your desired
            default password here
        # User.objects.create_user(username=username,
            password=password)
        CustUser.objects.create_user(username=username,
            password=password,student_id_id=student_id_id)
        # ScoreModel.objects.create(student_id=
            student_id_id)
class LoginView(FormView):
    template_name="login.html"
    form_class=LogForm
    def post(self,request,*args,**kwargs):
        log_form=LogForm(data=request.POST)
        if log_form.is_valid():
            us=log_form.cleaned_data.get('username')
            ps=log_form.cleaned_data.get('password')
            user=authenticate(request,username=us,
                password=ps)
            if user:
                login(request,user)
                if request.user.is_superuser == 1:
                    return redirect('h')
                else:
                    return redirect('sh')
            else:
                return render(request,'login.html',{"form
                    ":log_form})
        else:
            return render(request,'login.html',{"form":
                log_form})
```

```
class AddStudent(CreateView):
    template_name='addstudent.html'
    model=Student
    form_class=StudentForm
    success_url=reverse_lazy('stu')
class QuestView(TemplateView):
    template_name='test.html'
    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)
        id=kwargs.get('pk')
        context['stu']=Student.objects.get(id=id)
        context['ques']=Question.objects.all()
        return context
# class QuestViewAll(TemplateView):
#     template_name='Assignexam.html'
#     def get_context_data(self, **kwargs):
#         context = super().get_context_data(**kwargs)
#         context['stu']=Student.objects.all()
#         context['ques'] = Question.objects.all()
#         return context

# class QuestViewImageAll(TemplateView):
#     template_name='testallimage.html'
#     def get_context_data(self, **kwargs):
#         context = super().get_context_data(**kwargs)
#         context['stu']=Student.objects.all()
#         context['ques'] = QuestionImages.objects.all()
#         return context

# class QuestViewAudioAll(TemplateView):
#     template_name='testallaudio.html'
#     def get_context_data(self, **kwargs):
#         context = super().get_context_data(**kwargs)
#         context['stu']=Student.objects.all()
#         context['ques'] = QuestionAudio.objects.all()
#         return context

def Test(request,**kwargs):
    if request.method == 'POST':
        id=kwargs.get('pk')
        stu=Student.objects.get(id=id)
        que=Question.objects.all()
        assignment = StudentAnswer(student=stu, question=
            que)
        assignment.save()
        return redirect('det')
```

```
from django.core.mail import send_mail

def AssignView(request,**kwargs):
    if request.method == 'POST':
        # Get all available tests
        tests = Question.objects.all()

        # Get all students
        students = Student.objects.filter(disability__in
            =["Mobility Impairment", "Learning Disability
            ", "Autism Spectrum Disorder", "Speech
            Impairment", "Intellectual Disability"])
        students_with_email_sent = set()
        # Assign all tests to all students
        for test in tests:
            for student in students:

            assignment, created = StudentAnswer.objects.
                get_or_create(student=student, question=
                test)
             if created:
            assignment.save()
            subject = 'Test Assignment'
            message = f'You have been assigned the test.
                Check your dashboard for details.'
            from_email = 'testhelloability@gmail.com'
            to_email = [student.email]
            if student in students_with_email_sent:
                      continue
             send_mail(subject, message, from_email,
                 to_email, fail_silently=False)
             students_with_email_sent.add(student)
        return redirect('testall')

    # Retrieve all available tests
    tests = Question.objects.all()

    return render(request, 'Assignexam.html', {'tests':
        tests})
def AssignImageView(request,**kwargs):
    if request.method == 'POST':
        # Get all available tests

        tests = QuestionImages.objects.all()
```

```
        # Get all students
        students = Student.objects.filter(disability="
            Hearing Impairment")
        students_with_email_sent = set()
        # Assign all tests to all students
        for test in tests:
            for student in students:

                assignment, created = StudentAnswerImage.
                    objects.get_or_create(student=student,
                     question=test)
                if created:
                 assignment.save()
                 subject = 'Test Assignment'
                 message = f'You have been assigned the
                     test. Check your dashboard for
                     details.'
                 from_email = 'testhelloability@gmail.com
                     '
                 to_email = [student.email]
                 if student in students_with_email_sent:
                  continue
                 send_mail(subject, message, from_email,
                     to_email, fail_silently=False)
                 students_with_email_sent.add(student)

        return redirect('testallvisual')

    # Retrieve all available tests
    tests = QuestionImages.objects.all()

    return render(request, 'testallimage.html', {'tests':
        tests})
def AssignAudioView(request,**kwargs):
    if request.method == 'POST':
        # Get all available tests
        tests = QuestionAudio.objects.all()

        # Get all students
        students = Student.objects.filter(disability="
            Visual Impairment")
        students_with_email_sent = set()

        # Assign all tests to all students
        for test in tests:
            for student in students:
```

```
                    assignment, created = StudentAnswerAudio.
                        objects.get_or_create(student=student,
                         question=test)
                    if created:
                     assignment.save()
                     subject = 'Test Assignment'
                     message = 'You have been assigned the
                         test. Check your dashboard for
                         details.'
                     from_email = 'donkannukkadan@gmail.com'
                     to_email = [student.email]
                     if student in students_with_email_sent:
                       continue
                     # Send email
                      send_mail(subject, message, from_email,
                           to_email, fail_silently=False)

                     # Add the student to the set of students
                          with sent emails
                      students_with_email_sent.add(student)

            return redirect('testallhear')

        # Retrieve all available tests
        tests = QuestionAudio.objects.all()

        return render(request, 'testallaudio.html', {'tests':
            tests})

from random import sample

class Quesadd(CreateView):
    template_name="quesadd.html"
    model=Question
    form_class=QuesForm
    success_url=reverse_lazy('qans')
    def get_context_data(self, **kwargs) :
        context = super().get_context_data(**kwargs)
        context['tests'] = Question.objects.all()
        return context

class Quesimage(CreateView):
    template_name="quesimage.html"
    model=QuestionImages
    form_class=QuesFormImage
    success_url=reverse_lazy('qansimg')
```

```
    def get_context_data(self, **kwargs) :
        context = super().get_context_data(**kwargs)
        context['tests'] = QuestionImages.objects.all()
        return context

class Quesaudio(CreateView):
    template_name="quesaudio.html"
    model=QuestionAudio
    form_class=QuesFormAudio
    success_url=reverse_lazy('qansaudio')
    def get_context_data(self, **kwargs) :
        context = super().get_context_data(**kwargs)
        context['tests'] = QuestionAudio.objects.all()
        return context

class QuesUpdate(UpdateView):
    template_name="questionupdate.html"
    model=Question
    form_class=QuesForm
    success_url=reverse_lazy('qdel')
    def get_context_data(self, **kwargs) :
        context = super().get_context_data(**kwargs)
        context['tests']=Question.objects.all()
        return context

class QuesAnsUpdView(CreateView):
    template_name="quesansupdate.html"
    model=Answer
    form_class=QuesAnsForm
    success_url=reverse_lazy('qdel')
    def form_valid(self, form):
        # Save the main answer
        main_answer = form.save(commit=False)
        main_answer.save()

        # Get additional options from the form data
        option_texts = [self.request.POST.get(f'option_{i
            }') for i in range(1, 4)]

        # Create and save additional options
        for option_text in option_texts:
            if option_text:
                answer_option = Answer(question=form.
                    cleaned_data['question'], text=
                    option_text)
                answer_option.save()
```

```
        return super().form_valid(form)

class QuesAnsImageView(CreateView):
    template_name = "quesansimage.html"
    model = AnswerImages
    form_class = QuesAnsFormImg
    success_url = reverse_lazy('qimg')

    def form_valid(self, form):
        # Save the main answer
        main_answer = form.save(commit=False)
        main_answer.save()

        # Handle file uploads for options
        for i in range(1, 4):
            option_file = self.request.FILES.get(f'
                option_{i}')
            if option_file:
                AnswerImages.objects.create(
                    question=main_answer.question,
                    fileans=option_file
                )
        return super().form_valid(form)

class QuesAnsAudioView(CreateView):
    template_name="quesansaudio.html"
    model=AnswerAudio
    form_class=QuesAnsFormAudio
    success_url=reverse_lazy('qaudio')
    def form_valid(self, form):
        # Save the main answer
        main_answer = form.save(commit=False)
        main_answer.save()

        # Handle file uploads for options
        for i in range(1, 4):
            option_file = self.request.FILES.get(f'
                option_{i}')
            if option_file:
                AnswerAudio.objects.create(
                    question=main_answer.question,
                    fileans=option_file
                )

        return super().form_valid(form)
```

```
class DeleteView(View):
    def get(self,req,*args,**kwargs):
        id=kwargs.get('pk')
        dl=Question.objects.get(id=id)
        dl.delete()
        return redirect('qdel')

class DeleteImgView(View):
    def get(self,req,*args,**kwargs):
        id=kwargs.get('pk')
        dl=QuestionImages.objects.get(id=id)
        dl.delete()
        return redirect('qimg')

class DeleteAudioView(View):
    def get(self,req,*args,**kwargs):
        id=kwargs.get('pk')
        dl=QuestionAudio.objects.get(id=id)
        dl.delete()
        return redirect('qaudio')

# class Quesdel(TemplateView):
#     template_name="quesdel.html"
#     def get_context_data(self, **kwargs) :
#         context = super().get_context_data(**kwargs)
#         context['tests']=Question.objects.all()
#         return context

class SugView(TemplateView):
    template_name="suggestions.html"
    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)
        context['data']=Suggestion.objects.all().order_by
            ('cat')
        return context

class SuggTextView(CreateView):
    template_name="suggtext.html"
    model=Suggestion
    form_class=SugForm
    success_url=reverse_lazy('st')

class SuggVideoView(CreateView):
    template_name="suggvideo.html"
    model=Suggestion
```

```
    form_class=SugVideoForm
    success_url=reverse_lazy('sv')

class SuggAudioView(CreateView):
    template_name="suggaudio.html"
    model=Suggestion
    form_class=SugAudioForm
    success_url=reverse_lazy('sadd')

def view_video(request, video_id):
    video = get_object_or_404(Suggestion, pk=video_id)
    video_path = video.video.path
    response = FileResponse(open(video_path, 'rb')) #
        Adjust content_type as needed
    return response

def view_videoo(request, videoo_id):
    video = get_object_or_404(ScoreModel, pk=videoo_id)
    video_path = video.video.path
    response = FileResponse(open(video_path, 'rb')) #
        Adjust content_type as needed
    return response

def play_audio(request, audio_id):
    audio_recording = get_object_or_404(Suggestion, pk=
        audio_id)
    audio_file = audio_recording.audio
    response = FileResponse(open(audio_file.path, 'rb'))
    return response

def play_audioo(request, audio_id):
    audio_recording = get_object_or_404(ScoreModel, pk=
        audio_id)
    audio_file = audio_recording.audio
    response = FileResponse(open(audio_file.path, 'rb'))
    return response

class DeleteViewSug(View):
    def get(self,req,*args,**kwargs):
        id=kwargs.get('pk')
        dl=Suggestion.objects.get(id=id)
        dl.delete()
        return redirect('sadd')

class ChangePasswordViewHome(FormView):
    template_name="changepshome.html"
```

```
    form_class=ChangePasswordForm
    def post(self,request,*args,**kwargs):
        form_data=ChangePasswordForm(data=request.POST)
        if form_data.is_valid():
            current=form_data.cleaned_data.get("
                current_password")
            new=form_data.cleaned_data.get("new_password
                ")
            confirm=form_data.cleaned_data.get("
                confirm_password")
            user=authenticate(request,username=request.
                user.username,password=current)
            if user:
                if new==confirm:
                    user.set_password(new)
                    user.save()
                    logout(request)
                    return redirect("log")
                else:
                    return redirect("cp")
            else:
                return redirect("cp")
        else:
            return render(request,"changepassword.html
                ",{"form":form_data})

class SuggestionUpdateView(UpdateView):
    template_name='suggestionupdate.html'
    model=Suggestion
    form_class=SuggestionForm
    success_url=reverse_lazy('sadd')
```

PRODUCT MANAGEMENT SYSTEM

PROJECT REPORT

Submitted By

FEBIN TOM

Reg. No. CCAVBCA033

for the award of the Degree of

Bachelor of Computer Application (BCA)

in Computer Application
(University of Calicut)

under the guidance of

Ms. Vandana T.V

Assistant Professor

BACHELOR OF COMPUTER APPLICATION
DEPARTMENT OF COMPUTER SCIENCE
CHRIST COLLEGE(Autonomous)
IRINJALAKUDA, KERALA
2021-2024

DEPARTMENT OF COMPUTER SCIENCE

CHRIST COLLEGE (AUTONOMOUS)

IRINJALAKUDA

CERTIFICATE

This is to certify that the project report entitled ***"Product Management System"*** *is a bonafide record of the project work done by* ***Febin Tom*** *in partial fulfillment of the requirement for the sixth semester of* ***Bachelor of Computer Application*** *in Department of Computer Science of* ***CHRIST COLLEGE (AUTONOMOUS) IRINJALAKUDA***

.

Ms. Vandana T.V
Assistant Professor,CS
Internal Guide

Ms. Sini Thomas
Head of the Department
Computer Science

EXTERNAL EXAMINER **INTERNAL EXAMINER**

KALIPARAMBIL STORES

TO WHOMSOEVER IT MAY CONCERN

This is to certify that **Mr.FEBIN TOM**
(CCAVBCA033)
student of
CHRIST COLLEGE (AUTONOMOUS), IRINJALAKUDA
(Bachelor Of Computer Application)
has successfully completed their project
PRODUCT MANAGEMENT SYSTEM under the guidance of
Ms. VANDANA TV (guide) and implemented in
KALIPARAMBIL STORES on 29-01-2024
during the academic year 2021-2024.

MUJEEB KB

KALIPARAMBIL STORES
MANAGER

KALIPARAMBIL STORES

Proprietor

Vellikulangara Junction

DECLARATION

We hereby declare that this project work **"PRODUCT MANAGEMENT SYSTEM"** submitted to Christ College (Autonomous) Irinjalakuda, affiliated to Calicut University in partial fulfillment of the requirement for the award of the Bachelor of Computer Applications, is a record of original work done by us, under the guidance of Ms. VANDANA T.V, Department of computer Science.

Place: Irinjalakuda FEBIN TOM

ACKNOWLEDGEMENT

First and foremost we like to thank Lord almighty for his providence and for being the guiding light throughout the project. We wish to express my sincere gratitude to our beloved Department head for giving me all the facilities for our project. We take this opportunity to express my gratitude to the class teacher Ms. SOWMYA P.S and head of the department Ms. SINI THOMAS who has been supported us throughout the course of this project. We are thankful for her aspiring guidance and valuable advice during the project work. We express my sincere thanks to my project guide Ms. VANDANA T.V for supporting and guiding throughout the project.We would take this opportunity to specially thank all other faculty members for their constant and continuous motivation. Finally we would like to thank my family and friends for giving valuable advice and moral support throughout our project.

ABSTRACT

PRODUCT MANAGEMENT SYSTEM for Retail Stores represents a groundbreaking solution aimed at revolutionizing inventory management and enhancing operational efficiency within the retail sector. This innovative web application offers a comprehensive suite of features, including intuitive inventory management, proactive notifications for expiring products, promotion management capabilities, robust search functionality, seamless integration with point-of-sale systems, and insightful dashboard analytics. Through its user-friendly interface and advanced functionalities, the system empowers retail store owners and employees to optimize inventory processes, drive sales, and make data-driven decisions. With future enhancements focused on supply chain integration, mobile accessibility, and advanced analytics, the proposed system promises to redefine the retail experience and set new standards for success in the industry.

Contents

Chapter 1

1 Introduction

The Product Management System for Retail Stores heralds a new era of efficiency and optimization within the bustling world of retail. This innovative web application emerges as a comprehensive solution tailored to address the complex challenges inherent in managing store inventory and streamlining retail operations. At its core, this project aims to empower retail store owners and employees with a robust platform that seamlessly integrates inventory management, notification systems for expiring products, promotion management, product search functionality, electronic bill generation, and insightful dashboard analytics. By combining cutting-edge technology with intuitive design, the system revolutionizes the way retail businesses approach their day-to-day operations, fostering increased productivity, minimized wastage, and enhanced customer satisfaction. At the forefront of this project lies a user-friendly interface that serves as the gateway to a myriad of powerful features. Through a series of meticulously designed forms and interactive elements, users can effortlessly navigate the system, adding, editing, and removing products while inputting vital details such as product names, expiry dates, quantities, and prices. This meticulous attention to detail ensures that the inventory database remains accurate and up-to-date, providing users with a solid foundation upon which to build their retail endeavors. One of the standout features of the system is its proactive approach to managing expiring products. Through a sophisticated notification system, users receive timely alerts when products are nearing their expiry date, enabling them to take swift action to prevent wastage and capitalize on opportunities to drive sales through targeted promotions.

1.1 Overview

The Product Management System is born from a vision to revolutionize the way retail stores manage their inventory, sales, and operational tasks. With the ever-increasing complexity of product management in modern retail, our team recognized the need for a robust, user-friendly solution that integrates essential features to enhance efficiency and productivity.

Chapter 2

2 System Analysis

2.1 Purpose

This documentation serves as a comprehensive resource for understanding the design, functionality, and implementation of the Product Management System. Whether you are a developer looking to extend the system, a store manager seeking to optimize operations, or an investor evaluating the potential impact of our solution, this document provides the insights and information necessary to engage with our project effectively.

2.1.1 Existing System

The current Product Management System in use at our retail store represents a fundamental tool for managing inventory, sales, and operational tasks. Developed to address the challenges of product management in a retail environment, the system offers a range of essential features aimed at facilitating efficient store operations and enhancing customer satisfaction.

2.1.2 Proposed System

The proposed Product Management System for Retail Stores is a comprehensive solution designed to revolutionize inventory management and streamline retail operations. Key features include an intuitive user interface for efficient inventory management, automated notifications for expiring products, promotion management functionality to drive sales, robust search capabilities for quick access to product information, seamless integration with point-of-sale systems for smooth transactions, and insightful dashboard analytics for data-driven decision-making. Future scope includes integration with supply chain management for automated ordering and replenishment, development of a mobile application version for increased accessibility, and enhancement of analytics capabilities for deeper insights. Overall, the proposed system aims to empower retail store owners and employees with the tools they need to optimize their operations, minimize costs, and enhance customer satisfaction in today's competitive retail environment. Advantages of proposed system :unique page are provided, Automatic expiry notification when product reaches its expiry date and qr code is provided to each product get a overview about the product

2.2 Problem definition

To know what the problem is and what the needs are before developing it. Here we design a web application of retail stores for managing and tracking their product details

2.3 Feasibility study

After the problem is clearly understood and solutions proposed, the next step is to conduct the feasibility study. Feasibility study is defined as evaluation or analysis of the potential impact of the proposed project or program. The objective is to determine whether the proposed system is feasible. There are three aspects of feasibility study to which the proposed system is subjected as discussed below.

2.3.1 Technical Feasibility

Technical feasibility assess whether the current technical resources are sufficient for the new system. We can upgrade the level of technology for supporting our webapplication. We check whether the proposed system can be implemented in the present system without supporting the existing hardware.

2.3.2 Economical Feasibility

Economic feasibility determines whether the time and money are available to develop the system. There is no additional hardware used to develop the site, it is inexpensive to build.

2.3.3 Operational Feasibility

Operational feasibility determines if the human resources are available to operate the system once it has been installed. No extra training is needed to use this system. Anyone who has knowledge in internet in english language can easily use the system. The resources that are required to implement or install are already available with the organization.

Chapter 3

3 Software Requirement Specification

3.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the PRODUCT MANAGEMENT SYSTEM. It illustrate the purpose and complete description for the development of the system. It explain system constraints, interface and interactions with other external applications. This document is primarily intended to be proposed to a store for managing and tracking the product details and help the user to gain profit.

3.2 Scope

Our project has made it easier for store to manage products easily and systematically . We can get all information about the product by having a look at this web application. In the future drone delivery can also be included.

3.3 Overall Description

The system will include features such as user authentication, product management, expiry date notifications, discount management, electronic bill monitoring, and sales analytics, providing stakeholders with real-time insights into product performance and sales trends. By prioritizing non-functional requirements such as performance, security, usability, and reliability, the system will offer a secure, scalable, and user-friendly solution tailored to the needs of retail environments. With a focus on minimizing waste, optimizing inventory turnover, and enhancing customer satisfaction, this project seeks to empower retail businesses to thrive in an increasingly competitive market landscape.

3.3.1 Product Perspective

From a product perspective, the Product Management System for Retail Stores serves as a crucial tool to address the specific needs and challenges faced by retail businesses in managing their inventory, sales, and operational tasks. The system provides a centralized platform for store personnel to efficiently manage product information, monitor stock levels, and make informed decisions regarding pricing, promotions, and inventory optimization. With features such as expiry date notifications and dynamic discount management, the system enables proactive measures to minimize waste and maximize profitability. Additionally, the integration of sales analytics capabilities empowers stakeholders to gain valuable insights into product performance, customer preferences, and market trends, facilitating data-driven decision-making and strategic planning. By offering a comprehensive suite of functionalities and prioritizing user experience, security,

and scalability, the system aims to enhance operational efficiency, drive business growth, and ultimately contribute to the success and sustainability of retail businesses.

3.3.2 Product Functionality

Through this system admin can include various product data.User can getoverall control of the web application

3.3.3 Users and Characteristics

There are two types of users that interact with the site admin and employee. Both of them has the priviledge to login and register into web application.Both ofthe usercan add product.But only admin have the priviledge to access into the database

3.4 Specific Requirements

3.4.1 Hardware Requirements

- System: IBM-Compatible PC
- Processor: Pentium IV or above
- Speed: Above 1GHz
- RAM capacity: 512 MB
- Hard Dsk drive: 40 GB
- Keyboard: Standard
- Mouse: Standard
- Monitor: SVGA Color

3.4.2 Software Requirements

- Operating System:Windows or ubuntu
- Languages used: Python Django
- Database : MySql
- Technologies used: HTML,Javascript,CSS,Bootstrap

3.5 Functional Requirements

It contains three main modules.

- 1.Admin
- 2.User or Employee

Admin

An Admin account is used for editing or managing the web application dynamically by Admin panel. The admin can add new products, and have the access to database.

User

The user or The employee can register ar login the web apllication. Also the user can add product into the database and for the current status of the product. but they did not have the right to access into the database.

3.6 Non Functional Requirements

Non-functional requirements define the overall qualities or attributes of the resulting system. Non-functional system place restrictions on the product being developed, the developed process, and specify external constraints that the product must meet. Examples of non-functional requirements include safety, security, usability, reliability and performance requirements. Project management issues (costs, time and schedule) are often considered as non-functional requirements. The principal non - functional constraints which are relevant to critical systems :

- performance
- security
- safety
- usability

Performance

Performance requirements concern the speed of operation of a system.Type of performance requirements :

- Response requirements (how quickly the system reacts to a user input).
- Throughput requirements (how much the system can accomplish within a specified amount of time).
- Availability requirements (is the system available for service when requested by end users). The speed of operation of this system is adequate for the requirements.

Reliability

- Reliability is the ability of a system to perform its required function under stated conditions for a specified period of time.

- constraints on the runtime behavior of the system. This system is reliable because its functionalities can be done on the required conditions.

Safety

Safety requirements are not required which exclude unsafe situation from the possible solution of the system.

Usability

Usability is the ease with which a user can learn to operate, prepare inputs for, and interpret outputs of system or components. Usability requirements include :

- information error messages.
- well-formed user interfaces.

3.7 Interface Requirements

3.7.1 Hardware interfaces

The system must run over the internet, all the hardware shall require to connect internet will be hardware interface for the system. As for example modem, WAN-LAN.

3.7.2 Software interfaces

Software interface required for the working of the project is the appropriate operating system.

3.7.3 Communication interfaces

The user of the site communicate using the network connectivity and the data set is accessed

3.8 Security Requirements

- User accesses only their account.
- Validation of input is handled.
- This application containing the computer systems is physically secured against arms or surreptitious entry by intruders.
- Users must be authorized carefully to reduce changes of any such user giving access to an intruder in exchange for a bribe or other favour.

3.9 Platform Used

Windows 10 is a major version of the Microsoft Windows operating system that was released on July 29, 2015.It is built on the Windows NT kernel and follows windows 8.Part of the reason Microsoft decided to name the 2015 release "Windows 10"(and skipped "windows 9")is because the operating system is designed to be a new direction for Microsoft. One of the primary aims of windows 10 is to Unify the windows experience across multiple devices, such desktop computers, tablets, and smartphones. As part of this effort, Microsoft developed Windows10 Mobile alongside Windows 10 to replaces Windows Phone - Microsoft's previous mobile OS. Windows 10 also integrates other Microsoft services, such as Xbox Live and the Cortana voice recognition assistant.

3.10 Technologies Used

Python Django

Python Django is a powerful web framework that facilitates rapid development and clean, pragmatic design. It follows the "don't repeat yourself" (DRY) principle, enabling developers to build complex web applications with efficiency and simplicity. Django's batteries-included philosophy means it comes with many built-in features and functionalities, including an ORM (Object-Relational Mapping) for database interactions, a robust authentication system, and a powerful templating engine. Its versatility allows developers to create various types of web applications, from content management systems to e-commerce platforms. Django's scalability and security features make it a popular choice among developers for building secure and scalable web applications. Additionally, Django's active community and extensive documentation provide ample support for developers at all levels, making it an excellent framework for both beginners and experienced developers alike.

MySQL

MySQL, pronounced either "My S-Q-L" or "My Squel, " is an open source relational database management system. It is based on the structure query language (SQL), which is used for adding, removing and modifying information in the database. Standard SQL commands, such as ADD, DROP, INSERT,and UPDATE can be used with MySQL.

What distinguishes Python Django from something like client - side Javascript is that the code is executed on the server, generating HTML which is then sent to the client. The client would receive the results of running that script, but would not know what the underlying code was. You can even configure your web server to process all your HTML files with Python Django, and then there's really no way that users can tell what you have up your sleeve. The best things in using Python Django are that it is extremely simple for a newcomer, but offers many advanced features for a professional programmer.

Chapter 4

4 Design Document

4.1 Purpose

The Product Management System for Retail Stores web application serves as a comprehensive solution designed to optimize inventory management and enhance operational efficiency in retail settings. Its primary objectives include streamlining inventory management tasks by providing a centralized platform for adding, editing, and deleting products, as well as sending timely notifications for expiring products to prevent wastage. Additionally, the application facilitates promotion management, enabling users to offer discounts or special deals on expiring products to drive sales. It enhances the customer experience by offering features such as product search functionality and electric bill generation, ensuring a smooth transaction process. Furthermore, the application provides valuable insights through its dashboard analytics, empowering store owners to make informed decisions and optimize business strategies based on data-driven insights.

4.2 Scope

The scope of the Product Management System for Retail Stores web application encompasses a range of functionalities aimed at optimizing inventory management and enhancing retail operations. It includes features such as efficient inventory management through adding, editing, and deleting products, automated expiry notifications to prevent wastage, promotion management for expiring products to drive sales, product search functionality for quick access to information, electric bill generation for seamless transactions, and dashboard analytics providing valuable insights into store performance. By incorporating these features, the application aims to streamline processes, improve efficiency, and empower store owners with data-driven decision-making capabilities in their retail endeavors.

4.3 Overview

The purpose of this document is to help the reader to visualize the solution to the project presented.This document verifies how the design meet the requirement stimulated in the SRS document through design viewpoints.The design viewpoints will cover all design elements presented before. This document will provide a direct approach to the development of this project hence reducing feature creep and ponitedly determine the quality of the design.

4.4 Data Design

Database are the storehouses of data used in the software systems.The data is stored in tables inside the database.Several tables are created for the manipulation of the data for the system.Two essential settings for a database are

- Primary Key- The field that is unique for all the record occurrence.
- Foreign Key- The field used to set relation between tables.

Normalization

- Normalization is the process of organizing the data in the database.
- Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate the undesirable characteristics like Insertion, Update and Deletion Anomalies.
- Normalization divides the larger table into the smaller table and links them using relationship.
- The normal form is used to reduce redundancy from the database table.

Tables

Products

Name	DataType	Constraints	Description
id	int(11)	Primarykey	ID of product
name	varchar(25)	Notnull	Name of product
category	varchar(100)	Notnull	category of product
quantity	int(11)	Notnull	quantity of product
price	int(20)	Notnull	Prices of product
expiry_date	date	Notnull	expiry date of product
vendor	varchar(250)	Notnull	Name of vendor

Sales

Name	DataType	Constraints	Description
id	int(11)	Primarykey	ID of product
item	varchar(50)	Notnull	Name of item
price_item	int(11)	Notnull	Price of item
total_price	int(11)	Notnull	Total price of item
payment method	varchar(200)	Notnull	Payment method for product
customer name	varchar(20)	Notnull	Name of the customer
date	date	Notnull	Date of product purchase

Bill

Name	DataType	Constraints	Description
item	varchar(25)	Notnull	Item name of the product
quantity	int(9)	Notnull	Quantity of product
price	int(9)	Notnull	Price of the product
payment_method	varchar(25)	Notnull	payment method for the product

Staff

Name	DataType	Constraints	Description
id	int(11)	Primarykey	id of staff
user name	varchar(25)	Notnull	Name of staff
phone number	int(10)	Notnull	Phone number of the staff
status	varchar(30)	Notnull	Status of the staff
role	varchar(10)	Notnull	Role of the staff

Expiry date

Name	DataType	Constraints	Description
id	int(9)	Primarykey	ID of the product
name	varchar(25)	Foreignkey	Name from Product table
category	varchar(30)	Foreignkey	Category from Product table
quantity	int(11)	Foreignkey	Quantity from Product table
price	int(11)	Foreignkey	Price from Product table
expiry date	date	Foriegnkey	Expiry date from Product table
vendor	varchar(10)	Foriegnkey	Vendor from Product table

Chapter 5

5 Development of the System

The development of the Product Management System for Retail Stores web application involves utilizing a combination of front-end technologies like HTML, CSS, and JavaScript alongside a back-end framework such as Django or Flask, supported by a suitable database management system. Following an agile methodology, the development process encompasses stages of requirement gathering, system design, implementation, testing, deployment, and ongoing maintenance. Collaboration and communication among team members, stakeholders, and end-users are prioritized throughout, facilitated by project management tools and regular meetings. Security considerations are paramount, with the implementation of best practices for secure coding, data encryption, and access control to safeguard sensitive information. Ultimately, the development aims to deliver a robust, user-friendly solution that optimizes inventory management, enhances operational efficiency, and ensures data security in retail environments.

Chapter 6

6 System Testing

Testing is the penultimate step of software development.An elaborate testing of the data is prepared and the system is using the test data,while doing testing,errors are noted and correction is made.The users are trained to operate the developed system.Both hardware and software securities are made to run the developed system successfully.System testing is aimed at ensuring the system works accurately before the live operation commences.

6.1 Test Plan

6.1.1 Scope

This test plan will cover the following testing activities as identified in the testing strategy.

- White Box Testing

White box testing is a testing technique, that examines the program structure and derives test data from the program logic/code. In white box testing, the UI is bypassed. Inputs and outputs of ratings are tested directly at the code level and the results are compared against specifications. It Reveals errors in "hidden" code.

- Black Box Testing

Black - box testing is a method of software testing that examines the functionality of an application based on the specifications. Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end - user would.

- Unit Testing

The module includes four main components to undergo unit testing. It checks the syntax error, logic error and validity of the program correctness. This test will be performed by the developers.

- Integration Testing

After all components pass the unit testing, they get to be tested whether they work correctly when they are running concurrently and communicating to each other. Integration tests exercise an entire subsystem and ensure that a set of components play nicely together.

- Internal Data Testing

We will test the validity of the data before it enters the database to avoid any problems that may face us in the database. We will test the encryption of the personal information of all the users along with the user names and passwords to ensure maximum security of the users privacy.

6.1.2 Software risk issues

In this section, the plan is to test the risk involved in critical issues such as: Difficult to run on visual studio due to large loading time. Some inherent software risks such as complexity exist; also these issues need to be identified. Proper network connection Working of Mysql database

6.1.3 Features to be tested

- Test whether correct user name and password allows you to login.
- Test whether invalid user name and password prevents you from login.
- Test whether there is any connection problem in the Server.
- Test whether the student and admin details are entered correctly.
- Test whether invalid data entry allows saving data successfully.
- Test whether all pages are loaded correctly.
- Test whether watermark is embedded and extracted properly.
- Test whether the provided security works correctly.

6.2 Test consolidation

6.2.1 Test item

The items or features to be tested in the test cases are included in the document. Each and every user input is tested. The present condition of the system is tested. It is checked to make sure that environment is ready for the application to work.

6.2.2 Input specifications

These are the inputs required to execute the test case. All required inputs including data elements and values are given below.

Data	Values
Email	A valid email ID
Password	Combination of characters and numbers
Phone number	10 Digit number

Chapter 7

7 System Implementation and Maintenance

7.1 Implementation

System implementation is the conversion of new system into an operating one which involves creating compatible files, training clients and installing hardware. User training is crucial for minimizing resistance to change and giving chance to prove its worth. Training aids user friendly manuals and healthy screens provide the user with a good start. Software maintenance follows conversion to the extent that changes are necessary to maintain satisfactory operations relative to changes in the user's environment. Maintenance often includes minor enhancements or corrections to the problem that surface late in the systems operations. In the implementation phase, the team builds the components either from scratch or by composition. Given the architecture document meant from the design phase and the requirement document from the analysis phase, the team should build exactly what has been requested, though there is still room for innovation and flexibility. For example, a component may be narrowly designed for this particular system, or the component may be made more general to satisfy a reusability

- Careful planning
- Investigation of system and constraints
- Design the methods to achieve changeover.
- Training the staff in the changed phase.
- Evaluation of change over method.
- The method of implementation and time scale to be adopted are found out initially.

7.2 Maintenance

This phase occurs as a result of deploying the whole system at the end users organization. They will perform the beta testing at the end users and inform to the developers about any needed modification to the application. The customer records all the problems that are encountered during the beta testing and reports these to the developer at regular intervals.

7.2.1 Corrective Maintenance

Even with the best quality assurance activities, it is likely that the customer will uncover defects in the software. Corrective maintenance changes the software to correct the defects. Corrective Maintenance activity may consist of repair,

restoration or replacement of equipment. This activity will be result of a regular inspection, which identifies the failure in time for corrective maintenance to be planned and scheduled, then performed during a routine maintenance shutdown.

7.2.2 Adaptive Maintenance

Over time, the original environment(CPU, operating system, business rules, external product characteristics) for which the software was developed is likely to change. Adaptive maintenance results in modification to the software to accommodate changes to its external environment.

7.2.3 Enhanced Maintenance

As software is used, the customer/user will recognize additional functions that will provide the benefit. Perfect maintenance extends the software beyond its original functional requirements.

7.2.4 Preventive Maintenance

Computer software deteriorates due to change, and because of this preventive maintenance often called software re-engineering, must be conducted to enable the software to serve the needs of its end users. Preventive maintenance makes changes to computer programs so that they can be more easily corrected, adapted and enhanced.

Chapter 8

8 Conclusion and Future Scope

8.1 Conclusion

In conclusion, the Product Management System for Retail Stores stands as a transformative solution poised to redefine how retail businesses operate in the modern landscape. By combining intuitive design with powerful features such as inventory management, expiring product notifications, promotion management, and insightful analytics, the system empowers store owners and employees to streamline operations, minimize wastage, and drive profitability. With future enhancements focused on integration with supply chain management, mobile accessibility, and advanced analytics, the system is well-positioned to adapt and evolve alongside the changing needs of the retail industry. Ultimately, the proposed system promises to revolutionize the retail experience, fostering increased efficiency, agility, and customer satisfaction in an ever-evolving marketplace.

8.2 Future Scope

- Drone delivery
- Expansion to Multi-location Support
- Implementation of Customer Relationship Management (CRM) Features
- Integration with IoT Devices
- Integration with Supply Chain Management

Appendix

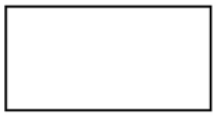
A Data Flow Diagrams

Data flow is the one of the best way of documenting the entire functionality of the system.For the system ,which will have data flows in and have some processing inside and then some data flow out from the system can be documented or represented effectively by means of data flow out from the system can be documented or represented effectively by means of data flow diagrams. The data flow diagram are a diagrammatic representation of the system,which has input,process and output.Once any system is represented using a data flow diagram we can identify the following things easily:

- Various entities interacting with the system are identified
- Flow of data from one entity to another is identified
- The various processes involved in between the interaction of two or more entities in the system are clearly pointed out
- The various data stores which hold the data in between the process,are clearly identified

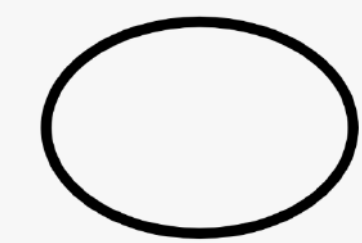
Some Data Flow Diagram charting forms:

A.1 External source or receiver



A source or sink is a person or part of organization ,which enters or receives information from the system,but is considered to be outlining the contest of data flow model.

A.2 Transform process

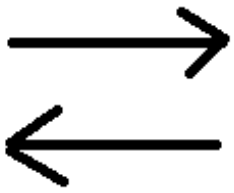


A process represents transformation where incoming data flowsare changed into outgoing data flow

A.3 Data Store

A data store is repository of data that is to be stored for use by one or more process may be as simple as buffer or queue or sophisticated as relational database. They should have clear names.If a process merely uses the contest of store and does not alter it,the arrowhead goes only form the store to the process. If a process alters the details in the store then double-headed arrow is used.

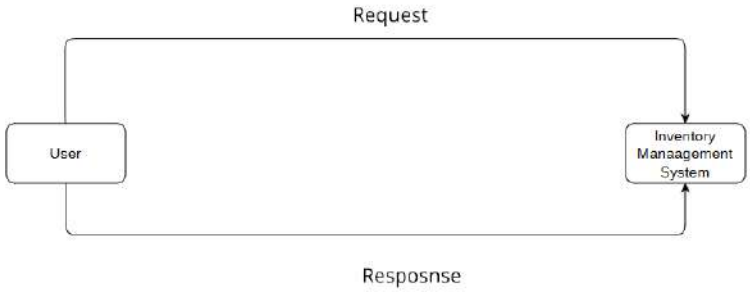
A.4 Data flow



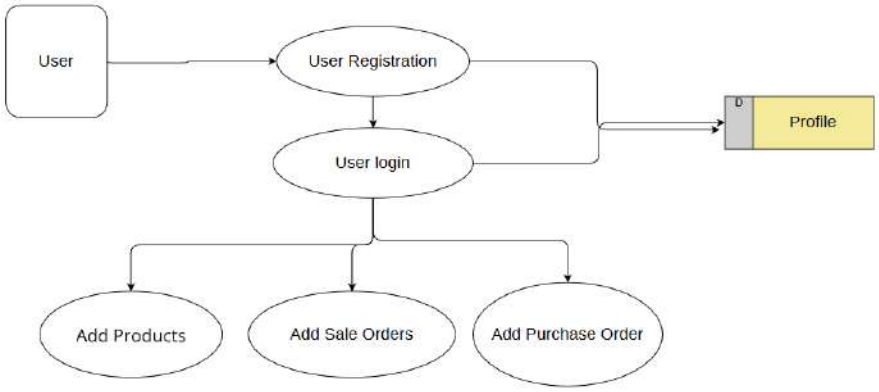
A data flow is a route, which enable packets of data to travel from one point to another.Data may flow,with arrowhead pointing in the direction of the flow

B Data Flow Diagrams

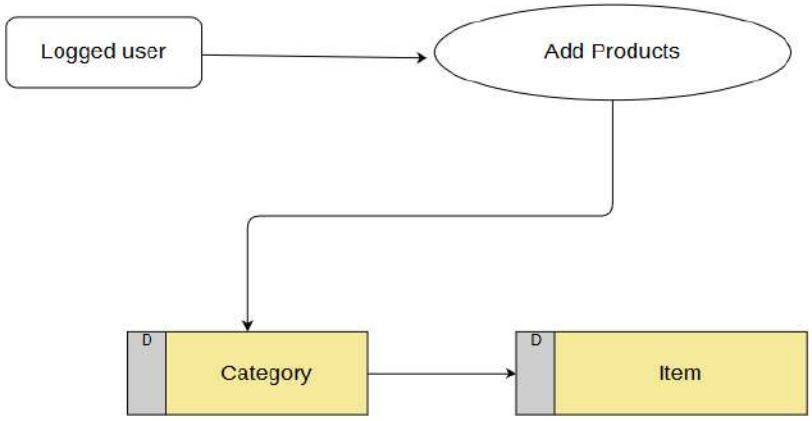
B.1 Level 0



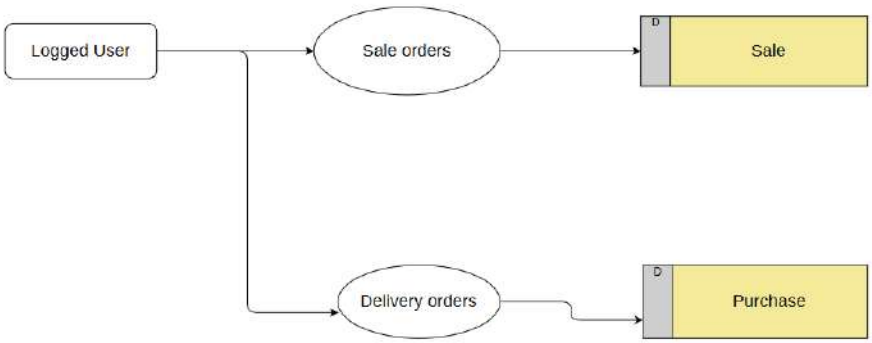
B.2 Level 1


User
User Registration
User login
D
Profile
Add Products
Add Sale Orders
Add Purchase Order

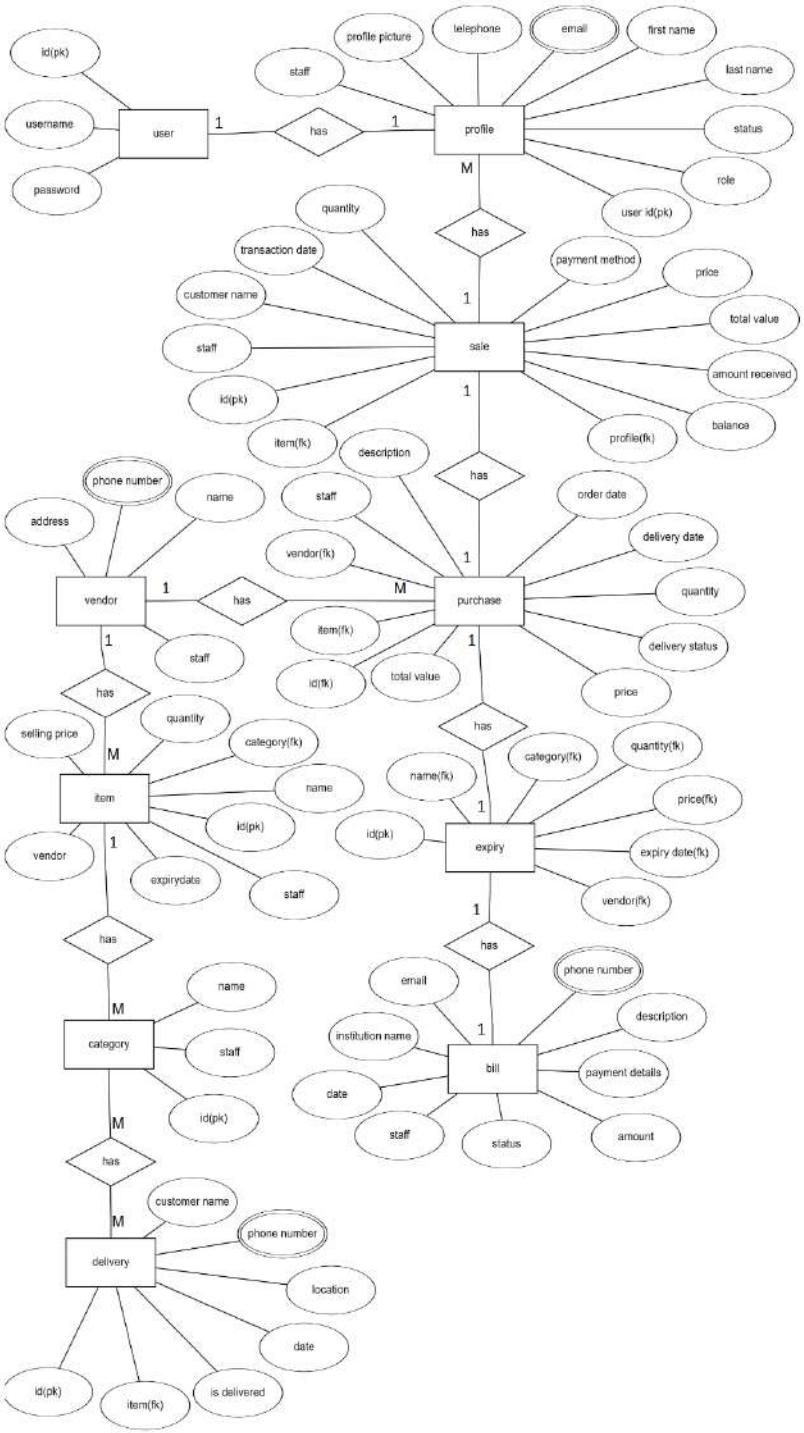
B.3 Level 2



B.4 Level 3

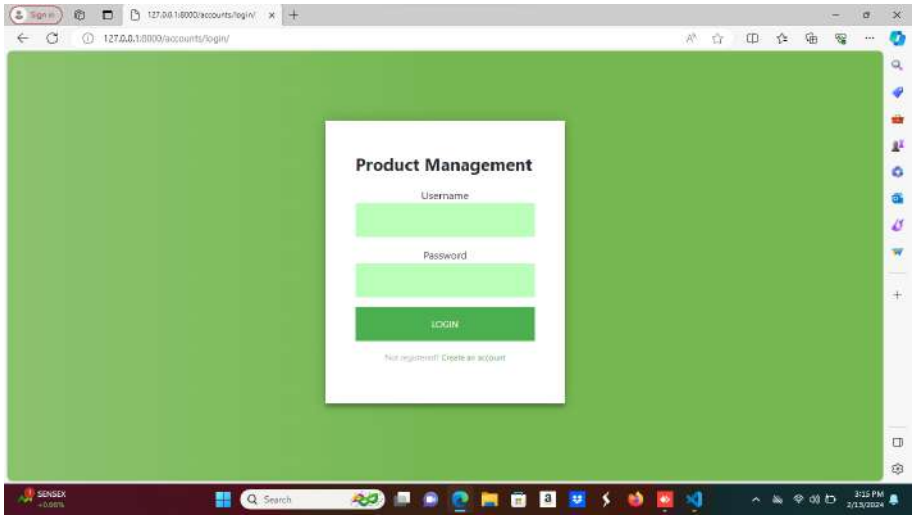


ER Diagram

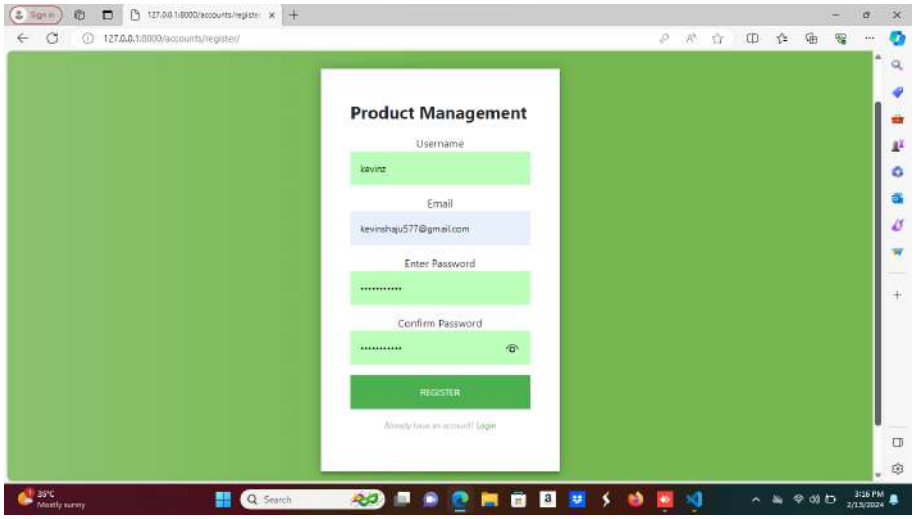

id(pk)
username
password
user
1
has
1
staff
profile picture
telephone
email
first name
last name
profile
status
role
user id(pk)
M
has
quantity
transaction date
customer name
staff
id(pk)
item(fk)
description
sale
payment method
price
total value
amount received
balance
profile(fk)
1
has
phone number
name
address
vendor
1
has
M
staff
vendor(fk)
order date
delivery date
quantity
delivery status
price
purchase
item(fk)
id(fk)
total value
1
has
selling price
quantity
category(fk)
name
id(pk)
item
M
1
vendor
expirydate
staff
has
name(fk)
category(fk)
quantity(fk)
price(fk)
expiry date(fk)
vendor(fk)
id(pk)
expiry
1
has
name
staff
id(pk)
category
M
has
email
phone number
description
institution name
payment details
date
staff
status
amount
bill
customer name
phone number
location
date
delivery
id(pk)
item(fk)
is delivered

C USER INTERFACES

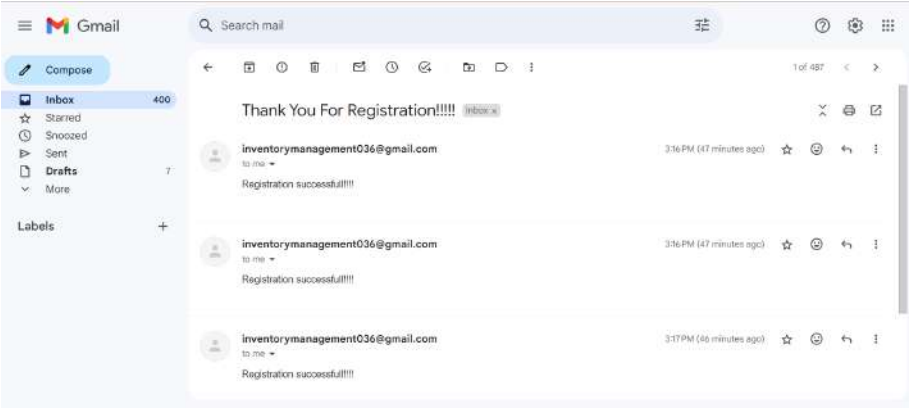
C.1 LOGIN



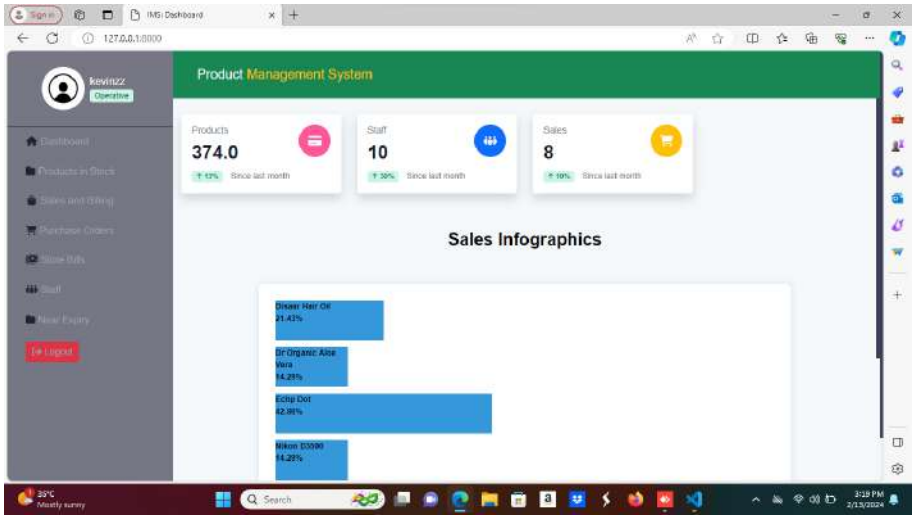
C.2 REGISTER


Product Management
Username
kevinz
Email
kevinshaju577@gmail.com
Enter Password
Confirm Password
REGISTER

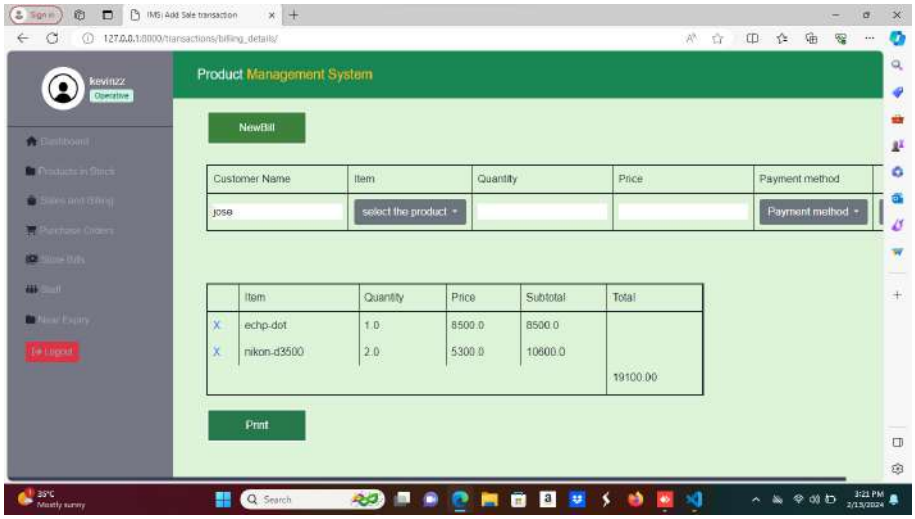
C.3 MAIL


Gmail
Search mail
Compose
Inbox 400
Starred
Snoozed
Sent
Drafts 7
More
Labels
Thank You For Registration!!!!!
inventorymanagement036@gmail.com
Registration successful!!!!
inventorymanagement036@gmail.com
Registration successful!!!!
inventorymanagement036@gmail.com
Registration successful!!!!

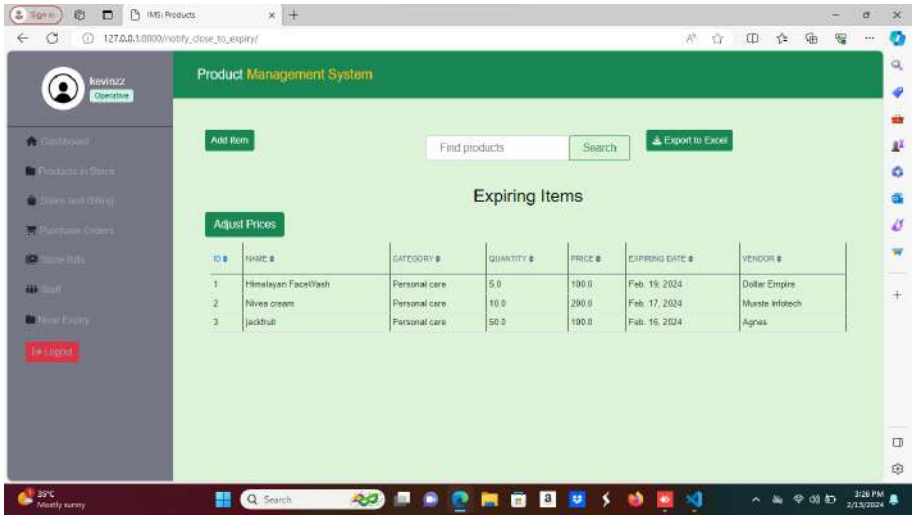
C.4 HOME


Product Management System
Products
374.0
Staff
10
Sales
8
Sales Infographics

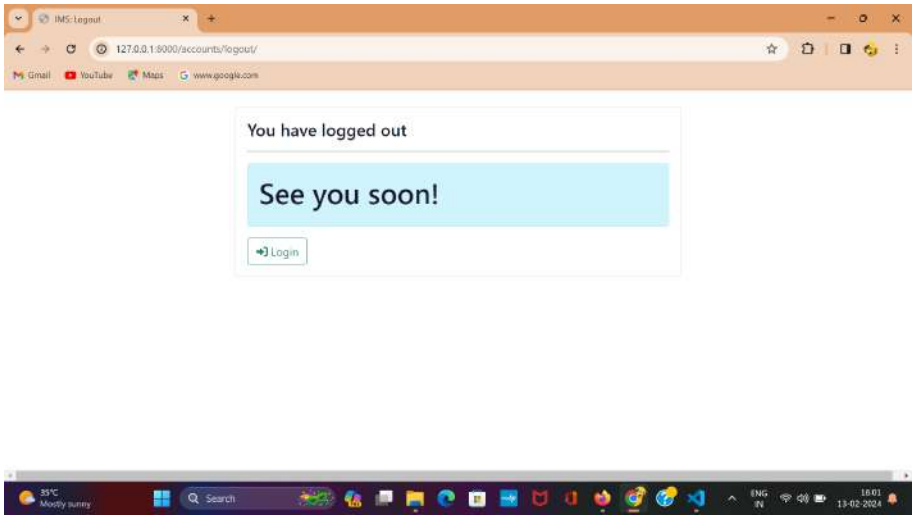
C.5 BILL


Product Management System
NewBill
Customer Name
Item
Quantity
Price
Payment method
jose
select the product
Payment method
Item
Quantity
Price
Subtotal
Total
echp-dot
1.0
8500.0
8500.0
nikon-d3500
2.0
5300.0
10600.0
19100.00
Print

C.6 EXPIRY


Product Management System
Add Item
Find products
Search
Export to Excel
Expiring Items
Adjust Prices
ID
NAME
CATEGORY
QUANTITY
PRICE
EXPIRING DATE
VENDOR
1 Himalayan FaceWash Personal care 5.0 100.0 Feb. 19, 2024 Dollar Empire
2 Nivea cream Personal care 10.0 200.0 Feb. 17, 2024 Murste Infotech
3 jackfruit Personal care 50.0 100.0 Feb. 16, 2024 Agnes

C.7 LOGOUT


IMS: Logout
127.0.0.1:8000/accounts/logout/
Gmail
YouTube
Maps
www.google.com
You have logged out
See you soon!
Login
35°C
Mostly sunny
Search
ENG
IN
16:01
13-02-2024

D CODE

login.html

```
[breaklines=true]
{% load crispy_forms_tags %}{% load static %}
<html>
  <head>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist
/css/bootstrap.min.css"
      rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3Q
DgpJLIm9Nao0Yz1ztcQTwFspd3yD
       65VohhpuuCOmLASjC"
      crossorigin="anonymous" />
  </head>
<body>
  <div class="login-page">
    <style>
    .login-page {
      width: 360px;
      padding: 8% 0 0;
      margin: auto;
    }
    .form {
      position: relative;
      z-index: 1;
      background: #FFFFFF;
      max-width: 360px;
      margin: 0 auto 100px;
      padding: 45px;
      text-align: center;
      box-shadow: 0 0 20px 0 rgba(0, 0, 0, 0.2), 0 5px 5px 0
 rgba(0, 0, 0, 0.24);
    }
    .form input {
      outline: 0;
      background: #bbffbb;
      width: 100%;
      border: 0;
      margin: 0 0 15px;
      padding: 15px;
      box-sizing: border-box;
      font-size: 14px;
    }
    .form button {
      text-transform: uppercase;
```

```
  outline: 0;
  background: #4CAF50;
  width: 100%;
  border: 0;
  padding: 15px;
  color: #FFFFFF;
  font-size: 14px;
  -webkit-transition: all 0.3 ease;
  transition: all 0.3 ease;
  cursor: pointer;
}
.form button:hover,.form button:active,.form button:focus {
  background: #43A047;
}
.form .message {
  margin: 15px 0 0;
  color: #b3b3b3;
  font-size: 12px;
}
.form .message a {
  color: #4CAF50;
  text-decoration: none;
}
.form .register-form {
  display: none;
}
.container {
  position: relative;
  z-index: 1;
  max-width: 300px;
  margin: 0 auto;
}
.container:before, .container:after {
  content: "";
  display: block;
  clear: both;
}
.container .info {
  margin: 50px auto;
  text-align: center;
}
.container .info h1 {
  margin: 0 0 15px;
  padding: 0;
  font-size: 36px;
  font-weight: 300;
```

```
      color: #1a1a1a;
    }
    .container .info span {
      color: #4d4d4d;
      font-size: 12px;
    }
    .container .info span a {
      color: #000000;
      text-decoration: none;
    }
    .container .info span .fa {
      color: #EF3B3A;
    }
    body {
      background: #76b852; /* fallback for old browsers */
      background: rgb(141,194,111);
      background: linear-gradient(90deg, rgba(141,194,111,1) 0%,
 rgba(118,184,82,1) 50%);
      -webkit-font-smoothing: antialiased;
      -moz-osx-font-smoothing: grayscale;
    }
    </style>
    <div class="form">
      <div>
        <strong class="navbar-brand" style="font-size: 26px;">
Product<span class=
             "text-warning">ms</span></strong>
      </div>
      <div class="text-danger">
        {% if messages %} {% for message in messages %}
        <div class="alert alert-{{message.tags}}">
            {{ message }}
        </div>
        {% endfor %} {% endif %}

    {% if form.non_field_errors %}
        <ul>
            {% for error in form.non_field_errors %}
                <li>{{ error }}</li>
            {% endfor %}
        </ul>
    {% endif %}
  </div>
      <form class="login-form" method="POST">
        {% csrf_token %}
        <p>{{ message }}</p>
```

```
        <label>Username</label>
        <span class="text-danger">{{ form.username.errors }}
</span>
        {{form.username}}
        <label>Password</label>
        <span class="text-danger">{{ form.password.errors }
}</span>
        {{form.password}}
        <button  type="submit" name="button">login
</button>
        <p class="message">Not registered? <a href=
            "{% url 'user-register' %}">Create an account
</a></p>
      </form>
    </div>
  </div>
</body>
</html>
```

register.html

```
{% load crispy_forms_tags %}{% load static %}
<html>
  <head>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@
5.0.2/dist/css/bootstrap.min.css"
      rel="stylesheet" integrity="sha384-EVSTQN3/azpr
G1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD
                    65VohhpuuCOmLASjC"
      crossorigin="anonymous" />
  </head>
<body>
  <div class="login-page">
    <style>
    .login-page {
      width: 360px;
      padding: 8% 0 0;
      margin: auto;
    }
    .form {
      position: relative;
      z-index: 1;
      background: #FFFFFF;
      max-width: 360px;
```

```
      margin: 0 auto 100px;
      padding: 45px;
      text-align: center;
      box-shadow: 0 0 20px 0 rgba(0, 0, 0, 0.2), 0
5px 5px 0 rgba(0, 0, 0, 0.24);
    }
    .form input {
      outline: 0;
      background: #bbffbb;
      width: 100%;
      border: 0;
      margin: 0 0 15px;
      padding: 15px;
      box-sizing: border-box;
      font-size: 14px;
    }
    .form button {
      text-transform: uppercase;
      outline: 0;
      background: #4CAF50;
      width: 100%;
      border: 0;
      padding: 15px;
      color: #FFFFFF;
      font-size: 14px;
      -webkit-transition: all 0.3 ease;
      transition: all 0.3 ease;
      cursor: pointer;
    }
    .form button:hover,.form button:active,
.form button:focus {
      background: #43A047;
    }
    .form .message {
      margin: 15px 0 0;
      color: #b3b3b3;
      font-size: 12px;
    }
    .form .message a {
      color: #4CAF50;
      text-decoration: none;
    }
    .form .register-form {
      display: none;
    }
    .container {
```

```
      position: relative;
      z-index: 1;
      max-width: 300px;
      margin: 0 auto;
    }
    .container:before, .container:after {
      content: "";
      display: block;
      clear: both;
    }
    .container .info {
      margin: 50px auto;
      text-align: center;
    }
    .container .info h1 {
      margin: 0 0 15px;
      padding: 0;
      font-size: 36px;
      font-weight: 300;
      color: #1a1a1a;
    }
    .container .info span {
      color: #4d4d4d;
      font-size: 12px;
    }
    .container .info span a {
      color: #000000;
      text-decoration: none;
    }
    .container .info span .fa {
      color: #EF3B3A;
    }
    body {
      background: #76b852; /* fallback for old browsers */
      background: rgb(141,194,111);
      background: linear-gradient(90deg,
rgba(141,194,111,1) 0%, rgba(118,184,82,1) 50%);
      -webkit-font-smoothing: antialiased;
      -moz-osx-font-smoothing: grayscale;
    }
    </style>
    <div class="form">
      <div>
        <strong class="navbar-brand" style=
"font-size: 26px;">Inventory<span class=
       "text-warning">ms</span></strong>
```

```
      </div>
      <div class="text-danger">
        {% if messages %} {% for message in messages %}
        <div class="alert alert-{{message.tags}}">
            {{ message }}
        </div>
        {% endfor %} {% endif %}

    {% if form.non_field_errors %}
        <ul>
            {% for error in form.non_field_errors %}
                <li>{{ error }}</li>
            {% endfor %}
        </ul>
    {% endif %}
  </div>
      <form class="login-form" method="POST">
        {% csrf_token %}
        <p>{{ message }}</p>
        <label>Username</label>
        <span class="text-danger">
{{ form.username.errors }}</span>
        {{form.username}}
        <label>Email</label>
        <span class="text-danger">{{ form.email.errors }}
</span>
        {{form.email}}
        <label>Enter Password</label>
        <span class="text-danger">{{ form.password.errors }}
</span>
        {{form.password1}}
        <label>Confirm Password</label>
        <span class="text-danger">{{ form.password2.errors }}
</span>
        {{form.password2}}
        <button  type="submit" name="button">Register
</button>
     <p class="message">Already have an account? <a href=
                            "{% url 'user-login' %}">Login</a></p>
      </form>
    </div>
  </div>
</body>
</html>
```

style.css

```
form input, select {
 outline: 0;
 background: #bbffbb;
 width: 100%;
 border: 0;
 margin: 0 0 15px;
 padding: 15px;
 box-sizing: border-box;
 font-size: 14px;
  }
#body-row {
 margin-left: 0;
 margin-right: 0;
}

#sidebar-container {
 min-height: 100vh;
 background-color: #4CAF50;
 padding: 0;
 /* flex: unset; */
}

.sidebar-expanded {
 width: 230px;
}

.sidebar-collapsed {
 /*width: 60px;*/
 width: 100px;
}

/* ----------| Menu item*/
#sidebar-container .list-group a {
 height: 50px;
 color: white;
}

/* ----------| Submenu item*/
#sidebar-container .list-group li.list-group-item {
 background-color: #4CAF50;
}

#sidebar-container .list-group .sidebar-submenu a {
 height: 45px;
 padding-left: 30px;
}
```

```
/* ----------| Separators */
.sidebar-separator-title {
 background-color: #4CAF50;
 height: 35px;
}

.sidebar-separator {
 background-color: #4CAF50;
 height: 25px;
}

.logo-separator {
 background-color: #4CAF50;
 height: 60px;
}

a.bg-dark {
 background-color: #4CAF50 !important;
}

@import 'https://fonts.googleapis.com/css2?
family=Inter:wght@300;400;500;600&display=
                swap" rel="stylesheet';

:root {
--dk-gray-100: #F3F4F6;
--dk-gray-200: #E5E7EB;
--dk-gray-300: #D1D5DB;
--dk-gray-400: #9CA3AF;
--dk-gray-500: #6B7280;
--dk-gray-600: #4B5563;
--dk-gray-700: #374151;
--dk-gray-800: #1F2937;
--dk-gray-900: #111827;
--dk-dark-bg: #313348;
--dk-darker-bg: #2a2b3d;
--navbar-bg-color: #6f6486;
--sidebar-bg-color: #252636;
--sidebar-width: 250px;
}

* {
margin: 0;
padding: 0;
box-sizing: border-box;
}
```

```
body {
font-family: 'Inter', sans-serif;
background-color: #def4d9;
}

a {
text-decoration: none;
link-style: none;
}
#wrapper {
margin-left: var(--sidebar-width);
transition: all .3s ease-in-out;
}

#wrapper.fullwidth {
margin-left: 0;
}



/** ---------------------------------
-- Sidebar
--------------------------------- */
.sidebar {
font-size: .925rem;
background-color: var(--sidebar-bg-color);
width: var(--sidebar-width);
transition: all .3s ease-in-out;
transform: translateX(0);
z-index: 9999999
}

.sidebar .close-aside {
position: absolute;
top: 7px;
right: 7px;
cursor: pointer;
color: #EEE;
}

.sidebar .sidebar-header {
border-bottom: 1px solid #2a2b3c
}

.sidebar .sidebar-header h5 a {
```

```
color: var(--dk-gray-300)
}

.sidebar .sidebar-header p {
color: var(--dk-gray-400);
font-size: .825rem;
}

.sidebar .search .form-control ~ i {
color: #2b2f3a;
right: 40px;
top: 22px;
}

.sidebar > ul > li {
padding: .7rem 1.75rem;
}

.sidebar ul > li > a {
color: var(--dk-gray-400);
text-decoration: none;
}

/* Start numbers */
.sidebar ul > li > a > .num {
line-height: 0;
border-radius: 3px;
font-size: 14px;
padding: 0px 5px
}

.sidebar ul > li > i {
font-size: 18px;
margin-right: .7rem;
color: var(--dk-gray-500);
}

.sidebar ul > li.has-dropdown > a:after {
content: '\eb3a';
font-family: unicons-line;
font-size: 1rem;
line-height: 1.8;
float: right;
color: var(--dk-gray-500);
transition: all .3s ease-in-out;
}
```

```
.sidebar ul .opened > a:after {
transform: rotate(-90deg);
}


.show-sidebar {
transform: translateX(-270px);
}

@media (max-width: 767px) {
.sidebar ul > li {
 padding-top: 12px;
 padding-bottom: 12px;
}

.sidebar .search {
 padding: 10px 0 10px 30px
}
}




/** ---------------------------------
-- welcome
--------------------------------- */
.welcome {
color: var(--dk-gray-300);
}

.welcome .content {
background-color: var(--dk-dark-bg);
}

.welcome p {
color: var(--dk-gray-400);
}




/** ---------------------------------
-- Statistics
--------------------------------- */
.statistics {
```

```
color: var(--dk-gray-200);
}

.statistics .box {
background-color: var(--dk-dark-bg);
}

.statistics .box i {
width: 60px;
height: 60px;
line-height: 60px;
}

.statistics .box p {
color: var(--dk-gray-400);
}


/** --------------------------------
-- Please don't do that in real-world projects!
-- overwrite Bootstrap variables instead.
-------------------------------- */

.navbar {
border: none !important;
}
.navbar .dropdown-menu {
right: auto !important;
left: 0 !important;
}
.navbar .navbar-nav>li>a {
color: #EEE !important;
line-height: 55px !important;
padding: 0 10px !important;
}
.navbar .navbar-brand {color:#FFF !important}
.navbar .navbar-nav>li>a:focus,
.navbar .navbar-nav>li>a:hover {color:
 #EEE !important}

.navbar .navbar-nav>.open>a,
.navbar .navbar-nav>.open>a:focus,
.navbar .navbar-nav>.open>a:hover
{background-color: transparent !important;
color: #FFF !important}
```

```
.navbar .navbar-brand {line-height: 55px
 !important; padding: 0 !important}
.navbar .navbar-brand:focus,
.navbar .navbar-brand:hover {color:
#FFF !important}
.navbar>.container .navbar-
brand, .navbar>.container-fluid .navbar-brand
               {margin: 0 !important}
@media (max-width: 767px) {
.navbar>.container-fluid .navbar-brand {
 margin-left: 15px !important;
}
.navbar .navbar-nav>li>a {
 padding-left: 0 !important;
}
.navbar-nav {
 margin: 0 !important;
}
.navbar .navbar-collapse,
.navbar .navbar-form {
 border: none !important;
}
}

.navbar .navbar-nav>li>a {
float: left !important;
}
.navbar .navbar-nav>li>a>span:not(.caret) {
background-color: #e74c3c !important;
border-radius: 50% !important;
height: 25px !important;
width: 25px !important;
padding: 2px !important;
font-size: 11px !important;
position: relative !important;
top: -10px !important;
right: 5px !important
}
.dropdown-menu>li>a {
padding-top: 5px !important;
padding-right: 5px !important;
}
.navbar .navbar-nav>li>a>i {
font-size: 18px !important;
}
```

```
/* Start media query */

@media (max-width: 767px) {
#wrapper {
 margin: 0 !important
}
.statistics .box {
 margin-bottom: 25px !important;
}
.navbar .navbar-nav .open
     .dropdown-menu>li>a {
 color: #CCC !important
}
.navbar .navbar-nav .open .
      dropdown-menu>li>a:hover {
 color: #FFF !important
}
.navbar .navbar-toggle{
 border:none !important;
 color: #EEE !important;
 font-size: 18px !important;
}
.navbar .navbar-toggle:focus, .navbar
 .navbar-toggle:hover
     {background-color: transparent !important}
}


::-webkit-scrollbar {
background: transparent;
width: 5px;
height: 5px;
}

::-webkit-scrollbar-thumb {
background-color: #3c3f58;
}

::-webkit-scrollbar-thumb:hover {
background-color: rgba(0, 0, 0, 0.3);
}
```

```
//sorting
table {
background-color: #000 !important;
}
table.dataTable thead .sorting:after,
table.dataTable thead .sorting:before,
table.dataTable thead .sorting_asc:after,
table.dataTable thead .sorting_asc:before,
table.dataTable thead .sorting_asc_disabled:after,
table.dataTable thead .sorting_asc_disabled:before,
table.dataTable thead .sorting_desc:after,
table.dataTable thead .sorting_desc:before,
table.dataTable thead .sorting_desc_disabled:after,
table.dataTable thead .sorting_desc_disabled:before {
bottom: .5em;
}

//update profile
 /*=========================================================
            Form Section
  =========================================================*/
  form {
    margin: 0 auto;
    width: 50%;
    border-bottom-right-radius: 6px;
    border-bottom-left-radius: 6px;
    padding-bottom: 10px;
  }
  /*changing title style*/
  h1 {
    padding-top: 30px;
    text-align: center;
    color: #FF5722;
    font-size: 26px;
  }

  /*Camera image to upload image*/
  .fa-camera-retro {
      color: #a7a7a7;
      width: 100px;
      height: 100px;
      margin-top: 20px;
      line-height: 100px;
      border: 2px solid #a7a7a7;
      cursor: pointer;
      -webkit-border-radius: 50%;
```

```
     border-radius: 50%;
     box-shadow: 0 15px 20px -10px
rgba(0, 0, 0, 0.3);
 }

 .fa-camera-retro:hover{
   border-color: #6f6f6f;
   box-shadow: 0 3px 5px -5px rgba(0, 0, 0, 0.3);
   transform: translateY(-1px);
   color: #6f6f6f;
 }



 /*=========================================================
           input Section
 ==========================================================*/

 /*Float Label main style*/
 .float-label input, .float-label select {
     -webkit-appearance: none;
     outline: none;
     border: none;
     margin: 0 auto;
     width: 100%;
     display: block;
     -moz-border-radius: 0;
     border-radius: 0;
     font-family: 'Lato', sans-serif;
     font-size: 18px;
     border-bottom: 1px solid rgba(0, 0, 0, 0.7);
     background: transparent;
     color: #757575;
     padding: 5px 15px 7px 10px;
 }

 .title {
   font-family: 'Lato', sans-serif;
   font-size: 18px;
   background: transparent;
   color: #757575;
   font-weight: normal;
   margin-left: 7px;
 }

 .gender, .birthday {
```

```
    text-align: left;
    padding-left: 16px;
    margin-top: -10px;
}

.gender div, .birthday div {
  display: inline-block;
  padding-left: 0;
}

/*add style to radio options*/
.option-input {
  -webkit-appearance: none;
  -moz-appearance: none;
  -ms-appearance: none;
  -o-appearance: none;
  appearance: none;
  position: relative;
  top: 13.33333px;
  right: 0;
  bottom: 0;
  left: 0;
  height: 15px;
  width: 15px;
  transition: all 0.15s ease-out 0s;
  background: #cbd1d8;
  border: none;
  color: #fff;
  cursor: pointer;
  display: inline-block;
  margin-right: 0.5rem;
  outline: none;
  position: relative;
  z-index: 1000;
}
.option-input:hover {
  background: #9faab7;
}
.option-input:checked {
  background: #77cee2;
}
.option-input:checked::before {
  height: 15px;
  width: 15px;
  position: absolute;
  content: '';
```

```
  display: inline-block;
  font-size: 12px;
  text-align: center;
  line-height: 15px;
}
.option-input:checked::after {
  -webkit-animation: click-wave 0.65s;
  -moz-animation: click-wave 0.65s;
  animation: click-wave 0.65s;
  background: #77cee2;
  content: '';
  display: block;
  position: relative;
  z-index: 100;
}
.option-input {
  border-radius: 50%;
}
.option-input::after {
  border-radius: 50%;
}
.radio-inline input[type=radio] {
  position: static;
  margin-left: 0;
  padding-left: 0;
}
.radio-inline {
  vertical-align: baseline;
}
.radio-inline span {
  line-height: 4;
  font-size: 16px;
  padding-left: 4px;
}

 /*Style birthday select */
.float-label select {
  display: inline-block;
  width: 24%;
  margin-left: 20px;
  font-size: 16px;
  color: black;
}
.float-label select:first-child{
  width: 30%;
}
```

```
.birthday .select {
  width: 82%;
}

.float-label > select option:first-child {
  color: #77cee2;
}


#updateProfile {
  padding: 20px 60px;
}

.user-left {
  padding-top: 40px;
}

 /*create a round container to hide overflow image*/
.user-left span {
  display: inline-block;
  width: 250px;
  height: 250px;
  overflow: hidden;
  text-align: center;
  border-radius: 100%;
  /*add shadow and border*/
  border: 2px solid #a9a9a9;
  cursor: pointer;
  -webkit-border-radius: 50%;
  border-radius: 50%;
  box-shadow: 0px 20px 25px -10px rgba(0, 0, 0, 0.3);
}

/*change image effect when hover*/
.user-left span:hover{
  border-color: #b3b3b3;
  box-shadow: 0 3px 5px -5px rgba(0, 0, 0, 0.3);
  transform: translateY(-1px);
}

/*change font margin and padding*/
.user-left h1 {
  padding-top: 0;
}

/*center image and make it round*/
```

```
  .user-image {
    left: 50%;
    margin-left: -100%;
    position: relative;
    width: auto important;
    height: 250px important;
  }

  /*change image opacity*/
  .user-image:hover {
    opacity: 0.5;
    filter: alpha(opacity=50);
  }

  .user-right {
    padding: 20px 20px 40px;
    min-height: 200px;
    /*margin-left: 80px;*/
  }

  /*Change profile title*/
  .user-right h2 {
    text-align: left;
    font-family: 'Roboto', Arial, sans-serif;
    font-weight: 400;
    font-size: 3em;
  }
  .user-right span {
    color: #d2d2d2;
  }

  .user-info {
    padding: 18px;
    width: 100%;
    min-height: 200px;
    border-radius: 3px;
    box-shadow: 0 0 0 0 transparent;
    box-shadow: 0 15px 20px -15px
 rgba(0, 0, 0, 0.3), 0 55px 50px -35px
                          rgba(210, 214, 213, 0.5);

    box-shadow: 0px 2px 2px 2px
rgba(210, 214, 213, 0.4);
  }
```

```
  td p{
    font-size: 20px;
    padding: 10px 15px 10px 10px;
    text-align: left;
  }

  #signOut {
    float: right;
  }
```

logout.html

```
[verbetim]
{% load crispy_forms_tags %}
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta http-equiv="X-UA-Compatible"
content="IE=edge">
        <meta name="viewport"
 content="width=device-width, initial-scale=1.0">
        {% load static %}

        <link rel="stylesheet" href=
     "{% static 'fontawesome/css/all.css' %}"
 type="text/css">
        <link rel="stylesheet" type="text/css" href=
"https://unpkg.com/@webpixels
                                  /css@1.1.5/dist/index.css">
        <link href="https://cdn.jsdelivr.net/npm/
bootstrap@5.0.2/dist/css/
      bootstrap.min.css" rel="stylesheet" integrity=
   "sha384-EVSTQN3/azprG1Anm3QDgp
JLIm9Nao0Yz1ztcQTwFspd
       3yD65VohhpuuCOmLASjC"
 crossorigin="anonymous">
        <title>IMS: Logout</title>
    </head>
    <body>
       <div class="row my-4">
         <div class="col-md-6 offset-md-3">
            <div class="border p-3 bg-white">
             <h4>You have logged out</h4>
```

```
                <hr>
               <div class="alert alert-info">
                  <h1>See you soon!</h1>
                     </div>
          <a class="btn btn-outline-success" href=
"{% url 'user-login' %}"><i class=
"fa-solid fa-right-to-bracket"></i> Login</a>
                 </div>
            </div>
        </div>
    </body>
</html>
```

bill.html

```
{% extends "store/base.html" %}
{% load render_table from django_tables2 %}
{% load querystring from django_tables2 %}
{% load static %}{% block title %}Bills
{%endblock title%}{% block content %}
    <div class="col container p-5">
      <style>
        input[type="text"],select{
            box-sizing: border-box;
            border: 1px solid #ccc;
            border-radius: 4px;
            display: inline-block;
            padding: 6px 20px;
            margin: 8px 0;
        }
        button[type="submit"]{
          padding: 6px 20px;
          margin: 8px 0;
      }
      </style>
      {% comment %} <form method="GET"
class="container">
        {{filter.form}}
        <button type="submit" class="btn btn-sm
btn-success fa fa-search"> Search</button>
      </form> {% endcomment %}
      <div>
     <a class="btn btn-sm btn-success" href=
"{% url 'bill-create' %}">
```

```
                          Add a bill record</a>
     <a class="float-end btn btn-sm btn-success" href=
                            "{% querystring '_export'='xlsx' %}">
          <i class="fa-solid fa-download"></i>
          Export to Excel
        </a>
      </div>
      <div class="m-2">
        <table class= "table table-sm table-responsive">
            <thead>
              <tr class="table-success">
                <th scope="col"><a href=
"{% querystring table.prefixed_order_by_field
=column.order_by_alias.next %}">ID
 <i class="fa-solid fa-sort"></i></a></th>
 <th scope="col">Name <i class="fa-solid fa-sort">
</i></th>
<th scope="col">Description <i class="fa-solid fa-sort">
</i></th>
<th scope="col">Contact number <i class=
"fa-solid fa-sort"></i></th>
 <th scope="col">email<i class="fa-solid fa-sort"></i></th>
 <th scope="col">Payment details<i class="fa-solid fa-sort">
</i></th>
 <th scope="col">Amount <i class="fa-solid fa-sort">

</i></th>
 <th scope="col">Status <i class="fa-solid fa-sort"></i></th>
 <th scope="col"> Action</th>
              </tr>
            </thead>
            <tbody>
            {% for bill in bills %}
              <tr>
                <th scope="row"><a> {{bill.id}}</a></th>
                <td>{{bill.institution_name}}</td>
                <td>{{bill.description}}</td>
                <td>{{bill.phone_number}}</td>
                <td>{{bill.email}}</td>
                <td>{{bill.payment_details}}</td>
                <td>{{bill.amount}}</td>
                <td>{% if bill.status == True%}
                  <span class="badge badge-pill
bg-soft-success text-success me-2">
                    Paid
                  </span>
```

```
                  {% else %}
                  <span class="badge badge-pill bg-soft
-danger text-danger me-2">
                    Pending
                  </span>
                  {% endif %}
                </td>
                <td>
                  <a class="text-info" href=
"{% url 'bill-update' bill.slug %}"><i class=
"fa-solid fa-pen"></i></a>
                  <a class="text-danger float-end" href=
"{% url 'bill-delete' bill.pk %}"><i class=
"fa-solid fa-trash"></i></a>
                </td>
              </tr>
            {% endfor %}
            </tbody>
          </table>
        </div>
          <div class="paginator">
            {% if is_paginated %}
            <ul class="pagination ">
                {% if page_obj.has_previous %}
                <li><a class="btn btn-sm accent-1
 btn-outline-success
 " href="?page={{ page_obj.previous_page_number }}
 ">&laquo;</a></li>
                {% else %}
                <li class="disabled "><span class=
"btn btn-sm btn-outline-success ">
                &laquo;</span></li>
             {% endif %} {% for i in paginator.page_range %}
 {% if page_obj.number == i %}
                <li class="active "><span class=
"btn btn-sm btn-success ">{{ i }}
                    <span class="sr-only ">(current
)</span></span>
                </li>
                {% else %}
      <li><a href="?page={{ i }} " class=
"btn btn-sm btn-outline-success ">{{ i }}</a></li>
                {% endif %} {% endfor %}
 {% if page_obj.has_next %}
                <li><a href="?page={{ page_obj.next_page_number }}
            " class="btn btn-sm btn-outline-success ">&raquo;</a></li>
```

```
                {% else %}
                <li class="disabled ">
<span class="btn btn-sm btn-outline-success ">
                &raquo;</span></li>
                {% endif %}
            </ul>
            {% endif %}
        </div>
    </div>
  {% endblock content %}
```

productlist.html

```
[breaklines=true]
{% extends "store/base.html" %}
{% load render_table from django_tables2 %}
{% load querystring from django_tables2 %}
{% load static %}
{% block title %}Products{%endblock title%}
{% block content %}
    <div class="col container p-5">
      <style>
        input[type="text"],select{
            box-sizing: border-box;
            border: 1px solid #ccc;
            border-radius: 4px;
            display: inline-block;
            padding: 6px 20px;
            margin: 8px 0;
        }
        button[type="submit"]{
          padding: 6px 20px;
          margin: 8px 0;
      }
      </style>
      <div class="row">
        <div class="col-md-4">
          <a class="btn btn-sm btn-success" href=
  "{% url 'product-create' %}">Add Item</a>
        </div>
        <div class="col-md-4">
          <form class="input-group" role=
"search " id="searchform" action=
"{% url 'item_search_list_view' %}
" method="get" accept-charset="utf-8">
            <div class="form-group">
```

```
                <div class="input-group ">
                    <input id="searchbox" name=
"q" type="text" class=
      "form-control " placeholder="Find products">
                    <span class="input-group-btn">
                      <button class=
                "btn btn-outline-success" type=
"submit">Search</i></button>
                    </span>
                </div>
            </div>
        </form>
        </div>
        <div class="col-md-4 float-end">
          <a class="btn btn-sm btn-success" href=
"{% querystring '_export'='xlsx' %}">
            <i class="fa-solid fa-download"></i>
            Export to Excel
          </a>
        </div>
      </div>
      <div class="m-2">
        <table class= "table table-sm table-responsive">
            <thead>
              <tr class="table-success">
                <th scope="col"><a href=
"{% querystring table.prefixed_order_by_field
=column.order_by_alias.next %}">ID
                                <i class="fa-solid fa-sort"></i></a></th>
                <th scope="col">Name <i class=
"fa-solid fa-sort"></i></th>
                <th scope="col">Category <i class=
"fa-solid fa-sort"></i></th>
                <th scope="col">Quantity <i class=
"fa-solid fa-sort"></i></i></th>
                <th scope="col">Price <i class=
"fa-solid fa-sort"></i></th>
                <th scope="col">Expiring date <i class=
"fa-solid fa-sort"></i></th>
                <th scope="col">Vendor <i class=
"fa-solid fa-sort"></i></th>
                <th scope="col"> Action</th>
              </tr>
            </thead>
            <tbody>
            {% for item in items %}
```

```
              <tr>
                <th scope="row"><a> {{item.id}}
</a></th>
                <td>{{item.name}}</td>
                <td>{{item.category}}</td>
                <td>{{item.quantity}}</td>
                <td>{{item.selling_price}}</td>
                <th scope="col">{{item.expiring_date}}
</th>
                <td>{{item.vendor}}</td>
                <td>
                  <a class="text-info" href=
"{% url 'product-update' item.slug %}">
<i class="fa-solid fa-pen"></i></a>
                  <a class="text-danger float-end" href=
"{% url 'product-delete' item.slug %}">
<i class="fa-solid fa-trash"></i></a>
                </td>
              </tr>
            {% endfor %}
            </tbody>
          </table>
        </div>
          <div class="paginator">
            {% if is_paginated %}
            <ul class="pagination ">
                {% if page_obj.has_previous %}
                <li><a class="btn btn-sm accent-1
btn-outline-success
       " href="?page={{ page_obj.previous_page_number }
} ">&laquo;</a></li>
                {% else %}
                <li class="disabled "><span class=
"btn btn-sm btn-outline-success ">
             &laquo;</span></li>
                {% endif %} {% for i in paginator.page_range %}
                        {% if page_obj.number == i %}
                <li class="active "><span class=
"btn btn-sm btn-success ">{{ i }}
                 <span class="sr-only ">(current)
</span></span>
                </li>
                {% else %}
                <li><a href="?page={{ i }} " class=
"btn btn-sm btn-outline-success ">
                      {{ i }}</a></li>
```

```
                {% endif %} {% endfor %} {% if page_obj.has_next %}
                <li><a href="?page={{ page_obj.next_page_number }}
               " class="btn btn-sm btn-outline-success ">&raquo;</a></li>
                {% else %}
                <li class="disabled "><span class=
"btn btn-sm btn-outline-success ">
          &raquo;</span></li>
                {% endif %}
            </ul>
            {% endif %}
        </div>
    </div>
  {% endblock content %}
```

manage.py

```
#!/usr/bin/env python
"""Django's command-line utility for administrative tasks."""
import os
import sys


def main():
    """Run administrative tasks."""
    os.environ.setdefault
('DJANGO_SETTINGS_MODULE', 'InventoryMS.settings')
    try:
        from django.core.management import
 execute_from_command_line
    except ImportError as exc:
        raise ImportError(
            "Couldn't import Django. Are you sure it's installed and "
            "available on your PYTHONPATH environment
 variable? Did you "
            "forget to activate a virtual environment?"
        ) from exc
    execute_from_command_line(sys.argv)


if __name__ == '__main__':
    main()
```

module.py

```
from django.db import models
```

```
from store.models import Item
from accounts.models import Profile, Vendor
from django_extensions.db.fields import AutoSlugField

# Create your models here.

PAYMENT_CHOICES = [
    ('MP', 'MPESA'),
    ('VISA', 'VISA'),
    ('CS', 'CASH'),
    ('VM', 'VOOMA'),
    ('BK', 'BANK')
]

DELIVERY_CHOICES = [
    ('P', 'PENDING'),
    ('S', 'SUCCESSFUL')
]

class Sale(models.Model):
    slug = AutoSlugField(unique=True , populate_from=
                'customer_name')
    item = models.ForeignKey(Item, on_delete=
                    models.CASCADE, blank=True, null=True)
    customer_name =
          models.CharField(max_length=20, null=True, blank=True)
    transaction_date =
       models.DateTimeField(auto_now=True, blank=True, null=True)
    quantity = models.FloatField(default=0.00, blank=False, null=False)
    payment_method = models.CharField(choices=
        PAYMENT_CHOICES, max_length=20, blank='True', null=True)
    price = models.FloatField(default=0.00, blank=False, null=False)
    total_value = models.FloatField(blank=True, null=True)
    amount_received = models.FloatField(default=False,
          blank=False, null=False)
    balance = models.FloatField(default=False, blank=False, null=False)
    profile = models.ForeignKey(Profile, verbose_name=
        ('Served by'), on_delete=models.CASCADE)

    def save(self, *args, **kwargs):
        amt_received = self.amount_received
        price = self.price
        quantity = self.quantity
        self.total_value = price * quantity
        self.balance = amt_received - self.total_value
        super().save(*args, **kwargs)
```

```
    def __str__(self):
        return str(self.item.name)

class Purchase(models.Model):
    slug = AutoSlugField(unique=True , populate_from='vendor')
    item = models.ForeignKey(Item, on_delete=models.CASCADE)
    description = models.TextField(max_length=300
, blank=True, null=True)
    vendor = models.ForeignKey
  (Vendor, related_name='vendor', on_delete=models
.CASCADE, blank=False, null=False)
    order_date = models.DateTimeField(auto_now_add=True)
    delivery_date = models.DateTimeField
    (auto_now=False, auto_now_add=False, blank=True,
null=True, verbose_name=
                                                    ('Delivery Date'))
    quantity = models.FloatField(default=0.00, blank=False, null=False)
    delivery_status = models.CharField(choices=
DELIVERY_CHOICES, max_length=3, default='P', blank=False,
                      null=False, verbose_name=
('Delivery Status'))
    price = models.FloatField(default=0.00, blank=False,
      null=False, verbose_name=
('Price per item(Ksh)'))
    total_value = models.FloatField()

    def save(self, *args, **kwargs):
        quantity = self.quantity
        price = self.price
        self.total_value = price * quantity
        return super().save(*args, **kwargs)

    def __str__(self):
        return self.item.name

    class Meta:
        ordering = ["order_date"]
```

setting.py

```
import os
from pathlib import Path
```

```
# Build paths inside the project like this:
 BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent


# Quick-start development settings -
 unsuitable for production
# See https://docs.djangoproject.com/
en/4.1/howto/deployment/checklist/

# SECURITY WARNING: keep the secret
 key used in production secret!
SECRET_KEY = 'django-insecure
-g_n2+2bznu6e@1wel!i
(&-4tp86_7lop5395ww+i4x%9*7^old'

# SECURITY WARNING:
 don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = []


# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',

    'phonenumber_field',
    'crispy_forms',
    'imagekit',
    'django_extensions',
    'django_filters',
    'django_tables2',

    'store.apps.StoreConfig',
    'accounts.apps.AccountsConfig',
    'transactions.apps.TransactionsConfig',
    'invoice.apps.InvoiceConfig',
    'bills.apps.BillsConfig',
```

```
]

MIDDLEWARE = [
  'django.middleware.security.SecurityMiddleware',
 'django.contrib.sessions.middleware.SessionMiddleware',
 'django.middleware.common.CommonMiddleware',
'django.middleware.csrf.CsrfViewMiddleware',
 'django.contrib.auth.middleware.AuthenticationMiddleware',
'django.contrib.messages.middleware.MessageMiddleware',
'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'InventoryMS.urls'

TEMPLATES = [
    {
 'BACKEND': 'django.template.backends
.django.DjangoTemplates',
        'DIRS': [],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
       'django.template.context_processors.debug',
        'django.template.context_processors.request',
         'django.contrib.auth.context_processors.auth',
          'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]

WSGI_APPLICATION = 'InventoryMS.wsgi.application'


# Database
# https://docs.djangoproject.com/
en/4.1/ref/settings/#databases


DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    }
}
```

```
# Password validation
# https://docs.djangoproject.com/
en/4.1/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.
password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth
.password_validation.MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth
.password_validation.CommonPasswordValidator',
    },
    {
        'NAME': 'django.contrib.auth.
password_validation.NumericPasswordValidator',
    },
]


# Internationalization
# https://docs.djangoproject.com/en/4.1/topics/i18n/

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_TZ = True

LOGIN_URL = 'user-login'
LOGIN_REDIRECT_URL = 'dashboard'
LOGOUT_URL = 'logout'

# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/4.1/howto/static-files/

STATIC_URL = 'static/'
STATICFILES_DIRS = [os.path.join(BASE_DIR,'static')
```

```
]
MEDIA_ROOT = os.path.join(BASE_DIR, 'static/images')
MEDIA_URL = '/images/'

# Default primary key field type
# https://docs.djangoproject.com/
en/4.1/ref/settings/#default-auto-field

DEFAULT_AUTO_FIELD =
 'django.db.models.BigAutoField'

# CRISPY_TEMPLATE_PACK = 'bootstrap4'



EMAIL_BACKEND =
 'django.core.mail.backends.smtp.EmailBackend'
EMAIL_HOST = 'smtp.gmail.com'
EMAIL_USE_TLS = True
EMAIL_PORT = 587
EMAIL_HOST_USER =
 "product management system753@gmail.com"
EMAIL_HOST_PASSWORD = "oyhzlplyolihopys"


# FCM_DJANGO_SETTINGS = {
#     "FCM_SERVER_KEY":
"BC7LHSnv6csnz2VKw5DMASW6n
D2n_FiCsQwjBPUmlOgN1AM
-Qehh2qslANJudWf7R-P1UbL
KwYZJyxu8OiGJwWE",
# }
```

setup.sh

```
[breaklines=true]
#!/bin/bash

echo " Setting up sales-and-inventory-management"

echo " Building Docker image..."
docker build -t sales-and-inventory-management:1.0 .
```

```
if [ $? -ne 0 ]; then
  echo " Error: Docker image build failed!"
  exit 1
fi

echo " Starting Docker containers in detached mode..."
docker run -d -p 8000:8000
 sales-and-inventory-management:1.0

if [ $? -ne 0 ]; then
  echo " Error: Docker container failed to start!"
  exit 1
fi

echo " Setup completed successfully!"
```

signal.py

```
[breaklines=true]
from django.db.models.signals import post_save
from django.dispatch import receiver

from django.contrib.auth.models import User
from .models import Profile


@receiver(post_save, sender=User)
def create_profile(sender, instance,
 created, **kwargs):

    if created:
        Profile.objects.create(user=instance)
        print('profile created!')

@receiver(post_save, sender=User)
def update_profile(sender, instance,
 created, **kwargs):

    if created == False:
        instance.profile.save()
        print('profile updated!')
```

table.py

```
[breaklines=true]
```

```
# tutorial/tables.py
import django_tables2 as tables
from .models import Sale, Purchase
from django.shortcuts import render

class SaleTable(tables.Table):
    class Meta:
        model = Sale
        template_name =
 "django_tables2/semantic.html"
       fields = ('item', 'customer_name',
'transaction_date', 'payment_method', 'quantity',
                    'price', 'total_value', 'amount_received',
 'balance', 'profile')
        order_by_field = 'sort'

class PurchaseTable(tables.Table):
    class Meta:
        model = Purchase
        template_name = "django_tables2/semantic.html"
        fields = ('item', 'vendor', 'order_date',
         'delivery_date', 'quantity',
          'delivery_status', 'price', 'total_value')
        order_by_field = 'sort'
```

url.py

```
[breaklines=true]
from django.urls import path
from . import views
from django.conf.urls.static import static
from django.conf import settings
from .views import (
    PurchaseListView,
    PurchaseDetailView,
    PurchaseCreateView,
    PurchaseUpdateView,
    PurchaseDeleteView,
    SaleListView,
    SaleDetailView,
    SaleCreateView,
    SaleUpdateView,
    SaleDeleteView,
)

urlpatterns = [
```

```
    path('purchases/',
PurchaseListView.as_view(), name="purchaseslist"),
    path('purchase/<slug:slug>/',
PurchaseDetailView.as_view(), name='purchase-detail'),
    path('new-purchase/',
PurchaseCreateView.as_view(), name='purchase-create'),
    path('purchase/<int:pk>/update/',
 PurchaseUpdateView.as_view(), name='purchase-update'),
    path('purchase/<int:pk>/delete/',
 PurchaseDeleteView.as_view(), name='purchase-delete'),
    path('sales/',SaleListView.as_view(), name="saleslist"),
    path('sale/<int:pk>/',
 SaleDetailView.as_view(), name='sale-detail'),
    path('new-sale/', SaleCreateView.as_view(),
 name='sale-create'),
    path('sale/<slug:slug>/update/',
SaleUpdateView.as_view(), name='sale-update'),
    path('sale/<slug:slug>/delete/',
 SaleDeleteView.as_view(), name='sale-delete'),

]
urlpatterns += static(settings.MEDIA_URL,document_roo
t=settings.MEDIA_ROOT)
```

view.py

```
[breaklines=true]
from django.shortcuts import render
from .models import *
from django.contrib.auth.models import User
from .filters import PurchaseFilter, SaleFilter
from django.core.paginator import Paginator, EmptyPage,
 PageNotAnInteger
from django.contrib.auth.mixins import LoginRequiredMixin
, UserPassesTestMixin
from django.views.generic.edit import FormMixin
from django_tables2 import SingleTableView
import django_tables2 as tables
from django.urls import reverse
from django.shortcuts import get_object_or_404
from django_tables2.export.views import ExportMixin
from django_tables2.export.export import TableExport
from .tables import PurchaseTable, SaleTable
from django.core.exceptions import ValidationError
from accounts.models import Profile
from django.views.generic import (
```

```
    ListView,
    DetailView,
    CreateView,
    UpdateView,
    DeleteView
)

class PurchaseListView(ExportMixin, tables.SingleTableView):
    """View to list purchases and export them."""
    model = Purchase
    table_class = SaleTable
    template_name = 'transactions/purchases_list.html'
    context_object_name = 'purchases'
    paginate_by = 10
    SingleTableView.table_pagination = False

class PurchaseDetailView(FormMixin, DetailView):
    """View to display details of a purchase."""
    model = Purchase
    template_name = 'transactions/sale_detail.html'

    def get_success_url(self):
        return reverse('sale-detail', kwargs={'slug':
        self.object.slug})

class PurchaseCreateView(LoginRequiredMixin, CreateView):
    """View to create a new purchase."""
    model = Purchase
    template_name = 'transactions/purchasescreate.html'
    fields = ['item', 'description', 'vendor', 'delivery_date'
, 'quantity', 'delivery_status']

    def form_valid(self, form):
        """Handles the form submission and updates the item's
 quantity."""
        item = form.cleaned_data['item']
        quantity = form.cleaned_data['quantity']

        total_value = item.selling_price * quantity

        form.instance.total_value = total_value
        form.instance.price = item.selling_price

        form.instance.balance = total_value

        form.instance.profile = self.request.user.profile
```

```
        item.quantity += quantity
        item.save()

        return super().form_valid(form)

    def get_success_url(self):
        return reverse('purchaseslist')

    def test_func(self):
        profile_list = Profile.objects.all()
        if self.request.user.profile in profile_list:
            return False
        else:
            return True

class PurchaseUpdateView(LoginRequiredMixin,
UserPassesTestMixin, UpdateView):
    """View to update a purchase."""
    model = Purchase
    template_name = 'transactions/purchaseupdate.html'
    fields = ['item', 'description', 'vendor', 'delivery_date',
 'quantity', 'price', 'delivery_status']

    def test_func(self):
        profiles = Profile.objects.all()
        if self.request.user.profile in profiles:
            return True
        else:
            return False

    def get_success_url(self):
            return reverse('purchaseslist')


class PurchaseDeleteView(LoginRequiredMixin,
 UserPassesTestMixin, DeleteView):
    """View to delete a purchase."""
    model = Purchase
    template_name = 'transactions/purchasedelete.html'

    def test_func(self):
        profiles = Profile.objects.all()
        if self.request.user.profile in profiles:
            return True
        else:
```

```
            return False
    def get_success_url(self):
            return reverse('purchaseslist')

class SaleListView(ExportMixin, tables.SingleTableView):
    """View to list sales and export them."""
    model = Sale
    table_class = SaleTable
    template_name = 'transactions/sales_list.html'
    context_object_name = 'sales'
    paginate_by = 10
    SingleTableView.table_pagination = False

class SaleDetailView(DetailView):
    """View to display details of a sale."""
    model = Sale
    template_name = 'transactions/saledetail.html'


class SaleCreateView(LoginRequiredMixin, CreateView):
    """View to create a new sale."""
    model = Sale
    template_name = 'transactions/salescreate.html'
    fields = ['item', 'customer_name', 'payment_method',
        'quantity', 'amount_received']

    def get_success_url(self):
        return reverse('saleslist')

    def form_valid(self, form):
        """Handles the form submission and
              validates product availability."""
        item = form.cleaned_data['item']
        quantity = form.cleaned_data['quantity']

        if item.quantity < quantity:
 raise ValidationError(f"Only {item.quantity} units of
     '{item.name}' are available.")

        price = item.selling_price

        total_price = price * quantity

        form.instance.price = price
        form.instance.total_value = total_price
```

```
        amount_received = form.cleaned_data['amount_received']
        balance = amount_received - total_price
        form.instance.balance = balance

        form.instance.profile = self.request.user.profile
        return super().form_valid(form)

    def test_func(self):
        profile_list = Profile.objects.all()
        if self.request.user.profile in profile_list:
            return False
        else:
            return True

class SaleUpdateView(LoginRequiredMixin, UserPassesTestMixin,
 UpdateView):
    """View to update a sale."""
    model = Sale
    template_name = 'transactions/sale_update.html'
    fields = ['item', 'customer_name', 'payment_method',
      'quantity', 'price',
                                              'amount_received']

    def test_func(self):
        """Checks if the user has the required permissions to
 access this view."""
        profiles = Profile.objects.all()
        if self.request.user.profile in profiles:
            return True
        else:
            return False

    def get_success_url(self):
        return reverse('saleslist')

    def form_valid(self, form)
 """Handles form submission and sets the profile of the sale."""
        form.instance.profile = self.request.user.profile
        return super().form_valid(form)

class SaleDeleteView(LoginRequiredMixin, UserPassesTestMixin,
 DeleteView):
    """View to delete a sale."""
    model = Sale
    template_name = 'transactions/saledelete.html'
```

```
    def test_func(self):

        """Checks if the user has the required permissions to
     access this view."""
        if self.request.user.is_superuser:
            return True
        else:
            return False
```

GET SCHEME

PROJECT REPORT

Submitted By

HIBIN DIXON

Reg. No. CCAVBCA037

for the award of the Degree of

Bachelor of Computer Application

(University of Calicut)

under the guidance of

Ms. Rasmi P.M

Assistant Professor

BACHELOR OF COMPUTER APPLICATION
DEPARTMENT OF COMPUTER SCIENCE
CHRIST COLLEGE(Autonomous)
IRINJALAKUDA, KERALA
INDIA

2021-2024

DEPARTMENT OF COMPUTER SCIENCE

CHRIST COLLEGE (AUTONOMOUS)

IRINJALAKUDA

CERTIFICATE

This is to certify that the project report entitled ***"Get Scheme"*** *is a bonfied record of the project work done by* ***Hibin Dixon*** *in partial fulfillment of the requirement for the sixth semester of* ***Bachelor of Computer Application*** *in Department of Computer Science of* **CHRIST COLLEGE (AUTONOMOUS) IRINJALAKUDA**

Ms. Rasmi P.M
Assistant Professor,CS
Internal Guide

Ms. Sini Thomas
Head of the Department
Computer Science

EXTERNAL EXAMINER

INTERNAL EXAMINER

DECLARATION

I hereby declare that this project work **"GET SCHEME"** submitted to Christ College (Autonomous) Irinjalakuda, affiliated to Calicut University in partial fulfillment of the requirement for the award of the Bachelor of Computer Application, is a record of original work done by me, under the guidance of Ms. RASMI P.M, Department of computer Science.

Place: Irinjalakuda HIBIN DIXON

ACKNOWLEDGEMENT

First and foremost i like to thank Lord almighty for his providence and for being the guiding light throughout the project. I wish to express my sincere gratitude to our beloved Department head for giving me all the facilities for our project. I take this opportunity to express my sincere gratitude to the class teacher Ms. SOUMYA P.S and head of the department Ms. SINI THOMAS who has been supported me throughout the course of this project. I am thankful for her aspiring guidance and valuable advice during the project work. I express my sincere thanks to my project guide Ms. RASMI P.M for supporting and guiding throughout the project.I would take this opportunity to specially thank all other faculty members for their constant and continuous motivation. Finally i would like to thank my family and friends for giving valuable advice and moral support throughout our project.

ABSTRACT

GET SCHEME is a mobile application designed to streamline access to government schemes for citizens . The app enables different categories of people to access various welfare programs offered by governments, categorized by sectors and demographics. Users can easily get schemes based on their eligibility criteria through the app, improving their access to social welfare opportunities and promoting inclusive governance.

Contents

Chapter 1

1 Introduction

Introducing our groundbreaking app: Get Scheme. Say goodbye to endless searches and confusion when it comes to accessing government assistance programs. Our app is your go-to resource for effortlessly discovering and accessing a multitude of government schemes tailored to your specific needs. Whether you're seeking financial aid, educational grants, healthcare benefits, or housing support, Get Scheme simplifies the process by providing a comprehensive database of available programs. With user-friendly navigation and personalized recommendations, we ensure that you never miss out on valuable opportunities for assistance. Take control of your access to government schemes today with our intuitive and empowering app.

.

1.1 Overview

Introducing our groundbreaking app: Get Scheme. Say goodbye to endless searches and confusion when it comes to accessing government assistance programs. Our app is your go-to resource for effortlessly discovering and accessing a multitude of government schemes tailored to your specific needs. Whether you're seeking financial aid, educational grants, healthcare benefits, or housing support, Get Scheme simplifies the process by providing a comprehensive database of available programs. With user-friendly navigation and personalized recommendations, we ensure that you never miss out on valuable opportunities for assistance. Take control of your access to government schemes today with our intuitive and empowering app.

Chapter 2

2 System Analysis

2.1 Purpose

The purpose of the Get Scheme app is to facilitate easy access to various government assistance programs and schemes, simplifying the process of discovering, managing, and applying for financial aid, educational grants, healthcare benefits, housing support, and more

2.1.1 Existing System

Currently, there is no existing application that comprehensively addresses the diverse needs of individuals seeking government assistance programs. While there are some platforms and websites that provide information about specific schemes or sectors, these offerings are often limited in scope and accessibility. Users may struggle to find relevant programs that match their unique circumstances, leading to frustration and disenchantment with the system.

2.1.2 Proposed System

The proposed system is a mobile application designed to simplify access to government assistance programs for individuals from various backgrounds and circumstances. Users will be able to register for an account, input personal details, and receive personalized recommendations for relevant schemes based on their eligibility criteria. The system will maintain a centralized database of government assistance programs across different sectors, providing real-time updates and deadlines. With a user-friendly interface, robust security measures, and community engagement initiatives, the platform aims to empower users to navigate the complexities of bureaucracy effectively and access the support they need to improve their lives.

2.2 Problem definition

The problem definition of the Get Scheme app revolves around the fragmented and often confusing process of accessing government assistance programs. Users currently face challenges in navigating multiple sources of information and eligibility criteria, leading to inefficiencies and missed opportunities for valuable assistance. The app aims to address these issues by providing a unified platform that simplifies the discovery, management, and application process for government schemes

2.3 Feasibility Study

After the problem is clearly understood and solutions proposed, the next step is to conduct the feasibility study. Feasibility study is defined as evaluation or analysis of the potential impact of the proposed project or program. The objective is to determine whether the proposed system is feasible. There are three aspects of feasibility study to which the proposed system is subjected as discussed below.

2.3.1 Technical Feasibility

Technical feasibility assess whether the current technical resources are sufficient for the new system. We can upgrade the level of technology for supporting our website. We check whether the proposed system can be implemented in the present system without supporting the existing hardware.

2.3.2 Economical Feasibility

Economic feasibility determines whether the time and money are available to develop the system. There is no additional hardware used to develop the site, it is inexpensive to build.

2.3.3 Operational Feasibility

Operational feasibility determines if the human resources are available to operate the system once it has been installed. No extra training is needed to use this system. Anyone who has knowledge in internet in english language can easily use the system. The resources that are required to implement or install are already available with the organization.

Chapter 3

3 Software Requirement Specification

3.1 Purpose

The purpose of defining software requirements for the Get Scheme app is to outline the specific functionalities, features, and performance criteria necessary to meet user needs effectively. This ensures that the development team understands the project objectives clearly and can design, implement, and test the software accordingly, ultimately delivering a robust and user-friendly application that fulfills its intended purpose of simplifying access to government schemes and enhancing financial well-being.

3.2 Scope

The scope of the Get Scheme project involves developing a user-friendly mobile application that consolidates information on government schemes, offers personalized recommendations, and guides users through the application process. It aims to streamline access to assistance programs, enhance financial literacy, and improve the overall user experience in navigating government initiatives.

3.3 Overall Description

The software requirements for our project aim to address the needs of individuals seeking access to government assistance programs in a user-friendly and efficient manner.The software requirements encompass the entire development lifecycle, from gathering user requirements to design, implementation, testing, deployment, and maintenance. The goal is to create a robust and user-centric platform that democratizes access to government assistance programs, promoting inclusivity, transparency, and efficiency.

3.3.1 Product Perspective

Our project operates within the broader context of government assistance programs and aims to streamline the process of accessing these programs for individuals.

3.3.2 Product Functionality

Our platform offers a range of functionalities aimed at simplifying the process of accessing government assistance programs and providing users with personalized support.

3.3.3 Users and Characteristics

Admin is able to add new schemes to the database.other category of users like farmers,students,government employees,women,military,sports can retrive the schemes based in their criteria of eligibilty.

3.4 Specific Requirements

3.4.1 Hardware Requirements

- System:Android Device
- Processor: Mediatek Helio g85
- Speed: Above 1GHz
- RAM capacity: 512 MB
- Hard Dsk drive: 2 GB

3.4.2 Software Requirements

- Operating System:Android
- Languages used: Python Django,Flutter
- Database : MySql
- Technologies used: HTML,Javascript,CSS

3.5 Functional Requirements

It contains three main modules.

- 1.Admin
- 2.User

Admin

An Admin account is used for adding new schemes. The new schemes which are added by the admin will be notified to the users through their registered email id.

User

The user can register or login the app.The new schemes based on the eligilbility criteria provided by the user will be displayed in the my scheme section.The new schemes which are added recently will be displayed in the new schemes section.The my profile section is used to edit the eligibilty criteria provided by the user.

3.6 Non Functional Requirements

Non-functional requirements define the overall qualities or attributes of the resulting system. Non-functional system place restrictions on the product being developed, the developed process, and specify external constraints that the product must meet. Examples of non-functional requirements include safety, security, usability, reliability and performance requirements. Project management issues (costs, time and schedule) are often considered as non-functional requirements. The principal non - functional constraints which are relevant to critical systems :

- Performance
- Security
- Safety
- Usability

Performance

Performance requirements concern the speed of operation of a system. Types of performance requirements :

- Response requirements (how quickly the system reacts to a user input).
- Throughput requirements (how much the system can accomplish within a specified amount of time).
- Availability requirements (is the system available for service when requested by end users). The speed of operation of this system is adequate for the requirements.

Reliability

- Reliability is the ability of a system to perform its required function under stated conditions for a specified period of time.
- Constraints on the runtime behavior of the system. This system is reliable because its functionalities can be done on the required conditions.

Safety

Safety requirements are not required which exclude unsafe situation from the possible solution of the system.

Usability

Usability is the ease with which a user can learn to operate, prepare inputs for, and interpret outputs of system or components. Usability requirements include :

- Information error messages.
- Well-formed user interfaces.

3.7 Interface Requirements

3.7.1 Hardware interfaces

The system must run over the internet, all the hardware shall require to connect internet will be hardware interface for the system. As for example modem, WAN-LAN.

3.7.2 Software interfaces

Software interface required for the working of the project is the appropriate operating system.

3.7.3 Communication interfaces

The user of the site communicate using the network connectivity and the data set is accessed

3.8 Security Requirements

- User accesses only their account.
- Validation of input is handled.
- This application containing the computer systems is physically secured against arms or surreptitious entry by intruders.
- Users must be authorized carefully to reduce changes of any such user giving access to an intruder in exchange for a bribe or other favour.

3.9 Platform Used

For the Get Scheme app project, the Android operating system serves as the foundation for delivering a seamless and accessible user experience. Leveraging the Android OS allows the development team to tap into a vast ecosystem of tools, libraries, and resources provided by Google, facilitating efficient app development and optimization for a wide range of devices. By adhering to Android's design guidelines and platform standards, the app ensures consistency and familiarity for users across different devices and versions of the OS. Moreover,

Android's robust security features enable the implementation of stringent measures to safeguard user data and privacy, instilling trust and confidence among users. Overall, the utilization of the Android OS in this project enables the delivery of a high-quality and feature-rich app that effectively serves its purpose of simplifying access to government schemes and enhancing financial well-being for users.

3.10 Technologies Used

Python Django

Python Django is a high-level web framework that simplifies the development of complex web applications by providing a clean and pragmatic design. It follows the "batteries-included" philosophy, offering built-in features for common web development tasks such as URL routing, database management, form handling, and user authentication. Django's emphasis on DRY (Don't Repeat Yourself) principle and its robust security features make it a popular choice for developers seeking efficiency, scalability, and rapid development. With its rich ecosystem of third-party packages and active community support, Django empowers developers to build sophisticated web applications with ease.

Flutter

Flutter is an open-source UI software development kit (SDK) created by Google for building natively compiled applications for mobile, web, and desktop from a single codebase. It uses the Dart programming language and offers a rich set of pre-designed widgets that enable developers to create visually stunning and highly performant user interfaces. Flutter's hot reload feature allows for rapid iteration and experimentation during development, resulting in faster development cycles. With its cross-platform capabilities and expressive UI framework, Flutter has gained popularity among developers for building beautiful and responsive applications across various platforms.

MySQL

MySQL, pronounced either "My S-Q-L" or "My Squel, " is an open source relational database management system. It is based on the structure query language (SQL), which is used for adding, removing and modifying information in the database. Standard SQL commands, such as ADD, DROP, INSERT,and UPDATE can be used with MySQL.

What distinguishes PHP from something like client - side Javascript is that the code is executed on the server, generating HTML which is then sent to the client. The client would receive the results of running that script, but would not know what the underlying code was. You can even configure your web server to process all your HTML files with PHP, and then there's really no way that users can tell what you have up your sleeve. The best things in using PHP are

that it is extremely simple for a newcomer, but offers many advanced features for a professional programmer.

Chapter 4

4 Design Document

4.1 Purpose

The purpose of the design document is to provide a comprehensive blueprint for the development team, outlining the architectural design, system components, and technical specifications required to build the software solution. This document serves as a roadmap that guides the development process from conception to implementation, ensuring alignment with project goals and requirements. By detailing the system's structure, interfaces, data flow, and functionality, the design document facilitates effective communication among stakeholders, fosters collaboration within the development team, and enables informed decision-making throughout the software development lifecycle. Additionally, the design document serves as a reference for future maintenance and updates, providing a clear understanding of the system's design rationale and facilitating efficient troubleshooting and modification as needed. Overall, the design document plays a crucial role in translating requirements into a coherent and actionable plan, ultimately leading to the successful delivery of a high-quality software solution that meets the needs of its users.

4.2 Scope

In the design document for the Get Scheme app project, the scope outlines the specific functionalities and features that the app will encompass. This includes defining the categories of government schemes to be covered, such as finance, education, healthcare, and housing. Additionally, the scope may detail the user interface design, including the layout of screens, navigation flow, and interactive elements.

4.3 Overview

the design document serves as a comprehensive guide for implementing the software solution, ensuring that all aspects of system design and architecture are properly documented and aligned with project requirements. It provides a roadmap for developers, designers, and testers to follow during the development process, facilitating efficient collaboration and successful project delivery.

4.4 Data Design

Database are the storehouses of data used in the software systems.The data is stored in tables inside the database.Several tables are created for the manipulation of the data for the system.Two essential settings for a database are

- Primary Key- The field that is unique for all the record occurrence.

- Foreign Key- The field used to set relation between tables.

Normalization

- Normalization is the process of organizing the data in the database.
- Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate the undesirable characteristics like Insertion, Update and Deletion Anomalies.
- Normalization divides the larger table into the smaller table and links them using relationship.
- The normal form is used to reduce redundancy from the database table.

Tables

User

Name	DataType	Constraints	Description
First name	char	Primarykey	Name
Lastname	char(100)	Notnull	Name
gender	char(250)	Notnul	Gender
Email	varchar(250)	Notnul	MailID
Password	varchar(250)	Notnul	Password

Scheme

Name	DataType	Constraints	Description
Scheme app	char	Primarykey	Name
Type	char	Notnull	Types
Description	varchar(250)	Notnul	Description
Start Age	int	Notnul	Age
End Age	int	Notnul	Age
Link	varchar	Notnul	Link

Chapter 5

5 Development of the System

The development of the Get Scheme system involves a systematic approach encompassing various stages. It begins with thorough requirement gathering, where project goals, functionalities, and user expectations are identified and documented. Following this, the design phase commences, where system architecture, database schema, and user interface designs are meticulously planned to ensure alignment with project requirements and user needs. Subsequently, developers begin coding the system, utilizing technologies like Django for backend development and Flutter for frontend implementation. Rigorous testing is then conducted to identify and rectify any issues, ensuring the system's reliability, functionality, and performance. Once testing is complete, the system is deployed to production environments, and ongoing maintenance and updates are performed to address evolving requirements and enhance user experience. Collaboration among developers, designers, testers, and stakeholders is crucial throughout the development process to ensure the successful delivery of the Get Scheme app.

Chapter 6

6 System Testing

Testing is the penultimate step of software development.An elaborate testing of the data is prepared and the system is using the test data,while doing testing,errors are noted and correction is made.The users are trained to operate the developed system.Both hardware and software securities are made to run the developed system successfully.System testing is aimed at ensuring the system works accurately before the live operation commences.

6.1 Test Plan

6.1.1 Scope

This test plan will cover the following testing activities as identified in the testing strategy.

- White Box Testing

White box testing is a testing technique, that examines the program structure and derives test data from the program logic/code. In white box testing, the UI is bypassed. Inputs and outputs of ratings are tested directly at the code level and the results are compared against specifications. It Reveals errors in "hidden" code.

- Black Box Testing

Black - box testing is a method of software testing that examines the functionality of an application based on the specifications. Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end - user would.

- Unit Testing

The module includes four main components to undergo unit testing. It checks the syntax error, logic error and validity of the program correctness. This test will be performed by the developers.

- Integration Testing

After all components pass the unit testing, they get to be tested whether they work correctly when they are running concurrently and communicating to each other. Integration tests exercise an entire subsystem and ensure that a set of components play nicely together.

- Internal Data Testing

We will test the validity of the data before it enters the database to avoid any problems that may face us in the database. We will test the encryption of the personal information of all the users along with the user names and passwords to ensure maximum security of the users privacy.

6.1.2 Software risk issues

The software risks of the Get Scheme app include potential security vulnerabilities, such as data breaches or unauthorized access to user information, which could compromise user privacy and trust. Additionally, software bugs or glitches may lead to system malfunctions or unexpected behavior, impacting user experience and satisfaction. Moreover, compatibility issues with different devices, operating systems, or third-party libraries could pose challenges in ensuring consistent functionality and performance across various platforms.

6.1.3 Features to be tested

- Test whether correct user name and password allows you to login.
- Test whether invalid user name and password prevents you from login.
- Test whether there is any connection problem in the Server.
- Test whether the user and admin details are entered correctly.
- Test whether invalid data entry allows saving data successfully.
- Test whether all pages are loaded correctly.
- Test whether the provided security works correctly.

6.2 Test consolidation

6.2.1 Test item

The items or features to be tested in the test cases are included in the document. Each and every user input is tested. The present condition of the system is tested. It is checked to make sure that environment is ready for the application to work.

6.2.2 Input specifications

These are the inputs required to execute the test case. All required inputs including data elements and values are given below.

Data	Values
Email	A valid email ID
Password	Combination of characters and numbers
Phone number	10 Digit number

Chapter 7

7 System Implementation and Maintenance

7.1 Implementation

System implementation is the conversion of new system into an operating one which involves creating compatible files, training clients and installing hardware. User training is crucial for minimizing resistance to change and giving chance to prove its worth. Training aids user friendly manuals and healthy screens provide the user with a good start. Software maintenance follows conversion to the extent that changes are necessary to maintain satisfactory operations relative to changes in the user's environment. Maintenance often includes minor enhancements or corrections to the problem that surface late in the systems operations. In the implementation phase, the team builds the components either from scratch or by composition. Given the architecture document meant from the design phase and the requirement document from the analysis phase, the team should build exactly what has been requested, though there is still room for innovation and flexibility. For example, a component may be narrowly designed for this particular system, or the component may be made more general to satisfy a reusability

- Careful planning
- Investigation of system and constraints
- Design the methods to achieve changeover.
- Training the staff in the changed phase.
- Evaluation of change over method.
- The method of implementation and time scale to be adopted are found out initially.

7.2 Maintenance

This phase occurs as a result of deploying the whole system at the end users organization. They will perform the beta testing at the end users and inform to the developers about any needed modification to the application. The customer records all the problems that are encountered during the beta testing and reports these to the developer at regular intervals.

7.2.1 Corrective Maintenance

Even with the best quality assurance activities, it is likely that the customer will uncover defects in the software. Corrective maintenance changes the software to correct the defects. Corrective Maintenance activity may consist of repair,

restoration or replacement of equipment. This activity will be result of a regular inspection, which identifies the failure in time for corrective maintenance to be planned and scheduled, then performed during a routine maintenance shutdown.

7.2.2 Adaptive Maintenance

Over time, the original environment(CPU, operating system, business rules, external product characteristics) for which the software was developed is likely to change. Adaptive maintenance results in modification to the software to accommodate changes to its external environment.

7.2.3 Enhanced Maintenance

As software is used, the customer/user will recognize additional functions that will provide the benefit. Perfect maintenance extends the software beyond its original functional requirements.

7.2.4 Preventive Maintenance

Computer software deteriorates due to change, and because of this preventive maintenance often called software re-engineering, must be conducted to enable the software to serve the needs of its end users. Preventive maintenance makes changes to computer programs so that they can be more easily corrected, adapted and enhanced.

Chapter 8

8 Conclusion and Future Scope

8.1 Conclusion

This app facilitate access to government schemes presents a significant opportunity to enhance citizen engagement and promote inclusivity in accessing essential services. By providing a user-friendly interface, timely updates, and personalized recommendations, the app can empower individuals to make informed decisions and avail themselves of relevant benefits. Moreover, such a platform can foster transparency, efficiency, and accountability within the government system, ultimately contributing to socio-economic development and welfare.

8.2 Future Scope

The future scope refers to the potential opportunities, advancements, and possibilities for growth, development, and expansion within a particular field, project, or concept. It encompasses various aspects, including technological advancements, market trends, policy changes, and societal needs, that can influence and shape future outcomes. Future scope entails exploring new avenues for innovation, addressing emerging challenges, and seizing opportunities for improvement and progress. It involves strategic planning, research, and forecasting to anticipate and capitalize on future trends and opportunities, thereby ensuring sustainable growth and success.

- Integration with Other Services: Integrating the app with other government services and platforms to create a comprehensive ecosystem for citizens to access various benefits seamlessly.
- Feedback Mechanisms: Establishing robust feedback mechanisms to gather insights from users, identify pain points, and continuously improve the app's usability and effectiveness.
- Partnerships and Collaborations: Collaborating with NGOs, community organizations, and private sector partners to amplify the app's impact, reach underserved populations, and leverage additional resources for implementation and promotion.
- Personalization and AI Recommendations: Utilizing artificial intelligence to provide personalized recommendations based on users' demographics, preferences, and past interactions with the app.

Appendix

A Data Flow Diagram

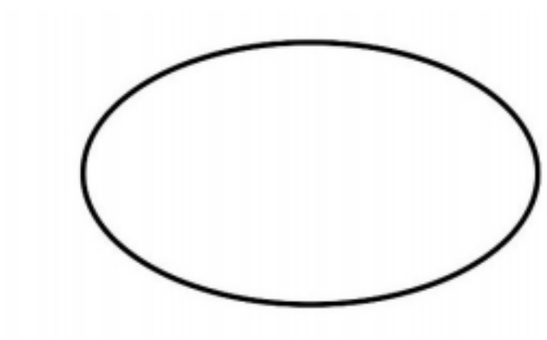
Data flow is the one of the best way of documenting the entire functionality of the system.For the system ,which will have data flows in and have some processing inside and then some data flow out from the system can be documented or represented effectively by means of data flow out from the system can be documented or represented effectively by means of data flow diagrams. The data flow diagram are a diagrammatic representation of the system,which has input,process and output.Once any system is represented using a data flow diagram we can identify the following things easily:

- Various entities interacting with the system are identified
- Flow of data from one entity to another is identified
- The various processes involved in between the interaction of two or more entities in the system are clearly pointed out
- The various data stores which hold the data in between the process,are clearly identified

Some Data Flow Diagram charting forms:

A.1 External source or receiver

A.2 Transform process

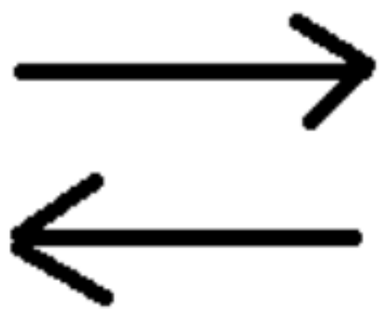


A process represents transformation where incoming data flowsare changed into outgoing data flow

A.3 Data Store

A data store is repository of data that is to be stored for use by one or more process may be as simple as buffer or queue or sophisticated as relational database. They should have clear names.If a process merely uses the contest of store and does not alter it,the arrowhead goes only form the store to the process. If a process alters the details in the store then double-headed arrow is used.

A.4 Data flow

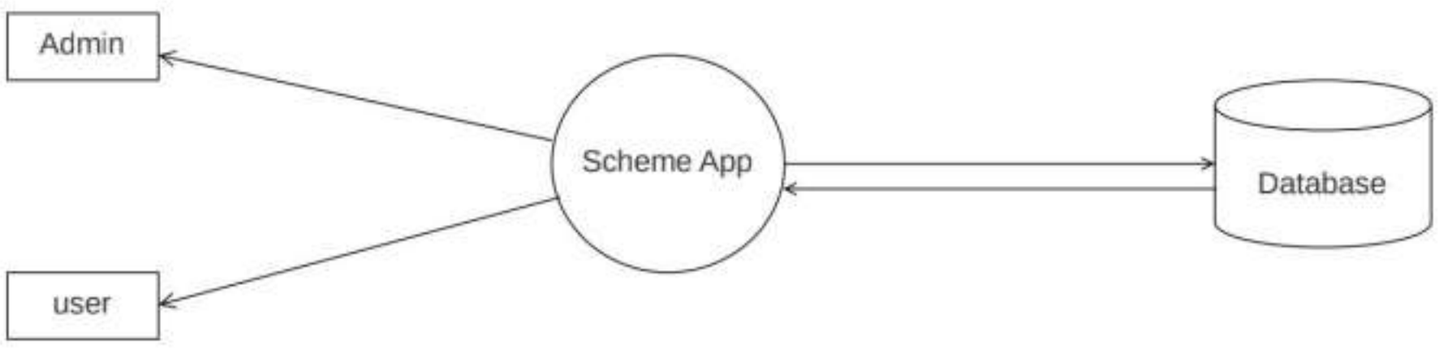


A data flow is a route, which enable packets of data to travel from one point to another.Data may flow,with arrowhead pointing in the direction of the flow

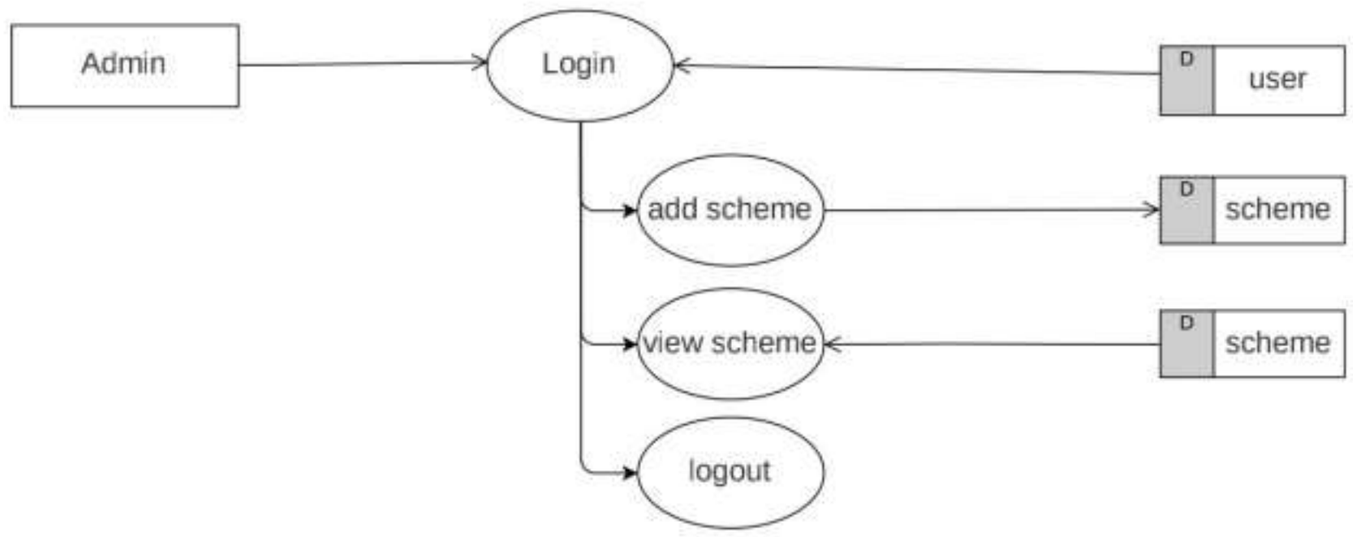
B Data Flow Diagrams

B.1 DFD1

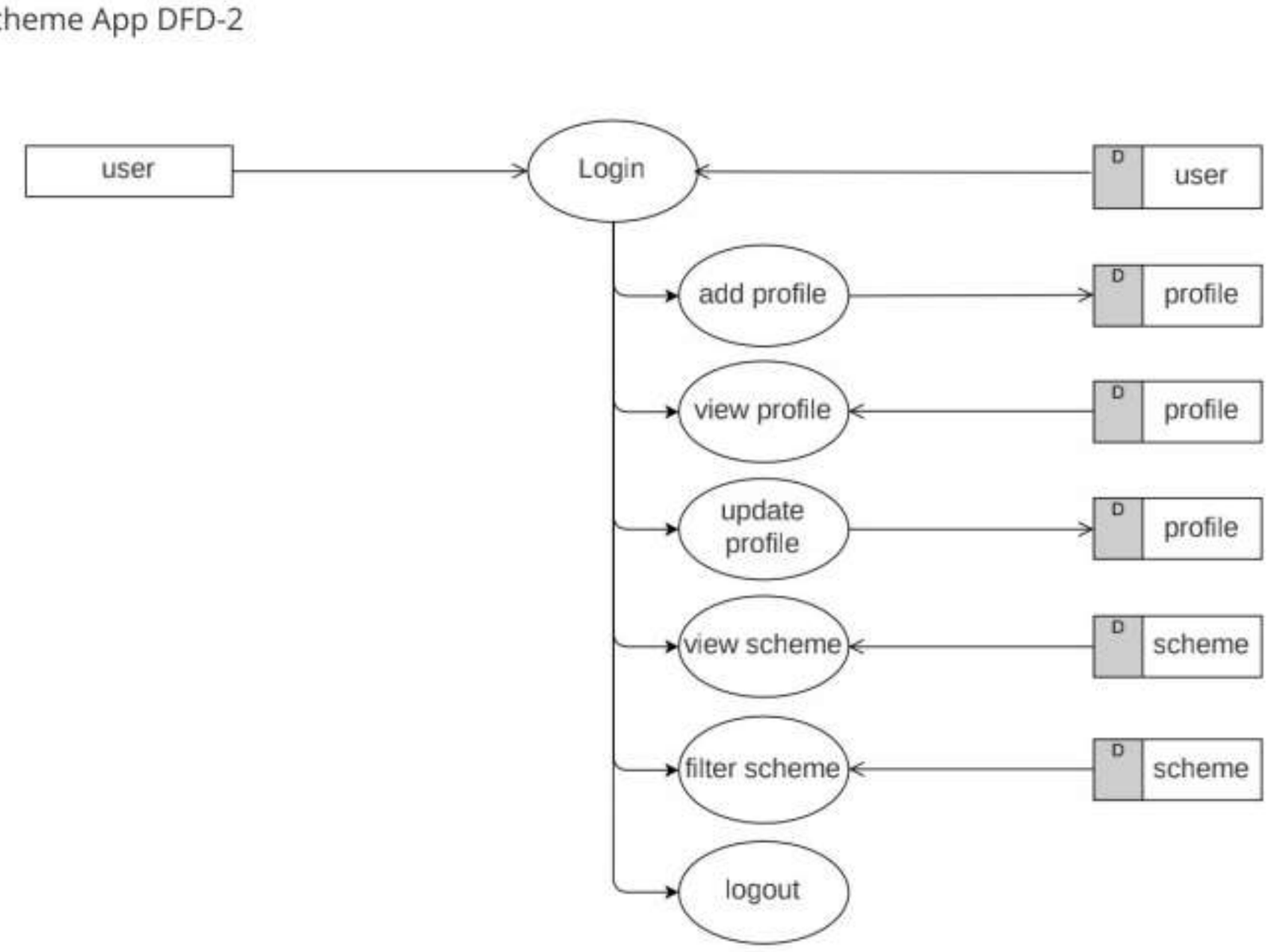
Scheme App DFD-0



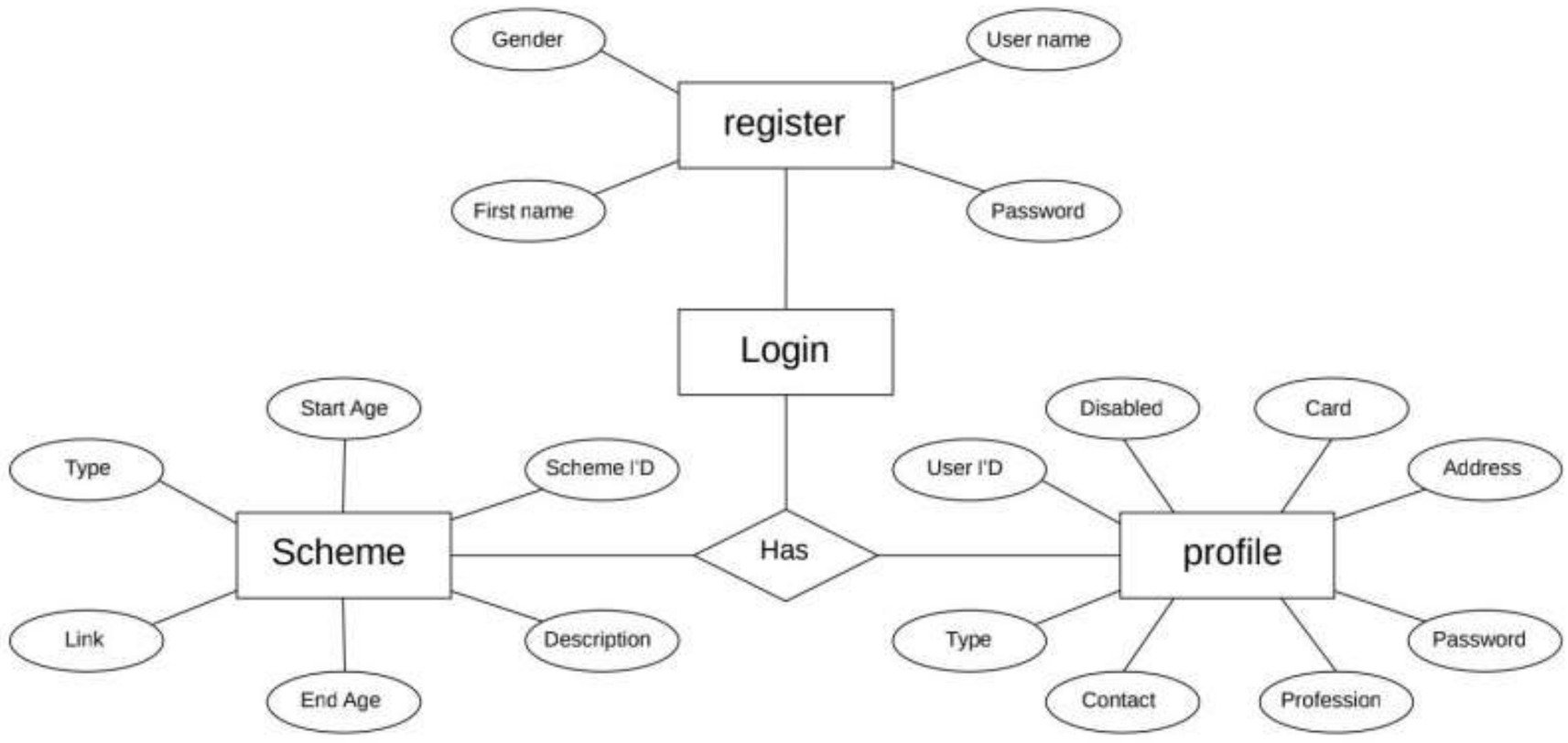
Scheme App DFD-1



B.2 DFD2

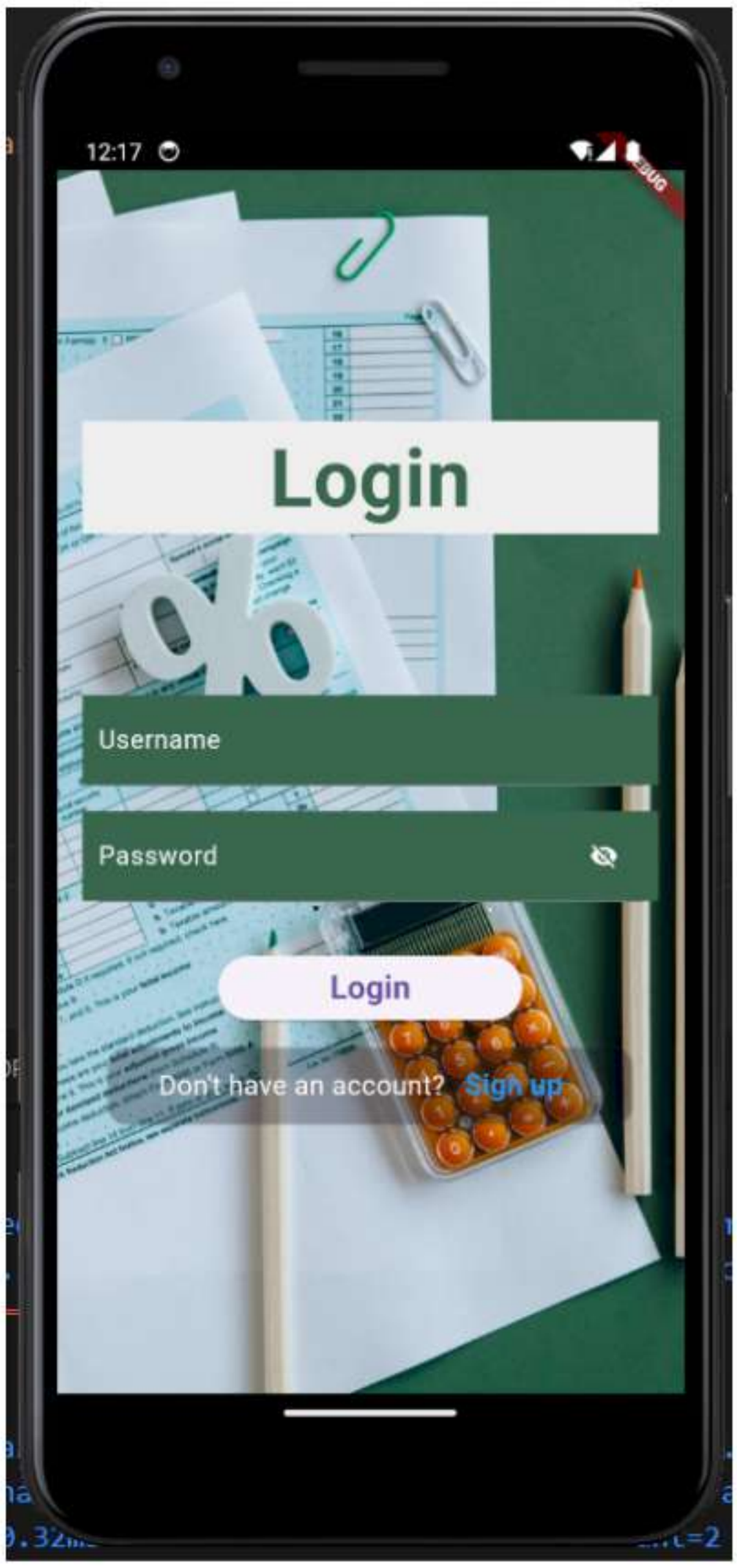

Scheme App DFD-2
user
Login
D
user
add profile
D
profile
view profile
D
profile
update profile
D
profile
view scheme
D
scheme
filter scheme
D
scheme
logout

B.3 ER Diagram

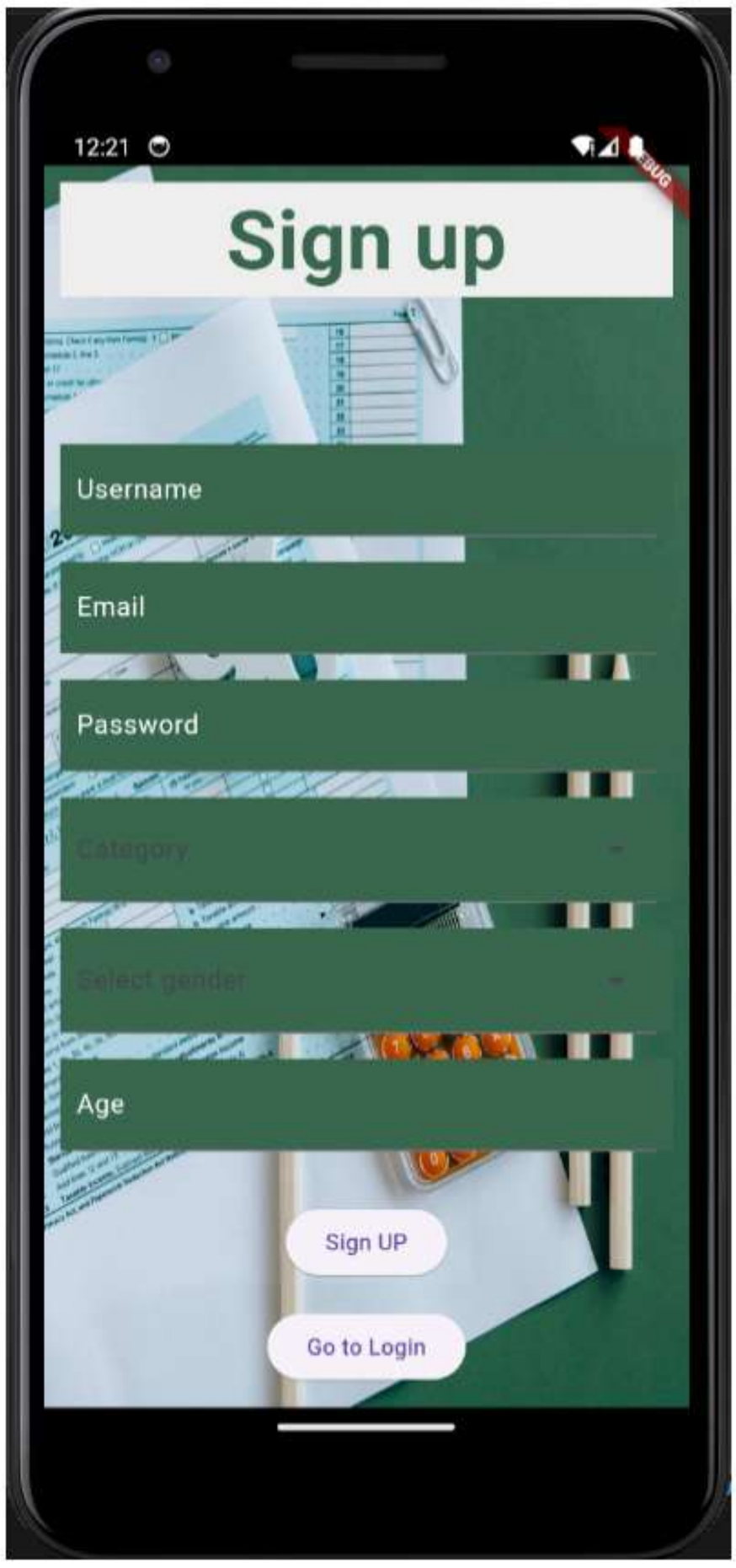


C USER INTERFACES

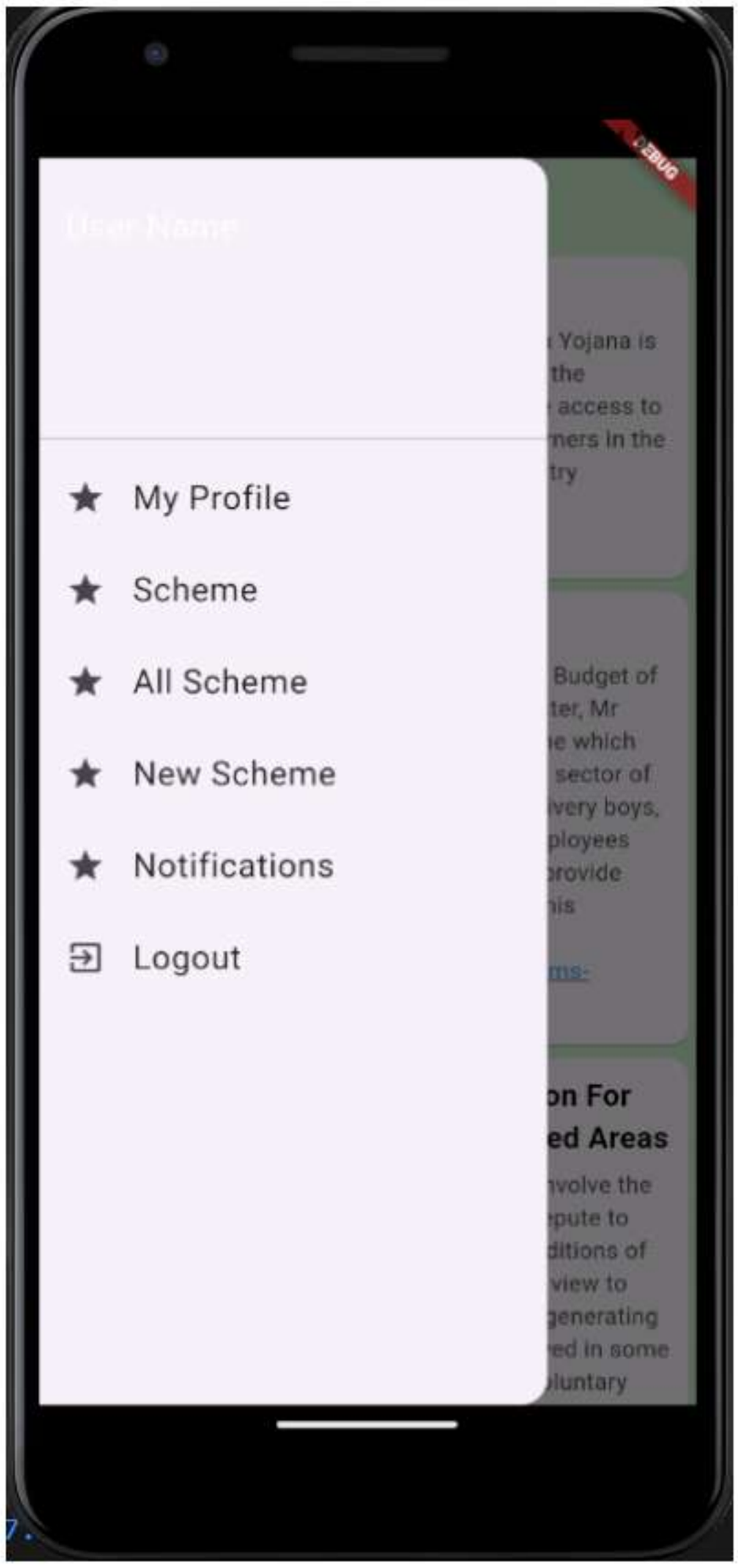
C.1 LOGIN



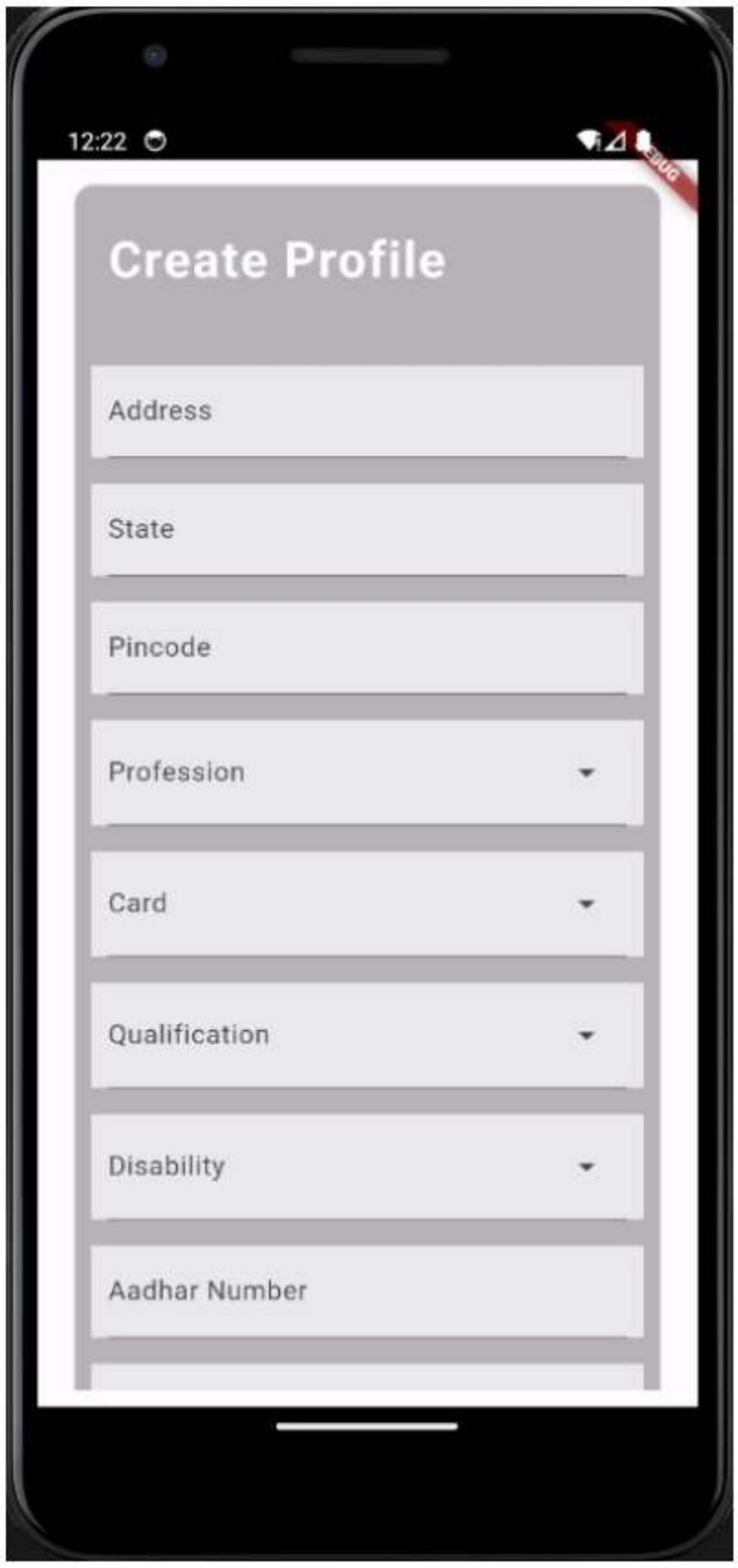
C.2 SIGNUP


12:21
DEBUG
Sign up
Username
Email
Password
Category
Select gender
Age
Sign UP
Go to Login

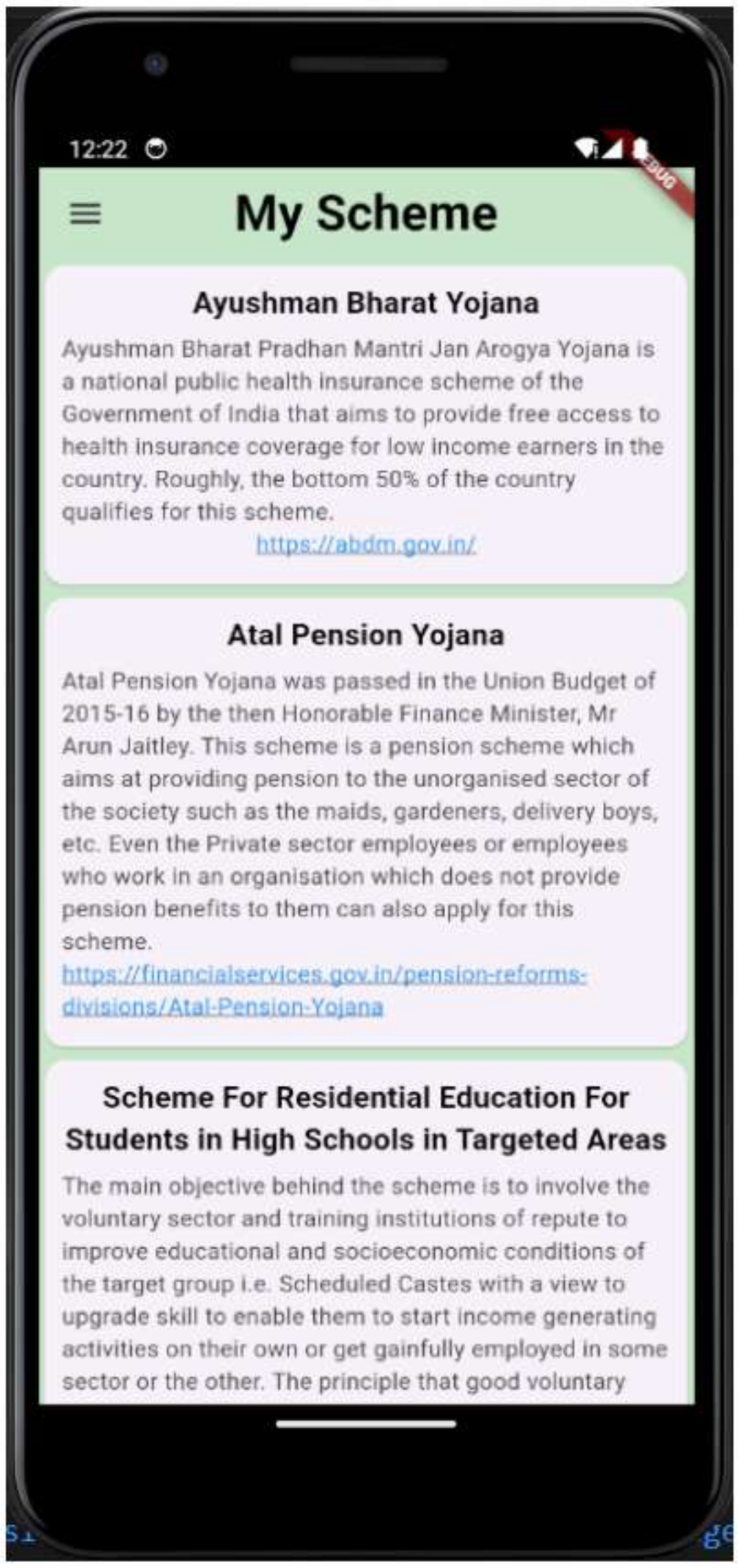
C.3 SIDE MENU


DEBUG
User Name
My Profile
Scheme
All Scheme
New Scheme
Notifications
Logout
Yojana is
the
access to
mers in the
try
Budget of
ter, Mr
e which
sector of
very boys,
ployees
provide
his
ms-
on For
ed Areas
nvolve the
pute to
ditions of
view to
generating
ed in some
luntary

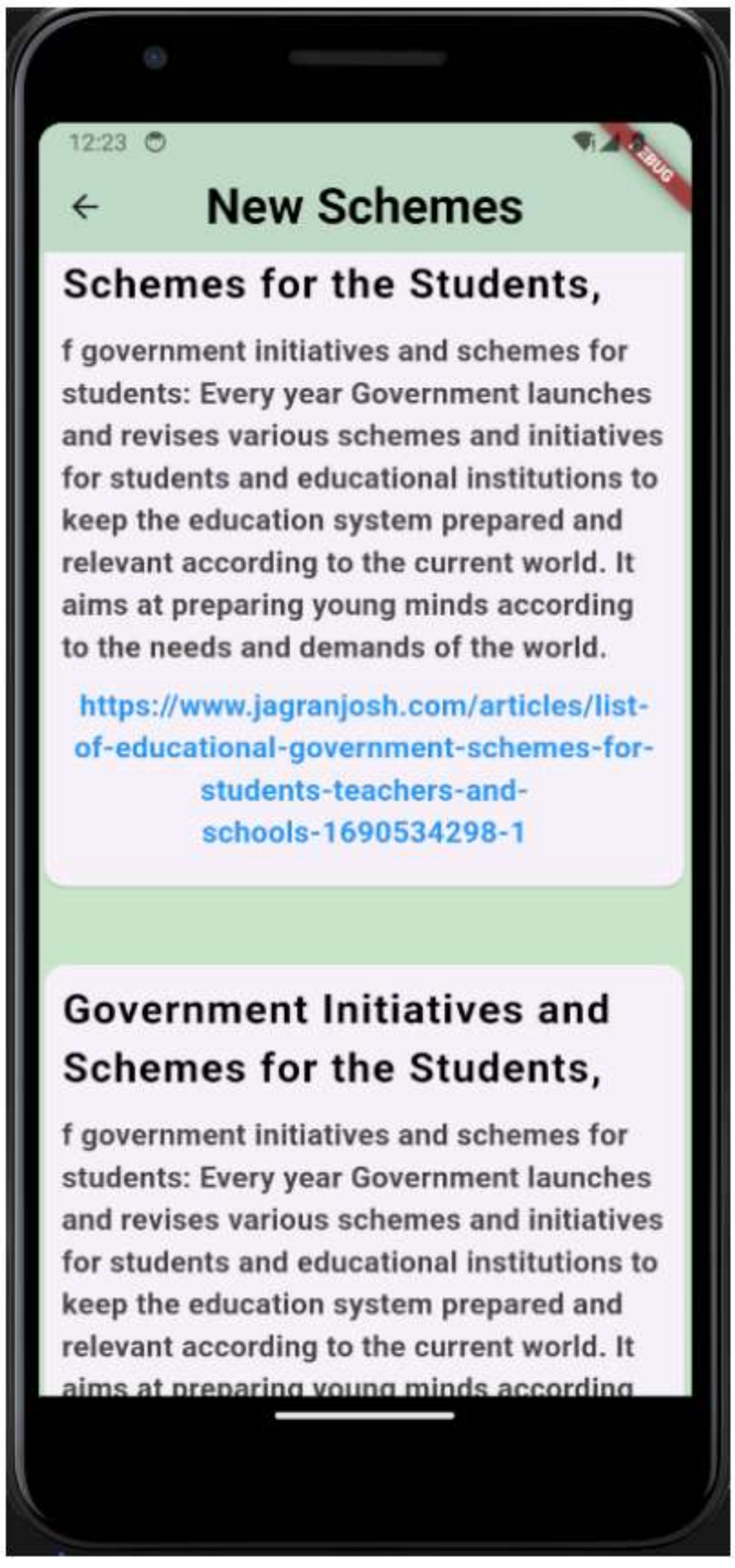
C.4 CREATE PROFILE



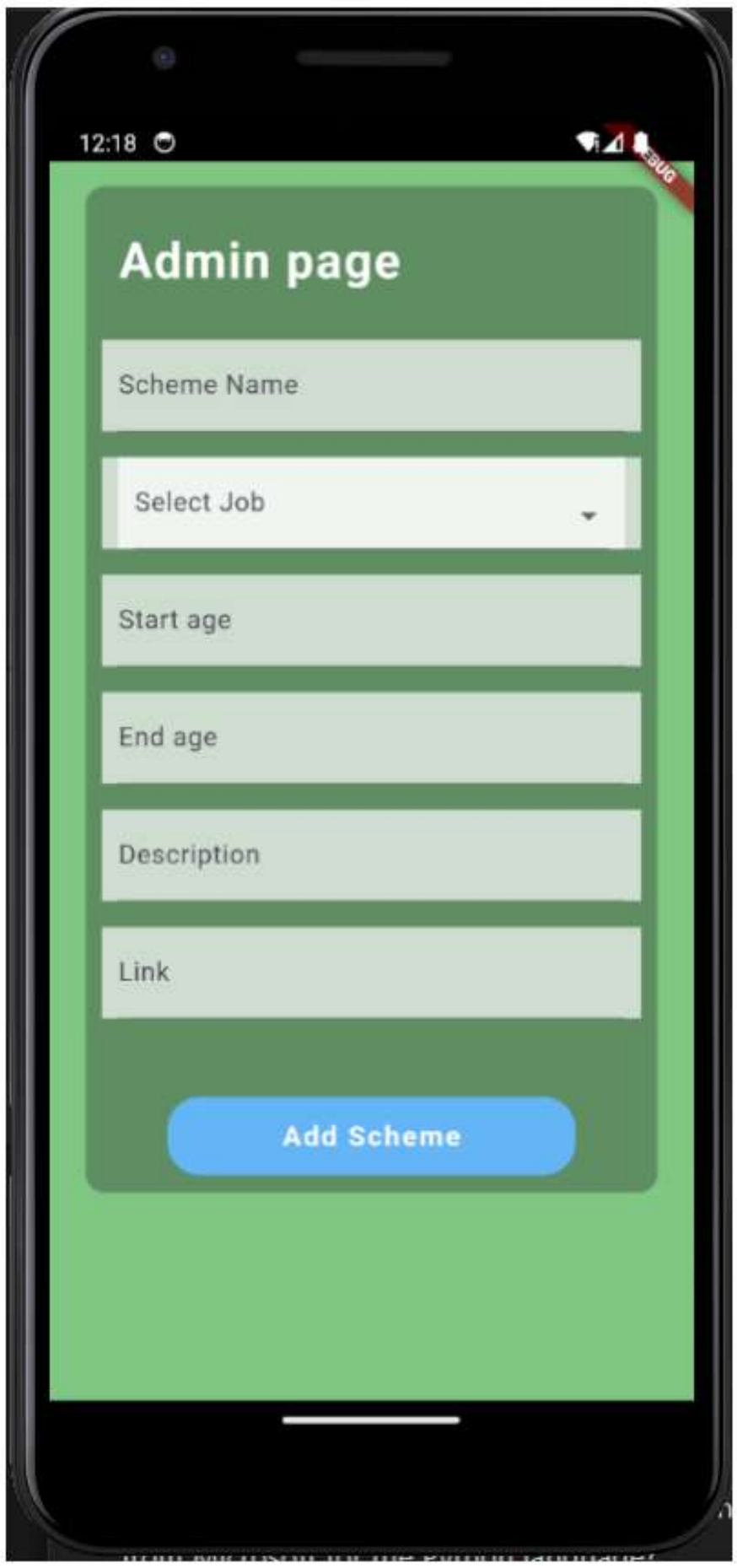
C.5 MY SCHEME


12:22
My Scheme
Ayushman Bharat Yojana
Ayushman Bharat Pradhan Mantri Jan Arogya Yojana is a national public health insurance scheme of the Government of India that aims to provide free access to health insurance coverage for low income earners in the country. Roughly, the bottom 50% of the country qualifies for this scheme.
https://abdm.gov.in/
Atal Pension Yojana
Atal Pension Yojana was passed in the Union Budget of 2015-16 by the then Honorable Finance Minister, Mr Arun Jaitley. This scheme is a pension scheme which aims at providing pension to the unorganised sector of the society such as the maids, gardeners, delivery boys, etc. Even the Private sector employees or employees who work in an organisation which does not provide pension benefits to them can also apply for this scheme.
https://financialservices.gov.in/pension-reforms-divisions/Atal-Pension-Yojana
Scheme For Residential Education For Students in High Schools in Targeted Areas
The main objective behind the scheme is to involve the voluntary sector and training institutions of repute to improve educational and socioeconomic conditions of the target group i.e. Scheduled Castes with a view to upgrade skill to enable them to start income generating activities on their own or get gainfully employed in some sector or the other. The principle that good voluntary

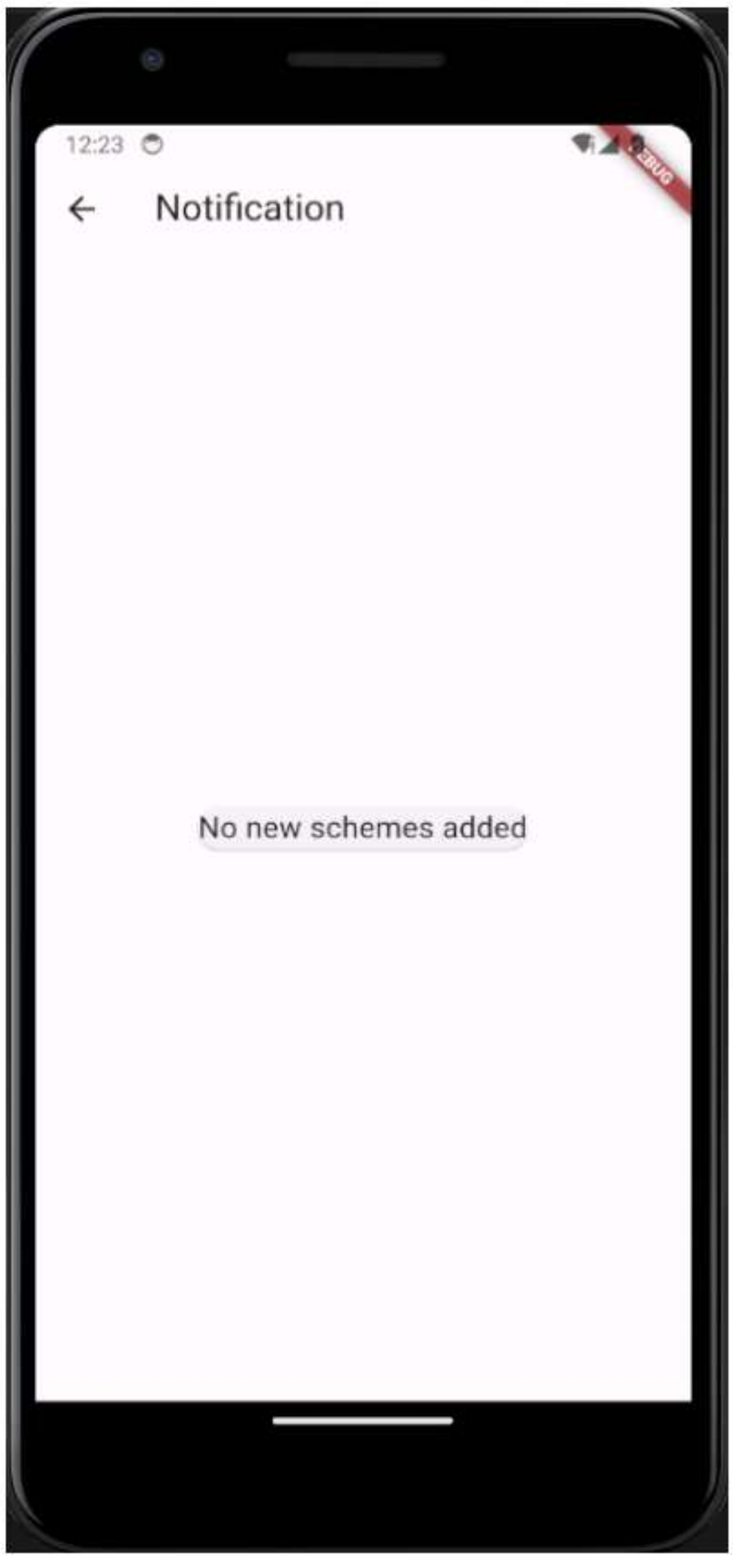
C.6 NEW SCHEME



C.7 ADMIN PAGE



C.8 NOTIFICATION



D CODE

admin.py

```
from django.contrib import admin
from .models import *
# Register your models here.
admin.site.register(CustomUser)
admin.site.register(ProfileDB)
admin.site.register(State)
```

apps.py

```
from django.apps import AppConfig
class AdminuiConfig(AppConfig):
    default_auto_field = 'django.db.models.BigAutoField'
    name = 'AdminUI'
```

models.py

```
from django.db import models
from django.contrib.auth.models import User,AbstractUser

class CustomUser(AbstractUser):
    GENDER_CHOICES = [
        ('Male', 'Male'),
        ('Female', 'Female'),
        ('Others', 'Others')
    ]
    gender = models.CharField(max_length=100, choices=
        GENDER_CHOICES, null=True)

    PROFESSION_CHOICES = [
        ('Farmer', 'Farmer'),
        ('Student', 'Student'),
        ('Disabled', 'Disabled'),
        ('Women', 'Women'),
        ('Teacher', 'Teacher'),
        ('Sports', 'Sports'),
        ('Military', 'Military'),
        ('Government Employee', 'Government Employee'),

    ]
    profession = models.CharField(max_length=100, choices
        =PROFESSION_CHOICES, default='Farmer')
    age = models.IntegerField(null=True)

class State(models.Model):
```

```
    name=models.CharField(max_length=100)

    class Meta:
        ordering =('name',)
        verbose_name_plural='States'

    def __str__(self):
            return self.name

# Create your models here.
class ProfileDB(models.Model):
    user = models.OneToOneField(CustomUser, on_delete=
        models.CASCADE, unique=True)
    Address = models.CharField(max_length=1000, null=True
        , blank=True)
    State = models.CharField(max_length=100,null=True)
    Pincode = models.IntegerField(null=True, blank=True)
    Aadhar_Number = models.BigIntegerField(null=True,
        blank=True)
    Contact = models.BigIntegerField(null=True, blank=
        True)
    options=(
        ('Farmer', 'Farmer'),
        ('Student', 'Student'),
        ('Disabled', 'Disabled'),
        ('Women', 'Women'),
        ('Teacher', 'Teacher'),
        ('Sports', 'Sports'),
        ('Military', 'Military'),
        ('Government Employee', 'Government Employee'),

    )
    profession = models.CharField(max_length=100,choices=
        options,default='Farmer')
    options=(
        ('APL','APL'),
        ('BPL','BPL'),
        ('AAY','AAY')
    )
    card=models.CharField(max_length=100,choices=options,
        null=True)
    options=(
        ('SSLC','SSLC'),
        ('PLUS TWO','PLUS TWO'),
        ('UG','UG'),
        ('PG','PG')
```

```
)
qual=models.CharField(max_length=100,choices=options,
    null=True)
options=(
    ('Mobility Impairment','Mobility Impairment'), #
        any full
    ('Visual Impairment','Visual Impairment'),  #
        audio
    ('Hearing Impairment','Hearing Impairment'), #
        image
    ('Learning Disability','Learning Disability'), #
        any
    ('Autism Spectrum Disorder','Autism Spectrum
        Disorder'), #any
    ('Speech Impairment','Speech Impairment'), #any
    ('Intellectual Disability','Intellectual
        Disability'), #any
    ('None','None')
)
disable=models.CharField(max_length=100,choices=
    options,null=True)
# Profile_Image = models.ImageField(upload_to='users/
    profile_images', null=True, blank=True)
```

serializers.py

```
from rest_framework import serializers
from rest_framework.validators import UniqueValidator
from .models import *
from rest_framework_simplejwt.serializers import
    TokenObtainPairSerializer
import json
# from AdminUI.models import ProfileDB



class UserSerializers(serializers.ModelSerializer):
    email = serializers.EmailField(
        validators=[UniqueValidator(queryset=CustomUser.
            objects.all())]
    )
    password=serializers.CharField(write_only=True)
    class Meta:
    model = CustomUser
    fields = (
        'username',
        'email',
```

```
            'profession',
            'gender',
            'age',
            'password'
        )
        extra_kwargs = {
            'password': {'write_only': True}  # Ensure
                password isn't included in responses
        }

    def create(self, validated_data):
        return CustomUser.objects.create_user(**
            validated_data)



class ProfileSerializer(serializers.ModelSerializer):
    class Meta:
        model = ProfileDB
        fields = (
            'user',
            'Address',
            'State',
            'Pincode',
            'profession',
            'card',
            'qual',
            'disable',
            'Aadhar_Number',
            'Contact',

        )


class LoginSerializer(serializers.Serializer):
    username = serializers.CharField()
    password = serializers.CharField(write_only=True)


class PasswordChangeSerializer(serializers.Serializer):
    old_password = serializers.CharField(required=True)
    new_password = serializers.CharField(required=True)

from rest_framework_simplejwt.serializers import
    TokenObtainPairSerializer
```

```
class MyTokenObtainPairSerializer(
    TokenObtainPairSerializer):
     @classmethod
     def get_token(cls, user):
          token = super().get_token(user)

          # Add custom claims to the token
          token['user_id'] = user.id
          token['username'] = user.username
          token['email'] = user.email
          token['is_superuser'] = user.is_superuser
          token['age'] = user.age
          token['gender'] = user.gender
          token['profession'] = user.profession
          # Add more custom claims as needed

          return token

     def validate(self, attrs):
          data = super().validate(attrs)

          # Include additional user details in the response
          data['user_id'] = self.user.id
          data['username'] = self.user.username
          data['email'] = self.user.email
          data['is_superuser'] = self.user.is_superuser
          data['age'] = self.user.age
          data['gender'] = self.user.gender
          data['profession'] = self.user.profession
          # Add more user details as needed

          return data
```

tests.py

```
from django.test import TestCase

# Create your tests here.
```

urls.py

```
from django.urls import path
from AdminUI import views
from .views import  *
from django.conf import settings
from django.conf.urls.static import static
```

```
urlpatterns = [
    path('register/',views.register_request.as_view(),
        name="register"), # registration
    path('profile/',ProfileView.as_view(),name='pro'),#
        profikle get put
    path('changepassword/',ChangePasswordView.as_view(),
        name="changeps"),#changepassword url
    path('logins/',LoginView.as_view(),name='log')

] + static(settings.MEDIA_URL, document_root=settings.
    MEDIA_ROOT)
```

views.py

```
from rest_framework import permissions
from django.shortcuts import render,redirect
from rest_framework import status,mixins,generics
from rest_framework.response import Response
from rest_framework.views import APIView
from rest_framework.viewsets import ModelViewSet
from rest_framework_simplejwt import authentication
from rest_framework.renderers import TemplateHTMLRenderer
    ,JSONRenderer
from django.contrib.auth import authenticate
from rest_framework.authtoken.models import Token
from rest_framework_simplejwt.views import
    TokenObtainPairView
from django.contrib.auth import authenticate, login
from Scheme.models import *
from Scheme.serializers import *
from django.contrib.auth import update_session_auth_hash
from rest_framework.renderers import MultiPartRenderer
from rest_framework.parsers import MultiPartParser
from .models import *
from .serializers import *
from django.contrib.auth.hashers import check_password

from rest_framework_simplejwt.views import
    TokenObtainPairView
from rest_framework.response import Response


class MyTokenObtainPairView(TokenObtainPairView):
    serializer_class = MyTokenObtainPairSerializer

    def post(self, request, *args, **kwargs):
```

```
        response = super().post(request, *args, **kwargs)
        serializer = MyTokenObtainPairSerializer(data=
            request.data)
        serializer.is_valid(raise_exception=True)

        # Include serialized user data in the response
        response.data.update(serializer.validated_data)
        return response


class register_request(mixins.ListModelMixin,mixins.
    CreateModelMixin,generics.GenericAPIView):
    queryset=CustomUser.objects.all()
    serializer_class=UserSerializers

    # def get(self,request):
    #     return self.list(request)

    def post(self,request,*args,**kwargs):
        return self.create(request)


class LoginView(APIView):
    serializer_class = LoginSerializer

    def post(self, request):
        serializer = self.serializer_class(data=request.
            data)

        if serializer.is_valid():
            username = serializer.validated_data['
                username']
            password = serializer.validated_data['
                password']
            user = authenticate(request, username=
                username, password=password)
            user = CustomUser.objects.get(username=
                username)

            if user is not None and check_password(
                password, user.password):
                login(request,user)
                return Response({'message': 'Login
                    successful'}, status=status.
                    HTTP_200_OK)
```

```
            else:
                return Response({'message': 'Invalid
                    credentials'}, status=status.
                    HTTP_401_UNAUTHORIZED)

        return Response(serializer.errors, status=status.
            HTTP_400_BAD_REQUEST)
```

```
class ProfileView(APIView):
    # permission_classes = [permissions.IsAuthenticated]
    # authentication_classes = [authentication.
        JWTAuthentication]
    serializer_class=ProfileSerializer


    def get(self,req,*args,**kwargs):
        prr=req.user.id
        print(prr)
        try:

            pr=ProfileDB.objects.get(user=req.user.id)
            print(pr)
            dser=ProfileSerializer(pr)
            return Response(data=dser.data)
        except:
            return Response({"msg":"Profile is not
                created"},status=status.
                HTTP_400_BAD_REQUEST)
    def post(self,req,*args,**kwargs):
        user=req.user.id
        ser=ProfileSerializer(data=req.POST)
        if ser.is_valid():
            ser.save()
            return Response({"Msg":"Profile Added"},
                status=status.HTTP_201_CREATED)
        else:
            return Response({"Msg": ser.errors},
                status=status.HTTP_400_BAD_REQUEST)

    def put(self,req,*args,**kwargs):
        try:
            pr=ProfileDB.objects.get(user=req.user.id)
            print(req.user.id)
            ser=ProfileSerializer(data=req.data,instance=
                pr)
            if ser.is_valid():
```

```
            ser.save()
            return Response({"msg":"Updated"})
        else:
            return Response({"msg":ser.errors},status
                =status.HTTP_422_UNPROCESSABLE_ENTITY)
    except:
        return Response({"msg":"Invalid ID"},status=
            status.HTTP_400_BAD_REQUEST)


class ChangePasswordView(APIView):
    # permission_classes = [permissions.IsAuthenticated]

    def post(self, request):
        user = request.user
        serializer = PasswordChangeSerializer(data=
            request.data)

        if serializer.is_valid():
            old_password = serializer.data.get("
                old_password")
            new_password = serializer.data.get("
                new_password")

            if not user.check_password(old_password):
                return Response({"old_password": ["Wrong
                    password."]}, status=status.
                    HTTP_400_BAD_REQUEST)

            user.set_password(new_password)
            user.save()
            update_session_auth_hash(request, user)
            return Response({"message": "Password updated
                successfully."}, status=status.
                HTTP_200_OK)

        return Response(serializer.errors, status=status.
            HTTP_400_BAD_REQUEST)
```

asgi.py

```
"""
ASGI config for GetScheme project.

It exposes the ASGI callable as a module-level variable
    named ``application``.
```

```
For more information on this file, see
https://docs.djangoproject.com/en/5.0/howto/deployment/
    asgi/
"""

import os

from django.core.asgi import get_asgi_application

os.environ.setdefault('DJANGO_SETTINGS_MODULE', '
    GetScheme.settings')

application = get_asgi_application()
```

settings.py

```
"""
Django settings for GetScheme project.

Generated by 'django-admin startproject' using Django
    4.1.4.

For more information on this file, see
https://docs.djangoproject.com/en/4.1/topics/settings/

For the full list of settings and their values, see
https://docs.djangoproject.com/en/4.1/ref/settings/
"""

from pathlib import Path
import os.path
import crispy_forms

# Build paths inside the project like this: BASE_DIR / '
    subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent


# Quick-start development settings - unsuitable for
    production
# See https://docs.djangoproject.com/en/4.1/howto/
    deployment/checklist/

# SECURITY WARNING: keep the secret key used in
    production secret!
SECRET_KEY = 'django-insecure-!w0yrs!6_bkr!qcez740ze&q*rz
    &_syfidoc^@iid3url9*6+o'
```

```
# SECURITY WARNING: don't run with debug turned on in
    production!
DEBUG = True

ALLOWED_HOSTS = ['10.0.2.2']
# ALLOWED_HOSTS = []


# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'AdminUI',
    'Scheme',
    'crispy_forms',
    'rest_framework',
    # 'rest_framework.authtoken',
    'rest_framework_simplejwt',
    'corsheaders'
]

# CRISPY_TEMPLATE_PACK = 'bootstrap4'

MIDDLEWARE = [
    'corsheaders.middleware.CorsMiddleware',
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware
        ',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.
       AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware
        ',
    'django.middleware.clickjacking.
        XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'GetScheme.urls'
```

```
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.
            DjangoTemplates',
        'DIRS': [os.path.join(BASE_DIR, 'templates')],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug
                    ',
                'django.template.context_processors.
                    request',
                'django.contrib.auth.context_processors.
                    auth',
                'django.contrib.messages.
                    context_processors.messages',
            ],
        },
    },
]

WSGI_APPLICATION = 'GetScheme.wsgi.application'


# Database
# https://docs.djangoproject.com/en/4.1/ref/settings/#
    databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'GETSCHEME',
        'USER': 'root',
        'PASSWORD': 'root',
        'HOST':'localhost',
        'PORT':'3306',
    }
}


# Password validation
# https://docs.djangoproject.com/en/4.1/ref/settings/#
    auth-password-validators
```

```
AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.
            UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.
            MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.
            CommonPasswordValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.
            NumericPasswordValidator',
    },
]


# Internationalization
# https://docs.djangoproject.com/en/4.1/topics/i18n/

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'GMT'

USE_I18N = True

USE_TZ = True


# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/4.1/howto/static-
    files/

STATIC_URL = 'static/'
MEDIA_URL = '/media/'
MEDIA_ROOT = os.path.join(BASE_DIR,'media/')

# Default primary key field type
# https://docs.djangoproject.com/en/4.1/ref/settings/#
    default-auto-field

DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
```

```
REST_FRAMEWORK = {
    'DEFAULT_AUTHENTICATION_CLASSES': [
        'rest_framework_simplejwt.authentication.
            JWTAuthentication',
    ],
}

from datetime import timedelta


SIMPLE_JWT = {
    'ACCESS_TOKEN_LIFETIME': timedelta(hours=10),
    'SLIDING_TOKEN_REFRESH_LIFETIME': timedelta(days=1),
    'SLIDING_TOKEN_LIFETIME': timedelta(days=1),
    'SLIDING_TOKEN_REFRESH_LIFETIME': timedelta(days=30),
}

AUTH_USER_MODEL = 'AdminUI.CustomUser'

CORS_ALLOW_METHODS = (
    "DELETE",
    "GET",
    "OPTIONS",
    "PATCH",
    "POST",
    "PUT",
)

EMAIL_BACKEND = 'django.core.mail.backends.smtp.
    EmailBackend'
EMAIL_HOST = 'smtp.gmail.com'  # Replace with your SMTP
    host
EMAIL_PORT = 587  # Replace with your SMTP port
EMAIL_USE_TLS = True  # Use TLS for secure connection
EMAIL_HOST_USER = 'hibindixon123123@gmail.com'  # Replace
     with your email
EMAIL_HOST_PASSWORD = 'altujgkcglvsttop'  # Replace with
    your email password
```

urls.py

```
"""GetScheme URL Configuration

The `urlpatterns` list routes URLs to views. For more
```

```
    information please see:
     https://docs.djangoproject.com/en/4.1/topics/http/
        urls/
Examples:
Function views
    1. Add an import:  from my_app import views
    2. Add a URL to urlpatterns:  path('', views.home,
        name='home')
Class-based views
    1. Add an import:  from other_app.views import Home
    2. Add a URL to urlpatterns:  path('', Home.as_view()
        , name='home')
Including another URLconf
    1. Import the include() function: from django.urls
        import include, path
    2. Add a URL to urlpatterns:  path('blog/', include('
        blog.urls'))
"""
from django.contrib import admin
from django.contrib.staticfiles.urls import
    staticfiles_urlpatterns, static
from django.urls import path, include
from rest_framework_simplejwt.views import (
    TokenObtainPairView,
    TokenRefreshView,
)

import AdminUI.urls
import Scheme.urls
from . import settings
from  rest_framework.authtoken import views
from AdminUI.views import *
from rest_framework.decorators import api_view,
    permission_classes
from rest_framework.permissions import AllowAny
from rest_framework_simplejwt.tokens import RefreshToken

@api_view(['GET'])
@permission_classes([AllowAny])
def get_tokens_for_user(request):

    # find the user base in params
    user = CustomUser.objects.first()

    refresh = RefreshToken.for_user(user)
```

```
    return Response({
        'refresh': str(refresh),
        'access': str(refresh.access_token),
    })

urlpatterns = [
    path('admin/', admin.site.urls),
    path('AdminUI/', include("AdminUI.urls")),
    path('Scheme/', include(Scheme.urls)),
    path('login/',MyTokenObtainPairView.as_view(),name='
        login'),
    path('token/', TokenObtainPairView.as_view(), name='
        token_obtain_pair'),
    path('token/refresh/', TokenRefreshView.as_view(),
        name='token_refresh'),
]
urlpatterns += staticfiles_urlpatterns()
urlpatterns += static(settings.MEDIA_URL, document_root=
    settings.MEDIA_ROOT)
```

wsgi.py

```
"""
WSGI config for GetScheme project.

It exposes the WSGI callable as a module-level variable
    named ``application``.

For more information on this file, see
https://docs.djangoproject.com/en/5.0/howto/deployment/
    wsgi/
"""

import os

from django.core.wsgi import get_wsgi_application

os.environ.setdefault('DJANGO_SETTINGS_MODULE', '
    GetScheme.settings')

application = get_wsgi_application()
```

admin.py

```
from django.contrib import admin

from Scheme.models import SchemesDB
```

```
# Register your models here.
admin.site.register(SchemesDB)
```

apps.py

```
from django.apps import AppConfig


class SchemeConfig(AppConfig):
    default_auto_field = 'django.db.models.BigAutoField'
    name = 'Scheme'
```

models.py

```
  from django.db import models
from django.utils import timezone
# Create your models here.

class SchemesDB(models.Model):
    Scheme_Name = models.CharField(max_length=100,null=
        True)
    options=(
        ('Disabled','Disabled'),
        ('Farmer','Farmer'),
        ('Age','Age'),
        ('Student','Student'),
        ('Government Employ','Government Employ'),
        ('General','General'),
        ('Women','Women'),
        ('Teacher','Teacher'),
        ('Sports','Sports'),
        ('Military','Military')
)
type = models.CharField(max_length=100,null=True,
    choices=options)
start_age=models.IntegerField(null=True)
end_age=models.IntegerField(null=True)
Description = models.CharField(max_length=1000,null=
    True)
Link = models.CharField(max_length=1000,null=True)
timestamp=models.DateTimeField(auto_now_add=True,null
    =True)


def contains_age(self, age_to_check):
    return self.start_age <= age_to_check <= self.
        end_age
```

serializers.py

```
from django.contrib.auth.models import User
from rest_framework import serializers

from Scheme.models import SchemesDB


class SchemeSerializer(serializers.ModelSerializer):
    class Meta:
        model = SchemesDB
        fields = (
            'Scheme_Name',
            'type',
            'start_age',
            'end_age',
            'Description',
            'Link'
        )
```

tests.py

```
from django.test import TestCase

# Create your tests here.
```

urls.py

```
from django.urls import path
from .views import *


urlpatterns = [
    path('scheme/',SchemeView.as_view(),name='scheme'),
    path('new_elements/',NewElementsAPIView.as_view(),
        name='elements'),
    path('myscheme/',FilterAPIView.as_view(),name='my'),
    path('notification/',NotificationsAPIView.as_view(),
        name='ntv')

]
```

views.py

```
from django.shortcuts import render
from rest_framework.views import APIView
from rest_framework_simplejwt import authentication
from rest_framework import permissions
from .models import *
from .serializers import *
from rest_framework.response import Response
```

```
from rest_framework import status
from django.utils import timezone
from AdminUI.models import ProfileDB
from django.http import JsonResponse
from AdminUI.models import CustomUser
from django.core.mail import send_mail
from django.conf import settings
# Create your views here.

class SchemeView(APIView):
    # permission_classes = [permissions.IsAuthenticated]
    # authentication_classes = [authentication.
      JWTAuthentication]

    def get(self,req,*args,**kwargs):
        try:
            pr=SchemesDB.objects.all()
            dser=SchemeSerializer(pr,many=True)
            return Response(data=dser.data)
        except:
            return Response({"msg":"Invalid ID"},status=
                status.HTTP_400_BAD_REQUEST)
    def post(self,req):
            ser=SchemeSerializer(data=req.POST)
            if ser.is_valid():
scheme_instance = ser.save()
users = CustomUser.objects.all()  #
    Replace with your actual User model
subject = 'New Scheme Created'
message = f'A new scheme has been created
    : '  # Adjust based on your Scheme
    model fields
from_email = 'hibindixon123123@gmail.com'
for user in users:
    to_email = [user.email]
    send_mail(subject, message,
        from_email, to_email,
        fail_silently=False)


return Response({"Msg":"Scheme Added"},
    status=status.HTTP_201_CREATED)
            else:
                return Response({"Msg": ser.errors},
                    status=status.HTTP_400_BAD_REQUEST)
```

```
import datetime

class NewElementsAPIView(APIView):
    def get(self, request):
        since_date = datetime.datetime.now() - datetime.
            timedelta(days=10)
        # current_time = timezone.now()
        if since_date:
            new_elements = SchemesDB.objects.filter(
                timestamp__gt=since_date)
        else:
            new_elements = SchemesDB.objects.all()

        serialized_data = SchemeSerializer(new_elements,
            many=True)
        return Response(serialized_data.data)


class NotificationsAPIView(APIView):
    def get(self, request):
        since_date = datetime.datetime.now() - datetime.
            timedelta(hours=5,minutes=41)
        # current_time = timezone.now()
        new_elements = SchemesDB.objects.filter(
            timestamp__gt=since_date).order_by("-timestamp
            ")
        if new_elements:
            msg="New schemes added check it..."
            response={"message":msg}
            return JsonResponse(response)
        else :
            msg="No new schemes added"
            response={"message":msg}
            return JsonResponse(response)


        # serialized_data = SchemeSerializer(new_elements
            , many=True)
        # return Response(data=serialized_data.data)


from django.shortcuts import get_object_or_404
class FilterAPIView(APIView):
    def get(self, request):

        id=request.user.id
```

```
pr = get_object_or_404(ProfileDB, user=id)
print(pr.id)
user_profession = pr.profession if hasattr(pr, '
    profession') else None
user_age = request.user.age if hasattr(request.
    user, 'age') else None
print(user_profession)
print(request.user.profession)
schemes = SchemesDB.objects.filter(type=
    user_profession,start_age__lte=user_age,
    end_age__gte=user_age)
filtered_schemes = [scheme for scheme in schemes
    if scheme.contains_age(user_age)]
serialized_data = SchemeSerializer(
    filtered_schemes, many=True)
return Response(serialized_data.data)
```

manage.py

```
#!/usr/bin/env python
"""Django's command-line utility for administrative tasks
    ."""
import os
import sys


def main():
    """Run administrative tasks."""
    os.environ.setdefault('DJANGO_SETTINGS_MODULE', '
        GetScheme.settings')
    try:
        from django.core.management import
            execute_from_command_line
    except ImportError as exc:
        raise ImportError(
            "Couldn't import Django. Are you sure it's
                installed and "
            "available on your PYTHONPATH environment
                variable? Did you "
            "forget to activate a virtual environment?"
        ) from exc
    execute_from_command_line(sys.argv)


if __name__ == '__main__':
    main()
```

control.dart

```
import 'package:get/get.dart';
import 'package:schemeapp/service/service.dart';

class SchemeController extends GetxController {
  var list = [].obs;

  @override
  void onInit() {
    getScheme();
    super.onInit();
  }

  void getScheme() async {
    try {
      var data = await HttpScheme.fetchscheme();
      if (data != null) {
        list.value = data;
      }
    } catch (e) {
      print(e);
    }
  }
}
```

filter_scheme_controller.dart

```
import 'package:get/get.dart';
import 'package:schemeapp/service/filter_scheme_serrvice.
    dart';
import 'package:schemeapp/service/service.dart';

class FilterSchemeController extends GetxController {
  var flist = [].obs;

  @override
  void onInit() {
    getfilterScheme();
    super.onInit();
  }

  void getfilterScheme() async {
    try {
      var data = await HttpFilterScheme.filterscheme();
      if (data != null) {
        flist.value = data;
      }
    } catch (e) {
```

```
        print(e);
      }
    }
}
```

filterscheme.dart

```
// To parse this JSON data, do
//
//     final filterScheme = filterSchemeFromJson(
    jsonString);

import 'dart:convert';

List<FilterScheme> filterSchemeFromJson(String str) =>
    List<FilterScheme>.from(json.decode(str).map((x) =>
    FilterScheme.fromJson(x)));

String filterSchemeToJson(List<FilterScheme> data) =>
    json.encode(List<dynamic>.from(data.map((x) => x.
    toJson())));

class FilterScheme {
    String? schemeName;
    String? type;
    int? startAge;
    int? endAge;
    String? description;
    String? link;

    FilterScheme({
        this.schemeName,
        this.type,
        this.startAge,
        this.endAge,
        this.description,
        this.link,
    });

    factory FilterScheme.fromJson(Map<String, dynamic>
    json) => FilterScheme(
        schemeName: json["Scheme_Name"],
        type: json["type"],
        startAge: json["start_age"],
        endAge: json["end_age"],
        description: json["Description"],
        link: json["Link"],
```

```
);

Map<String, dynamic> toJson() => {
        "Scheme_Name": schemeName,
        "type": type,
        "start_age": startAge,
        "end_age": endAge,
        "Description": description,
        "Link": link,
    };
}
```

modell.dart

```
// To parse this JSON data, do
//
//     final schmeModel = schmeModelFromJson(jsonString);

import 'dart:convert';

List<SchmeModel> schmeModelFromJson(String str) => List<
    SchmeModel>.from(json.decode(str).map((x) =>
    SchmeModel.fromJson(x)));

String schmeModelToJson(List<SchmeModel> data) => json.
    encode(List<dynamic>.from(data.map((x) => x.toJson())
    ));

class SchmeModel {
    String? schemeName;
    String? schmeModelNew;
    String? description;
    String? link;

    SchmeModel({
        this.schemeName,
        this.schmeModelNew,
        this.description,
        this.link,
    });

    factory SchmeModel.fromJson(Map<String, dynamic> json
        ) => SchmeModel(
    schemeName: json["Scheme_Name"],
    schmeModelNew: json["new"],
    description: json["Description"],
    link: json["Link"],
```

```
);

Map<String, dynamic> toJson() => {
      "Scheme_Name": schemeName,
      "new": schmeModelNew,
      "Description": description,
      "Link": link,
    };
}
```

adminl.dart

```
import 'package:flutter/material.dart';
import 'package:url_launcher/url_launcher.dart';

void main() {
  runApp(MaterialApp(home: Admin()));
}

class Admin extends StatefulWidget {
  @override
  State<Admin> createState() => _AdminState();
}

class _AdminState extends State<Admin> {
  TextEditingController linkController =
    TextEditingController();

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text('Upload Page'),
    ),
    body: Padding(
      padding: const EdgeInsets.all(16.0),
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          TextField(
            controller: linkController,
            decoration: InputDecoration(labelText: '
                Enter Link'),
    ),
    SizedBox(height: 20),
    ElevatedButton(
      onPressed: () {
```

```
                String uploadedLink = linkController.text
                    ;
                _launchURL(uploadedLink);
              },
              child: Text('Upload Link'),
            ),
          ],
            ),
          ),
        );
    }

    _launchURL(String link) async {
      if (await canLaunch(link)) {
        await launch(link);
      } else {
        throw 'Could not launch $link';
      }
    }
}
```

edit_profile.dart

```
import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;

void main() {
  runApp(MyHomePage());
}

class MyHomePage extends StatefulWidget {
  @override
  _MyHomePageState createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {
  TextEditingController usernameController =
      TextEditingController();
  // TextEditingController emailController =
      TextEditingController();
  TextEditingController stateController =
      TextEditingController();
  TextEditingController pinController =
      TextEditingController();
  TextEditingController addressController =
      TextEditingController();
  TextEditingController contactController =
```

```
    TextEditingController();
TextEditingController aadharController =
    TextEditingController();

Future<void> postData() async {
  final url = 'http://10.0.2.2:8000/AdminUI/profile/';

final response = await http.post(
  Uri.parse(url),
  body: {
    'user': usernameController.text,
    'Address': addressController.text,
    'State': stateController.text,
    'Pincode': pinController.text,
    'Aadhar_Number': addressController.text,
    'Contact': contactController.text,
    'profession': aadharController.text,
    'card': aadharController.text,
    'qual': aadharController.text,
    'disable': aadharController.text,
  },
);

if (response.statusCode == 201) {
  print('Data posted successfully');
  } else {
    print('Failed to post data. Status code: ${response
        .statusCode}');
  }
}

String profesion = 'Farmer';
String card = 'APL';
String qualification = 'SSLC';
String disable = "None";

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text('HTTP Post Example'),
    ),
body:
      // Padding(
      //   padding: const EdgeInsets.all(16.0),
      //   child: Column(
```

```
//      mainAxisAlignment: MainAxisAlignment.
    center,
//      children: [
//        TextField(
//          controller: usernameController,
//          decoration: InputDecoration(
    labelText: 'Username'),
//        ),
//        TextField(
//          controller: emailController,
//          decoration: InputDecoration(
    labelText: 'Email'),
//        ),
//        TextField(
//          controller: stateController,
//          decoration: InputDecoration(
    labelText: 'State'),
//        ),
//        TextField(
//          controller: pinController,
//          decoration: InputDecoration(
    labelText: 'Pin'),
//        ),
//        TextField(
//          controller: addressController,
//          decoration: InputDecoration(
    labelText: 'Address'),
//        ),
//        TextField(
//          controller: contactController,
//          decoration: InputDecoration(
    labelText: 'Contact'),
//        ),
//        TextField(
//          controller: aadharController,
//          decoration: InputDecoration(
    labelText: 'Aadhar'),
//        ),
//        SizedBox(height: 20),
//        ElevatedButton(
//          onPressed: postData,
//          child: Text('Post Data'),
//        ),
//      ],
//    ),
// ),
```

```
    SingleChildScrollView(
  child: Padding(
    padding: const EdgeInsets.all(16.0),
    child: Column(
      children: [
        // username
  TextFormField(
    controller: usernameController,
    decoration: InputDecoration(labelText: '
        Username'),
  ),

  // Address
  TextFormField(
    controller: addressController,
    decoration: InputDecoration(labelText: '
        Address'),
  ),

  // state
  TextFormField(
    controller: stateController,
    decoration: InputDecoration(labelText: '
        stste'),
  ),

  // Pincod
  TextFormField(
    controller: pinController,
    decoration: InputDecoration(labelText: '
        Pincode'),
  ),

  // Adhar number
  TextFormField(
    controller: aadharController,
    decoration: InputDecoration(labelText: '
        Adhar number'),
  ),

  // Contact
  TextFormField(
    controller: contactController,
    decoration: InputDecoration(labelText: '
        Contact'),
```

```
),

SizedBox(height: 20.0),

// proffesion
DropdownButtonFormField<String>(
  value: profesion,
  onChanged: (String? newValue) {
    setState(() {
      profesion = newValue!;
    });
  },
  items: <String>[
    'Farmer',
    'Student',
    'Disability',
    'Government Employ'
  ].map<DropdownMenuItem<String>>((String
      value) {
    return DropdownMenuItem<String>(
      value: value,
      child: Text(value),
    );
  }).toList(),
  decoration: InputDecoration(labelText: '
      Profession'),
),
SizedBox(height: 20.0),

// card
DropdownButtonFormField<String>(
  value: card,
  onChanged: (String? newValue) {
    setState(() {
      card = newValue!;
    });
  },
  items: <String>[
    'APL',
    'BPL',
    'AAY',
  ].map<DropdownMenuItem<String>>((String
      value) {
    return DropdownMenuItem<String>(
      value: value,
      child: Text(value),
```

```
        );
      }).toList(),
      decoration: InputDecoration(labelText: '
          Card'),
    ),
    SizedBox(height: 20.0),

    // Qualifivcation
    DropdownButtonFormField<String>(
      value: qualification,
      onChanged: (String? newValue) {
        setState(() {
          qualification = newValue!;
        });
      },
      items: <String>['SSLC', 'PLUS TWO', 'UG',
          'PG']
          .map<DropdownMenuItem<String>>((
              String value) {
        return DropdownMenuItem<String>(
          value: value,
          child: Text(value),
        );
      }).toList(),
      decoration: InputDecoration(labelText: '
          Qualification'),
    ),
    SizedBox(height: 20.0),

    // Disable
    DropdownButtonFormField<String>(
      value: disable,
      onChanged: (String? newValue) {
        setState(() {
          disable = newValue!;
        });
      },
      items: <String>[
        "Mobility Impairment",
        "Visual Impairment",
        "Hearing Impairment",
        "Learning Disability",
        "Autism Spectrum Disorder",
        "Speech Impairment",
        "Intellectual Disability",
        "None",
```

```
].map<DropdownMenuItem<String>>((String
    value) {
  return DropdownMenuItem<String>(
    value: value,
    child: Text(value),
  );
}).toList(),
                    decoration: InputDecoration(labelText: '
                        Disable'),
                  ),
                  SizedBox(height: 20.0),

                  ElevatedButton(
                    onPressed: () {
                      //
                    },
                    child: Text('Save'),
                  ),
                ],
              ),
            ),
          ),
        ),
      );
  }
}
```

home.dart

```
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'package:schemeapp/controller/control.dart';
import 'package:schemeapp/model/model.dart';
import 'package:schemeapp/screens/user/loginpage.dart';
import 'package:schemeapp/views/Mysche.dart';
import 'package:schemeapp/views/uu.dart';
import 'package:url_launcher/url_launcher.dart';

class SchemePage extends StatelessWidget {
  final SchemeController schemeController = Get.put(
      SchemeController());

  @override
  Widget build(BuildContext context) {
    return SafeArea(
      child: Scaffold(
        drawer: Drawer(
          child: ListView(
```

```
padding: EdgeInsets.zero,
children: <Widget>[
  SizedBox(
    height: 10,
  ),
  DrawerHeader(
    decoration: BoxDecoration(
  image: DecorationImage(
    image: NetworkImage(
      "https://qph.cf2.quoracdn.net/main-
          qimg-89
          dc4d0e21c42c7f0778582ee45c7440-
          pjlq",
    ),
    fit: BoxFit.fill,
  ),
),
child: Text(
  'User Name', // Replace with the actual
      user name
  style: TextStyle(
    color: Colors.white,
    fontSize: 20,
  ),
),
),
ListTile(
  leading: Icon(Icons.star),
  title: Text('My Profile', style:
      TextStyle(fontSize: 20)),
  onTap: () {
    Navigator.of(context).push(
      // MaterialPageRoute(builder: (
          context) => EditProfile()),
      MaterialPageRoute(builder: (context)
          => Profile()),
    );
  },
),
ListTile(
  leading: Icon(Icons.star),
  title: Text('Scheme', style: TextStyle(
    fontSize: 20)),
onTap: () {
  Navigator.of(context).push(
    MaterialPageRoute(builder: (context)
```

```
                => FilterSchemePage()),
          );
        },
      ),
      ListTile(
        leading: Icon(Icons.star),
        title: Text('All Scheme', style:
            TextStyle(fontSize: 20)),
        onTap: () {
          Navigator.of(context).push(
            MaterialPageRoute(builder: (context)
                => SchemePage()),
          );
        },
      ),
      ListTile(
        leading: Icon(Icons.exit_to_app),
        title: Text('Logout', style: TextStyle(
            fontSize: 20)),
        onTap: () {
          Navigator.of(context).push(
            MaterialPageRoute(builder: (context)
                => LoginPage()),
          );
        },
          ),
        ],
      ),
    ),
    backgroundColor: Colors.green[100],
    appBar: AppBar(
      backgroundColor: Colors.green[100],
      centerTitle: true,
      title: Text(
        "My Scheme",
        style: TextStyle(
          fontSize: 30.0,
          fontWeight: FontWeight.bold,
          color: Colors.black,
        ),
      ),
    ),
    body: Obx(
      () => ListView.builder(
        itemCount: schemeController.list.length,
        itemBuilder: (context, index) {
```

```
SchmeModel scheme = schemeController.list[
    index];

return Card(
  child: Container(
    padding: EdgeInsets.all(10),
    child: Column(
children: [
  Text(
    '${scheme.schemeName}',
    textAlign: TextAlign.center,
    style: TextStyle(
      fontSize: 18,
      fontWeight: FontWeight.bold,
      color: Colors.green,
    ),
  ),
  SizedBox(height: 5),
  Container(
    child: Text('${scheme.description
        }'),
  ),
  GestureDetector(
    onTap: () {
      launchURL(scheme.link);
    },
    child: Text(
      '${scheme.link}',
      style: TextStyle(
        color: Colors.blue,
        decoration: TextDecoration.
            underline,
      ),
    ),
  ),
),
SizedBox(height: 5),
// Container(
//   width: 80,
//   height: 35,
//   child: Card(
//     shape:
    RoundedRectangleBorder(
//       borderRadius: BorderRadius
    .circular(20),
//     ),
//     shadowColor: const Color.
```

```
        fromARGB(255, 246, 16, 0),
//        child: Center(
//            child: Text('${scheme.
    schmeModelNew ?? ''}')),
//    ),
// ),
// SizedBox(height: 5),
                    ],
                  ),
                ),
              ),
            ),
          ),
        ),
      );
}

void launchURL(String? link) async {
  if (await canLaunch(link!)) {
    await launch(link,
        forceSafariVC: false, forceWebView: false,
            enableJavaScript: true);
  } else {
    throw 'Could not launch $link';
  }
}
```

loginpage.dart

```
import 'dart:convert';
import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;
import 'package:schemeapp/screens/user/Siginpage.dart';
import 'package:schemeapp/screens/user/home.dart';
import 'package:schemeapp/screens/user/homepage.dart';
import 'package:schemeapp/service/filter_scheme_serrvice.
    dart';
import 'package:schemeapp/views/scheme.dart';

class LoginPage extends StatefulWidget {
  @override
  _LoginPageState createState() => _LoginPageState();
}

class _LoginPageState extends State<LoginPage> {
```

```
final _usernameController = TextEditingController();
final _passwordController = TextEditingController();

void login() async {
  final response = await http.post(
    Uri.parse('http://10.0.2.2:8000/login/'),
    body: {
      'username': _usernameController.text,
      'password': _passwordController.text,
    },
  );

  if (response.statusCode == 200) {
    var responseMap = jsonDecode(response.body);
    String accessToken = responseMap['access'];

    print("accesssssstoken is :$accessToken");
    print(response.body);

    Navigator.push(
      context,
      MaterialPageRoute(
        builder: (context) => SchemePage(),
      ),
    );
  } else {
    ScaffoldMessenger.of(context).showSnackBar(
      SnackBar(
        content: Text('Invalid credentials. Please try
            again.'),
      ),
    );
  }
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    body: Container(
      decoration: BoxDecoration(
        image: DecorationImage(
          image: NetworkImage(
"https://images.unsplash.com/photo
    -1613125700782-8394bec3e89d?q=80&w
    =1887&auto=format&fit=crop&ixlib=rb
    -4.0.3&ixid=
```

```
                    M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA
                    %3D%3D") ,
            fit: BoxFit.cover,
        ),
    ),
    child: Padding(
      padding: const EdgeInsets.all(16.0),
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          Text(
            'Login',
            style: TextStyle(
              fontSize: 24,
              fontWeight: FontWeight.bold,
              color: Colors.white,
            ),
          ),
          TextField(
            controller: _usernameController,
            decoration: InputDecoration(labelText: '
                Username'),
          ),
          SizedBox(height: 16),
          TextField(
            controller: _passwordController,
            obscureText: true,
            decoration: InputDecoration(labelText: '
                Password'),
    ),
    SizedBox(height: 24),
    ElevatedButton(
      onPressed: login,
      child: Text('Login'),
    ),
    TextButton(
      onPressed: () {
        Navigator.of(context).push(
            MaterialPageRoute(
          builder: (context) => Siginup(),
        ));
      },
      child: Text(
        "Don't have an account? Sign in",
        style: TextStyle(color: Colors.white,
            fontSize: 15),
```

```
                    ) ,
                ) ,
            ] ,
        ) ,
    ) ,
) ,
) ;
  }
}
```

siginpage.dart

```
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'package:http/http.dart' as http;
import 'dart:convert';

import 'package:schemeapp/screens/user/loginpage.dart';

void main() {
  runApp(MaterialApp(
    home: Siginup(),
  ));
}

class Siginup extends StatefulWidget {
  @override
  _MyHomePageState createState() => _MyHomePageState();
}

class _MyHomePageState extends State<Siginup> {
  final TextEditingController usernameController =
      TextEditingController();
  final TextEditingController emailController =
      TextEditingController();
  final TextEditingController passwordController =
      TextEditingController();
  final TextEditingController jobController =
      TextEditingController();
  final TextEditingController ageController =
      TextEditingController();
  final TextEditingController genderController =
      TextEditingController();

  List<String> jobList = ['Farmer', 'Student', 'Disabled
      ', 'Other'];
  // List<String> ageList = ['18-25', '26-35', '36-50',
```

```
    '51+'];
List<String> genderlist = ["Male", "Female"];

Future<void> postData() async {
  final String url =
      'http://10.0.2.2:8000/AdminUI/register/'; //
         Replace with your API endpoint

  // final Map<String, String> data = {
  //   'username': usernameController.text,
  //   'email': emailController.text,
  //   'password': passwordController.text,
  //   'profession': jobController.text,
  //   'age': ageController.text,
  // };

  final response = await http.post(
    Uri.parse(url),
    headers: {
      'Content-Type':
          'application/x-www-form-urlencoded', //
             Adjust the content type if needed
    },
    body: {
      'username': usernameController.text.trim(),
      'email': emailController.text.trim(),
      'password': passwordController.text.trim(),
      'age': ageController.text.trim(),
      'profession': jobController.text.trim(),
      'gender': genderController.text.trim()
    },
  );
  if (response.statusCode == 201) {
    Navigator.push(
        context,
        MaterialPageRoute(
          builder: (context) => LoginPage(),
        ));
    print("success");
  } else {
    print(response.statusCode);
    print("Error");
  }
}

@override
```

```
Widget build(BuildContext context) {
  return SafeArea(
    child: Scaffold(
      body: Container(
        height: double.infinity,
        decoration: BoxDecoration(
          image: DecorationImage(
            image: NetworkImage(
              "https://images.unsplash.com/photo
                  -1613125700782-8394bec3e89d?q=80&w
                  =1887&auto=format&fit=crop&ixlib=rb
                  -4.0.3&ixid=
                  M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA
                  %3D%3D",
            ),
            fit: BoxFit.fill,
          ),
        ),
        child: Padding(
          padding: const EdgeInsets.symmetric(vertical:
              10, horizontal: 10),
          child: SingleChildScrollView(
            scrollDirection: Axis.vertical,
            child: Column(
              mainAxisAlignment: MainAxisAlignment.
                spaceBetween,
              children: [
                Container(
                  alignment: Alignment.center,
                  child: Text(
                    "Sign in",
                    style: TextStyle(
                        fontSize: 30,
                        color: Colors.white,
                        fontWeight: FontWeight.bold),
                  ),
                  height: 50,
                  color: Colors.transparent,
                  width: double.maxFinite,
                ),
                SizedBox(
                  height: 100,
                ),
                TextField(
                  controller: usernameController,
                  decoration: InputDecoration(
```

```
            labelText: 'Username',
            labelStyle: TextStyle(color:
                Colors.black)),
    ),
    SizedBox(height: 16),
    TextField(
      controller: emailController,
      decoration: InputDecoration(
          labelText: 'Email',
          labelStyle: TextStyle(color:
              Colors.black)),
    ),
    SizedBox(height: 16),
    TextField(
      controller: passwordController,
      decoration: InputDecoration(
          labelText: 'Password',
          labelStyle: TextStyle(color:
              Colors.black)),
      obscureText: true,
    ),
    SizedBox(height: 16),
    DropdownTextField(
      controller: jobController,
      items: jobList,
      hintText: 'Select Job',
    ),
    SizedBox(height: 16),
    DropdownTextField(
      controller: genderController,
      items: genderlist,
      hintText: 'Select gender',
    ),
    SizedBox(height: 16),
    TextField(
      controller: ageController,
      decoration: InputDecoration(
          labelText: 'Age',
          labelStyle: TextStyle(color:
              Colors.black)),
      obscureText: false,
    ),
    SizedBox(height: 32),
    ElevatedButton(
      onPressed: () {
        postData();
```

```
                    },
                    child: Text('Post Data'),
                  ),
                ],
              ),
            ),
          ),
        ),
      );
    }
  }

  class DropdownTextField extends StatelessWidget {
    final TextEditingController controller;
    final List<String> items;
    final String hintText;

    const DropdownTextField({
      Key? key,
      required this.controller,
      required this.items,
      required this.hintText,
    }) : super(key: key);

    @override
    Widget build(BuildContext context) {
      return TextField(
        controller: controller,
        decoration: InputDecoration(
          labelText: hintText,
          suffixIcon: PopupMenuButton<String>(
            icon: const Icon(Icons.arrow_drop_down),
            onSelected: (value) {
              controller.text = value;
            },
            itemBuilder: (BuildContext context) {
              return items.map<PopupMenuItem<String>>((
                  String value) {
                return PopupMenuItem<String>(
                  value: value,
                  child: Text(value),
                );
              }).toList();
            },
          ),
```

```
        ),
      );
    }
}
```

splash.dart

```
import 'dart:async';
import 'package:flutter/material.dart';
import 'package:schemeapp/screens/user/loginpage.dart';

class SplashPage extends StatefulWidget {
  @override
  _SplashPageState createState() => _SplashPageState();
}

class _SplashPageState extends State<SplashPage> {
  @override
  void initState() {
    super.initState();

    Timer(
      Duration(seconds: 6),
      () => Navigator.pushReplacement(
        context,
        MaterialPageRoute(builder: (context) => LoginPage
            ()),
      ),
    );
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    body: Center(
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
         Image.network("https://www.logopeople.in/wp-
             content/uploads/2013/01/government-of-india
             .jpg")

        ],
      ),
    ),
  );
}
```

```
}

void main() {
  runApp(MaterialApp(
    home: SplashPage(),
    debugShowCheckedModeBanner: false,
  ));
}
```

adminpage.dart

```
import 'package:flutter/material.dart';

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Admin(),
    );
  }
}

class Admin extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Form Page'),
      ),
      body: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.stretch,
          children: [
            TextFormField(
              decoration: InputDecoration(labelText: '
                  Main Heading'),
            ),
            SizedBox(height: 16.0),
            TextFormField(
              decoration: InputDecoration(labelText: 'Age
                  '),
              keyboardType: TextInputType.number,
```

```
),
SizedBox(height: 16.0),
TextFormField(
  decoration: InputDecoration(labelText: '
      Description'),
),
SizedBox(height: 16.0),
                TextFormField(
                  decoration: InputDecoration(labelText: '
                      Category'),
                ),
                SizedBox(height: 32.0),
                ElevatedButton(
                  onPressed: () {

                  },
                  child: Text('Submit'),
                ),
              ],
            ),
          ),
        );
    }
}
```

filter_scheme_servicce.dart

```
import 'package:http/http.dart' as https;
import 'package:schemeapp/model/filterscheme.dart';
import 'package:schemeapp/screens/user/loginpage.dart';

// class HttpFilterScheme {
//   static String authToken =
//       ''; // Static variable to store the
    authentication token

//   static void setAuthToken(String token) {
//     authToken = token;
//   }

class HttpFilterScheme {
  static Future<dynamic> filterscheme() async {
    // var token = LoginPage.authToken;
    var response = await https.get(
      Uri.parse("http://10.0.2.2:8000/Scheme/myscheme/"),
      headers: {
        "Authorization":
```

```
"Bearer "eyJhbGciOiJI" "UzI1NiIsInR5cCI6Ik" "
    pXVCJ9.eyJ0b2tlbl9" "0eXBlIjoiYWNjZXNz" "
    IiwiZXhwIjoxN" "zA1MDc2Njc" "2LCJpYXQiOjE"
     "3MDUwNDA2N" "zYsImp0aSI6ImE1" "
    MTYxMDY4MTEz" "YzRkMzZiN2E" "
    zMTRhYWE5Y2NlZD" "VkIiwidXNlcl" "9
    pZCI6MSwidX" "Nlcm5hbWUiO" "
        iJhZG1pbiIsInBhc3" "N3b3JkIjoic" "
        GJrZGYyX3NoYTI" "1NiQ2MDAwM" "
        DAkQ2JMZUowam" "NROGxhVWJET1lu" "
        dHdlTSR0aEk4Y" "k9MZzNKWGw3" "
        Q0Z5eDZJV1JBVUJ" "ydm8rZHRPRlUvN" "
        y9GaytHRnE0PSIsIm" "F1dGgiOnR" "ydWV9.99
        hg34" "HC6zTC9nOu1q2_" "9-
        Vh4S6_J36cWilDV28AFHs"
    },
  );
  if (response.statusCode == 200) {
    var data = response.body;
    print(response.body);
    return filterSchemeFromJson(data);
  }
 }
}
```

service.dart

```
import 'package:http/http.dart' as https;
import 'package:schemeapp/model/model.dart';

class HttpScheme {
  static Future<dynamic> fetchscheme() async {
    var response = await https
        .get(Uri.parse("http://10.0.2.2:8000/Scheme/
            scheme/?format=json"),
        // headers: {"auth":"Bearer rtdyfgujoikl"}
        );
    if (response.statusCode == 200) {
      var data = response.body;
      print(response.body);
      return schmeModelFromJson(data);
    }
  }
}
```

Mysche.dart

```
import 'package:flutter/material.dart';
```

```
import 'package:get/get.dart';
import 'package:schemeapp/controller/
    filter_scheme_controller.dart';
import 'package:schemeapp/model/filterscheme.dart';
import 'package:url_launcher/url_launcher.dart';

class FilterSchemePage extends StatelessWidget {
  final FilterSchemeController filterSchemeController =
      Get.put(FilterSchemeController());

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.green[100],
      appBar: AppBar(
        backgroundColor: Colors.green[100],
        centerTitle: true,
        title: Text(
          "Filter Scheme",
          style: TextStyle(
            fontSize: 30.0,
            fontWeight: FontWeight.bold,
            color: Colors.black,
          ),
        ),
      ),
      body: Obx(
        () => ListView.builder(
          itemCount: filterSchemeController.flist.length,
          itemBuilder: (context, index) {
            FilterScheme filterScheme =
                filterSchemeController.flist[index];

            return Padding(
              padding: const EdgeInsets.symmetric(
                  vertical: 20),
              child: Card(
                child: Container(
                  // height: 70,
                  alignment: Alignment.center,
                  padding: EdgeInsets.all(10),
                  child: Column(
                    crossAxisAlignment:
                        CrossAxisAlignment.start,
                    mainAxisAlignment: MainAxisAlignment.
                        center,
```

```
children: [
  Text(
    '${filterScheme.schemeName}',
    textAlign: TextAlign.left,
    style: TextStyle(
      fontSize: 25,
      letterSpacing: 1.1,
      fontWeight: FontWeight.bold,
      color: Colors.black,
    ),
),
SizedBox(height: 10),
Text(
  '${filterScheme.description}',
  textAlign: TextAlign.left,
  style: TextStyle(
    fontSize: 18,
    fontWeight: FontWeight.bold,
    color: Colors.grey[800],
  ),
),
SizedBox(height: 10),
InkWell(
  onTap: () => launchURL(
      filterScheme.link),
  child: Text(
    '${filterScheme.link}',
    textAlign: TextAlign.center,
    style: TextStyle(
      fontSize: 18,
      fontWeight: FontWeight.bold,
      color: Colors.blue,
    ),
  ),
),
SizedBox(height: 5),
// Container(
//   child: Text('${scheme.
    description}'),
// ),
// GestureDetector(
//   onTap: () {
//     launchURL(scheme.link);
//   },
//   child: Text(
//     '${scheme.link}',
```

```
//       style: TextStyle(
//         color: Colors.blue,
//         decoration: TextDecoration
    .underline,
//       ),
//     ),
// ),
// SizedBox(height: 5),
// Container(
//   width: 80,
//   height: 35,
//   child: Card(
//     shape:
    RoundedRectangleBorder(
//       borderRadius: BorderRadius
    .circular(20),
//       ),
//       shadowColor: const Color.
    fromARGB(255, 246, 16, 0),
//       child: Center(child: Text('$
    {scheme.schmeModelNew}')),
//   ),
// ),

SizedBox(height: 5),
                    ],
                  ),
                ),
              );
            },
          ),
        ),
      ),
    );
}

void launchURL(String? link) async {
  if (await canLaunch(link!)) {
    await launch(link,
        forceSafariVC: false, forceWebView: false,
            enableJavaScript: true);
  } else {
    throw 'Could not launch $link';
  }
}
```

uu.dart

```
import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;

import 'package:schemeapp/screens/user/loginpage.dart';

void main() {
  runApp(MaterialApp(
    home: Profile(),
  ));
}

class Profile extends StatefulWidget {
  @override
  _ProfilePageState createState() => _ProfilePageState();
}

class _ProfilePageState extends State<Profile> {
  final TextEditingController usernameController =
      TextEditingController();
  final TextEditingController addressController =
      TextEditingController();
  final TextEditingController stateController =
      TextEditingController();
  final TextEditingController pinController =
      TextEditingController();
  final TextEditingController aadharController =
      TextEditingController();
  final TextEditingController contactController =
      TextEditingController();
  final TextEditingController professionController =
      TextEditingController();
  final TextEditingController cardController =
      TextEditingController();
  final TextEditingController qualController =
      TextEditingController();
  final TextEditingController disableController =
      TextEditingController();

  List<String> profession = ['Farmer', 'Student', '
      Disabled', 'Other'];
  // List<String> ageList = ['18-25', '26-35', '36-50',
      '51+'];
  List<String> card = ["Apl", "Bpl", "Aay"];
  List<String> qual = ["sslc", "Plus tow", "ug", "pg"];
```

```
List<String> disable = [
  "Mobility impairment",
  "visual impairment",
  "Hearing impairment",
  "Learing Disabilty",
  "Autism Spectrum Disoder",
  "Speach Imaariment",
  "Inrellectual Disability"
];

Future<void> postData() async {
  final String url =
      'http://10.0.2.2:8000/AdminUI/profile/'; //
         Replace with your API endpoint

  // final Map<String, String> data = {
  //   'username': usernameController.text,
  //   'email': emailController.text,
  //   'password': passwordController.text,
  //   'profession': jobController.text,
  //   'age': ageController.text,
  // };

  final response = await http.post(
    Uri.parse(url),
headers: {
  'Content-Type':
      'application/x-www-form-urlencoded', //
         Adjust the content type if needed
},
body: {
  // 'username': usernameController.text.trim(),
  // 'email': emailController.text.trim(),
  // 'password': passwordController.text.trim(),
  // 'age': ageController.text.trim(),
  // 'profession': jobController.text.trim(),
  // 'gender': genderController.text.trim()

  'user': usernameController.text.trim(),
  'Address': addressController.text.trim(),
  'State': stateController.text.trim(),
  'Pincode': pinController.text.trim(),
  'Aadhar_Number': aadharController.text.trim(),
  'Contact': contactController.text.trim(),
  'profession': professionController.text.trim(),
  'card': cardController.text.trim(),
```

```
        'qual': qualController.text.trim(),
        'disable': disableController.text.trim(),
      },
    );
    if (response.statusCode == 201) {
      Navigator.push(
          context,
          MaterialPageRoute(
            builder: (context) => LoginPage(),
          ));
      print("success");
    } else {
      print(response.statusCode);
      print("Error");
    }
  }

  @override
  Widget build(BuildContext context) {
    return SafeArea(
      child: Scaffold(
        body: Container(
          height: double.infinity,
          decoration: BoxDecoration(),
          child: SingleChildScrollView(
            child: Padding(
              padding: const EdgeInsets.symmetric(
                  horizontal: 10),
              child: SingleChildScrollView(
                scrollDirection: Axis.vertical,
                child: Column(
                  mainAxisAlignment: MainAxisAlignment.
                      spaceBetween,
                  children: [
                    Container(
                      alignment: Alignment.center,
                      child: Text(
                        "Profile",
                        style: TextStyle(
                            fontSize: 30,
                            color: Colors.black,
                            fontWeight: FontWeight.bold),
                      ),
                      height: 50,
                      color: Colors.transparent,
                      width: double.maxFinite,
```

```
),
// SizedBox(
//   height: 100,
// ),
TextField(
  controller: usernameController,
  decoration: InputDecoration(
      labelText: 'Username',
      labelStyle: TextStyle(color:
          Colors.black)),
),
SizedBox(height: 16),
TextField(
  controller: addressController,
  decoration: InputDecoration(
      labelText: 'address',
      labelStyle: TextStyle(color:
          Colors.black)),
),
SizedBox(height: 16),
TextField(
  controller: stateController,
  decoration: InputDecoration(
      labelText: 'state',
      labelStyle: TextStyle(color:
          Colors.black)),
  obscureText: true,
),
SizedBox(height: 16),
TextField(
  controller: pinController,
  decoration: InputDecoration(
      labelText: 'pincode',
      labelStyle: TextStyle(color:
          Colors.black)),
),
SizedBox(height: 16),
TextField(
  controller: aadharController,
  decoration: InputDecoration(
      labelText: 'aadhar no',
      labelStyle: TextStyle(color:
          Colors.black)),
),
SizedBox(height: 16),
TextField(
```

```
            controller: professionController,
            decoration: InputDecoration(
                labelText: 'job',
                labelStyle: TextStyle(color:
                    Colors.black)),
        ),
        SizedBox(height: 16),
        DropdownTextField(
          controller: cardController,
          items: card,
          hintText: 'Ration card',
        ),
        SizedBox(height: 16),
        TextField(
          controller: qualController,
          decoration: InputDecoration(
              labelText: 'qualificcation',
              labelStyle: TextStyle(color:
                  Colors.black)),
        ),
        SizedBox(height: 16),
        TextField(
          controller: disableController,
          decoration: InputDecoration(
              labelText: 'category',
              labelStyle: TextStyle(color:
                  Colors.black)),
        ),
        SizedBox(height: 16),
        TextField(
          controller: contactController,
          decoration: InputDecoration(
              labelText: 'contact',
              labelStyle: TextStyle(color:
                  Colors.black)),
          obscureText: false,
        ),
        SizedBox(height: 32),
        ElevatedButton(
          onPressed: () {
            postData();
          },
            child: Text('Post Data'),
          ),
        ],
      ),
```

```
                    ) ,
                ) ,
            ) ,
        ) ,
    ) ,
  ) ;
}
```

```
class DropdownTextField extends StatelessWidget {
  final TextEditingController controller;
  final List<String> items;
  final String hintText;

  const DropdownTextField({
    Key? key,
    required this.controller,
    required this.items,
    required this.hintText,
  }) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return TextField(
      controller: controller,
      decoration: InputDecoration(
        labelText: hintText,
        suffixIcon: PopupMenuButton<String>(
          icon: const Icon(Icons.arrow_drop_down),
          onSelected: (value) {
            controller.text = value;
          },
          itemBuilder: (BuildContext context) {
            return items.map<PopupMenuItem<String>>((
                String value) {
              return PopupMenuItem<String>(
                value: value,
                child: Text(value),
              );
            }).toList();
          },
        ),
      );
  }
}}
```

main.dart

```
import 'package:flutter/material.dart';
import 'package:schemeapp/screens/user/Siginpage.dart';
import 'package:schemeapp/screens/user/admin.dart';
import 'package:schemeapp/screens/user/homepage.dart';
import 'package:schemeapp/screens/user/loginpage.dart';
import 'package:schemeapp/screens/user/splash.dart';
import 'package:schemeapp/views/Mysche.dart';

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: SplashPage(),
    );
  }
}
```

WEBSITE OF ABILITY CONNECT

PROJECT REPORT

Submitted By

LEYON T JOHN

Reg. No.CCAVBCA042

For the award of the Degree of

Bachelor of COMPUTER APPLICATION(BCA)

in Computer Science

(University of Calicut)

under the guidance of

Ms. Rasmi P M

Assistant Professor

BACHELOR OF COMPUTER APPLICATION
DEPARTMENT OF COMPUTER SCIENCE
CHRIST COLLEGE
IRINJALAKUDA, KERALA

2021-2024

DEPARTMENT OF COMPUTER SCIENCE

CHRIST COLLEGE (AUTONOMOUS)

IRINJALAKUDA

CERTIFICATE

This is to certify that the project report entitled ***"Website of Ability Connect"*** *is a bonfied record of the project work done by* ***LEYON T JOHN*** *in partial fulfillment of the requirement for the sixth semester of* ***Bachelor of Computer Science*** *in Department of Computer Science of* **CHRIST COLLEGE (AUTONOMOUS) IRINJALAKUDA**

Ms. Rasmi P M
Assistant Professor,CS
Internal Guide

Ms. Sini Thomas
Head of the Department
Computer Science

EXTERNAL EXAMINER

INTERNAL EXAMINER

DECLARATION

We here by declare that this project work **"WEBSITE OF ABILITY CONNECT"** submitted by Christ College (Autonomous)Irinjalakuda,affliated to Calicut University in partial fullfillment of the requirement for the award of the Bachelor of Computer Application, is a record of original work done by us,under the guidance of Ms.RASMI P M,Department of Computer Science.

Place: Irinjalakuda LEYON T JOHN

ACKNOWLEDGEMENT

First and foremost we like to thank Lord almighty for his providence and for being the guiding light throughout the project. We wish to express my sincere gratitude to our beloved Department head for giving me all the facilities for our project. We take this opportunity to express my gratitude to the class teacher Ms.SOWMYA P.S and head of the department Ms.SINI THOMAS who has been supported us throughout the course of this project. We are thankful for her aspiring guidance and valuable advice during the project work. We express my sincere thanks to my project guide Ms.RASMI P M for supporting and guiding throughout the project.We would take this opportunity to specially thank all other faculty members for their constant and continuous motivation. Finally we would like to thank my family and friends for giving valuable advice and moral support throughout our project.

ABSTRACT

WEBSITE OF ABILITY CONNECT is a innovative website introduced as a learning platform for differently abled students in assistance of Computer Science Department of Christ College(Autonomous) Irinjalakuda. The website is enriched with two kind of login facilities - student login, teachers login. It is a learning platform and the main features are- student registration, learning and exam dashboards and so on. All these features make this website more adaptable and user-friendly.

Contents

Chapter 1

1 Introduction

Education is the cornerstone of empowerment and growth, yet education systems often fail to accommodate the diverse needs of students with various disabilities .Our website endeavors to help the learning process by crafting inclusive assessments specifically tailored to each student's abilities, while also introducing a spectrum of learning methods—audio, text, video, and interactive activities that can help foster a learning environment.

.

1.1 Overview

The objective of the Ability Connect Website is to design an simple and adaptable website that helps the users in their learning process by tailoring different modes through which they can access education . Different modes of examinations, in-website voice navigation and other features helps website to be user friendly and create a learning environment for differently abled students .

Chapter 2

2 System Analysis

2.1 Purpose

The main purpose of the website owned by Don Davis, Adhithyan T.J, Leyon T.John ,Mithra prothasis is to make a user friendly website as a inclusive learning platform for diffrently abled students.

2.1.1 Existing System

In the existing system, there may be limited opportunities for disabled children to learn or develop their skills independently, especially at a young age. While schools and special education programs may offer support from teachers and other aides, there may be gaps in providing opportunities for self-directed learning and skill development outside of structured classroom settings. Additionally, parents of disabled children may face challenges in providing supplemental learning experiences or resources at home due to lack of time, knowledge, or access to suitable materials. Limitations of existing system : Self-directed learning and skill development outside of structured classroom settings and difficulties in accesing diffrent modes of learning according to disabilities.

2.1.2 Proposed System

The materials will be designed to be accessible to children with various disabilities. Users can specify any disabilities or special needs they have to ensure that the platform can customize them.The website will incorporate learning technologies to adjust the difficulty level and pacing of activities based on the student's performance and progress. This ensures that each student is appropriately challenged and supported throughout their learning journey. Proposed system aims to create an inclusive and empowering learning environment for disabled children. The website will provide feedback to students on their performance and progress, including scores, achievements, and areas for improvement. Parents or teachers may also have access to progress reports to monitor the student's development and provide additional support as needed. Other Features : A message system that sends notification to user mail about the new contents or exams assigned.

2.2 Problem definition

To know what the problem is and what the needs are before developing it. Here we design a website as a learning platform for differently abled students.

2.3 Feasibility Study

After the problem is clearly understood and solutions proposed, the next step is to conduct the feasibility study. Feasibility study is defined as evaluation or analysis of the potential impact of the proposed project or program. The objective is to determine whether the proposed system is feasible. There are three aspects of feasibility study to which the proposed system is subjected as discussed below.

2.3.1 Technical Feasibility

Technical feasibility assess whether the current technical resources are sufficient for the new system. We can upgrade the level of technology for supporting our website. We check whether the proposed system can be implemented in the present system without supporting the existing hardware.

2.3.2 Economical Feasibility

Economic feasibility determines whether the time and money are available to develop the system. There is no additional hardware used to develop the site, it is inexpensive to build.

2.3.3 Operational Feasibility

Operational feasibility determines if the human resources are available to operate the system once it has been installed. No extra training is needed to use this system. Anyone who has knowledge in internet in english language can easily use the system. The resources that are required to implement or install are already available with the organization.

Chapter 3

3 Software Requirement Specification

3.1 Purpose

The purpose of creating an educational website for differently-abled students is multifaceted and revolves around addressing the unique needs and challenges that these students may face. The educational website for differently-abled students is to create an inclusive, accessible, and supportive online learning environment that empowers individuals with diverse abilities to thrive academically and personally. The project aims to bridge gaps, break down barriers, and contribute to a more inclusive educational landscape.

3.2 Scope

The scope of the "Educational Website for Differently-Abled Students" project is a comprehensive initiative aimed at developing an inclusive and accessible online learning platform. The project will focus on catering to the unique needs of differently-abled students, educators, and administrators, providing a supportive and engaging environment for learning. It also aims to create a holistic and inclusive educational platform that not only meets the immediate needs of its users but also lays the groundwork for ongoing growth and enhancement.

3.3 Overall Description

The project for developing an "Educational Website for Differently-Abled Students" is a visionary initiative with the primary goal of creating an inclusive and accessible online learning platform. This comprehensive website is designed to cater to the diverse educational needs of students with varying abilities, ensuring they have equal access to quality learning resources and a supportive community. At the heart of the project is the creation of an adaptive learning environment. The website will feature a dynamic system that tailors educational content to the individual needs, preferences, and learning styles of differently-abled students.

3.3.1 Product Perspective

The "Educational Website for Differently Abled Students" involves considering how the website fits into the broader context of educational technology and the needs of its users.

3.3.2 Product Functionality

Through this website teachers can upload study materials and quizzes as a part of exam and teacher can see the result of each student. The students get notified

if new material or new test been assigned. And the main highlight is that this website can be voice controlled specially for blind students.

3.3.3 Users and Characteristics

There are two types of users admin or teacher and student. In admin page the teacher can upload new study materials and tests, teacher can also see the growth of each student. And teachers can give suggestions for each type of student. In the student page they can see all the study materials and tests. There will be an entertainment section where there will be small stories, contents and games.

3.4 Specific Requirements

3.4.1 Hardware Requirements

- System: IBM-Compatible PC
- Processor: Intel Core i3 Or Above
- Speed: Above 1GHz
- RAM capacity: 4 GB Or Above
- Hard Disk Drive: 256 GB Or Above

3.4.2 Software Requirements

- Front End : HTML, CSS, Javascript
- Back End : Python
- Database : Sqlite3
- IDE : Visual Studio Code

3.5 Functional Requirements

It contains two main modules.

- 1.Teacher/Admin
- 2.Student

Admin

An admin account is used for editing or managing the website dynamically by admin panel. The admin can add new study materials, new tests and they can also review student performance and assign necessary suggestions according to their performance. The admin will be acting as teacher role.

Student

The user or the student can login the website, and can access the study materials assignd by the admin or teacher. The will be tests, suggestions, and small entertainment section for leisure time.

3.6 Non Functional Requirements

Non-functional requirements define the overall qualities or attributes of the resulting system. Non-functional system place restrictions on the product being developed, the developed process, and specify external constraints that the product must meet. Examples of non-functional requirements include safety, security, usability, reliability and performance requirements. Project management issues (costs, time and schedule) are often considered as non-functional requirements. The principal non - functional constraints which are relevant to critical systems :

- performance
- security
- safety
- usability

Performance

Performance requirements concern the speed of operation of a system. Types of performance requirements :

- Response requirements (how quickly the system reacts to a user input).
- Throughput requirements (how much the system can accomplish within a specified amount of time).
- Availability requirements (is the system available for service when requested by end users). The speed of operation of this system is adequate for the requirements.

Reliability

- Reliability is the ability of a system to perform its required function under stated conditions for a specified period of time.
- constraints on the runtime behavior of the system. This system is reliable because its functionalities can be done on the required conditions.

Safety

Safety requirements are not required which exclude unsafe situation from the possible solution of the system.

Usability

Usability is the ease with which a user can learn to operate, prepare inputs for, and interpret outputs of system or components. Usability requirements include :

- information error messages.
- well-formed user interfaces.

3.7 Interface Requirements

3.7.1 Hardware interfaces

The system must run over the internet, all the hardware shall require to connect internet will be hardware interface for the system. As for example modem, WAN-LAN.

3.7.2 Software interfaces

Software interface required for the working of the project is the appropriate operating system.

3.7.3 Communication interfaces

The user of the site communicate using the network connectivity and the data set is accessed

3.8 Security Requirements

- User accesses only their account.
- Validation of input is handled.
- This application containing the computer systems is physically secured against arms or surreptitious entry by intruders.
- Users must be authorized carefully to reduce changes of any such user giving access to an intruder in exchange for a bribe or other favour.

3.9 Platform Used

Windows 10 is a major version of the Microsoft Windows operating system that was released on July 29, 2015.It is built on the Windows NT kernel and follows windows 8.Part of the reason Microsoft decided to name the 2015 release "Windows 10"(and skipped "windows 9")is because the operating system is designed to be a new direction for Microsoft. One of the primary aims of windows 10 is to Unify the windows experience across multiple devices, such desktop computers, tablets, and smartphones. As part of this effort, Microsoft developed Windows10 Mobile alongside Windows 10 to replaces Windows Phone

- Microsoft's previous mobile OS. Windows 10 also integrates other Microsoft services, such as Xbox Live and the Cortana voice recognition assistant.

3.10 Technologies Used

JavaScript

JavaScript is a programming language that enables interactivity and dynamic behavior on web pages. It can be used to manipulate the HTML and CSS of a webpage, handle user interactions, fetch data from servers, and much more.HTML and CSS can be embedded with JavaScript to create a working efficient website. HTML is the standard markup language for creating web pages. It provides the structure of a webpage by using elements or tags to define different parts of the content. CSS is used to style the HTML elements and define their appearance on the webpage. It allows you to control the layout, colors, fonts, and other visual aspects of your website.

Sqlite3

SQLite is a lightweight, serverless, self-contained, and embedded SQL database engine. It's widely used in various applications due to its simplicity, efficiency, and ease of integration. In the context of a website, SQLite can be utilized to store and manage data on the server-side, providing persistence for web applications. The Sqlite libraries provide APIs for performing CRUD (Create, Read, Update, Delete) operations, executing SQL queries, and managing database connections. By integrating SQLite into your website, you can efficiently store and manage data, enabling features like user authentication, data persistence, and dynamic content generation.

Chapter 4

4 Design Document

4.1 Purpose

The purpose of the website is to provide accessible and inclusive educational resources for disabled students with visual and auditory impairments. The website aims to support these students in their studies by offering:

- Accessible Study Materials: The website provides notes and educational materials in various accessible formats, such as voice notes for blind students and video/photo classes for students with hearing impairments.
- Interactive Learning Tools: Students can engage with interactive learning tools tailored to their specific needs, including audio descriptions for visual content and subtitles or sign language interpretation for video content.
- Assessments and Tests: The website offers assessments and tests designed to accommodate the needs of disabled students, ensuring fair evaluation of their understanding and progress.
- Teacher Support and Suggestions: Teachers can provide personalized support and suggestions to disabled students through the platform, offering guidance and assistance to help them succeed academically.
- Voice Command Input: For blind students, the website incorporates voice command input functionality, allowing them to navigate the platform and interact with content using voice commands, enhancing accessibility and usability.

4.2 Scope

The website aims to provide accessible study materials, including voice notes for blind students and video/photo classes for those with hearing impairments. It facilitates interactive learning through tests, offers teacher suggestions, and incorporates voice command input for blind users, fostering an inclusive educational environment for disabled students.

4.3 Overview

The purpose of this document is to help the reader to visualize the solution to the project presented.This document verifies how the design meet the requirement stimulated in the SRS document through design viewpoints.The design viewpoints will cover all design elements presented before. This document will provide a direct approach to the development of this project hence reducing feature creep and ponitedly determine the quality of the design.

4.4 Data Design

Database are the storehouses of data used in the software systems.The data is stored in tables inside the database.Several tables are created for the manipulation of the data for the system.Two essential settings for a database are

- Primary Key- The field that is unique for all the record occurrence.
- Foreign Key- The field used to set relation between tables.

Normalization

- Normalization is the process of organizing the data in the database.
- Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate the undesirable characteristics like Insertion, Update and Deletion Anomalies.
- Normalization divides the larger table into the smaller table and links them using relationship.
- The normal form is used to reduce redundancy from the database table.

Tables

Login

Name	DataType	Constraints	Description
username	varchar(100)	Notnull	Username of student
password	varchar(100)	Notnull	Password of student

Student

Name	DataType	Constraints	Description
studentid	Charfield(50)	Primarykey	ID of users
firstname	Charfield(100)	Notnull	First name of users
lastname	Charfield(100)	Notnul	Last name of users
email	Emailfield)	Notnull	Email of the users
gender	Charfield(100)	Notnull	Gender of the user
age	PositiveIntegerfield	Notnull	Age of user
disability	Charfield(100)	Notnull	Disability of user
access technology	Charfield(200)	Notnul	Users access technology

Question Table

Name	DataType	Constraints	Description
type	Charfield	Not Null	Question Type
text_id	Charfield	Foreignkey	Text Question
image	Filefield	Foreignkey	Image Question
audio	Filefield	Foreignkey	Audio Question

Scoremodel Table

Name	DataType	Constraints	Description
student	Charfield	Foreignkey	Student Name
score	Integerfield	Foreignkey	Exam Score
category	Charfield	Foreignkey	Category
suggestion	Textfield	Not Null	Suggestions

Suggestions

Name	DataType	Constraints	Description
suggestion	Textfield	Notnull	Suggestions
category	Charfield	Foreignkey	Category
video	Filefield	Foreignkey	Suggested Video
audio	Filefield)	Foreignkey	Suggested audio

Chapter 5

5 Development of the System

The website will feature a user-friendly interface with options for visually impaired students to access voice notes and hearing-impaired students to engage in video and photo classes. It will include an integrated testing platform, teacher feedback system, and voice command functionality for blind students, aiming to enhance accessibility and support disabled students in their studies.

Chapter 6

6 System Testing

Testing is the penultimate step of software development.An elaborate testing of the data is prepared and the system is using the test data,while doing testing,errors are noted and correction is made.The users are trained to operate the developed system.Both hardware and software securities are made to run the developed system successfully.System testing is aimed at ensuring the system works accurately before the live operation commences.

6.1 Test Plan

6.1.1 Scope

This test plan will cover the following testing activities as identified in the testing strategy.

- White Box Testing

White box testing is a testing technique, that examines the program structure and derives test data from the program logic/code. In white box testing, the UI is bypassed. Inputs and outputs of ratings are tested directly at the code level and the results are compared against specifications. It Reveals errors in "hidden" code.

- Black Box Testing

Black - box testing is a method of software testing that examines the functionality of an application based on the specifications. Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end - user would.

- Unit Testing

The module includes four main components to undergo unit testing. It checks the syntax error, logic error and validity of the program correctness. This test will be performed by the developers.

- Integration Testing

After all components pass the unit testing, they get to be tested whether they work correctly when they are running concurrently and communicating to each other. Integration tests exercise an entire subsystem and ensure that a set of components play nicely together.

- Internal Data Testing

We will test the validity of the data before it enters the database to avoid any problems that may face us in the database. We will test the encryption of the personal information of all the users along with the user names and passwords to ensure maximum security of the users privacy.

6.1.2 Software risk issues

In this section, the plan is to test the risk involved in critical issues such as: Difficult to run on visual studio due to large loading time. Some inherent software risks such as complexity exist; also these issues need to be identified. Proper network connection Working of Mysql database

6.1.3 Features to be tested

- Test whether correct user name and password allows you to login.
- Test whether invalid user name and password prevents you from login.
- Test whether there is any connection problem in the Server.
- Test whether the student and admin details are entered correctly.
- Test whether invalid data entry allows saving data successfully.
- Test whether all pages are loaded correctly.
- Test whether watermark is embedded and extracted properly.
- Test whether the provided security works correctly.

6.2 Test consolidation

6.2.1 Test item

The items or features to be tested in the test cases are included in the document. Each and every user input is tested. The present condition of the system is tested. It is checked to make sure that environment is ready for the application to work.

6.2.2 Input specifications

These are the inputs required to execute the test case. All required inputs including data elements and values are given below.

Data	Values
Email	A valid email ID
Password	Combination of characters and numbers
Phone number	10 Digit number

Chapter 7

7 System Implementation and Maintenance

7.1 Implementation

System implementation is the conversion of new system into an operating one which involves creating compatible files, training clients and installing hardware. User training is crucial for minimizing resistance to change and giving chance to prove its worth. Training aids user friendly manuals and healthy screens provide the user with a good start. Software maintenance follows conversion to the extent that changes are necessary to maintain satisfactory operations relative to changes in the user's environment. Maintenance often includes minor enhancements or corrections to the problem that surface late in the systems operations. In the implementation phase, the team builds the components either from scratch or by composition. Given the architecture document meant from the design phase and the requirement document from the analysis phase, the team should build exactly what has been requested, though there is still room for innovation and flexibility. For example, a component may be narrowly designed for this particular system, or the component may be made more general to satisfy a reusability

- Careful planning
- Investigation of system and constraints
- Design the methods to achieve changeover.
- Training the staff in the changed phase.
- Evaluation of change over method.
- The method of implementation and time scale to be adopted are found out initially.

7.2 Maintenance

This phase occurs as a result of deploying the whole system at the end users organization. They will perform the beta testing at the end users and inform to the developers about any needed modification to the application. The customer records all the problems that are encountered during the beta testing and reports these to the developer at regular intervals.

7.2.1 Corrective Maintenance

Even with the best quality assurance activities, it is likely that the customer will uncover defects in the software. Corrective maintenance changes the software to correct the defects. Corrective Maintenance activity may consist of repair,

restoration or replacement of equipment. This activity will be result of a regular inspection, which identifies the failure in time for corrective maintenance to be planned and scheduled, then performed during a routine maintenance shutdown.

7.2.2 Adaptive Maintenance

Over time, the original environment(CPU, operating system, business rules, external product characteristics) for which the software was developed is likely to change. Adaptive maintenance results in modification to the software to accommodate changes to its external environment.

7.2.3 Enhanced Maintenance

As software is used, the customer/user will recognize additional functions that will provide the benefit. Perfect maintenance extends the software beyond its original functional requirements.

7.2.4 Preventive Maintenance

Computer software deteriorates due to change, and because of this preventive maintenance often called software re-engineering, must be conducted to enable the software to serve the needs of its end users. Preventive maintenance makes changes to computer programs so that they can be more easily corrected, adapted and enhanced.

Chapter 8

8 Conclusion and Future Scope

8.1 Conclusion

There are three types of users that interact with the system Admin,Teacher and Student.Each of these three types has different uses of the system so each of them has their own panel.Admin can manage all the features of website dynamically by login on admin panel.Teacher can view the details about their event assigned by the admin and publish the result of the event.And the student can register for events and view the results of the events.

8.2 Future Scope

- Advanced adaptive learning technologies can be implemented.
- Advanced AI Chatbots

Appendix

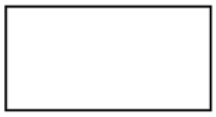
A Data Flow Diagram

Data flow is the one of the best way of documenting the entire functionality of the system.For the system ,which will have data flows in and have some processing inside and then some data flow out from the system can be documented or represented effectively by means of data flow out from the system can be documented or represented effectively by means of data flow diagrams. The data flow diagram are a diagrammatic representation of the system,which has input,process and output.Once any system is represented using a data flow diagram we can identify the following things easily:

- Various entities interacting with the system are identified
- Flow of data from one entity to another is identified
- The various processes involved in between the interaction of two or more entities in the system are clearly pointed out
- The various data stores which hold the data in between the process,are clearly identified

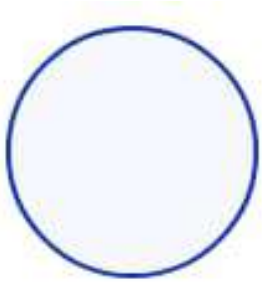
Some Data Flow Diagram charting forms:

A.1 External source or receiver



A source or sink is a person or part of organization ,which enters or receives information from the system,but is considered to be outlining the contest of data flow model.

A.2 Transform process

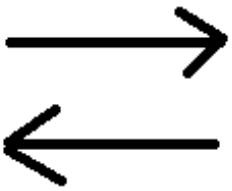


A process represents transformation where incoming data flowsare changed into outgoing data flow

A.3 Data Store

A data store is repository of data that is to be stored for use by one or more process may be as simple as buffer or queue or sophisticated as relational database. They should have clear names.If a process merely uses the contest of store and does not alter it,the arrowhead goes only form the store to the process. If a process alters the details in the store then double-headed arrow is used.

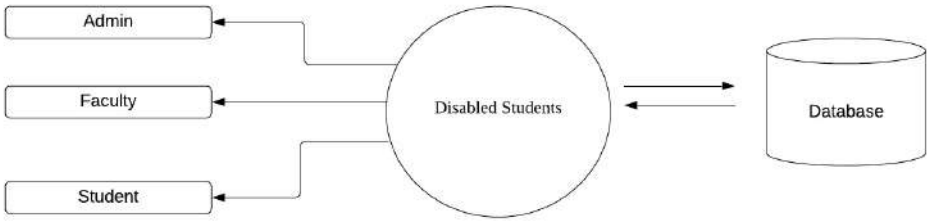
A.4 Data flow



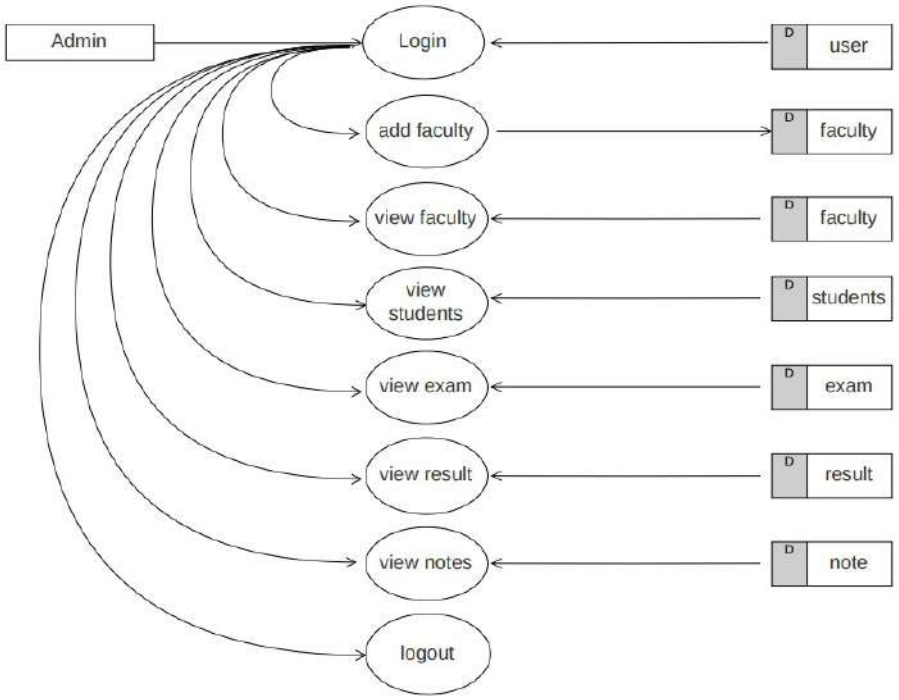
A data flow is a route, which enable packets of data to travel from one point to another.Data may flow,with arrowhead pointing in the direction of the flow

B Data Flow Diagrams

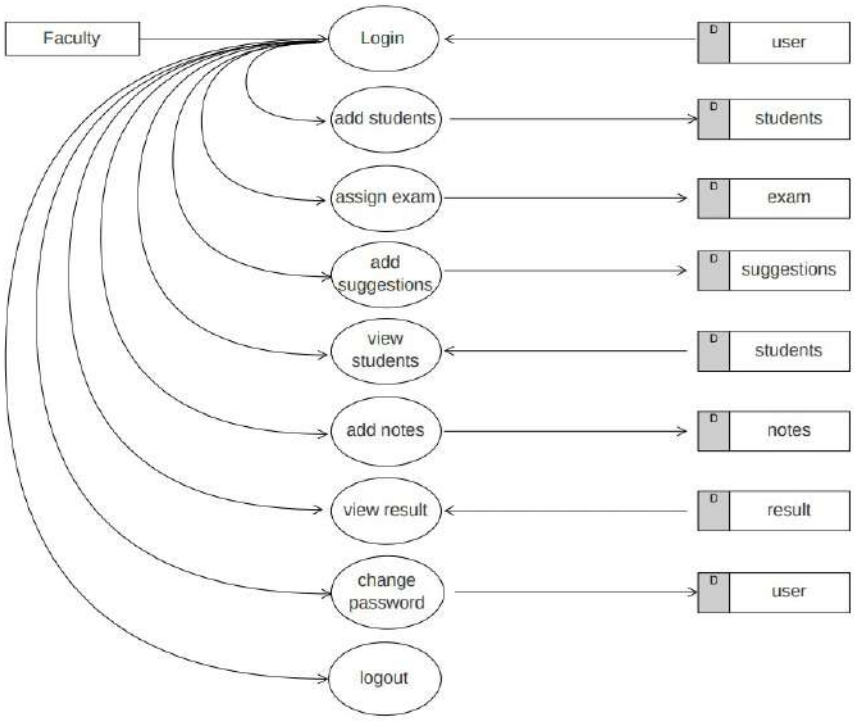
B.1 Level 0



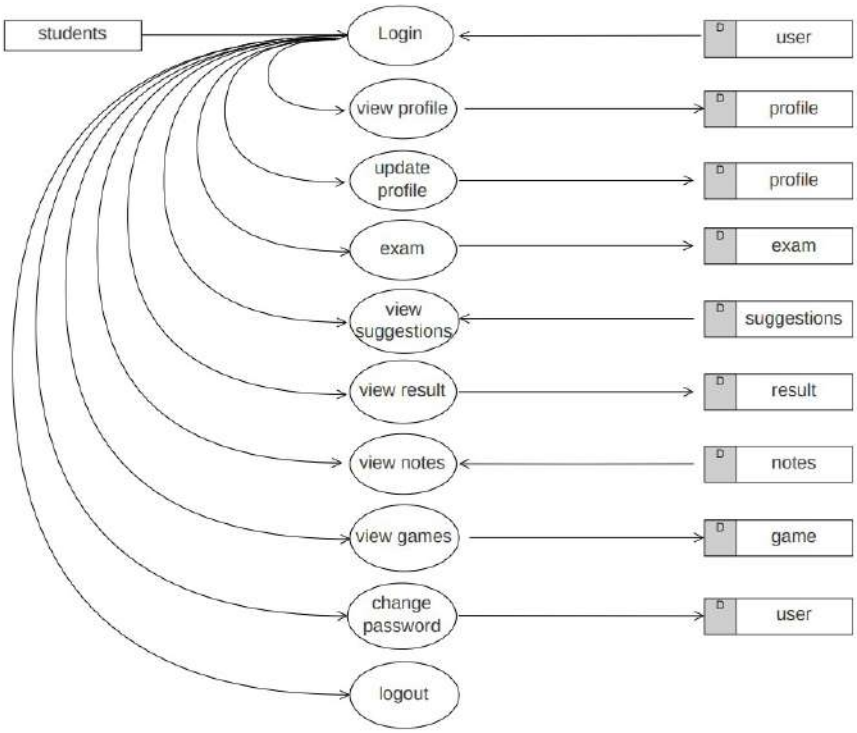
B.2 Level 1.1 - Admin



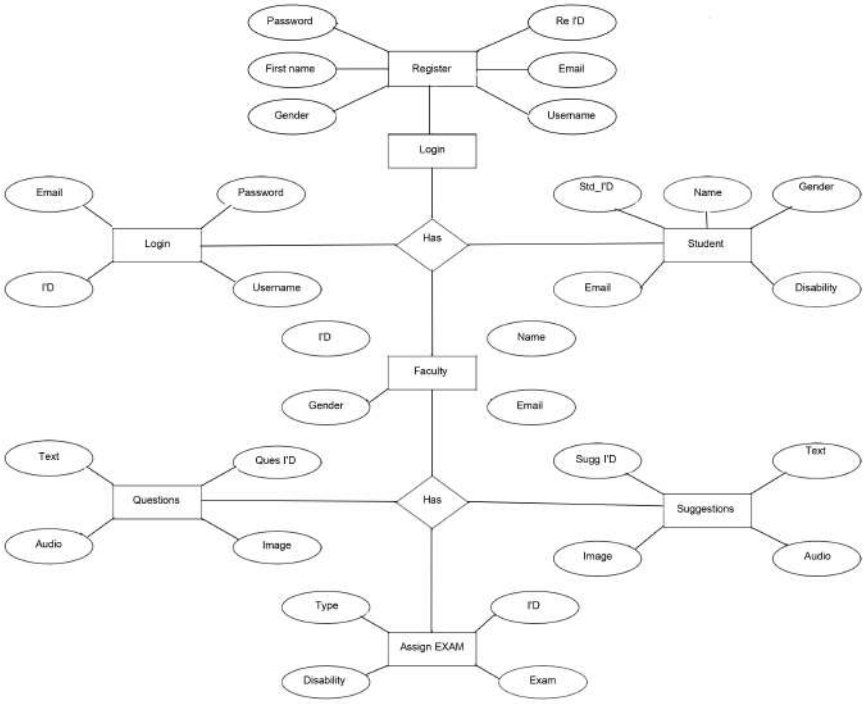
B.3 Level 1.2 - Faculty



B.4 Level 1.3 - Students

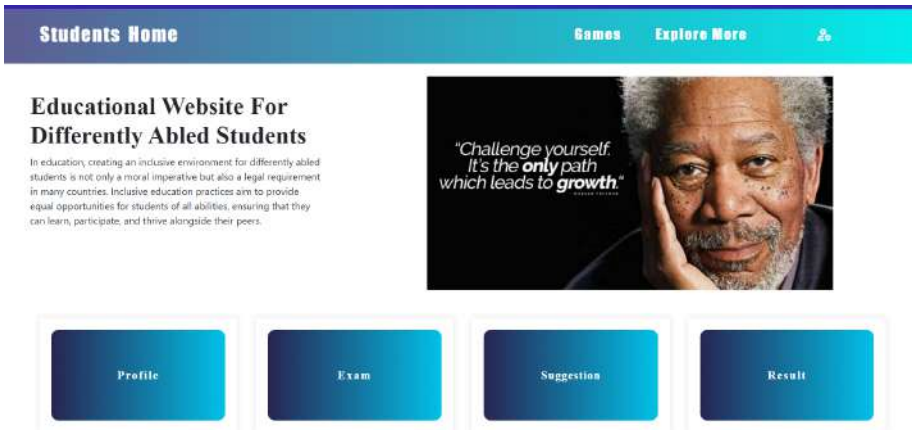


B.5 ER DIAGRAM

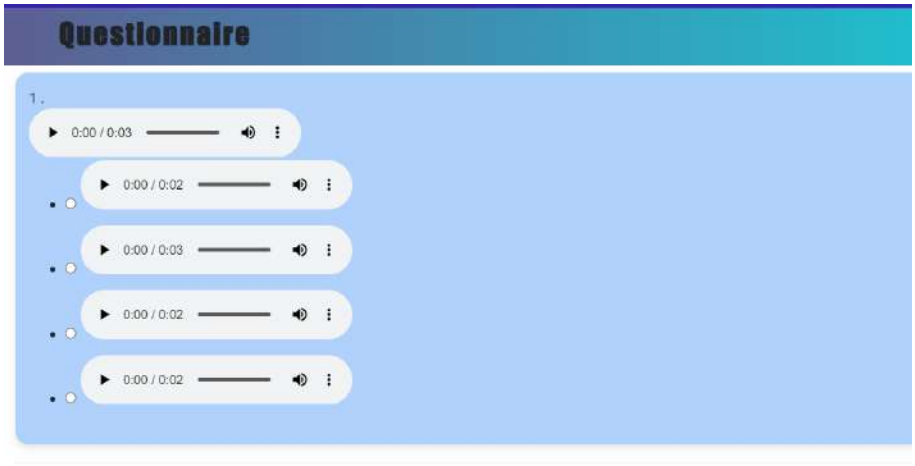


C USER INTERFACES

C.1 HOME



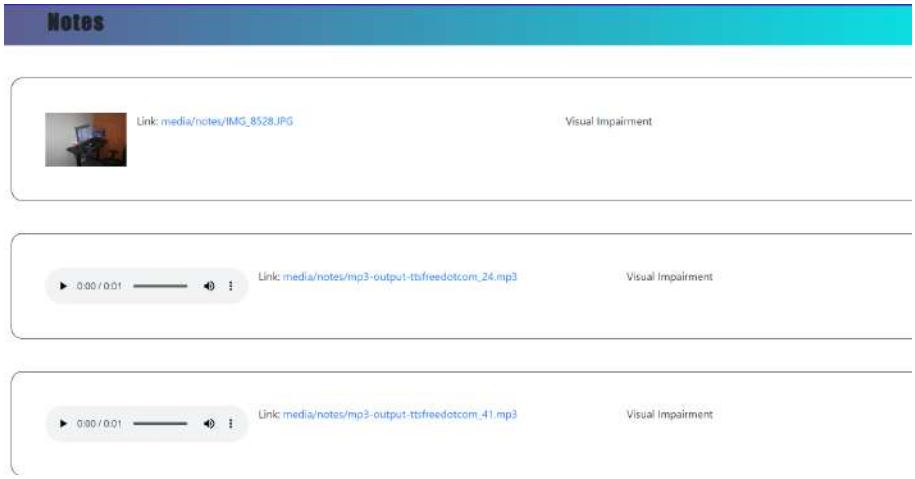
C.2 EXAM



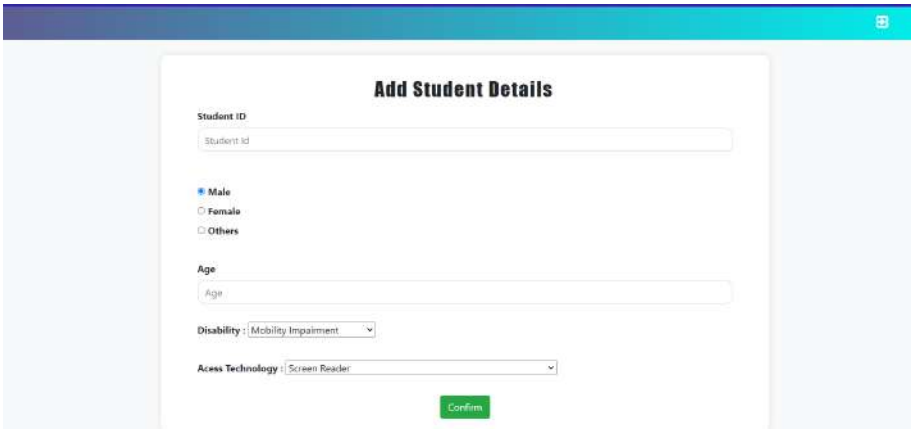
C.3 RESULT


Result
TestScore
1
Category
Very Poor
Result is here Check it....
No.
Question
Result
1
0:00 / 0:03
False
2
0:00 / 0:03
False
3
0:00 / 0:02
False
4
0:00 / 0:06
True

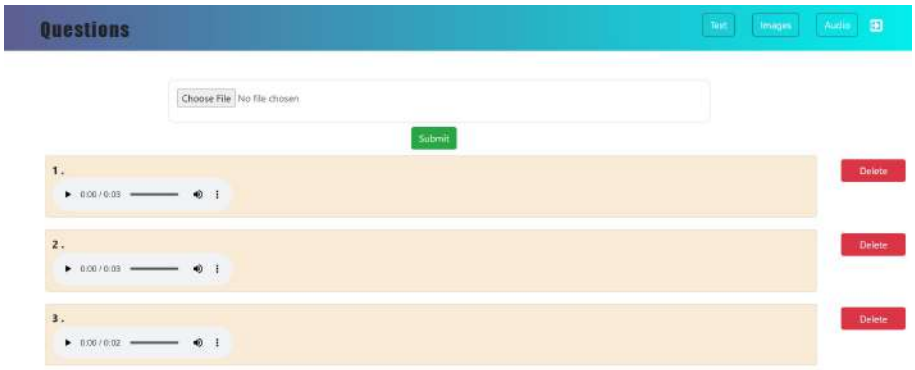
C.4 NOTES


Notes
Link: media/notes/IMG_8528.JPG
Visual Impairment
0:00 / 0:01
Link: media/notes/mp3-output-ttsfreedotcom_24.mp3
Visual Impairment
0:00 / 0:01
Link: media/notes/mp3-output-ttsfreedotcom_41.mp3
Visual Impairment

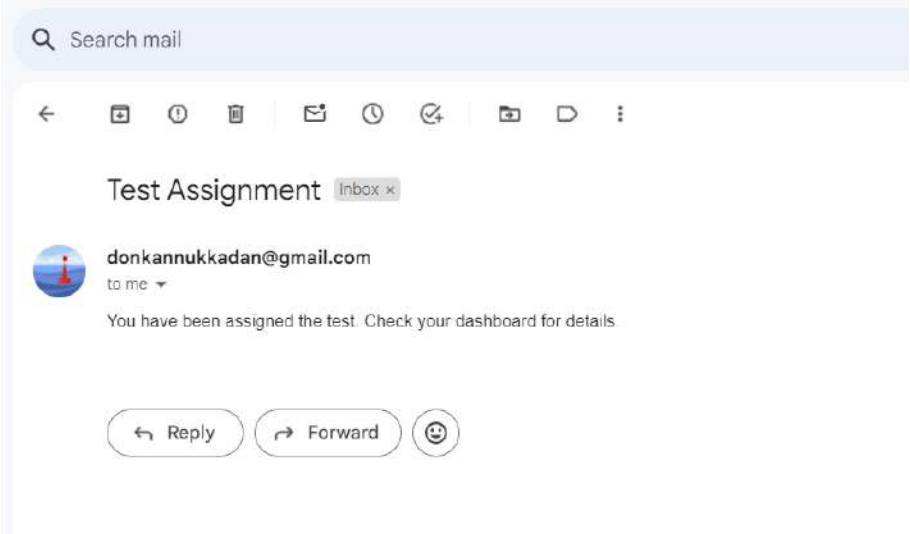
C.5 REGISTRATION



C.6 QUESTIONS WINDOW



C.7 NOTIFICATION



D CODE

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width,
      initial-scale=1.0">
  <!-- <link rel="stylesheet" href="styles.css"> -->
  <script src="toggle-sideNav.js" defer></script>
  <script src="featured-games.js" defer></script>

  <title>Game for mentally challenged people</title>
  <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,
        initial-scale=1, shrink-to-fit=no">
    <!-- Optional JavaScript -->
    <!-- jQuery first, then Popper.js, then Bootstrap JS
        -->
    <script src="https://code.jquery.com/jquery-3.3.1.
        slim.min.js" integrity="sha384-q8i/X+965
        DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8
        abtTE1Pi6jizo" crossorigin="anonymous"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/
        popper.js/1.14.7/umd/popper.min.js" integrity="
        sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9W
      O1clHTMGa3JDZwrnQq4sF86dIHNDz0W1" crossorigin="
          anonymous"></script>
    <script src="https://stackpath.bootstrapcdn.com/
        bootstrap/4.3.1/js/bootstrap.min.js" integrity="
        sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/
        nJGzIxFDsf4x0xIM+B07jRM" crossorigin="anonymous
        "></script>
    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://stackpath.
        bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min
        .css" integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH
        /1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
        crossorigin="anonymous">
    <link rel="stylesheet" href="https://fonts.googleapis
        .com/css2?family=Material+Symbols+Sharp:opsz,wght,
        FILL,GRAD@48,700,0,200" />
```

```
    <link rel="stylesheet" href="https://fonts.googleapis
        .com/css2?family=Material+Symbols+Outlined:opsz,
        wght,FILL,GRAD@24,700,0,200" />
   <link rel="stylesheet" href="https://fonts.googleapis.
       com/css2?family=Material+Symbols+Rounded:opsz,wght,
       FILL,GRAD@40,600,0,200" />
  </head>
<style>
  #play-now{
    text-decoration: none;
  }
</style>
<body>
{% load static %}
<div style="margin-left:95%;text-decoration: none;padding
    : 4px;border-radius: 0.5rem;" ><a href="{% url 'sh'
    %}" class="text-black ml-3" style="color: black;"><
    span class="material-symbols-outlined">
   exit_to_app
  </span></a></div>
    <section class="hero-wrapper">
        <div class="container">
          <p class="greeting " style="color: green;">
            Brain Games For Disabled Students!!!
          </p>
        </div>
      </section>
    <section id="play-now" class="program-wrapper">
      <div class="program-container container">
        <h3 class="title">
            Featured Games
        </h3>
        <div class="program-content">
          <div class="row">
            <div class="col">
                  <a href="{% url 'animal' %}" ><div
                      class="program-detail">
                    <img src="{% static 'images/first.png
                        ' %}" alt="reloading" width="350px
                        " height="450px">
                    <h5 class="mt-1">who am i ?</h5>
                  <p>A great game for the brain! It
                      improves visual scanning, planning,
                      and spatial memory!! </p>
                </div></a>
            </div>
```

```
            <div class="col">
              <a href="{% url 'math' %}"><div class="
                  program-detail">
                <img src="{% static 'images/third.jpg'
                    %}" alt="icon of a speaker" width="550
                    px" height="450px">
                <h5 class="mt-1">Fun with numbers</h5>
              <p>Helps to quickly solve the elementary
                  math problems!! </p>
            </div></a>
            </div>
        </div>
        <div class="row mt-5">
         <div class="col">
           <a href="{% url 'memory' %}"><div class="
               program-detail">
           <img src="{% static 'images/forth.jpg' %}"
              alt="icon of a speaker" width="500px">
           <h5>Behind the scenes</h5>
         <p>Helps to improve Visual Scanning and memory
             while remembering the cards!! </p>
         </div></a>
         </div>
        </div>
         </div>
    </section>
  <script>
   function playWelcomeMessage() {
         const welcomeMessage = new
            SpeechSynthesisUtterance('welcome games');
         window.speechSynthesis.speak(welcomeMessage);
         }
   playWelcomeMessage();
  var recognition = new webkitSpeechRecognition();
  recognition.continuous = true;
  recognition.lang = 'en-US';
  recognition.onresult = function(event) {
    var result = event.results[event.results.length -
        1][0].transcript.toLowerCase();
    console.log("result",result);
    var currentQuestionIndex=0;
     if (result.includes('back to home')) {
        // Redirect to the home page
        window.location.href = '{% url "sh" %}';
    }
};
```

```
 recognition.onerror = function(event) {
      console.error('Speech recognition error:', event.
          error);
  };
  recognition.onend = function() {
      // Restart recognition when it ends
      recognition.start();
  };
  // Start speech recognition
  recognition.start();
  </script>
</body>
</html>
```

views.py

```
from typing import Any
from django.forms.models import BaseModelForm
from django.http import HttpResponse
from django.shortcuts import render,redirect,
    HttpResponseRedirect
from .models import *
from .models import Question
from .forms import *
from django.views.generic import FormView,CreateView,
    UpdateView,TemplateView,View
from django.contrib.auth import authenticate,login,logout
from django.urls import reverse_lazy
from django.contrib.auth.hashers import make_password
from django.http import FileResponse
from django.shortcuts import get_object_or_404
from .models import Suggestion
from student_app.forms import ChangePasswordForm
from django.forms import formset_factory
# Create your views here.

import pandas as pd
import random
from django.db.models.signals import post_save
from django.dispatch import receiver
from django.contrib.auth import get_user_model
from django.http import JsonResponse

class MainHome(TemplateView):
    template_name="mainhome.html"
```

```
class ClearDataView(View):
    def post(self, request):
        StudentAnswer.objects.all().delete()
        StudentAnswerImage.objects.all().delete()
        StudentAnswerAudio.objects.all().delete()
        ScoreModel.objects.all().delete()
        return JsonResponse({'message': 'Data cleared
            successfully'})
@receiver(post_save, sender=Student)
def create_user_from_student(sender, instance, created,
    **kwargs):
    if created:
        # User = get_user_model()
        CustUser= get_user_model()
        student_id_id=instance.id
        username = instance.std_id
        password = 'admin@123'  # Set your desired
            default password here
        # User.objects.create_user(username=username,
            password=password)
        CustUser.objects.create_user(username=username,
            password=password,student_id_id=student_id_id)
        # ScoreModel.objects.create(student_id=
            student_id_id)
class LoginView(FormView):
    template_name="login.html"
    form_class=LogForm
    def post(self,request,*args,**kwargs):
        log_form=LogForm(data=request.POST)
        if log_form.is_valid():
            us=log_form.cleaned_data.get('username')
            ps=log_form.cleaned_data.get('password')
            user=authenticate(request,username=us,
                password=ps)
            if user:
                login(request,user)
                if request.user.is_superuser == 1:
                    return redirect('h')
                else:
                    return redirect('sh')
            else:
                return render(request,'login.html',{"form
                    ":log_form})
        else:
            return render(request,'login.html',{"form":
                log_form})
```

```
class AddStudent(CreateView):
    template_name='addstudent.html'
    model=Student
    form_class=StudentForm
    success_url=reverse_lazy('stu')
class QuestView(TemplateView):
    template_name='test.html'
    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)
        id=kwargs.get('pk')
        context['stu']=Student.objects.get(id=id)
        context['ques']=Question.objects.all()
        return context
# class QuestViewAll(TemplateView):
#     template_name='Assignexam.html'
#     def get_context_data(self, **kwargs):
#         context = super().get_context_data(**kwargs)
#         context['stu']=Student.objects.all()
#         context['ques'] = Question.objects.all()
#         return context

# class QuestViewImageAll(TemplateView):
#     template_name='testallimage.html'
#     def get_context_data(self, **kwargs):
#         context = super().get_context_data(**kwargs)
#         context['stu']=Student.objects.all()
#         context['ques'] = QuestionImages.objects.all()
#         return context

# class QuestViewAudioAll(TemplateView):
#     template_name='testallaudio.html'
#     def get_context_data(self, **kwargs):
#         context = super().get_context_data(**kwargs)
#         context['stu']=Student.objects.all()
#         context['ques'] = QuestionAudio.objects.all()
#         return context

def Test(request,**kwargs):
    if request.method == 'POST':
        id=kwargs.get('pk')
        stu=Student.objects.get(id=id)
        que=Question.objects.all()
        assignment = StudentAnswer(student=stu, question=
            que)
        assignment.save()
        return redirect('det')
```

```
from django.core.mail import send_mail

def AssignView(request,**kwargs):
    if request.method == 'POST':
        # Get all available tests
        tests = Question.objects.all()

        # Get all students
        students = Student.objects.filter(disability__in
            =["Mobility Impairment", "Learning Disability
            ", "Autism Spectrum Disorder", "Speech
            Impairment", "Intellectual Disability"])
        students_with_email_sent = set()
        # Assign all tests to all students
        for test in tests:
            for student in students:

            assignment, created = StudentAnswer.objects.
                get_or_create(student=student, question=
                test)
             if created:
            assignment.save()
            subject = 'Test Assignment'
            message = f'You have been assigned the test.
                Check your dashboard for details.'
            from_email = 'testhelloability@gmail.com'
            to_email = [student.email]
            if student in students_with_email_sent:
                      continue
             send_mail(subject, message, from_email,
                 to_email, fail_silently=False)
             students_with_email_sent.add(student)
        return redirect('testall')

    # Retrieve all available tests
    tests = Question.objects.all()

    return render(request, 'Assignexam.html', {'tests':
        tests})
def AssignImageView(request,**kwargs):
    if request.method == 'POST':
        # Get all available tests

        tests = QuestionImages.objects.all()
```

```
        # Get all students
        students = Student.objects.filter(disability="
            Hearing Impairment")
        students_with_email_sent = set()
        # Assign all tests to all students
        for test in tests:
            for student in students:

                assignment, created = StudentAnswerImage.
                    objects.get_or_create(student=student,
                     question=test)
                if created:
                 assignment.save()
                 subject = 'Test Assignment'
                 message = f'You have been assigned the
                     test. Check your dashboard for
                     details.'
                 from_email = 'testhelloability@gmail.com
                     '
                 to_email = [student.email]
                 if student in students_with_email_sent:
                  continue
                 send_mail(subject, message, from_email,
                     to_email, fail_silently=False)
                 students_with_email_sent.add(student)

        return redirect('testallvisual')

    # Retrieve all available tests
    tests = QuestionImages.objects.all()

    return render(request, 'testallimage.html', {'tests':
        tests})
def AssignAudioView(request,**kwargs):
    if request.method == 'POST':
        # Get all available tests
        tests = QuestionAudio.objects.all()

        # Get all students
        students = Student.objects.filter(disability="
            Visual Impairment")
        students_with_email_sent = set()

        # Assign all tests to all students
        for test in tests:
            for student in students:
```

```
                    assignment, created = StudentAnswerAudio.
                        objects.get_or_create(student=student,
                         question=test)
                    if created:
                     assignment.save()
                     subject = 'Test Assignment'
                     message = 'You have been assigned the
                         test. Check your dashboard for
                         details.'
                     from_email = 'donkannukkadan@gmail.com'
                     to_email = [student.email]
                     if student in students_with_email_sent:
                       continue
                     # Send email
                      send_mail(subject, message, from_email,
                           to_email, fail_silently=False)

                     # Add the student to the set of students
                          with sent emails
                      students_with_email_sent.add(student)

            return redirect('testallhear')

        # Retrieve all available tests
        tests = QuestionAudio.objects.all()

        return render(request, 'testallaudio.html', {'tests':
            tests})

from random import sample

class Quesadd(CreateView):
    template_name="quesadd.html"
    model=Question
    form_class=QuesForm
    success_url=reverse_lazy('qans')
    def get_context_data(self, **kwargs) :
        context = super().get_context_data(**kwargs)
        context['tests'] = Question.objects.all()
        return context

class Quesimage(CreateView):
    template_name="quesimage.html"
    model=QuestionImages
    form_class=QuesFormImage
    success_url=reverse_lazy('qansimg')
```

```
    def get_context_data(self, **kwargs) :
        context = super().get_context_data(**kwargs)
        context['tests'] = QuestionImages.objects.all()
        return context

class Quesaudio(CreateView):
    template_name="quesaudio.html"
    model=QuestionAudio
    form_class=QuesFormAudio
    success_url=reverse_lazy('qansaudio')
    def get_context_data(self, **kwargs) :
        context = super().get_context_data(**kwargs)
        context['tests'] = QuestionAudio.objects.all()
        return context

class QuesUpdate(UpdateView):
    template_name="questionupdate.html"
    model=Question
    form_class=QuesForm
    success_url=reverse_lazy('qdel')
    def get_context_data(self, **kwargs) :
        context = super().get_context_data(**kwargs)
        context['tests']=Question.objects.all()
        return context

class QuesAnsUpdView(CreateView):
    template_name="quesansupdate.html"
    model=Answer
    form_class=QuesAnsForm
    success_url=reverse_lazy('qdel')
    def form_valid(self, form):
        # Save the main answer
        main_answer = form.save(commit=False)
        main_answer.save()

        # Get additional options from the form data
        option_texts = [self.request.POST.get(f'option_{i
            }') for i in range(1, 4)]

        # Create and save additional options
        for option_text in option_texts:
            if option_text:
                answer_option = Answer(question=form.
                    cleaned_data['question'], text=
                    option_text)
                answer_option.save()
```

```
        return super().form_valid(form)

class QuesAnsImageView(CreateView):
    template_name = "quesansimage.html"
    model = AnswerImages
    form_class = QuesAnsFormImg
    success_url = reverse_lazy('qimg')

    def form_valid(self, form):
        # Save the main answer
        main_answer = form.save(commit=False)
        main_answer.save()

        # Handle file uploads for options
        for i in range(1, 4):
            option_file = self.request.FILES.get(f'
                option_{i}')
            if option_file:
                AnswerImages.objects.create(
                    question=main_answer.question,
                    fileans=option_file
                )
        return super().form_valid(form)

class QuesAnsAudioView(CreateView):
    template_name="quesansaudio.html"
    model=AnswerAudio
    form_class=QuesAnsFormAudio
    success_url=reverse_lazy('qaudio')
    def form_valid(self, form):
        # Save the main answer
        main_answer = form.save(commit=False)
        main_answer.save()

        # Handle file uploads for options
        for i in range(1, 4):
            option_file = self.request.FILES.get(f'
                option_{i}')
            if option_file:
                AnswerAudio.objects.create(
                    question=main_answer.question,
                    fileans=option_file
                )

        return super().form_valid(form)
```

```
class DeleteView(View):
    def get(self,req,*args,**kwargs):
        id=kwargs.get('pk')
        dl=Question.objects.get(id=id)
        dl.delete()
        return redirect('qdel')

class DeleteImgView(View):
    def get(self,req,*args,**kwargs):
        id=kwargs.get('pk')
        dl=QuestionImages.objects.get(id=id)
        dl.delete()
        return redirect('qimg')

class DeleteAudioView(View):
    def get(self,req,*args,**kwargs):
        id=kwargs.get('pk')
        dl=QuestionAudio.objects.get(id=id)
        dl.delete()
        return redirect('qaudio')

# class Quesdel(TemplateView):
#     template_name="quesdel.html"
#     def get_context_data(self, **kwargs) :
#         context = super().get_context_data(**kwargs)
#         context['tests']=Question.objects.all()
#         return context

class SugView(TemplateView):
    template_name="suggestions.html"
    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)
        context['data']=Suggestion.objects.all().order_by
            ('cat')
        return context

class SuggTextView(CreateView):
    template_name="suggtext.html"
    model=Suggestion
    form_class=SugForm
    success_url=reverse_lazy('st')

class SuggVideoView(CreateView):
    template_name="suggvideo.html"
    model=Suggestion
```

```
    form_class=SugVideoForm
    success_url=reverse_lazy('sv')

class SuggAudioView(CreateView):
    template_name="suggaudio.html"
    model=Suggestion
    form_class=SugAudioForm
    success_url=reverse_lazy('sadd')

def view_video(request, video_id):
    video = get_object_or_404(Suggestion, pk=video_id)
    video_path = video.video.path
    response = FileResponse(open(video_path, 'rb')) #
        Adjust content_type as needed
    return response

def view_videoo(request, videoo_id):
    video = get_object_or_404(ScoreModel, pk=videoo_id)
    video_path = video.video.path
    response = FileResponse(open(video_path, 'rb')) #
        Adjust content_type as needed
    return response

def play_audio(request, audio_id):
    audio_recording = get_object_or_404(Suggestion, pk=
        audio_id)
    audio_file = audio_recording.audio
    response = FileResponse(open(audio_file.path, 'rb'))
    return response

def play_audioo(request, audio_id):
    audio_recording = get_object_or_404(ScoreModel, pk=
        audio_id)
    audio_file = audio_recording.audio
    response = FileResponse(open(audio_file.path, 'rb'))
    return response

class DeleteViewSug(View):
    def get(self,req,*args,**kwargs):
        id=kwargs.get('pk')
        dl=Suggestion.objects.get(id=id)
        dl.delete()
        return redirect('sadd')

class ChangePasswordViewHome(FormView):
    template_name="changepshome.html"
```

```
    form_class=ChangePasswordForm
    def post(self,request,*args,**kwargs):
        form_data=ChangePasswordForm(data=request.POST)
        if form_data.is_valid():
            current=form_data.cleaned_data.get("
                current_password")
            new=form_data.cleaned_data.get("new_password
                ")
            confirm=form_data.cleaned_data.get("
                confirm_password")
            user=authenticate(request,username=request.
                user.username,password=current)
            if user:
                if new==confirm:
                    user.set_password(new)
                    user.save()
                    logout(request)
                    return redirect("log")
                else:
                    return redirect("cp")
            else:
                return redirect("cp")
        else:
            return render(request,"changepassword.html
                ",{"form":form_data})

class SuggestionUpdateView(UpdateView):
    template_name='suggestionupdate.html'
    model=Suggestion
    form_class=SuggestionForm
    success_url=reverse_lazy('sadd')
```

VENTURE VISTA

PROJECT REPORT

Submitted By

MARIA JOSEPH

Reg. No. CCAVBCA043

for the award of the Degree of

Bachelor of Computer Application (B.C.A.)

(University of Calicut)

under the guidance of

Ms. Soumya P.S

Assistant Professor

Bachelor of Computer Application
DEPARTMENT OF COMPUTER SCIENCE
CHRIST COLLEGE(Autonomous)
IRINJALAKUDA, KERALA
INDIA

2021-24

DEPARTMENT OF COMPUTER SCIENCE

CHRIST COLLEGE (AUTONOMOUS)

IRINJALAKUDA

CERTIFICATE

This is to certify that the project report entitled ***"VENTURE VISTA"*** *is a bonafide record of the project work done by* ***MARIA JOSEPH*** *in partial fulfillment of the requirement for the sixth semester of* ***Bachelor of Computer Application*** *in Department of Computer Science of* ***CHRIST COLLEGE (AUTONOMOUS) IRINJALAKUDA***

.

Ms. SOUMYA P.S
Assistant Professor,CS
Internal Guide

Ms. SINI THOMAS
Head of the Department
Computer Science

EXTERNAL EXAMINER

INTERNAL EXAMINER

DECLARATION

We hereby declare that this project work **"VENTURE VISTA"** submitted to Christ College (Autonomous) Irinjalakuda, affiliated to Calicut University in partial fulfillment of the requirement for the award of the Bachelor of Computer Application, is a record of original work done by us, under the guidance of Ms. SOUMYA P.S, Department of computer Science.

Place: Irinjalakuda MARIA JOSEPH

ACKNOWLEDGEMENT

First and foremost I like to thank Lord almighty for his providence and for being the guiding light throughout the project. I wish to express my sincere gratitude to my beloved Department head for giving me all the facilities for my project. I take this opportunity to express my gratitude to the class teacher Ms. SOUMYA P.S and head of the department Ms. SINI THOMAS who has been supported us throughout the course of this project. I are thankful for her aspiring guidance and valuable advice during the project work. I express my sincere thanks to my project guide Ms. SOUMYA P.S for supporting and guiding throughout the project.I would take this opportunity to specially thank all other faculty members for their constant and continuous motivation. Finally I would like to thank my family and friends for giving valuable advice and moral support throughout my project.

ABSTRACT

VENTURE VISTA Our travel guide website aims to inspire and assist travelers in exploring new destinations worldwide. From detailed city guides to off-the-beaten-path adventures, we provide a wealth of information to help users plan unforgettable trips. With curated recommendations for accommodations, dining, activities, and transportation, our platform caters to all types of travelers, security scores,live money exchange rates. By combining expert insights with user reviews and ratings, we strive to be a one-stop resource for discovering and planning your next adventure.

Contents

Chapter 1

1 Introduction

This is an web application that helps the users to know more about the major attractions near to them. The application provides all the details about tourist places or major attractions like their opening time, closing time, history and other details. The app also provides all the details of the hotels available for the users. The app can also provide the best route to the places you have selected. The app will come in handy for those who travel a lot, or like to travel to new places.

1.1 Overview

The objective of the TRAVEL GUIDE APPLICATION is to design an simple and adaptable application that helps the users to know more about the major attractions near to them. The web application includes many interesting features such as providing place suggestions to visit which best suites for the preferences you have given.It is a User friendly application.

Chapter 2

2 System Analysis

2.1 Purpose

A travel guide application serves as a comprehensive resource for travelers, offering a wide range of information and features to enhance their travel experiences. Its primary purpose is to provide users with detailed insights into various destinations, including popular attractions, historical sites, cultural experiences, dining options, accommodations, and local services.

By leveraging the app, users can efficiently plan their trips, exploring different destinations based on their interests and preferences. They can access detailed descriptions, reviews, and ratings for various attractions and services, helping them make informed decisions about their travel itinerary.

Furthermore, a travel guide application often includes features such as offline maps, itinerary planners, currency converters, and language translators, which can be invaluable for travelers navigating unfamiliar environments. These features can help users navigate their destinations more effectively, communicate with locals, and manage their travel budgets.

Overall, a travel guide application aims to simplify the travel planning process, inspire users to explore new destinations, and enhance their overall travel experiences by providing them with relevant and reliable information.
.

2.1.1 Existing System

The existing system are apps that provides data about tour packages and so on. Apps like this, which provide all details of the places are not available. Limitations of existing system :We cannot get all the details about the places we are looking for from a single application.

2.1.2 Proposed System

The proposed system is an web application which provides all the details related to the places you are looking for your next trip. We also provide a feature that gives you suggestions of the places near you to visit, based on the preferences you have provided. Advantages of proposed system :The application gives all the details about the places you are looking for.It also

provide place suggestions to visit which best suites for the preferences you have given.

2.2 Problem definition

To know what the problem is and what the needs are before developing it. Here we design an application for travel guiding.

2.3 FEASIBILITY STUDY

After the problem is clearly understood and solutions proposed, the next step is to conduct the feasibility study. Feasibility study is defined as evaluation or analysis of the potential impact of the proposed project or program. The objective is to determine whether the proposed system is feasible. There are three aspects of feasibility study to which the proposed system is subjected as discussed below.

2.3.1 Technical Feasibility

Technical feasibility assess whether the current technical resources are sufficient for the new system. We can upgrade the level of technology for supporting our website. We check whether the proposed system can be implemented in the present system without supporting the existing hardware.

2.3.2 Economical Feasibility

Economic feasibility determines whether the time and money are available to develop the system. There is no additional hardware used to develop the site, it is inexpensive to build.

2.3.3 Operational Feasibility

Operational feasibility determines if the human resources are available to operate the system once it has been installed. No extra training is needed to use this system. Anyone who has knowledge in internet in English language can easily use the system. The resources that are required to implement or install are already available with the organization.

Chapter 3

3 Software Requirement Specification

3.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the TRAVEL GUIDE APPLICATION. It illustrate the purpose and complete description for the development of the system. It explain system constraints, interface and interactions with other external applications. This document is primarily intended to be proposed to provide all the details related to the places you are looking for your next trip.

3.2 Scope

The application gives all the details about the places you are looking for.It also provide place suggestions to visit which best suites for the preferences you have given.

3.3 Overall Description

This section give an overview of our application, TRAVEL GUIDE APPLICATION. This is an web application that helps the users to know more about the major attractions near to them. The application provides all the details about tourist places or major attractions like their opening time, closing time, history and other details. The app also provides all the details of the hotels available for the users. The app can also provide the best route to the places you have selected. The app will come in handy for those who travel a lot, or like to travel to new places.

3.3.1 Product Perspective

TRAVEL GUIDE APPLICATION is mainly used to know the details about your next trip. It also provide features that gives you suggestions to visit according to the preferences you have given.

3.3.2 Product Functionality

Through this system admin can upload various data. User can view the major attractions and their history and other details available in an area.

Users can also view all the information about the hotels available in a specific area.

3.3.3 Users and Characteristics

There are two types of users that interact with the application, people who wish to travel and the guide. Each of these have different tasks which is performed. Admin is able to upload details of places and can view the feedback.

3.4 Specific Requirements

3.4.1 Hardware Requirements

- System: IBM-Compatible PC
- Processor: Intel Core i3
- Speed: Above 1GHz
- RAM capacity: 4 GB
- Hard Dsk drive: 500 GB

3.4.2 Software Requirements

- Front End : React
- Back End : Djanko Python
- Database : MySql

3.5 Functional Requirements

It contains three main modules.

- 1.Admin
- 2.User

Admin

The admin will upload the details of the places. All the details that are provided in the app is uploaded by the admin. The admin can view user's feedbacks and can take suitable decisions based on the user's decisions.

User

The users can view the major attractions and their history and other details available in an area. Users can also view all the information about the hotels available in a specific area. The users can get the best route available to the selected place from their current location, which is calculated by the app and shown in the map. Users can also get notifications on the places that are suggested by the app based on the preferences they have given.

3.6 Non Functional Requirements

Non-functional requirements define the overall qualities or attributes of the resulting system. Non-functional system place restrictions on the product being developed, the developed process, and specify external constraints that the product must meet. Examples of non-functional requirements include safety, security, usability, reliability and performance requirements. Project management issues (costs, time and schedule) are often considered as non-functional requirements. The principal non - functional constraints which are relevant to critical systems :

- performance
- security
- safety
- usability

Performance

Performance requirements concern the speed of operation of a system. Types of performance requirements :

- Response requirements (how quickly the system reacts to a user input).
- Throughput requirements (how much the system can accomplish within a specified amount of time).
- Availability requirements (is the system available for service when requested by end users). The speed of operation of this system is adequate for the requirements.

Reliability

- Reliability is the ability of a system to perform its required function under stated conditions for a specified period of time.

- constraints on the runtime behavior of the system. This system is reliable because its functionalities can be done on the required conditions.

Safety

Safety requirements are not required which exclude unsafe situation from the possible solution of the system.

Usability

Usability is the ease with which a user can learn to operate, prepare inputs for, and interpret outputs of system or components. Usability requirements include :

- information error messages.
- well-formed user interfaces.

3.7 Interface Requirements

3.7.1 Hardware interfaces

The system must run over the internet, all the hardware shall require to connect internet will be hardware interface for the system. As for example modem, WAN-LAN.

3.7.2 Software interfaces

Software interface required for the working of the project is the appropriate operating system.

3.7.3 Communication interfaces

The user of the site communicate using the network connectivity and the data set is accessed

3.8 Security Requirements

- User accesses only their account.
- Validation of input is handled.
- This application containing the computer systems is physically secured against arms or surreptitious entry by intruders.
- Users must be authorized carefully to reduce changes of any such user giving access to an intruder in exchange for a bribe or other favour.

3.9 Platform Used

Windows 10 is a major version of the Microsoft Windows operating system that was released on July 29, 2015.It is built on the Windows NT kernel and follows windows 8.Part of the reason Microsoft decided to name the 2015 release "Windows 10"(and skipped "windows 9")is because the operating system is designed to be a new direction for Microsoft. One of the primary aims of windows 10 is to Unify the windows experience across multiple devices, such desktop computers, tablets, and smartphones. As part of this effort, Microsoft developed Windows10 Mobile alongside Windows 10 to replaces Windows Phone - Microsoft's previous mobile OS. Windows 10 also integrates other Microsoft services, such as Xbox Live and the Cortana voice recognition assistant.

3.10 Technologies Used

Djanko

Django is a free and open-source web framework, written in Python, that follows the model-template-views (MTV) architectural pattern. It is used to build database-driven websites. Django takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel.

React

React.js is an open-source JavaScript library used to build user interfaces (UIs) for single-page applications. React manages the view layer and is used for the development of both web and mobile applications.React is a component-based library, which means that UIs are built by combining reusable components. Components are self-contained pieces of code that can be used to render HTML, CSS, and JavaScript.React uses a virtual DOM, which is a lightweight representation of the real DOM. The virtual DOM is used to efficiently update the UI when data changes.React is a popular choice for building UIs because it is fast, scalable, and easy to learn.

MySQL

MySQL, pronounced either "My S-Q-L" or "My Squel, " is an open source relational database management system. It is based on the structure query

language (SQL), which is used for adding, removing and modifying information in the database. Standard SQL commands, such as ADD, DROP, INSERT,and UPDATE can be used with MySQL.

What distinguishes PHP from something like client - side Javascript is that the code is executed on the server, generating HTML which is then sent to the client. The client would receive the results of running that script, but would not know what the underlying code was. You can even configure your web server to process all your HTML files with PHP, and then there's really no way that users can tell what you have up your sleeve. The best things in using PHP are that it is extremely simple for a newcomer, but offers many advanced features for a professional programmer.

Chapter 4

4 Design Document

4.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the TRAVEL GUIDE APPLICATION.. It illustrate the purpose and complete description for the development of the system. It explain system constraints, interface and interactions with other external applications. This document is primarily intended help the users to know more about the major attractions near to them.This section give an overview of our system, "9 to 5" APPLICATION. The application provides all the details about tourist places or major attractions like their opening time, closing time, history and other details. The app also provides all the details of the hotels available for the users. The app can also provide the best route to the places you have selected. The app will come in handy for those who travel a lot, or like to travel to new places.

4.2 Scope

The application gives all the details about the places you are looking for.It also provide place suggestions to visit which best suites for the preferences you have given.

4.3 Overview

The purpose of this document is to help the reader to visualize the solution to the project presented.This document verifies how the design meet the requirement stimulated in the SRS document through design viewpoints.The design viewpoints will cover all design elements presented before. This document will provide a direct approach to the development of this project hence reducing feature creep and ponitedly determine the quality of the design.

4.4 Data Design

Database are the storehouses of data used in the software systems.The data is stored in tables inside the database.Several tables are created for the manipulation of the data for the system.Two essential settings for a database are

- Primary Key- The field that is unique for all the record occurrence.
- Foreign Key- The field used to set relation between tables.

Normalization

- Normalization is the process of organizing the data in the database.
- Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate the undesirable characteristics like Insertion, Update and Deletion Anomalies.
- Normalization divides the larger table into the smaller table and links them using relationship.
- The normal form is used to reduce redundancy from the database table.

Chapter 5

5 Development of the System

The system is an web application which provides all the details related to the places you are looking for your next trip. We also provide a feature that gives you suggestions of the places near you to visit, based on the preferences you have provided.

Chapter 6

6 System Testing

Testing is the penultimate step of software development.An elaborate testing of the data is prepared and the system is using the test data,while doing testing,errors are noted and correction is made.The users are trained to operate the developed system.Both hardware and software securities are made to run the developed system successfully.System testing is aimed at ensuring the system works accurately before the live operation commences.

6.1 Test Plan

6.1.1 Scope

This test plan will cover the following testing activities as identified in the testing strategy.

- White Box Testing

White box testing is a testing technique, that examines the program structure and derives test data from the program logic/code. In white box testing, the UI is bypassed. Inputs and outputs of ratings are tested directly at the code level and the results are compared against specifications. It Reveals errors in "hidden" code.

- Black Box Testing

Black - box testing is a method of software testing that examines the functionality of an application based on the specifications. Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end - user would.

- Unit Testing

The module includes four main components to undergo unit testing. It checks the syntax error, logic error and validity of the program correctness. This test will be performed by the developers.

- Integration Testing

After all components pass the unit testing, they get to be tested whether they work correctly when they are running concurrently and communicating to each other. Integration tests exercise an entire subsystem and ensure that a set of components play nicely together.

- Internal Data Testing

We will test the validity of the data before it enters the database to avoid any problems that may face us in the database. We will test the encryption of the personal information of all the users along with the user names and passwords to ensure maximum security of the users privacy.

6.1.2 Software risk issues

In this section, the plan is to test the risk involved in critical issues such as: Difficult to run on visual studio due to large loading time. Some inherent software risks such as complexity exist; also these issues need to be identified. Proper network connection Working of Mysql database

6.1.3 Features to be tested

- Test whether correct user name and password allows you to login.
- Test whether invalid user name and password prevents you from login.
- Test whether there is any connection problem in the Server.
- Test whether the student and admin details are entered correctly.
- Test whether invalid data entry allows saving data successfully.
- Test whether all pages are loaded correctly.
- Test whether watermark is embedded and extracted properly.
- Test whether the provided security works correctly.

6.2 Test consolidation

6.2.1 Test item

The items or features to be tested in the test cases are included in the document. Each and every user input is tested. The present condition of the system is tested. It is checked to make sure that environment is ready for the application to work.

6.2.2 Input specifications

These are the inputs required to execute the test case. All required inputs including data elements and values are given below.

Data	Values
Email	A valid email ID
Password	Combination of characters and numbers
Phone number	10 Digit number

Chapter 7

7 System Implementation and Maintenance

7.1 Implementation

System implementation is the conversion of new system into an operating one which involves creating compatible files, training clients and installing hardware. User training is crucial for minimizing resistance to change and giving chance to prove its worth. Training aids user friendly manuals and healthy screens provide the user with a good start. Software maintenance follows conversion to the extent that changes are necessary to maintain satisfactory operations relative to changes in the user's environment. Maintenance often includes minor enhancements or corrections to the problem that surface late in the systems operations. In the implementation phase, the team builds the components either from scratch or by composition. Given the architecture document meant from the design phase and the requirement document from the analysis phase, the team should build exactly what has been requested, though there is still room for innovation and flexibility. For example, a component may be narrowly designed for this particular system, or the component may be made more general to satisfy a reusability

- Careful planning
- Investigation of system and constraints
- Design the methods to achieve changeover.
- Training the staff in the changed phase.
- Evaluation of change over method.
- The method of implementation and time scale to be adopted are found out initially.

7.2 Maintenance

This phase occurs as a result of deploying the whole system at the end users organization. They will perform the beta testing at the end users and inform to the developers about any needed modification to the application. The customer records all the problems that are encountered during the beta testing and reports these to the developer at regular intervals.

7.2.1 Corrective Maintenance

Even with the best quality assurance activities, it is likely that the customer will uncover defects in the software. Corrective maintenance changes the software to correct the defects. Corrective Maintenance activity may consist of repair, restoration or replacement of equipment. This activity will be result of a regular inspection, which identifies the failure in time for corrective maintenance to be planned and scheduled, then performed during a routine maintenance shutdown.

7.2.2 Adaptive Maintenance

Over time, the original environment(CPU, operating system, business rules, external product characteristics) for which the software was developed is likely to change. Adaptive maintenance results in modification to the software to accommodate changes to its external environment.

7.2.3 Enhanced Maintenance

As software is used, the customer/user will recognize additional functions that will provide the benefit. Perfect maintenance extends the software beyond its original functional requirements.

7.2.4 Preventive Maintenance

Computer software deteriorates due to change, and because of this preventive maintenance often called software re-engineering, must be conducted to enable the software to serve the needs of its end users. Preventive maintenance makes changes to computer programs so that they can be more easily corrected, adapted and enhanced.

Chapter 8

8 Conclusion and Future Scope

8.1 Conclusion

There are two types of users that interact with the system Admin, and the traveler.Each of these two types has different uses of the system so each of them has their own panel.Admin can manage all the features of application dynamically by login on admin panel. The users can view the major attractions and their history and other details available in an area. Users can also view all the information about the hotels available in a specific area.

8.2 Future Scope

- Gives all the details about the places you are looking for.
- Provide place suggestions to visit which best suites for the preferences you have given.

Appendix

A Data Flow Diagram

Data flow is the one of the best way of documenting the entire functionality of the system.For the system ,which will have data flows in and have some processing inside and then some data flow out from the system can be documented or represented effectively by means of data flow out from the system can be documented or represented effectively by means of data flow diagrams. The data flow diagram are a diagrammatic representation of the system,which has input,process and output.Once any system is represented using a data flow diagram we can identify the following things easily:

- Various entities interacting with the system are identified
- Flow of data from one entity to another is identified
- The various processes involved in between the interaction of two or more entities in the system are clearly pointed out
- The various data stores which hold the data in between the process,are clearly identified

Some Data Flow Diagram charting forms:

A.1 External source or receiver

A source or sink is a person or part of organization ,which enters or receives information from the system,but is considered to be outlining the contest of data flow model.

A.2 Transform process

A process represents transformation where incoming data flowsare changed into outgoing data flow

A.3 Data Store

A data store is repository of data that is to be stored for use by one or more process may be as simple as buffer or queue or sophisticated as relational database. They should have clear names.If a process merely uses the contest of store and does not alter it,the arrowhead goes only form the store to the

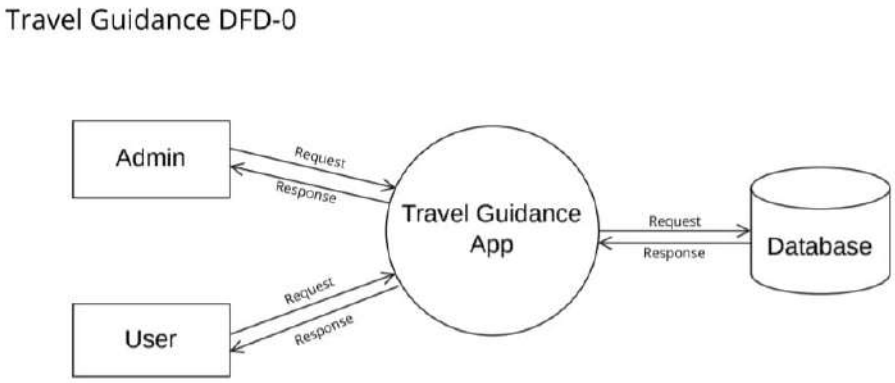
process. If a process alters the details in the store then double-headed arrow is used.

A.4 Data flow

A data flow is a route, which enable packets of data to travel from one point to another.Data may flow,with arrowhead pointing in the direction of the flow

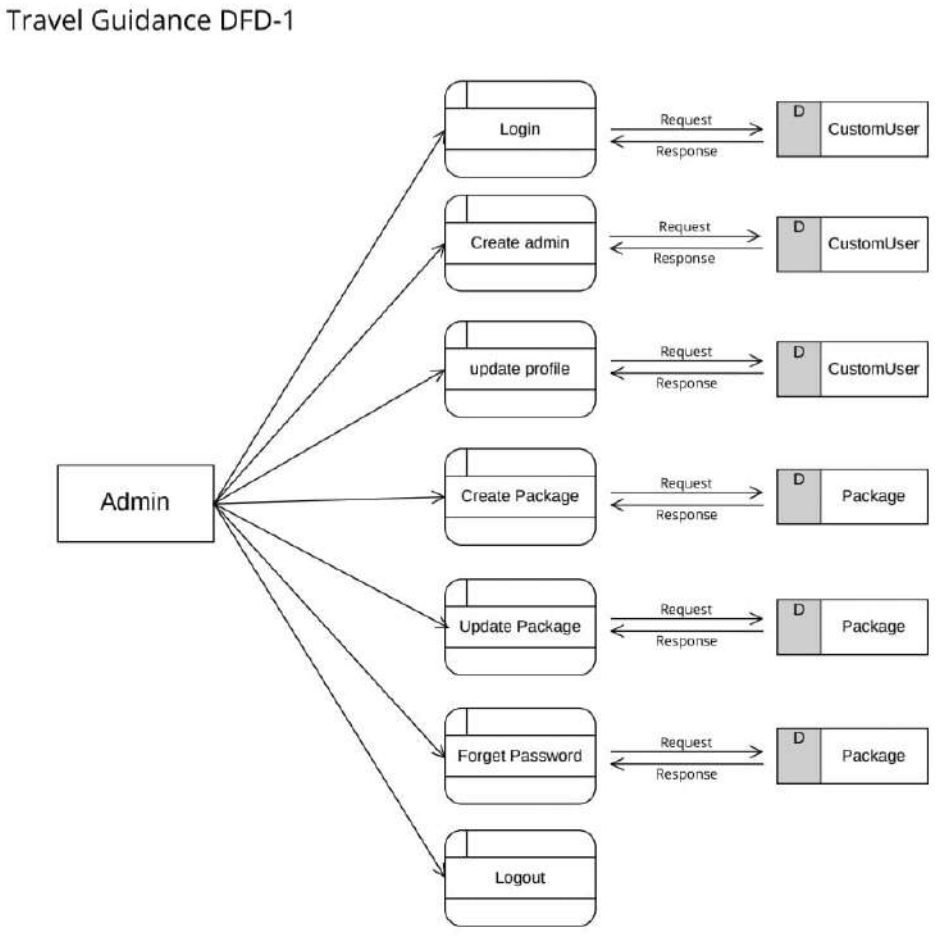
B dfd0

B.1 Level 0



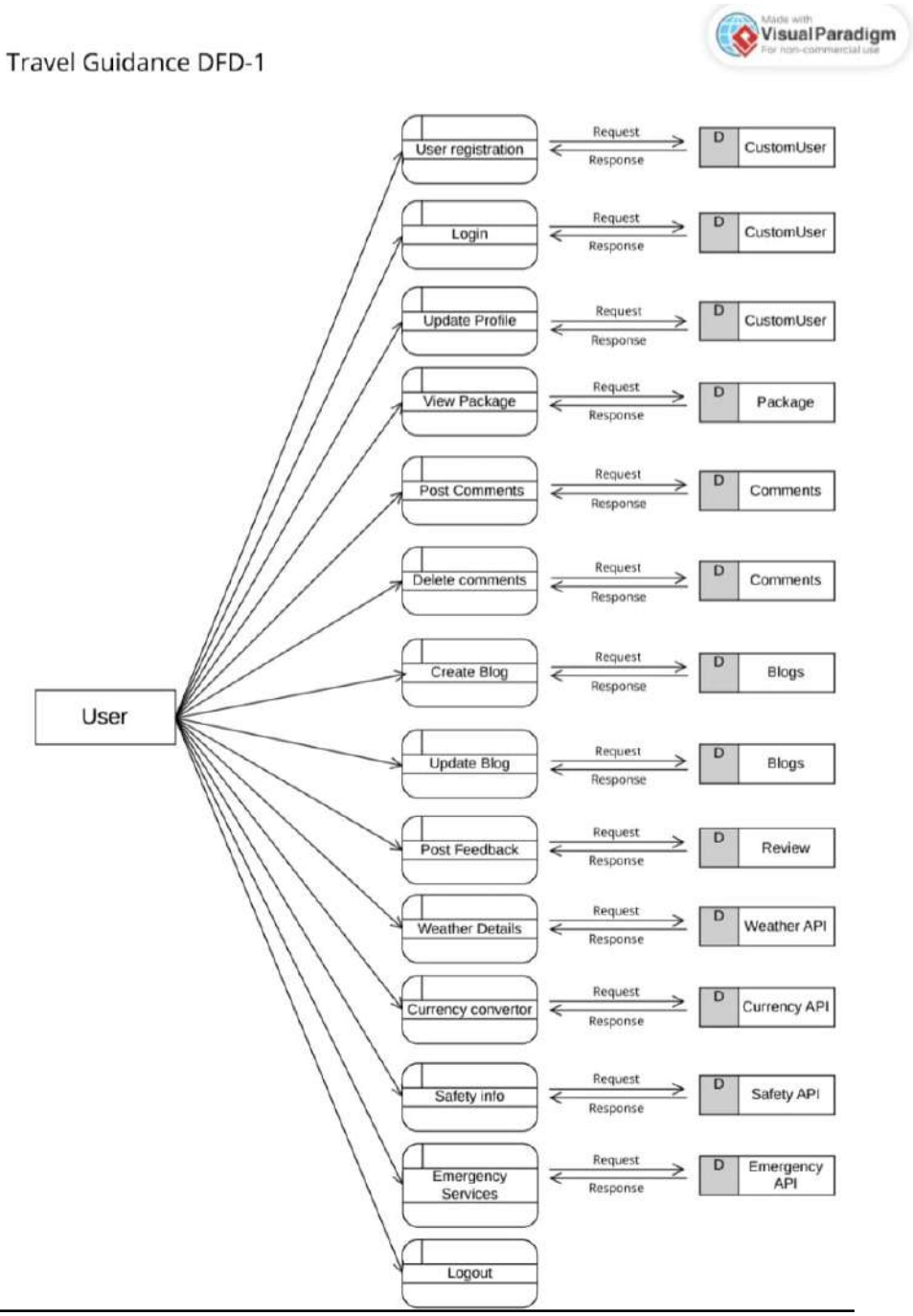
C Data Flow Diagram of Admin

C.1 Level 0

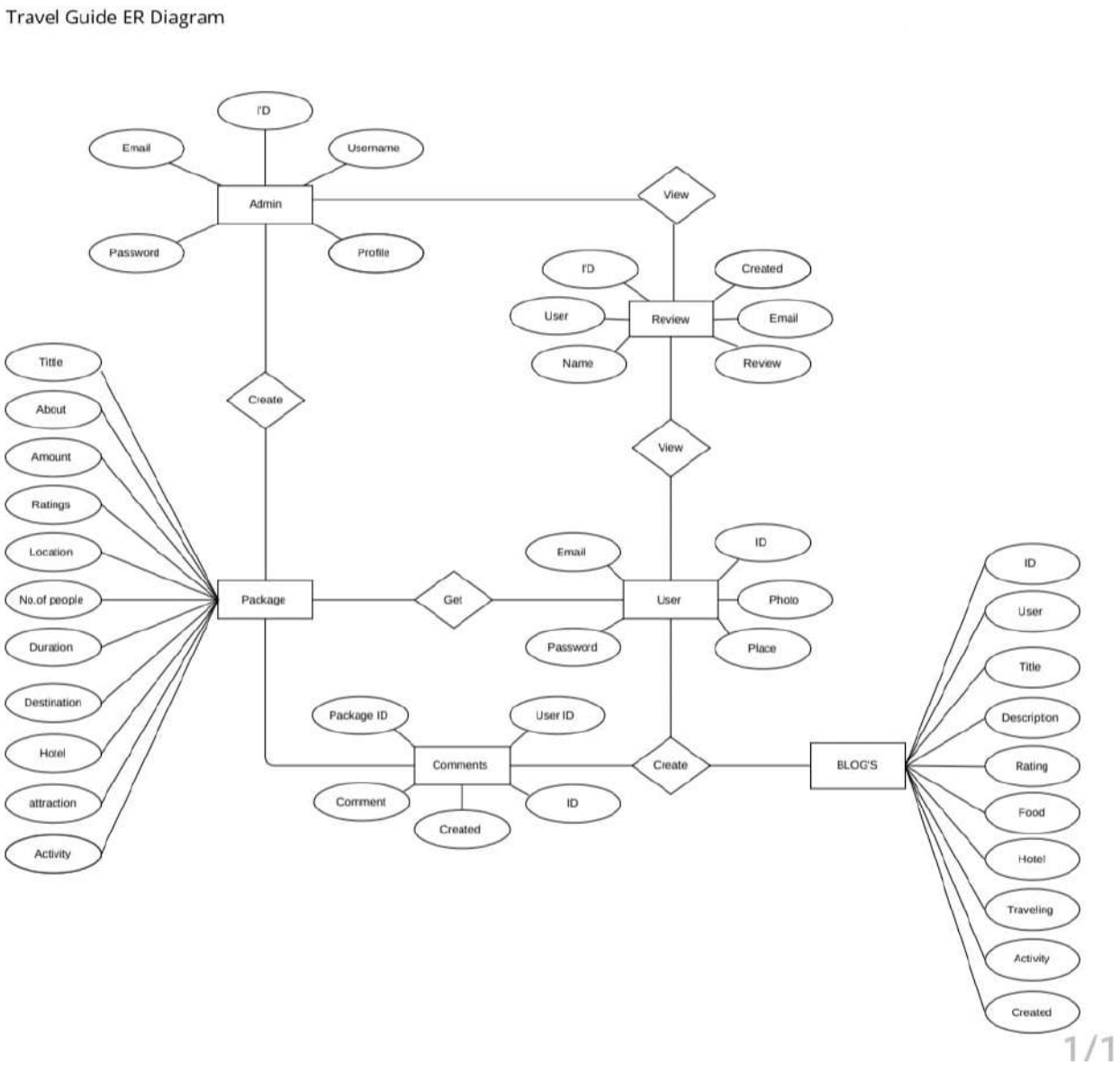


D Data Flow Diagram of User

D.1 Level 0

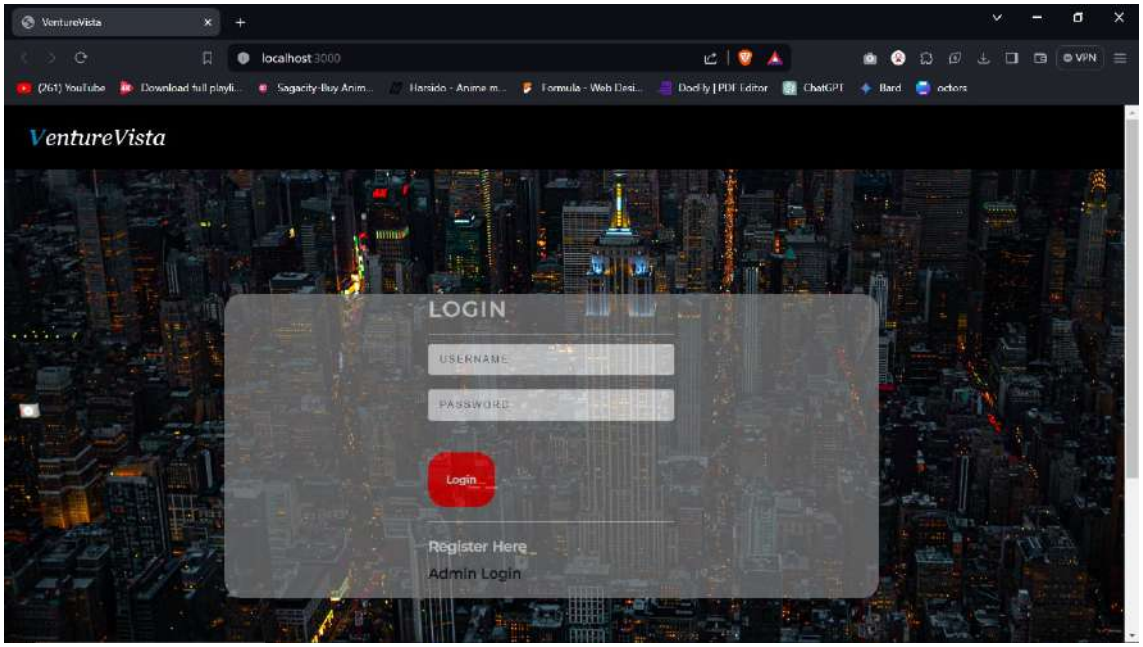


E E-R Diagram

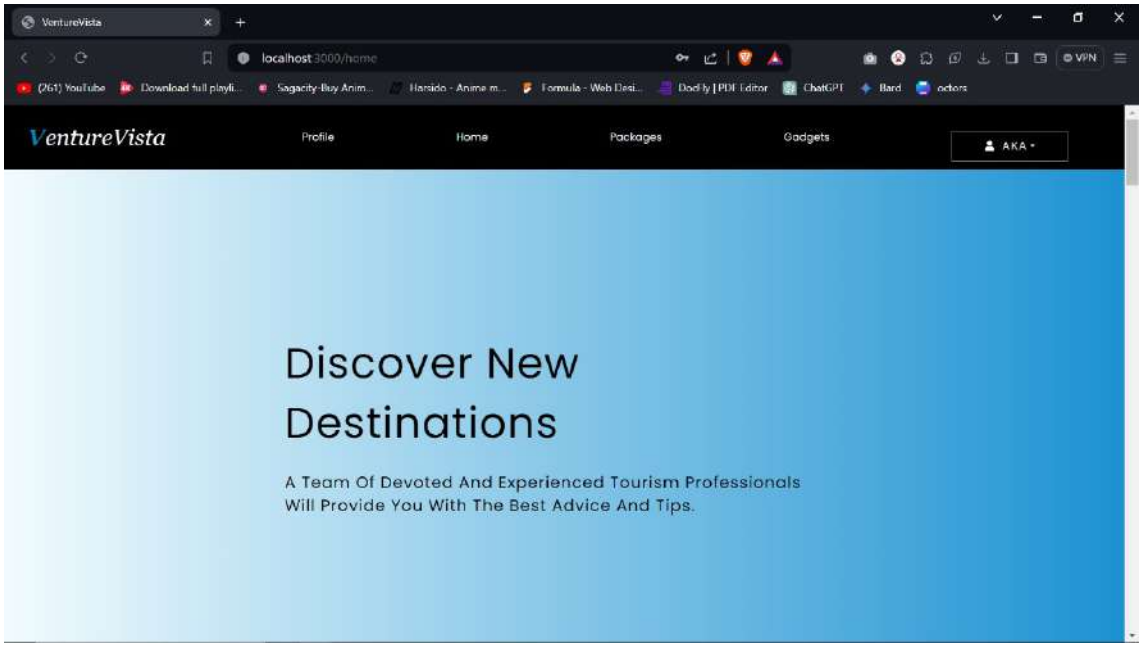

Travel Guide ER Diagram
I'D
Email
Username
Admin
Password
Profile
View
I'D
Created
User
Review
Email
Name
Review
Tittle
About
Amount
Ratings
Location
No.of people
Duration
Destination
Hotel
attraction
Activity
Create
View
Email
ID
Package
Get
User
Photo
Password
Place
Package ID
User ID
Comments
Create
BLOG'S
Comment
ID
Created
ID
User
Title
Description
Rating
Food
Hotel
Traveling
Activity
Created
1/1

F USER INTERFACES

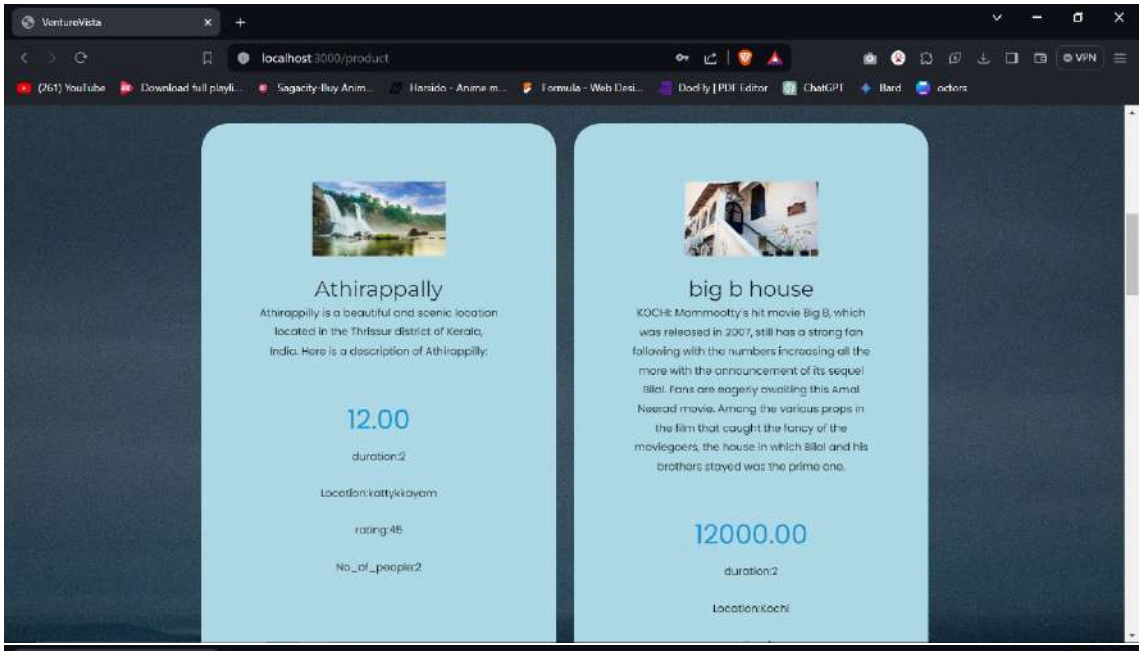
F.1 LOGIN

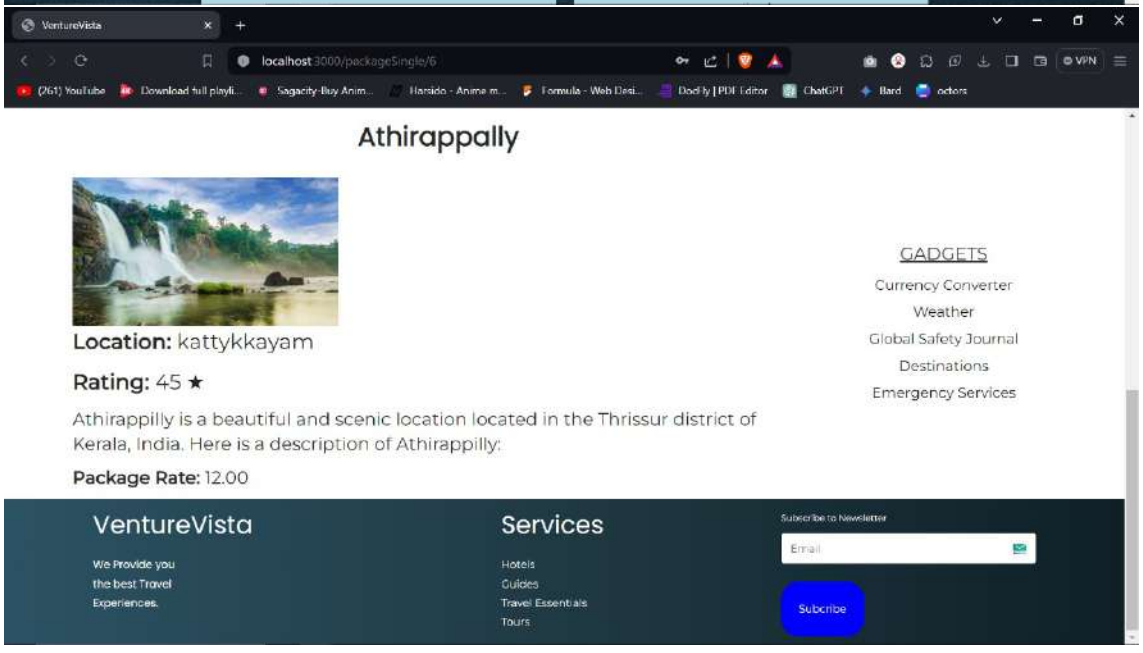


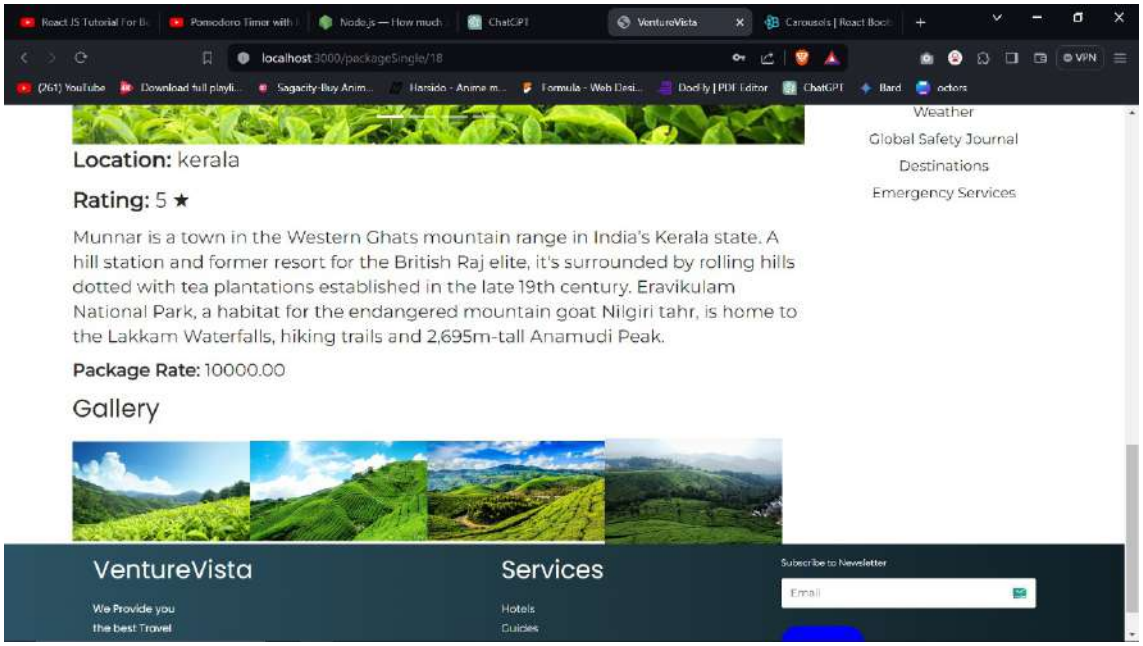
F.2 HOME


VentureVista
localhost:3000/home
VentureVista
Profile
Home
Packages
Gadgets
AKA
Discover New Destinations
A Team Of Devoted And Experienced Tourism Professionals
Will Provide You With The Best Advice And Tips.

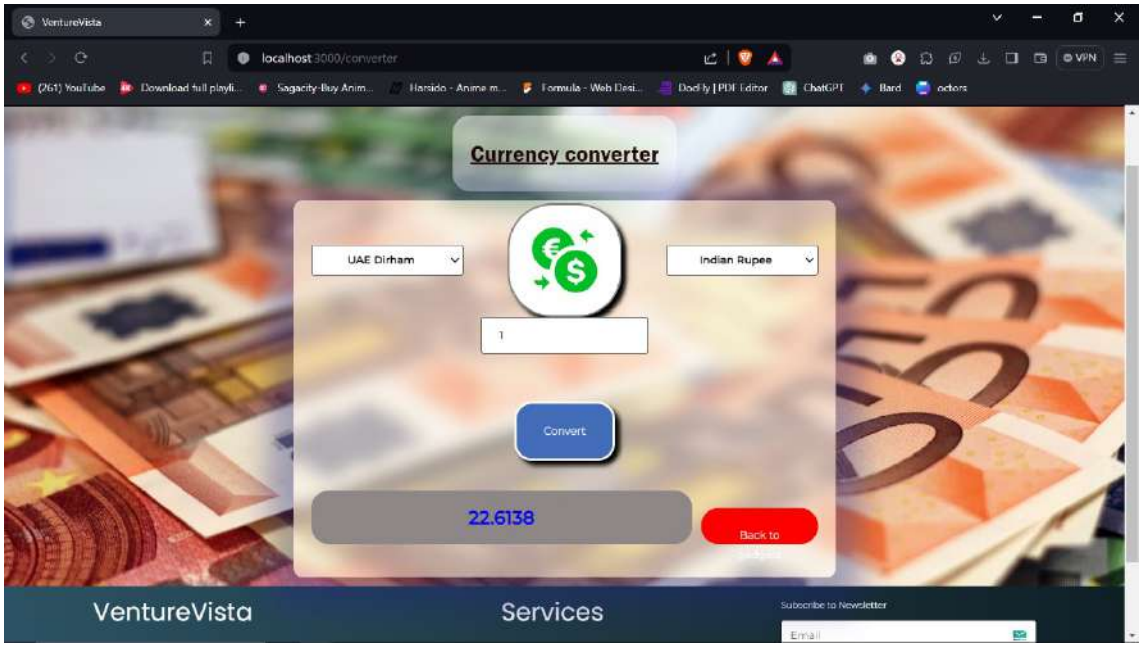
F.3 PACKAGES


VentureVista
localhost:3000/product
Athirappally
Athirappilly is a beautiful and scenic location located in the Thrissur district of Kerala, India. Here is a description of Athirappilly:
12.00
duration:2
Location:kattykkayam
rating:45
No_of_people:2
big b house
KOCHI: Mammootty's hit movie Big B, which was released in 2007, still has a strong fan following with the numbers increasing all the more with the announcement of its sequel Bilal. Fans are eagerly awaiting this Amal Neerad movie. Among the various props in the film that caught the fancy of the moviegoers, the house in which Bilal and his brothers stayed was the prime one.
12000.00
duration:2
Location:Kochi

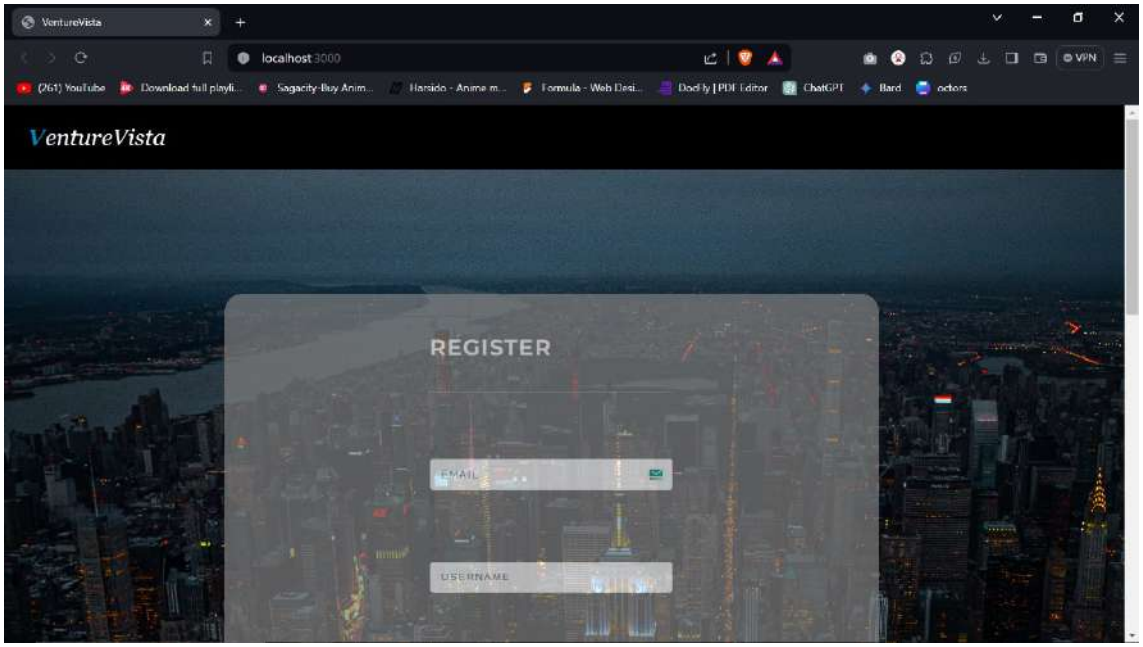

VentureVista
localhost:3000/packageSingle/6
Athirappally
Location: kattykkayam
Rating: 45 ★
Athirappilly is a beautiful and scenic location located in the Thrissur district of Kerala, India. Here is a description of Athirappilly:
Package Rate: 12.00
GADGETS
Currency Converter
Weather
Global Safety Journal
Destinations
Emergency Services
VentureVista
We Provide you the best Travel Experiences.
Services
Hotels
Guides
Travel Essentials
Tours
Subscribe to Newsletter
Email
Subcribe


localhost:3000/packageSingle/18
Weather
Global Safety Journal
Destinations
Emergency Services
Location: kerala
Rating: 5 ★
Munnar is a town in the Western Ghats mountain range in India's Kerala state. A hill station and former resort for the British Raj elite, it's surrounded by rolling hills dotted with tea plantations established in the late 19th century. Eravikulam National Park, a habitat for the endangered mountain goat Nilgiri tahr, is home to the Lakkam Waterfalls, hiking trails and 2,695m-tall Anamudi Peak.
Package Rate: 10000.00
Gallery
VentureVista
We Provide you the best Travel
Services
Hotels
Guides
Subscribe to Newsletter
Email

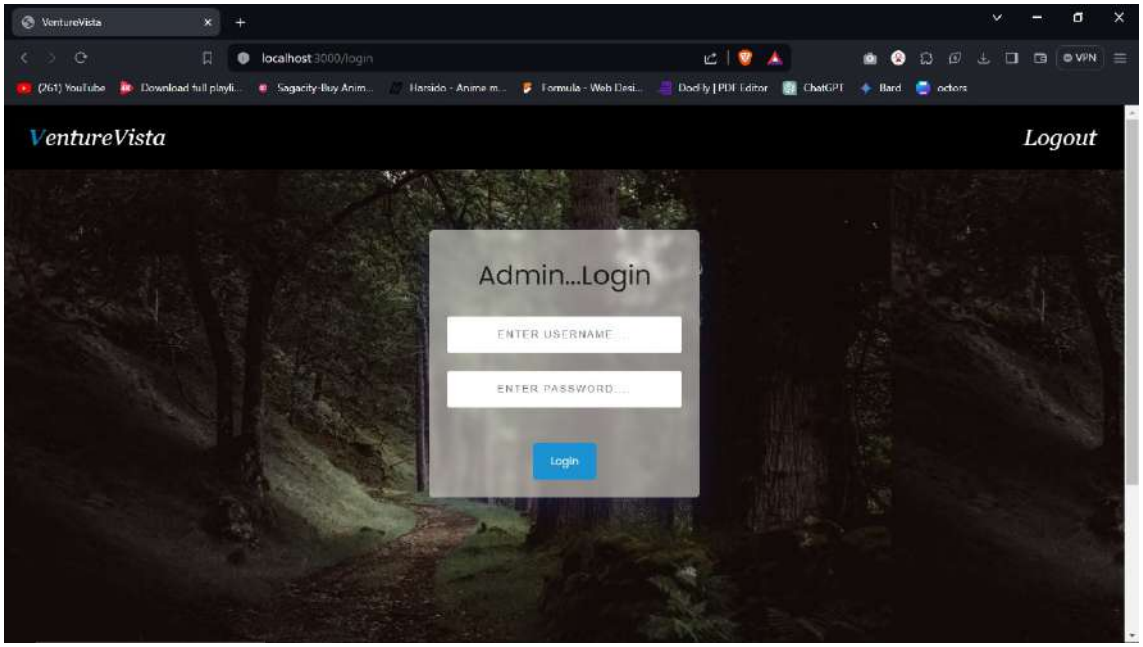
F.4 CURRENCY CONVERTER



F.5 REGISTER


VentureVista
localhost:3000
VentureVista
REGISTER
EMAIL
USERNAME

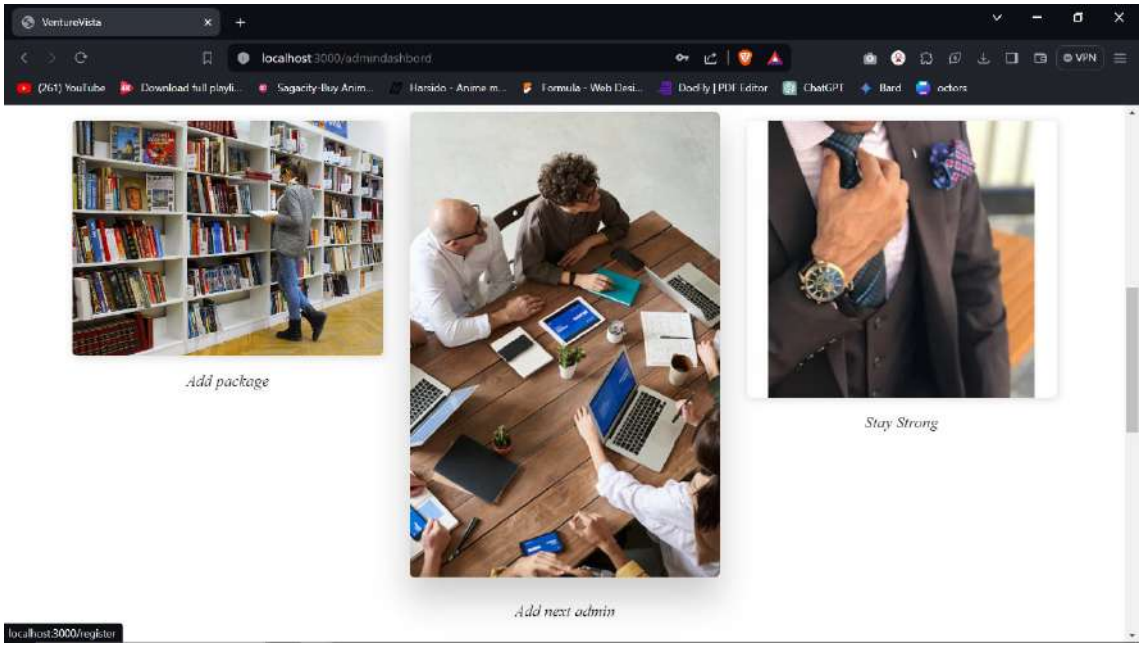
F.6 ADMIN LOGIN



F.7 ADMIN


VentureVista
localhost:3000/admindashboard
VentureVista
Logout
WELCOME MR.ADMIN
THE MAGICAL POWERES OF
YOU ARE....

F.8 ADMIN OPTIONS



G CODE

FRONTEND

```
//
// Source code recreated from a .class
file by
IntelliJ IDEA
// (powered by Fernflower decompiler)
//

import React, { useState, useEffect }
from
"react";
import { Modal, Button, Carousel } from
"react-bootstrap";
import { BrowserRouter, Route, Link }
from
"react-router-dom";
import Image from
"../images/athirappalli.jpeg";
import "./product.css";
function Product({ product }) {
const [show, setShow] = useState(false);
const handleClose = () =>
setShow(false);
const handleShow = () => setShow(true);  console.log(product)

  return (
    <>
      <div>
        <div className="product--card">
<img src={product.attraction}
alt="Product
Shoes" className="product--img" />
          <h4>{product.title}</h4>
          <p>
            {product.about}
          </p>
          <h2>{product.amount}</h2>
```

```
<p>duration:{product.duration}</p><p>Location:{product.location}
</p><p>rating:{product.rating}</p><p>
No_of_people:{product.no_of_people}</p>
        </div>
      </div>
    </>
  );
}

export default Product;
import React from 'react'
import './gadgets.css'
function AdminPowers() {
  return (
<body data-spy="scroll"
data-target=".onpage-navigation"
data-offset="60">
      <main>

<section class="bg-dark-30
showcase-page-header module parallax-bg"id='re' data-background="">
          <div class="titan-caption">
<div class="caption-content"><div class="font-alt mb-30
titan-title-size-1">Our best gadgets to
assits
you</div>
<div class="font-alt mb-40
titan-title-size-4">feel free to rush
trough
them</div>
            </div>
          </div>
        </section>
<div class="main showcase-page"><section class="module-medium"
id="demos"> <div class="container">
<div class="row multi-columns-row"><div
class="col-md-4 col-sm-6 col-xs-12"><a
class="content-box" href="/converter">
<div class="content-box-image"><img
src="https://blog.cdn.cmarix.com/blog/wp-content
```

```
/uploads/2023/07/develop-a-currency-converter-mobile-app.webp"
alt="Currency Converter" /></div>
<h3 class="content-box-title
font-serif">Currency
Converter</h3></a></div>
<div class="col-md-4 col-sm-6
col-xs-12"><a
class="content-box" href="Wheather">
<div class="content-box-image"><img
src="https://play-lh.googleusercontent.com/znc7FD1B
qyFR92b8n6hyROZrAR3FsTyV_ThO79hyuQG-Nb45z2qHFDTnf1HXH50DYg"
alt="Agency" /></div>
<h3 class="content-box-title
font-serif">wheather app</h3></a></div>
<div class="col-md-4 col-sm-6
col-xs-12"><a
class="content-box" href="/journal">
<div class="content-box-image"><img
src="https://s.yimg.com/ny/api/res/1.2/rx7Bjb7IghNsT35t13jufQ--
/YXBwaWQ9aGlnaGxhbmRlcjt3PTY0MDtoPTQ4MA--
/https://s.yimg.com/os/creatr-images/2020-05/45815870-9238-11ea-8fff-
1180915abec1"
alt="Portfolio" /></div>
<h3 class="content-box-title
font-serif">Global Safety
Journal</h3></a></div>
<div class="col-md-4 col-sm-6
col-xs-12"><a
class="content-box"
href="/destinations">
<div class="content-box-image"><img
src="https://images.pexels.com/photos/7235910/pexels-photo-
7235910.jpeg?auto=compress&cs=tinysrgb&w=300"
alt="Restaurant" /></div>
<h3 class="content-box-title
font-serif">Destinations</h3></a></div>
<div class="col-md-4 col-sm-6
col-xs-12"><a
class="content-box" href="/emergency">
<div class="content-box-image"><img
```

```
src="https://cdn.dribbble.com/userupload/7319839/
file/original-86f171450ede48437f8b1cfd8ac352f1.png?
crop=0x0-2880x2160&resize=400x300&vertical=center"
alt="Finance" /></div>
<h3 class="content-box-title
font-serif">Emergency
services</h3></a></div>
{/* <div class="col-md-4 col-sm-6
col-xs-12"><a class="content-box"
href="/product">
<div class="content-box-image"><img
src="https://cdn.dribbble.com/userupload/10920096/file/

original-4b166c456ae22ae068a4c961d6053a74.png?resize=400x0"
alt="Landing"/></div>
<h3 class="content-box-title
font-serif">Travel
packages</h3></a></div> */}
              </div>
            </div>
          </section>
        </div>
<div class="scroll-up"><a
href="#totop"><i
class="fa
fa-angle-double-up"></i></a></div>
      </main>
<script
src="assets/lib/jquery/dist/jquery.js"></script>
<script
src="assets/lib/bootstrap/dist/js/bootstrap.min.js"></script>
<script
src="assets/lib/wow/dist/wow.js"></script>
<script
src="assets/lib/jquery.mb.ytplayer/dist/jquery.mb.YTPlayer.js"></script>
<script
src="assets/lib/isotope/dist/isotope.pkgd.js"></script>
<script
src="assets/lib/imagesloaded/imagesloaded.pkgd.js"></script>
<script
```

```
src="assets/lib/flexslider/jquery.flexslider.js"></script>
<script
src="assets/lib/owl.carousel/dist/owl.carousel.min.js"></script>
<script
src="assets/lib/smoothscroll.js"></script>
<script
src="assets/lib/magnific-popup/dist/jquery.magnific-popup.js"></script>
<script
src="assets/lib/simple-text-rotator/jquery.simple-text-rotator.min.js"></script>
<script
src="assets/js/plugins.js"></script><script
src="assets/js/main.js"></script>
</body>

  )
}

export default AdminPowers
import React, { useState, useEffect }
from
"react";
import { useNavigate } from
"react-router-dom";
import axios from "axios";
import Loader from
"../components/Loader";
import Error from "../components/Error";import Success from
"../components/Success";
import "./login.css";
import { Link } from "react-router-dom";// import Loginimg from
"../images/Login.svg";function
LoginScreen() {

const [FormData, setFormData] =
useState({

  });
const navigate=useNavigate()
const [loading, setLoading] =
useState(false); const [error, setError]
```

```
= useState("");
const [success, setSuccess] =
useState("");
const handlechange=(e)=>{
  setFormData({
    ...FormData,
    [e.target.name]:e.target.value
  })
}
const Login=async(e)=>{
const {username,password}=FormData

await axios.post('admin-login/',{
  username,
  password
}).then((res)=>{
  if(res.status===200){
    console.log(res.status)
    if(res.status==200){
localStorage.setItem('adminkey',res.data.access)
localStorage.setItem('isadmin',res.data.is_superuser)
alert("admin login
successfull..welcome..."+res.data.username)
navigate('/admindashbord')
    }
    else{
alert("detected unautherized entry...")    }
    setLoading(false);
  }

}).catch(()=>{
alert("detected unauthorized entry..")  })
}

  return (
    <div className="login--screen">
{/* <img src={Loginimg} alt="login img"
className="login--img" /> */}
      {loading && <Loader></Loader>}
```

```
<div className="row
justify-content-center"> <div
className="col-md-5 mt-5">
{error.length > 0 && <Error
msg={error}></Error>}
          <div className="bs">
            <h2>Admin...Login</h2>

            <input
              type="text"
              className="form-control"
              name="username"
placeholder="Enter username...."
onChange={handlechange}
              style={{
                textAlign: "center",
                padding: "20px",
                marginBottom: "20px",
              }}
            />
            <input
              type="password"
              className="form-control"
placeholder="enter Password...."
name="password"
              onChange={handlechange}
              style={{
                textAlign: "center",
                padding: "20px",
                marginBottom: "20px",
              }}
            />
            {loading ? (
<div>Login...Please Wait...</div> ) : (
<button className="login--btn"
onClick={Login}>
                Login
              </button>
            )}
          </div>
```

```
        </div>
      </div>
    </div>
  );
}

export default LoginScreen;
```

BACKEND

```
//
// Source code recreated from a .class
file by
IntelliJ IDEA
// (powered by Fernflower decompiler)
//
from django.db import models

# Create your models here.


from django.contrib.auth.models import
AbstractUser
from django.db import models

class CustomUser(AbstractUser):

phone = models.CharField(max_length=15,
blank=True)
place = models.CharField(max_length=255,blank=True)
nickname =
models.CharField(max_length=30,
blank=True)
color = models.CharField(max_length=30,
blank=True)
image =
models.ImageField(upload_to='user_images/',
blank=True, null=True)
```

```
    def _str_(self):
        return self.username



class Package(models.Model):
admin = models.ForeignKey(CustomUser,
on_delete=models.CASCADE)
title = models.CharField(max_length=255)    about = models.TextField()
amount =
models.DecimalField(max_digits=10,
decimal_places=2)
    rating = models.IntegerField()
location =
models.CharField(max_length=255)duration
=
models.CharField(max_length=50)no_of_people
= models.CharField(max_length=50)hotel =
models.ImageField(upload_to='package_photos/',
blank=True, null=True)
destination =
models.ImageField(upload_to='package_photos/',
blank=True, null=True)
activity =
models.ImageField(upload_to='package_photos/',
blank=True, null=True)
attraction =
models.ImageField(upload_to='package_photos/',
blank=True, null=True)

    def _str_(self):
        return self.title


class Comments(models.Model):
user = models.ForeignKey(CustomUser,
on_delete=models.CASCADE)
package = models.ForeignKey(Package,
on_delete=models.CASCADE)
```

```
    comment = models.TextField()
created =
models.DateField(auto_now_add=True)
    def _str_(self):
        return self.comment


class Blogs(models.Model):
user = models.ForeignKey(CustomUser,
on_delete=models.CASCADE)
title = models.CharField(max_length=255)    description = models.TextField()
    rating = models.IntegerField()
food =
models.ImageField(upload_to='blog_photos/',
blank=True, null=True)
hotel =
models.ImageField(upload_to='blog_photos/',
blank=True, null=True)
travelling =
models.ImageField(upload_to='blog_photos/',
blank=True, null=True)
activity =
models.ImageField(upload_to='blog_photos/',
blank=True, null=True)
created =
models.DateField(auto_now_add=True)
    def _str_(self):
        return self.title

class Review(models.Model):

name =
models.CharField(max_length=255,blank=True,null=True)
email=models.CharField(max_length=255,blank=True,null=True)
    review = models.TextField()
created =
models.DateField(auto_now_add=True)
    def _str_(self):
        return self.review
"""
```

```
URL configuration for travel_guide
project.

The urlpatterns list routes URLs to
views. For
more information please see:
https://docs.djangoproject.com/en/5.0/topics/http/urls/
Examples:
Function views
1. Add an import: from my_app import
views2. Add a URL to urlpatterns:
path('',
views.home, name='home')
Class-based views
1. Add an import: from other_app.views
import
Home
2. Add a URL to urlpatterns: path('',
Home.as_view(), name='home')
Including another URLconf
1. Import the include() function: from
django.urls import include, path
2. Add a URL to urlpatterns:
path('blog/',
include('blog.urls'))
"""
from django.contrib import admin
from django.urls import path
from . import views
from rest_framework_simplejwt import
views as
jwt_views

urlpatterns = [
path('',views.UserRegistrationView.as_view(),
name='user-registration'),
path('admin-register/',
views.SuperuserRegistrationView.as_view(),
name='superuser-registration'),
path('user-login/',
```

```
views.UserloginView.as_view(),
name='user-token_obtain_pair'),
path('admin-login/',
views.AdminloginView.as_view(),
name='admin-token_obtain_pair'),
path('api/token/refresh/',
jwt_views.TokenRefreshView.as_view(),
name='token_refresh'),
path('user-details/',
views.UserDetailsView.as_view(),
name='user-details'),
path('update-user/',
views.UpdateUserDetailsView.as_view(),
name='update-user-details'),

path('send-otp/',
views.PasswordResetOTPSendView.as_view(),
name='update-user-details'),
path('otp-validation/',
views.OTPValidationView.as_view(),
name='otp-validation'),
path('change-password/',
views.ChangePasswordView.as_view(),
name='new-password'),

path('package-crud/',
views.PackageCRUDView.as_view()),
path('package-crud/<int:pk>/',
views.PackageCRUDView.as_view(),),

path('list-packages/',
views.ListPackagesView.as_view(),
name='list-packages'),
path('package-detail/<int:pk>/',
views.PackageDetailView.as_view(),
name='package-detail-view'),

path('create-comment/<int:pk>/',
views.CreateCommentView.as_view(),
name='create-comment'),
```

```
path('list-comments/<int:pk>/',
views.ListCommentsView.as_view(),
name='list-comments'),
path('delete-comment/<int:pk>/',
views.DeleteCommentView.as_view(),
name='delete-comment'),


path('create-blog/',
views.CreateBlogView.as_view(),
name='create-blog'),
path('list-blogs/',
views.ListBlogsView.as_view(),
name='list-blogs'),
path('blog-detail/<int:pk>/',
views.BlogDetailView.as_view(),
name='blog-detail'),

path('list-user-blogs/',
views.ListUserBlogsView.as_view(),
name='list-user-blogs'),
path('update-blog/<int:pk>/',
views.UpdateBlogView.as_view(),
name='update-blog'),
path('delete-blog/<int:pk>/',
views.DeleteBlogView.as_view(),
name='delete-blog'),


path('create-review/',
views.CreatReviewView.as_view(),),
path('list-reviews/',
views.ListReviewView.as_view(),),
path('review-detail/<int:pk>/',
views.ReviewDetailView.as_view(),),


]
from rest_framework import serializers
from .models import
```

```
CustomUser,Package,Comments,Blogs,Reviewfrom
rest_framework_simplejwt.serializers
import TokenObtainPairSerializer
from django.core.exceptions import
ObjectDoesNotExist


class
UserTokenObtainPairSerializer(TokenObtainPairSerializer):
    @classmethod
    def get_token(cls, user):
        token = super().get_token(user)
        # adding custom claims
token['username'] = user.usernametoken['is_superuser'] =
user.is_superuser return token

    def validate(self, attrs):
        data = super().validate(attrs)
        user = self.user
        if user.is_superuser==False:
data["is_superuser"] = user.is_superuser
data["username"] = user.username
            return data
        else:
raise serializers.ValidationError("Only
common
users are allowed to log in here.")


class
AdminTokenObtainPairSerializer(TokenObtainPairSerializer):
    @classmethod
    def get_token(cls, user):
        token = super().get_token(user)
        # adding custom claims
token['username'] = user.usernametoken['is_superuser'] =
user.is_superuser return token

    def validate(self, attrs):
        data = super().validate(attrs)
```

```
        user = self.user
        if user.is_superuser==True:
data["is_superuser"] =
user.is_superuserdata["username"] =
user.username return data
        else:
raise serializers.ValidationError(" Onlyadministrators are allowed to log in
here.")



class
CustomUserSerializer(serializers.ModelSerializer):
password_confirmation =
serializers.CharField(write_only=True)
email =
serializers.EmailField(required=True)
image=serializers.ImageField(max_length=None,use_url=True,required=False)
    class Meta:
        model = CustomUser
fields = ('id', 'username', 'email',
'phone',
'place', 'image',
'password','password_confirmation')
extra_kwargs = {'password':
{'write_only':
True}}

    def validate(self, data):
        try:
CustomUser.objects.get(email=data['email'])raise
serializers.ValidationError({'email':'Email
already exists.'})
        except ObjectDoesNotExist:
            pass

if data['password'] !=
data['password_confirmation']:
raise
serializers.ValidationError({'password_confirmation':"Passwords
```

```
do not match."})
        return data

    def create(self, validated_data):
validated_data.pop('password_confirmation',
None)
user =
CustomUser.objects.create_user(**validated_data)
        return user

class
CustomUserupdateSerializer(serializers.ModelSerializer):
image=serializers.ImageField(max_length=None,use_url=True,required=False)
    class Meta:
        model = CustomUser
fields =
['username','image','phone','place','email']

    def validate_email(self, value):
if
CustomUser.objects.filter(email__iexact=value)
.exclude(pk=self.instance.pk).exists():
raise serializers.ValidationError("This
email
address is already in use.")
        return value


class
PackageSerializer(serializers.ModelSerializer):
    class Meta:
        model = Package
        exclude = ['admin']

class
PackagelistSerializer(serializers.ModelSerializer):
    class Meta:
        model = Package
        fields = "_all_"
```

```
class
CommentSerializer(serializers.ModelSerializer):
    class Meta:
        model = Comments
        fields = ['comment']

class
CommentlistSerializer(serializers.ModelSerializer):
    class Meta:
        model = Comments
        fields = "_all_"

class
BlogSerializer(serializers.ModelSerializer):
    class Meta:
        model = Blogs
        exclude = ['user']

class
BloglistSerializer(serializers.ModelSerializer):
    class Meta:
        model = Blogs
        fields = "_all_"


class
ReviewSerializer(serializers.ModelSerializer):
    class Meta:
        model = Review
        fields = "_all_"

class
ReviewlistSerializer(serializers.ModelSerializer):
    class Meta:
        model = Review
        fields = "_all_"
from django.contrib import admin

# Register your models here
from .models import
```

```
CustomUser,Package,Comments,Blogs,Review# admin.site.register(CustomUser)
class
CustomUseryadmin(admin.ModelAdmin):
list_display =
['username','is_superuser']admin.site.register(CustomUser,CustomUseryadmin)

class Packageadmin(admin.ModelAdmin):
list_display =
['admin','title','amount','location','duration']
admin.site.register(Package,Packageadmin)

class Commentsadmin(admin.ModelAdmin):
list_display =
['user','comment','package']admin.site.register(Comments,Commentsadmin)

class Blogsadmin(admin.ModelAdmin):
    list_display = ['user','title']
admin.site.register(Blogs,Blogsadmin)



class Reviewadmin(admin.ModelAdmin):
    list_display = ['name','review']
admin.site.register(Review,Reviewadmin)
```

UNLOCKING TRACEABILITY AND TRANSPARENCY IN SUPPLY CHAINS WITH BLOCKCHAIN TECHNOLOGY

PROJECT REPORT

Submitted By

MEGNA BIJU

Reg. No. CCAVBCA044

For the award of the Degree of

Bachelor (BCA)

of Computer Application

(University of Calicut)

under the guidance of

Ms.Priyanga K.K

Assistant Professor

BCA (Bachelor of Computer Application)
DEPARTMENT OF COMPUTER SCIENCE
CHRIST COLLEGE (AUTONOMOUS)
IRINJALAKUDA, KERALA

2021-2024

DEPARTMENT OF COMPUTER SCIENCE

CHRIST COLLEGE (AUTONOMOUS)

IRINJALAKUDA

CERTIFICATE

This is to certify that the project report entitled ***Unlocking Traceability and Transparency in Supply Chains with Blockchain Technology*** *is a bonfied record of the project work done by* ***Megna Biju*** *in partial fulfillment of the requirement for the sixth semester of* ***Bachelor of Computer Application*** *in Department of Computer Science of* **CHRIST COLLEGE (AUTONOMOUS) IRINJALAKUDA**.

Ms.Priyanga K.K
Assistant Professor,CS
Internal Guide

Ms.Sini Thomas
Head of the Department
Computer Science

EXTERNAL EXAMINER

INTERNAL EXAMINER

2023 3rd International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON)

29th – 31st December 2023

Certificate

This is to certify that Dr./Prof./Mr./Ms. *Megna Biju* *has presented paper entitled* ***Unlocking Traceability and Transparency in Retail Supply Chains with Blockchain Technology*** *in 2023 3rd International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON) during 29th & 31st December 2023.*

Prof. Zahara Amreen
Coordinator

Prof. Mohammed Fazale Elahi
Convener

Dr. Zahir Hasan
Conference Chair

DECLARATION

I hereby declare that this project work **"Unlocking Traceability and Transparency in Supply Chains with Blockchain Technology"** submitted to Christ College (Autonomous) Irinjalakuda, affliated to Calicut University in partial fullfillment of the requirement for the award of the Bachelor of Computer Application, is a record of original work done by me, under the guidance of Ms.PRIYANGA K.K, Assistant Professor, Department of Computer Science.

Place: Irinjalakuda MEGNA BIJU

ACKNOWLEDGEMENT

First and foremost, I would like to thank God almighty for his guidance and support throughout the project.I wish to express my sincere gratitude to my beloved Department for providing me all the facilities for my project. I take this opportunity to express my gratitude to the class teacher Ms.SOUMYA P.S and HOD(Head Of Department) Ms.SINI THOMAS who supported me throughout the course of this project.I am thankful for their aspiring guidance and valuable advice during the project work.I express my sincere thanks to my project guide Ms.PRIYANGA K.K for supporting and guiding me throughout the project.I would also like to take this opportunity, to specially thank, all other faculty members for their constant and continuous motivation.Finally, I would like to thank my family and friends for giving valuable advices and moral support throughout my project.

ABSTRACT

Unlocking Traceability and Transparency in Supply Chains with Blockchain Technology is a topic that investigates and proposes a blockchain based traceability framework. Traceability is very crucial for multi-tier production where customers demand transparency and quality assurance. This study proposes a blockchain-based traceability framework to address issues like information asymmetry and low visibility.It suggests using smart contracts and transaction validation rules to ensure secure information sharing.The system aims to build trust among supplychain partners by using a distributed ledger to store and authenticate transactions, offering flexibility and transparency.

Contents

Chapter 1

1 Introduction

Supply chains are intricate systems critical to global operations, but they often lack transparency and traceability, leading to issues like disruptions, counterfeits, and ethical concerns. Blockchain technology, known for its decentralized and immutable record-keeping, offers a solution. This project explores blockchain's potential in enhancing supply chain traceability and transparency across industries like manufacturing, healthcare, food, and luxury goods. It analyses real-world case studies, discusses benefits and limitations, and aims to inform businesses, policymakers, and researchers about the transformative impact of blockchain on supply chains, emphasizing its role in building trust and improving operations.By facilitating transparent and secure traceability throughout the supply chain, blockchain not only enables better information sharing but also fosters trust among supply chain partners spread across the globe.
Additionally, existing methodologies for blockchain-based supply chain traceability and transparency primarily focus on leveraging blockchain's fundamental principles to create a transparent and immutable record of supply chain activities. These methodologies involve the deployment of blockchain networks to record every transaction and product movement, ensuring data transparency. To tackle data privacy issues, existing approaches recommend employing encryption, off-chain storage, and data access controls to protect sensitive information. Smart contracts are commonly utilized to automate and enforce supply chain agreements, reducing manual intervention and enhancing transparency. Regulatory compliance is acknowledged as an essential aspect, with a focus on adapting to existing legal frameworks.
This system suggests using smart contracts and transaction validation rules to ensure secure information sharing. It also aims to build trust among supply chain partners by using a distributed ledger to store and authenticate transactions, offering flexibility and transparency. Traceability and Transparency are the key factors of Blockchain technology.

1.1 Overview

This project explores how blockchain technology can improve supply chain traceability and transparency across various industries. By analyzing real

world case studies, it highlights blockchain's potential to foster trust and enhance operations. The Textile and Clothing sector, facing challenges like opaque supply chains, can benefit from blockchain's ability to ensure transparent and secure traceability, addressing issues such as product recalls and labor concerns. Blockchain technology offers several features that make it particularly suitable for transactions:

- Decentralization: Blockchain operates on a decentralized network of computers (nodes) that collectively validate and record transactions. This removes the need for a central authority, reducing the risk of manipulation or control by a single entity.
- Immutability: Once a transaction is recorded on the blockchain, it becomes extremely difficult to alter or delete. Each block contains a cryptographic hash of the previous block, creating a chain of blocks that are interlinked and resistant to tampering.
- Transparency: All transactions on a blockchain are transparent and can be viewed by anyone with access to the network. This transparency helps to foster trust among participants and can reduce the risk of fraud.
- Security: Blockchain uses cryptographic techniques to secure transactions and ensure the integrity of the network. Transactions are verified by multiple nodes before being added to the blockchain, making it extremely difficult for malicious actors to manipulate the system.
- Smart Contracts: Smart contracts are self-executing contracts with the terms of the agreement directly written into code.They automatically execute and enforce the terms of the agreement when predefined conditions are met, eliminating the need for intermediaries and reducing transaction costs.

Chapter 2

2 System Analysis

2.1 Purpose

The purpose is to explore the potential of blockchain technology in enhancing supply chain traceability and transparency across various industries. It aims to inform stakeholders about the transformative impact of blockchain on supply chains, emphasizing its role in building trust and improving operations.
Supply chains are complex networks crucial for global operations, but they often lack transparency and traceability, leading to issues like disruptions and counterfeits, and ethical concerns. Blockchain technology, with its decentralized and immutable record-keeping, presents a solution. It also smart contracts and cryptographic techniques which enhances security.

2.1.1 Existing System

Blockchain's core features, including cryptographic equations and smart contracts, ensure data integrity and automate agreements within supply chains. Consensus mechanisms like Proof of Work (PoW) and immutable ledgers provide tamper resistance, addressing concerns over data manipulation. Existing methodologies focus on leveraging these principles to create transparent records of supply chain activities using blockchain networks. They tackle challenges such as scalability, interoperability, and data privacy through various techniques like sharding, encryption, and standardized formats. Smart contracts streamline processes and enhance transparency, while regulatory compliance remains a key consideration.
Overall, these methodologies aim to integrate blockchain effectively, considering scalability, interoperability, privacy and regulatory aspects to establish robust and transparent supply chains.

2.1.2 Proposed System

The proposed methodology builds upon blockchain's core features to address challenges in traditional supply chains. It emphasizes the creation of decentralized and immutable ledgers to record every transaction and product movement, ensuring transparency and data integrity. To enhance

scalability, techniques like sharding and sidechains are proposed, along with cross-chain interoperability standards. Privacy concerns are addressed through technologies like zero-knowledge proofs and secure multi-party computation. Smart contracts automate and enforce agreements, reducing manual intervention and disputes. Regulatory compliance is prioritized through collaboration with regulators and pilot projects to refine solutions within regulatory boundaries. We use SQLite3 as database.

Chapter 3

3 Feasibility Study

After the problem is clearly understood and solutions proposed, the next step is to conduct the feasibility study. Feasibility study is defined as evaluation or analysis of the potential impact of the proposed project or program. The objective is to determine whether the proposed system is feasible. There are three aspects of feasibility study to which the proposed system is subjected as discussed below.

3.1 Technical Feasibility

Technical feasibility evaluates whether the existing technical infrastructure is capable of supporting the implementation of blockchain solutions. It involves analyzing whether upgrades or modifications to current systems are necessary to accommodate blockchain integration. Additionally, it examines whether the proposed blockchain system can seamlessly integrate with the existing infrastructure without requiring significant hardware or software changes.

3.2 Economical Feasibility

Economic feasibility focuses on determining whether the required resources, including time and financial investments, are available to develop and deploy the blockchain-based supply chain solution. It entails assessing the cost effectiveness of implementing blockchain technology compared to traditional supply chain management approaches. Since blockchain implementation typically does not require additional hardware investments and offers cost saving benefits in the long run, it is deemed economically feasible.

3.3 Operational Feasibility

Operational feasibility evaluates whether the organization possesses the necessary human resources and capabilities to operate and maintain the blockchain based supply chain solution effectively. It assesses the training requirements for personnel and determines whether the system can be easily adopted and used by stakeholders. Given that the blockchain solutions aim to simplify processes and enhance transparency, minimal additional training is typically required for users familiar with

internet usage and the English language. Moreover, the organization already possesses the necessary resources for implementation, further enhancing operational feasibility.

Chapter 4

4 Software Requirement Specification

4.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the 'Traceability and Transparency in Supply Chains with Blockchain Technology'. It illustrate the purpose and complete description for the development of the system.It explains system constraints, interface and interactions with other external applications. This document is primarily intended to be proposed to make supply chains more transparent and traceable.

4.2 Scope

Our project has made it easier to increase the traceability and transparency of supply chains using blockchain technology. It has helped to increase the trust among consumers towards supply chains.

4.3 Overall Description

Our project emphasizes the importance of blockchain in addressing challenges like scalability, interoperability, data privacy, and regulatory compliance in supply chain management. It also highlights real - world implementations and the transformative impact of blockchain on improving operational efficiency, trust among stakeholders, and compliance with regulatory frameworks. It suggests future research directions, including integrating blockchain with emerging technologies like AI, IoT, and big data analytics to further enhance supply chain management.

4.3.1 Product Perspective

Our project is mainly used to build the trust among different stakeholders and suppliers so we can make the supply chain market boost its way through the economy.

4.3.2 Product Functionality

Through this system we can trace the payment details in our website. It will help in making our system more transparent.

4.3.3 Users and Characteristics

There are two users - admin and user. The admin can regulate everything in the system and they can add new new products. The users or the customers can purchase anything from the website and can make the payments through the wallet installed in it.

4.4 Specific Requirements

4.4.1 Hardware Requirements

- System: LAPTOP-OE7M0PNN
- Processor: 12th Gen Intel(R) Core(TM) i5-1235U
- Speed: Above 1GHz
- RAM capacity: 8 GB
- Hardisk drive: 212 GB

4.4.2 Software Requirements

- Operating System:Windows
- Frontend: React js
- Backend: Python
- Database : SQLite3
- Technologies used: HTML,Javascript

4.5 Functional Requirements

It contains four main modules.

- 1.Home
- 2.API
- 3.User
- 4.Product

Home

Home module provides top-level menus where visitors can go deeper into various areas of the site. It contains cart, products,etc. Home module contains the admin who manages the system.

API

The Module API provides functions that return information about the current operating environment (module, version, and instance). The Modules API also has functions that retrieve the address of a module, a version, or an instance. It also sends API calls to the frontend and backend.

User

The user module allows users to register, log in, and log out. Users benefit from being able to sign on because this associates content they create with their account and allows various permissions to be set for their roles. User can also interact with the system, select product, make payment etc.

Product

The Products module is a catalogue of the products and their descriptions. The product module contains all the products, details about the products,updation etc.

4.6 Non Functional Requirements

Non-functional requirements define the overall qualities or attributes of the resulting system. Non-functional system place restrictions on the product being developed, the developed process, and specify external constraints that the product must meet. Examples of non-functional requirements include safety, security, usability, reliability and performance requirements. Project management issues (costs, time and schedule) are often considered as non-functional requirements. The principal non - functional constraints which are relevant to critical systems :

- performance
- security
- safety

- usability

Performance

Performance requirements concern the speed of operation of a system. Types of performance requirements :

- Response requirements (how quickly the system reacts to a user input).
- Throughput requirements (how much the system can accomplish within a specified amount of time).
- Availability requirements (is the system available for service when requested by end users). The speed of operation of this system is adequate for the requirements.

Reliability

- Reliability is the ability of a system to perform its required function under stated conditions for a specified period of time.
- constraints on the runtime behavior of the system. This system is reliable because its functionalities can be done on the required conditions.

Safety

Safety requirements are not required which exclude unsafe situation from the possible solution of the system.

Usability

Usability is the ease with which a user can learn to operate, prepare inputs for, and interpret outputs of system or components. Usability requirements include :

- information error messages.
- well-formed user interfaces.

4.7 Interface Requirements

4.7.1 Hardware interfaces

The system must run over the internet, all the hardware shall require to connect internet will be hardware interface for the system. As for example modem, WAN-LAN.

4.7.2 Software interfaces

Software interface required for the working of the project is the appropriate operating system.

4.7.3 Communication interfaces

The user of the site communicate using the network connectivity and the data set is accessed.

4.8 Security Requirements

- User accesses only their account.
- Validation of input is handled.
- This application containing the computer systems is physically secured against arms or surreptitious entry by intruders.
- Users must be authorized carefully to reduce changes of any such user giving access to an intruder in exchange for a bribe or other favour.

4.9 Platform Used

Windows 11 is the latest version of Microsoft's operating system, succeeding Windows 10. It brings a redesigned user interface with a centered Start menu, improved window management features like Snap layouts and virtual desktops, enhanced gaming capabilities with DirectStorage and Auto HDR, and better integration with Microsoft Teams. It also introduces support for Android apps through the Microsoft Store and emphasizes security with features like Secure Boot, TPM 2.0, and enhanced protection against ransomware attacks. Overall, Windows 11 aims to provide a more modern, streamlined, and secure computing experience for users.

4.10 Technologies Used

Python

Python is an interpreted, object-oriented, and high-level programming language with dynamic semantics. Its built-in data structures, combined with dynamic typing and dynamic binding, make it attractive for Rapid Application Development. Additionally, Python serves as a scripting or glue language to connect existing components together. Python is a powerful, readable, and versatile language used in various domains, including machine learning, web development, and data analysis. Its ease of use and extensive standard library contribute to its popularity and widespread adoption.

JavaScript

JavaScript is a versatile programming language commonly used for web development. React, a popular JavaScript library, is used for building user interfaces, particularly single-page applications. React allows developers to create reusable UI components and efficiently manage the state of an application. It uses a virtual DOM for optimal performance, updating only the necessary parts of the UI when changes occur. React is often combined with other libraries and frameworks, such as Redux for state management and React Router for navigation, to create complex web applications.

HTML

HTML (Hypertext Markup Language) is the standard markup language for creating web pages and web applications. It provides a structure for content on the web by using a system of tags and attributes to define elements such as headings, paragraphs, links, images, and forms. HTML documents are interpreted by web browsers to render the visual presentation of web pages. HTML is an essential building block of the web and is often combined with CSS (Cascading Style Sheets) for styling and JavaScript for interactivity to create dynamic and visually appealing websites.

SQLite3

SQLite 3 is a lightweight, embedded relational database management system (RDBMS) that is widely used in various applications due to its simplicity, efficiency, and portability. Developed as a self-contained, serverless, and zero-configuration database engine, SQLite 3 is designed to be seamlessly integrated into applications, making it a popular choice for embedded

systems, mobile devices, and desktop software. Its compact nature doesn't compromise on capabilities, as it supports standard SQL features and provides reliable ACID transactions. Additionally, SQLite 3's open-source nature and cross-platform compatibility contribute to its widespread adoption across a diverse range of software development projects. Whether powering mobile apps, web browsers, or desktop applications, SQLite 3's ease of use and robust performance make it a versatile and dependable database solution.

Chapter 5

5 Design Document

5.1 Purpose

For our system, we created a demo website for implementing the blockchain technology in transactions. We came up with a system where we implement blockchain technology in supply chains to make it more traceable and transparent. We came up with a system and implemented it in a demo website in the payment section where blockchain is implemented in transactions, which helps in making the transactions more transparent and trustworthy.
Purpose of this document is to give the detailed description of the architecture and system design for the software.The system design document shows the software will be structured to satisfy the requirement identified in the software requirement specification.It is a translation of requirements into a description of the software structure,software component, interfaces and data necessary for the implementation phase.In a complete software design document,each requirements must be traceable to one or more design entities.

5.2 Scope

The demo website implemented in our system can be adopted by many retail companies and it will help boost trust in consumers and stakeholders.

5.3 Overview

The purpose of this document is to help the reader to visualize the solution to the project presented. This document verifies how the design meet the requirement stimulated in the SRS document through design viewpoints. The design viewpoints will cover all design elements presented before.
This document will provide a direct approach to the development of this project hence reducing feature creep and ponitedly determine the quality of the design.

5.4 Data Design

Database are the storehouses of data used in the software systems.The data is stored in tables inside the database.Several tables are created for the manipulation of the data for the system.Two essential settings for a database are:

- Primary Key- The field that is unique for all the record occurrence.
- Foreign Key- The field used to set relation between tables.

Normalization

- Normalization is the process of organizing the data in the database.
- Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate the undesirable characteristics like Insertion, Update and Deletion Anomalies.
- Normalization divides the larger table into the smaller table and links them using relationship.
- The normal form is used to reduce redundancy from the database table.

Tables

User

Name	DataType	Constraints	Description
id	integer	Primarykey	ID of user
email	string	unique	Email of the users
username	string	unique	Username of user
firstName	string	Notnull	Firstname of user
lastName	string	Notnull	Lastname of user
dateJoined	date	Notnull	Date of join
lastLogin	date	Notnull	Date of last login
password	string	Notnull	Pasword of user
Mobile No.	integer	Notnull	phone number of user
companyName	string	Notnull	Name of the company

Products

Name	DataType	Constraints	Description
id	string	Primarykey	ID of product(auto generated)
name	string(60)	Notnull	Name of product
price	float	Notnull	Price of product
description	string(200)	null	Description of product
image	string(255)	null	Image of product
Warranty_period	integer	Default[0]	period of warranty
created_at	date	Notnull	Date of manufacture
modified_at	date	Notnull	Date of modification
user_id	primary	Foriegn	id from users table

Cart

Name	DataType	Constraints	Description
id	string	Primarykey	cart id (auto generated)
product_id	string	Foreignkey	id from products table
quantity	integer	Default	No. of items in cart
date_added	date	Notnull	date the item was added
user_id	string	Foriegn key	id from user table

NFT details

Name	DataType	Constraints	Description
id	string	Primarykey	auto generated
product_id	string	foriegn key	id from products table
token_url	string(200)	Default[' ']	url of the tokens
expiry_date	date	Default[' ']	expiry date of token
user_id	string	Foriegn key	id from user table
token_id	integer	Unique, Default	id of token
acc_address	string(300)	Default[' ']	Address of the account
redeem	boolean	Default[' TRUE ']	only returns true

Order items

Name	DataType	Constraints	Description
id	string	Primarykey	Auto generated
product_id	string	Foreignkey	id from product table
quantity	integer	Notnull	No. of items ordered
order_date	date	Notnull	date the item was ordered
user_id	string	Foreign key	auto generated

Payment details

Name	DataType	Constraints	Description
id	string	Primarykey	Auto generated
orders_id	string	Foreignkey	id from order items
total_amount	float	Notnull	total amount to be paid
status	boolean	Default[' TRUE ']	status of payment
payment_date	date	Notnull	Date of payment

Order details

Name	DataType	Constraints	Description
id	string	Primarykey	Auto generated
payment_id	string	Foreignkey	id from payment details

Shipping Address

Name	DataType	Constraints	Description
id	string	Primarykey	auto generated
user_id	string	Foriegn key	id from user table
address	string	Notnull	address to be shipped to
city	string	Notnull	city to be shipped to
state	string	Notnull	state to be shipped to
postal_code	integer	Notnull	postal code to be shipped to
date_added	date	Notnull	date to be shipped

Track Repairs

Name	DataType	Constraints	Description
id	string	Primarykey	auto generated
orders_id	string	Foreign key	id from order items
description	string	null	Description of repairs
percent_work_done	float	Notnull	Percentage of repair done
Details_id	string	Foriegn key	Details about payment and order

Chapter 6

6 Development of the System

This system can be decomposed into many number of submodules. The sub modules of our system 'Unlocking Traceability and Transparency in Supply Chains with Blockchain Technology' are home, API, product and user. Each sub module have specific objectives,to avoid unwanted coupling between modules and to increase cohesion, modules are again decomposed into submodules. We have also implemented a demo website to use this system.

Chapter 7

7 System Testing

Testing is the penultimate step of software development.An elaborate testing of the data is prepared and the system is using the test data,while doing testing, errors are noted and correction is made.The users are trained to operate the developed system.Both hardware and software securities are made to run the developed system successfully.System testing is aimed at ensuring the system works accurately before the live operation commences.

7.1 Test Plan

7.1.1 Scope

This test plan will cover the following testing activities as identified in the testing strategy.

- White Box Testing

White box testing is a testing technique, that examines the program structure and derives test data from the program logic/code. In white box testing, the UI is bypassed. Inputs and outputs of ratings are tested directly at the code level and the results are compared against specifications. It reveals errors in "hidden" code.

- Black Box Testing

Black box testing is a method of software testing that examines the functionality of an application based on the specifications. Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end - user would.

- Unit Testing

The module includes four main components to undergo unit testing. It checks the syntax error, logic error and validity of the program correctness. This test will be performed by the developers.

- Integration Testing

After all components pass the unit testing, they get to be tested whether they work correctly when they are running concurrently and communicating to each other. Integration tests exercise an entire subsystem and ensure that a set of components play nicely together.

- Internal Data Testing

We will test the validity of the data before it enters the database to avoid any problems that may face us in the database. We will test the encryption of the personal information of all the users along with the user names and passwords to ensure maximum security of the users privacy.

7.1.2 Software risk issues

In this section, the plan is to test the risk involved in critical issues such as:

- Difficult to run on visual studio due to large loading time
- Some inherent software risks such as complexity exist; also these issues need to be identified.
- Proper network connection
- Working of SQLite3 database

7.1.3 Features to be tested

- Test whether correct user name and password allows you to login.
- Test whether invalid user name and password prevents you from login.
- Test whether there is any connection problem in the Server.
- Test whether invalid data entry allows saving data successfully.
- Test whether products are added to cart successfully.
- Test whether all the data entered are valid.
- Test whether all pages are loaded correctly.
- Test whether the provided security works correctly.

7.2 Test consolidation

7.2.1 Test item

The items or features to be tested in the test cases are included in the document. Each and every user input is tested. The present condition of the system is tested. It is checked to make sure that environment is ready for the application to work.

7.2.2 Input specifications

These are the inputs required to execute the test case. All required inputs including data elements and values are given below.

Data	Values
username	A valid username
Password	Combination of characters and numbers

Chapter 8

8 System Implementation and Maintenance

8.1 Implementation

System implementation is the conversion of new system into an operating one which involves creating compatible files, training clients and installing hardware. User training is crucial for minimizing resistance to change and giving chance to prove its worth. Training aids user friendly manuals and healthy screens provide the user with a good start. Software maintenance follows conversion to the extent that changes are necessary to maintain satisfactory operations relative to changes in the user's environment. Maintenance often includes minor enhancements or corrections to the problem that surface late in the systems operations.
In the implementation phase, the team builds the components either from scratch or by composition. Given the architecture document meant from the design phase and the requirement document from the analysis phase, the team should build exactly what has been requested, though there is still room for innovation and flexibility. For example, a component may be narrowly designed for this particular system, or the component may be made more general to satisfy a reusability

- Careful planning
- Investigation of system and constraints
- Design the methods to achieve changeover.
- Training the staff in the changed phase.
- Evaluation of change over method.
- The method of implementation and time scale to be adopted are found out initially.

8.2 Maintenance

This phase occurs as a result of deploying the whole system at the end users organization. They will perform the beta testing at the end users and inform to the developers about any needed modification to the application. The customer records all the problems that are encountered during the beta testing and reports these to the developer at regular intervals.

8.2.1 Corrective Maintenance

Even with the best quality assurance activities, it is likely that the customer will uncover defects in the software. Corrective maintenance changes the software to correct the defects. Corrective Maintenance activity may consist of repair, restoration or replacement of equipment. This activity will be result of a regular inspection, which identifies the failure in time for corrective maintenance to be planned and scheduled, then performed during a routine maintenance shutdown.

8.2.2 Adaptive Maintenance

Over time, the original environment(CPU, operating system, business rules, external product characteristics) for which the software was developed is likely to change. Adaptive maintenance results in modification to the software to accommodate changes to its external environment.

8.2.3 Enhanced Maintenance

As software is used, the customer/user will recognize additional functions that will provide the benefit. Perfect maintenance extends the software beyond its original functional requirements.

8.2.4 Preventive Maintenance

Computer software deteriorates due to change, and because of this preventive maintenance often called software re-engineering, must be conducted to enable the software to serve the needs of its end users. Preventive maintenance makes changes to computer programs so that they can be more easily corrected, adapted and enhanced.

Chapter 9

9 Conclusion and Future Scope

9.1 Conclusion

This system explores blockchain's impact on enhancing supply chain traceability and transparency, highlighting its transformative potential through real-world case studies. It acknowledges challenges like scalability, data privacy, and regulatory compliance. Ongoing research focuses on scalability solutions, interoperability standards, and privacy-enhancing technologies. Collaboration among academia, industry, and regulators is crucial. Blockchain's integration with AI/ML holds promise for optimizing supply chain processes. Additionally, advancements in cross-chain interoperability, energy-efficient consensus mechanisms, and integration with emerging technologies like IoT, big data analytics, and edge computing are key for improving supply chain management.

9.2 Future Scope

- Integration of DL with Blockchain
- Emerging Technology Integration

Chapter 10

10 References

1. Chen, X., Wang, L., & Wang, L. (2019). Blockchain-based Supply Chain Traceability: A Systematic Review. IEEE Transactions on Industrial Informatics, 15(8), 4608-4619.

2. Durst, S., Klußmann, S., & Arndt, H. (2021). Supply Chain Management 4.0: A Comprehensive Survey and Blockchain-Based Case Study. Sustainability, 13(6), 3210.

3. Idowu, A. (2020). Enhancing Food Traceability in the Supply Chain with Blockchain: A Case Study of Walmart. International Journal of Information Management, 55, 102177.

4. Kim, H. M., Laskowski, M., & Eksioglu, S. (2021). Enhancing Supply Chain Transparency with Blockchain Technology: A Case Study of Maersk's TradeLens Platform. International Journal of Logistics Management, 32(2), 274-291.

5. Marcella, R., Zimbra, D., & Micale, R. (2020). Blockchain for Ethical Sourcing in the Diamond Industry: A Case Study of De Beers' Tracr. Resources Policy, 68, 101814.

6. Mettler, M. (2016). Blockchain Technology in Healthcare: The Revolution Starts Here. IEEE Pulse, 8(4), 35-38.

7. Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. Retrieved from https://bitcoin.org/bitcoin.pdf

8. Kim, Y., & Laskowski, M. (2018). "Toward an ontology-driven block chain design for supply chain provenance." IEEE Transactions on Engineering Management, 65(4), 545-558.

9. Garg, S., & Baweja, K. (2019). "Blockchain-based supply chain management for sustainable business practices." Production Planning & Control, 30(11-12), 964-979.

10. Qu, W., Wu, B., & Wang, H. (2020). "A blockchain-based quality traceability system for agri-food supply chain." Journal of Food Engineering, 289, 110187.

11. Yao, L., Liu, Y., & Fan, S. (2019). "A blockchain-based supply chain quality management framework: A case study of an online retailer." International Journal of Production Research, 57(7), 2045-2063.

12. Tapscott, D., & Tapscott, A. (2016). "Blockchain revolution: how the technology behind bitcoin is changing money, business, and the world." Penguin.

13. Kshetri, N. (2018). "Will blockchain emerge as a tool to break the poverty chain in the Global South?" Third World Quarterly, 39(8), 1439-1458.

14. Al-Mansoori, A., & Al-Fuqaha, A. (2019). "Blockchain technology for social impact: Opportunities and challenges ahead." IEEE Transactions on Technology and Society, 1(1), 42-47.

15. Li, X., & Liang, G. (2017). "Blockchain-based system for secure data storage with private keyword search." IEEE Transactions on Services Computing, 11(5), 817-831.

16. Mengelkamp, E., Notheisen, B., & Weinhardt, C. (2018). "A blockchain based smart grid: towards sustainable local energy markets." Computer Science-Research and Development, 33(1-2), 207-214.

17. Li, W., & Mannan, M. (2017). "Securing internet of things with lightweight blockchain." IEEE Cloud Computing, 4(4), 32-39.

18. Christidis, K., & Devetsikiotis, M. (2016). "Blockchains and smart contracts for the internet of things." IEEE Access, 4, 2292-2303.

19. Noor, A. K., Salim, S. S., & Shuja, J. (2020). "An IoT and blockchain based food traceability system: A case study of Saudi Arabian dates." IEEE Access, 8, 190207-190218.

20. Zheng, Z., Xie, S., Dai, H. N., Chen, X., & Wang, H. (2018). "Blockchain challenges and opportunities: A survey." International Journal of Web and Grid Services, 14(4), 352-375.

21. Park, S., Lee, K., Kim, S., & Kim, H. (2019). "Blockchain-based supply chain traceability system for food safety." Electronics, 8(6), 668.

22. Tse, D. (2021). "Blockchain-Based Supply Chain Management: A Case Study in the Automotive Industry." Journal of Open Innovation: Technology, Market, and Complexity, 7(1), 38.

23. Zhang, C., Xu, H., & Zhao, M. (2018). "A new era of mass data and blockchain Internet of Things." IEEE Transactions on Industrial Informatics, 15(3), 1430-1438.

24. Jiao, S., Zhang, J., & Jiang, J. (2019). "A blockchain-based supply chain quality management framework: A case study of online retailer." International Journal of Production Research, 57(7), 2045-2063.

25. Korpela, K., Hallikas, J., & Dahlberg, T. (2017). "Digital supply chain transformation toward blockchain integration." Production Planning & Control, 28(11-12), 929-944.

Appendix

A Data Flow Diagram

Data flow is the one of the best way of documenting the entire functionality of the system.For the system, which will have data flows in and have some processing inside and then some data flow out from the system can be documented or represented effectively by means of data flow out from the system can be documented or represented effectively by means of data flow diagrams. The data flow diagram are a diagrammatic representation of the system,which has input,process and output.Once any system is represented using a data flow diagram we can identify the following things easily:

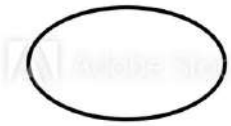
- Various entities interacting with the system are identified
- Flow of data from one entity to another is identified
- The various processes involved in between the interaction of two or more entities in the system are clearly pointed out
- The various data stores which hold the data in between the process,are clearly identified

Some Data Flow Diagram charting forms are given below:

A.1 External source or receiver

A source or sink is a person or part of organization ,which enters or receives information from the system,but is considered to be outlining the contest of data flow model.

A.2 Transform process

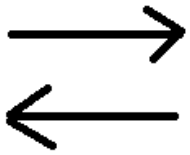


A process represents transformation where incoming data flows are changed into outgoing data flow.

A.3 Data Store

A data store is repository of data that is to be stored for use by one or more process may be as simple as buffer or queue or sophisticated as relational database. They should have clear names.If a process merely uses the contest of store and does not alter it,the arrowhead goes only form the store to the process. If a process alters the details in the store then double headed arrow is used.

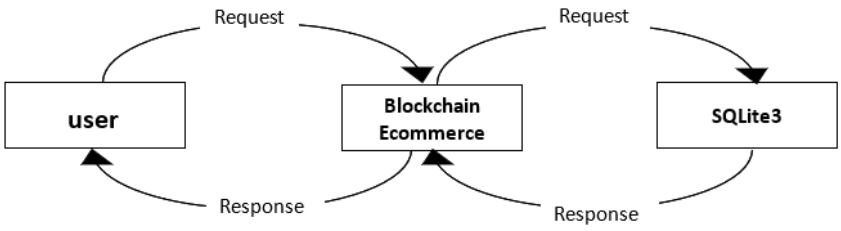
A.4 Data flow



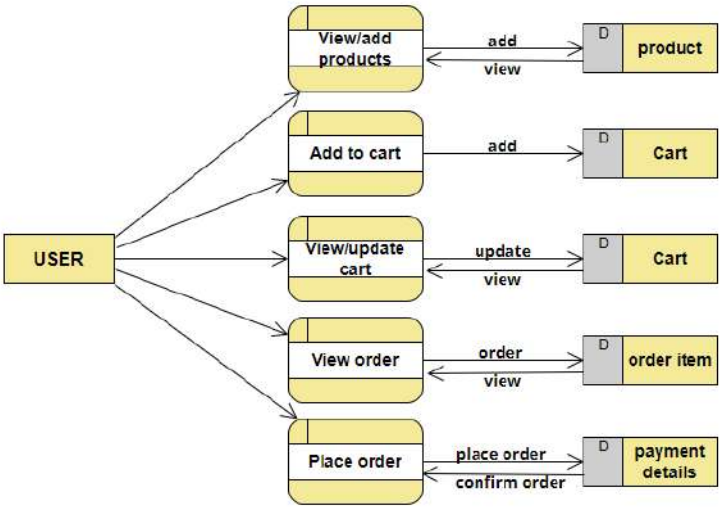
A data flow is a route, which enable packets of data to travel from one point to another.Data may flow,with arrowhead pointing in the direction of the flow.

A.5 Diagrams

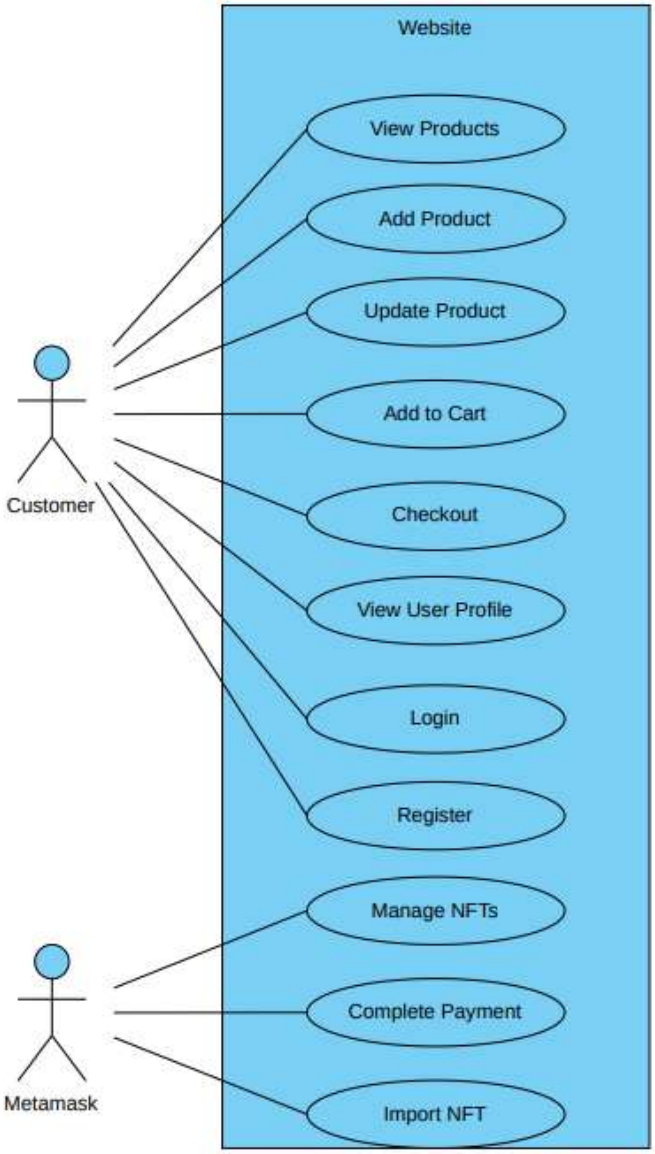
A.5.1 Level 0



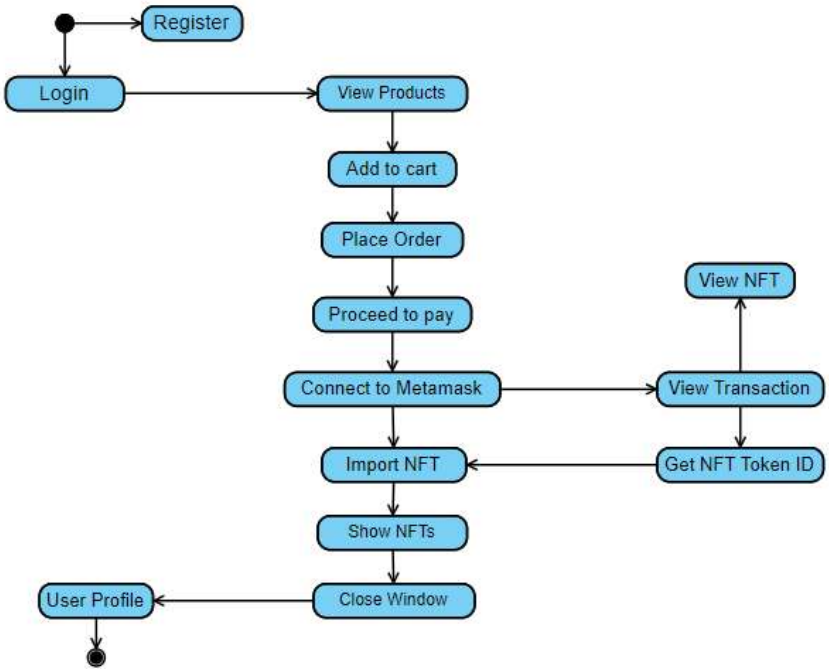
A.5.2 Level 1


USER
View/add products
Add to cart
View/update cart
View order
Place order
add
view
add
update
view
order
view
place order
confirm order
D
product
Cart
Cart
order item
payment details

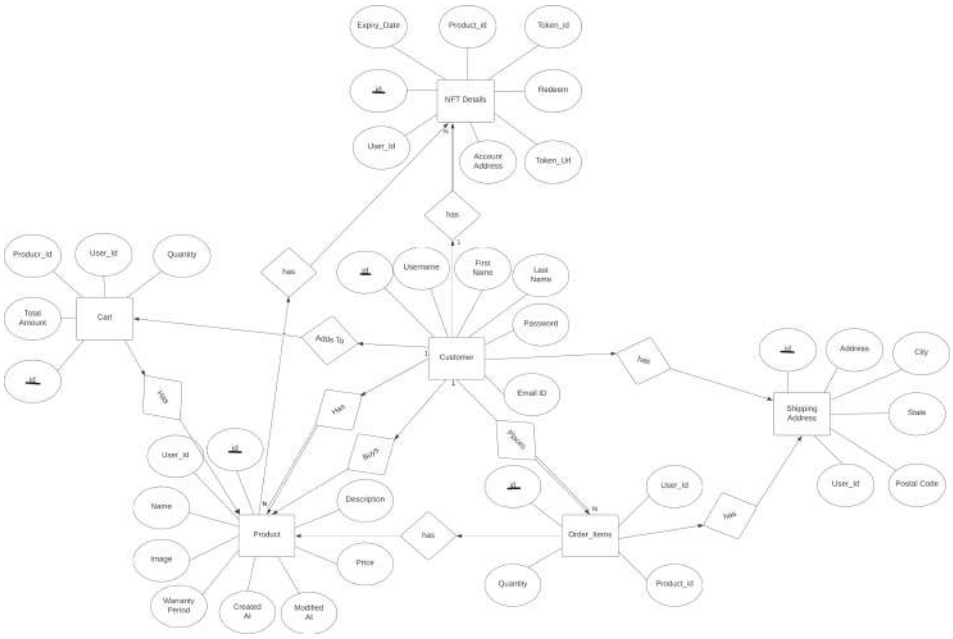
B Use Case Diagram



C Activity Diagram



D ER Diagram


Expiry_Date
Product_id
Token_id
NFT Details
Redeem
User_Id
Account Address
Token_Uri
has
Producr_id
User_Id
Quantity
Total Amount
Cart
Adds To
Customer
Username
First Name
Last Name
Password
Email ID
has
Address
City
State
Shipping Address
User_Id
Postal Code
Has
Has
Buys
Places
User_Id
Name
Description
Image
Price
Warranty Period
Created At
Modified At
Product
has
Order_Items
User_Id
Product_id
Quantity
has

E Front Page Of The IEEE Paper Published

2023 3rd International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON)
Karnataka, India. Dec 29-31, 2023

Unlocking Traceability and Transparency in Retail Supply Chains with Blockchain Technology

Devana Pramod
Department of Computer Science,
Christ college (Autonomous),
Irinjalakuda, Kerala-680125

Hana Nasreen
Department of Computer Science,
Christ college (Autonomous),
Irinjalakuda, Kerala-680125

Liya Johnson
Department of Computer Science,
Christ college (Autonomous),
Irinjalakuda, Kerala-680125

Megna Biju
Department of Computer Science,
Christ college (Autonomous),
Irinjalakuda, Kerala-680125

Priyanga K.K
Department of Computer Science,
Christ college (Autonomous),
Irinjalakuda, Kerala-680125
priyasumesh111@gmail.com

***Abstract*: Traceability has emerged as a prime requirement for a multi-tier and multi-site production. It enables visibility and caters to the consumer requirements of transparency and quality assurance. Textile and clothing industry is one such example that requires traceability implementation to address prevailing problems of information asymmetry and low visibility. Customers find it difficult to access product data that can facilitate ethical buying practices or assure product authenticity. Besides, it is challenging for stakeholders to share crucial information in an insecure environment with risk of data manipulations and fear of losing information advantage. In this context, this study investigates and proposes a blockchain-based traceability framework for traceability in multi-tier textile and clothing supply chain. It conceptualizes the interaction of supply chain partners, and related network architecture at the organizational level and smart contract and transaction validation rules at the operational level. To illustrate the application of the proposed framework, the study presents an example of organic cotton supply chain using blockchain with customized smart contract and transaction rules. It finally demonstrates the applicability of the developed blockchain by testing it under two parameters. The proposed system can build a technology-based trust among the supply chain partners, where the distributed ledger can be used to store and authenticate supply chain transactions. Further, the blockchain-based traceability system would provide a unique opportunity, flexibility, and authority to all partners to trace-back their supply network and create transparent and sustainable supply chain.**

Keywords: Blockchain, Traceability, Manufacturing, Textile and Clothing, Information sharing, Supply chain

I. Introduction

Supply chains are intricate systems critical to global operations, but they often lack transparency and traceability, leading to issues like disruptions, counterfeits, and ethical concerns. Blockchain technology, known for its decentralized and immutable record-keeping, offers a solution. This paper explores blockchain's potential in enhancing supply chain traceability and transparency across industries like manufacturing, healthcare, food, and luxury goods. [1] It analyses real-world case studies, discusses benefits and limitations, and aims to inform businesses, policymakers, and researchers about the transformative impact of blockchain on supply chains, emphasizing its role in building trust and improving operations.

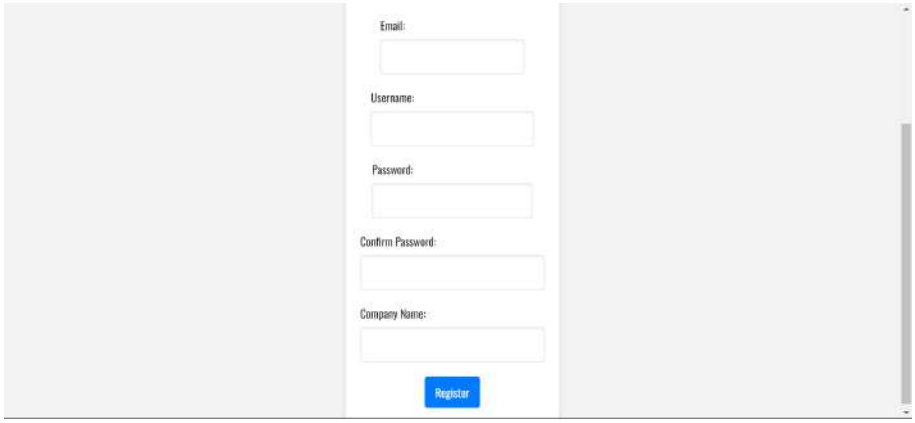
Industries worldwide are grappling with mounting pressure to transition towards sustainable production practices. The Textile and Clothing (T&C) sector, in particular, faces a unique set of challenges stemming from volatile consumer demands, intense competition, and the complexities of opaque supply chains. These issues have given rise to problems like product recalls and concerns over labour practices. [2] Enter blockchain technology, offering a robust solution. By facilitating transparent and secure traceability throughout the supply chain, blockchain not only enables better information sharing but also fosters trust among supply chain partners spread across the globe. What's noteworthy is that this blockchain-based approach is not limited to the T&C industry alone; with some adjustments, it can be applied to various other industries, making it a versatile tool for promoting sustainability and transparency across different sectors.

Additionally, existing methodologies for blockchain-based supply chain traceability and transparency primarily focus on leveraging blockchain's fundamental principles to create a transparent and immutable record of supply chain activities. These methodologies involve the deployment of blockchain networks to record every transaction and product movement, ensuring data transparency. [3] However, challenges related to scalability are often addressed through various consensus mechanisms and data partitioning techniques. While interoperability remains a concern, some methodologies suggest the use of standardized data formats and interoperability frameworks to facilitate communication between disparate blockchain networks. To tackle data privacy issues, existing approaches recommend employing encryption, off-chain storage, and data access controls to protect sensitive information. Smart contracts are commonly utilized to automate and enforce supply chain agreements, reducing manual intervention and enhancing transparency. Regulatory compliance is acknowledged as an essential aspect, with a focus on adapting to existing legal frameworks. [3] Some methodologies also propose conducting pilot projects to navigate regulatory challenges and refine blockchain-based solutions within regulatory boundaries. Overall, existing methodologies aim to create robust and transparent supply chains by integrating blockchain technology while considering scalability, interoperability, privacy, and regulatory aspects.

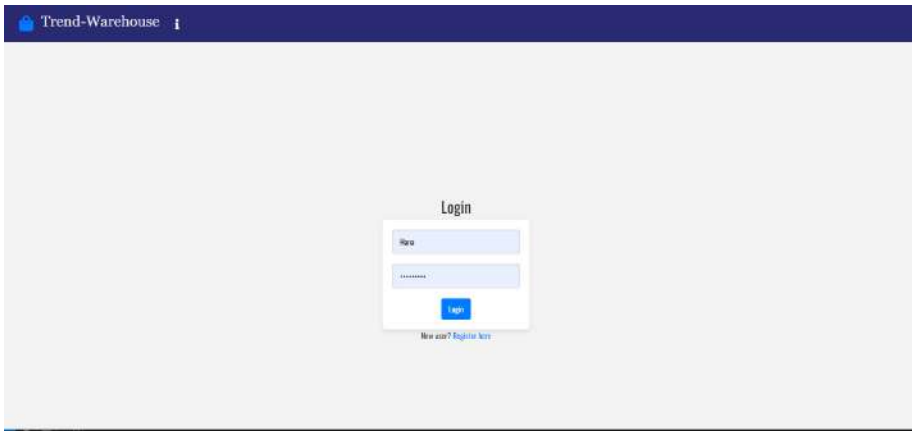
979-8-3503-1912-5/23/$31.00 ©2023 IEEE

F USER INTERFACES

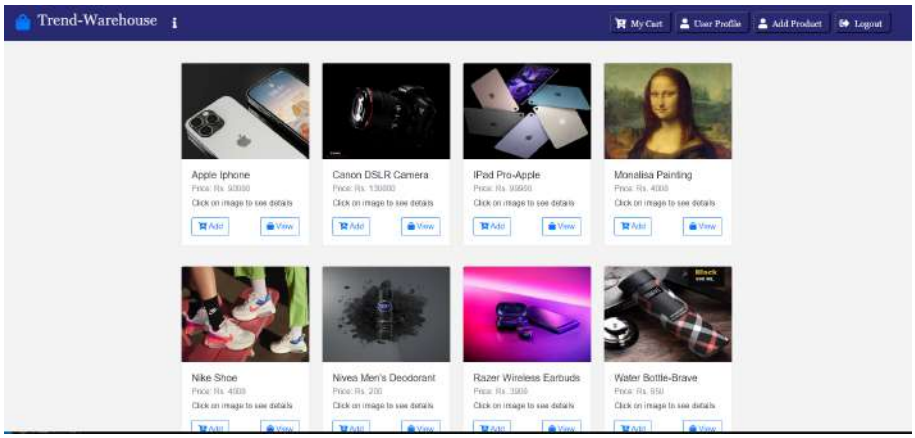
F.1 REGISTRATION



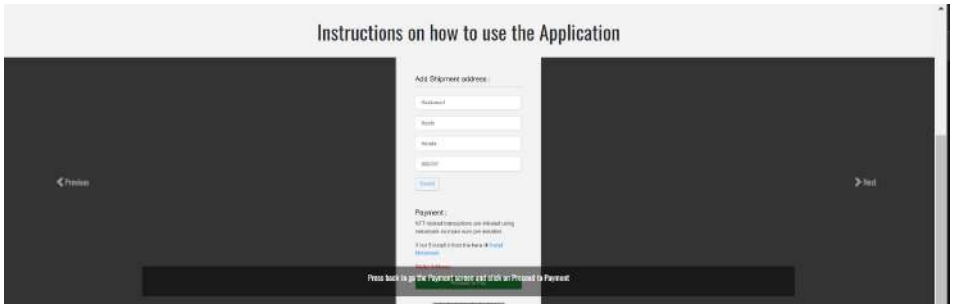
F.2 LOGIN



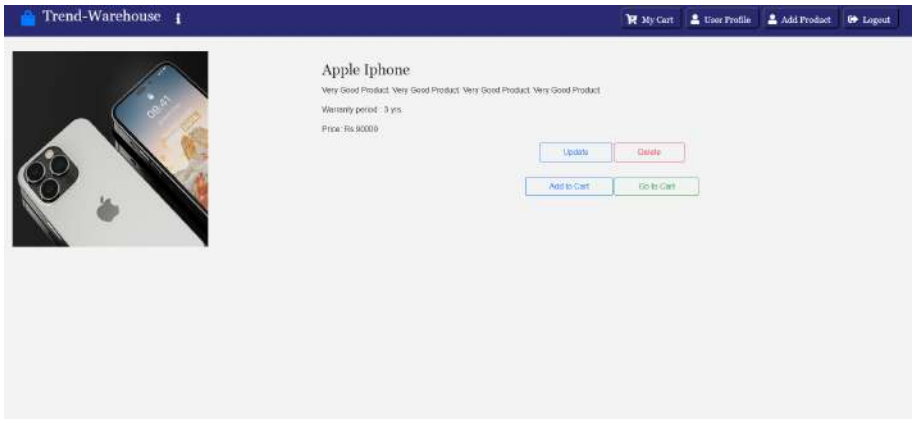
F.3 HOME


Trend-Warehouse
My Cart
User Profile
Add Product
Logout
Apple Iphone
Canon DSLR Camera
IPad Pro-Apple
Monalisa Painting
Nike Shoe
Nivea Men's Deodorant
Razer Wireless Earbuds
Water Bottle-Brave

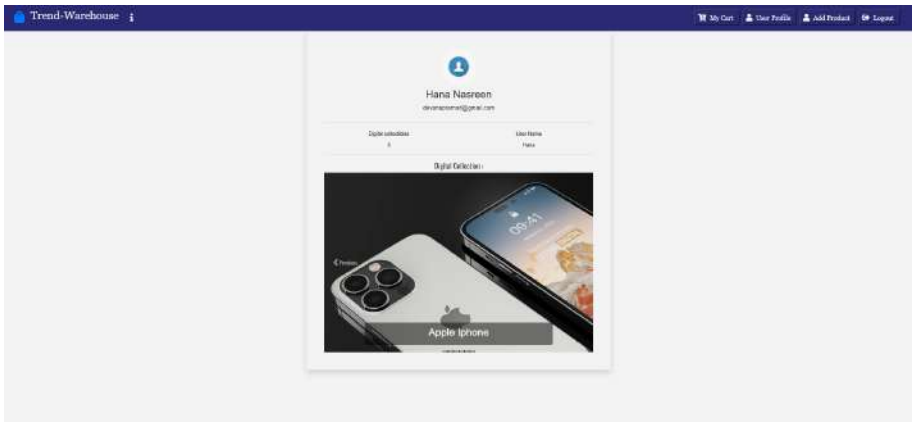
F.4 INSTRUCTIONS



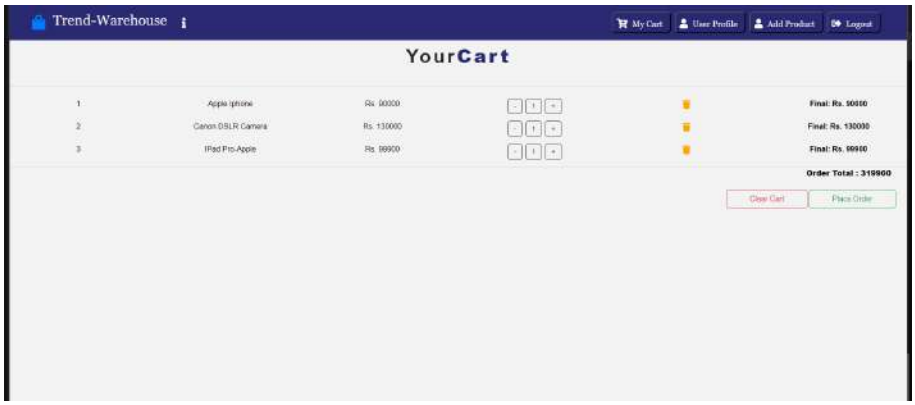
F.5 PRODUCT DESCRIPTION



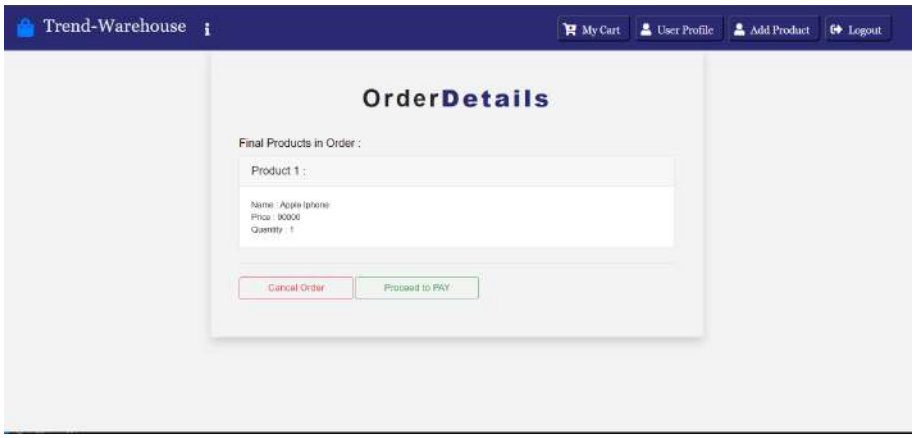
F.6 USER PROFILE



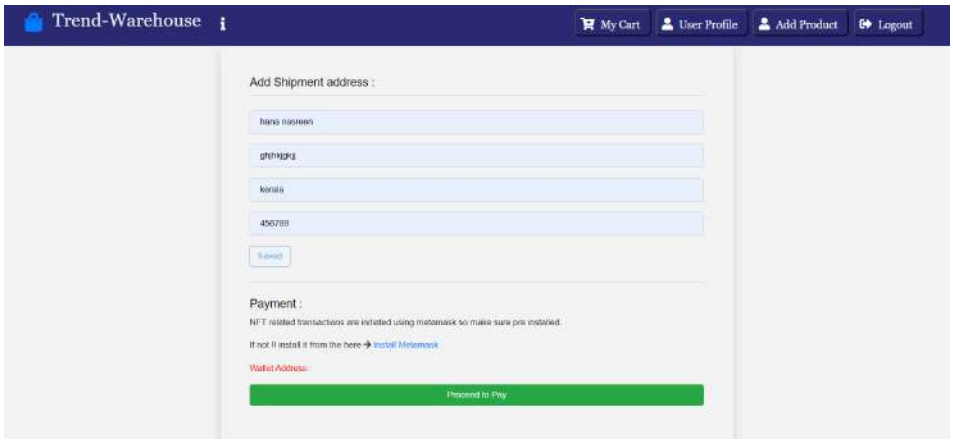
F.7 CART DETAILS

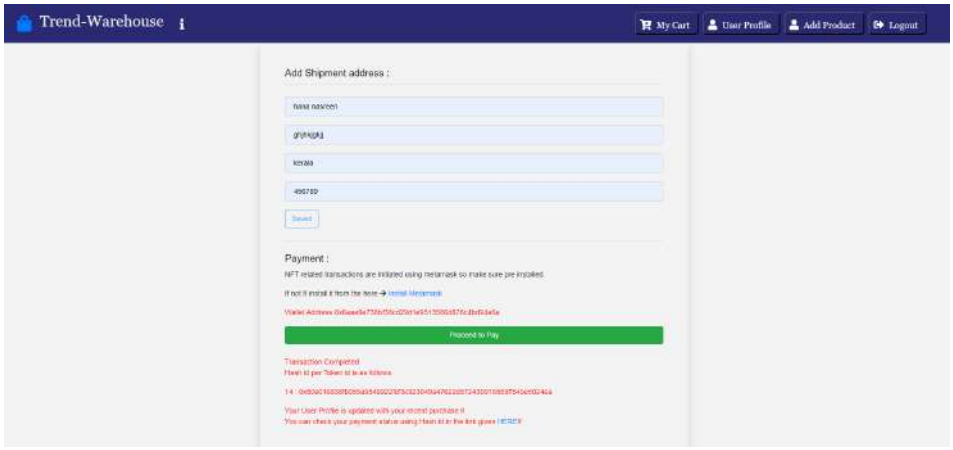


F.8 ORDER DETAILS

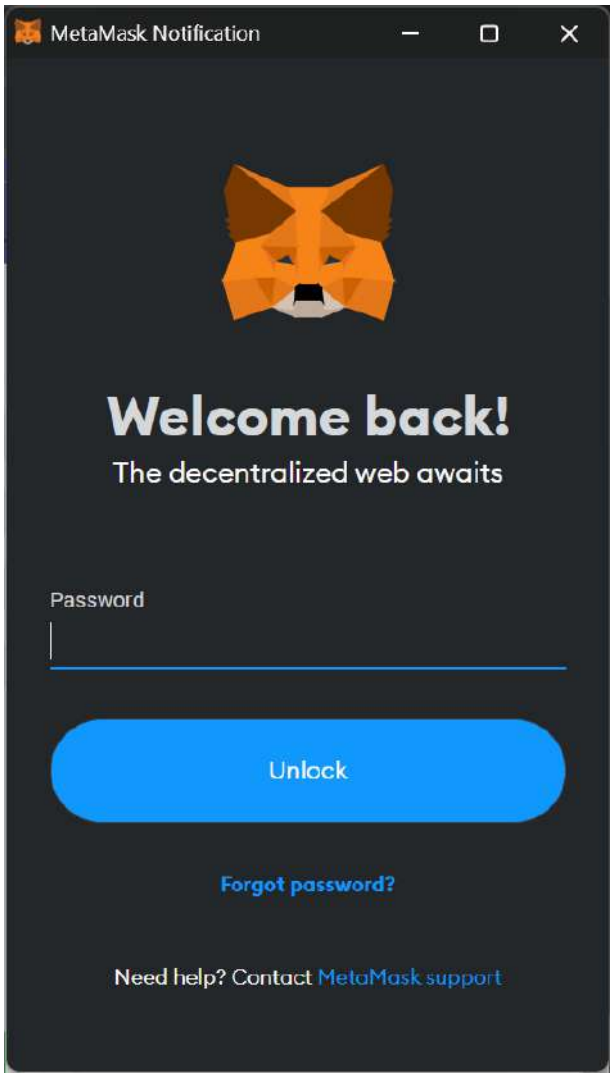


F.9 PAYMENT

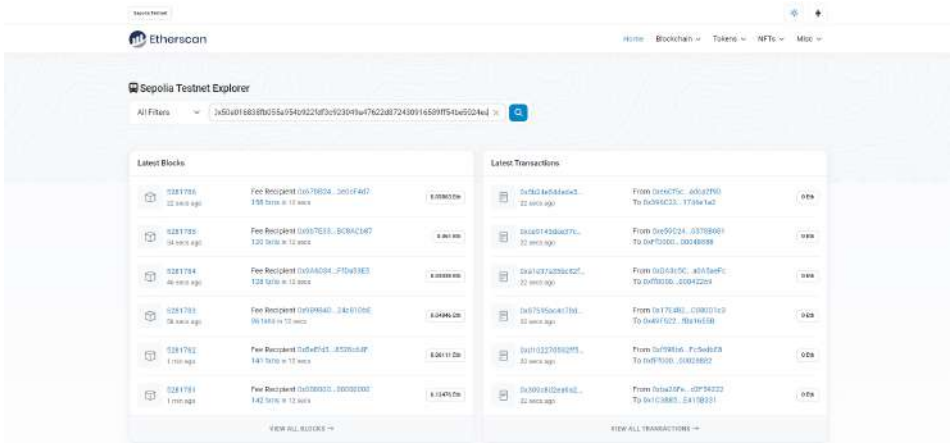

Trend-Warehouse
My Cart
User Profile
Add Product
Logout
Add Shipment address :
hana nasreen
ghhkgkg
kerala
456789
Saved
Payment :
NFT related transactions are initated using metamask so make sure pre installed.
If not ll install it from the here → Install Metamask
Wallet Address:
Proceed to Pay

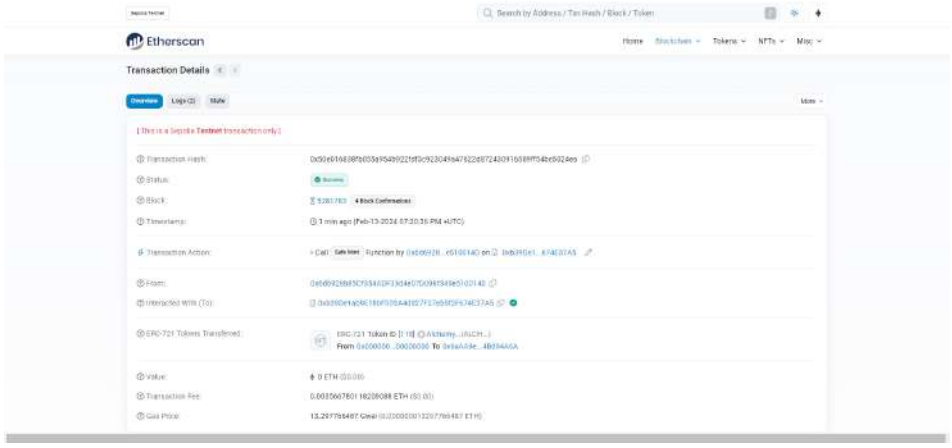

Trend-Warehouse
My Cart
User Profile
Add Product
Logout
Add Shipment address :
hana nasreen
ghhkgkg
kerala
456789
Saved
Payment :
Proceed to Pay

F.10 METAMASK

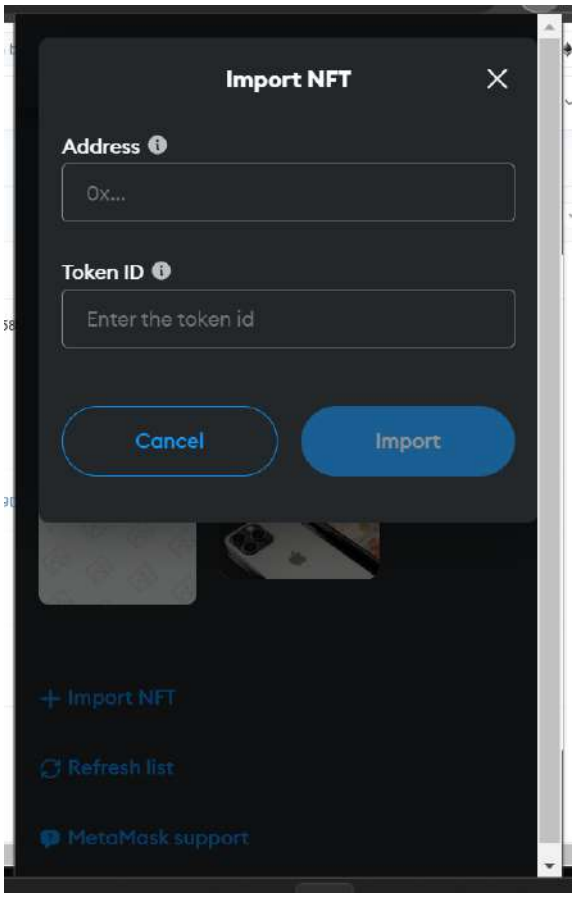


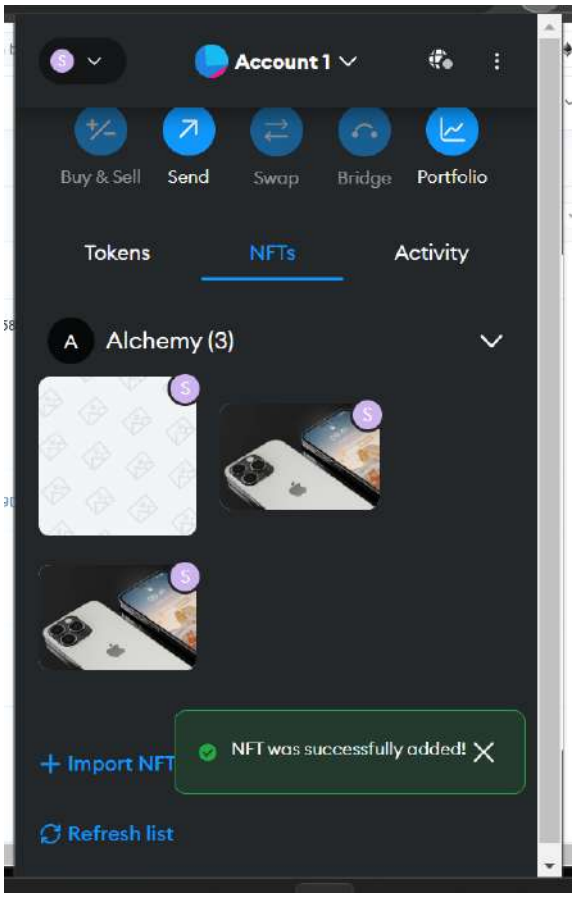
F.11 WALLET


Etherscan
Sepolia Testnet Explorer
Latest Blocks
Latest Transactions


Etherscan
Transaction Details

F.12 NFT DETAILS





G CODE

models.py

```
from django.db import models
from django.contrib.auth.hashers import make_password, check_password
from home.models import Account


class User(models.Model):
    User_type = (
        ('R', 'Retailer'),
        ('C', 'Customer'),
    )
    email = models.EmailField(max_length=60, unique=True)
    username = models.CharField(max_length=30, unique=True)
    first_name = models.CharField(max_length=30, null=True)
    last_name = models.CharField(max_length=30, null=True)
    date_joined = models.DateTimeField(auto_now_add=True)
    last_login = models.DateTimeField(auto_now=True)
    password = models.CharField(max_length=30)
    mobile_no = models.IntegerField(blank=True, null=True)
    company_name = models.CharField(max_length=50, null=True)

    def __str__(self):
        return self.username

    def set_password(self, raw_password):
        self.password = raw_password

    def check_password(self, raw_password):
        return raw_password==self.password


def get_product_image_filepath(self, filename):
    return 'product_images/' + str(self.pk) + '/product_image.png'


class Product(models.Model):
```

```
    name = models.CharField(max_length=60)
    price = models.FloatField()
    description = models.CharField(max_length=200, null=True)
    image = models.ImageField(max_length=255, upload_to=
                                        get_product_image_filepath,
         null=True, blank=True)
    warranty_period = models.IntegerField(default=0)
    created_at = models.DateTimeField(auto_now_add=True)
    modified_at = models.DateTimeField(auto_now=True)
    user_id = models.ForeignKey(User, on_delete=models.CASCADE, default=3)

    def get_product_image_filename(self):
        substring = f'product_images/{self.pk}/'
        if substring in self.image:
            result = self.image[str(self.image).index(substring):]
        else:
            result = None  # or some other default value or action

        return result

    def __str__(self):
        return self.name


class Order_Items(models.Model):
    product_id = models.ForeignKey(Product, on_delete=models.CASCADE)
    user_id = models.ForeignKey(User, on_delete=models.CASCADE)
    quantity = models.IntegerField()
    order_date = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return str(self.user_id)


class Payment_Details(models.Model):
    order_id = models.ForeignKey(Order_Items, on_delete=models.CASCADE)
    total_amount = models.FloatField()
    status = models.BooleanField(default=True)
    payment_date = models.DateTimeField(auto_now_add=True)
```

```
    def __str__(self):
        return str(self.status)


class Order_Details(models.Model):
    payment_id = models.ForeignKey(Payment_Details, on_delete=
        models.CASCADE)

    def __str__(self):
        return str(self.payment_id)
```

views.py

```
from django.shortcuts import render
from rest_framework import status, generics
from rest_framework.permissions import AllowAny
from .models import User
from .serializers import *
```

urls.py

```
from django.urls import path,include
from . import views
from rest_framework import routers

urlpatterns = [
    path('product/',views.product_list),
    path('product/<int:id>/add/',views.AddProduct),
    path('product/<int:id>/',views.product_detail),
    path('product/<int:rid>/<int:pid>/update/',views.update_product_detail),
    path('product/<int:id>/delete/',views.delete_product),
    path('cart/<int:id>/',views.cart_list),
    path('cart/<int:uid>/<int:pid>/delete/',views.remove_item),
    path('cart/<int:uid>/<int:pid>/add/',views.AddItem),
    path('cart/<int:id>/clear/',views.clear_cart),
    path('cart/<int:uid>/<int:pid>/',views.update_quant_and_total),
    path('order/<int:id>/',views.order_items),
```

```
    path('order/<int:id>/add/',views.AddOrder),
    path('order/<int:id>/ship/',views.ship_address),
    path('order/<int:id>/detail/',views.order_details),
    path('cart/<int:uid>/<int:pid>/dec/',views.dec_quant_and_total),
    path('cart/<int:id>/total/',views.total_cart_price),
    path('order/<int:id>/cancel/',views.cancel_order),
    path('token/<int:uid>/<str:id>/',views.pinata_file_upload),
    path('user/<int:id>/',views.user_detail),
    path('login/', views.login_view),
    path('register/', views.RegisterView.as_view()),
    path('nft/<int:id>/',views.valid_nft),
    path('nft/<int:tid>/<str:uname>/<str:id>/update/',views.update_nft_detail),
    path('warranty/<int:tid>/',views.update_warranty),
    path('redeem/<int:tid>/',views.update_redeem),
]
```

views.py

```
from django.contrib.auth import authenticate, login
from django.http import JsonResponse
from django.views.decorators.csrf import csrf_exempt
from django.views.decorators.http import require_http_methods
from user.serializers import UserSerializer, ProductSerializer,
 Order_ItemsSerializer, Payment_DetailsSerializer, \
    Order_DetailsSerializer, RegisterSerializer
from user.models import User, Product, Order_Items,
 Payment_Details, Order_Details
from product.models import Cart, Shipping_Address, Track_Repairs, NFT_Details
from product.serializer import CartSerializer, Shipping_AddressSerializer,
 Track_RepairsSerializer, \
    NFT_DetailsSerializer
from rest_framework.decorators import api_view
from rest_framework.views import APIView
from rest_framework.response import Response
from rest_framework import status
from django.db.models import F
import random, requests, json
from web3 import Web3, HTTPProvider
from web3.gas_strategies.rpc import rpc_gas_price_strategy
```

```
from datetime import datetime, timedelta, timezone
from web3.middleware import geth_poa_middleware
from dateutil import relativedelta
from django.core.mail import send_mail
from django.conf import settings
import re

@csrf_exempt
@require_http_methods(["POST"])
def login_view(request):
    data = json.loads(request.body)
    username = data.get('username')
    password = data.get('password')
    user = authenticate(request, username=username, password=password)
    if user is not None:
        login(request, user)
        return JsonResponse({"message": "Login successful!", "user_id": user.id},
 status=200)
    else:
        return JsonResponse({"message": "Invalid username or password."},
   status=400)


class RegisterView(APIView):
    def post(self, request, format='json'):
        serializer = RegisterSerializer(data=request.data)
        if serializer.is_valid():
            user = serializer.save()
            if user:
           return Response(serializer.data, status=status.HTTP_201_CREATED)
       return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)


@api_view(['GET'])
def user_detail(request, id):
    try:
        user = User.objects.get(pk=id)
    except User.DoesNotExist:
        return Response(status=status.HTTP_404_NOT_FOUND)
    serializer = UserSerializer(user)
```

```
    return Response(serializer.data)


@api_view(['GET'])
def product_list(request):
    product = Product.objects.all().order_by('name')
    serializer = ProductSerializer(product, many=True)
    return JsonResponse(serializer.data, safe=False)


@api_view(['POST'])
def AddProduct(request, id):
    data = request.data.copy()
    data['user_id'] = id
    serializer = ProductSerializer(data=data)
    if serializer.is_valid():
        serializer.save()
      return Response(serializer.data, status=status.HTTP_201_CREATED)
    else:
      return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)


@api_view(['GET'])
def product_detail(request, id):
    try:
        product = Product.objects.get(pk=id)
    except Product.DoesNotExist:
        return Response(status=status.HTTP_404_NOT_FOUND)

    serializer = ProductSerializer(product)
    return Response(serializer.data)


@api_view(['PUT'])
def update_product_detail(request, rid, pid):
    try:
        product = Product.objects.get(pk=pid)
    except Product.DoesNotExist:
        return Response(status=status.HTTP_404_NOT_FOUND)
    data = request.data.copy()
```

```
    data['user_id'] = rid
    serializer = ProductSerializer(product, data=data)
    if serializer.is_valid():
        serializer.save()
        return Response(serializer.data)
    return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)


@api_view(['DELETE'])
def delete_product(request, id):
    try:
        product = Product.objects.get(pk=id)
    except Product.DoesNotExist:
        return Response(status=status.HTTP_404_NOT_FOUND)

    product.delete()
    return Response(status=status.HTTP_204_NO_CONTENT)


@api_view(['GET'])
def cart_list(request, id):
    try:
        cart = Cart.objects.filter(user_id=id)
    except Cart.DoesNotExist:
        return Response(status=status.HTTP_404_NOT_FOUND)
    serializer = CartSerializer(cart, many=True)
    ls = []
    total = 0
    for i in serializer.data:
        dict1 = {}
        product = Product.objects.get(id=i['product_id'])
        dict1['id'] = i['id']
        dict1['date_added'] = i['date_added']
        dict1['product_id'] = i['product_id']
        dict1['user_id'] = i['user_id']
        dict1['name'] = product.name
        dict1['price'] = product.price
        dict1['quantity'] = i['quantity']
        dict1['total_amount'] = i['total_amount']
        dict1['inCart'] = True
```

```
        ls.append(dict1)
    return Response(ls)


@api_view(['GET'])
def total_cart_price(request, id):
    try:
        cart = Cart.objects.filter(user_id=id)
    except Cart.DoesNotExist:
        return Response(status=status.HTTP_404_NOT_FOUND)
    serializer = CartSerializer(cart, many=True)
    ls = []
    total = 0
    for i in serializer.data:
        dict1 = {}
        product = Product.objects.get(id=i['product_id'])
        dict1['id'] = i['id']
        dict1['date_added'] = i['date_added']
        dict1['product_id'] = i['product_id']
        dict1['user_id'] = i['user_id']
        dict1['name'] = product.name
        dict1['price'] = product.price
        dict1['quantity'] = i['quantity']
        dict1['total_amount'] = i['total_amount']
        total += i['total_amount']
        ls.append(dict1)
    final = []
    dict2 = {'total': total}
    final.append(dict2)
    return Response(final)


@api_view(['DELETE'])
def remove_item(request, uid, pid):
    try:
        cart = Cart.objects.filter(user_id=uid, product_id=pid)
    except Cart.DoesNotExist:
        return Response(status=status.HTTP_404_NOT_FOUND)
    cart.delete()
    return Response(status=status.HTTP_204_NO_CONTENT)
```

```
@api_view(['POST'])
def AddItem(request,uid,pid):
    request.data['user_id']=uid
    request.data['product_id']=pid
    request.data['quantity'] = 1
    product = Product.objects.get(pk=pid)
    serializer1 = ProductSerializer(product)
    request.data['total_amount']=serializer1.data['price']
    serializer=CartSerializer(data=request.data)
    if serializer.is_valid():
        serializer.save()
        return Response(serializer.data,status=status.HTTP_201_CREATED)


@api_view(['DELETE'])
def clear_cart(request, id):
    try:
        cart = Cart.objects.filter(user_id=id)
    except Cart.DoesNotExist:
        return Response(status=status.HTTP_404_NOT_FOUND)
    cart.delete()
    return Response(status=status.HTTP_204_NO_CONTENT)


@api_view(['PUT'])
def update_quant_and_total(request, uid, pid):
    try:
        cart = Cart.objects.filter(user_id=uid)
    except Cart.DoesNotExist:
        return Response(status=status.HTTP_404_NOT_FOUND)

    serializer = CartSerializer(cart, many=True)
    updated_cart_data = []

    for cart_item in serializer.data:
        if cart_item['product_id'] == pid:
            cart_instance = Cart.objects.get(id=cart_item['id'])
            product = Product.objects.get(id=cart_item['product_id'])
```

```
            new_quantity = cart_item['quantity'] + 1
            new_total_amount = cart_item['total_amount'] + product.price

            # Prevent quantity from going below zero
            if new_quantity >= 0:
                cart_instance.quantity = new_quantity
                cart_instance.total_amount = new_total_amount
                updated_cart_data.append(cart_instance)

    if updated_cart_data:
        request.data['user_id'] = uid
        request.data['product_id'] = pid
        request.data['quantity'] = max(0, updated_cart_data[0].quantity)
 # Ensure quantity is not negative
        request.data['total_amount'] = updated_cart_data[0].total_amount

        serializer = CartSerializer(updated_cart_data[0], data=request.data)

    if serializer.is_valid():
        serializer.save()
        return Response(serializer.data)
    return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)


@api_view(['PUT'])
def dec_quant_and_total(request, uid, pid):
    try:
        cart = Cart.objects.filter(user_id=uid)
    except Cart.DoesNotExist:
        return Response(status=status.HTTP_404_NOT_FOUND)
    serializer = CartSerializer(cart, many=True)
    ls = []
    for i in serializer.data:
        if i['product_id'] == pid:
            if i['quantity'] > 1:
                dict1 = {}
                cart1 = Cart.objects.get(id=i['id'])
                product = Product.objects.get(id=i['product_id'])
                dict1['quantity'] = i['quantity'] - 1
```

```
                dict1['total_amount'] = i['total_amount'] - product.price
                ls.append(dict1)
    request.data['user_id'] = uid
    request.data['product_id'] = pid
    request.data['quantity'] = ls[0]['quantity']
    request.data['total_amount'] = ls[0]['total_amount']
    serializer = CartSerializer(cart1, data=request.data)
    if serializer.is_valid():
        serializer.save()
        return Response(serializer.data)
    return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)



@api_view(['GET'])
def order_items(request, id):
    order = Order_Items.objects.filter(user_id=id)
    serializer = Order_ItemsSerializer(order, many=True)
    ls = []
    for i in serializer.data:
        dict1 = {}
        product = Product.objects.get(id=i['product_id'])
        dict1['id'] = i['id']
        dict1['user_id'] = i['user_id']
        dict1['order_date'] = i['order_date']
        dict1['name'] = product.name
        dict1['price'] = product.price * i['quantity']
        dict1['quantity'] = i['quantity']
        ls.append(dict1)
    return Response(ls)


@api_view(['POST'])
def AddOrder(request, id):
    try:
        cart = Cart.objects.filter(user_id=id)
        order = Order_Items.objects.filter(user_id=id)
    except Cart.DoesNotExist or Order_Items.DoesNotExist:
        return Response(status=status.HTTP_404_NOT_FOUND)
    order.delete()
```

```
    serializer = CartSerializer(cart, many=True)
    ls = []
    for i in serializer.data:
        dict1 = {}
        dict1['product_id'] = i['product_id']
        dict1['user_id'] = i['user_id']
        dict1['quantity'] = i['quantity']
        ls.append(dict1)
    for i in range(len(ls)):
        request.data['user_id'] = ls[i]['user_id']
        request.data['product_id'] = ls[i]['product_id']
        request.data['quantity'] = ls[i]['quantity']
        serializer = Order_ItemsSerializer(data=request.data)
        if serializer.is_valid():
            serializer.save()
    return Response(ls)


@api_view(['DELETE'])
def cancel_order(request, id):
    try:
        order = Order_Items.objects.filter(user_id=id)
    except Order_Items.DoesNotExist:
        return Response(status=status.HTTP_404_NOT_FOUND)
    order.delete()
    return Response(status=status.HTTP_204_NO_CONTENT)


@api_view(['POST'])
def ship_address(request, id):
    try:
        ship = Shipping_Address.objects.filter(user_id=id)
        order = Order_Items.objects.filter(user_id=id)
        ship.delete()
    except Order_Items.DoesNotExist:
        return Response(status=status.HTTP_404_NOT_FOUND)
    request.data['user_id'] = id
    serializer = Shipping_AddressSerializer(data=request.data)
    if serializer.is_valid():
        serializer.save()
```

```
        return Response(serializer.data)
    return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)


@api_view(['POST'])
def order_details(request, id):
    request.data['payment_id'] = id
    serializer = Order_DetailsSerializer(data=request.data)
    if serializer.is_valid():
        serializer.save()
        return Response(serializer.data, status=status.HTTP_201_CREATED)


@api_view(['POST'])
def pinata_file_upload(request, uid, id):
    oorder = Order_Items.objects.filter(user_id=uid)
    oserializer = Order_ItemsSerializer(oorder, many=True)
    ls = []
    for i in oserializer.data:
        dicto = {}
        product = Product.objects.get(id=i['product_id'])
        dicto['id'] = i['id']
        dicto['product_id'] = i['product_id']
        dicto['user_id'] = i['user_id']
        dicto['order_date'] = i['order_date']
        dicto['name'] = product.name
        dicto['price'] = product.price * i['quantity']
        dicto['quantity'] = i['quantity']
        ls.append(dicto)
    final = {}
    try1 = []
    for i in ls:
        print('outer')
        finallist = []
        try2 = {}
        for j in range(i['quantity']):
            print('inner')
            product = Product.objects.filter(pk=i['product_id'])
            serializer = ProductSerializer(product, many=True)
            img_name = serializer.data[0]['image'].split('/')[-1]
```

```
            file_path = "D:/blockchain 2nd part/Backend/blockchain" + \
                        serializer.data[0]['image']
            url = "https://api.pinata.cloud/pinning/pinFileToIPFS"
            payload = {}
            files = [
                ('file', (img_name, open(file_path, 'rb'), 'image/png'))
            ]
            headers = {
             'Authorization': 'Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c
            2VySW5mb3JtYXRpb24iOnsiaWQiOiJkYTk2OTc4ZS02NWJiLTQwOGEtYTAzMS0yZGFh
            N2ViY2IyMDMiLCJlbWFpbCI6ImpvZWxzYW1qb2huc29uQHlhaG9vLmNvbSIsImVtYWls
            X3ZlcmlmaWVkIjp0cnVlLCJwaW5fcG9saWN5Ijp7InJlZ2lvbnMiOlt7ImlkIjoiRlJB
            MSIsImRlc2lyZWRSZXBsaWNhdGlvbkNvdW50IjoxfSx7ImlkIjoiTllDMSIsImRlc2l
            yZWRSZXBsaWNhdGlvbkNvdW50IjoxfV0sInZlcnNpb24iOjF9LCJtZmFfZW5hYmxlZ
            CI6ZmFsc2UsInN0YXR1cyI6IkFDVElWRSJ9LCJhdXRoZW50aWNhdGlvblR5cGU
            iOiJzY29wZWRLZXkiLCJzY29wZWRLZXlLZXkiOiI3MWEzNjJkYjMxMGY4YzdlNTB
            kMyIsInNjb3BlZEtleVNlY3JldCI6IjM2MjllZGJhMzc2Y2E0MDkwZWRiMzg4NjE2Z
            TA2MTFhODJjYzRjYjIxNTU2ZjIxMWVkZWQzZjA2ZGNmYWJjNzUiLCJpYXQiOjE
          3MDc4NDE1Njh9.Xhc2XxMBq5dIKWW-593JBH571oTMVxgtx4Op5Tt6KWE'
            }

            response = requests.request("POST", url, headers=headers,
data=payload, files=files)
            hashval = response.text.split('"')[3]
            url = "https://api.pinata.cloud/pinning/pinJSONToIPFS"
            img_url = "https://amethyst-realistic-camel-859.mypinata.cloud
/ipfs/" + hashval + "?pinataGatewayToken=MgL8ssK-
torTK_vfvilq04TZ6BpK1cdpQFCc6fRLD-K8dQ4g8f2SplukNJ-sGXrJ"
            Product_name = serializer.data[0]['name']
            time = serializer.data[0]['warranty_period'] * 31556952
            x = datetime.now() + timedelta(seconds=time)
            payload = json.dumps({
                "description": "Token URL Generation",
                "external_url": img_url,
                "image": img_url,
                "name": Product_name + "-Warranty NFT",
                "attributes": [
                    {
                        "trait_type": "Expiry",
                        "value": str(x)
```

```
                    }
                ]
            })
            headers = {
              'Authorization': 'Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1
          c2VySW5mb3JtYXRpb24iOnsiaWQiOiJkYTk2OTc4ZS02NWJiLTQwOGEtYTAzMS0yZGFh
          2ViY2IyMDMiLCJlbWFpbCI6ImpvZWxzYW1qb2huc29uQHlhaG9vLmNvbSIsImVtY
           WlsX3ZlcmlmaWVkIjp0cnVlLCJwaW5fcG9saWN5Ijp7InJlZ2lvbnMiOlt7ImlkI
           joiRlJBMSIsImRlc2lyZWRSZXBsaWNhdGlvbkNvdW50IjoxfSx7ImlkIjoiTllDM
          SIsImRlc2lyZWRSZXBsaWNhdGlvbkNvdW50IjoxfV0sInZlcnNpb24iOjF9LCJtZmFfZW5
          hYmxlZCI6ZmFsc2UsInN0YXR1cyI6IkFDVElWRSJ9LCJhdXRoZW50aWNhdGlvblR5cG
          UiOiJzY29wZWRLZXkiLCJzY29wZWRLZXlLZXkiOiI3MWEzNjJkYjMxMGY4YzdlNTBkMy
          IsInNjb3BlZEtleVNlY3JldCI6IjM2MjllZGJhMzc2Y2E0MDkwZWRiMzg4NjE2ZTA2M
          TFhODJjYzRjYjIxNTU2ZjIxMWVkZWQzZjA2ZGNmYWJjNzUiLCJpYXQiOjE3MDc4
            NDE1Njh9.Xhc2XxMBq5dIKWW-593JBH571oTMVxgtx40p5Tt6KWE',
                'Content-Type': 'application/json'
            }

            response = requests.request("POST", url, headers=headers,
        data=payload)
            hashval2 = response.text.split('"')[3]
            json_url = "https://amethyst-realistic-camel-859.mypinata.cloud
/ipfs/" + hashval2 + "?pinataGatewayToken=MgL8ssK-torTK_vfvilq04
TZ6BpK1cdpQFCc6fRLD-K8dQ4g8f2SplukNJ-sGXrJ"

            w3 = Web3(HTTPProvider('https://eth-sepolia.g.alchemy.com/v2/
GtRs0mmRk7tpkqbRpthhc5HGQAi8wDGC'))
            w3.middleware_onion.inject(geth_poa_middleware, layer=0)
            abi = '''[
    {
        "inputs": [],
        "stateMutability": "nonpayable",
        "type": "constructor"
    },
    {
        "anonymous": false,
        "inputs": [
            {
                "indexed": true,
                "internalType": "address",
```

```
                "name": "owner",
                "type": "address"
            },
            {
                "indexed": true,
                "internalType": "address",
                "name": "approved",
                "type": "address"
            },
            {
                "indexed": true,
                "internalType": "uint256",
                "name": "tokenId",
                "type": "uint256"
            }
        ],
        "name": "Approval",
        "type": "event"
    },
    {
        "anonymous": false,
        "inputs": [
            {
                "indexed": true,
                "internalType": "address",
                "name": "owner",
                "type": "address"
            },
            {
                "indexed": true,
                "internalType": "address",
                "name": "operator",
                "type": "address"
            },
            {
                "indexed": false,
                "internalType": "bool",
                "name": "approved",
                "type": "bool"
            }
```

```
        ],
        "name": "ApprovalForAll",
        "type": "event"
    },
    {
        "anonymous": false,
        "inputs": [
            {
                "indexed": true,
                "internalType": "address",
                "name": "previousOwner",
                "type": "address"
            },
            {
                "indexed": true,
                "internalType": "address",
                "name": "newOwner",
                "type": "address"
            }
        ],
        "name": "OwnershipTransferred",
        "type": "event"
    },
    {
        "anonymous": false,
        "inputs": [
            {
                "indexed": true,
                "internalType": "address",
                "name": "from",
                "type": "address"
            },
            {
                "indexed": true,
                "internalType": "address",
                "name": "to",
                "type": "address"
            },
            {
                "indexed": true,
```

```
                "internalType": "uint256",
                "name": "tokenId",
                "type": "uint256"
            }
        ],
        "name": "Transfer",
        "type": "event"
    },
    {
        "inputs": [
            {
                "internalType": "uint256",
                "name": "tid",
                "type": "uint256"
            }
        ],
        "name": "applyExpiryDiscount",
        "outputs": [],
        "stateMutability": "nonpayable",
        "type": "function"
    },
    {
        "inputs": [
            {
                "internalType": "address",
                "name": "to",
                "type": "address"
            },
            {
                "internalType": "uint256",
                "name": "tokenId",
                "type": "uint256"
            }
        ],
        "name": "approve",
        "outputs": [],
        "stateMutability": "nonpayable",
        "type": "function"
    },
    {
```

```
        "inputs": [
            {
                "internalType": "address",
                "name": "owner",
                "type": "address"
            }
        ],
        "name": "balanceOf",
        "outputs": [
            {
                "internalType": "uint256",
                "name": "",
                "type": "uint256"
            }
        ],
        "stateMutability": "view",
        "type": "function"
    },
    {
        "inputs": [
            {
                "internalType": "uint256",
                "name": "tokenId",
                "type": "uint256"
            }
        ],
        "name": "getApproved",
        "outputs": [
            {
                "internalType": "address",
                "name": "",
                "type": "address"
            }
        ],
        "stateMutability": "view",
        "type": "function"
    },
    {
        "inputs": [
            {
```

```
                "internalType": "uint256",
                "name": "startwarr",
                "type": "uint256"
            },
            {
                "internalType": "uint256",
                "name": "warrPeriod",
                "type": "uint256"
            }
        ],
        "name": "getTime",
        "outputs": [
            {
                "internalType": "uint256",
                "name": "",
                "type": "uint256"
            }
        ],
        "stateMutability": "view",
        "type": "function"
    },
    {
        "inputs": [
            {
                "internalType": "address",
                "name": "owner",
                "type": "address"
            },
            {
                "internalType": "address",
                "name": "operator",
                "type": "address"
            }
        ],
        "name": "isApprovedForAll",
        "outputs": [
            {
                "internalType": "bool",
                "name": "",
                "type": "bool"
```

```
            }
        ],
        "stateMutability": "view",
        "type": "function"
    },
    {
        "inputs": [],
        "name": "name",
        "outputs": [
            {
                "internalType": "string",
                "name": "",
                "type": "string"
            }
        ],
        "stateMutability": "view",
        "type": "function"
    },
    {
        "inputs": [],
        "name": "owner",
        "outputs": [
            {
                "internalType": "address",
                "name": "",
                "type": "address"
            }
        ],
        "stateMutability": "view",
        "type": "function"
    },
    {
        "inputs": [
            {
                "internalType": "uint256",
                "name": "tokenId",
                "type": "uint256"
            }
        ],
        "name": "ownerOf",
```

```
        "outputs": [
            {
                "internalType": "address",
                "name": "",
                "type": "address"
            }
        ],
        "stateMutability": "view",
        "type": "function"
    },
    {
        "inputs": [],
        "name": "renounceOwnership",
        "outputs": [],
        "stateMutability": "nonpayable",
        "type": "function"
    },
    {
        "inputs": [
            {
                "internalType": "uint256",
                "name": "tid",
                "type": "uint256"
            }
        ],
        "name": "replacement",
        "outputs": [
            {
                "internalType": "uint256",
                "name": "",
                "type": "uint256"
            }
        ],
        "stateMutability": "nonpayable",
        "type": "function"
    },
    {
        "inputs": [
            {
                "internalType": "address",
```

```
                "name": "to",
                "type": "address"
            },
            {
                "internalType": "uint256",
                "name": "tokenId",
                "type": "uint256"
            },
            {
                "internalType": "uint256",
                "name": "warrantyDuration",
                "type": "uint256"
            },
            {
                "internalType": "string",
                "name": "uri",
                "type": "string"
            }
        ],
        "name": "safeMint",
        "outputs": [],
        "stateMutability": "nonpayable",
        "type": "function"
    },
    {
        "inputs": [
            {
                "internalType": "address",
                "name": "from",
                "type": "address"
            },
            {
                "internalType": "address",
                "name": "to",
                "type": "address"
            },
            {
                "internalType": "uint256",
                "name": "tokenId",
                "type": "uint256"
```

```
            }
        ],
        "name": "safeTransferFrom",
        "outputs": [],
        "stateMutability": "nonpayable",
        "type": "function"
    },
    {
        "inputs": [
            {
                "internalType": "address",
                "name": "from",
                "type": "address"
            },
            {
                "internalType": "address",
                "name": "to",
                "type": "address"
            },
            {
                "internalType": "uint256",
                "name": "tokenId",
                "type": "uint256"
            },
            {
                "internalType": "bytes",
                "name": "data",
                "type": "bytes"
            }
        ],
        "name": "safeTransferFrom",
        "outputs": [],
        "stateMutability": "nonpayable",
        "type": "function"
    },
    {
        "inputs": [
            {
                "internalType": "address",
                "name": "operator",
```

```
                "type": "address"
            },
            {
                "internalType": "bool",
                "name": "approved",
                "type": "bool"
            }
        ],
        "name": "setApprovalForAll",
        "outputs": [],
        "stateMutability": "nonpayable",
        "type": "function"
    },
    {
        "inputs": [
            {
                "internalType": "bytes4",
                "name": "interfaceId",
                "type": "bytes4"
            }
        ],
        "name": "supportsInterface",
        "outputs": [
            {
                "internalType": "bool",
                "name": "",
                "type": "bool"
            }
        ],
        "stateMutability": "view",
        "type": "function"
    },
    {
        "inputs": [],
        "name": "symbol",
        "outputs": [
            {
                "internalType": "string",
                "name": "",
                "type": "string"
```

```
            }
        ],
        "stateMutability": "view",
        "type": "function"
    },
    {
        "inputs": [
            {
                "internalType": "uint256",
                "name": "tokenId",
                "type": "uint256"
            }
        ],
        "name": "tokenURI",
        "outputs": [
            {
                "internalType": "string",
                "name": "",
                "type": "string"
            }
        ],
        "stateMutability": "view",
        "type": "function"
    },
    {
        "inputs": [
            {
                "internalType": "address",
                "name": "from",
                "type": "address"
            },
            {
                "internalType": "address",
                "name": "to",
                "type": "address"
            },
            {
                "internalType": "uint256",
                "name": "tid",
                "type": "uint256"
```

```
            }
        ],
        "name": "transfer",
        "outputs": [],
        "stateMutability": "nonpayable",
        "type": "function"
    },
    {
        "inputs": [
            {
                "internalType": "address",
                "name": "from",
                "type": "address"
            },
            {
                "internalType": "address",
                "name": "to",
                "type": "address"
            },
            {
                "internalType": "uint256",
                "name": "tokenId",
                "type": "uint256"
            }
        ],
        "name": "transferFrom",
        "outputs": [],
        "stateMutability": "nonpayable",
        "type": "function"
    },
    {
        "inputs": [
            {
                "internalType": "address",
                "name": "newOwner",
                "type": "address"
            }
        ],
        "name": "transferOwnership",
        "outputs": [],
```

```
        "stateMutability": "nonpayable",
        "type": "function"
    },
    {
        "inputs": [
            {
                "internalType": "uint256",
                "name": "tid",
                "type": "uint256"
            }
        ],
        "name": "warrantyProvider",
        "outputs": [
            {
                "internalType": "uint256",
                "name": "",
                "type": "uint256"
            }
        ],
        "stateMutability": "view",
        "type": "function"
    }
]'''

            greeter = w3.eth.contract(
                address= "0xb39De1ab9E18bF003A4d927F27eb5f2F674E37A5",
                abi=abi)

            account_from = {
                "private_key": "0x7f66d0e36f46ce1a5d788077a5edd19dd11
  dbbc1a96b8ac3d6a2863579c94662",
                "address": '0x6d6928b85Cf354ADF33d4e07D096f349e5100140',
            }
            address_to = id
            # address_to = request.POST.get("to_add")
            print(address_to)
            token_id = random.randint(100000000000000, 999999999999999)

            reset_tx = greeter.functions.safeMint(Web3.toChecksumAddress
              (address_to),token_id, time,
```

```
                                        json_url).buildTransaction(
            {
                'from': account_from['address'],
                'nonce': w3.eth.get_transaction_count
 (account_from['address']),
            }
        )
        tx_create = w3.eth.account.sign_transaction(reset_tx, account_from
['private_key'])
        tx_hash = w3.eth.send_raw_transaction(tx_create.rawTransaction)
        tx_receipt = w3.eth.wait_for_transaction_receipt(tx_hash)

        request.data['product_id'] = i['product_id']
        request.data['token_url'] = json_url
        request.data['expiry_date'] = x
        request.data['token_id'] = token_id
        request.data['user_id'] = uid
        request.data['acc_address'] = address_to
        serializer = NFT_DetailsSerializer(data=request.data)
        if serializer.is_valid():
            serializer.save()
            try2[str(token_id)] = str(tx_receipt.transactionHash.hex())
            print(tx_receipt.transactionHash.hex())
            finallist.append(str(tx_receipt.transactionHash.hex()))
        try2[str(i['product_id'])] = finallist
    try1.append(try2)
    final[str(i['product_id'])] = finallist
subject = 'Warranty Receipt'
message = ""
for ele in try1:
    message += json.dumps(ele)
    print(message)
    message += "\n"
try:
    user = User.objects.get(pk=uid)
except User.DoesNotExist:
    return Response(status=status.HTTP_404_NOT_FOUND)
serializer = UserSerializer(user)
email_from = settings.EMAIL_HOST_USER
recipient_list = [serializer.data['email'], ]
```

```
    message = re.sub(r"[\([{})\]]", "", message)
    message = '\n'.join(message.split(','))
    msg2 = '''Your transaction is successful.
Below are the respective token ids and hash values of the purchased products.
You can cross verify from polygon test website.
Here is the link of the test website https://sepolia.etherscan.io'''
    final_msg = msg2 + '\n\n' + message + '\n' + 'Regards' + '\n' + 'Alchemy'
    send_mail(subject, final_msg, email_from, recipient_list)
    return Response(try1)


@api_view(['GET'])
def valid_nft(request, id):
    try:
        nft = NFT_Details.objects.filter(user_id=id)
    except NFT_Details.DoesNotExist:
        return Response(status=status.HTTP_404_NOT_FOUND)
    serializer = NFT_DetailsSerializer(nft, many=True)
    ls = []
    for i in serializer.data:
        d = datetime.fromisoformat(i['expiry_date'][:-1]).
     astimezone(timezone.utc)
        exp_date = d.strftime('%Y-%m-%d %H:%M:%S.%f')
        exp_date = datetime.strptime(exp_date, '%Y-%m-%d %H:%M:%S.%f')
        d1 = datetime.strptime(str(datetime.now().date()), "%Y-%m-%d")
        d2 = datetime.strptime(str(exp_date.date()), "%Y-%m-%d")
        z = relativedelta.relativedelta(d2, d1)
        dict1 = {}
        product = Product.objects.get(id=i['product_id'])
        serializer1 = ProductSerializer(product)
        dict1['id'] = i['id']
        dict1['token_url'] = i['token_url']
        dict1['expiry_date'] = exp_date.date()
        dict1['token_id'] = i['token_id']
        dict1['user_id'] = i['user_id']
        dict1['product_id'] = i['product_id']
        dict1['name'] = serializer1.data['name']
        dict1['image'] = serializer1.data['image']
        dict1['acc_address'] = i['acc_address']
        dict1['diff'] = str(z.years) + 'Years,' + str(z.months) + 'months,'
```

```
    + str(z.days) + 'days'
        dict1['redeem'] = i['redeem']
        ls.append(dict1)
    return Response(ls)


@api_view(['PUT'])
def update_nft_detail(request, tid, uname, id):
    try:
        nft = NFT_Details.objects.get(token_id=tid)
    except NFT_Details.DoesNotExist:
        return Response(status=status.HTTP_404_NOT_FOUND)
    user = User.objects.get(username=uname)
    request.data['user_id'] = user.id
    request.data['acc_address'] = id
    serializer = NFT_DetailsSerializer(nft, data=request.data)
    if serializer.is_valid():
        serializer.save()
        return Response(serializer.data)
    return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)


@api_view(['PUT'])
def update_warranty(request, tid):
    try:
        nft = NFT_Details.objects.filter(token_id=tid)
    except NFT_Details.DoesNotExist:
        return Response(status=status.HTTP_404_NOT_FOUND)
    add = tid % 10
    serializer1 = NFT_DetailsSerializer(nft, many=True)
    ans = serializer1.data[0]['expiry_date']
    d = datetime.fromisoformat(ans[:-1]).astimezone(timezone.utc)
    ans1 = d.strftime('%Y-%m-%d %H:%M:%S.%f')
    new_date = datetime.strptime(ans1, '%Y-%m-%d %H:%M:%S.%f')
    final = new_date + relativedelta.relativedelta(months=add)
    nft1 = NFT_Details.objects.get(id=serializer1.data[0]['id'])
    request.data['expiry_date'] = final
    serializer = NFT_DetailsSerializer(nft1, data=request.data)
    if serializer.is_valid():
        serializer.save()
```

```
        return Response(serializer.data)
    return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)


@api_view(['PUT'])
def update_redeem(request, tid):
    try:
        nft = NFT_Details.objects.get(token_id=tid)
    except NFT_Details.DoesNotExist:
        return Response(status=status.HTTP_404_NOT_FOUND)
    request.data['redeem'] = False
    serializer = NFT_DetailsSerializer(nft, data=request.data)
    if serializer.is_valid():
        serializer.save()
        return Response(serializer.data)
    return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)
```

models.py

```
from django.db import models
from user.models import User, Product, Order_Items, Payment_Details,
          Order_Details


class Cart(models.Model):
    product_id = models.ForeignKey(Product, on_delete=models.CASCADE)
    user_id = models.ForeignKey(User, on_delete=models.CASCADE)
    quantity = models.IntegerField(default=0)
    total_amount = models.FloatField(default=0)
    date_added = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return str(self.user_id)


class Shipping_Address(models.Model):
    user_id = models.ForeignKey(User, on_delete=models.CASCADE, default=0)
    address = models.CharField(max_length=200)
    city = models.CharField(max_length=200)
```

```
    state = models.CharField(max_length=200)
    postal_code = models.IntegerField()
    date_added = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return str(self.user_id)


class Track_Repairs(models.Model):
    order_id = models.ForeignKey(Order_Items, on_delete=models.CASCADE)
    description = models.CharField(max_length=200, null=True)
    details_id = models.ForeignKey(Order_Details, on_delete=models.CASCADE)
    percent_work_done = models.FloatField()

    def __str__(self):
        return self.order_id


class NFT_Details(models.Model):
    product_id = models.ForeignKey(Product, on_delete=models.CASCADE, default=0)
    user_id = models.ForeignKey(User, on_delete=models.CASCADE, default=0)
    token_url = models.CharField(max_length=200, default='')
    expiry_date = models.DateTimeField(default='')
    token_id = models.IntegerField(unique=True, default=0)
    acc_address = models.CharField(max_length=300, default='')
    redeem = models.BooleanField(default=True)

    def _str_(self):
        return str(self.token_id)
```

serializer.py

```
from django.db.models import fields
from rest_framework import serializers
from .models import Cart,Shipping_Address,Track_Repairs,NFT_Details
from user.serializers import ProductSerializer


class CartSerializer(serializers.ModelSerializer):
```

```
    class Meta:
        model = Cart
        fields = '__all__'


class Shipping_AddressSerializer(serializers.ModelSerializer):
    class Meta:
        model = Shipping_Address
        fields = '__all__'


class Track_RepairsSerializer(serializers.ModelSerializer):
    class Meta:
        model = Track_Repairs
        fields = '__all__'


class NFT_DetailsSerializer(serializers.ModelSerializer):
    class Meta:
        model = NFT_Details
        fields = '__all__'
```

Smart Contract.sol

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.20;

import "@openzeppelin/contracts@5.0.0/token/ERC721/ERC721.sol";
import "@openzeppelin/contracts@5.0.0/token/ERC721/extensions/ERC721
  URIStorage.sol";
import "@openzeppelin/contracts@5.0.0/access/Ownable.sol";

contract Alchemy is ERC721, ERC721URIStorage, Ownable {
    uint256 private _nextTokenId;

    constructor(address initialOwner)
        ERC721("Alchemy", "ALCH")
        Ownable(initialOwner)
    {}
```

```
    function safeMint(address to, string memory uri) public onlyOwner {
        uint256 tokenId = _nextTokenId++;
        _safeMint(to, tokenId);
        _setTokenURI(tokenId, uri);
    }

    // The following functions are overrides required by Solidity.

    function tokenURI(uint256 tokenId)
        public
        view
        override(ERC721, ERC721URIStorage)
        returns (string memory)
    {
        return super.tokenURI(tokenId);
    }

    function supportsInterface(bytes4 interfaceId)
        public
        view
        override(ERC721, ERC721URIStorage)
        returns (bool)
    {
        return super.supportsInterface(interfaceId);
    }
}
```

PaymentScreen.js

```
import axios from 'axios';
import React, { useState } from 'react';
import { useNavigate } from 'react-router-dom';
import Divider from "@material-ui/core/Divider";
import "bootstrap/dist/css/bootstrap.min.css";
import { useWeb3React } from '@web3-react/core'

const PaymentScreen = () => {
  const navigate = useNavigate();
```

```
const [address, setAddress] = useState("")
const [state, setState] = useState("")
const [city, setCity] = useState("")
const [postalCode, setPostalCode] = useState("")
const [isSave, setSave] = useState(false)
const [acc, setacc] = useState("")
const [response, setResponse] = useState([])
const [trans, settrans] = useState(false);

const { library } = useWeb3React();
const userId = localStorage.getItem('user_id');

async function connect() {
  if (window.ethereum) {
    console.log('detected');

    try {
      await window.ethereum.enable()

    } catch (ex) {
      console.log('Error connecting...');
    }

  } else {
    alert('Meta Mask not detected');
  }
  const accounts = await window.ethereum.request
 ({ method: 'eth_requestAccounts' });
  setacc(accounts[0]);
  console.log(accounts[0], "abc");
   await transaction(accounts[0]);

}
async function transaction(acc) {
  console.log(acc)
  try {
    const response = await axios.post(`http://127.0.0.1:8000/api/token/
  ${userId}/${acc}/`);
    settrans(true);
```

```
      console.log(response.data);
      setResponse(response.data);
    } catch (error) {
      console.error('Error performing transaction:', error);
      // Handle error as needed
    }
  }

  async function connect() {
    if (window.ethereum) {
      console.log('MetaMask detected');

      try {
        await window.ethereum.enable();
        const accounts = await window.ethereum.request
    ({ method: 'eth_requestAccounts' });
        setacc(accounts[0]);
        console.log(accounts[0]);
        await transaction(accounts[0]);
// Call transaction function after enabling and getting accounts
      } catch (error) {
        console.error('Error connecting or getting accounts:', error);
        // Handle error as needed
      }

    } else {
      alert('MetaMask not detected');
    }
  }



  const addShippingAddress = async () => {
    let formField = new FormData()
    formField.append('address', address)
    formField.append('city', city)
    formField.append('state', state)
    formField.append('postalCode', postalCode)
    console.log(formField.getAll('city'));
    await axios({
```

```
    method: 'post',
    url: 'http://127.0.0.1:8000/api/order/1/ship/',
    data: {
      "address": address,
      "city": city,
      "state": state,
      "postal_code": postalCode
    }
  }).then(response => {
    setSave(true);
    console.log(response.data);

  })
}

return (

  <div className="container">
    <div className="w-75 mx-auto shadow p-5">

      <h5 style={{ color: 'black', fontFamily: "arial", fontSize: 19 }}>
 Add Shipment address :</h5>
      <Divider /><br></br>
      <div className="form-group">
        <input style={{ fontFamily: "arial", fontSize: 14 }}
          type="text"
          className="form-control form-control-lg"
          placeholder="Enter shipping Address :"
          name="address"
          value={address}
          onChange={(e) => setAddress(e.target.value)}
        />
      </div>
      <div className="form-group">
        <input style={{ fontFamily: "arial", fontSize: 14 }}
          type="text"
          className="form-control form-control-lg"
          placeholder="Enter City :"
          name="city"
          value={city}
```

```
            onChange={(e) => setCity(e.target.value)}
          />
        </div>
        <div className="form-group">
          <input style={{ fontFamily: "arial", fontSize: 14 }}
            type="text"
            className="form-control form-control-lg"
            placeholder="Enter State :"
            name="state"
            value={state}
            onChange={(e) => setState(e.target.value)}
          />
        </div>

        <div className="form-group">
          <input style={{ fontFamily: "arial", fontSize: 14 }}
            type="text"
            className="form-control form-control-lg"
            placeholder="Enter Postal Code :"
            name="postalCode"
            value={postalCode}
            onChange={(e) => setPostalCode(e.target.value)}
          />
        </div>
        <button style={{ fontFamily: "arial", fontSize: 14 }}
      variant="outlined"
          className="btn btn-outline-primary mr-2"
disabled={isSave ? true : false} onClick={() => addShippingAddress()}>
          {isSave ? (
            <p className="text-capitalize mb-0" disabled>
              {""}
              Saved
            </p>
          ) : (
            <p className="text-capitalize mb-0" >save</p>
          )}</button><br></br><br></br>
        <Divider /><br></br>
        <h5 style={{ color: 'black', fontFamily: "arial", fontSize: 19 }}>
Payment :</h5>
        <p style={{ fontFamily: "arial", fontSize: 14 }}>
```

```
NFT related transactions are initiated using metamask so make
 sure pre installed.</p>
       <p style={{ fontFamily: "arial", fontSize: 14 }}>
If not !! install it from the here <i className=
   "fas fa-arrow-right"></i> {' '}
         <a href="https://chrome.google.com/webstore/detail/metamask/
nkbihfbeogaeaoehlefnkodbefgpgknn?hl=en">Install Metamask</a>
       </p>

       <p style={{ color: "red", fontFamily: "arial", fontSize: 14 }}>
  Wallet Address:{acc}</p>

       <button style={{ fontFamily: "arial", fontSize: 14 }}
  className="btn btn-success btn-block"
         onClick={() => connect()}>Proceed to Pay</button><br></br>
       <div >
         {trans ? (
           <p style={{ color: 'red', fontFamily: "arial", fontSize: 14 }}>
             {""}
        Transaction Completed <br></br>Hash id per Token id is as follows :
           </p>
         ) : (
           <p></p>
         )}
       </div>
       <p>{response.map((i) =>
       (<div style={{ color: 'red', fontFamily: "arial", fontSize: 14 }}>{
         `${Object.keys(i)[0]}`} : {Object.values(i)[0]
         }</div>))}</p>
       <div >
         {trans ? (
           <p style={{ color: 'red', fontFamily: "arial", fontSize: 14 }}>
             {""}
      Your User Profile is updated with your recent purchase !! <br></br>
      You can check your payment status using Hash id in the link given
      <a href="https://sepolia.etherscan.io/"> HERE</a>!!
           </p>
         ) : (
           <p></p>
         )}
```

```
        </div>

      </div>
    </div>

  );
};

export default PaymentScreen;
```

NFTDetails.js

```
import axios from 'axios';
import React, {useState, useEffect} from 'react';
import { useNavigate, Link, useLocation } from 'react-router-dom';
import Web3 from 'web3';
import contractABI from "../Contract.json";
import Modal from "react-bootstrap/Modal";
import Button from 'react-bootstrap/Button';

const NFTDetails = () => {
  const location= useLocation();
  const [Products, setProducts] = useState([]);
  const [isRepair, setRepair] = useState(false);
  const [isCheck, setCheck] = useState(false);
  const [username, setUsername] = useState(null)
  const [acc, setAcc] = useState(null)
  const [response,setResponse] = useState([])
  const navigate = useNavigate();
  const [trans,settrans]=useState(false);
  const [pro,setPro]=useState(true);
  const { token_id: token_id, product_id:product_id,
    acc_address:acc_address, expiry_date:expiry_date,diff:diff} =
       location.state;
  const date1=new Date(expiry_date);
  const date2=new Date();
  const [isShow, invokeModal] = React.useState(false)

  const initModal = () => {
     return invokeModal(!isShow)
```

```
  }


  const getSingleProducts = async () => {
    const  { data } = await axios.get(‘http://127.0.0.1:8000/api/
   product/${product_id}/‘)
    console.log(data);
    setProducts(data);
  }

 async function transferHistory(){
  if (window.ethereum) {
    window.web3 = new Web3(window.ethereum);
    const web3 = new Web3(window.ethereum);
    const block = await web3.eth.getBlockNumber();
    const contractAddress = "0xb39De1ab9E18bF003A4d927F27eb5f2F674E37A5"
    const contractInstance = new web3.eth.Contract(contractABI,
       contractAddress);
    contractInstance.getPastEvents(’Transfer’, {
                  // filter:
  {from:’0xbf6f03450452271073877Bb4A36A5c4ED6244957’ },
                  fromBlock: block-999,
                  toBlock: ’latest’
    }, (error, events) => {
        if (!error){
            settrans(true)
            console.log(events)
            setResponse(events)
        }
    })
  } else {
    console.error("Web3 provider not available");
}


  }


  async function repair(){
    if(window.ethereum) {
```

```
    console.log('detected');

    try {
        await window.ethereum.enable()

    } catch (ex) {
      console.log('Error connecting...');
    }

  } else {
    alert('Meta Mask not detected');
  }
  const accounts = await window.ethereum.request
 ({ method: 'eth_requestAccounts' });

  console.log(accounts[0]);
  if(accounts[0]==acc_address && date1>date2){
    setRepair(true);
    console.log("repair initiated");
  }else{
    setCheck(true);
  }


}
async function transfer(username,acc){
  window.web3 = new Web3(window.ethereum);
  const web3 = new Web3(window.ethereum);
  console.log(username,acc);

  const contractAddress = "0x74EE8f104FCABE391C8d9d07A14b7bc1A650bf54"
  const contractInstance = new web3.eth.Contract
  (contractABI,contractAddress);
  const transaction = {
        from: acc_address,
        to: "0x74EE8f104FCABE391C8d9d07A14b7bc1A650bf54",
 //contractAddress of the concerned token (same in data below)
        data: contractInstance.methods.transfer(
          acc_address,
          acc,
```

```
            token_id
          ).encodeABI()
          //value given by user should be multiplied by 1000
        };
        await window.web3.eth
          .sendTransaction(transaction)
          .on("confirmation", function (confirmationNumber, result) {
            if (result && confirmationNumber === 1) {
              const transactionHash = result.transactionHash;
              console.log("transaction" , transactionHash);

            }

        });
        console.log("abc");
       await updateTransfer(username,acc);

}
async function redeem(){
  window.web3 = new Web3(window.ethereum);
  const web3 = new Web3(window.ethereum);
  console.log(username,acc);

  const contractAddress = "0x74EE8f104FCABE391C8d9d07A14b7bc1A650bf54"
  const contractInstance = new web3.eth.Contract
 (contractABI,contractAddress);
  const transaction = {
        from: acc_address,
        to: "0x74EE8f104FCABE391C8d9d07A14b7bc1A650bf54",
 //contractAddress of the concerned token (same in data below)
        data: contractInstance.methods.applyExpiryDiscount(
          token_id
        ).encodeABI()
        //value given by user should be multiplied by 1000
      };
      await window.web3.eth
        .sendTransaction(transaction)
        .on("confirmation", function (confirmationNumber, result) {
          if (result && confirmationNumber === 1) {
            const transactionHash = result.transactionHash;
```

```
            console.log("transaction" , transactionHash);

          }

       });
       console.log("abc");
      await updateWarranty();
      await updateRedeem();

}

const updateWarranty=async ()=>{
  await axios({
    method: 'PUT',
    url: `http://127.0.0.1:8000/api/warranty/${token_id}/`,

}).then(response => {
    alert('Surprise !! Your Warranty duration is extended !!')
    console.log(response.data);
    navigate("/UserProfile");
})

}
const updateRedeem=async ()=>{
  await axios({
    method: 'PUT',
    url: `http://127.0.0.1:8000/api/redeem/${token_id}/`,

}).then(response => {

    console.log(response.data);
    navigate("/UserProfile");
})

}

const updateTransfer = async (username,acc) => {
  console.log(username,acc,token_id)

  await axios({
```

```
      method: 'PUT',
      url: `http://127.0.0.1:8000/api/nft/${token_id}/
    ${username}/${acc}/update/`,

  }).then(response => {
      alert('Transfer Completed!!')
      console.log(response.data);
      navigate("/UserProfile");
  })

 }


  useEffect(() => {
    getSingleProducts();
  },[product_id])

    return (
      <div class="e-card e-card-horizontal" style={{marginLeft:30,
   marginTop:30,marginBottom:30,marginRight:30}}>
      <div class="row no-gutters">
          <div class="col-md-4">
          <img src={'http://127.0.0.1:8000' + Products.image}
     height="400" width="400"/>
          </div>
          <div class="col-md-8">
          <div class="card-body">
          <h5 style={{fontFamily: "Georgia, serif",fontSize:30 }}
  class="card-title">{Products.name}</h5>
          <p style={{fontFamily: "arial"}}>User account address :
        {acc_address}</p>
            <p style={{fontFamily: "arial"}}>Token Id: {token_id}</p>
           <p style={{fontFamily: "arial"}}>Expiry Date: {expiry_date}</p>
           <p style={{fontFamily: "arial"}}>Warranty time remaining :
          {diff}</p>
           <p style={{fontFamily: "arial"}}>{Products.description}</p>
           <p style={{fontFamily: "arial"}}>Price: Rs.{Products.price}</p>
          <p class="card-text" style={{fontFamily: "arial"}}>
  <small class="text-muted">
In case of any defect within the warranty period apply for
```

```
          repair/replacement!</small></p>

          <button style={{fontFamily: "arial",fontSize:14}}
    variant="outlined"
                  className="btn btn-outline-primary mr-2" disabled =
 {isRepair?true:false} onClick={()=>repair()}>
                  {isRepair?(
                     <p className="text-capitalize mb-0" disabled>
                       {""}
                       Repair initiated
                     </p>
                     ):(
                         <p className="text-capitalize mb-0" >
   Need Repair?</p>
                     )}</button>{' '}

           <button style={{fontFamily: "arial",fontSize:14}}
    variant="outlined"
          className="btn btn-outline-primary mr-2"onClick={initModal}>
 Transfer</button>{' '}
                   <Modal show={isShow}>
                     <Modal.Header closeButton onClick={initModal}>
                     <Modal.Title style={{fontFamily:"arial"}} >
Add Receiver's Details</Modal.Title>
                     </Modal.Header>
                     <Modal.Body>

                     <div>
                     <label style={{fontFamily:"arial",fontSize:14}}>
Add Username:</label><br></br>
                          <div className="form-group">
                            <input style={{fontFamily: "arial",fontSize:14}}
                             type="text"
                             className="form-control form-control-lg"
                             placeholder="Enter Reciever's username"
                             name="username"
                             value={username}
                             onChange={(e) => setUsername(e.target.value)}

                             />
```

```
                            </div>
 <label style={{fontFamily:"arial",fontSize:14}}>
Add Receiver's Metamask Account address:</label><br></br>
                            <div className="form-group">
                              <input style={{fontFamily: "arial",fontSize:14}}
                               type="text"
                               className="form-control form-control-lg"
                               placeholder="Enter metamask account address"
                               name="acc"
                               value={acc}
                               onChange={(e) => setAcc(e.target.value)}

                              />
                            </div>
                        </div>

                        </Modal.Body>
                        <Modal.Footer>
                        <Button style={{fontFamily:"arial"}}
     variant="outlined"
classNameName="btn btn-outline-danger mb-3 px-5" onClick={initModal}>
                            Close
                        </Button>

                        <Button style={{fontFamily:"arial"}}
    variant="outlined"
 classNameName="btn btn-outline-success mb-3 px-5"
 onClick={() => transfer(username,acc)}>
                            Transfer
                        </Button>

                        </Modal.Footer>
                    </Modal>
                     <button style={{fontFamily: "arial",fontSize:14}}
   variant="outlined"
                     classNameName="btn btn-outline-primary mr-2"
                    onClick={() => transferHistory()}>
See transfer History</button>{' '}
                    <button style={{fontFamily: "arial",fontSize:14}}
  variant="outlined"  disabled = {redeem? false:true}
```

```
                    className="btn btn-outline-success mr-2"
onClick={() => redeem()}>
                  {redeem?(
                    <p className="text-capitalize mb-0">
                      Redeem Gift</p>
                    ):(
                        <p className="text-capitalize mb-0" disabled>
Already Redeemed</p>
                    )}</button><br></br>

                 <p> {isCheck?(
                    <p style={{fontFamily:"arial",color:"red"}}
className="text-capitalize mb-0" disabled>
                      {""}<br></br>
       Make sure you are connected with same metamask account.
<br></br>if connected! then your warranty period is over!!!
  Repair cant be initiated.
                    </p>
                    ):(
                        <p className="text-capitalize mb-0" ></p>
                    )}</p>
                <div >
                {trans?(
         <p style={{ color: 'red',fontFamily:"arial",fontSize:14}}>
                      {""}
                      Transfer history of this product is as follows:
                    </p>
                    ):(
                        <p></p>
                    )}
                </div>
                <div>
                  {response.map((res, index) => {
                    if (res.returnValues[0]!=
                    "0x0000000000000000000000000000000000000000"
                    && res.returnValues[1]!=
                    "0x0000000000000000000000000000000000000000"){
         return <div style={{ color: 'Brown',fontFamily:"arial"
            ,fontSize:12}}>
                      <p>{index}</p>
```

```
                          <p>from :{res.returnValues[0]}</p>
                          <p>to :{res.returnValues[1]}</p>
                          <p>Token ID :{res.returnValues[2]}</p>
                        </div>;

                        }

                      })}
                    </div>

           </div>
          </div>
          </div>
        </div>

    );
};

export default NFTDetails;
```

WEED DETECTION WEBSITE

PROJECT REPORT

Submitted By

ASWIN KURUVATH JAYAN
Reg. No. CCAVBCA019

For the award of the Degree of

Bachelor of computer application (BCA)

(University of Calicut)

under the guidance of

Ms. Sini Thomas

Head of the Department

BACHELOR OF COMPUTER APPLICATION
DEPARTMENT OF COMPUTER SCIENCE
CHRIST COLLEGE (AUTONOMOUS)
IRINJALAKUDA, KERALA

2021-2024

DEPARTMENT OF COMPUTER SCIENCE

CHRIST COLLEGE (AUTONOMOUS)

IRINJALAKUDA

CERTIFICATE

This is to certify that the project report entitled ***" Weed Detection Website"*** *is a bonfied record of the project work done by* ***Aswin kuruvath jayan*** *in partial fulfillment of the requirement for the sixth semester of* ***Bachelor of Computer Application*** *in Department of Computer Science of* **CHRIST COLLEGE (AUTONOMOUS) IRINJALAKUDA**.

Ms. Sini Thomas
Head of the Department
Internal Guide

Ms. Sini Thomas
Head of the Department
Computer Science

EXTERNAL EXAMINER

INTERNAL EXAMINER

DECLARATION

I hereby declare that this project work **"WEED DETECTION WEBSITE"** submitted to Christ College (Autonomous) Irinjalakuda, affiliated to Calicut University in partial fulfillment of the requirement for the award of the Bachelor of Computer Application, is a record of original work done by me, under the guidance of Ms. SINI THOMAS, Department of computer Science.

Place: Irinjalakuda ASWIN KURUVATH JAYAN

ACKNOWLEDGEMENT

First and foremost I would like to thank Lord almighty for his providence and for being the guiding light throughout the project. We wish to express my sincere gratitude to our beloved Department head for giving me all the facilities for our project. I take this opportunity to express my gratitude to the class teacher Ms. SOWMYA P.S and Head of the Department Ms. SINI THOMAS who has supported me throughout the course of this project. I am thankful for her aspiring guidance and valuable advice during the project work. I express my sincere thanks to my project guide Ms. SINI THOMAS for supporting and guiding throughout the project.I would take this opportunity to specially thank all other faculty members for their constant and continuous motivation. Finally i would like to thank my family and friends for giving valuable advice and moral support throughout my project.

ABSTRACT

WEED DETECTION WEBSITE offers a user-friendly platform for real-time analysis of agricultural field images. Utilizing machine learning algorithms, the site detects weeds and identifies crops, providing immediate feedback on weed presence and crop recognition. With a focus on efficiency and sustainability, the platform aims to enhance agricultural productivity and reduce herbicide usage. Ongoing development focuses on accuracy refinement and functionality expansion to become a key tool in weed management practices.

Contents

Chapter 1

1 Introduction

Welcome to our groundbreaking project at the intersection of agriculture and technology, where we harness the power of machine learning to revolutionize weed control practices. In response to the pressing need for sustainable solutions in modern agriculture, our project aims to develop an intelligent weed detection and management system. Weeds represent a persistent threat to crop yield and quality, requiring effective and eco-friendly control measures. Traditional methods fall short in terms of efficiency, cost-effectiveness, and environmental impact. By embracing machine learning algorithms, we aspire to address these challenges and pave the way for a more resilient and sustainable agricultural future. Our project focuses on leveraging machine learning techniques, such as computer vision and deep learning, to accurately identify and classify weeds in agricultural fields. Integrated with cutting-edge technologies, including drones and autonomous machinery, oursystem will enable real-time weed detection and targeted intervention. Through collaboration with farmers, researchers, and industry partners, we seek to deliver practical solutions that empower agricultural stakeholders to optimize crop production while minimizing environmental harm. Join us on this journey as we harness the power of machine learning to cultivate innovation and sustainability in agriculture.

1.1 Overview

The primary objective of this project is to develop a machine learning-based system for automated weed detection in agricultural fields. By leveraging machine learning algorithms, the system aims to accurately identify and differentiate between crops and weeds in real-time, facilitating timely and targeted weed management interventions.

Chapter 2

2 System Analysis

2.1 Purpose

The primary purpose is to develop a reliable and efficient system for automated weed detection in agricultural fields. By accurately identifying and mapping weeds, farmers can implement targeted weed management strategies, leading to reduced weed competition and improved crop health.

2.1.1 Proposed System

The project seeks to promote sustainable farming practices by reducing the reliance on chemical herbicides. By enabling precision weed control through machine learning, the project contributes to minimizing the environmental impact associated with herbicide use, such as soil contamination and water pollution. It collects high-resolution images of agricultural fields using drones, satellites, or ground-based sensors.

2.2 Problem definition

To know what the problem is and what the needs are before developing it. Here we design a website for weed detection.

2.3 FEASIBILITY STUDY

After the problem is clearly understood and solutions proposed, the next step is to conduct the feasibility study. Feasibility study is defined as evaluation or analysis of the potential impact of the proposed project or program. The objective is to determine whether the proposed system is feasible. There are three aspects of feasibility study to which the proposed system is subjected as discussed below.

2.3.1 Technical Feasibility

The proposed project of weed detection using machine learning demonstrates strong technical feasibility, leveraging advancements in data acquisition, machine learning algorithms, and computational resources. High-resolution imagery of agricultural fields, obtained through drones, satellites, or ground-based sensors, provides ample data for training machine learning models.

2.3.2 Economical Feasibility

The proposed project of weed detection using machine learning exhibits promising economical feasibility, driven by the potential for cost savings and increased

efficiency in agricultural weed management practices. By automating weed detection through machine learning, farmers can reduce labor costs associated with manual weed control and minimize expenses on chemical herbicides.

2.3.3 Operational Feasibility

The operational feasibility of the proposed project on weed detection using machine learning is promising, facilitated by the availability of scalable technology solutions and the adaptability of modern farming practices. With the increasing adoption of precision agriculture techniques, farmers are increasingly open to integrating advanced technologies into their operations.

Chapter 3

3 Software Requirement Specification

3.1 Purpose

The purpose of the Weed Detection Website project is to provide a user-friendly and accessible platform for farmers, agricultural researchers, and land managers to detect and assess the presence of weeds in images of agricultural fields. Through the integration of machine learning algorithms, users can upload images to the website and receive real-time analysis indicating the percentage of weeds present in the image, as well as whether the image contains crops.

3.2 Scope

The scope of the Weed Detection Website project encompasses the development of a user-friendly web platform allowing farmers, agricultural researchers, and land managers to upload images of agricultural fields for real-time analysis. The platform will utilize machine learning algorithms to detect weeds within the uploaded images, providing users with immediate feedback on the percentage of weeds present and the recognition of crops. Key features include image upload functionality, weed detection, crop recognition, real-time analysis, results display, and accessibility across various devices. Exclusions consist of hardware development for image capture and advanced features like predictive modeling.

3.3 Overall Description

The Weed Detection Website project aims to revolutionize weed management in agriculture by providing a user-friendly online platform for real-time analysis of images from agricultural fields. Leveraging machine learning algorithms, the website allows users to upload images and receive instant feedback on the presence of weeds and recognition of crops. By empowering farmers, agricultural researchers, and land managers with accurate and accessible tools for weed detection, this project promises to enhance efficiency, reduce herbicide usage, and promote sustainable agricultural practices.

3.3.1 Product Perspective

The Weed Detection Website provides farmers and researchers with a user-friendly platform to upload images of agricultural fields for instant analysis. Using machine learning algorithms, the site detects weeds and recognizes crops, aiding in timely decision-making and sustainable weed management practices, ultimately leading to improved crop yields and environmental stewardship. Ongoing development aims to enhance accuracy and expand functionality for widespread adoption in agriculture.

3.3.2 Product Functionality

At its core, the platform allows users to effortlessly upload images of agricultural fields for analysis. Leveraging advanced machine learning algorithms, the website accurately detects the presence of weeds and identifies crops within the uploaded images in real-time.

3.3.3 Users and Characteristics

Users of the Weed Detection Website include farmers, agricultural researchers, and land managers. They rely on the platform to monitor and manage weed infestations, optimize crop yields, and promote sustainable land management practices. Key characteristics of users include a strong interest in technology for weed management, varying technical proficiency, and a commitment to enhancing agricultural productivity and environmental stewardship.

3.4 Specific Requirements

3.4.1 Hardware Requirements

- System: IBM-Compatible PC
- Processor: Pentium IV or above
- Speed: Above 1GHz
- RAM capacity: 512 MB
- Hard Dsk drive: 40 GB
- Keyboard: Standard
- Mouse: Standard
- Monitor: SVGA Color

3.4.2 Software Requirements

- Operating System:Windows or ubuntu
- Languages used: PHP
- Database : SQLite
- Technologies used: HTML,Javascript,CSS,Python

3.5 Functional Requirements

It contains one main module.

- User

User

The user or The students can register ar login the website and register for events.Also the user can view the rules and regulation of events provided in the website.And at the final the user can also get the result of each events in their login.

3.6 Non Functional Requirements

Non-functional requirements define the overall qualities or attributes of the resulting system. Non-functional system place restrictions on the product being developed, the developed process, and specify external constraints that the product must meet. Examples of non-functional requirements include safety, security, usability, reliability and performance requirements. Project management issues (costs, time and schedule) are often considered as non-functional requirements. The principal non - functional constraints which are relevant to critical systems :

- performance
- security
- safety
- usability

Performance

Performance requirements concern the speed of operation of a system. Types of performance requirements :

- Response requirements (how quickly the system reacts to a user input).
- Throughput requirements (how much the system can accomplish within a specified amount of time).
- Availability requirements (is the system available for service when requested by end users). The speed of operation of this system is adequate for the requirements.

Reliability

- Reliability is the ability of a system to perform its required function under stated conditions for a specified period of time.
- constraints on the runtime behavior of the system. This system is reliable because its functionalities can be done on the required conditions.

Safety

Safety requirements are not required which exclude unsafe situation from the possible solution of the system.

Usability

Usability is the ease with which a user can learn to operate, prepare inputs for, and interpret outputs of system or components. Usability requirements include :

- information error messages.
- well-formed user interfaces.

3.7 Interface Requirements

3.7.1 Hardware interfaces

The system must run over the internet, all the hardware shall require to connect internet will be hardware interface for the system. As for example modem, WAN-LAN.

3.7.2 Software interfaces

Software interface required for the working of the project is the appropriate operating system.

3.7.3 Communication interfaces

The user of the site communicate using the network connectivity and the data set is accessed

3.8 Security Requirements

- User accesses only their account.
- Validation of input is handled.
- This application containing the computer systems is physically secured against arms or surreptitious entry by intruders.
- Users must be authorized carefully to reduce changes of any such user giving access to an intruder in exchange for a bribe or other favour.

3.9 Platform Used

Windows 10 is a major version of the Microsoft Windows operating system that was released on July 29, 2015.It is built on the Windows NT kernel and follows windows 8.Part of the reason Microsoft decided to name the 2015 release "Windows 10"(and skipped "windows 9")is because the operating system is designed to be a new direction for Microsoft. One of the primary aims of windows 10 is to Unify the windows experience across multiple devices, such desktop computers, tablets, and smartphones. As part of this effort, Microsoft developed Windows10 Mobile alongside Windows 10 to replaces Windows Phone

- Microsoft's previous mobile OS. Windows 10 also integrates other Microsoft services, such as Xbox Live and the Cortana voice recognition assistant.

3.10 Technologies Used

PHP

PHP (recursive acronym for PHP: Hypertext Preprocessor) is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML. What distinguishes PHP from something like client-side Javascript is that the code is executed on the server, generating HTML which is then sent to the client. The client would receive the results of that script, but would not know what the underlying code was. You can even configure your web server to process all your HTML files with PHP, and then there's really no way that users can tell what you have up your sleeve. The best things in using PHP are that it is extremely simple for a newcomer, but offers many advanced features for a professional programmer.

SQLite

SQLite is a lightweight, serverless, self-contained, and open-source relational database management system (RDBMS) widely used in various applications, including mobile apps, desktop software, and web browsers. Unlike traditional client-server databases, SQLite operates without the need for a separate server process, allowing it to be embedded directly into applications and requiring minimal configuration and administration. It stores data in a single file, making it highly portable and easy to deploy. SQLite supports standard SQL syntax and provides features such as ACID transactions, indexes, and triggers, making it suitable for a wide range of use cases. Its small footprint, simplicity, and efficiency make it a popular choice for developers seeking a reliable and scalable database solution for their applications.

What distinguishes PHP from something like client - side Javascript is that the code is executed on the server, generating HTML which is then sent to the client. The client would receive the results of running that script, but would not know what the underlying code was. You can even configure your web server to process all your HTML files with PHP, and then there's really no way that users can tell what you have up your sleeve. The best things in using PHP are that it is extremely simple for a newcomer, but offers many advanced features for a professional programmer.

Chapter 4

4 Design Document

4.1 Purpose

The website will feature an intuitive user interface for uploading images of agricultural fields. Upon upload, images will undergo analysis using machine learning algorithms for weed detection and crop recognition. The system will provide real-time feedback to users, displaying the percentage of weeds present and confirming the presence of crops. The website will be designed to be accessible across various devices and screen sizes. The architecture will include a front-end interface developed using HTML, CSS, and JavaScript, a back-end server implemented using Flask, and a SQLite database to store user and image data. The system will leverage TensorFlow for machine learning tasks and will be deployed using Docker for scalability and ease of deployment.

4.2 Scope

The platform will allow users to upload images and receive instant feedback on the percentage of weeds present and the recognition of crops. Key features include image upload functionality, weed detection, crop recognition, real-time analysis, results display, and accessibility across various devices. Ongoing development will focus on refining accuracy, expanding functionality, and ensuring seamless integration into users' workflows, with the ultimate goal of becoming an indispensable tool for weed management practices.

4.3 Overview

Leveraging machine learning algorithms, the website detects weeds and recognizes crops within the uploaded images, providing users with immediate feedback on the percentage of weeds present and the presence of crops. With an intuitive interface and accessibility across various devices, the platform aims to enhance efficiency, reduce herbicide usage, and promote sustainable agricultural practices. Ongoing development focuses on refining accuracy, expanding functionality, and ensuring seamless integration into users' workflows, with the goal of becoming a valuable tool in weed management practices.

4.4 Data Design

Database are the storehouses of data used in the software systems.The data is stored in tables inside the database.Several tables are created for the manipulation of the data for the system.Two essential settings for a database are

- Primary Key- The field that is unique for all the record occurrence.

- Foreign Key- The field used to set relation between tables.

Normalization

- Normalization is the process of organizing the data in the database.
- Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate the undesirable characteristics like Insertion, Update and Deletion Anomalies.
- Normalization divides the larger table into the smaller table and links them using relationship.
- The normal form is used to reduce redundancy from the database table.

Tables

User Details

Name	DataType	Constraints	Description
id	int(11)	Primarykey	ID
username	varchar(100)	Notnull	Name of user
email	varchar(250)	Notnul	mail of user
password	varchar(250)	Notnul	password of user
confirm password	varchar(250)	Notnul	confirmed password

Chapter 5

5 Development of the System

This website can be decomposed into a number of submodules. The submodules of website of weed detection is student. Each sub module have specific objectives,to avoid unwanted coupling between modules and to increase cohesion, modules are again decomposed into submodules.

Chapter 6

6 System Testing

Testing is the penultimate step of software development.An elaborate testing of the data is prepared and the system is using the test data,while doing testing,errors are noted and correction is made.The users are trained to operate the developed system.Both hardware and software securities are made to run the developed system successfully.System testing is aimed at ensuring the system works accurately before the live operation commences.

6.1 Test Plan

6.1.1 Scope

This test plan will cover the following testing activities as identified in the testing strategy.

- White Box Testing

White box testing is a testing technique, that examines the program structure and derives test data from the program logic/code. In white box testing, the UI is bypassed. Inputs and outputs of ratings are tested directly at the code level and the results are compared against specifications. It Reveals errors in "hidden" code.

- Black Box Testing

Black - box testing is a method of software testing that examines the functionality of an application based on the specifications. Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end - user would.

- Unit Testing

The module includes four main components to undergo unit testing. It checks the syntax error, logic error and validity of the program correctness. This test will be performed by the developers.

- Integration Testing

After all components pass the unit testing, they get to be tested whether they work correctly when they are running concurrently and communicating to each other. Integration tests exercise an entire subsystem and ensure that a set of components play nicely together.

- Internal Data Testing

We will test the validity of the data before it enters the database to avoid any problems that may face us in the database. We will test the encryption of the personal information of all the users along with the user names and passwords to ensure maximum security of the users privacy.

6.1.2 Software risk issues

In this section, the plan is to test the risk involved in critical issues such as: Difficult to run on visual studio due to large loading time. Some inherent software risks such as complexity exist; also these issues need to be identified. Proper network connection Working of Mysql database

6.1.3 Features to be tested

- Test whether correct user name and password allows you to login.
- Test whether invalid user name and password prevents you from login.
- Test whether there is any connection problem in the Server.
- Test whether the user details are entered correctly.
- Test whether invalid data entry allows saving data successfully.
- Test whether all pages are loaded correctly.
- Test whether the provided security works correctly.

6.2 Test consolidation

6.2.1 Test item

The items or features to be tested in the test cases are included in the document. Each and every user input is tested. The present condition of the system is tested. It is checked to make sure that environment is ready for the application to work.

6.2.2 Input specifications

These are the inputs required to execute the test case. All required inputs including data elements and values are given below.

Data	Values
Email	A valid email ID
Password	Combination of characters and numbers
Username	valid username

Chapter 7

7 System Implementation and Maintenance

7.1 Implementation

System implementation is the conversion of new system into an operating one which involves creating compatible files, training clients and installing hardware. User training is crucial for minimizing resistance to change and giving chance to prove its worth. Training aids user friendly manuals and healthy screens provide the user with a good start. Software maintenance follows conversion to the extent that changes are necessary to maintain satisfactory operations relative to changes in the user's environment. Maintenance often includes minor enhancements or corrections to the problem that surface late in the systems operations. In the implementation phase, the team builds the components either from scratch or by composition. Given the architecture document meant from the design phase and the requirement document from the analysis phase, the team should build exactly what has been requested, though there is still room for innovation and flexibility. For example, a component may be narrowly designed for this particular system, or the component may be made more general to satisfy a reusability

- Careful planning
- Investigation of system and constraints
- Design the methods to achieve changeover.
- Training the staff in the changed phase.
- Evaluation of change over method.
- The method of implementation and time scale to be adopted are found out initially.

7.2 Maintenance

This phase occurs as a result of deploying the whole system at the end users organization. They will perform the beta testing at the end users and inform to the developers about any needed modification to the application. The customer records all the problems that are encountered during the beta testing and reports these to the developer at regular intervals.

7.2.1 Corrective Maintenance

Even with the best quality assurance activities, it is likely that the customer will uncover defects in the software. Corrective maintenance changes the software to correct the defects. Corrective Maintenance activity may consist of repair,

restoration or replacement of equipment. This activity will be result of a regular inspection, which identifies the failure in time for corrective maintenance to be planned and scheduled, then performed during a routine maintenance shutdown.

7.2.2 Adaptive Maintenance

Over time, the original environment(CPU, operating system, business rules, external product characteristics) for which the software was developed is likely to change. Adaptive maintenance results in modification to the software to accommodate changes to its external environment.

7.2.3 Enhanced Maintenance

As software is used, the customer/user will recognize additional functions that will provide the benefit. Perfect maintenance extends the software beyond its original functional requirements.

7.2.4 Preventive Maintenance

Computer software deteriorates due to change, and because of this preventive maintenance often called software re-engineering, must be conducted to enable the software to serve the needs of its end users. Preventive maintenance makes changes to computer programs so that they can be more easily corrected, adapted and enhanced.

Chapter 8

8 Conclusion and Future Scope

8.1 Conclusion

The main objective of the project is to develop a system that can capture and scan images of crop fields and identify weeds from crops. Methodology: The project uses image processing and machine learning techniques to detect weeds. The image processing techniques include color segmentation, edge detection, and morphological operations. The machine learning techniques include feature extraction, classification, and localization. Dataset: The project uses a custom dataset of plant images, which contains images of different crop and weed species. The dataset is divided into training, validation, and testing sets.

8.2 Future Scope

- Live detection

Appendix

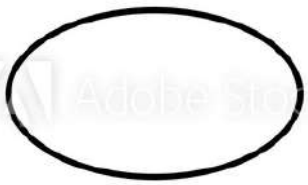
A ACTIVITY DIAGRAM

An activity diagram is a type of UML (Unified Modeling Language) diagram that visually represents the flow of control or the sequence of activities in a system, process, or workflow. It is used to model the behavior of a system and illustrate how different activities or actions relate to each other. Activity diagrams consist of nodes, which represent activities or actions, and edges, which represent the flow of control between activities. Nodes can also have additional properties, such as conditions or constraints, that determine when they are executed. Activity diagrams are particularly useful for visualizing complex processes, identifying bottlenecks, and improving the efficiency of workflows.

A.1 External source or receiver

A source or sink is a person or part of organization ,which enters or receives information from the system,but is considered to be outlining the contest of data flow model.

A.2 Transform process

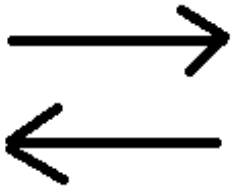


A process represents transformation where incoming data flowsare changed into outgoing data flow

A.3 Input/Output

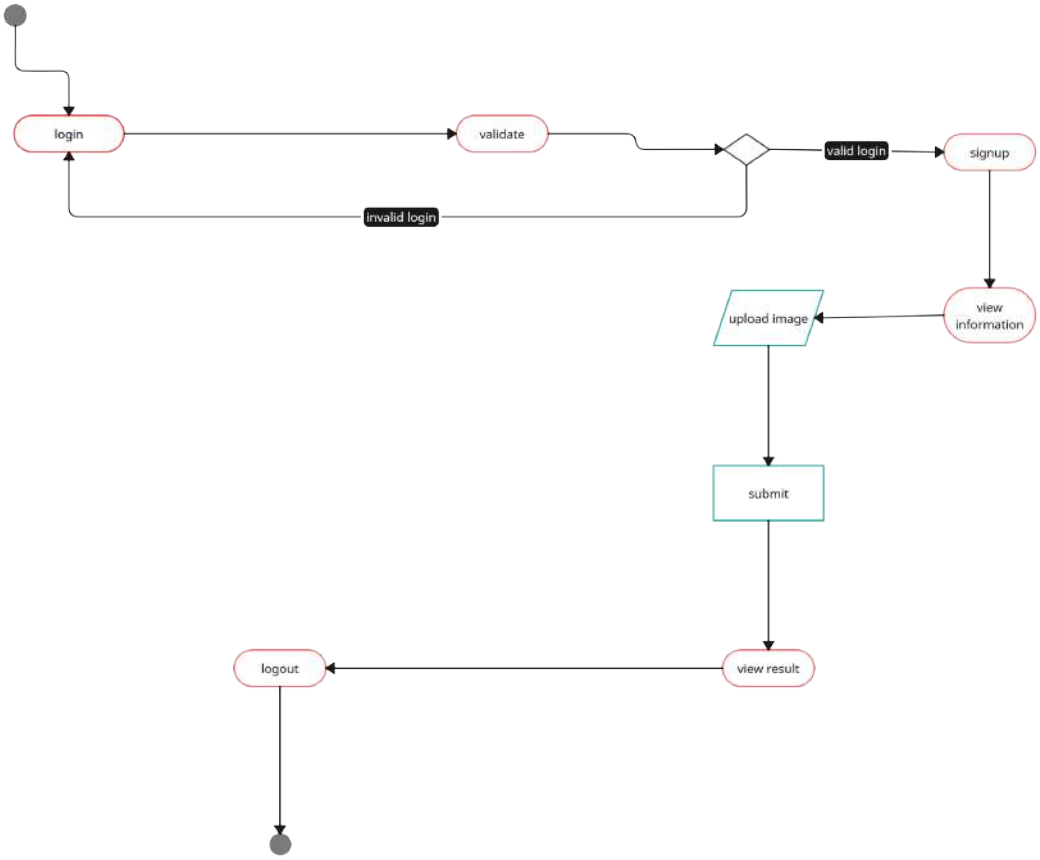
The input/output symbol represents a point in the process where data enters or exits the system. It is used to indicate the input or output of data, information, or materials at a specific point in the workflow. The input/output symbol is typically represented as a rectangle with a thin vertical line on the left side for inputs and a thin vertical line on the right side for outputs.

A.4 Data flow



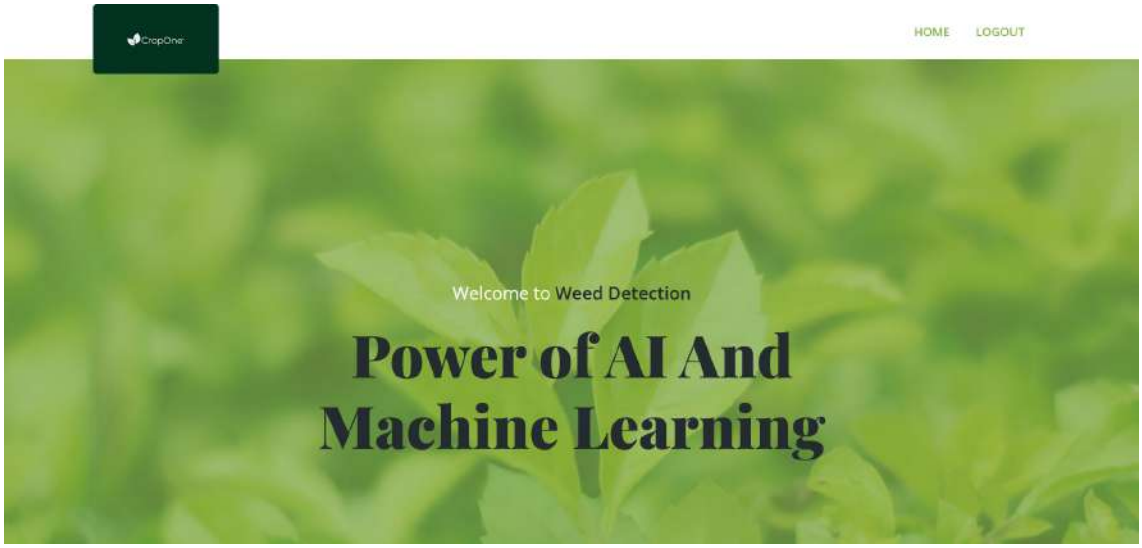
A data flow is a route, which enable packets of data to travel from one point to another.Data may flow,with arrowhead pointing in the direction of the flow

A.5 Diagram

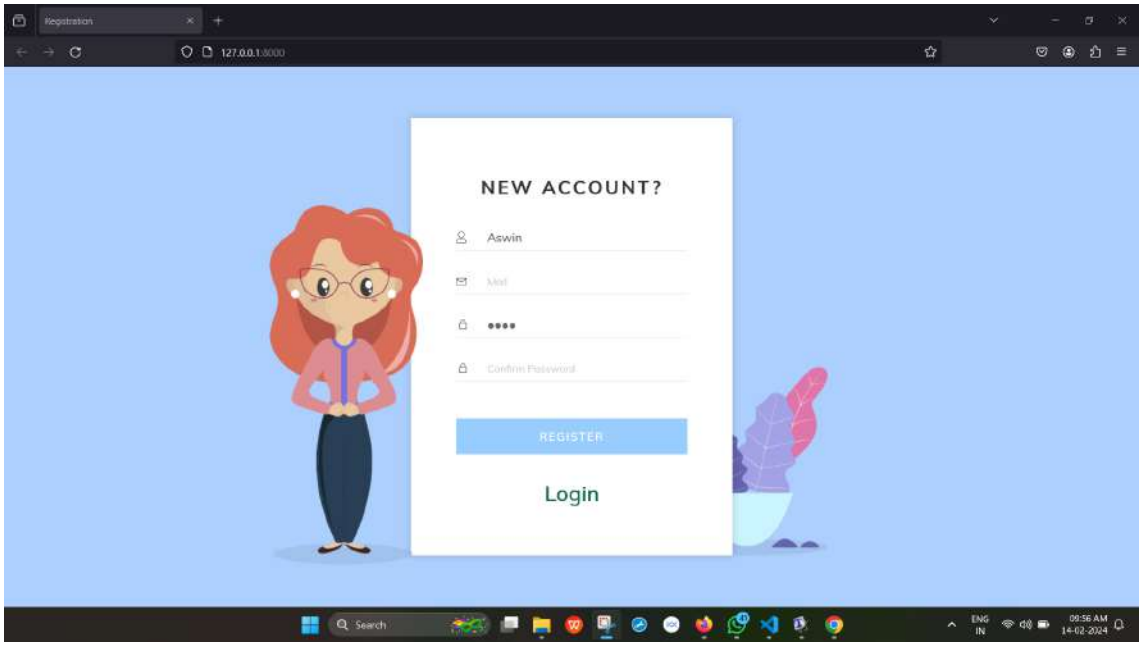

login
validate
valid login
signup
invalid login
view information
upload image
submit
view result
logout

B USER INTERFACES

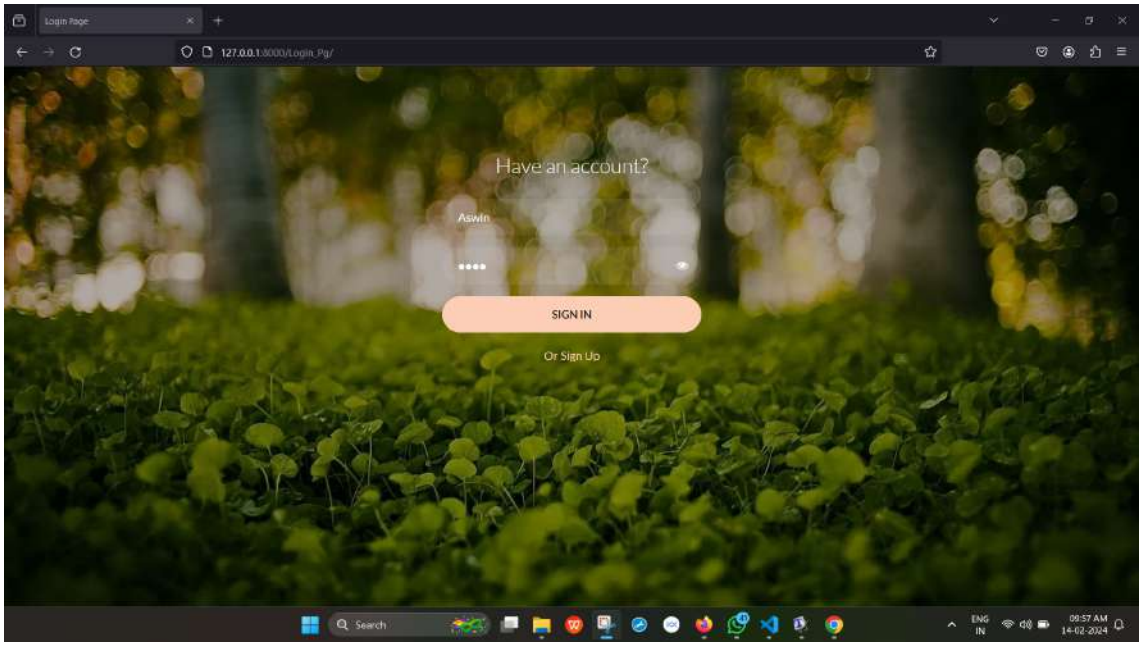
B.1 HOME



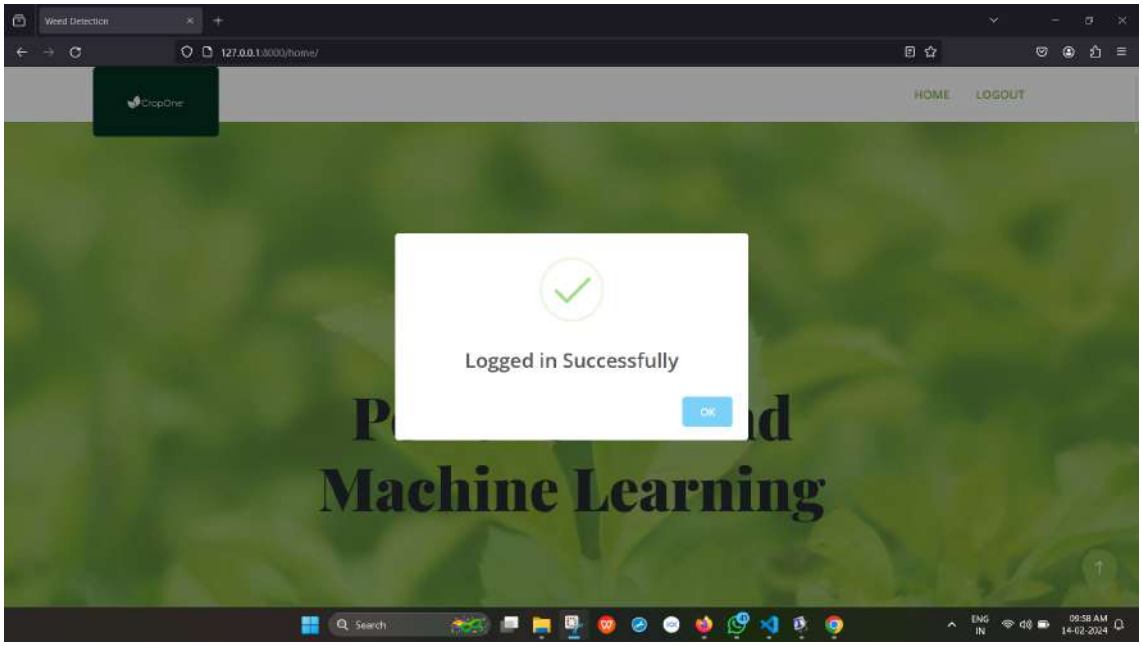
B.2 REGISTER PAGE



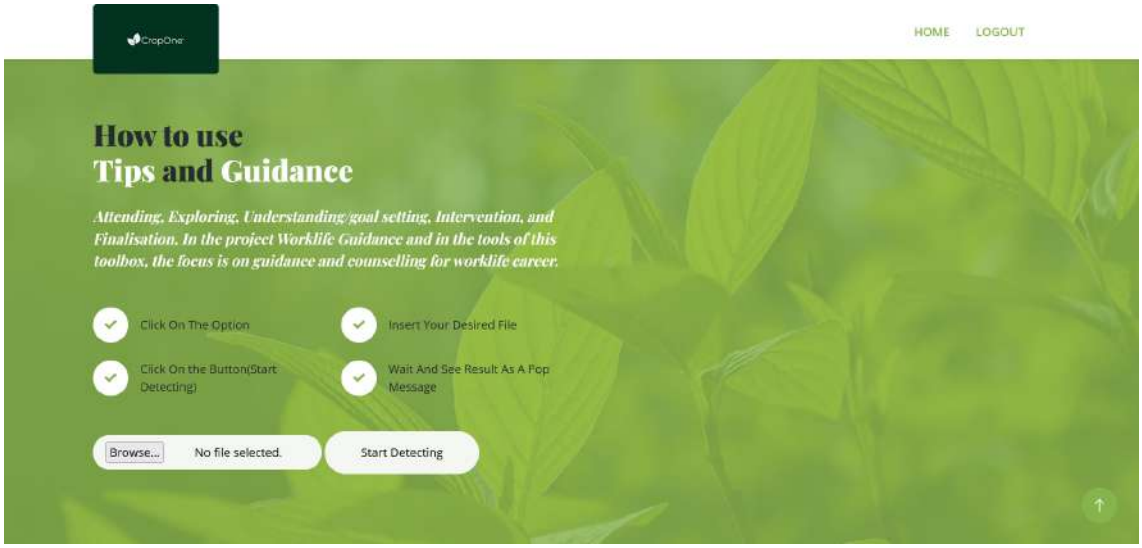
B.3 SIGNUP PAGE



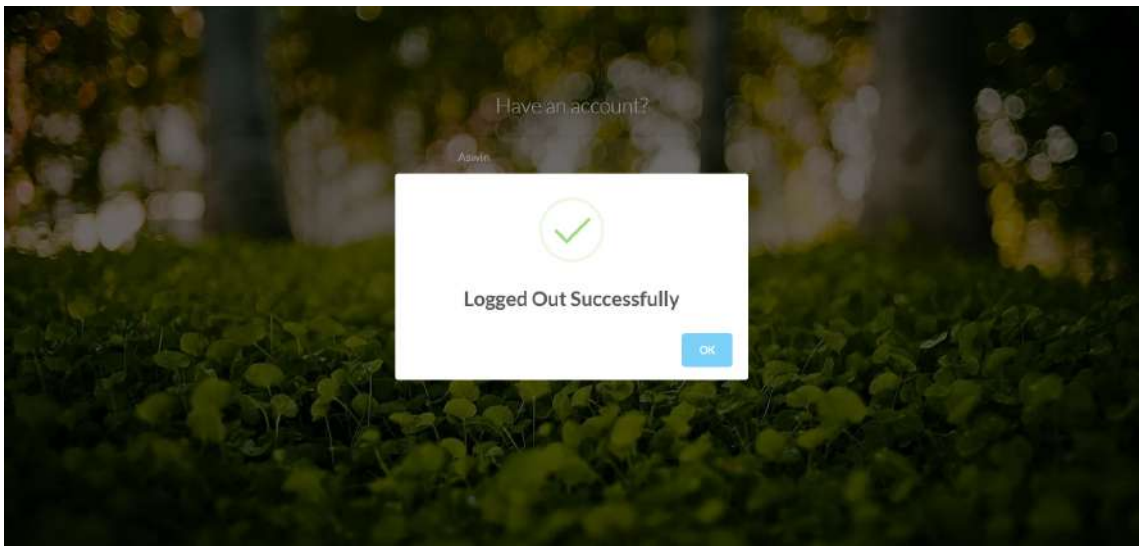
B.4 LOGIN CONFIRMATION



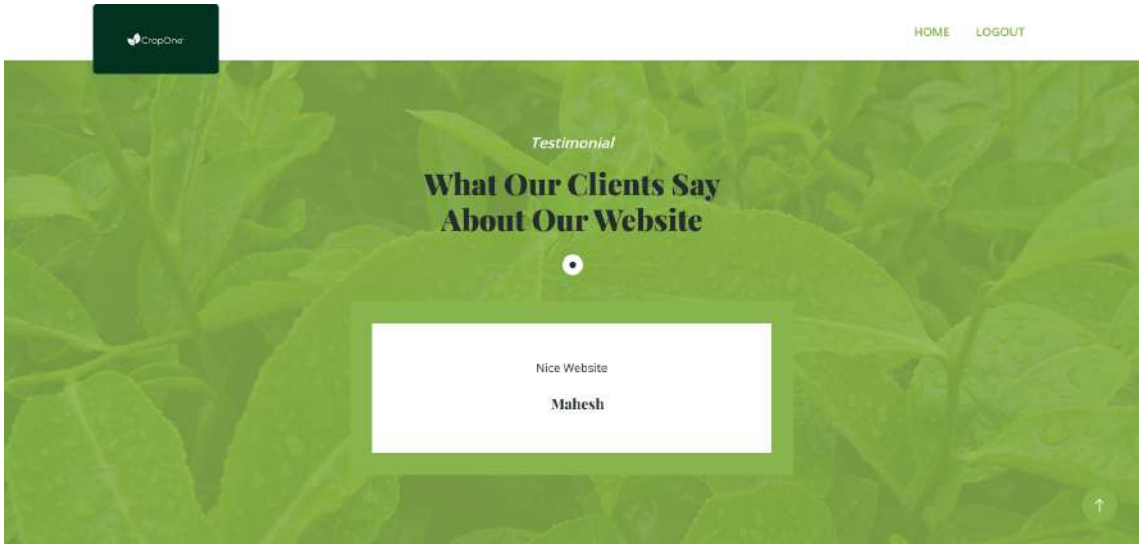
B.5 USER INPUT



B.6 LOGOUT CONFIRMATION



B.7 REVIEW


CropOne
HOME
LOGOUT
Testimonial
What Our Clients Say
About Our Website
Nice Website
Mahesh

C CODE

style.css

```
/********** Template CSS **********/
:root {
    --primary: #88B44E;
    --secondary: #FB9F38;
    --light: #F5F8F2;
    --dark: #252C30;
}

.back-to-top {
    position: fixed;
    display: none;
    right: 30px;
    bottom: 30px;
    z-index: 99;
}

.fw-medium {
    font-weight: 600;
}

.fw-bold {
    font-weight: 700;
}

.fw-black {
    font-weight: 900;
}


/*** Spinner ***/
#spinner {
    opacity: 0;
    visibility: hidden;
    transition: opacity .5s ease-out, visibility 0s linear .5s;
    z-index: 99999;
}

#spinner.show {
    transition: opacity .5s ease-out, visibility 0s linear 0s;
    visibility: visible;
    opacity: 1;
}
```

```
/*** Button ***/
.btn {
    transition: .5s;
    font-weight: 500;
}

.btn-primary,
.btn-outline-primary:hover {
    color: var(--light);
}

.btn-secondary,
.btn-outline-secondary:hover {
    color: var(--dark);
}

.btn-square {
    width: 38px;
    height: 38px;
}

.btn-sm-square {
    width: 32px;
    height: 32px;
}

.btn-lg-square {
    width: 48px;
    height: 48px;
}

.btn-square,
.btn-sm-square,
.btn-lg-square {
    padding: 0;
    display: flex;
    align-items: center;
    justify-content: center;
    font-weight: normal;
}


/*** Navbar ***/
.sticky-top {
```

```
    top: -150px;
    transition: .5s;
}

.navbar .navbar-brand {
    position: absolute;
    padding: 0;
    width: 170px;
    height: 135px;
    top: 0;
    left: 0;
}

.navbar .navbar-nav .nav-link {
    margin-right: 35px;
    padding: 25px 0;
    color: var(--dark);
    font-weight: 600;
    text-transform: uppercase;
    outline: none;
}

.navbar .navbar-nav .nav-link:hover,
.navbar .navbar-nav .nav-link.active {
    color: var(--primary);
}

.navbar .dropdown-toggle::after {
    border: none;
    content: "\f107";
    font-family: "Font Awesome 5 Free";
    font-weight: 900;
    vertical-align: middle;
    margin-left: 8px;
}

@media (max-width: 991.98px) {
    .navbar .navbar-brand {
        width: 126px;
        height: 100px;
    }

    .navbar .navbar-nav .nav-link  {
        margin-right: 0;
        padding: 10px 0;
    }
```

```
    .navbar .navbar-nav {
        margin-top: 75px;
        border-top: 1px solid #EEEEEE;
    }
}

@media (min-width: 992px) {
    .navbar .nav-item .dropdown-menu {
        display: block;
        border: none;
        margin-top: 0;
        top: 150%;
        opacity: 0;
        visibility: hidden;
        transition: .5s;
    }

    .navbar .nav-item:hover .dropdown-menu {
        top: 100%;
        visibility: visible;
        transition: .5s;
        opacity: 1;
    }
}


/*** Header ***/
.carousel-caption {
    top: 0;
    left: 0;
    right: 0;
    bottom: 0;
    display: flex;
    align-items: center;
    background: rgba(136, 180, 78, .7);
    z-index: 1;
}

.carousel-control-prev,
.carousel-control-next {
    width: 15%;
}

.carousel-control-prev-icon,
.carousel-control-next-icon {
```

```
    width: 3.5rem;
    height: 3.5rem;
    border-radius: 3.5rem;
    background-color: var(--dark);
    border: 15px solid var(--dark);
}

@media (max-width: 768px) {
    #header-carousel .carousel-item {
        position: relative;
        min-height: 450px;
    }

    #header-carousel .carousel-item img {
        position: absolute;
        width: 100%;
        height: 100%;
        object-fit: cover;
    }
}

.page-header {
background: linear-gradient(rgba(136, 180, 78, .7), rgba(136,
180, 78, .7)), url(../img/carousel-1.jpg) center center
no-repeat;
    background-size: cover;
}

.page-header .breadcrumb-item+.breadcrumb-item::before {
    color: var(--light);
}

.page-header .breadcrumb-item,
.page-header .breadcrumb-item a {
    font-size: 18px;
    color: var(--light);
}


/*** Section Title ***/
.section-title {
    position: relative;
    margin-bottom: 3rem;
    padding-bottom: 2rem;
}
```

```
.section-title::before {
    position: absolute;
    content: "";
    width: 50%;
    height: 2px;
    bottom: 0;
    left: 0;
    background: var(--primary);
}

.section-title::after {
    position: absolute;
    content: "";
    width: 28px;
    height: 28px;
    bottom: -13px;
    left: calc(25% - 13px);
    background: var(--dark);
    border: 10px solid #FFFFFF;
    border-radius: 28px;
}

.section-title.text-center::before {
    left: 25%;
}

.section-title.text-center::after {
    left: calc(50% - 13px);
}


/*** Products ***/
.product {
background: linear-gradient(rgba(136, 180, 78, .1), rgba(136,
180, 78, .1)),
    url(../img/product-bg.png) left bottom no-repeat;
    background-size: auto;
}

.product-carousel .owl-nav {
    display: flex;
    justify-content: center;
    margin-top: 30px;
}

.product-carousel .owl-nav .owl-prev,
```

```
.product-carousel .owl-nav .owl-next {
    margin: 0 10px;
    width: 55px;
    height: 55px;
    display: flex;
    align-items: center;
    justify-content: center;
    color: #FFFFFF;
    background: var(--primary);
    border-radius: 55px;
    box-shadow: 0 0 45px rgba(0, 0, 0, .15);
    font-size: 25px;
    transition: .5s;
}

.product-carousel .owl-nav .owl-prev:hover,
.product-carousel .owl-nav .owl-next:hover {
    background: #FFFFFF;
    color: var(--primary);
}


/*** About ***/
.video {
background: linear-gradient(rgba(136, 180, 78, .85), rgba(136,
180, 78, .85)),
    url(../img/video-bg.jpg) center center no-repeat;
    background-size: cover;
}

.btn-play {
    position: relative;
    display: block;
    box-sizing: content-box;
    width: 65px;
    height: 75px;
    border-radius: 100%;
    border: none;
    outline: none !important;
    padding: 28px 30px 30px 38px;
    background: #FFFFFF;
}

.btn-play:before {
    content: "";
    position: absolute;
```

```
    z-index: 0;
    left: 50%;
    top: 50%;
    transform: translateX(-50%) translateY(-50%);
    display: block;
    width: 120px;
    height: 120px;
    background: #FFFFFF;
    border-radius: 100%;
    animation: pulse-border 1500ms ease-out infinite;
}

.btn-play:after {
    content: "";
    position: absolute;
    z-index: 1;
    left: 50%;
    top: 50%;
    transform: translateX(-50%) translateY(-50%);
    display: block;
    width: 120px;
    height: 120px;
    background: #FFFFFF;
    border-radius: 100%;
    transition: all 200ms;
}

.btn-play span {
    display: block;
    position: relative;
    z-index: 3;
    width: 0;
    height: 0;
    left: 13px;
    border-left: 40px solid var(--primary);
    border-top: 28px solid transparent;
    border-bottom: 28px solid transparent;
}

@keyframes pulse-border {
    0% {
transform: translateX(-50%) translateY(-50%) translateZ(0)
scale(1);
        opacity: 1;
    }
```

```
    100% {
transform: translateX(-50%) translateY(-50%) translateZ(0)
scale(2);
        opacity: 0;
    }
}

.modal-video .modal-dialog {
    position: relative;
    max-width: 800px;
    margin: 60px auto 0 auto;
}

.modal-video .modal-body {
    position: relative;
    padding: 0px;
}

.modal-video .close {
    position: absolute;
    width: 30px;
    height: 30px;
    right: 0px;
    top: -30px;
    z-index: 999;
    font-size: 30px;
    font-weight: normal;
    color: #FFFFFF;
    background: #000000;
    opacity: 1;
}


/*** Store ***/
.store-item .store-overlay {
    position: absolute;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: center;
    background: rgba(138, 180, 78, .3);
    opacity: 0;
```

```
    transition: .5s;
}

.store-item:hover .store-overlay {
    opacity: 1;
}


/*** Contact ***/
.contact .btn-square {
    width: 100px;
    height: 100px;
    border: 20px solid var(--light);
    background: var(--primary);
    border-radius: 50px;
}


/*** Testimonial ***/
.testimonial {
background: linear-gradient(rgba(136, 180, 78, .85), rgba(136,
180, 78, .85)),
    url(../img/testimonial-bg.jpg) center center no-repeat;
    background-size: cover;
}

.testimonial-item {
    margin: 0 auto;
    max-width: 600px;
    text-align: center;
    background: #FFFFFF;
    border: 30px solid var(--primary);
}

.testimonial-item img {
    width: 60px !important;
    height: 60px !important;
    border-radius: 60px;
}

.testimonial-carousel .owl-dots {
    margin-top: 35px;
    display: flex;
    align-items: flex-end;
    justify-content: center;
}
```

```
.testimonial-carousel .owl-dot {
    position: relative;
    display: inline-block;
    margin: 0 5px;
    width: 15px;
    height: 15px;
    background: var(--primary);
    border-radius: 15px;
    transition: .5s;
}

.testimonial-carousel .owl-dot.active {
    width: 30px;
    background: var(--dark);
}


/*** Footer ***/
.footer {
    color: #B0B9AE;
}

.footer .btn.btn-link {
    display: block;
    margin-bottom: 5px;
    padding: 0;
    text-align: left;
    color: #B0B9AE;
    font-weight: normal;
    text-transform: capitalize;
    transition: .3s;
}

.footer .btn.btn-link::before {
    position: relative;
    content: "\f105";
    font-family: "Font Awesome 5 Free";
    font-weight: 900;
    color: var(--primary);
    margin-right: 10px;
}

.footer .btn.btn-link:hover {
    color: var(--light);
    letter-spacing: 1px;
```

```
    box-shadow: none;
}

.copyright {
    color: #B0B9AE;
}

.copyright {
    background: #252525;
}

.copyright a:hover {
    color: #FFFFFF !important;
}
```

home.html

```
<!DOCTYPE html>
{% load static %}
<html lang="en">

<head>
<meta charset="utf-8">
<title>Weed Detection</title>
<meta content="width=device-width, initial-scale=1.0"
name="viewport">
<meta content="" name="keywords">
<meta content="" name="description">

<!-- Favicon -->
<link href="{% static 'img/favicon.ico' %}" rel="icon">

<!-- Google Web Fonts -->
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com"
crossorigin>
<link href="https://fonts.googleapis.com/css2?
family=Open+Sans:wght@400;600&family=Playfair+Display:wght@700;900&display=swap"
rel="stylesheet">

<!-- Icon Font Stylesheet -->
<link
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.10.0/css/all.min.css"
rel="stylesheet">
<link
```

```
href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.4.1/font/bootstrap-icons.css"
rel="stylesheet">

<!-- Libraries Stylesheet -->
<link href="{% static 'lib/animate/animate.min.css' %}"
rel="stylesheet">
<link href="{% static
'lib/owlcarousel/assets/owl.carousel.min.css' %}"
rel="stylesheet">

<!-- Customized Bootstrap Stylesheet -->
<link href="{% static 'css/bootstrap.min.css' %}"
rel="stylesheet">

<!-- Template Stylesheet -->
<link href="{% static 'css/style.css' %}" rel="stylesheet">
</head>

<body>
<!-- Spinner Start -->
<div id="spinner" class="show bg-white position-fixed
translate-middle w-100 vh-100 top-50
start-50 d-flex align-items-center justify-content-center">
<div class="spinner-border text-primary" role="status"
style="width: 3rem; height: 3rem;"></div>
</div>
<!-- Spinner End -->


<!-- Navbar Start -->
<div class="container-fluid bg-white sticky-top">
<div class="container">
<nav class="navbar navbar-expand-lg bg-white navbar-light py-2
py-lg-0">
<a href="index.html" class="navbar-brand">
<img class="img-fluid" src="{% static 'img/crop1.jpg' %}"
alt="Logo" width="200px" height="60px">
</a>
<button type="button" class="navbar-toggler ms-auto me-0"
data-bs-toggle="collapse" data-bs-target="#navbarCollapse">
<span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="navbarCollapse">
<div class="navbar-nav ms-auto">
<a href="{% url 'home' %}" class="nav-item nav-link
active">Home</a>
```

```
{% if request.session.username %}
<a href="{% url 'Logout_fn' %}" class="nav-item nav-link
active">Logout</a>
{% else %}
<a href="{% url 'RegistrationForm' %}" class="nav-item nav-link
active">Register</a>
{% endif %}

</div>
</div>



</nav>
</div>
</div>
<!-- Navbar End -->


<!-- Carousel Start -->
<div class="container-fluid px-0 mb-5">
<div id="header-carousel" class="carousel slide carousel-fade"
data-bs-ride="carousel">
<div class="carousel-inner">
<div class="carousel-item active">
<img class="w-100" src="{% static 'img/carousel-1.jpg' %}"
alt="Image">
<div class="carousel-caption">
<div class="container">
<div class="row justify-content-center">
<div class="col-lg-7 text-center">
<p class="fs-4 text-white animated zoomIn">Welcome to
<strong class="text-dark">Weed Detection</strong></p>
<h1 class="display-1 text-dark mb-4 animated zoomIn">
Power of AI And Machine Learning</h1>
</div>
</div>
</div>
</div>
</div>

</div>
</div>
</div>
<!-- Carousel End -->
```

```
<!-- About Start -->
<div class="container-xxl py-5">
<div class="container">
<div class="row g-5">
<div class="col-lg-6">
<div class="row g-3">
<div class="col-6 text-end">
<img class="img-fluid bg-white w-200 mb-3 wow fadeIn"
data-wow-delay="0.1s" src="{% static 'img/cropandweed.jpeg' %}"
alt="">
<img class="img-fluid bg-white w-300 wow fadeIn"
data-wow-delay="0.2s" src="{% static 'img/cropandweed2.jpeg' %}"
alt="">
</div>
<div class="col-6">
<img class="img-fluid bg-white w-200 mb-3 wow fadeIn"
data-wow-delay="0.3s" src="{% static 'img/cropandweed3.jpeg' %}"
alt="">
<img class="img-fluid bg-white w-300 wow fadeIn"
data-wow-delay="0.4s" src="{% static 'img/cropandweed4.jpeg' %}"
alt="">
</div>
</div>
</div>
<div class="col-lg-6 wow fadeIn" data-wow-delay="0.5s">
<div class="section-title">
<p class="fs-5 fw-medium fst-italic text-primary">About
Project</p>
<h1 class="display-6">What is weed detection?</h1>
</div>
<div class="row g-3 mb-4">
<div class="col-sm-4">
<img class="img-fluid bg-white w-150" src="{% static
'img/cropandweed5.jpeg' %}" alt="">
</div>
<div class="col-sm-8">
<h5>Weed detection systems are important solutions to one of the
existing
agricultural problems|unmechanized weed control. </h5>
<p class="mb-0"> Weed detection also helps provide a means of
reducing or eliminating herbicide use, mitigating agricultural
environmental
and health impact, and improving sustainability.</p>
</div>
</div>
```

```
<div class="border-top mb-4"></div>
<div class="row g-3">
<div class="col-sm-8">
<h5>What is the objective for weed detection project?</h5>
<p class="mb-0">The main objective has been to obtain a formula
so that a
weed detection system can be developed through binary
classifications.
The initial step of image processing is the detection of green
plants in order to
eliminate all the soil in the image, reducing information that
is not necessary.</p>
</div>
<div class="col-sm-4">
<img class="img-fluid bg-white w-150" src="{% static
'img/cropandweed5.jpeg' %}" alt="">
</div>
</div>
</div>
</div>
</div>
</div>
<!-- About End -->


<!-- Products Start -->
<div class="container-fluid product py-5 my-5">
<div class="container py-5">
<div class="section-title text-center mx-auto wow fadeInUp"
data-wow-delay="0.1s"
 style="max-width: 500px;">
<p class="fs-5 fw-medium fst-italic text-primary">Our Source</p><h1 class="display-6">
What is the role of AI in
agriculture?</h1>
</div>
<div class="owl-carousel product-carousel wow fadeInUp"
data-wow-delay="0.5s">
<a href="" class="d-block product-item rounded">
<img src="img/product-1.jpg" alt="">
<div class="bg-white shadow-sm text-center p-4 position-relative
mt-n5 mx-4">
<h4 class="text-primary">AI Specification</h4>
<span class="text-body">By analyzing data from various sources,
AI can help farmers make data-driven decisions,
optimize resource usage, and reduce environmental impact.AI can
be used to develop
```

```
 image recognition systems that can.
identify weeds with a high degree of accuracy</span>
</div>
</a>
<a href="" class="d-block product-item rounded">
<img src="img/product-2.jpg" alt="">
<div class="bg-white shadow-sm text-center p-4 position-relative
mt-n5 mx-4">
<h4 class="text-primary">General Science</h4>
<span class="text-body">For example, the World Economic Forum
has reported that
 AI integration in agriculture could bring about
a 60% decrease in pesticide usage and a 50% reduction in water
usage.
AI algorithms can analyze the chemical composition of soil
samples to determine which nutrients may be lacking.
AI can also identify or even predict crop diseases. </span>
</div>
</a>
<a href="" class="d-block product-item rounded">
<img src="img/product-3.jpg" alt="">
<div class="bg-white shadow-sm text-center p-4 position-relative
mt-n5 mx-4">
<h4 class="text-primary">Weed</h4>
<span class="text-body">Weeds are unwanted and undesirable
 plants which interfere with the. utilization of land and water
 resources and thus adversely affect human welfare.
They can also be referred as plants out of place. Weeds compete
with the
beneficial and desired vegetation in crop lands, forests,
aquatic systems etc.</span>
</div>
</a>
</div>
</div>
</div>
<!-- Products End -->


<!-- Article Start -->
<div class="container-xxl py-5">
<div class="container">
<div class="row g-5">
<div class="col-lg-5 wow fadeIn" data-wow-delay="0.1s">
<img class="img-fluid" src="{% static 'img/article.jpg' %}"
alt="">
```

```
</div>
<div class="col-lg-6 wow fadeIn" data-wow-delay="0.5s">
<div class="section-title">
<p class="fs-5 fw-medium fst-italic text-primary">Featured
Acticle</p>
<h1 class="display-6">The history of Weed Detection</h1>
</div>
<p class="mb-4">In olden days weed detection was done
by employing some men, especially for weed removal purpose.
They will detect the weeds by checking each and every plant
field
. Then they will pluck them out manually using their
hands or spades. In proposed system the weeds can be detected
by image processing techniques. The main aim of the
project is to identify weed affected area for more seeding.
Since there are different issues with respect to the
developed yields on the field, a standout amongst the majority
is about
weeds which go as a hindrance in the development of the
harvests. Weeds may break down the life and nature of the yields
if
it is not controlled legitimately in time.
This proposed thought centers around to reduce the work and also
the time
needed to identify weeds and expel the same.
During this process they will consider Edge detection and
colour segmentation process.
They will analyze each and every crop with their intensity
colour,
Intensity, edge, Size etc., are been obtained as output.
The colour of the edges and veins of the crop and weed are
white after segmentation process and edge detection,
whereas the rest of the image is totally black in colour.
After the process of Edge detection and Colour
segmentation it has undergone the process of Filtering.
The Filtering can be a type of process in which each and every
crop can be identified and their gain value,
tradeoff's, edge, frequency of crop and weed can be identified
and their threshold values can obtained </p>

</div>
</div>
</div>
</div>
<!-- Article End -->
```

```
<!-- Video Start -->
<div class="container-fluid video my-5">
<div class="container">
<div class="row g-0">
<div class="col-lg-6 py-5 wow fadeIn" data-wow-delay="0.1s">
<div class="py-5">
<h1 class="display-6 mb-4">How to use <br><span
class="text-white">
Tips</span> and <span class="text-white">Guidance</span></h1>
<h5 class="fw-normal lh-base fst-italic text-white
mb-5">Attending, Exploring
, Understanding/goal setting, Intervention, and Finalisation.
In the project Worklife Guidance and in the tools of this
toolbox,
the focus is on guidance and counselling for worklife
career.</h5>
<div class="row g-4 mb-5">
<div class="col-sm-6">
<div class="d-flex align-items-center">
<div class="flex-shrink-0 btn-lg-square bg-white text-primary
rounded-circle me-3">
<i class="fa fa-check"></i>
</div>
<span class="text-dark">Click On The Option</span>
</div>
</div>
<div class="col-sm-6">
<div class="d-flex align-items-center">
<div class="flex-shrink-0 btn-lg-square bg-white text-primary
rounded-circle me-3">
<i class="fa fa-check"></i>
</div>
<span class="text-dark">Insert Your Desired File</span>
</div>
</div>
<div class="col-sm-6">
<div class="d-flex align-items-center">
<div class="flex-shrink-0 btn-lg-square bg-white text-primary
rounded-circle me-3">
<i class="fa fa-check"></i>
</div>
<span class="text-dark">Click On the Button(Start
Detecting)</span>
</div>
```

```
</div>
<div class="col-sm-6">
<div class="d-flex align-items-center">
<div class="flex-shrink-0 btn-lg-square bg-white text-primary
rounded-circle me-3">
<i class="fa fa-check"></i>
</div>
<span class="text-dark">Wait And See Result As A Pop
Message</span>
</div>
</div>
</div>
<form action="{% url 'detect_weed_or_crop' %}" method="post"
enctype="multipart/form-data">
{% csrf_token %}
<input type="file" class="btn btn-light rounded-pill py-2 px-3"
required name="image_file">
{% if request.session.username %}
<button class="btn btn-light rounded-pill py-3 px-5">Start
Detecting</button>
</form>
{% else %}


<a href="{% url 'RegistrationForm' %}" class="btn btn-light
rounded-pill py-3 px-5">Register</a>



{% endif %}


<div class="modal fade" id="exampleModal" tabindex="-1"
aria-labelledby="exampleModalLabel" aria-hidden="true">
<div class="modal-dialog">
<div class="modal-content">
<div class="modal-header">
<h1 class="modal-title fs-5" id="exampleModalLabel">Result</h1>
<button type="button" class="btn-close" data-bs-dismiss="modal"
aria-label="Close"></button>
</div>
<div class="modal-body">
{{result.result}}
</div>
<div class="modal-footer">
<button type="button" class="btn btn-secondary"
```

```
data-bs-dismiss="modal">Close</button>

</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>


<!-- Testimonial Start -->
<div class="container-fluid testimonial py-5 my-5">

<div class="container py-5">
<div class="section-title text-center mx-auto wow fadeInUp"
 data-wow-delay="0.1s" style="max-width: 500px;">
<p class="fs-5 fw-medium fst-italic text-white">Testimonial</p>
<h1 class="display-6">What Our Clients Say About Our
Website</h1>
</div>

<div class="owl-carousel testimonial-carousel wow fadeInUp"
data-wow-delay="0.5s">
{% for i in x %}
<div class="testimonial-item p-4 p-lg-5">
<p class="mb-4">{{i.Description}}</p>
<div class="d-flex align-items-center justify-content-center">
<div class="text-start ms-3">
<h5>{{i.username}}</h5>
</div>
</div>
</div>
{% endfor %}
</div>

</div>

</div>
<!-- Testimonial End -->


<!-- Contact Start -->
<div class="container-xxl contact py-5">
```

```
<div class="container">
<div class="section-title text-center mx-auto wow fadeInUp"
 data-wow-delay="0.1s" style="max-width: 500px;">
<h1 class="display-6">Overview</h1>
</div>
<div class="row justify-content-center wow fadeInUp"
data-wow-delay="0.1s">
<div class="col-lg-8">
<p class="text-center mb-5">Artificial intelligence,
specifically deep learning, is a fast-growing research field
today. One of its
various applications is object recognition,
making use of computer vision. The combination of these
two technologies leads to the purpose of this thesis.
 In this project, a system for the identification of
different crops and weeds has been developed as an
 alternative to the system present on the FarmBot
company's robots. This is done by accessing the
 images through the FarmBot API, using computer
vision for image processing, and artificial intelligence
for the application of transfer learning to a RCNN
that performs the plants identification autonomously.
 The results obtained show that the system
works with an accuracy of 78.10
% for the main crop and 53.12% and 44.76% for the two weeds
considered. Moreover, the coordinates of the weeds
 are also given as results. The performance of the
resulting system is compared both with
 similar projects found during research, and with the current
version of the FarmBot weed detector. </p>
<div class="row g-5">
<div class="col-md-4 text-center wow fadeInUp"
data-wow-delay="0.3s">
<div class="btn-square mx-auto mb-3">
<i class="fa fa-envelope fa-2x text-white"></i>
</div>
<p class="mb-2">name@example.com</p>
<p class="mb-0">name1@example.com</p>
</div>
<div class="col-md-4 text-center wow fadeInUp"
data-wow-delay="0.4s">
<div class="btn-square mx-auto mb-3">
<i class="fa fa-phone fa-2x text-white"></i>
</div>
<p class="mb-2">+012 345 67890</p>
</div>
```

```
<div class="col-md-4 text-center wow fadeInUp"
data-wow-delay="0.5s">
<div class="btn-square mx-auto mb-3">
<i class="fa fa-map-marker-alt fa-2x text-white"></i>
</div>
<p class="mb-2">Irinjalakuda</p>
<p class="mb-0">Thrissur, Kerala</p>
</div>
</div>
</div>
</div>
</div>
</div>
<!-- Contact Start -->

<form action="{% url 'ReviewSave' %}" method="post">
{% csrf_token %}

<div style="margin-left:450px;">
<div class="form-floating">
<input type="hidden" name="uname"
value="{{request.session.username}}">
<input class="form-control" placeholder="Insert Your Review
Here"
id="message" type="text" name="txt" style="height:
120px;width:550px;">
<label for="message">Review</label>
</div>
<br>
<div class="col-12">
{% if request.session.username %}
<button class="btn btn-primary rounded-pill py-3 px-4"
style="margin-left:200px;" type="submit">
Submit Review</button>

{% else %}
<a href="{% url 'RegistrationForm' %}" class="btn btn-primary
rounded-pill py-3 px-4"
style="margin-left:200px;">Register</a>


{% endif %}


</div>
</div>
```

```
</form>


<!-- Footer Start -->
<div class="container-fluid bg-dark footer mt-5 py-5 wow fadeIn"
data-wow-delay="0.1s">
<div class="container py-5">
<div class="row g-5">
<div class="col-lg-3 col-md-6">
<h4 class="text-primary mb-4">Authencity</h4>
<p class="mb-2"><i class="fa fa-map-marker-alt text-primary
me-3">
</i>Fort Street, Ernakulam , Kerala</p>
<p class="mb-2"><i class="fa fa-phone-alt text-primary me-3
"></i>+012 345 67890</p>
<p class="mb-2"><i class="fa fa-envelope text-primary me-3">
</i>name@example.com</p>
<div class="d-flex pt-3">
<a class="btn btn-square btn-primary rounded-circle me-2"
href="#">
<i class="fab fa-twitter"></i></a>
<a class="btn btn-square btn-primary rounded-circle me-2"
href="#">
<i class="fab fa-facebook-f"></i></a>
<a class="btn btn-square btn-primary rounded-circle me-2"
href="#">
<i class="fab fa-youtube"></i></a>
<a class="btn btn-square btn-primary rounded-circle me-2"
href="#">
<i class="fab fa-linkedin-in"></i></a>
</div>
</div>
<div class="col-lg-3 col-md-6">
<h4 class="text-primary mb-4">Quick Links</h4>
<a class="btn btn-link" href="{% url 'home' %}">Home</a>
{% if request.session.username%}
<a class="btn btn-link" href="{% url 'Logout_fn' %}">Logout</a>
{% else %}
<a class="btn btn-link" href="{% url 'RegistrationForm' %}">Sign
Up</a>
{% endif %}
</div>
<div class="col-lg-3 col-md-6">
<h4 class="text-primary mb-4">Business Hours</h4>
<p class="mb-1">Monday - Saturday</p>
<h6 class="text-light">24/5</h6>
```

```
<p class="mb-1">Saturday</p>
<h6 class="text-light">09:00 am - 12:00 pm</h6>
<p class="mb-1">Sunday</p>
<h6 class="text-light">Closed</h6>
</div>
</div>
</div>
</div>
<!-- Footer End -->


<!-- Copyright Start -->
<div class="container-fluid copyright py-4">
<div class="container">
<div class="row">
<div class="col-md-6 text-center text-md-start mb-3 mb-md-0">
&copy; <a class="fw-medium" href="#">Weed Detection</a>, All
Right Reserved.
</div>
<div class="col-md-6 text-center text-md-end">
<!--/*** This template is free as long as you keep the footer
author's credit
link/attribution link/backlink. If you'd like to use the
template without the
footer author's credit link/attribution link/backlink, you can
purchase the
Credit Removal License from
"https://htmlcodex.com/credit-removal".
Thank you for your support. ***/-->
</div>
</div>
</div>
</div>
<!-- Copyright End -->


<!-- Back to Top -->
<a href="#" class="btn btn-lg btn-primary btn-lg-square
rounded-circle back-to-top">
<i class="bi bi-arrow-up"></i></a>


<!-- JavaScript Libraries -->
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.1/jquery.min.js"></script>
<script
```

```
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0/dist/js/bootstrap.bundle.min.js">
</script>
<script src="{% static 'lib/wow/wow.min.js' %}"></script>
<script src="{% static 'lib/easing/easing.min.js' %}"></script>
<script src="{% static 'lib/waypoints/waypoints.min.js'
%}"></script>
<script src="{% static 'lib/owlcarousel/owl.carousel.min.js'
%}"></script>

<!-- Template Javascript -->
<script src="{% static 'js/main.js' %}"></script>
<script
src="https://unpkg.com/sweetalert/dist/sweetalert.min.js"></script>
{% if messages %}
{% for i in messages %}
{% if i.tags == 'warning' %}
<script>
swal('{{i}}','', 'warning');
</script>
{% elif i.tags == 'error' %}
<script>
swal('{{i}}','', 'error');
</script>
{% else %}
<script>
swal('{{i}}','', 'success');
</script>
{% endif %}

{% endfor %}
{% endif %}
</body>


</html>
```

login.html

```
<!DOCTYPE html>
{% load static %}
<html>
<head>
<meta charset="utf-8">
<title>Registration</title>
<meta name="viewport" content="width=device-width,
```

```
initial-scale=1.0">

<!-- LINEARICONS -->
<link rel="stylesheet" href="{% static
'assets/fonts/linearicons/style.css' %}">

<!-- STYLE CSS -->
<link rel="stylesheet" href="{% static 'assets/css/style.css'
%}">
</head>

<body>

<div class="wrapper">
<div class="inner">
<img src="{% static 'assets/images/image-1.png' %}" alt=""
class="image-1">
<form action="{% url 'Registration_save' %}" method="post">
{% csrf_token %}
<h3>New Account?</h3>
<div class="form-holder">
<span class="lnr lnr-user"></span>
<input type="text" class="form-control" placeholder="Username"
name="uname" required>
</div>

<div class="form-holder">
<span class="lnr lnr-envelope"></span>
<input type="email" class="form-control" placeholder="Mail"
name="email" required>
</div>
<div class="form-holder">
<span class="lnr lnr-lock"></span>
<input type="password" class="form-control"
placeholder="Password" name="password" required>
</div>
<div class="form-holder">
<span class="lnr lnr-lock"></span>
<input type="password" class="form-control" placeholder="Confirm
Password" name="cpassword" required>
</div>
<button type="submit">
<span>Register</span>
</button>
<br><br>
```

```
<h1 style="text-align:center;"><a style="color: rgb(23, 107,
79);" href="{% url 'Login_Pg' %}">Login</a></h1>
</form>

<img src="{% static 'assets/images/image-2.png' %}" alt=""
class="image-2">
</div>

</div>

<script src="{% static 'assets/js/jquery-3.3.1.min.js'
%}"></script>
<script src="{% static 'assets/js/main.js' %}"></script>
<script
src="https://unpkg.com/sweetalert/dist/sweetalert.min.js"></script>
{% if messages %}
{% for i in messages %}
{% if i.tags == 'warning' %}
<script>
swal('{{i}}','', 'warning');
</script>
{% elif i.tags == 'error' %}
<script>
swal('{{i}}','', 'error');
</script>
{% else %}
<script>
swal('{{i}}','', 'success');
</script>
{% endif %}

{% endfor %}
{% endif %}
</body><!-- This templates was made by Colorlib
(https://colorlib.com) -->
</html>

\\
```

Registrartion.html

```
<!DOCTYPE html>
{% load static %}
<html>
<head>
<meta charset="utf-8">
```

```
<title>Registration</title>
<meta name="viewport" content="width=device-width,
initial-scale=1.0">

<!-- LINEARICONS -->
<link rel="stylesheet" href="{% static
'assets/fonts/linearicons/style.css' %}">

<!-- STYLE CSS -->
<link rel="stylesheet" href="{% static 'assets/css/style.css'
%}">
</head>

<body>

<div class="wrapper">
<div class="inner">
<img src="{% static 'assets/images/image-1.png' %}" alt=""
class="image-1">
<form action="{% url 'Registration_save' %}" method="post">
{% csrf_token %}
<h3>New Account?</h3>
<div class="form-holder">
<span class="lnr lnr-user"></span>
<input type="text" class="form-control" placeholder="Username"
name="uname" required>
</div>

<div class="form-holder">
<span class="lnr lnr-envelope"></span>
<input type="email" class="form-control" placeholder="Mail"
name="email" required>
</div>
<div class="form-holder">
<span class="lnr lnr-lock"></span>
<input type="password" class="form-control"
placeholder="Password" name="password" required>
</div> <div class="form-holder">
<span class="lnr lnr-lock"></span>
<input type="password" class="form-control" placeholder="Confirm
Password" name="cpassword" required>
</div>
<button type="submit">
<span>Register</span>
</button>
<br><br>
```

```
<h1 style="text-align:center;"><a style="color: rgb(23, 107,
79);" href="{% url 'Login_Pg' %}">Login</a></h1>
</form>

<img src="{% static 'assets/images/image-2.png' %}" alt=""
class="image-2">
</div>

</div>

<script src="{% static 'assets/js/jquery-3.3.1.min.js'
%}"></script>
<script src="{% static 'assets/js/main.js' %}"></script>
<script
src="https://unpkg.com/sweetalert/dist/sweetalert.min.js"></script>
{% if messages %}
{% for i in messages %}
{% if i.tags == 'warning' %}
<script>
swal('{{i}}','', 'warning');
</script>
{% elif i.tags == 'error' %}
<script>
swal('{{i}}','', 'error');
</script>
{% else %}
<script>
swal('{{i}}','', 'success');
</script>
{% endif %}

{% endfor %}
{% endif %}
</body>
</html>
```

apps.py

```
from django.apps import AppConfig


class FrontendConfig(AppConfig):
    default_auto_field = 'django.db.models.BigAutoField'
    name = 'Frontend'
```

models.py

```
from django.db import models

# Create your models here.
class Review(models.Model):
username=models.CharField(max_length=500,null=True,blank=True)
Description=models.CharField(max_length=5000,null=True,blank=True)

class Registration(models.Model):
username=models.CharField(max_length=100,null=True,blank=True
)Email=models.EmailField(max_length=100,null=True,blank=True)
Password=models.CharField(max_length=8,null=True,blank=True)
Confirm_Password=models.CharField(max_length=8,null=True,blank=True)
```

urls.py

```
from django.urls import path
from Frontend import views

urlpatterns=[
    path('home/',views.home,name="home"),
path('detect_weed_or_crop/',views.detect_weed_or_crop,name="detect_weed_or_crop"),

    path('ReviewSave/',views.ReviewSave,name="ReviewSave"),
    path('',views.RegistrationForm,name="RegistrationForm"),

path('Registration_save/',views.Registration_save,name="Registration_save"),
    path('Login_Pg/',views.Login_Pg,name="Login_Pg"),

    path('Login_fun/',views.Login_fun,name="Login_fun"),
    path('Logout_fn/',views.Logout_fn,name="Logout_fn"),
]
```

views.py

```
from django.shortcuts import render,redirect
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import numpy as np
from keras.applications.imagenet_utils import preprocess_input
from django.contrib import messages
from PIL import Image
from Frontend.models import Review,Registration
```

```
from django.http import HttpResponse,JsonResponse
from tempfile import NamedTemporaryFile
import io
from django.shortcuts import redirect,render
from keras.applications.imagenet_utils import preprocess_input
from django.contrib import messages
from Frontend.models import Review,Registration
import tensorflow as tf
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.mobilenet_v2 import
preprocess_input
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.image as mpimg




def home(request):
    x=Review.objects.all()
    return render(request,'Home.html',{'x':x})


def detect_weed_or_crop(request):
    if request.method == "POST":
        try:
            model_path = 'weed_final.h5'
            model = tf.keras.models.load_model(model_path)

            uploaded_file = request.FILES['image_file']
            with NamedTemporaryFile(delete=False) as temp_file:
                temp_file.write(uploaded_file.read())
                temp_file_path = temp_file.name
img = image.load_img(temp_file_path, target_size=(224, 224))
  img_array = image.img_to_array(img)
            img_array = np.expand_dims(img_array, axis=0)
            img_data = preprocess_input(img_array)

            predictions = model.predict(img_data)

            class_labels = ["crop", "weed", "objects"]
            class_probabilities = predictions[0]

class_percentages = [prob * 100 for prob in class_probabilities]
            threshold = 0.5
```

```
detected_classes = [class_labels[i] for i, prob in
enumerate(class_probabilities) if prob > threshold]

            img = mpimg.imread(temp_file_path)
            plt.imshow(img)
            plt.axis('off')

if "crop" in detected_classes and "weed" in detected_classes:
  crop_index = class_labels.index("crop")
                weed_index = class_labels.index("weed")

                crop_percentage = class_percentages[crop_index]
                weed_percentage = class_percentages[weed_index]

                weed_to_crop_ratio = weed_percentage /
       crop_percentage if crop_percentage != 0 else 0

title = f'Weed detected in crop: Crop: {crop_percentage:.2f}
%, Weed: {weed_percentage:.2f}%, Weed to Crop Ratio:
{weed_to_crop_ratio:.2f}'
                title_color = 'purple'
            elif "crop" in detected_classes:
                crop_index = class_labels.index("crop")
                crop_percentage = class_percentages[crop_index]
title = f'Crop detected: {crop_percentage:.2f}
%'              title_color = 'green'
            elif "weed" in detected_classes:
                weed_index = class_labels.index("weed")
                weed_percentage = class_percentages[weed_index]
title = f'Weed detected: {weed_percentage:.2f
}%'                title_color = 'red'
            else:
                title = 'Neither weed or crop detected'
                title_color = 'black'

context = {'title': title, 'title_color': title_color,
'class_percentages': class_percentages,
                        'detected_classes': detected_classes}
            print(title)
            messages.success(request, title)

            return render(request, "Home.html", context)
        except Exception as e:
            messages.error(request, f"Error: {str(e)}")
            return render(request, "Home.html")
```

```
    messages.error(request, "Invalid Request")
    return render(request, "Home.html")

def ReviewSave(req):
    if req.method=="POST":
        nm=req.POST.get('uname')
        des=req.POST.get('txt')
        x=Review(username=nm,Description=des)
        x.save()
        messages.success(req,"Review Submitted Successfully")
        return redirect(home)

def RegistrationForm(req):
    return render(req,"Registration.html")



def Registration_save(request):
    if request.method == "POST":
        nm = request.POST.get('uname')
        em = request.POST.get('email')
        passw = request.POST.get('password')
        con = request.POST.get('cpassword')
        if passw != con:
messages.error(request, "Password and confirm password do not
match.")
            return redirect(RegistrationForm)
registration = Registration(username=nm, Email=em,
Password=passw, Confirm_Password=con)
        registration.save()
        messages.success(request,"Registered Succesfully")
        return redirect(Login_Pg)

def Login_Pg(req):
    return render(req,"Login_Pg.html")

def Login_fun(request):
    if request.method=="POST":
        nm=request.POST.get('uname')
        pwd=request.POST.get('password')
if
Registration.objects.filter(username=nm,Password=pwd).exists():
            request.session['username']=nm
            request.session['Password']=pwd
            messages.success(request, "Logged in Successfully")
            return redirect(home)
```

```
        else:
            messages.warning(request, "Check Your Credentials")
            return redirect(Login_Pg)
    else:
messages.warning(request, "Check Your Credentials Or Sign Up ")
 return redirect(Login_Pg)

def Logout_fn(request):
    del request.session['username']
    del request.session['Password']
    messages.success(request, "Logged Out Successfully")
    return redirect(Login_Pg)
```

manage.py

```
#!/usr/bin/env python
"""Django's command-line utility for administrative tasks."""
import os
import sys


def main():
    """Run administrative tasks."""
os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'Weed.settings')    try:
from django.core.management import execute_from_command_line
 except ImportError as exc:
        raise ImportError(
"Couldn't import Django. Are you sure it's installed and
""available on your PYTHONPATH environment variable? Did you "
"forget to activate a virtual environment?"
        ) from exc
    execute_from_command_line(sys.argv)


if __name__ == '__main__':
    main()
```

urls2.py

```
"""Weed URL Configuration

The ‘urlpatterns‘ list routes URLs to views. For more
information please see:
    https://docs.djangoproject.com/en/3.2/topics/http/urls/
```

```
Examples:
Function views
    1. Add an import:  from my_app import views
2. Add a URL to urlpatterns:  path('', views.home, name='home')Class-based views
    1. Add an import:  from other_app.views import Home
2. Add a URL to urlpatterns: path('', Home.as_view(),
name='home')
Including another URLconf
1. Import the include() function: from django.urls import
include, path
2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
"""
from django.contrib import admin
from django.urls import path,include
from django.contrib.staticfiles.urls import
staticfiles_urlpatterns,static
from . import settings
import Frontend.urls
from Frontend import views
urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include(Frontend.urls)),
    path('/',views.RegistrationForm,name="RegistrationForm")
]

urlpatterns+=staticfiles_urlpatterns()
urlpatterns+=static(settings.MEDIA_URL,document_root=settings.MEDIA_ROOT)
```

Dr.BABY
VACCINATION APPLICATION

PROJECT REPORT

Submitted By

MILAN MATHACHAN

Reg. No. CCAVBCA007

Bachelors

in Computer Application
(University of Calicut)

under the guidance of

Mr.Thoufeeq Ansari

Assistant Professor

Bachelors in Computer Application
DEPARTMENT OF COMPUTER SCIENCE
CHRIST COLLEGE(Autonomous)
IRINJALAKUDA, KERALA
INDIA

2021 - 2024

DEPARTMENT OF COMPUTER SCIENCE

CHRIST COLLEGE (AUTONOMOUS)

IRINJALAKUDA

CERTIFICATE

This is to certify that the project report entitled ***"Dr. Baby-Vaccination Application"*** *is a bonfied record of the project work done by* ***Milan Mathachan*** *in partial fulfillment of the requirement for the sixth semester of* ***Bachelor of Computer Application*** *in Department of Computer Science of* **CHRIST COLLEGE (AUTONOMOUS) , IRINJALAKUDA**

Ms. Thoufeeq Ansari
Assistant Professor,CS
Internal Guide

Ms. Sini Thomas
Head of the Department
Computer Science

EXTERNAL EXAMINER

INTERNAL EXAMINER

DECLARATION

I here by declare that this project work **"Dr.BABY - VACCINATION APPLICATION"** submitted by Christ College (Autonomous)Irinjalakuda , affliated to Calicut University in partial fullfillment of the requirement for the award of the Bachelor of Computer Application, is a record of original work done by me,under the guidance of Mr.THOUFEEQ ANSARI,Department of Computer Science.

Place : Irinjalakuda MILAN MATHACHAN

ACKNOWLEDGEMENT

First and foremost I would like to thank Lord almighty for his providence and for being the guiding light throughout the project. I wish to express my sincere gratitude to our beloved Department head for giving me all the facilities for our project. I take this opportunity to express my gratitude to the class teacher Ms. SOWMYA P.S and head of the department Ms. SINI THOMAS who has been a great support for me throughout the course of this project. I am very thankful for her aspiring guidance and valuable advice during the project work. I would like to express my sincere thanks to our project guide Mr. THOUFEEQ ANSARI for supporting and guiding throughout the project. I take this opportunity to specially thank all other faculty members for their constant and continuous motivation. Finally I would like to thank my family and friends for giving valuable advice and moral support throughout our project.

ABSTRACT

Dr. BABY is a comprehensive mobile application designed to simplify and streamline the process of managing baby vaccinations. With a user-friendly interface and intuitive features, Dr.Baby aims to alleviate the burden on parents and caregivers by providing timely reminders, educational resources, and appointment scheduling tools. The application utilizes personalized vaccination schedules based on the baby's age, ensuring adherence to recommended immunization timelines. Additionally, Dr.Baby offers real-time updates on vaccine availability and information on local healthcare providers and clinics. By promoting vaccination compliance and empowering parents with valuable knowledge, Dr.Baby strives to contribute to improved public health outcomes and safeguarding the well-being of infants worldwide.

Contents

Chapter 1

1 Introduction

We have developed an application, suitable for both iOS and androind user. Our application mainly focuses on the vaccinations mandatory for the new born babies. The application contains a calender , booking system and a chatbot. It gives out a notification when is the time for the baby's vaccination schedule. This app is designed to assist the parents of the new born with vaccination tracking.

1.1 Overview

The objective of our Dr. BABY - VACCINATION APPLICATION is to ensure that all the new born's do not miss out on the mandatory vaccinations that can be life treatening. The application includes many features that is required to assist the parents with the vaccinations, such as a chatbot, calender scheduling. It is a User friendly application.

Chapter 2

2 System Analysis

2.1 Purpose

The main purpose of this application is to make the vaccination journey of the new born's easier and effortless for the parents.

2.1.1 Existing System

There are many applications for women to keep track of periods, vaccination apps developed during the covid era and so on, but an app for the parents of the new born are still not available. Limitations of existing system :We cannot the information as well as a schedule for the vaccination for the baby.

2.1.2 Proposed System

The proposed system is an flutter - based application that is developed to help parents of the new born to track all the mandatory vaccinations. We also provide a lot of other health related services along with the vaccination scheduling.

2.2 Problem definition

To know what the problem is and what the needs are before developing it. Here we designed an application for vaccinations specially for the new-born babies.

2.3 FEASIBILITY STUDY

After the problem is clearly understood and solutions proposed, the next step is to conduct the feasibility study. Feasibility study is defined as evaluation or analysis of the potential impact of the proposed project or program. The objective is to determine whether the proposed system is feasible. There are three aspects of feasibility study to which the proposed system is subjected as discussed below.

2.3.1 Technical Feasibility

Technical feasibility assess whether the current technical resources are sufficient for the new system. We can upgrade the level of technology for supporting our application. We checked whether the proposed system can be implemented in the present system without supporting the existing hardware.

2.3.2 Economical Feasibility

Economic feasibility determines whether the time and money are available to develop the system. There is no additional hardware used to develop the site,so it is inexpensive to build.

2.3.3 Operational Feasibility

Operational feasibility determines if the human resources are available to operate the system once it has been installed. No extra training is needed to use this system. Anyone who has knowledge in internet in English language can easily use the system. The resources that are required to implement or install are already available with the organization.

Chapter 3

3 Software Requirement Specification

3.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the Dr.BABY - VACCINATION APPLICATION. It illustrate the purpose and complete description for the development of the system. It explain system constraints, interface and interactions with other external applications. This document is primarily intended to be proposed to help with vaccination journey and bookings of the vaccination trp

3.2 Scope

The application gives all the details about the vaccinaton dates and booking system.It also provides a chatbot that helps with any doubts arising to the parents.

3.3 Overall Description

Dr.BABY is a comprehensive mobile application designed to streamline and enhance the vaccination experience for parents and caregivers of infants. The app is dedicated to ensuring that babies receive timely and appropriate vaccinations, helping to protect them from various preventable diseases. It serves as a one-stop solution for managing vaccination schedules, accessing reliable information, and facilitating communication with healthcare providers.

3.3.1 Product Perspective

Dr.BABY APPLICATION is mainly used to know the details about your next vaccination trip. It also provide features that gives you a booking system as well as a chatbot that helps with all the users doubts.

3.3.2 Product Functionality

Through this system, admin can upload various data based on the vaccination help services, health advices etc. User will be able to view their vaccine schedule in a calender and will also be directed to a booking page to book the vaccine appointememt.

3.3.3 Users and Characteristics

There are two users - admin and parent. The admin can regulate everything in the system and they can add new hospitals and booking slots. The users/parent

will be able to schedule their vaccination trips as well as book vaccination slots at the nearest hospitals.

3.4 Specific Requirements

3.4.1 Hardware Requirements

- System: LAPTOP-S2BJ4TFP
- Processor: Intel Core i7-1065G7 CPU
- Speed: 1.30 GHz - 1.50GHz
- RAM capacity: 12 GB
- Hard Dsk drive: 952 GB

3.4.2 Software Requirements

- Front End : Flutter
- Back End : Python Django
- Database : MySql
- IDE : Visual Studio Code

3.5 Functional Requirements

It contains two main modules :

- 1.Admin
- 2.Parent

Admin

The admin will be able to handle the database in person. It can also add on more and more hospital location, booking area's etc. The admin also has access to user database that enable the admin to remove any malicious users.

Parent

The users/parent will be able to view the calender that has all the vaccine schedules that has been already set up according to the DOB of the child from child profile. They also have access to various other supporting services. Chatbot and booking system is few of the services that assists the user/parent for a smooth vaccinaton scheduling, booking etc.

3.6 Non Functional Requirements

Non-functional requirements define the overall qualities or attributes of the resulting system. Non-functional system place restrictions on the product being developed, the developed process, and specify external constraints that the product must meet. Examples of non-functional requirements include safety, security, usability, reliability and performance requirements. Project management issues (costs, time and schedule) are often considered as non-functional requirements. The principal non - functional constraints which are relevant to critical systems :

- performance
- security
- safety
- usability

Performance

Performance requirements concern the speed of operation of a system. Types of performance requirements :

- Response requirements (how quickly the system reacts to a user input).
- Throughput requirements (how much the system can accomplish within a specified amount of time).
- Availability requirements (is the system available for service when requested by end users). The speed of operation of this system is adequate for the requirements.

Reliability

- Reliability is the ability of a system to perform its required function under stated conditions for a specified period of time.
- constraints on the runtime behavior of the system. This system is reliable because its functionalities can be done on the required conditions.

Safety

Safety requirements are not required which exclude unsafe situation from the possible solution of the system.

Usability

Usability is the ease with which a user can learn to operate, prepare inputs for, and interpret outputs of system or components.

Usability requirements include :

- information error messages.
- well-formed user interfaces.

3.7 Interface Requirements

3.7.1 Hardware interfaces

The system must run over the internet, all the hardware shall require to connect internet will be hardware interface for the system. As for example modem, WAN-LAN.

3.7.2 Software interfaces

Software interface required for the working of the project is the appropriate operating system.

3.7.3 Communication interfaces

The user of the site communicate using the network connectivity and the data set is accessed

3.8 Security Requirements

- User accesses only their account.
- Validation of input is handled.
- This application containing the computer systems is physically secured against arms or surreptitious entry by intruders.
- Users must be authorized carefully to reduce changes of any such user giving access to an intruder in exchange for a bribe or other favour.

3.9 Platform Used

Windows 11, the latest operating system from Microsoft, introduces a refreshed and modernized user interface with a centered Start menu, rounded corners, and enhanced Snap layouts for improved multitasking. The taskbar has been redesigned, featuring a simplified and more streamlined look. Windows 11 also brings advancements in gaming with DirectStorage and Auto HDR, providing a more immersive gaming experience. The Microsoft Store has undergone a significant overhaul, offering a wider range of apps and a more user-friendly interface. Additionally, Windows 11 focuses on productivity enhancements, such as the integration of Microsoft Teams directly into the taskbar for seamless communication. With its fresh design and enhanced features, Windows 11 aims to deliver a more intuitive and enjoyable computing experience for users.

3.10 Technologies Used

Flutter

Flutter is an open-source UI software development toolkit created by Google, providing a comprehensive framework for building natively compiled applications across mobile, web, and desktop from a single codebase. Launched in 2017, Flutter has gained significant popularity among developers due to its fast development cycles, expressive and flexible UI, and native performance. Using Dart as its programming language, Flutter enables developers to create visually appealing and responsive applications that can run seamlessly on different platforms, fostering a streamlined and efficient development process. With a rich set of pre-designed widgets, a strong community support, and continuous updates, Flutter empowers developers to build high-quality, cross-platform applications that deliver a consistent user experience across various devices.

SQLite 3

SQLite is a lightweight, self-contained, and serverless relational database management system (RDBMS) that is widely used for embedded systems, mobile applications, and small to medium-sized projects. Developed as a C library, SQLite offers a simple and efficient solution for storing and retrieving data without the need for a separate server process. It operates on a single ordinary disk file and supports standard SQL syntax, providing users with a familiar relational database experience. SQLite is known for its ease of use, portability, and minimal setup requirements, making it a popular choice for developers seeking a straightforward and compact database solution for various applications.

Python Django

Django is a high-level web framework for building robust and scalable web applications using the Python programming language. Developed with the principles of simplicity, flexibility, and maintainability in mind, Django follows the Model-View-Controller (MVC) architectural pattern, known as Model-View-Template (MVT) in Django's terminology. It comes equipped with a wide range of built-in features, including an Object-Relational Mapping (ORM) system for database management, a powerful templating engine, and a comprehensive administration interface. Django encourages the implementation of clean, reusable code through its "Don't Repeat Yourself" (DRY) philosophy, enabling developers to focus on application logic rather than boilerplate code. With a vibrant and supportive community, Django has become a popular choice for web developers, offering a quick and efficient way to create feature-rich web applications.

Chapter 4

4 Design Document

4.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the Dr.BABY APPLICATION. It illustrate the purpose and complete description for the development of the system. It explain system constraints, interface and interactions with other external applications. This document is primarily intended to help the users to schedule see their upcoming vaccines. . The application provides all the details about the vaccinations, date on which the vaccines are to be taken, booking system and other services. The app also provides all the details of the vaccination to be taken.

4.2 Scope

The scope of this application can be comprehensive, addressing various aspects related to the vaccination process, healthcare, and information management for infants and toddlers.

4.3 Overview

The purpose of this document is to help the reader to visualize the solution to the project presented.This document verifies how the design meet the requirement stimulated in the SRS document through design viewpoints.The design viewpoints will cover all design elements presented before. This document will provide a direct approach to the development of this project hence reducing feature creep and ponitedly determine the quality of the design.

4.4 Data Design

Database are the storehouses of data used in the software systems.The data is stored in tables inside the database.Several tables are created for the manipulation of the data for the system.Two essential settings for a database are

- Primary Key- The field that is unique for all the record occurrence.
- Foreign Key- The field used to set relation between tables.

Normalization

- Normalization is the process of organizing the data in the database.
- Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate the undesirable characteristics like Insertion, Update and Deletion Anomalies.

- Normalization divides the larger table into the smaller table and links them using relationship.
- The normal form is used to reduce redundancy from the database table.

Tables

Name	DataType	Constraints	Description
UserID	int(11)	Primary key	ID of users
username	varchar(50)	Not null	Name of users
password	varchar(250)	Not null	Password of users
email	varchar(100)	Not null	Email of the users
FirstName	varchar(200)	Not null	First name of user
LastName	varchar(20)	Not null	Last name of user
Phone	varchar(50)	Not null	Contact number
Role	varchar(250)	Not null	Relation to baby

Table 1: User Account

Name	DataType	Constraints	Description
BabyID	int(11)	Primarykey	ID of baby
UserID	int(9)	Foreignkey	ID from User table
FirstName	varchar(200)	Notnull	First name of baby
LastName	varchar(20)	Notnull	Last name of baby
Date of Birth	date()	Notnull	Babies date of birth
Gender	char(1)	Notnull	Gender of the baby
Location	varcher(5)	Notnull	Place of baby

Table 2: Babies Profile

Name	DataType	Constraints	Description
VaccineID	int(11)	Foreignkey	id from vaccine name table
VaccineName	varchar(50)	Notnull	Name of vaccines
Vaccine_Available	varchar(50)	Notnull	Available Vaccines

Table 3: Vaccine Table

Name	DataType	Constraints	Description
Vaccine_Status_ID	int(11)	Primarykey	id for vaccine status table
VaccineName	varchar(50)	Foreignkey	Reference to vaccine name table
BabyID	int(11)	Foreignkey	Id of the baby,refernce to baby profile

Table 4: Vaccine Status

Name	DataType	Constraints	Description
Vaccine_Booking_ID	int(11)	Primarykey	Id for vaccine booking table
ParentName	varchar(50)	Foreignkey	Reference to user profile table
Parent_Email	varchar(50)	Notnull	Email for vaccine booking
Hospital	varchar(50)	Foreignkey	Rreference to hospital table
Vaccine_Available	varchar(50)	Foreignkey	Reference to vaccine table
Booking_Date	date()	Notnull	Date of vaccine

Table 5: Vaccine Booking

Name	DataType	Constraints	Description
Hospital_ID	int(11)	Primarykey	Id for hospital table
Hospital	varchar(50)	Notnull	Hospital name
Slots_Available	varchar(50)	Notnull	Vaccine Slots
Vaccine_Available	varchar(50)	Foreignkey	Reference to vaccine table

Table 6: Hospital Table

Chapter 5

5 Development of the System

The development of the baby vaccination app involves a comprehensive approach to ensure the health and well-being of infants. The app is designed to streamline and simplify the vaccination process, providing parents with a user-friendly platform to keep track of their baby's immunization schedule. It includes features such as personalized vaccination reminders, informative content about each vaccine, and a secure digital record of the child's vaccination history.Development considerations encompass data security, user accessibility, and collaboration with healthcare professionals to ensure accurate and reliable information.

Chapter 6

6 System Testing

Testing is the penultimate step of software development.An elaborate testing of the data is prepared and the system is using the test data,while doing testing,errors are noted and correction is made.The users are trained to operate the developed system.Both hardware and software securities are made to run the developed system successfully.System testing is aimed at ensuring the system works accurately before the live operation commences.

6.1 Test Plan

6.1.1 Scope

This test plan will cover the following testing activities as identified in the testing strategy.

- White Box Testing

White box testing is a testing technique, that examines the program structure and derives test data from the program logic/code. In white box testing, the UI is bypassed. Inputs and outputs of ratings are tested directly at the code level and the results are compared against specifications. It Reveals errors in "hidden" code.

- Black Box Testing

Black - box testing is a method of software testing that examines the functionality of an application based on the specifications. Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end - user would.

- Unit Testing

The module includes four main components to undergo unit testing. It checks the syntax error, logic error and validity of the program correctness. This test will be performed by the developers.

- Integration Testing

After all components pass the unit testing, they get to be tested whether they work correctly when they are running concurrently and communicating to each other. Integration tests exercise an entire subsystem and ensure that a set of components play nicely together.

- Internal Data Testing

We will test the validity of the data before it enters the database to avoid any problems that may face us in the database. We will test the encryption of the personal information of all the users along with the user names and passwords to ensure maximum security of the users privacy.

6.1.2 Software risk issues

In this section, the plan is to test the risk involved in critical issues such as: Difficult to run on visual studio due to large loading time. Some inherent software risks such as complexity exist, Proper network connection,Working of SQLite 3 database.

6.1.3 Features to be tested

- Test whether correct user name and password allows you to login.
- Test whether invalid user name and password prevents you from login.
- Test whether there is any connection problem in the Server.
- Test whether the parent and admin details are entered correctly.
- Test whether invalid data entry allows saving data successfully.
- Test whether all pages are loaded correctly.
- Test whether the provided security works correctly.

6.2 Test consolidation

6.2.1 Test item

The items or features to be tested in the test cases are included in the document. Each and every user input is tested. The present condition of the system is tested. It is checked to make sure that environment is ready for the application to work.

6.2.2 Input specifications

These are the inputs required to execute the test case. All required inputs including data elements and values are given below.

Data	Values
Email	A valid email ID
Password	Combination of characters and numbers
Phone number	10 Digit number

Chapter 7

7 System Implementation and Maintenance

7.1 Implementation

System implementation is the conversion of new system into an operating one which involves creating compatible files, training clients and installing hardware. User training is crucial for minimizing resistance to change and giving chance to prove its worth. Training aids user friendly manuals and healthy screens provide the user with a good start. Software maintenance follows conversion to the extent that changes are necessary to maintain satisfactory operations relative to changes in the user's environment. Maintenance often includes minor enhancements or corrections to the problem that surface late in the systems operations. In the implementation phase, the team builds the components either from scratch or by composition. Given the architecture document meant from the design phase and the requirement document from the analysis phase, the team should build exactly what has been requested, though there is still room for innovation and flexibility. For example, a component may be narrowly designed for this particular system, or the component may be made more general to satisfy a reusability

- Careful planning
- Investigation of system and constraints
- Design the methods to achieve changeover.
- Training the staff in the changed phase.
- Evaluation of change over method.
- The method of implementation and time scale to be adopted are found out initially.

7.2 Maintenance

This phase occurs as a result of deploying the whole system at the end users organization. They will perform the beta testing at the end users and inform to the developers about any needed modification to the application. The customer records all the problems that are encountered during the beta testing and reports these to the developer at regular intervals.

7.2.1 Corrective Maintenance

Even with the best quality assurance activities, it is likely that the customer will uncover defects in the software. Corrective maintenance changes the software to correct the defects. Corrective Maintenance activity may consist of repair,

restoration or replacement of equipment. This activity will be result of a regular inspection, which identifies the failure in time for corrective maintenance to be planned and scheduled, then performed during a routine maintenance shutdown.

7.2.2 Adaptive Maintenance

Over time, the original environment(CPU, operating system, business rules, external product characteristics) for which the software was developed is likely to change. Adaptive maintenance results in modification to the software to accommodate changes to its external environment.

7.2.3 Enhanced Maintenance

As software is used, the parent/user will recognize additional functions that will provide the benefit. Perfect maintenance extends the software beyond its original functional requirements.

7.2.4 Preventive Maintenance

Computer software deteriorates due to change, and because of this preventive maintenance often called software re-engineering, must be conducted to enable the software to serve the needs of its end users. Preventive maintenance makes changes to computer programs so that they can be more easily corrected, adapted and enhanced.

Chapter 8

8 Conclusion and Future Scope

8.1 Conclusion

In conclusion, the baby vaccination app represents a pivotal tool in the realm of child healthcare, offering a streamlined and technologically advanced solution for parents and healthcare providers. As the app continues to evolve, its future scope extends beyond the confines of vaccination tracking, encompassing personalized health records, AI-driven predictive analysis, telemedicine integration and etc. By fostering a sense of community, providing real-time health monitoring, and adhering to robust security measures, the app stands poised to contribute significantly to global immunization efforts and enhance the overall well-being of children. As we navigate the ever-changing landscape of healthcare technology, the baby vaccination app serves as a beacon of innovation and empowerment, bridging the gap between parents and healthcare, ultimately ensuring a healthier and brighter future for the youngest members of our society.

8.2 Future Scope

Here are some potential future developments:

- Personalized Health Records:

The app could expand to include a broader range of health information, allowing parents to maintain a comprehensive digital health record for their child. This could include growth charts, developmental milestones, and other relevant healthcare data.

- AI Integration for Predictive Analysis:

Incorporating artificial intelligence (AI) could enable predictive analysis based on health data, helping parents and healthcare providers anticipate potential health issues and take preventive measures.

- Telemedicine Integration:

Facilitating telemedicine consultations within the app could allow parents to connect with healthcare professionals for advice or consultations without physically visiting a clinic, especially in remote or underserved areas.

- Global Immunization Tracking:

Collaborating with healthcare organizations and governments to contribute to global efforts in tracking and managing vaccination coverage, aiding in the prevention of the spread of vaccine-preventable diseases.

Appendix

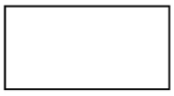
A Data Flow Diagram

Data flow is the one of the best way of documenting the entire functionality of the system.For the system ,which will have data flows in and have some processing inside and then some data flow out from the system can be documented or represented effectively by means of data flow out from the system can be documented or represented effectively by means of data flow diagrams. The data flow diagram are a diagrammatic representation of the system,which has input,process and output.Once any system is represented using a data flow diagram we can identify the following things easily:

- Various entities interacting with the system are identified
- Flow of data from one entity to another is identified
- The various processes involved in between the interaction of two or more entities in the system are clearly pointed out
- The various data stores which hold the data in between the process,are clearly identified

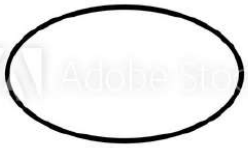
Some Data Flow Diagram charting forms:

A.1 External source or receiver



A source or sink is a person or part of organization ,which enters or receives information from the system,but is considered to be outlining the contest of data flow model.

A.2 Transform process

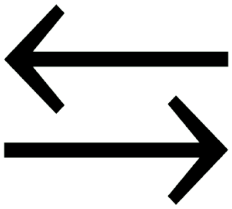


A process represents transformation where incoming data flowsare changed into outgoing data flow

A.3 Data Store

A data store is repository of data that is to be stored for use by one or more process may be as simple as buffer or queue or sophisticated as relational database. They should have clear names.If a process merely uses the contest of store and does not alter it,the arrowhead goes only form the store to the process. If a process alters the details in the store then double-headed arrow is used.

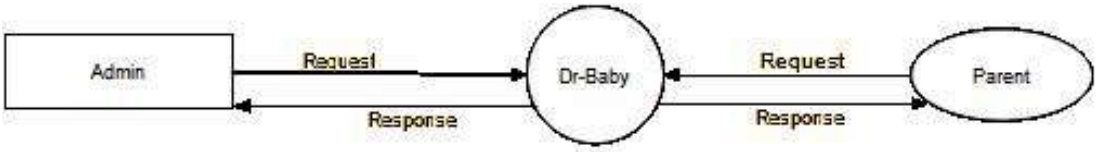
A.4 Data flow



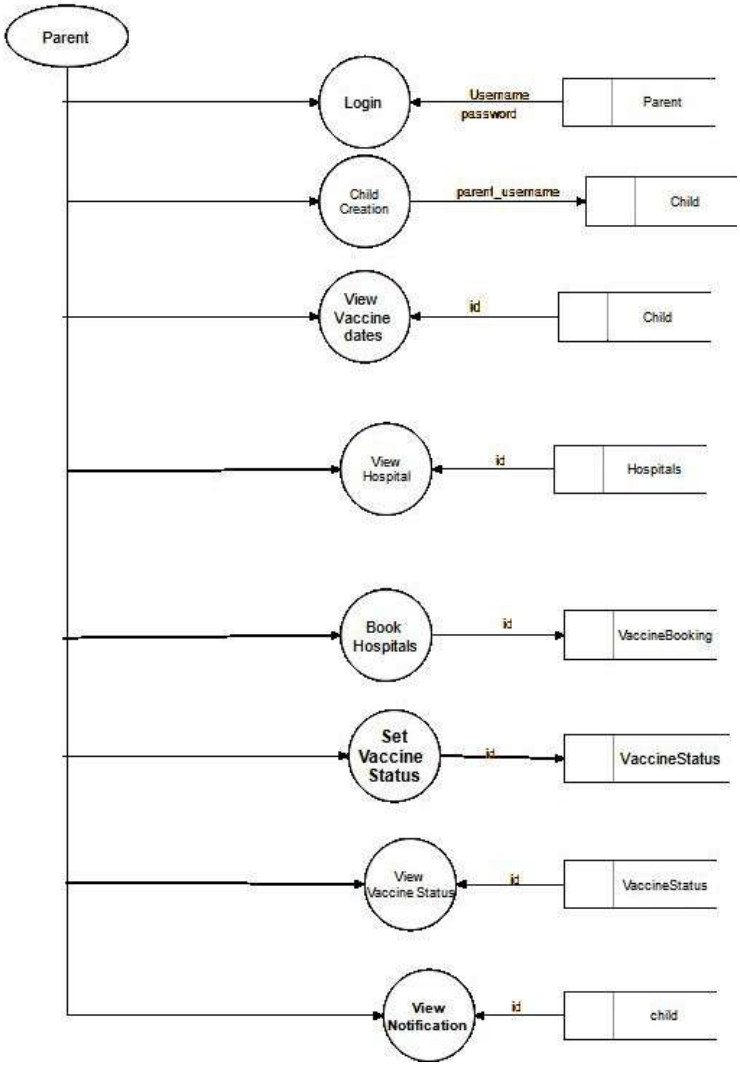
A data flow is a route, which enable packets of data to travel from one point to another.Data may flow,with arrowhead pointing in the direction of the flow

B Data Flow Diagrams

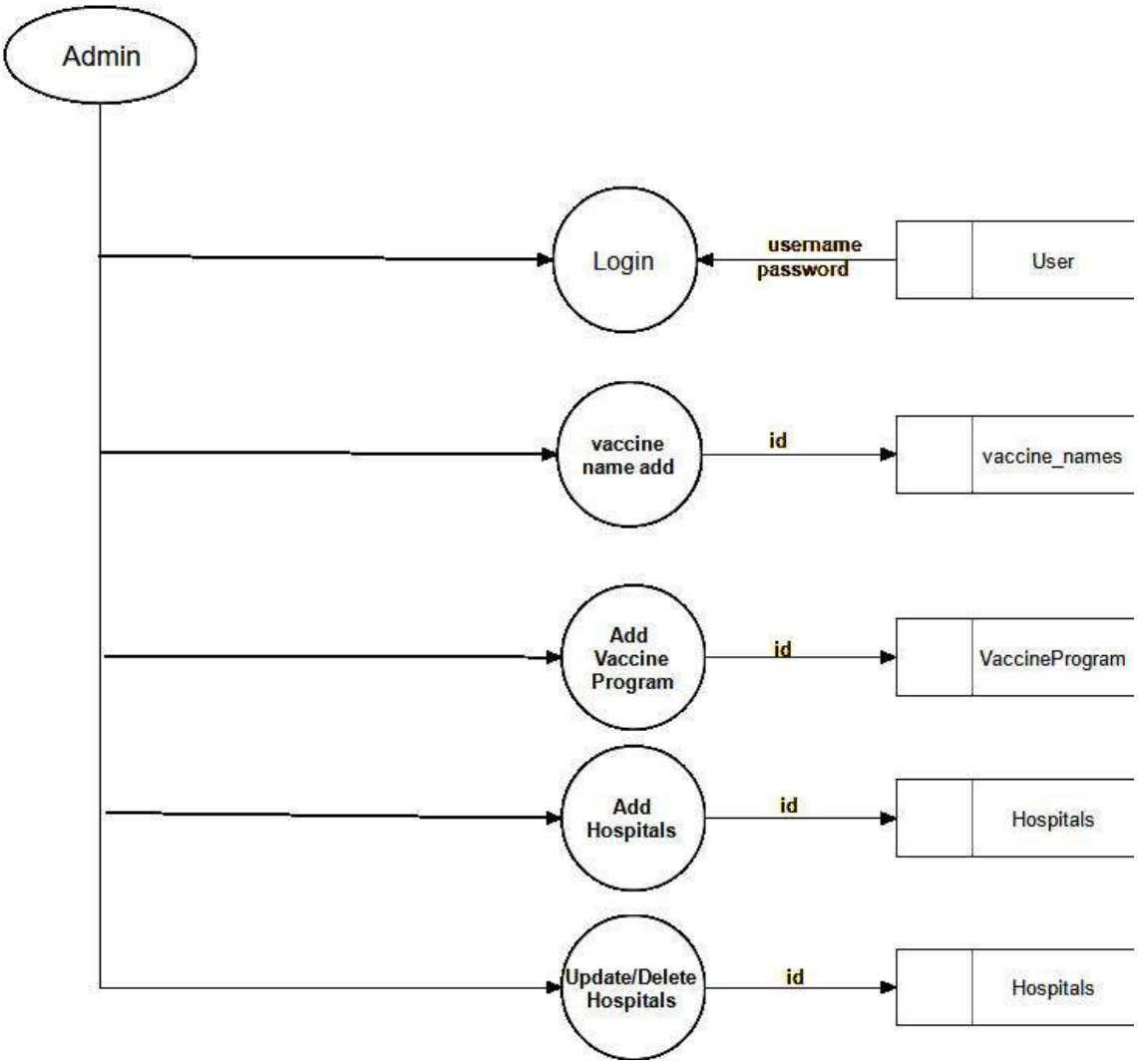
B.1 Level 0



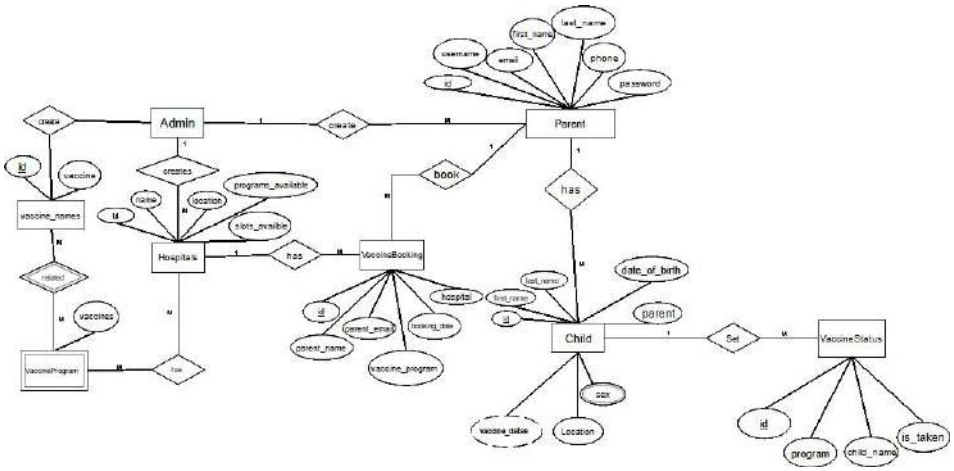
B.2 Level 1



B.3 Level 2

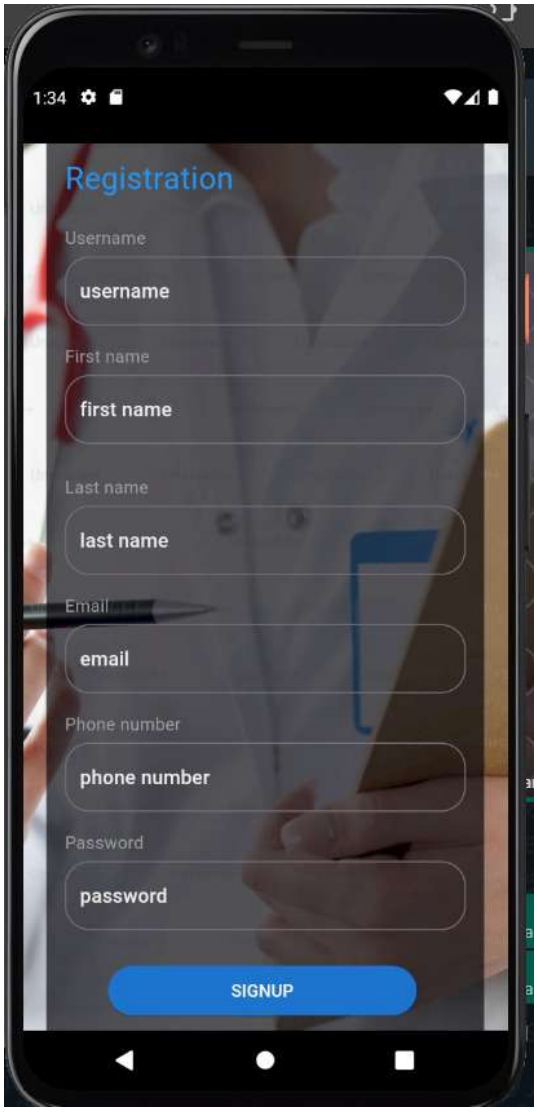

Admin
Login
username
password
User
vaccine
name add
id
vaccine_names
Add
Vaccine
Program
id
VaccineProgram
Add
Hospitals
id
Hospitals
Update/Delete
Hospitals
id
Hospitals

B.4 E-R Diagram

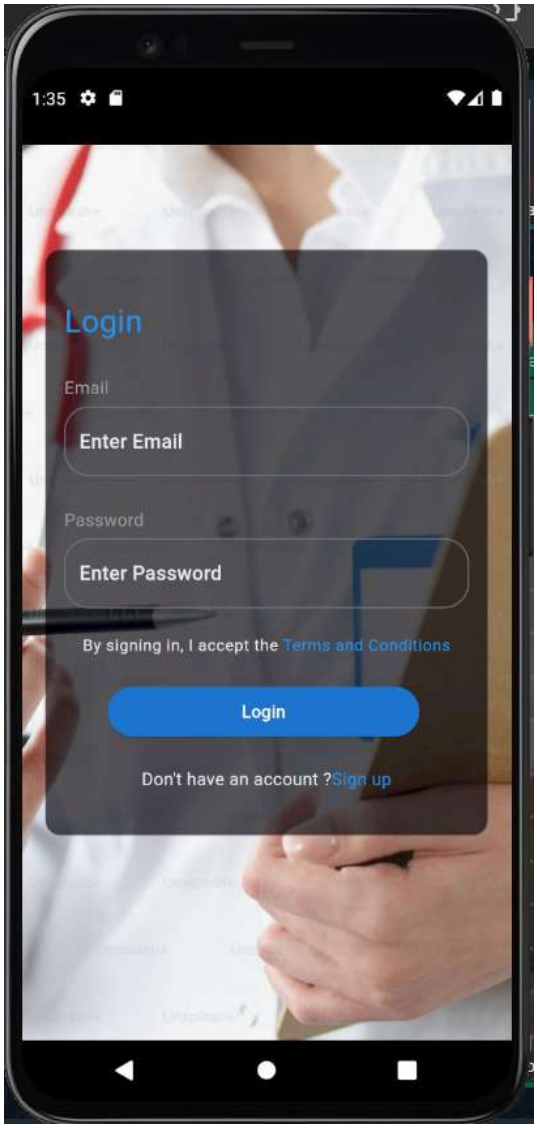


C Interface

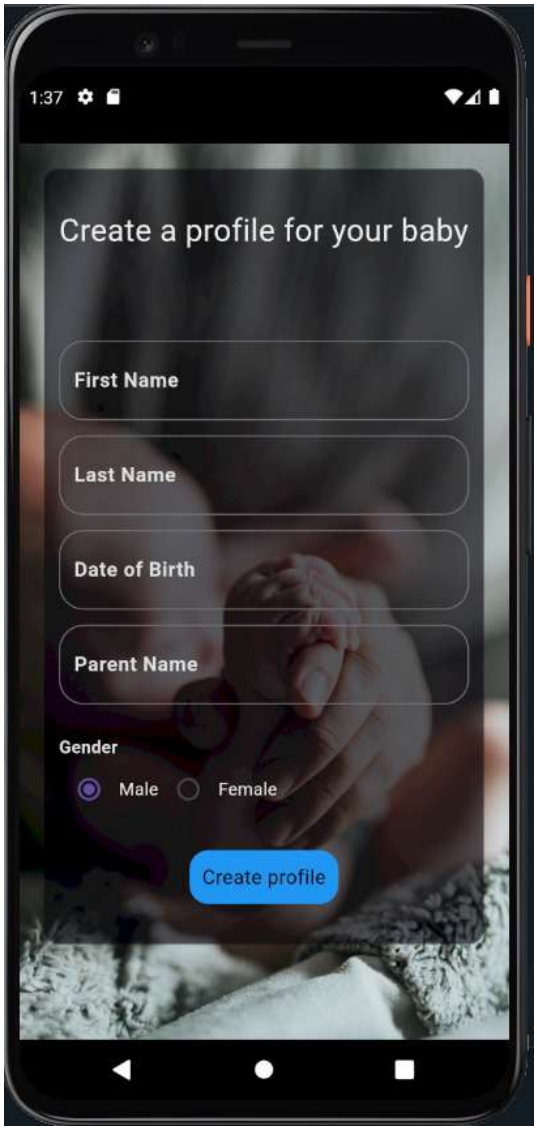
C.1 Registration



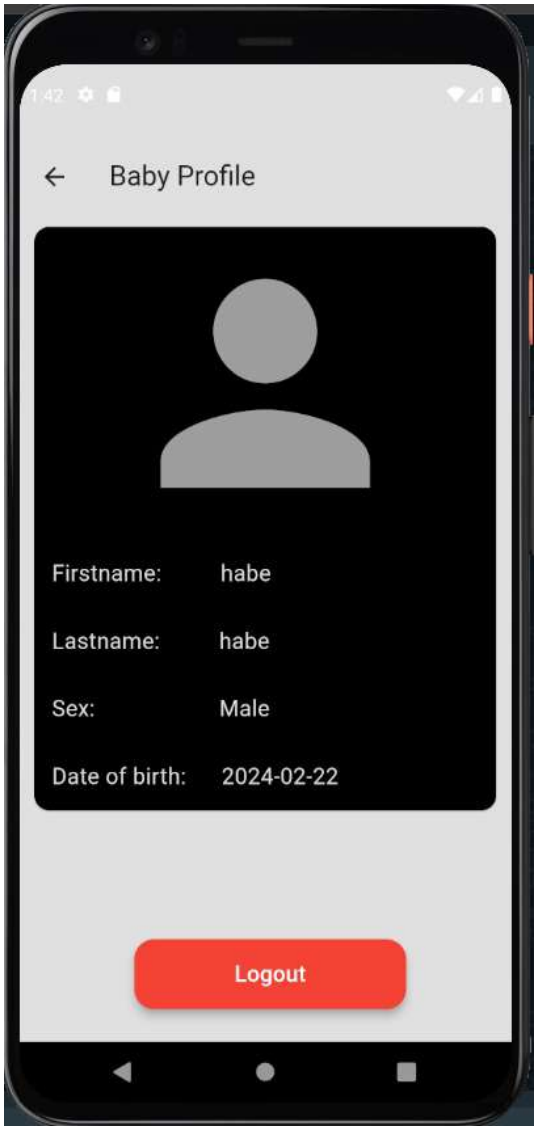
C.2 Login


1:35
Login
Email
Enter Email
Password
Enter Password
By signing in, I accept the Terms and Conditions
Login
Don't have an account ?Sign up

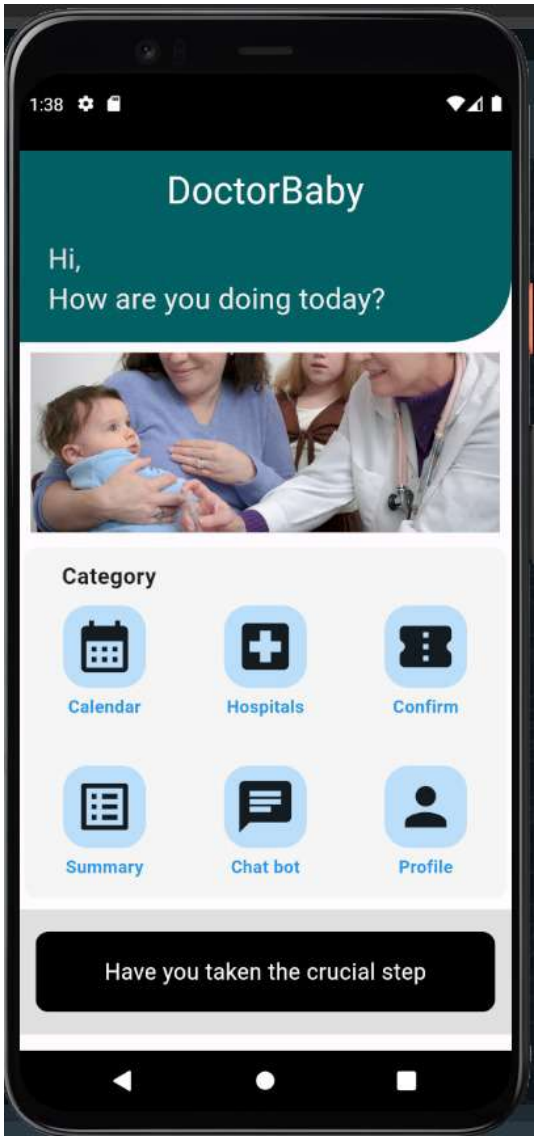
C.3 Profile


1:37
Create a profile for your baby
First Name
Last Name
Date of Birth
Parent Name
Gender
Male
Female
Create profile

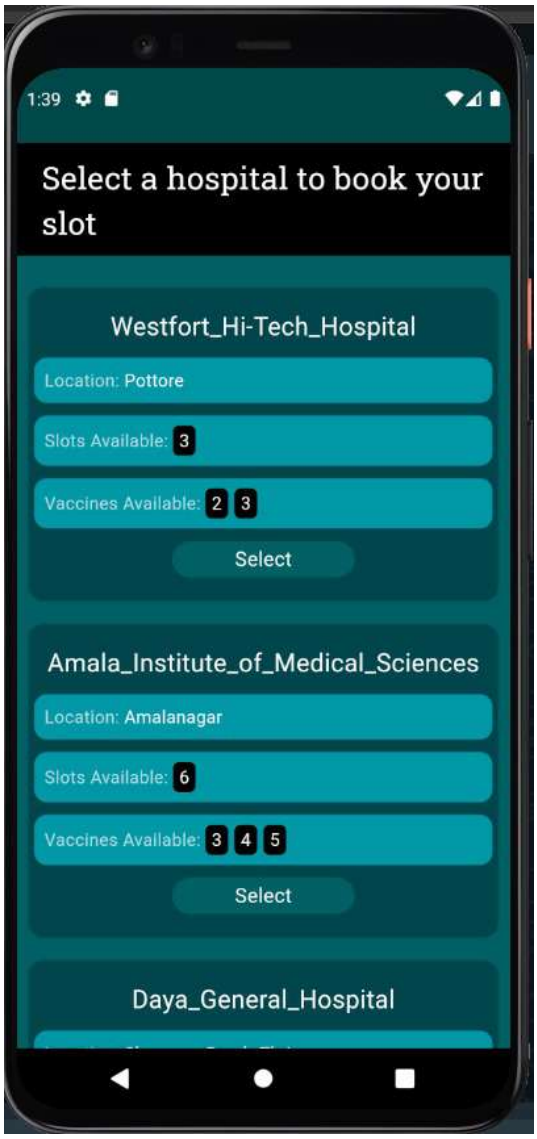
C.4 User Profile



C.5 Home


1:38
DoctorBaby
Hi,
How are you doing today?
Category
Calendar
Hospitals
Confirm
Summary
Chat bot
Profile
Have you taken the crucial step

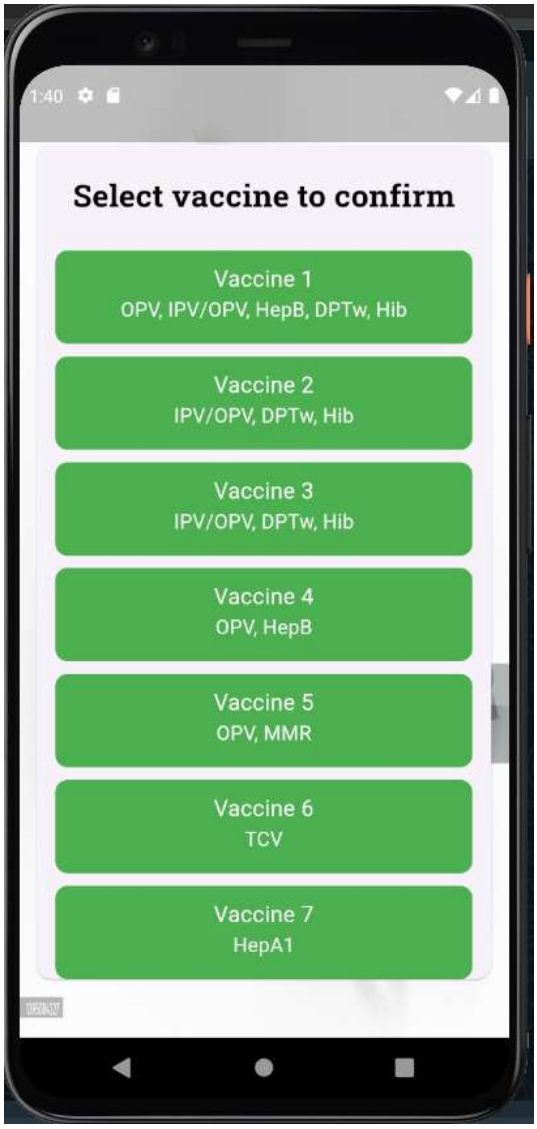
C.6 Hospital


1:39
Select a hospital to book your slot
Westfort_Hi-Tech_Hospital
Location: Pottore
Slots Available: 3
Vaccines Available: 2 3
Select
Amala_Institute_of_Medical_Sciences
Location: Amalanagar
Slots Available: 6
Vaccines Available: 3 4 5
Select
Daya_General_Hospital

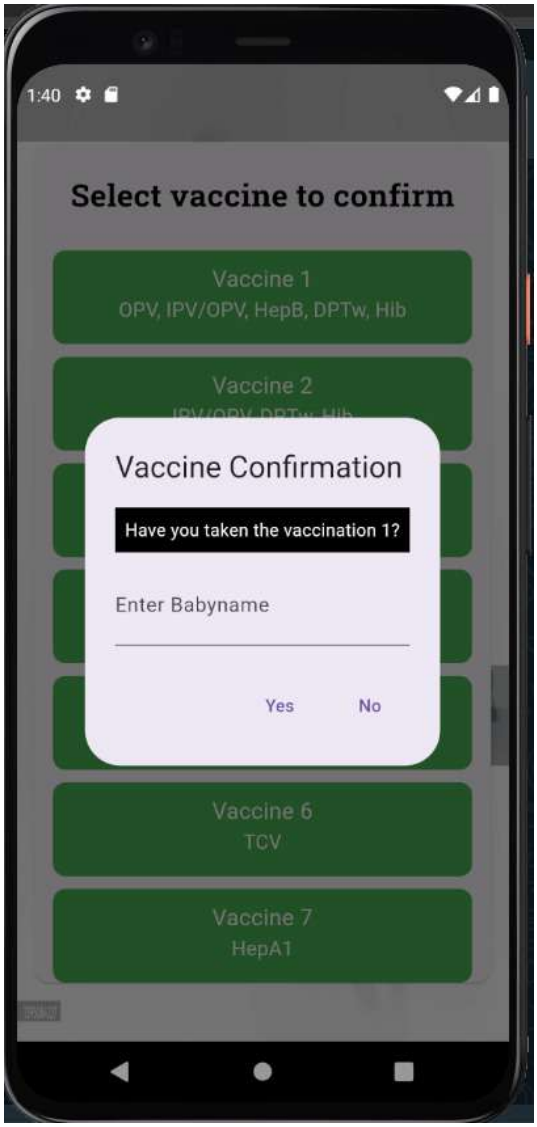
C.7 Calendar


1:38
Calendar
February 2024
2 weeks
Sun Mon Tue Wed Thu Fri Sat
28 29 30 31 1 2 3
4 5 6 7 8 9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 1 2
Vaccine Dates:
2/4/2024
29/4/2024
2/5/2024
21/5/2024

C.8 Vaccine Confirm

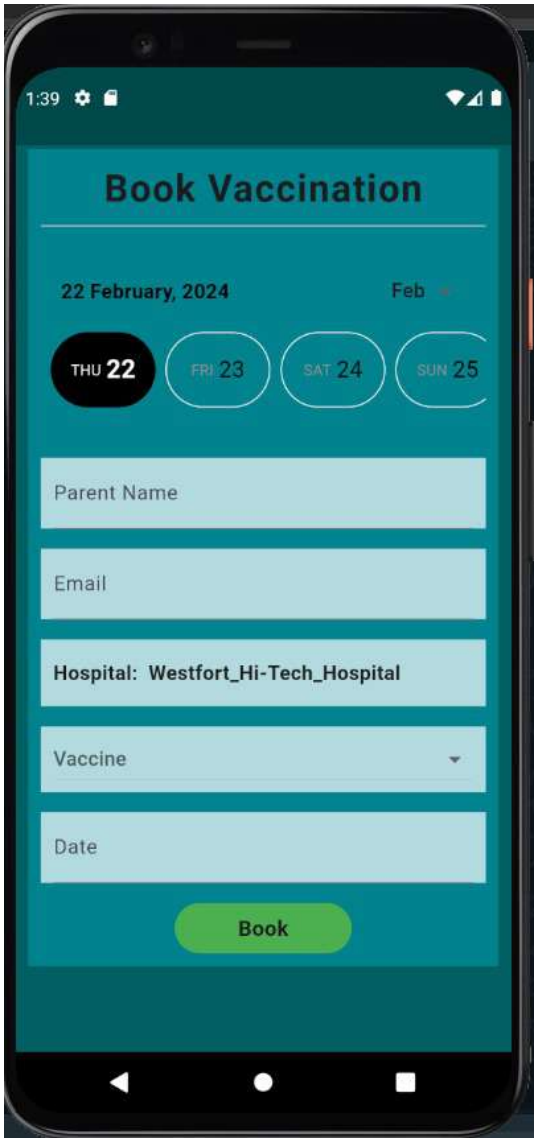


C.9 Confirmation



C.10 Chatbot

C.11 Vaccine Booking



D CODE

tasks.py

```
from celery import shared_task
from django.contrib.auth.models import User
from django.utils import timezone
import datetime
from datetime import timedelta
from django.core.mail import send_mail
from babyvaccinepro import settings
from .models import *

from datetime import datetime, timedelta
from django.utils import timezone




@shared_task(bind=True)
def send_mail_based_on_dates(self):
    recently_registered_user = User.objects.order_by
    ('-date_joined').first()

    if recently_registered_user:
        recent_user_email = recently_registered_user.email
        children = Child.objects.filter
        (parent=recently_registered_user)

        for child in children:
            # Map health review dates to corresponding program
               IDs based on child's date of birth
            health_review_program_mapping = {
                (child.date_of_birth + timedelta(days=40)): [1],
                (child.date_of_birth + timedelta(days=67)): [2],
                (child.date_of_birth + timedelta(days=70)): [3],
                (child.date_of_birth + timedelta(days=89)): [4],
                (child.date_of_birth + timedelta(days=180)): [5],
                (child.date_of_birth + timedelta(days=304)): [6],
                (child.date_of_birth + timedelta(days=363)): [7],

            }

            for rev_date, program_ids in health_review_program
   _mapping.items():
```

```
                if rev_date == timezone.localtime(timezone.
  now()).date():
                    user = child.parent
                    mail_subject = "Health Review Date Reminder"

                    # Filter programs based on the list of
      program IDs
                    programs = VaccinePrograms.objects.filter(
  id__in=program_ids)

                    # Construct the message with vaccines and
                        their names
                    message = f"Hi {user.username}, This is
   a reminder for a health review appointment today
      for {child.first_name}.\n"

                    if programs.exists():
                        for program in programs:
                            message += f"\nVaccines Are\n"

    # Use all() to get all related vaccines
                            for vaccine_info in program.vaccines.all():
                                        vaccine_name = vaccine_
    info.vaccine
                                        message += f"{vaccine_name}\n"

                    else:
                        message += "No relevant vaccination
       programs found for this appointment."

                    to_email = recent_user_email
                    send_mail(
                        subject=mail_subject,
                        message=message,
                        from_email=settings.EMAIL_HOST_USER,
                        recipient_list=[to_email],
                        fail_silently=True,
                    )
                    return "done"
```

views.py

```
from django.contrib.auth import authenticate, login, logout
from django.shortcuts import render
from rest_framework.views import APIView
from .serializer import *
from rest_framework.response import Response
```

```
from rest_framework import generics
from django.db import transaction

from rest_framework import status

from rest_framework.authentication import BasicAuthentication,
TokenAuthentication

from rest_framework.authtoken.models import Token
from django.contrib.auth.hashers import check_password
from django.http import HttpResponse

from django.core.mail import send_mail
from babyvaccinepro.settings import EMAIL_HOST_USER

from django.shortcuts import get_object_or_404
from django.http import JsonResponse

from chatterbot import ChatBot
from chatterbot.trainers import ListTrainer
from PyPDF2 import PdfReader
from langchain.embeddings.openai import OpenAIEmbeddings
from langchain.text_splitter import CharacterTextSplitter
from langchain.vectorstores import FAISS
from langchain.chains.question_answering import load_qa_chain
from langchain.llms import OpenAI
import os



#register parent
class Registeruser(APIView):
    def post(self, request):
        serializer = UserSerializer(data=request.data)
        if serializer.is_valid():
            user = serializer.save()
            # Create a token for the registered user
            token, created = Token.objects.get_or_create(user=user)

            # Get the email of the registered user
            user_email = user.email

            # Your email sending logic using the user's email
            subject = 'Welcome to Dr.baby'
            message = f'Congratulations,\n' \
```

```
                    f'You have successfully registered with our
            website.\n' \
                    f'username: {user.email}\n' \
                    f'WELCOME'

            send_mail(subject, message, settings.EMAIL_HOST_USER,
   [user_email], fail_silently=False)

            # Assuming token creation or any other
               response data you want to send back
            return Response({'data': serializer.data, 'token':
    token.key}, status=status.HTTP_201_CREATED)

        return Response(serializer.errors, status=status.
        HTTP_400_BAD_REQUEST)


#user login

class LoginView(APIView):
    serializer_class = loginserializer

    def post(self, request):
        serializer = self.serializer_class(data=request.data)

        if serializer.is_valid():
            email = serializer.validated_data['email']
            password = serializer.validated_data['password']
            user = authenticate(request, email=email, password
   =password)
            user = User.objects.get(email=email)

            if user is not None and check_password(password,
   user.password):
                login(request,user)
                return Response({'message': 'Login successful'},
                status=status.HTTP_200_OK)
            else:
                return Response({'message': 'Invalid credentials'},
                status=status.HTTP_401_UNAUTHORIZED)

        return Response(serializer.errors, status=status.
        HTTP_400_BAD_REQUEST)


class logoutview(APIView):
```

```
    def post(self,request):
        logout(request)
        return Response({'msg':'logout successfully'})


#child details get and post
class ChildListCreateView(generics.ListCreateAPIView):
    queryset = Child.objects.all()
    serializer_class = ChildSerializer

#child details delete and update

class ChildDetailView(generics.RetrieveUpdateDestroyAPIView):
    queryset = Child.objects.all()
    serializer_class = ChildSerializer


class VaxNameListCreateView(generics.ListCreateAPIView):
    queryset = VaxName.objects.all()
    serializer_class = VaxNameSerializer

class VaxNameDetailView(generics.RetrieveUpdateDestroyAPIView):
    queryset = VaxName.objects.all()
    serializer_class = VaxNameSerializer
```

```
from .tasks import *
from django_celery_beat.models import PeriodicTask,CrontabSchedule
import json


#celery function to send mail

def send_mail_to_parent(request):
    send_mail_based_on_dates.delay()
    return JsonResponse({"message": "Email sending initiated."},
    status=200)

# class SendMailToParentView(APIView):
#     def post(self, request):
```

```
#         send_mail_based_on_dates.delay()
#         return Response({"message": "Email sending initiated."},
           status=status.HTTP_200_OK)


# class SendMailToParentView(APIView):
#     def get(self, request):
#         send_mail_based_on_dates.delay()
#         return Response({"message": "Email sending initiated."}
           , status=status.HTTP_200_OK)
```

```
class VaxCycleAPIView(generics.ListCreateAPIView):
   queryset = Vax_Cycle.objects.all()
   serializer_class = VaxCycleSerializer


class VaxCycleDelete_Update(generics.RetrieveUpdateDestroyAPIView):
    queryset = Vax_Cycle.objects.all()
    serializer_class = VaxCycleSerializer

class VaxAPIView(generics.ListCreateAPIView):
   queryset=Vax.objects.all()
   serializer_class=VaxSerializer

class VaxDelete_Update(generics.RetrieveUpdateDestroyAPIView):
   queryset=Vax.objects.all()
   serializer_class=VaxSerializer
```

```
#vaccine dates


def vaccination_dates_view(request, child_id):
    child = get_object_or_404(Child, pk=child_id)
    vaccination_dates = child.get_vaccination_dates()
```

```
    # Example: Convert vaccination dates to strings for
       JSON response
    vaccination_dates_str = [str(date) for date in vaccination
    _dates]

    return JsonResponse({'vaccination_dates': vaccination_
    dates_str})



class ChatbotAPI(APIView):
    def __init__(self):
        super().__init__()
        # Read PDF and initialize necessary components
        os.environ["OPENAI_API_KEY"] = "sk-y5f7syPdWNeTUUSrDZteT
        3BlbkFJW1Xes6rz21Zmo0hwDxxl"
        pdfreader = PdfReader(r"C:\flutter\backend2\dr-babyvaccine
        \Dr.baby.pdf")
        raw_text = ''
        for page in pdfreader.pages:
            content = page.extract_text()

            if content:
                raw_text += content

        text_splitter = CharacterTextSplitter(
            separator="\n",
            chunk_size=800,
            chunk_overlap=200,
            length_function=len,
        )
        texts = text_splitter.split_text(raw_text)

        embeddings = OpenAIEmbeddings()
        self.document_search = FAISS.from_texts(texts, embeddings)

        self.chain = load_qa_chain(OpenAI(), chain_type="stuff")

    def post(self, request):
        user_input = request.data.get('user_input')

        if user_input.lower() == 'exit':
            return Response({"response": "Goodbye!"},
            status=status.HTTP_200_OK)
```

```
        # Your existing response logic
        bot_response = self.get_response(user_input)
        return Response({"response": bot_response}, status=
        status.HTTP_200_OK)

    def get_response(self, user_input):
        if user_input.lower() in ["hi", "hello", "hey", "hy", "hai"]:
            return "Hello, welcome to Dr Baby. How can I
            assist you today!"
        elif user_input.lower() in ["bye", "by", "thank you",
      "thanks"]:
            return "bye"
        else:
            docs = self.document_search.similarity_search
           (user_input)
            return self.chain.run(input_documents=docs, question
   =user_input)


#vaccine names

class VaccineListView(APIView):
    def post(self,request):
        a=VaccineNameSerializer(data=request.data)
        if a.is_valid():
            a.save()
        return Response(a.data)
    def get(self,request):
        qs=vaccine_names.objects.all()
        a=VaccineNameSerializer(qs,many=True)
        return Response(a.data)



class VaccineProgramsListCreateView(generics.ListCreateAPIView):
    queryset = VaccinePrograms.objects.all()
    serializer_class = VaccineProgramSerializer

class VaccineProgramsDetailView(generics.Retrieve
UpdateDestroyAPIView):
    queryset = VaccinePrograms.objects.all()
    serializer_class = VaccineProgramSerializer
```

```
class HospitalsAPIView(APIView):
    def get(self, request, *args, **kwargs):
        # Filter hospitals with available slots
        hospitals = Hospitals.objects.filter(slots_available__gt=0)
        serializer = HospitalsSerializer(hospitals, many=True)
        return Response(serializer.data)

    def post(self, request, *args, **kwargs):
        serializer = HospitalsSerializer(data=request.data)

        if serializer.is_valid():
            # Save Hospitals instance
            hospital_instance = serializer.save()

            # If vaccine_names data is present, add it to the
               hospital_instance
            vaccine_names_data = request.data.get('programs_
            available', [])
            if vaccine_names_data:
                # Retrieve existing VaccinePrograms instances based
                   on provided IDs
                programs_instances = VaccinePrograms.objects.filter
                (id__in=vaccine_names_data)

                # Add the existing VaccinePrograms instances to
                   hospital_instance
                hospital_instance.programs_available.set(
                programs_instances)

                # Get the vaccine names in the response
                vaccine_names_response = [vaccine.vaccine for
                program in programs_instances for vaccine in
                program.vaccines.all()]
                serializer.data['vaccine_names'] = vaccine_names_
                response

            return Response(serializer.data, status=status.
            HTTP_201_CREATED)

        return Response(serializer.errors, status=status.
        HTTP_400_BAD_REQUEST)
```

```
class VaccineBookingList(APIView):
    def get(self, request):
        bookings = VaccineBooking.objects.all()
        serializer = VaccineBookingSerializer(bookings, many=True)
        return Response(serializer.data)

    def post(self, request):
        serializer = VaccineBookingSerializer(data=request.data)

        if serializer.is_valid():
            hospital_id = request.data.get('hospital')
            program_id = request.data.get('vaccine_program')

            try:
                hospital = Hospitals.objects.get(pk=hospital_id)
                program = VaccinePrograms.objects.get(pk=program_id)
            except (Hospitals.DoesNotExist, VaccinePrograms.
   DoesNotExist):
                return Response({'message': 'Hospital or vaccine
                program not found'}, status=status.
        HTTP_404_NOT_FOUND)

            if hospital.slots_available > 0 and program in
            hospital.programs_available.all():
                with transaction.atomic():
                    hospital.slots_available -= 1
                    hospital.save()

                    serializer.validated_data['hospital'] = hospital
                    serializer.validated_data['vaccine_program']
  = program
                    vaccine_booking = serializer.save()

                    send_booking_confirmation_mail(vaccine_
                     booking.parent_email, vaccine_booking.
            hospital.name,
                     vaccine_booking.vaccine_
                     program.id)

                return Response({'message': 'Booking successful'},
                status=status.HTTP_201_CREATED)
            else:
                return Response({'message':
                'Invalid booking request'},
                status=status.HTTP_400_BAD_REQUEST)
```

```
        return Response(serializer.errors, status=
        status.HTTP_400_BAD_REQUEST)

def send_booking_confirmation_mail(parent_email, hospital_name,
 vaccine_program_id):
    vaccine_program = VaccinePrograms.objects.get(
    pk=vaccine_program_id)

    subject = 'Vaccine Booking Confirmation'
    message = f'Thank you for booking the vaccine program at
    {hospital_name}!\n'
    message += 'Vaccines included:\n'
    for vaccine in vaccine_program.vaccines.all():
        message += f'- {vaccine}\n'

    from_email = 'amrithababy142@gmail.com'
    # Replace with your email
    recipient_list = [parent_email]

    send_mail(subject, message, from_email, recipient_list,
    fail_silently=False)

    return Response({'message': 'Booking successful'},
   status=status.HTTP_201_CREATED)




from django.http import Http404

class VaccineProgramsAPI(APIView):
    def get_object(self, pk):
        try:
            return VaccinePrograms.objects.get(pk=pk)
        except VaccinePrograms.DoesNotExist:
            raise Http404

    def get(self, request, pk=None, format=None):
        if pk:
            # Get a specific vaccine program and its status
            vaccine_program = self.get_object(pk)
            statuses = VaccineStatus.objects.filter(
             program=vaccine_program)

            data = []
```

```
            for status in statuses:
                serializer_status = VaccineStatusSerializer(status)
                program_data = {
                    'program': VaccineProgramSerializer
                    (vaccine_program).data,
                    'statuses': [
                        {
                            'status': {
                                'id': status.id,
                                'program': status.program.id,
                                'child_name': status.
                                 child_name.first_name,
                                'is_taken': status.is_taken
                            }
                        }
                    ]
                }
                data.append(program_data)

            if not data:
                return Response({'is_taken': False})

            return Response(data)
        else:
            # Get status for all vaccine programs
            vaccine_programs = VaccinePrograms.objects.all()
            data = []

            for program in vaccine_programs:
                statuses = VaccineStatus.objects.filter(program
                =program)
                program_data = {
                    'program': VaccineProgramSerializer(program)
                    .data,
                    'statuses': []
                }

                for status in statuses:
                    status_data = {
                        'status': {
                            'id': status.id,
                            'program': status.program.id,
                            'child_name': status.child_name.
                            first_name,
                            'is_taken': status.is_taken
                        }
```

```
                }
                program_data['statuses'].append(status_data)

            data.append(program_data)

        return Response(data)

def post(self, request, pk=None, format=None):
    if pk:
        # Update the status of a specific vaccine program
        vaccine_program = self.get_object(pk)
        child_name = request.data.get('child_name')
      # Retrieve the child name from the request data

        try:
            child = Child.objects.get(first_name=child_name)
         # Retrieve the Child instance using the name
        except Child.DoesNotExist:
            return Response({'error': f'Child with name
            {child_name} not found'}, status=status
            .HTTP_404_NOT_FOUND)

        status, created = VaccineStatus.objects.get_or_create
        (program=vaccine_program, child_name=child)

        # Use 'is_taken' as the field name, and ensure
            it's a boolean
        is_taken = request.data.get('is_taken', False)
        is_taken = is_taken.lower() == 'true'
        # Convert to boolean if needed

        status.is_taken = is_taken
        status.save()

        # Check if the vaccine is taken and return
           the appropriate status
        result_status = 'Taken' if status.is_taken else
        'Pending'

        return Response({'status': result_status})
    else:
        # Create a new vaccine status with child details
        return Response({'error':'Invalid request'})
```

```
class HospitalDetailAPIView(APIView):
    def get_object(self, pk):
        try:
            return Hospitals.objects.get(pk=pk)
        except Hospitals.DoesNotExist:
            raise Http404

    def get(self, request, pk, format=None):
        hospital = self.get_object(pk)
        serializer = HospitalsSerializer(hospital)
        return Response(serializer.data)

    def put(self, request, pk, format=None):
        hospital = self.get_object(pk)
        serializer = HospitalsSerializer(hospital,
        data=request.data)
        if serializer.is_valid():
            serializer.save()
            return Response(serializer.data)
        return Response(serializer.errors, status=status.
        HTTP_400_BAD_REQUEST)

    def delete(self, request, pk, format=None):
        hospital = self.get_object(pk)
        hospital.delete()
        return Response(status=status.HTTP_204_NO_CONTENT)



class ChildVaccineStatusAPI(APIView):
    def get(self, request, child_name, format=None):
        try:
            child = Child.objects.get(first_name=child_name)
        except Child.DoesNotExist:
            return Response({'error': f'Child with name
           {child_name} not found'}, status=status.HTTP_404_NOT_FOUND)

        # Get all VaccineStatus entries for the given child
        statuses = VaccineStatus.objects.filter(child_name=child)

        # Get all available VaccinePrograms
        all_programs = VaccinePrograms.objects.all()

        data = []

        for program in all_programs:
```

```
            # Check if the child has a status for this program
            status_for_program = statuses.filter(program=program)
            .first()

            # If the child has a status, include it in the response
            if status_for_program:
                program_data = {
                    # 'program': VaccineProgramSerializer(status
                    _for_program.program).data,
                    'status': {
                        'id': status_for_program.id,
                        'program': status_for_program.program.id,
                        'child_name': status_for_program.
                child_name.first_name,
                        'is_taken': status_for_program.is_taken
                    }
                }
            else:
                # If the child doesn't have a status, create a
                   placeholder entry with 'is_taken' set to False
                program_data = {
                    # 'program': VaccineProgramSerializer
                        (program).data,
                    'status': {
                        'id': None,
                        'program': program.id,
                        'child_name': child.first_name,
                        'is_taken': False
                    }
                }

            data.append(program_data)

        return Response(data)
```

dates_controller.dart

```
import 'dart:convert';

import 'package:doctor_baby/model/dates_model.dart';
import 'package:doctor_baby/services/dates_service.dart';
import 'package:get/get.dart';

class DatesController extends GetxController{

  var dates = <DateTime>[].obs;
```

```
  @override
  void onInit() {
    getDates();
    super.onInit();
  }

  void getDates() async{
    try {
      var data = await Httpdates.fetchDates();
      var jsondata = json.decode(data);
      VaccinationDates vaccinationDates = VaccinationDates.from
      Json(jsondata);
      dates.assignAll(vaccinationDates.vaccinationDates ?? []);
      print(dates);
    } catch (e) {
      print(e);
    }
  }
}
```

dates_model.dart

```
import 'dart:convert';

VaccinationDates vaccinationDatesFromJson(String str) =>
VaccinationDates.fromJson(json.decode(str));

String vaccinationDatesToJson(VaccinationDates data) =>
 json.encode(data.toJson());

class VaccinationDates {
    List<DateTime>? vaccinationDates;

    VaccinationDates({
        this.vaccinationDates,
    });

    factory VaccinationDates.fromJson(Map<String, dynamic> json) =>
VaccinationDates(
        vaccinationDates: json["vaccination_dates"] == null ? [] :
List<DateTime>.from(json["vaccination_dates"]!.map((x) =>
 DateTime.parse(x))),
    );

    Map<String, dynamic> toJson() => {
        "vaccination_dates": vaccinationDates == null ? [] :
```

```
List<dynamic>.from(vaccinationDates!.map((x) => "${x.year
.toString().padLeft(4, '0')}
-${x.month.toString().padLeft(2, '0')}-${x.day.toString().
padLeft(2, '0')}")),
    };
}
```

dates_service.dart

```
import 'package:doctor_baby/view/profile.dart';
import 'package:http/http.dart' as https;

class Httpdates{
  static Future<dynamic> fetchDates() async{
    var url = "http://10.0.2.2:8000/babyapp/childcreate/
${ProfilePage.userId}/vaccination-dates/";
    print("URL: $url");
    var response = await https.get(Uri.parse(url));
    if(response.statusCode==200){
      return response.body;
    }else{
      throw Exception();
    }
  }
}
```

calender.dart

```
import 'package:doctor_baby/controller/dates_controller.dart';
import 'package:doctor_baby/view/chat.dart';
import 'package:doctor_baby/view/mail/mail.dart';
import 'package:doctor_baby/view/program_status_view.dart';
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'package:get/get_core/src/get_main.dart';
import 'package:table_calendar/table_calendar.dart';
import 'package:intl/intl.dart';


class VaccineCalendar extends StatefulWidget {

  final DatesController datesController = Get.put(DatesController());

  @override
  _VaccineCalendarState createState() => _VaccineCalendarState();
  }
```

```
class _VaccineCalendarState extends State<VaccineCalendar> {
  CalendarFormat _calendarFormat = CalendarFormat.month;
  DateTime _focusedDay = DateTime.now();
  DateTime? _selectedDay;
  Map<DateTime, List<String>> _events = {};
  DateTime? _selectedListViewDate;


  @override
  void initState() {
    super.initState();

  sendMailToParent();

    datesController.dates.forEach((date) {
      _events[date] = ['Vaccine Date'];
    });

    DateTime birthDate = DateTime.now();
 }

  List<String> _getEventsForDay(DateTime date) {
    return _events[date] ?? [];
  }

  void _showEventDetails(DateTime selectedDay) {
    List<String> events = _events[selectedDay] ?? [];
    if (events.isNotEmpty) {
      showDialog(
        context: context,
        builder: (BuildContext context) {
          return AlertDialog(
            title: Text('Events on ${selectedDay.toString().
            split(' ')[0]}'),
            content: Column(
              crossAxisAlignment: CrossAxisAlignment.start,
              mainAxisSize: MainAxisSize.min,
              children: events.map((event) {
                return Text('- $event');
              }).toList(),
            ),
            actions: <Widget>[
              TextButton(
                onPressed: () {
                  Navigator.of(context).pop();
                },
```

```
              child: Text('Close'),
            ),
          ],
        );
      },
    );
  }
}

final DatesController datesController = Get.put(
DatesController());

@override
Widget build(BuildContext context) {
  return SafeArea(
    child: Scaffold(

      // floatingActionButton: FloatingActionButton(onPressed: (){
      //   Navigator.of(context).push(MaterialPageRoute(builder:
            (context)=>ChatBot()));
      // },child: Icon(Icons.message),),

      backgroundColor: Colors.white,
      appBar: AppBar(
        title: Text("Calendar", style: TextStyle(fontWeight:
        FontWeight.bold),),
      ),
      // drawer: Drawer(
      // child: Column(
      //   children: [
      //     TextButton(onPressed: () => Get.to(Programsview()),
            child: Text("Confirm"))
      //   ],
      // ),
    // )
      body: Expanded(
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            // Container(
            //   alignment: Alignment.center,
            //   child: Text("Doctor baby",
            //   style: TextStyle(
            //     fontSize: 25,
            //     color: Colors.black,
            //     fontWeight: FontWeight.bold
```

```
//   ),)),

TableCalendar(
  firstDay: DateTime.utc(2021, 1, 1),
  lastDay: DateTime.utc(2025, 12, 31),
  focusedDay: _focusedDay,
  calendarFormat: _calendarFormat,
  eventLoader: _getEventsForDay,
  headerStyle: HeaderStyle(
    formatButtonVisible: true,
  formatButtonDecoration: BoxDecoration
  (color: Colors.grey),
  leftChevronIcon: Icon(Icons.chevron_left,
  color: Colors.black),
  rightChevronIcon: Icon(Icons.chevron_right,
  color: Colors.black),
    titleTextStyle: TextStyle(color: Colors.black),
  ),
  calendarStyle: CalendarStyle(
    selectedDecoration: BoxDecoration(
      color: Colors.red,
      shape: BoxShape.circle,
    ),
    todayDecoration: BoxDecoration(
      color: Colors.green,
      shape: BoxShape.circle,
    ),
    defaultTextStyle: TextStyle(color: Colors.black),
  ),
  onFormatChanged: (format) {
    setState(() {
      _calendarFormat = format;
    });
  },
  selectedDayPredicate: (day) {
    return isSameDay(_selectedDay, day);
  },
  onDaySelected: (selectedDay, focusedDay) {
    setState(() {
      _selectedDay = selectedDay;
      _focusedDay = focusedDay;
    });
    _showEventDetails(selectedDay);
  },
  calendarBuilders: CalendarBuilders(
    markerBuilder: (context, date, events) {
```

```
        final markers = <Widget>[];

        if (_events[date] != null && _events[date]!.
        isNotEmpty) {
          markers.add(
            Positioned(
              bottom: 1,
              child: Container(
                height: 6,
                width: 6,
                decoration: BoxDecoration(
                  color: Colors.yellow,
                  shape: BoxShape.circle,
                ),
              ),
            ),
          );
        }

        return Row(children: markers);
      },
    ),
  ),
  // SizedBox(height: 10),
  Padding(
    padding: const EdgeInsets.all(8.0),
    child: Text(
      'Vaccine Dates:',
      style: TextStyle(letterSpacing: 1.2,fontSize:
    20, fontWeight: FontWeight.bold, color: Colors.black),
    ),
  ),
  Expanded(
    child: Container(
      // padding: EdgeInsets.all(8),
      decoration: BoxDecoration(
        color: Colors.grey[400],
        borderRadius: BorderRadius.only(
          topLeft: Radius.circular(25),
          topRight: Radius.circular(25),
        ),
      ),
      child:
      Obx(() =>
      ListView.builder(
        itemCount: datesController.dates.length,
```

```
                    itemBuilder: (context, index) {
                      DateTime date = datesController.dates[index];
                      return Card(
                        margin: EdgeInsets.symmetric(vertical: 5,
                        horizontal: 10),
                        color: Colors.grey[900],
                        child: ListTile(
                          title: Row(mainAxisAlignment:
                          MainAxisAlignment.center,
                            children: [
                              Text(
                                '${date.day}/${date.month}/
                                 ${date.year}',
                                style: TextStyle(
                                  color: Colors.white,
                                  letterSpacing: 1.2,
                                  fontSize: 17
                                ),
                              ),
                            ],
                          ),
                          onTap: () {
                            setState(() {
                              _selectedListViewDate = date;
                              _selectedDay = date;
                              _focusedDay = date;
                            });
                            _showEventDetails(date);
                          },
                        ),
                      );
                    },
                  ),
                ),
              ),)
            ],
          ),
        ),
      ),
    );
  }
}
```

hospital_controller.dart

```
import 'package:doctor_baby/services/hospital_service.dart';
import 'package:get/get.dart';
```

```
class HospitalsController extends GetxController{

  var hospitalsList = [].obs;

  @override
  void onInit() {
    getHospitals();
    super.onInit();
  }

  void getHospitals() async{
    try {
      var datas = await HospitalService.fetchHopitals();
      if(datas!=null){
        hospitalsList.value=datas;
      }
    } catch (e) {
      print(e);
    }
  }

}
```

hospital_model.dart

```
// To parse this JSON data, do
//
//     final hospitals = hospitalsFromJson(jsonString);

import 'dart:convert';

List<Hospitals> hospitalsFromJson(String str) =>
List<Hospitals>.from(json.decode(str).map((x) =>
Hospitals.fromJson(x)));

String hospitalsToJson(List<Hospitals> data) =>
json.encode(List<dynamic>.from(data.map((x) => x.toJson())));

class Hospitals {
    int? id;
    String? name;
    String? location;
    int? slotsAvailable;
    List<int>? programsAvailable;
    List<ProgramsDetail>? programsDetails;
```

```
    Hospitals({
        this.id,
        this.name,
        this.location,
        this.slotsAvailable,
        this.programsAvailable,
        this.programsDetails,
    });

    factory Hospitals.fromJson(Map<String, dynamic> json) =>
    Hospitals(
        id: json["id"],
        name: json["name"],
        location: json["location"],
        slotsAvailable: json["slots_available"],
        programsAvailable: json["programs_available"] == null ?
 [] : List<int>.from(json["programs_available"]!.map((x) => x)),
        programsDetails: json["programs_details"] == null ?
 [] : List<ProgramsDetail>.from(json["programs_details"]!.map((x) =>
ProgramsDetail.fromJson(x))),
    );

    Map<String, dynamic> toJson() => {
        "id": id,
        "name": name,
        "location": location,
        "slots_available": slotsAvailable,
        "programs_available": programsAvailable == null ?
          [] : List<dynamic>.from(programsAvailable!.map((x) => x)),
        "programs_details": programsDetails == null ?
          [] : List<dynamic>.from(programsDetails!.map((x) =>

 x.toJson())),
    };
}

class ProgramsDetail {
    List<Vaccine>? vaccines;

    ProgramsDetail({
        this.vaccines,
    });

    factory ProgramsDetail.fromJson(Map<String, dynamic> json)
    => ProgramsDetail(
        vaccines: json["vaccines"] == null ? [] : List<Vaccine>.
```

```
        from(json["vaccines"]!.map((x) => vaccineValues.map[x]!)),
    );

    Map<String, dynamic> toJson() => {
        "vaccines": vaccines == null ? [] : List<dynamic>
        .from(vaccines!.map((x) => vaccineValues.reverse[x])),
    };
}

enum Vaccine {
    DP_TW,
    HEP_B,
    HIB,
    IPV_OPV,
    MMR,
    OPV,
    TCV
}

final vaccineValues = EnumValues({
    "DPTw": Vaccine.DP_TW,
    "HepB": Vaccine.HEP_B,
    "Hib": Vaccine.HIB,
    "IPV/OPV": Vaccine.IPV_OPV,
    "MMR": Vaccine.MMR,
    "OPV": Vaccine.OPV,
    "TCV": Vaccine.TCV
});

class EnumValues<T> {
    Map<String, T> map;
    late Map<T, String> reverseMap;

    EnumValues(this.map);

    Map<T, String> get reverse {
        reverseMap = map.map((k, v) => MapEntry(v, k));
        return reverseMap;
    }
}
```

hospital_service.dart

```
import 'package:doctor_baby/model/hospitals_model.dart';
import 'package:http/http.dart' as https;

class HospitalService{
```

```
  static Future<dynamic> fetchHopitals()async{
    var response = await https.get(Uri.parse(
    "http://10.0.2.2:8000/babyapp/hospitals/"));
    if(response.statusCode == 200){
      var data = response.body;
      return hospitalsFromJson(data);
    }else{
      throw Exception();
    }
  }
}
```

hospital_view.dart

```
import 'package:doctor_baby/controller/hospital_controller.dart';
import 'package:doctor_baby/model/hospitals_model.dart';
import 'package:doctor_baby/model/program_status_model.dart';
import 'package:doctor_baby/view/booking_screen.dart';
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'package:google_fonts/google_fonts.dart';

void main(){
  runApp(GetMaterialApp(home: Hospitalsview(),));
}

class Hospitalsview extends StatelessWidget {

  final HospitalsController hospitalsController =
 Get.put(HospitalsController());

  Hospitalsview({super.key});

  @override
  Widget build(BuildContext context) {
    return Scaffold(
        backgroundColor: Colors.cyan[900],
      body: Column(
        children: [

          SizedBox(height: 60),

          Container(padding: EdgeInsets.symmetric(horizontal: 20),
            alignment: Alignment.centerLeft,
            height: 90,
            decoration: BoxDecoration(
              color:Colors.black,
```

```
      ),
    child: Text("Select a hospital to book your slot",
   style: GoogleFonts.robotoSlab(
    fontSize: 25,
    color: Colors.white,
  ),)
  ),

SizedBox(height: 15),

  Expanded(
    child: Obx(() =>
      ListView.builder(padding: EdgeInsets.symmetric(
      horizontal: 5),
        itemCount: hospitalsController.hospitalsList.length,
        itemBuilder: (context, index) {
        Hospitals hospitals = hospitalsController.
       hospitalsList[index];
        return Padding(
          padding: const EdgeInsets.symmetric(vertical: 5),
          child: InkWell(
            onTap: ()=> Get.to(Booking(hospitalname:
            hospitals.name!, hospitalId: hospitals.id!,)),
            child: Card(
              color:Colors.transparent,
              child: Container(
                alignment: Alignment.center,
                padding: EdgeInsets.symmetric(horizontal: 5),
                height: 250,
                child: Column(
                  mainAxisAlignment: MainAxisAlignment.center,
                  children: [
                    Container(
                      alignment: Alignment.center,
                      child: Text(hospitals.name!,
                      style: TextStyle(
                        color: Colors.white,
                        fontSize: 20),)),

    SizedBox(height: 10,),

     Column(
     mainAxisAlignment: MainAxisAlignment.spaceEvenly,
      children: [
      Container(
      decoration: BoxDecoration(
```

```
                color: Colors.cyan[700],
                borderRadius: BorderRadius.circular(10)),
                padding: EdgeInsets.all(8),
                // alignment: Alignment.centerLeft,
                 child: Row(
                  children: [
                  Text("Location: ",
                  style: TextStyle(color: Colors.white70),
                   ),
                  Text(hospitals.location!,
                   style: TextStyle(color: Colors.white),
                     ),
                      ],
                      )
                     ),

               SizedBox(height: 10,),

                Container(
                decoration: BoxDecoration(
                color: Colors.cyan[700],
                borderRadius: BorderRadius.circular(10)),
                padding: EdgeInsets.all(8),
                // alignment: Alignment.centerLeft,
                child: Row(
                children: [
                          Text("Slots Available: ",
                          style: TextStyle(color: Colors.white70),),
                           Container(
                           decoration: BoxDecoration(
                            color: Colors.black,
                             borderRadius: BorderRadius.circular(5)
                             ),
                            padding: EdgeInsets.symmetric(horizontal: 5,
                            vertical: 2),
                            child: Text("${hospitals.slotsAvailable!}",
                            style: TextStyle(color: Colors.white),),
                                        ),
                                      ],
                                    )
                                    ),

                                    SizedBox(height: 10,),

Container(
decoration: BoxDecoration(
```

```
    color: Colors.cyan[700],
    borderRadius: BorderRadius.circular(10),
  ),
  padding: EdgeInsets.all(8),
  child: Row(
    children: [
      Text("Vaccines Available: ", style: TextStyle(
      color: Colors.white70),),
      Row(
        children: hospitals.programsAvailable!.map((program) {
          return Padding(
            padding: EdgeInsets.only(right: 5),
            child: Container(
              padding: EdgeInsets.symmetric(horizontal: 5,
              vertical: 2),
              decoration: BoxDecoration(
                color: Colors.black,
                borderRadius: BorderRadius.circular(5),
              ),
              child: Text(
                "$program",
                style: TextStyle(color:
                Colors.white),
              ),
            ),
          );
        }).toList(),
      ),
    ],
  ),
)
```

```
                          ],
                        ),
                        SizedBox(height: 10,),

                        Container(
                          decoration: BoxDecoration(
                          color: Colors.cyan[900],
                          borderRadius: BorderRadius.circular(20)),
                          padding: EdgeInsets.symmetric(
                          horizontal: 50, vertical: 3),
                          child: Text("Select", style:
                          TextStyle(fontSize: 16,
                          color: Colors.white),),
```

```
                          )
                        ],
                      ),
                    ),
                  ),
                ),
              );
            },),
          ),
        ),
      ],
    ),
  );
}
```

program_status_controller.dart

```
import 'package:doctor_baby/services/hospital_service.dart';
import 'package:doctor_baby/services/program_status_service.dart';
import 'package:get/get.dart';

class ProgramController extends GetxController{

  var programlist = [].obs;

  @override
  void onInit() {
    getprograms();
    super.onInit();
  }

  void getprograms() async{
    try {
      var datas = await ProgramService.fetchPrograms();
      if(datas!=null){
        programlist.value=datas;
      }
    } catch (e) {
      print(e);
    }
  }

}
```

program_status_model.dart

```
// To parse this JSON data, do
```

```
//
//     final programStatus = programStatusFromJson(jsonString);

import 'dart:convert';

List<ProgramStatus> programStatusFromJson(String str) =>
List<ProgramStatus>.from(json.decode(str).map((x) =>

ProgramStatus.fromJson(x)));

String programStatusToJson(List<ProgramStatus> data) =>
 json.encode(List<dynamic>.from(data.map((x) => x.toJson())));

class ProgramStatus {
    Program? program;
    List<StatusElement>? statuses;

    ProgramStatus({
        this.program,
        this.statuses,
    });

    factory ProgramStatus.fromJson(Map<String, dynamic> json) =>
 ProgramStatus(
        program: json["program"] == null ? null : Program.fromJson
        (json["program"]),
        statuses: json["statuses"] == null ? [] :
        List<StatusElement>.
        from(json["statuses"]!.map((x) =>
       StatusElement.fromJson(x))),
    );

    Map<String, dynamic> toJson() => {
        "program": program?.toJson(),
        "statuses": statuses == null ? [] : List<dynamic>.
         from(statuses!.map((x) => x.toJson())),
    };
}

class Program {
    int? id;
    List<Vaccine>? vaccines;

    Program({
        this.id,
        this.vaccines,
```

```
    });

    factory Program.fromJson(Map<String, dynamic> json) => Program(
        id: json["id"],
        vaccines: json["vaccines"] == null ? [] : List<Vaccine>.
        from(json["vaccines"]!.map((x) => Vaccine.fromJson(x))),
    );

    Map<String, dynamic> toJson() => {
        "id": id,
        "vaccines": vaccines == null ? [] :
         List<dynamic>.from(vaccines!.map((x) => x.toJson())),
    };
}

class Vaccine {
    int? id;
    String? vaccine;

    Vaccine({
        this.id,
        this.vaccine,
    });

    factory Vaccine.fromJson(Map<String, dynamic> json) => Vaccine(
        id: json["id"],
        vaccine: json["vaccine"],
    );

    Map<String, dynamic> toJson() => {
        "id": id,
        "vaccine": vaccine,
    };
}

class StatusElement {
    StatusStatus? status;

    StatusElement({
        this.status,
    });

    factory StatusElement.fromJson(Map<String, dynamic> json) =>
    StatusElement(
     status: json["status"] == null ? null : StatusStatus.
    fromJson(json["status"]),
```

```
    );

    Map<String, dynamic> toJson() => {
        "status": status?.toJson(),
    };
}

class StatusStatus {
    int? id;
    int? program;
    String? childName;
    bool? isTaken;

    StatusStatus({
        this.id,
        this.program,
        this.childName,
        this.isTaken,
    });

    factory StatusStatus.fromJson(Map<String, dynamic> json) =>
   StatusStatus(
        id: json["id"],
        program: json["program"],
        childName: json["child_name"],
        isTaken: json["is_taken"],
    );

    Map<String, dynamic> toJson() => {
        "id": id,
        "program": program,
        "child_name": childName,
        "is_taken": isTaken,
    };
}
```

program_status_service.dart

```
import 'package:doctor_baby/model/program_status_model.dart';
import 'package:http/http.dart' as https;

class ProgramService{
  static Future<dynamic> fetchPrograms()async{
    var response = await https.get(Uri.parse(
   "http://10.0.2.2:8000/babyapp/vaccinestatus/"));
    if(response.statusCode == 200){
      var data = response.body;
```

```
      return programStatusFromJson(data);
    }else{
      throw Exception();
    }
  }
}
```

program_status_view.dart

```
import 'dart:convert';

import 'package:doctor_baby/controller/program
%20_status_controller.dart';
import 'package:doctor_baby/model/program_status_model.dart';
import 'package:doctor_baby/view/calendar.dart';
import 'package:doctor_baby/view/home.dart';
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:http/http.dart' as http;

void main(){
  runApp(GetMaterialApp(home: Programsview(),));
}

class VaccineSelectionController extends GetxController {
  RxString selectedVaccineId = "".obs;

  void setSelectedVaccineId(String id) {
    selectedVaccineId.value = id;
  }
}

final VaccineSelectionController vaccineSelectionController =
Get.put(VaccineSelectionController());


class Programsview extends StatelessWidget {

  final ProgramController programController = Get.put(
ProgramController());
  final TextEditingController dropController=
TextEditingController();
  TextEditingController nameController = TextEditingController();

  static String enteredName = "";
```

```
Programsview({super.key});

@override
Widget build(BuildContext context) {
  return Scaffold(
    // appBar: AppBar(
      // backgroundColor: Colors.black,
    body: Container(
      height: double.infinity,
        decoration: BoxDecoration(
          image: DecorationImage(
            image: NetworkImage(
              "https://media.istockphoto.com/id/1395084227/photo/
             covid-19-vaccine.jpg?s=2048x2048&w=is&k=20&c=
             kiPagHTCx5T6itBE_ek4AtZA8QeiXrUIaSt2VuNom1E=",
            ),
            fit: BoxFit.fill,
          ),
        ),
      // decoration: BoxDecoration(image: DecorationImage
      (image: AssetImage("assets/baby.png"))),
      child: SingleChildScrollView(
        child: Column(
          children: [

          SizedBox(height: 100),

            Padding(
              padding: const EdgeInsets.all(10.0),
              child: Card(
                child: Container(alignment: Alignment.center,
                  height: 550,
                  color: Colors.transparent,
                  child: Column(
                    children: [

                Container(
                  padding: EdgeInsets.symmetric(horizontal: 20),
                  alignment: Alignment.center,
                  height: 90,
                  decoration: BoxDecoration(
                    color:Colors.transparent,
                    borderRadius: BorderRadius.only(
                    bottomLeft: Radius.circular(20),
                    bottomRight: Radius.circular(20)
```

```
                    )),
                  child: Text("Select vaccine to confirm",
                 style: GoogleFonts.robotoSlab(
                  fontSize: 25,
                  color: Colors.black,
                  fontWeight: FontWeight.bold
                ),)
                ),

                        Expanded(
                          child: Obx(() =>
                            ListView.builder(padding:
                            EdgeInsets.symmetric(horizontal: 5),
                              itemCount: programController.
                              programlist.length,
                              itemBuilder: (context, index) {
                              ProgramStatus program =
                              programController.programlist[index];
                              return Padding(
                                padding: const EdgeInsets.
                                symmetric(vertical: 5, horizontal: 5),
                                child: InkWell(
                                  onTap: () {
   // vaccineSelectionController.setSelectedVaccineId(program.
program!.id.toString());
  String selectedVaccineId = program.program!.id.toString();
  showDialog(
           context: context,
           builder: (BuildContext context) {
           return Container(
           child: Padding(
            padding: const EdgeInsets.symmetric(vertical: 50),
            child: AlertDialog(
            title: Text("Vaccine Confirmation"),
            content: SingleChildScrollView(
            child: Column(
            children: [
             Container(
             alignment: Alignment.center,
             width: double.infinity,
             padding: EdgeInsets.all(8),
             color: Colors.black,
            child: Text("Have you taken the vaccination
            ${program.program!.id.toString()}?", style:
            TextStyle(color: Colors.white),)),
```

```
SizedBox(height: 10,),

TextField(
controller: nameController,
decoration: InputDecoration(labelText: 'Enter Babyname'),
                                                                                ),
],
),
),
actions: [
TextButton(
onPressed: () async {

enteredName = nameController.text.trim();

String username = nameController.text.trim();

String yes = "True";

await post(selectedVaccineId, username, yes);
// Get.to(VaccineCalendar());
},
child: Text("Yes"),
),
TextButton(
onPressed: () {
Navigator.of(context).pop();
},
child: Text("No"),
),
],
),
),
);
},
);
} ,
child: Padding(
padding: const EdgeInsets.symmetric(horizontal: 20),
child: Container(
decoration: BoxDecoration(
color:Colors.green,
borderRadius: BorderRadius.circular(10)),
child: Container(
alignment: Alignment.center,
padding: EdgeInsets.symmetric(horizontal: 5,vertical: 10),
```

```
height: 50,
child: Row(
mainAxisAlignment: MainAxisAlignment.spaceAround,
children: [
Container(
alignment: Alignment.center,
\child: Text("Vaccine ${program.program!.id.toString()}",
style: TextStyle(
color: Colors.white,
fontSize: 20),)),
                                          ],
                                        ),
                                      ),
                                    ),
                                  ),
                                ),
                              ),
                            );
                          },),
                        ),
                      ),
                    ],
                  ),
                ),
              ),
            ],
          ),
        ),
      ),
    );
  }
}

Future<void> post(String vaccineId, String username,
 String yes) async {
    final url = Uri.parse('http://10.0.2.2:8000/babyapp/
    setvaccinestatus/$vaccineId/');
    final response = await http.post(
      url,
      headers: {'Content-Type': 'application/json'},
      body:
      jsonEncode(
        {
        'is_taken': yes,
        'child_name': username
        }
```

```
        ),
    );

    if (response.statusCode == 200) {
      print('Confirmation successful');
      Get.snackbar("Vaccination", "Confirm");
      Get.to(Home());
    } else {
      print('Error: ${response.statusCode}');
    }
  }
```

summary_controller.dart

```
import 'package:doctor_baby/services/summary_service.dart';
import 'package:doctor_baby/services/vaccine_service.dart';
import 'package:get/get.dart';

class SummaryController extends GetxController{

  var summaryList = [].obs;

  @override
  void onInit() {
    getSummary();
    super.onInit();
  }

  void getSummary() async{
    try {
      var datas = await VaccineSummary.fetchSummary();
      if(datas!=null){
        summaryList.value=datas;
      }
    } catch (e) {
      print(e);
    }
  }

}
```

summary_model.dart

```
// To parse this JSON data, do
//
//     final summary = summaryFromJson(jsonString);
```

```
import 'dart:convert';

List<Summary> summaryFromJson(String str) =>
List<Summary>.from(json.decode(str).map((x) => Summary.fromJson(x)));

String summaryToJson(List<Summary> data) =>
json.encode(List<dynamic>.from(data.map((x) => x.toJson())));

class Summary {
    Status? status;

    Summary({
        this.status,
    });

    factory Summary.fromJson(Map<String, dynamic> json) => Summary(
        status: json["status"] == null ? null :
        Status.fromJson(json["status"]),
    );

    Map<String, dynamic> toJson() => {
        "status": status?.toJson(),
    };
}

class Status {
    int? id;
    int? program;
    String? childName;
    bool? isTaken;

    Status({
        this.id,
        this.program,
        this.childName,
        this.isTaken,
    });

    factory Status.fromJson(Map<String, dynamic> json) => Status(
        id: json["id"],
        program: json["program"],
        childName: json["child_name"],
        isTaken: json["is_taken"],
    );

    Map<String, dynamic> toJson() => {
```

```
        "id": id,
        "program": program,
        "child_name": childName,
        "is_taken": isTaken,
    };
}
```

summary_service.dart

```
import 'package:doctor_baby/model/vaccines_model.dart';
import 'package:doctor_baby/view/program_status_view.dart';
import 'package:http/http.dart' as https;

class VaccineSummary{
  static Future<dynamic> fetchSummary()async{
    var response = await https.get(Uri.parse(
    "http://10.0.2.2:8000/babyapp/child_vaccine_status/nazrin/"));
    if(response.statusCode == 200){
      var data = response.body;
      return vaccinesFromJson(data);
    }else{
      throw Exception();
    }
  }
}
```

booking_summary.dart

```
import 'package:doctor_baby/model/summary_model.dart';
import 'package:doctor_baby/view/profile.dart';
import 'package:doctor_baby/view/program_status_view.dart';
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'package:http/http.dart' as http;


class SummaryScreen extends StatefulWidget {
  @override
  _SummaryScreenState createState() => _SummaryScreenState();
}

class _SummaryScreenState extends State<SummaryScreen> {
  final _summaryController = Get.put(SummaryController());

  @override
  void initState() {
    super.initState();
```

```
    _summaryController.fetchSummaryData();
  }

  @override
  Widget build(BuildContext context) {
    return SafeArea(
      child: Scaffold(
        backgroundColor: Colors.grey,

        body: Padding(
          padding: const EdgeInsets.symmetric(vertical: 25),
          child: Card(color: Colors.black54,
            child: Container(
              padding: EdgeInsets.symmetric(vertical: 10),
              // height: 600,
              child: Column(mainAxisAlignment: MainAxisAlignment.center,
                children: [

                  Text("Vaccine Summary", style: TextStyle(color:
                   Colors.grey[200], fontSize: 25, fontWeight:
                   FontWeight.bold),),
                  SizedBox(height: 20),

                  Expanded(
                    child: Obx(
                      () {
                        if (_summaryController.isLoading.value) {
                          return Center(child:
                          CircularProgressIndicator());
                        } else {
                          return ListView.builder(
                            itemCount: _summaryController.
                            summaryList.length,
                            itemBuilder: (context, index) {
                              var summary = _summaryController.
                              summaryList[index];

                              String isTakenDisplay = summary.
                              status?.isTaken == true ?
                              'Completed' : 'Pending';
                              var isTakenColor =
                            summary.status?.isTaken == true ?
                            Colors.green : Colors.red;

return
Padding(
```

```
padding: const EdgeInsets.symmetric(vertical: 5, horizontal: 10),
child: Container(
decoration: BoxDecoration(
color: Colors.black,
borderRadius: BorderRadius.circular(10)),
padding: EdgeInsets.symmetric(vertical: 12,horizontal: 10),
width: double.infinity,
child: Row(
children: [
Container(
width: 150,
padding: EdgeInsets.all(10),
child: Text('Vaccine ${summary.status?.program}'
,style: TextStyle(fontSize: 20, color: Colors.white)),
),
Expanded(
child: Container(
decoration: BoxDecoration(
color: isTakenColor,
borderRadius: BorderRadius.circular(20)),
alignment: Alignment.center,
padding: EdgeInsets.all(10),
child: Text('${isTakenDisplay}',style: TextStyle(fontSize: 20),)),
                                        )
                                      ],
                                    ),
                                  ),
                                );
ListTile(
title: Text('Vaccine ${summary.status?.program}'),
// subtitle: Text('Is Taken: ${summary.status?.isTaken}'),
subtitle: Text('Taken: ${isTakenDisplay}'),
                                );
                              },
                            );
                          }
                        },
                      ),
                    ),
                  ],
                ),
              ),
            ),
          ),
        ),
      ),
    );
```

```
  }
}

class SummaryController extends GetxController {
  RxBool isLoading = true.obs;
  RxList<Summary> summaryList = <Summary>[].obs;

  Future<void> fetchSummaryData() async {
    try {

      String? firstName = await Util.getFirstName();

      final response = await http.get(Uri.parse(
     'http://10.0.2.2:8000/babyapp/child_vaccine_status/$firstName/'));
      if (response.statusCode == 200) {
        final List<Summary> summaries = summaryFromJson(response.body);
        summaryList.assignAll(summaries);
      } else {
        throw Exception('Failed to load data');
      }
    } catch (e) {
      print('Error: $e');
    } finally {
      isLoading.value = false;
    }
  }
}
```

vaccine_comtroller.dart

```
import 'package:doctor_baby/services/vaccine_service.dart';
import 'package:get/get.dart';

class VaccineController extends GetxController{

  var vaccineList = [].obs;

  @override
  void onInit() {
    getVaccines();
    super.onInit();
  }

  void getVaccines() async{
    try {
      var datas = await VaccineService.fetchVaccine();
      if(datas!=null){
```

```
        vaccineList.value=datas;
      }
    } catch (e) {
      print(e);
    }
  }

}
```

vaccine_model.dart

```
// To parse this JSON data, do
//
//     final vaccines = vaccinesFromJson(jsonString);

import 'dart:convert';

List<Vaccines> vaccinesFromJson(String str) =>
 List<Vaccines>.from(json.decode(str).map((x) => Vaccines.fromJson(x)));

String vaccinesToJson(List<Vaccines> data) =>
 json.encode(List<dynamic>.from(data.map((x) => x.toJson())));

class Vaccines {
    int? id;
    List<Vaccine>? vaccines;

    Vaccines({
        this.id,
        this.vaccines,
    });

    factory Vaccines.fromJson(Map<String, dynamic> json) => Vaccines(
        id: json["id"],
        vaccines: json["vaccines"] == null ? [] : List<Vaccine>.
        from(json["vaccines"]!.map((x) => Vaccine.fromJson(x))),
    );

    Map<String, dynamic> toJson() => {
        "id": id,
        "vaccines": vaccines == null ? [] : List<dynamic>.
         from(vaccines!.map((x) => x.toJson())),
    };
}

class Vaccine {
    int? id;
```

```
    String? vaccine;

    Vaccine({
        this.id,
        this.vaccine,
    });

    factory Vaccine.fromJson(Map<String, dynamic> json) => Vaccine(
        id: json["id"],
        vaccine: json["vaccine"],
    );

    Map<String, dynamic> toJson() => {
        "id": id,
        "vaccine": vaccine,
    };
}
```

vaccine_service.dart

```
import 'package:doctor_baby/model/vaccines_model.dart';
import 'package:http/http.dart' as https;

class VaccineService{
  static Future<dynamic> fetchVaccine()async{
    var response = await https.get(Uri.parse(
    "http://10.0.2.2:8000/babyapp/vaccine_programs/"));
    if(response.statusCode == 200){
      var data = response.body;
      return vaccinesFromJson(data);
    }else{
      throw Exception();
    }
  }
}
```

chat.dart

```
import 'dart:convert';
import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;
import 'package:get/get.dart';

class ChatController extends GetxController {
  RxList<Map<String, String>> chat = <Map<String, String>>[].obs;

  TextEditingController textController = TextEditingController();
```

```
  RxString text = ''.obs;
  RxString prediction = ''.obs;

  Future<void> postData(String text) async {
    var url = "http://10.0.2.2:8000/babyapp/chatbot/";
    chat.add({
      "responeby": "user",
      "text": text,
    });
    try {
      final jsonResponse = await http.post(
        Uri.parse(url),
        body: {'user_input': text.trim()},
      );

      if (jsonResponse.statusCode == 200) {
        Map<String, dynamic> parsedResponse = jsonDecode
        (jsonResponse.body);
        String responseValue = await parsedResponse['response'];
        chat.add({
          "responeby": "bot",
          "text": responseValue,
        });
        print(responseValue);
      } else {
        print("Error: ${jsonResponse.statusCode}");
      }
    } catch (error) {
      print("Error: $error");
    }
  }
}

class ChatBot extends StatelessWidget {
  final ChatController chatController = Get.put(ChatController());

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        centerTitle: true,
        backgroundColor: Colors.cyan[900],
        title: Text(
          "Baby Helper",
          style: TextStyle(fontWeight: FontWeight.bold, color:
          Colors.white),
```

```
        ),
      ),
      body: SingleChildScrollView(
        child: Column(
          children: [
Obx(() => ListView.builder(
shrinkWrap: true,
physics: NeverScrollableScrollPhysics(),
itemCount: chatController.chat.length,
itemBuilder: (context, index) {
String chatText = chatController.chat[index]["text"]!;
return Align(
alignment: chatController.chat[index]["responeby"] == "bot"
? Alignment.centerLeft
: Alignment.centerRight,
child: Container(
child: Padding(
padding: const EdgeInsets.all(8.0),
child: Container(
decoration: BoxDecoration(
color: chatController.chat[index]["responeby"] == "bot"
? Colors.blue
: Colors.grey,
borderRadius: chatController.chat[index]["responeby"] ==
"bot"
? BorderRadius.only(
bottomLeft: Radius.circular(10),
bottomRight: Radius.circular(10),
topRight: Radius.circular(10))
: BorderRadius.only(
bottomLeft: Radius.circular(10),
bottomRight: Radius.circular(10),
topLeft: Radius.circular(10)),
                            ),
                            child: Padding(
                              padding: const EdgeInsets.only(
                                  top: 8, bottom: 8, left: 15,
                                  right: 15),
                              child: Text(
                                chatText,
                                style: TextStyle(
                                    fontSize: 18,
                                    color: Colors.white,
                                    fontWeight: FontWeight.bold),
                              ),
                            ),
```

```
                    ),
                  ),
                ),
              );
            },
          )),
      Padding(
        padding: EdgeInsets.only(
          bottom: MediaQuery.of(context).viewInsets.bottom,
        ),
      ),
    ],
  ),
),
bottomNavigationBar: Padding(
  padding: EdgeInsets.only(
    bottom: MediaQuery.of(context).viewInsets.bottom,
  ),
  child: Container(
    height: 60,
    decoration: BoxDecoration(
      color: Colors.grey[200],
      borderRadius: BorderRadius.circular(20),
    ),
    child: Padding(
      padding: const EdgeInsets.symmetric(horizontal: 20),
      child: Row(
        children: [
          Expanded(
            child: TextFormField(
              controller: chatController.textController,
              onChanged: (value) => chatController.text.
              value = value,
              decoration: InputDecoration(
                border: InputBorder.none,
                hintText: "Enter here ..",
              ),
            ),
          ),
          SizedBox(width: 10),
          InkWell(
            onTap: () {
              chatController.postData(chatController.
              text.value);
              chatController.textController.text = "";
            },
```

```
                  child: CircleAvatar(
                    radius: 18,
                    backgroundColor: Colors.blue,
                    child: Center(child: Icon(Icons.send)),
                  ),
                ),
              ],
            ),
          ),
        ),
      ),
    );
  }
}

void main() {
  runApp(GetMaterialApp(home: ChatBot()));
}
```

home.dart

```
import 'package:doctor_baby/view/booking_summary.dart';
import 'package:doctor_baby/view/calendar.dart';
import 'package:doctor_baby/view/chat.dart';
import 'package:doctor_baby/view/components/carousals.dart';
import 'package:doctor_baby/view/hospitals_view.dart';
import 'package:doctor_baby/view/program_status_view.dart';
import 'package:flutter/material.dart';
import 'package:get/get.dart';

void main(){
  runApp(GetMaterialApp(home: Home(),debugShowCheckedModeBanner:
  false,));
}

class Home extends StatefulWidget {
  const Home({super.key});

  @override
  State<Home> createState() => _HomeState();
}

class _HomeState extends State<Home> {

  var name = [
    "Calendar",
    "Hospitals",
```

```
    "Confirm",
    "Summary",
    "Chat bot",
    "Profile",
  ];

  var icon = [
    Icons.calendar_month,
    Icons.local_hospital,
    Icons.confirmation_num_rounded,
    Icons.list_alt,
    Icons.chat,
    Icons.person,
  ];

  var color = [
    Colors.blue[100],
    Colors.red[300],
    Colors.green[100],
    Colors.pink[100],
    Colors.purple[200],
    Colors.cyan[100],
  ];

  var pages = [
    VaccineCalendar(),
    Hospitalsview(),
    Programsview(),
    SummaryScreen(),
    ChatBot(),
    ChatBot(),
  ];


  @override
  Widget build(BuildContext context) {
    return SafeArea(
      child: Scaffold(
        body: SingleChildScrollView(
          child: Column(
          children: [

            Container(
              decoration: BoxDecoration(
                color: Colors.cyan[900],
                borderRadius: BorderRadius.only(bottomRight:
```

```
               Radius.circular(50))
              ),
              child: Column(
                children: [
                  SizedBox(height: 10,),

                  Container(
                    alignment: Alignment.center,
                    width: double.infinity,
                    child: Text("DoctorBaby", style: TextStyle(

color: Colors.white, fontSize: 30),),
                    ),
            Container(
              width: double.infinity,
              padding: EdgeInsets.symmetric(horizontal: 15,
              vertical: 10),
              height: 100,
              child: Padding(
                padding: const EdgeInsets.all(8.0),
                child: Column(
                  mainAxisAlignment: MainAxisAlignment.center,
                  crossAxisAlignment: CrossAxisAlignment.start,
                  children: [
                    Text("Hi User, ",style: TextStyle(color:
                    Colors.grey[300], fontSize: 22),),
                    Text("How are you doing today?",style:
                    TextStyle(color: Colors.grey[200], fontSize: 23))
                  ],
                ),
              ),
            ),
                ],
              ),
            ),


            Carousal(),

            Padding(
              padding: const EdgeInsets.all(4.0),
              child: Container(
                decoration: BoxDecoration(
                color: Colors.grey[100],
                  borderRadius: BorderRadius.circular(10)),
                height:300,
```

```
  child: Column(
    children: [
      Padding(
        padding: const EdgeInsets.symmetric(
       horizontal: 30, vertical: 10),
        child: Container(
          alignment: Alignment.centerLeft,
          height: 25,
          width: double.infinity,
          child: Text("Category", style: TextStyle(
          fontSize: 18, fontWeight: FontWeight.bold),),
        ),
      ),
      SizedBox(height: 10,),
      Expanded(
child: GridView.builder(
  physics: NeverScrollableScrollPhysics(),
  itemCount: 6,
  gridDelegate:
  SliverGridDelegateWithFixedCrossAxisCount(crossAxisCount: 3),
  itemBuilder: (context, index){
    return Padding(
      padding: const EdgeInsets.all(1.0),
      child: Column(
        children: [
          InkWell(onTap: () => Get.to(pages[index]),
            child: Container(
              decoration: BoxDecoration(
              color: Colors.blue[100],
                borderRadius: BorderRadius.circular(20)
              ),
              padding: EdgeInsets.all(8),
              child: Icon(icon[index], size: 50),
            ),
          ),
          SizedBox(height: 5,),
          Text(name[index], style: TextStyle(
          color: Colors.blue, fontWeight:
          FontWeight.bold),)
        ],
      ),
    );
  }),
      )
    ],
  ),
```

```
  ),
),

Padding(
  padding: const EdgeInsets.symmetric(
  horizontal: 10, vertical: 5),
  child: Container(
    decoration: BoxDecoration(
    color: Colors.grey[300],
      borderRadius: BorderRadius.circular(10)
    ),
    width: double.infinity,
    child: Expanded(
      child: Padding(
        padding: const EdgeInsets.all(2),
        child: Row(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Card(
              // color: Colors.grey[900],
              child: Container(
                width: 140,
                padding: EdgeInsets.all(10),
                child: Column(
                  mainAxisAlignment: MainAxisAlignment.center,
                  children: [

                  Text("Quick and",style:
                   TextStyle(color: Colors.black)),
                   // SizedBox(height: 3,),
                   Text("easy",style:
                    TextStyle(color: Colors.black)),
                   // SizedBox(height: 3,),
                   Text("appointments",style:
                   TextStyle(color: Colors.black)),
                ],),
              ),
            ),
            SizedBox(width: 30),
            Card(
              // color: Colors.grey[900],
              child: Container(
                width: 140,
                padding: EdgeInsets.all(10),
                child: Column(
                  crossAxisAlignment: CrossAxisAlignment.center,
```

```
                              mainAxisAlignment: MainAxisAlignment.center,
                              children: [

                              Text("Safe and", style:
                              TextStyle(color: Colors.black),),
                              // SizedBox(height: 3,),
                              Text("effective",style:
                              TextStyle(color: Colors.black)),
                              // SizedBox(height: 3,),
                              Text("vaccines",style:
                               TextStyle(color: Colors.black)),
                            ],),
                          ),
                        ),
                      ],
                    ),
                  )),
              ),
            )

          ],
                ),
        )
      ),
    );

  }
}
```

profile.dart

```
import 'package:doctor_baby/view/bottom_nav.dart';
import 'package:doctor_baby/view/calendar.dart';
import 'package:doctor_baby/view/home.dart';
import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;
import 'dart:convert';
import 'dart:io';
import 'package:image_picker/image_picker.dart';
import 'package:shared_preferences/shared_preferences.dart';

class ProfilePage extends StatefulWidget {

  static int? userId;

  @override
  _ProfilePageState createState() => _ProfilePageState();
```

```
}

class _ProfilePageState extends State<ProfilePage> {
  final _firstnameController = TextEditingController();
  final _lastnameController = TextEditingController();
  final _dobController = TextEditingController();
  final _parentNameController = TextEditingController();
  File? _profileImage;
  String _selectedGender = 'Male';



  Future<void> _saveProfile() async {
    if (_validateForm()) {
      final url = 'http://10.0.2.2:8000/babyapp/childcreate/';

      final requestData = {
        'first_name': _firstnameController.text,
        'last_name': _lastnameController.text,
        'date_of_birth': _dobController.text,
        'sex': _selectedGender,
        'parent_username': _parentNameController.text,
      };

      try {
        final response = await http.post(
          Uri.parse(url),
          body: json.encode(requestData),
          headers: {
            'Content-Type': 'application/json',
          },
        );

        if (response.statusCode == 201) {
          print(response.body);
          _showSnackBar('Profile created successfully!');
          final Map<String, dynamic> responseData =
          json.decode(response.body);

          // userId = responseData['id'];
          ProfilePage.userId = responseData["id"];


          await saveFirstName(_firstnameController.text);
```

```
      print('Retrieved ID: ${ProfilePage.userId}');
        Navigator.of(context).push(MaterialPageRoute(builder:
         (context) => Home()));
      } else {
        _showSnackBar('Failed to create profile. Please try again.');
      }
    } catch (e) {
      print('Exception: $e');
      _showSnackBar('Failed to create profile. Please try again.');
    }
  }
}


Future<void> saveFirstName(String firstName) async {
  SharedPreferences prefs = await SharedPreferences.getInstance();
  prefs.setString('firstName', firstName);
}



bool _validateForm() {
  if (_firstnameController.text.isEmpty ||
      _lastnameController.text.isEmpty ||
      _dobController.text.isEmpty ||
      _parentNameController.text.isEmpty ||
      _selectedGender.isEmpty) {
    _showSnackBar('Please fill in all fields.');
    return false;
  }
  return true;
}

void _showSnackBar(String message) {
  ScaffoldMessenger.of(context).showSnackBar(
    SnackBar(content: Text(message)),
  );
}

 Future<void> _selectDate(BuildContext context) async {
  final DateTime? picked = await showDatePicker(
    context: context,
    initialDate: DateTime.now(),
    firstDate: DateTime(1900),
    lastDate: DateTime.now(),
  );
```

```
    if (picked != null && picked != DateTime.now()) {
      setState(() {
        _dobController.text = picked.toLocal().toString().

split(' ')[0];
      });
    }
  }

  Future<void> _pickImage() async {
    final picker = ImagePicker();
    final pickedFile = await picker.pickImage(source:
   ImageSource.gallery);

    if (pickedFile != null) {
      setState(() {
        _profileImage = File(pickedFile.path);
      });
    }
  }
  @override
  Widget build(BuildContext context) {
    return SafeArea(
      child: Scaffold(
        body: Container(
          decoration: BoxDecoration(image: DecorationImage
         (image: NetworkImage("https://images.unsplash.com/
          photo-1582486225644-aeacf6aa0b1bq=80&w=
         1887&auto=format&fit=crop&ixlib=rb-4.0.3&ixid=

M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA
%3D%3D"), fit: BoxFit.cover)),
          height: double.infinity,
          child: Padding(
            padding: const EdgeInsets.all(16.0),
            child: SingleChildScrollView(
              child: Card(
                color: Colors.black45,
                child: Padding(
                  padding: const EdgeInsets.all(12.0),
                  child: Form(
                    child: Column(
                      crossAxisAlignment: CrossAxisAlignment.start,
                      children: [
```

```
  SizedBox(height: 20),
Text("Create a profile for your baby",
style: TextStyle(color: Colors.grey[200],
fontSize: 25),),

  SizedBox(height: 70),

  TextFormField(style: TextStyle(color:
  Colors.grey[100]),
    controller: _firstnameController,
    decoration: InputDecoration(labelText:
    'First Name',
      labelStyle: TextStyle(fontWeight:
     FontWeight.bold, color: Colors.grey[300]),
      border: OutlineInputBorder
    (borderRadius: BorderRadius
          .circular(20) ),),

    validator: (value) {
      if (value == null || value.isEmpty) {
        return 'Please enter your first name.';
      }
      return null;
    },
  ),
  SizedBox(height: 12),
  TextFormField(style: TextStyle(color:
   Colors.grey[100]),
    controller: _lastnameController,
    decoration: InputDecoration(labelText:
   'Last Name',
      labelStyle: TextStyle(fontWeight:
      FontWeight.bold, color: Colors.grey[300]),
      border: OutlineInputBorder(
      borderRadius: BorderRadius
          .circular(20) ),),

    validator: (value) {
      if (value == null || value.isEmpty) {
        return 'Please enter your last name.';
      }
      return null;
    },
  ),
  SizedBox(height: 12),
  TextFormField(style: TextStyle(
```

```
                        color: Colors.grey[100]),
                          controller: _dobController,
                          decoration: InputDecoration(
                            labelText: 'Date of Birth',  labelStyle:
                          TextStyle(fontWeight: FontWeight.bold,
                           color: Colors.grey[300]),
                            border: OutlineInputBorder(
                            borderRadius: BorderRadius
                                .circular(20) ),),
                          onTap: () => _selectDate(context),
                          readOnly: true,
                        ),
                        SizedBox(height: 12),
                        TextFormField(style: TextStyle(color:
                          Colors.grey[100]),
                          controller: _parentNameController,
                          decoration: InputDecoration(labelText:
                         'Parent Name',  labelStyle: TextStyle(fontWeight:
                         FontWeight.bold,color: Colors.grey[300]),
                            border: OutlineInputBorder(
                          borderRadius: BorderRadius
                                .circular(20) ),),
                          validator: (value) {
                            if (value == null || value.isEmpty) {
                              return 'Please enter the parent name.';
                            }
                            return null;
                          },
                        ),
                        SizedBox(height: 24),

                        Text("Gender", style: TextStyle(fontWeight:

FontWeight.bold, color: Colors.grey[300])),
                        Row(
                          children: [
                            Radio(
                              value: 'Male',
                              groupValue: _selectedGender,
                              onChanged: (value) {
                                setState(() {
                                  _selectedGender = value.toString();
                                });
                              },
                            ),
                            Text('Male', style: TextStyle(
```

```
                            color: Colors.grey[200]),),
                            Radio(
                              value: 'Female',
                              groupValue: _selectedGender,
                              onChanged: (value) {
                                setState(() {
                                  _selectedGender = value.toString();
                                });
                              },
                            ),
                            Text('Female', style: TextStyle(

color: Colors.grey[200]),),
                          ],
                        ),


                        SizedBox(height: 25),

                        Center(
                          child: GestureDetector(
                            onTap: _saveProfile,
                            child: Container(
                              padding: EdgeInsets.all(10),
                              child: Text(
                                "Create profile",
                                style: TextStyle(fontSize: 16,
                                color: Colors.black),
                              ),
                              decoration: BoxDecoration(
                                borderRadius: BorderRadius.circular(15),
                                color: Colors.blue,
                              ),
                            ),
                          ),
                        ),
                        SizedBox(height: 20),
                      ],
                    ),
                  ),
                ),
              ),
            ),
          ),
        ),
      ),
```

```
    );
  }
}


  class Util {
  static Future<String?> getFirstName() async {
    SharedPreferences prefs = await SharedPreferences.getInstance();
    return prefs.getString('firstName');
  }
}


void main() {
  runApp(MaterialApp(
    home: ProfilePage(),
  ));
}
```

mail.dart

```
import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;

void sendMailToParent() async {
  var url = 'http://10.0.2.2:8000/babyapp/send_mail_date/';

  try {
    var response = await http.get(
      Uri.parse(url),
      headers: <String, String>{
        'Content-Type': 'application/json; charset=UTF-8',
      },
    );

    if (response.statusCode == 200) {
      print('Email sending initiated.');
    } else {
      print('Failed to send email. Status code: ${response.statusCode}');
    }
  } catch (e) {
    print('Error sending email: $e');
  }
}
```

WEBSITE OF ABILITY CONNECT

PROJECT REPORT

Submitted By

MITHRA PROTHASIS

Reg. No. CCAVBCA008

For the award of the Degree of

Bachelor of COMPUTER APPLICATION(BCA)

in Computer Science

(University of Calicut)

under the guidance of

Ms. Rasmi P M

Assistant Professor

BACHELOR OF COMPUTER APPLICATION
DEPARTMENT OF COMPUTER SCIENCE
CHRIST COLLEGE
IRINJALAKUDA, KERALA

2021-2024

DEPARTMENT OF COMPUTER SCIENCE

CHRIST COLLEGE (AUTONOMOUS)

IRINJALAKUDA

CERTIFICATE

This is to certify that the project report entitled ***"Website of Ability Connect"*** *is a bonfied record of the project work done by* ***MITHRA PROTHASIS*** *in partial fulfillment of the requirement for the sixth semester of* ***Bachelor of Computer Science*** *in Department of Computer Science of* **CHRIST COLLEGE (AUTONOMOUS) IRINJALAKUDA**

Ms. Rasmi P M
Assistant Professor,CS
Internal Guide

Ms. Sini Thomas
Head of the Department
Computer Science

EXTERNAL EXAMINER

INTERNAL EXAMINER

DECLARATION

We here by declare that this project work **"WEBSITE OF ABILITY CONNECT"** submitted by Christ College (Autonomous)Irinjalakuda,affliated to Calicut University in partial fullfillment of the requirement for the award of the Bachelor of Computer Application, is a record of original work done by us,under the guidance of Ms.RASMI P M,Department of Computer Science.

Place: Irinjalakuda MITHRA PROTHASIS

ACKNOWLEDGEMENT

First and foremost we like to thank Lord almighty for his providence and for being the guiding light throughout the project. We wish to express my sincere gratitude to our beloved Department head for giving me all the facilities for our project. We take this opportunity to express my gratitude to the class teacher Ms.SOWMYA P.S and head of the department Ms.SINI THOMAS who has been supported us throughout the course of this project. We are thankful for her aspiring guidance and valuable advice during the project work. We express my sincere thanks to my project guide Ms.RASMI P M for supporting and guiding throughout the project.We would take this opportunity to specially thank all other faculty members for their constant and continuous motivation. Finally we would like to thank my family and friends for giving valuable advice and moral support throughout our project.

ABSTRACT

WEBSITE OF ABILITY CONNECT is a innovative website introduced as a learning platform for differently abled students in assistance of Computer Science Department of Christ College(Autonomous) Irinjalakuda. The website is enriched with two kind of login facilities - student login, teachers login. It is a learning platform and the main features are- student registration, learning and exam dashboards and so on. All these features make this website more adaptable and user-friendly.

Contents

Chapter 1

1 Introduction

Education is the cornerstone of empowerment and growth, yet education systems often fail to accommodate the diverse needs of students with various disabilities .Our website endeavors to help the learning process by crafting inclusive assessments specifically tailored to each student's abilities, while also introducing a spectrum of learning methods—audio, text, video, and interactive activities that can help foster a learning environment.

.

1.1 Overview

The objective of the Ability Connect Website is to design an simple and adaptable website that helps the users in their learning process by tailoring different modes through which they can access education . Different modes of examinations, in-website voice navigation and other features helps website to be user friendly and create a learning environment for differently abled students .

Chapter 2

2 System Analysis

2.1 Purpose

The main purpose of the website owned by Don Davis, Adhithyan T.J, Leyon T.John ,Mithra prothasis is to make a user friendly website as a inclusive learning platform for diffrently abled students.

2.1.1 Existing System

In the existing system, there may be limited opportunities for disabled children to learn or develop their skills independently, especially at a young age. While schools and special education programs may offer support from teachers and other aides, there may be gaps in providing opportunities for self-directed learning and skill development outside of structured classroom settings. Additionally, parents of disabled children may face challenges in providing supplemental learning experiences or resources at home due to lack of time, knowledge, or access to suitable materials. Limitations of existing system : Self-directed learning and skill development outside of structured classroom settings and difficulties in accesing diffrent modes of learning according to disabilities.

2.1.2 Proposed System

The materials will be designed to be accessible to children with various disabilities. Users can specify any disabilities or special needs they have to ensure that the platform can customize them.The website will incorporate learning technologies to adjust the difficulty level and pacing of activities based on the student's performance and progress. This ensures that each student is appropriately challenged and supported throughout their learning journey. Proposed system aims to create an inclusive and empowering learning environment for disabled children. The website will provide feedback to students on their performance and progress, including scores, achievements, and areas for improvement. Parents or teachers may also have access to progress reports to monitor the student's development and provide additional support as needed. Other Features : A message system that sends notification to user mail about the new contents or exams assigned.

2.2 Problem definition

To know what the problem is and what the needs are before developing it. Here we design a website as a learning platform for differently abled students.

2.3 Feasibility Study

After the problem is clearly understood and solutions proposed, the next step is to conduct the feasibility study. Feasibility study is defined as evaluation or analysis of the potential impact of the proposed project or program. The objective is to determine whether the proposed system is feasible. There are three aspects of feasibility study to which the proposed system is subjected as discussed below.

2.3.1 Technical Feasibility

Technical feasibility assess whether the current technical resources are sufficient for the new system. We can upgrade the level of technology for supporting our website. We check whether the proposed system can be implemented in the present system without supporting the existing hardware.

2.3.2 Economical Feasibility

Economic feasibility determines whether the time and money are available to develop the system. There is no additional hardware used to develop the site, it is inexpensive to build.

2.3.3 Operational Feasibility

Operational feasibility determines if the human resources are available to operate the system once it has been installed. No extra training is needed to use this system. Anyone who has knowledge in internet in english language can easily use the system. The resources that are required to implement or install are already available with the organization.

Chapter 3

3 Software Requirement Specification

3.1 Purpose

The purpose of creating an educational website for differently-abled students is multifaceted and revolves around addressing the unique needs and challenges that these students may face. The educational website for differently-abled students is to create an inclusive, accessible, and supportive online learning environment that empowers individuals with diverse abilities to thrive academically and personally. The project aims to bridge gaps, break down barriers, and contribute to a more inclusive educational landscape.

3.2 Scope

The scope of the "Educational Website for Differently-Abled Students" project is a comprehensive initiative aimed at developing an inclusive and accessible online learning platform. The project will focus on catering to the unique needs of differently-abled students, educators, and administrators, providing a supportive and engaging environment for learning. It also aims to create a holistic and inclusive educational platform that not only meets the immediate needs of its users but also lays the groundwork for ongoing growth and enhancement.

3.3 Overall Description

The project for developing an "Educational Website for Differently-Abled Students" is a visionary initiative with the primary goal of creating an inclusive and accessible online learning platform. This comprehensive website is designed to cater to the diverse educational needs of students with varying abilities, ensuring they have equal access to quality learning resources and a supportive community. At the heart of the project is the creation of an adaptive learning environment. The website will feature a dynamic system that tailors educational content to the individual needs, preferences, and learning styles of differently-abled students.

3.3.1 Product Perspective

The "Educational Website for Differently Abled Students" involves considering how the website fits into the broader context of educational technology and the needs of its users.

3.3.2 Product Functionality

Through this website teachers can upload study materials and quizzes as a part of exam and teacher can see the result of each student. The students get notified

if new material or new test been assigned. And the main highlight is that this website can be voice controlled specially for blind students.

3.3.3 Users and Characteristics

There are two types of users admin or teacher and student. In admin page the teacher can upload new study materials and tests, teacher can also see the growth of each student. And teachers can give suggestions for each type of student. In the student page they can see all the study materials and tests. There will be an entertainment section where there will be small stories, contents and games.

3.4 Specific Requirements

3.4.1 Hardware Requirements

- System: IBM-Compatible PC
- Processor: Intel Core i3 Or Above
- Speed: Above 1GHz
- RAM capacity: 4 GB Or Above
- Hard Disk Drive: 256 GB Or Above

3.4.2 Software Requirements

- Front End : HTML, CSS, Javascript
- Back End : Python
- Database : Sqlite3
- IDE : Visual Studio Code

3.5 Functional Requirements

It contains two main modules.

- 1.Teacher/Admin
- 2.Student

Admin

An admin account is used for editing or managing the website dynamically by admin panel. The admin can add new study materials, new tests and they can also review student performance and assign necessary suggestions according to their performance. The admin will be acting as teacher role.

Student

The user or the student can login the website, and can access the study materials assignd by the admin or teacher. The will be tests, suggestions, and small entertainment section for leisure time.

3.6 Non Functional Requirements

Non-functional requirements define the overall qualities or attributes of the resulting system. Non-functional system place restrictions on the product being developed, the developed process, and specify external constraints that the product must meet. Examples of non-functional requirements include safety, security, usability, reliability and performance requirements. Project management issues (costs, time and schedule) are often considered as non-functional requirements. The principal non - functional constraints which are relevant to critical systems :

- performance
- security
- safety
- usability

Performance

Performance requirements concern the speed of operation of a system. Types of performance requirements :

- Response requirements (how quickly the system reacts to a user input).
- Throughput requirements (how much the system can accomplish within a specified amount of time).
- Availability requirements (is the system available for service when requested by end users). The speed of operation of this system is adequate for the requirements.

Reliability

- Reliability is the ability of a system to perform its required function under stated conditions for a specified period of time.
- constraints on the runtime behavior of the system. This system is reliable because its functionalities can be done on the required conditions.

Safety

Safety requirements are not required which exclude unsafe situation from the possible solution of the system.

Usability

Usability is the ease with which a user can learn to operate, prepare inputs for, and interpret outputs of system or components. Usability requirements include :

- information error messages.
- well-formed user interfaces.

3.7 Interface Requirements

3.7.1 Hardware interfaces

The system must run over the internet, all the hardware shall require to connect internet will be hardware interface for the system. As for example modem, WAN-LAN.

3.7.2 Software interfaces

Software interface required for the working of the project is the appropriate operating system.

3.7.3 Communication interfaces

The user of the site communicate using the network connectivity and the data set is accessed

3.8 Security Requirements

- User accesses only their account.
- Validation of input is handled.
- This application containing the computer systems is physically secured against arms or surreptitious entry by intruders.
- Users must be authorized carefully to reduce changes of any such user giving access to an intruder in exchange for a bribe or other favour.

3.9 Platform Used

Windows 10 is a major version of the Microsoft Windows operating system that was released on July 29, 2015.It is built on the Windows NT kernel and follows windows 8.Part of the reason Microsoft decided to name the 2015 release "Windows 10"(and skipped "windows 9")is because the operating system is designed to be a new direction for Microsoft. One of the primary aims of windows 10 is to Unify the windows experience across multiple devices, such desktop computers, tablets, and smartphones. As part of this effort, Microsoft developed Windows10 Mobile alongside Windows 10 to replaces Windows Phone

- Microsoft's previous mobile OS. Windows 10 also integrates other Microsoft services, such as Xbox Live and the Cortana voice recognition assistant.

3.10 Technologies Used

JavaScript

JavaScript is a programming language that enables interactivity and dynamic behavior on web pages. It can be used to manipulate the HTML and CSS of a webpage, handle user interactions, fetch data from servers, and much more.HTML and CSS can be embedded with JavaScript to create a working efficient website. HTML is the standard markup language for creating web pages. It provides the structure of a webpage by using elements or tags to define different parts of the content. CSS is used to style the HTML elements and define their appearance on the webpage. It allows you to control the layout, colors, fonts, and other visual aspects of your website.

Sqlite3

SQLite is a lightweight, serverless, self-contained, and embedded SQL database engine. It's widely used in various applications due to its simplicity, efficiency, and ease of integration. In the context of a website, SQLite can be utilized to store and manage data on the server-side, providing persistence for web applications. The Sqlite libraries provide APIs for performing CRUD (Create, Read, Update, Delete) operations, executing SQL queries, and managing database connections. By integrating SQLite into your website, you can efficiently store and manage data, enabling features like user authentication, data persistence, and dynamic content generation.

Chapter 4

4 Design Document

4.1 Purpose

The purpose of the website is to provide accessible and inclusive educational resources for disabled students with visual and auditory impairments. The website aims to support these students in their studies by offering:

- Accessible Study Materials: The website provides notes and educational materials in various accessible formats, such as voice notes for blind students and video/photo classes for students with hearing impairments.
- Interactive Learning Tools: Students can engage with interactive learning tools tailored to their specific needs, including audio descriptions for visual content and subtitles or sign language interpretation for video content.
- Assessments and Tests: The website offers assessments and tests designed to accommodate the needs of disabled students, ensuring fair evaluation of their understanding and progress.
- Teacher Support and Suggestions: Teachers can provide personalized support and suggestions to disabled students through the platform, offering guidance and assistance to help them succeed academically.
- Voice Command Input: For blind students, the website incorporates voice command input functionality, allowing them to navigate the platform and interact with content using voice commands, enhancing accessibility and usability.

4.2 Scope

The website aims to provide accessible study materials, including voice notes for blind students and video/photo classes for those with hearing impairments. It facilitates interactive learning through tests, offers teacher suggestions, and incorporates voice command input for blind users, fostering an inclusive educational environment for disabled students.

4.3 Overview

The purpose of this document is to help the reader to visualize the solution to the project presented.This document verifies how the design meet the requirement stimulated in the SRS document through design viewpoints.The design viewpoints will cover all design elements presented before. This document will provide a direct approach to the development of this project hence reducing feature creep and ponitedly determine the quality of the design.

4.4 Data Design

Database are the storehouses of data used in the software systems.The data is stored in tables inside the database.Several tables are created for the manipulation of the data for the system.Two essential settings for a database are

- Primary Key- The field that is unique for all the record occurrence.
- Foreign Key- The field used to set relation between tables.

Normalization

- Normalization is the process of organizing the data in the database.
- Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate the undesirable characteristics like Insertion, Update and Deletion Anomalies.
- Normalization divides the larger table into the smaller table and links them using relationship.
- The normal form is used to reduce redundancy from the database table.

Tables

Login

Name	DataType	Constraints	Description
username	varchar(100)	Notnull	Username of student
password	varchar(100)	Notnull	Password of student

Student

Name	DataType	Constraints	Description
studentid	Charfield(50)	Primarykey	ID of users
firstname	Charfield(100)	Notnull	First name of users
lastname	Charfield(100)	Notnul	Last name of users
email	Emailfield)	Notnull	Email of the users
gender	Charfield(100)	Notnull	Gender of the user
age	PositiveIntegerfield	Notnull	Age of user
disability	Charfield(100)	Notnull	Disability of user
access technology	Charfield(200)	Notnul	Users access technology

Question Table

Name	DataType	Constraints	Description
type	Charfield	Not Null	Question Type
text_id	Charfield	Foreignkey	Text Question
image	Filefield	Foreignkey	Image Question
audio	Filefield	Foreignkey	Audio Question

Scoremodel Table

Name	DataType	Constraints	Description
student	Charfield	Foreignkey	Student Name
score	Integerfield	Foreignkey	Exam Score
category	Charfield	Foreignkey	Category
suggestion	Textfield	Not Null	Suggestions

Suggestions

Name	DataType	Constraints	Description
suggestion	Textfield	Notnull	Suggestions
category	Charfield	Foreignkey	Category
video	Filefield	Foreignkey	Suggested Video
audio	Filefield)	Foreignkey	Suggested audio

Chapter 5

5 Development of the System

The website will feature a user-friendly interface with options for visually impaired students to access voice notes and hearing-impaired students to engage in video and photo classes. It will include an integrated testing platform, teacher feedback system, and voice command functionality for blind students, aiming to enhance accessibility and support disabled students in their studies.

Chapter 6

6 System Testing

Testing is the penultimate step of software development.An elaborate testing of the data is prepared and the system is using the test data,while doing testing,errors are noted and correction is made.The users are trained to operate the developed system.Both hardware and software securities are made to run the developed system successfully.System testing is aimed at ensuring the system works accurately before the live operation commences.

6.1 Test Plan

6.1.1 Scope

This test plan will cover the following testing activities as identified in the testing strategy.

- White Box Testing

White box testing is a testing technique, that examines the program structure and derives test data from the program logic/code. In white box testing, the UI is bypassed. Inputs and outputs of ratings are tested directly at the code level and the results are compared against specifications. It Reveals errors in "hidden" code.

- Black Box Testing

Black - box testing is a method of software testing that examines the functionality of an application based on the specifications. Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end - user would.

- Unit Testing

The module includes four main components to undergo unit testing. It checks the syntax error, logic error and validity of the program correctness. This test will be performed by the developers.

- Integration Testing

After all components pass the unit testing, they get to be tested whether they work correctly when they are running concurrently and communicating to each other. Integration tests exercise an entire subsystem and ensure that a set of components play nicely together.

- Internal Data Testing

We will test the validity of the data before it enters the database to avoid any problems that may face us in the database. We will test the encryption of the personal information of all the users along with the user names and passwords to ensure maximum security of the users privacy.

6.1.2 Software risk issues

In this section, the plan is to test the risk involved in critical issues such as: Difficult to run on visual studio due to large loading time. Some inherent software risks such as complexity exist; also these issues need to be identified. Proper network connection Working of Mysql database

6.1.3 Features to be tested

- Test whether correct user name and password allows you to login.
- Test whether invalid user name and password prevents you from login.
- Test whether there is any connection problem in the Server.
- Test whether the student and admin details are entered correctly.
- Test whether invalid data entry allows saving data successfully.
- Test whether all pages are loaded correctly.
- Test whether watermark is embedded and extracted properly.
- Test whether the provided security works correctly.

6.2 Test consolidation

6.2.1 Test item

The items or features to be tested in the test cases are included in the document. Each and every user input is tested. The present condition of the system is tested. It is checked to make sure that environment is ready for the application to work.

6.2.2 Input specifications

These are the inputs required to execute the test case. All required inputs including data elements and values are given below.

Data	Values
Email	A valid email ID
Password	Combination of characters and numbers
Phone number	10 Digit number

Chapter 7

7 System Implementation and Maintenance

7.1 Implementation

System implementation is the conversion of new system into an operating one which involves creating compatible files, training clients and installing hardware. User training is crucial for minimizing resistance to change and giving chance to prove its worth. Training aids user friendly manuals and healthy screens provide the user with a good start. Software maintenance follows conversion to the extent that changes are necessary to maintain satisfactory operations relative to changes in the user's environment. Maintenance often includes minor enhancements or corrections to the problem that surface late in the systems operations. In the implementation phase, the team builds the components either from scratch or by composition. Given the architecture document meant from the design phase and the requirement document from the analysis phase, the team should build exactly what has been requested, though there is still room for innovation and flexibility. For example, a component may be narrowly designed for this particular system, or the component may be made more general to satisfy a reusability

- Careful planning
- Investigation of system and constraints
- Design the methods to achieve changeover.
- Training the staff in the changed phase.
- Evaluation of change over method.
- The method of implementation and time scale to be adopted are found out initially.

7.2 Maintenance

This phase occurs as a result of deploying the whole system at the end users organization. They will perform the beta testing at the end users and inform to the developers about any needed modification to the application. The customer records all the problems that are encountered during the beta testing and reports these to the developer at regular intervals.

7.2.1 Corrective Maintenance

Even with the best quality assurance activities, it is likely that the customer will uncover defects in the software. Corrective maintenance changes the software to correct the defects. Corrective Maintenance activity may consist of repair,

restoration or replacement of equipment. This activity will be result of a regular inspection, which identifies the failure in time for corrective maintenance to be planned and scheduled, then performed during a routine maintenance shutdown.

7.2.2 Adaptive Maintenance

Over time, the original environment(CPU, operating system, business rules, external product characteristics) for which the software was developed is likely to change. Adaptive maintenance results in modification to the software to accommodate changes to its external environment.

7.2.3 Enhanced Maintenance

As software is used, the customer/user will recognize additional functions that will provide the benefit. Perfect maintenance extends the software beyond its original functional requirements.

7.2.4 Preventive Maintenance

Computer software deteriorates due to change, and because of this preventive maintenance often called software re-engineering, must be conducted to enable the software to serve the needs of its end users. Preventive maintenance makes changes to computer programs so that they can be more easily corrected, adapted and enhanced.

Chapter 8

8 Conclusion and Future Scope

8.1 Conclusion

There are three types of users that interact with the system Admin,Teacher and Student.Each of these three types has different uses of the system so each of them has their own panel.Admin can manage all the features of website dynamically by login on admin panel.Teacher can view the details about their event assigned by the admin and publish the result of the event.And the student can register for events and view the results of the events.

8.2 Future Scope

- Advanced adaptive learning technologies can be implemented.
- Advanced AI Chatbots

Appendix

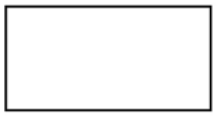
A Data Flow Diagram

Data flow is the one of the best way of documenting the entire functionality of the system.For the system ,which will have data flows in and have some processing inside and then some data flow out from the system can be documented or represented effectively by means of data flow out from the system can be documented or represented effectively by means of data flow diagrams. The data flow diagram are a diagrammatic representation of the system,which has input,process and output.Once any system is represented using a data flow diagram we can identify the following things easily:

- Various entities interacting with the system are identified
- Flow of data from one entity to another is identified
- The various processes involved in between the interaction of two or more entities in the system are clearly pointed out
- The various data stores which hold the data in between the process,are clearly identified

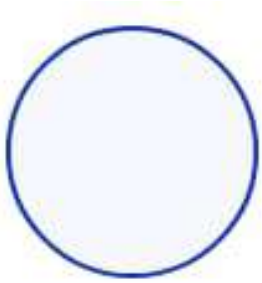
Some Data Flow Diagram charting forms:

A.1 External source or receiver



A source or sink is a person or part of organization ,which enters or receives information from the system,but is considered to be outlining the contest of data flow model.

A.2 Transform process

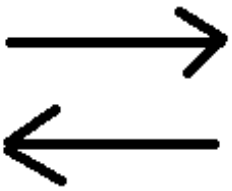


A process represents transformation where incoming data flowsare changed into outgoing data flow

A.3 Data Store

A data store is repository of data that is to be stored for use by one or more process may be as simple as buffer or queue or sophisticated as relational database. They should have clear names.If a process merely uses the contest of store and does not alter it,the arrowhead goes only form the store to the process. If a process alters the details in the store then double-headed arrow is used.

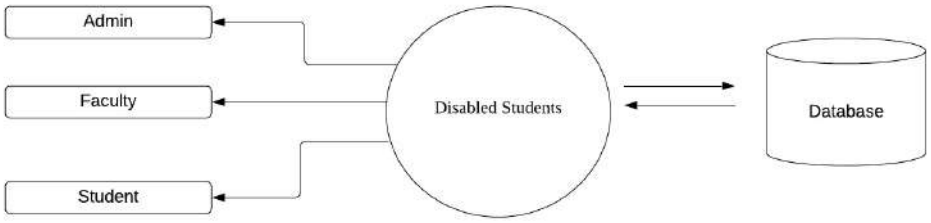
A.4 Data flow



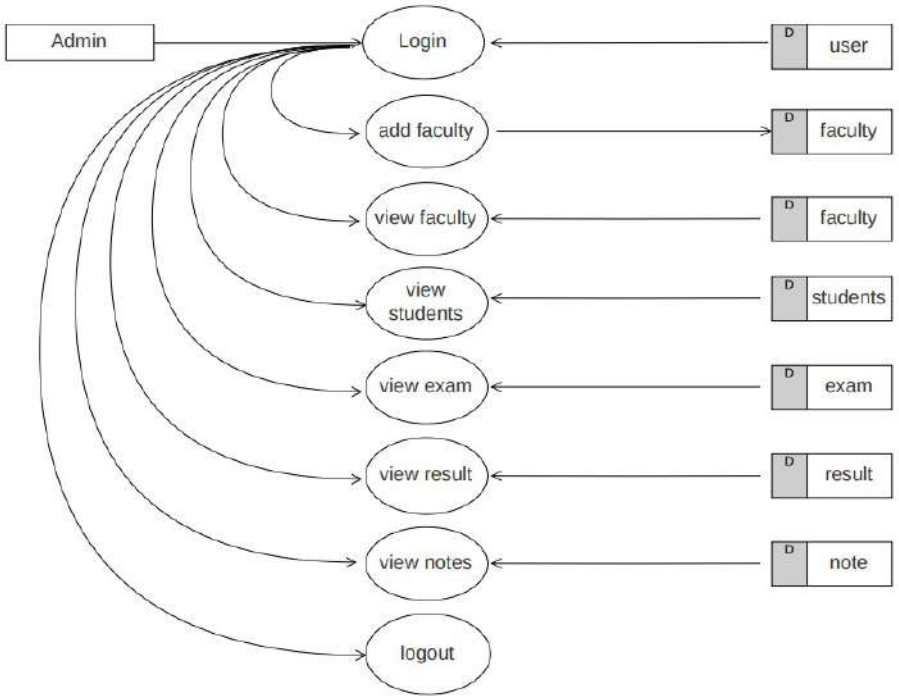
A data flow is a route, which enable packets of data to travel from one point to another.Data may flow,with arrowhead pointing in the direction of the flow

B Data Flow Diagrams

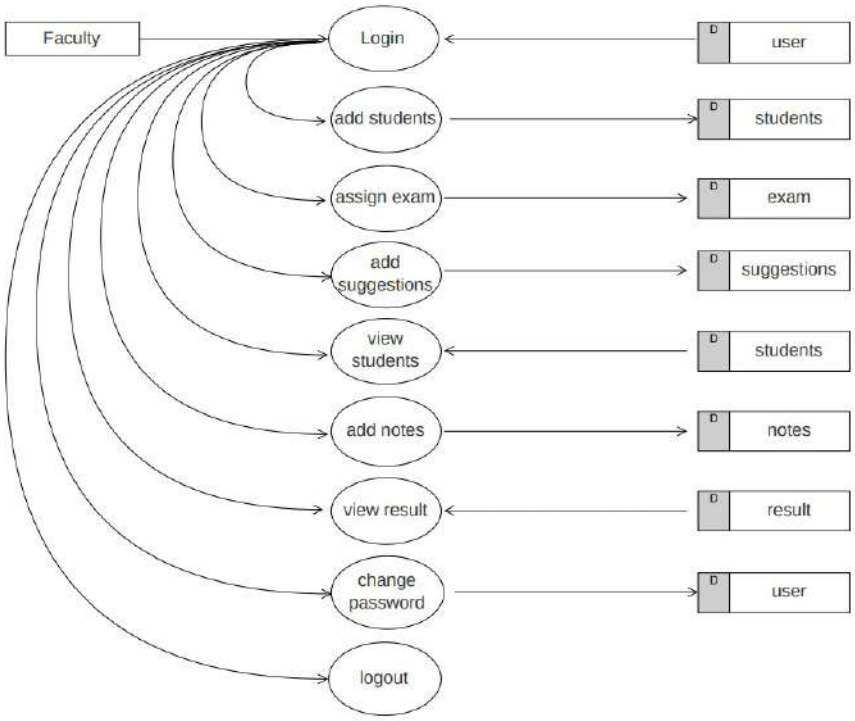
B.1 Level 0



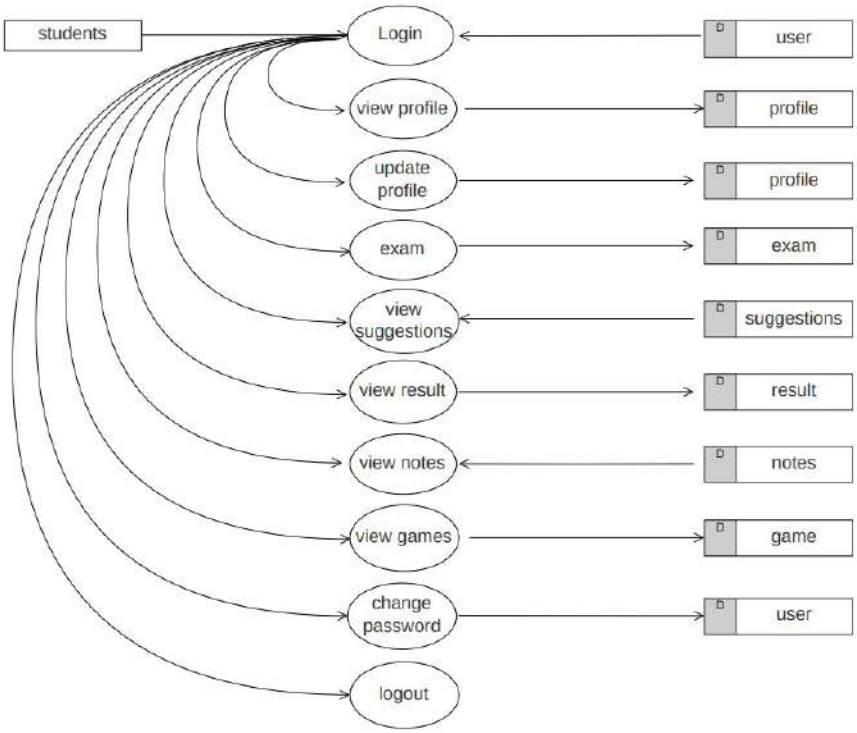
B.2 Level 1.1 - Admin



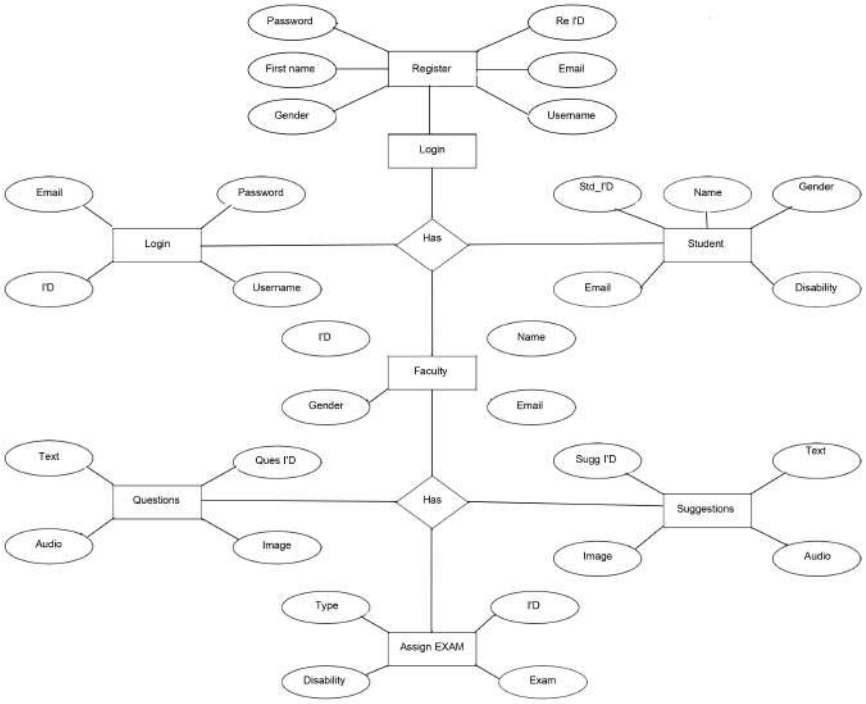
B.3 Level 1.2 - Faculty



B.4 Level 1.3 - Students

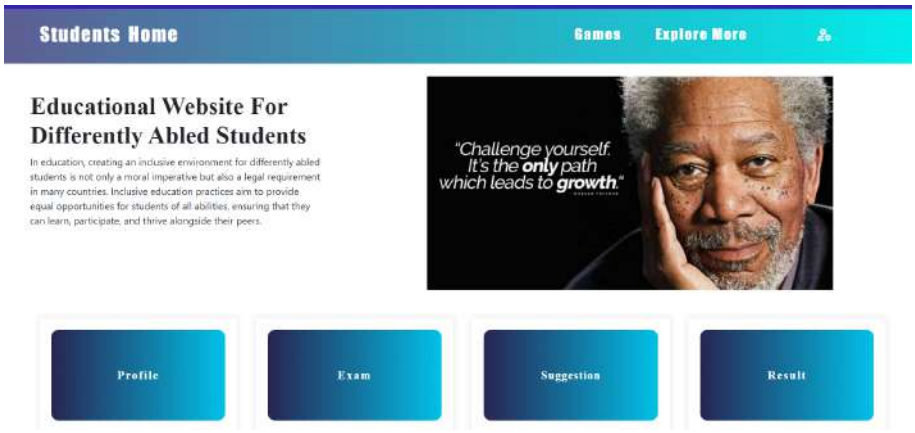


B.5 ER DIAGRAM

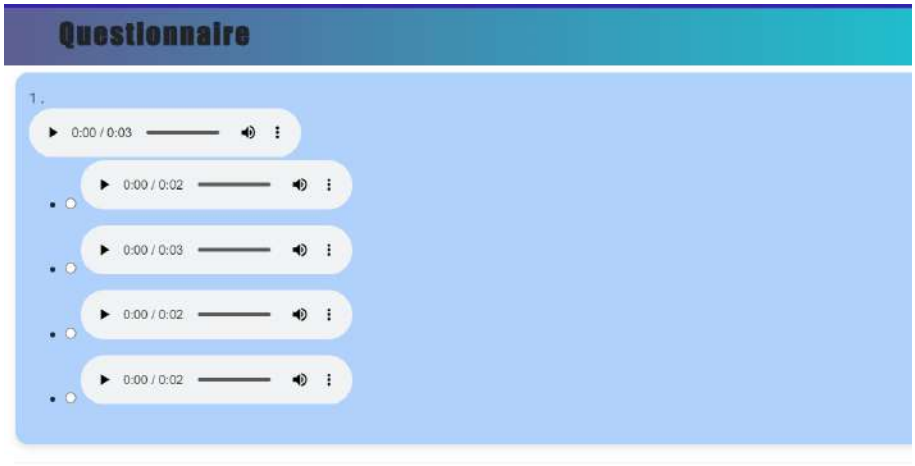


C USER INTERFACES

C.1 HOME



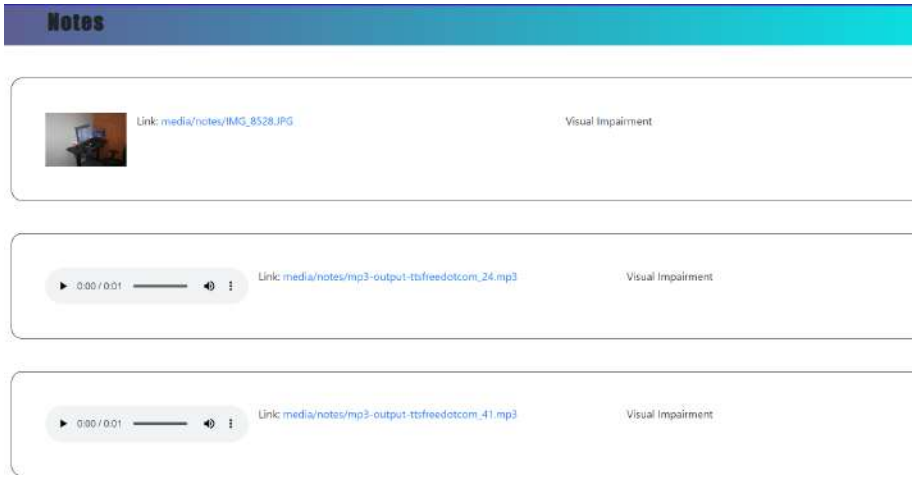
C.2 EXAM



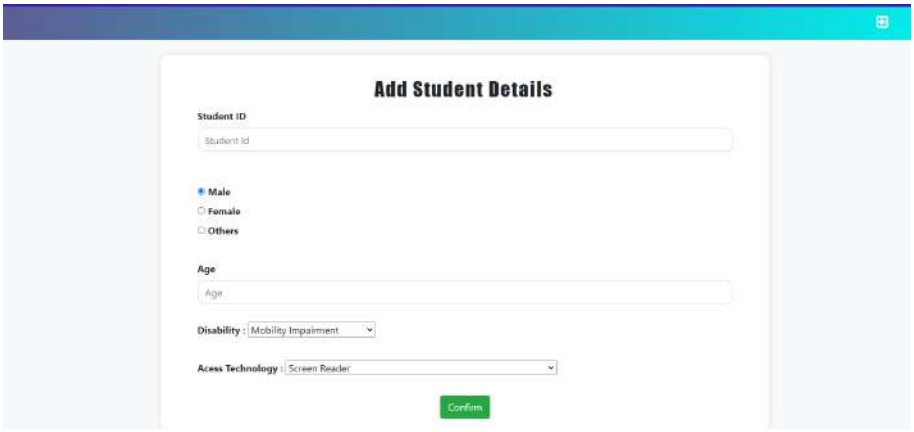
C.3 RESULT


Result
TestScore
1
Category
Very Poor
Result is here Check it....
No.
Question
Result
1
0:00 / 0:03
False
2
0:00 / 0:03
False
3
0:00 / 0:02
False
4
0:00 / 0:06
True

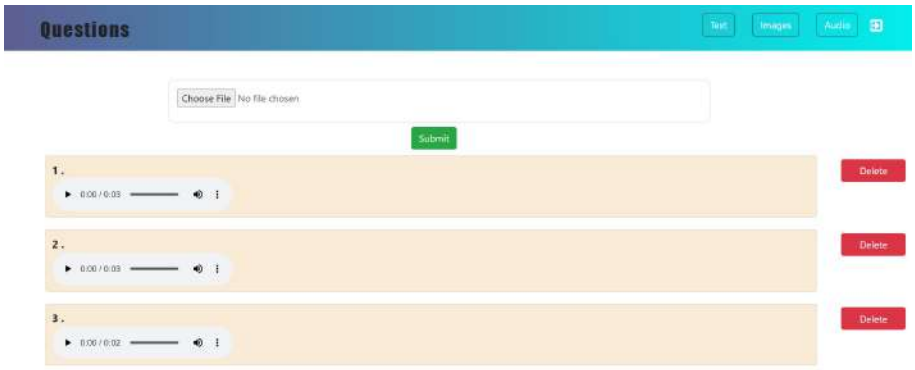
C.4 NOTES


Notes
Link: media/notes/IMG_8528.JPG
Visual Impairment
0:00 / 0:01
Link: media/notes/mp3-output-ttsfreedotcom_24.mp3
Visual Impairment
0:00 / 0:01
Link: media/notes/mp3-output-ttsfreedotcom_41.mp3
Visual Impairment

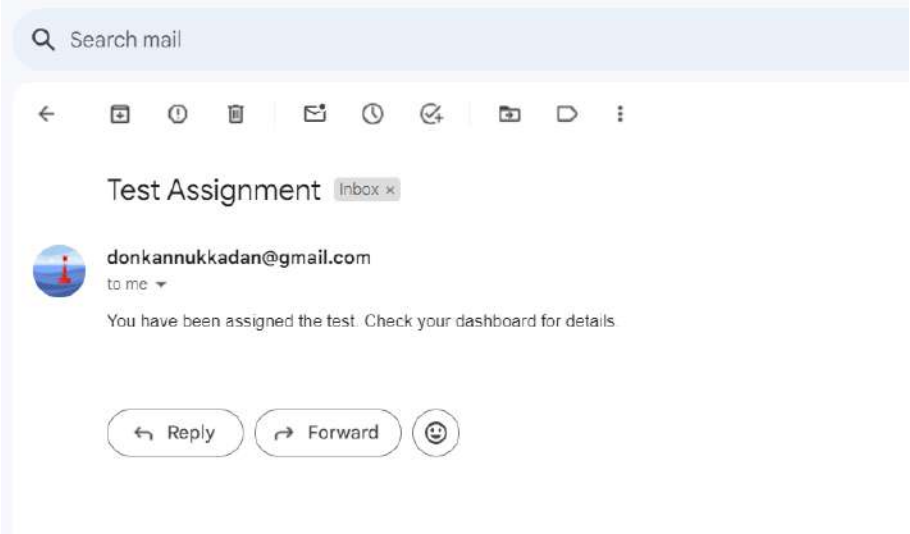
C.5 REGISTRATION



C.6 QUESTIONS WINDOW



C.7 NOTIFICATION



D CODE

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width,
      initial-scale=1.0">
  <!-- <link rel="stylesheet" href="styles.css"> -->
  <script src="toggle-sideNav.js" defer></script>
  <script src="featured-games.js" defer></script>

  <title>Game for mentally challenged people</title>
  <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,
        initial-scale=1, shrink-to-fit=no">
    <!-- Optional JavaScript -->
    <!-- jQuery first, then Popper.js, then Bootstrap JS
        -->
    <script src="https://code.jquery.com/jquery-3.3.1.
        slim.min.js" integrity="sha384-q8i/X+965
        DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8
        abtTE1Pi6jizo" crossorigin="anonymous"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/
        popper.js/1.14.7/umd/popper.min.js" integrity="
        sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9W
      O1clHTMGa3JDZwrnQq4sF86dIHNDz0W1" crossorigin="
          anonymous"></script>
    <script src="https://stackpath.bootstrapcdn.com/
        bootstrap/4.3.1/js/bootstrap.min.js" integrity="
        sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/
        nJGzIxFDsf4x0xIM+B07jRM" crossorigin="anonymous
        "></script>
    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://stackpath.
        bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min
        .css" integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH
        /1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
        crossorigin="anonymous">
    <link rel="stylesheet" href="https://fonts.googleapis
        .com/css2?family=Material+Symbols+Sharp:opsz,wght,
        FILL,GRAD@48,700,0,200" />
```

```
    <link rel="stylesheet" href="https://fonts.googleapis
        .com/css2?family=Material+Symbols+Outlined:opsz,
        wght,FILL,GRAD@24,700,0,200" />
   <link rel="stylesheet" href="https://fonts.googleapis.
       com/css2?family=Material+Symbols+Rounded:opsz,wght,
       FILL,GRAD@40,600,0,200" />
  </head>
<style>
  #play-now{
    text-decoration: none;
  }
</style>
<body>
{% load static %}
<div style="margin-left:95%;text-decoration: none;padding
    : 4px;border-radius: 0.5rem;" ><a href="{% url 'sh'
    %}" class="text-black ml-3" style="color: black;"><
    span class="material-symbols-outlined">
   exit_to_app
  </span></a></div>
    <section class="hero-wrapper">
        <div class="container">
          <p class="greeting " style="color: green;">
            Brain Games For Disabled Students!!!
          </p>
        </div>
      </section>
    <section id="play-now" class="program-wrapper">
      <div class="program-container container">
        <h3 class="title">
            Featured Games
        </h3>
        <div class="program-content">
          <div class="row">
            <div class="col">
                  <a href="{% url 'animal' %}" ><div
                      class="program-detail">
                    <img src="{% static 'images/first.png
                        ' %}" alt="reloading" width="350px
                        " height="450px">
                    <h5 class="mt-1">who am i ?</h5>
                  <p>A great game for the brain! It
                      improves visual scanning, planning,
                      and spatial memory!! </p>
                </div></a>
            </div>
```

```
            <div class="col">
              <a href="{% url 'math' %}"><div class="
                  program-detail">
                <img src="{% static 'images/third.jpg'
                    %}" alt="icon of a speaker" width="550
                    px" height="450px">
                <h5 class="mt-1">Fun with numbers</h5>
              <p>Helps to quickly solve the elementary
                  math problems!! </p>
            </div></a>
            </div>
        </div>
        <div class="row mt-5">
         <div class="col">
           <a href="{% url 'memory' %}"><div class="
               program-detail">
           <img src="{% static 'images/forth.jpg' %}"
              alt="icon of a speaker" width="500px">
           <h5>Behind the scenes</h5>
         <p>Helps to improve Visual Scanning and memory
             while remembering the cards!! </p>
         </div></a>
         </div>
        </div>
         </div>
    </section>
  <script>
   function playWelcomeMessage() {
         const welcomeMessage = new
            SpeechSynthesisUtterance('welcome games');
         window.speechSynthesis.speak(welcomeMessage);
         }
   playWelcomeMessage();
  var recognition = new webkitSpeechRecognition();
  recognition.continuous = true;
  recognition.lang = 'en-US';
  recognition.onresult = function(event) {
    var result = event.results[event.results.length -
        1][0].transcript.toLowerCase();
    console.log("result",result);
    var currentQuestionIndex=0;
     if (result.includes('back to home')) {
        // Redirect to the home page
        window.location.href = '{% url "sh" %}';
    }
};
```

```
 recognition.onerror = function(event) {
      console.error('Speech recognition error:', event.
          error);
  };
  recognition.onend = function() {
      // Restart recognition when it ends
      recognition.start();
  };
  // Start speech recognition
  recognition.start();
  </script>
</body>
</html>
```

views.py

```
from typing import Any
from django.forms.models import BaseModelForm
from django.http import HttpResponse
from django.shortcuts import render,redirect,
    HttpResponseRedirect
from .models import *
from .models import Question
from .forms import *
from django.views.generic import FormView,CreateView,
    UpdateView,TemplateView,View
from django.contrib.auth import authenticate,login,logout
from django.urls import reverse_lazy
from django.contrib.auth.hashers import make_password
from django.http import FileResponse
from django.shortcuts import get_object_or_404
from .models import Suggestion
from student_app.forms import ChangePasswordForm
from django.forms import formset_factory
# Create your views here.

import pandas as pd
import random
from django.db.models.signals import post_save
from django.dispatch import receiver
from django.contrib.auth import get_user_model
from django.http import JsonResponse

class MainHome(TemplateView):
    template_name="mainhome.html"
```

```
class ClearDataView(View):
    def post(self, request):
        StudentAnswer.objects.all().delete()
        StudentAnswerImage.objects.all().delete()
        StudentAnswerAudio.objects.all().delete()
        ScoreModel.objects.all().delete()
        return JsonResponse({'message': 'Data cleared
            successfully'})
@receiver(post_save, sender=Student)
def create_user_from_student(sender, instance, created,
    **kwargs):
    if created:
        # User = get_user_model()
        CustUser= get_user_model()
        student_id_id=instance.id
        username = instance.std_id
        password = 'admin@123'  # Set your desired
            default password here
        # User.objects.create_user(username=username,
            password=password)
        CustUser.objects.create_user(username=username,
            password=password,student_id_id=student_id_id)
        # ScoreModel.objects.create(student_id=
            student_id_id)
class LoginView(FormView):
    template_name="login.html"
    form_class=LogForm
    def post(self,request,*args,**kwargs):
        log_form=LogForm(data=request.POST)
        if log_form.is_valid():
            us=log_form.cleaned_data.get('username')
            ps=log_form.cleaned_data.get('password')
            user=authenticate(request,username=us,
                password=ps)
            if user:
                login(request,user)
                if request.user.is_superuser == 1:
                    return redirect('h')
                else:
                    return redirect('sh')
            else:
                return render(request,'login.html',{"form
                    ":log_form})
        else:
            return render(request,'login.html',{"form":
                log_form})
```

```
class AddStudent(CreateView):
    template_name='addstudent.html'
    model=Student
    form_class=StudentForm
    success_url=reverse_lazy('stu')
class QuestView(TemplateView):
    template_name='test.html'
    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)
        id=kwargs.get('pk')
        context['stu']=Student.objects.get(id=id)
        context['ques']=Question.objects.all()
        return context
# class QuestViewAll(TemplateView):
#     template_name='Assignexam.html'
#     def get_context_data(self, **kwargs):
#         context = super().get_context_data(**kwargs)
#         context['stu']=Student.objects.all()
#         context['ques'] = Question.objects.all()
#         return context

# class QuestViewImageAll(TemplateView):
#     template_name='testallimage.html'
#     def get_context_data(self, **kwargs):
#         context = super().get_context_data(**kwargs)
#         context['stu']=Student.objects.all()
#         context['ques'] = QuestionImages.objects.all()
#         return context

# class QuestViewAudioAll(TemplateView):
#     template_name='testallaudio.html'
#     def get_context_data(self, **kwargs):
#         context = super().get_context_data(**kwargs)
#         context['stu']=Student.objects.all()
#         context['ques'] = QuestionAudio.objects.all()
#         return context

def Test(request,**kwargs):
    if request.method == 'POST':
        id=kwargs.get('pk')
        stu=Student.objects.get(id=id)
        que=Question.objects.all()
        assignment = StudentAnswer(student=stu, question=
            que)
        assignment.save()
        return redirect('det')
```

```
from django.core.mail import send_mail

def AssignView(request,**kwargs):
    if request.method == 'POST':
        # Get all available tests
        tests = Question.objects.all()

        # Get all students
        students = Student.objects.filter(disability__in
            =["Mobility Impairment", "Learning Disability
            ", "Autism Spectrum Disorder", "Speech
            Impairment", "Intellectual Disability"])
        students_with_email_sent = set()
        # Assign all tests to all students
        for test in tests:
            for student in students:

            assignment, created = StudentAnswer.objects.
                get_or_create(student=student, question=
                test)
             if created:
            assignment.save()
            subject = 'Test Assignment'
            message = f'You have been assigned the test.
                Check your dashboard for details.'
            from_email = 'testhelloability@gmail.com'
            to_email = [student.email]
            if student in students_with_email_sent:
                      continue
             send_mail(subject, message, from_email,
                 to_email, fail_silently=False)
             students_with_email_sent.add(student)
        return redirect('testall')

    # Retrieve all available tests
    tests = Question.objects.all()

    return render(request, 'Assignexam.html', {'tests':
        tests})
def AssignImageView(request,**kwargs):
    if request.method == 'POST':
        # Get all available tests

        tests = QuestionImages.objects.all()
```

```
        # Get all students
        students = Student.objects.filter(disability="
            Hearing Impairment")
        students_with_email_sent = set()
        # Assign all tests to all students
        for test in tests:
            for student in students:

                assignment, created = StudentAnswerImage.
                    objects.get_or_create(student=student,
                     question=test)
                if created:
                 assignment.save()
                 subject = 'Test Assignment'
                 message = f'You have been assigned the
                     test. Check your dashboard for
                     details.'
                 from_email = 'testhelloability@gmail.com
                     '
                 to_email = [student.email]
                 if student in students_with_email_sent:
                  continue
                 send_mail(subject, message, from_email,
                     to_email, fail_silently=False)
                 students_with_email_sent.add(student)

        return redirect('testallvisual')

    # Retrieve all available tests
    tests = QuestionImages.objects.all()

    return render(request, 'testallimage.html', {'tests':
        tests})
def AssignAudioView(request,**kwargs):
    if request.method == 'POST':
        # Get all available tests
        tests = QuestionAudio.objects.all()

        # Get all students
        students = Student.objects.filter(disability="
            Visual Impairment")
        students_with_email_sent = set()

        # Assign all tests to all students
        for test in tests:
            for student in students:
```

```
                    assignment, created = StudentAnswerAudio.
                        objects.get_or_create(student=student,
                         question=test)
                    if created:
                     assignment.save()
                     subject = 'Test Assignment'
                     message = 'You have been assigned the
                         test. Check your dashboard for
                         details.'
                     from_email = 'donkannukkadan@gmail.com'
                     to_email = [student.email]
                     if student in students_with_email_sent:
                       continue
                     # Send email
                      send_mail(subject, message, from_email,
                           to_email, fail_silently=False)

                     # Add the student to the set of students
                          with sent emails
                      students_with_email_sent.add(student)

            return redirect('testallhear')

        # Retrieve all available tests
        tests = QuestionAudio.objects.all()

        return render(request, 'testallaudio.html', {'tests':
            tests})

from random import sample

class Quesadd(CreateView):
    template_name="quesadd.html"
    model=Question
    form_class=QuesForm
    success_url=reverse_lazy('qans')
    def get_context_data(self, **kwargs) :
        context = super().get_context_data(**kwargs)
        context['tests'] = Question.objects.all()
        return context

class Quesimage(CreateView):
    template_name="quesimage.html"
    model=QuestionImages
    form_class=QuesFormImage
    success_url=reverse_lazy('qansimg')
```

```
    def get_context_data(self, **kwargs) :
        context = super().get_context_data(**kwargs)
        context['tests'] = QuestionImages.objects.all()
        return context

class Quesaudio(CreateView):
    template_name="quesaudio.html"
    model=QuestionAudio
    form_class=QuesFormAudio
    success_url=reverse_lazy('qansaudio')
    def get_context_data(self, **kwargs) :
        context = super().get_context_data(**kwargs)
        context['tests'] = QuestionAudio.objects.all()
        return context

class QuesUpdate(UpdateView):
    template_name="questionupdate.html"
    model=Question
    form_class=QuesForm
    success_url=reverse_lazy('qdel')
    def get_context_data(self, **kwargs) :
        context = super().get_context_data(**kwargs)
        context['tests']=Question.objects.all()
        return context

class QuesAnsUpdView(CreateView):
    template_name="quesansupdate.html"
    model=Answer
    form_class=QuesAnsForm
    success_url=reverse_lazy('qdel')
    def form_valid(self, form):
        # Save the main answer
        main_answer = form.save(commit=False)
        main_answer.save()

        # Get additional options from the form data
        option_texts = [self.request.POST.get(f'option_{i
            }') for i in range(1, 4)]

        # Create and save additional options
        for option_text in option_texts:
            if option_text:
                answer_option = Answer(question=form.
                    cleaned_data['question'], text=
                    option_text)
                answer_option.save()
```

```
        return super().form_valid(form)

class QuesAnsImageView(CreateView):
    template_name = "quesansimage.html"
    model = AnswerImages
    form_class = QuesAnsFormImg
    success_url = reverse_lazy('qimg')

    def form_valid(self, form):
        # Save the main answer
        main_answer = form.save(commit=False)
        main_answer.save()

        # Handle file uploads for options
        for i in range(1, 4):
            option_file = self.request.FILES.get(f'
                option_{i}')
            if option_file:
                AnswerImages.objects.create(
                    question=main_answer.question,
                    fileans=option_file
                )
        return super().form_valid(form)

class QuesAnsAudioView(CreateView):
    template_name="quesansaudio.html"
    model=AnswerAudio
    form_class=QuesAnsFormAudio
    success_url=reverse_lazy('qaudio')
    def form_valid(self, form):
        # Save the main answer
        main_answer = form.save(commit=False)
        main_answer.save()

        # Handle file uploads for options
        for i in range(1, 4):
            option_file = self.request.FILES.get(f'
                option_{i}')
            if option_file:
                AnswerAudio.objects.create(
                    question=main_answer.question,
                    fileans=option_file
                )

        return super().form_valid(form)
```

```
class DeleteView(View):
    def get(self,req,*args,**kwargs):
        id=kwargs.get('pk')
        dl=Question.objects.get(id=id)
        dl.delete()
        return redirect('qdel')

class DeleteImgView(View):
    def get(self,req,*args,**kwargs):
        id=kwargs.get('pk')
        dl=QuestionImages.objects.get(id=id)
        dl.delete()
        return redirect('qimg')

class DeleteAudioView(View):
    def get(self,req,*args,**kwargs):
        id=kwargs.get('pk')
        dl=QuestionAudio.objects.get(id=id)
        dl.delete()
        return redirect('qaudio')

# class Quesdel(TemplateView):
#     template_name="quesdel.html"
#     def get_context_data(self, **kwargs) :
#         context = super().get_context_data(**kwargs)
#         context['tests']=Question.objects.all()
#         return context

class SugView(TemplateView):
    template_name="suggestions.html"
    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)
        context['data']=Suggestion.objects.all().order_by
            ('cat')
        return context

class SuggTextView(CreateView):
    template_name="suggtext.html"
    model=Suggestion
    form_class=SugForm
    success_url=reverse_lazy('st')

class SuggVideoView(CreateView):
    template_name="suggvideo.html"
    model=Suggestion
```

```
    form_class=SugVideoForm
    success_url=reverse_lazy('sv')

class SuggAudioView(CreateView):
    template_name="suggaudio.html"
    model=Suggestion
    form_class=SugAudioForm
    success_url=reverse_lazy('sadd')

def view_video(request, video_id):
    video = get_object_or_404(Suggestion, pk=video_id)
    video_path = video.video.path
    response = FileResponse(open(video_path, 'rb')) #
        Adjust content_type as needed
    return response

def view_videoo(request, videoo_id):
    video = get_object_or_404(ScoreModel, pk=videoo_id)
    video_path = video.video.path
    response = FileResponse(open(video_path, 'rb')) #
        Adjust content_type as needed
    return response

def play_audio(request, audio_id):
    audio_recording = get_object_or_404(Suggestion, pk=
        audio_id)
    audio_file = audio_recording.audio
    response = FileResponse(open(audio_file.path, 'rb'))
    return response

def play_audioo(request, audio_id):
    audio_recording = get_object_or_404(ScoreModel, pk=
        audio_id)
    audio_file = audio_recording.audio
    response = FileResponse(open(audio_file.path, 'rb'))
    return response

class DeleteViewSug(View):
    def get(self,req,*args,**kwargs):
        id=kwargs.get('pk')
        dl=Suggestion.objects.get(id=id)
        dl.delete()
        return redirect('sadd')

class ChangePasswordViewHome(FormView):
    template_name="changepshome.html"
```

```
    form_class=ChangePasswordForm
    def post(self,request,*args,**kwargs):
        form_data=ChangePasswordForm(data=request.POST)
        if form_data.is_valid():
            current=form_data.cleaned_data.get("
                current_password")
            new=form_data.cleaned_data.get("new_password
                ")
            confirm=form_data.cleaned_data.get("
                confirm_password")
            user=authenticate(request,username=request.
                user.username,password=current)
            if user:
                if new==confirm:
                    user.set_password(new)
                    user.save()
                    logout(request)
                    return redirect("log")
                else:
                    return redirect("cp")
            else:
                return redirect("cp")
        else:
            return render(request,"changepassword.html
                ",{"form":form_data})

class SuggestionUpdateView(UpdateView):
    template_name='suggestionupdate.html'
    model=Suggestion
    form_class=SuggestionForm
    success_url=reverse_lazy('sadd')
```

WEBSITE OF ABILITY CONNECT

PROJECT REPORT

Submitted By

MITHRA PROTHASIS

Reg. No. CCAVBCA008

For the award of the Degree of

Bachelor of COMPUTER APPLICATION(BCA)

in Computer Science

(University of Calicut)

under the guidance of

Ms. Rasmi P M

Assistant Professor

BACHELOR OF COMPUTER APPLICATION
DEPARTMENT OF COMPUTER SCIENCE
CHRIST COLLEGE
IRINJALAKUDA, KERALA

2021-2024

DEPARTMENT OF COMPUTER SCIENCE

CHRIST COLLEGE (AUTONOMOUS)

IRINJALAKUDA

CERTIFICATE

This is to certify that the project report entitled ***"Website of Ability Connect"*** *is a bonfied record of the project work done by* ***MITHRA PROTHASIS*** *in partial fulfillment of the requirement for the sixth semester of* ***Bachelor of Computer Science*** *in Department of Computer Science of* CHRIST COLLEGE (AUTONOMOUS) IRINJALAKUDA

Ms. Rasmi P M
Assistant Professor, CS
Internal Guide

26/2/24

Ms. Sini Thomas
Head of the Department
Computer Science

25/03/2024

EXTERNAL EXAMINER

21/3/24

INTERNAL EXAMINER

DECLARATION

We here by declare that this project work "**WEBSITE OF ABILITY CONNECT**" submitted by Christ College (Autonomous)Irinjalakuda,affiliated to Calicut University in partial fullfillment of the requirement for the award of the Bachelor of Computer Application, is a record of original work done by us,under the guidance of Ms.RASMI P M,Department of Computer Science.

Place: Irinjalakuda

MITHRA PROTHASIS

ACKNOWLEDGEMENT

First and foremost we like to thank Lord almighty for his providence and for being the guiding light throughout the project. We wish to express my sincere gratitude to our beloved Department head for giving me all the facilities for our project. We take this opportunity to express my gratitude to the class teacher Ms.SOWMYA P.S and head of the department Ms.SINI THOMAS who has been supported us throughout the course of this project. We are thankful for her aspiring guidance and valuable advice during the project work. We express my sincere thanks to my project guide Ms.RASMI P M for supporting and guiding throughout the project.We would take this opportunity to specially thank all other faculty members for their constant and continuous motivation. Finally we would like to thank my family and friends for giving valuable advice and moral support throughout our project.

ABSTRACT

WEBSITE OF ABILITY CONNECT is a innovative website introduced as a learning platform for differently abled students in assistance of Computer Science Department of Christ College(Autonomous) Irinjalakuda. The website is enriched with two kind of login facilities - student login, teachers login. It is a learning platform and the main features are- student registration, learning and exam dashboards and so on. All these features make this website more adaptable and user-friendly.

Contents

Chapter 1

1 Introduction

Education is the cornerstone of empowerment and growth, yet education systems often fail to accommodate the diverse needs of students with various disabilities .Our website endeavors to help the learning process by crafting inclusive assessments specifically tailored to each student's abilities, while also introducing a spectrum of learning methods—audio, text, video, and interactive activities that can help foster a learning environment.

1.1 Overview

The objective of the Ability Connect Website is to design an simple and adaptable website that helps the users in their learning process by tailoring different modes through which they can access education . Different modes of examinations, in-website voice navigation and other features helps website to be user friendly and create a learning environment for differently abled students .

Chapter 2

2 System Analysis

2.1 Purpose

The main purpose of the website owned by Don Davis, Adhithyan T.J, Leyon T.John ,Mithra prothasis is to make a user friendly website as a inclusive learning platform for diffrently abled students.

2.1.1 Existing System

In the existing system, there may be limited opportunities for disabled children to learn or develop their skills independently, especially at a young age. While schools and special education programs may offer support from teachers and other aides, there may be gaps in providing opportunities for self-directed learning and skill development outside of structured classroom settings. Additionally, parents of disabled children may face challenges in providing supplemental learning experiences or resources at home due to lack of time, knowledge, or access to suitable materials. Limitations of existing system : Self-directed learning and skill development outside of structured classroom settings and difficulties in accesing diffrent modes of learning according to disabilities.

2.1.2 Proposed System

The materials will be designed to be accessible to children with various disabilities. Users can specify any disabilities or special needs they have to ensure that the platform can customize them.The website will incorporate learning technologies to adjust the difficulty level and pacing of activities based on the student's performance and progress. This ensures that each student is appropriately challenged and supported throughout their learning journey. Proposed system aims to create an inclusive and empowering learning environment for disabled children. The website will provide feedback to students on their performance and progress, including scores, achievements, and areas for improvement. Parents or teachers may also have access to progress reports to monitor the student's development and provide additional support as needed. Other Features : A message system that sends notification to user mail about the new contents or exams assigned.

2.2 Problem definition

To know what the problem is and what the needs are before developing it. Here we design a website as a learning platform for differently abled students.

2.3 Feasibility Study

After the problem is clearly understood and solutions proposed, the next step is to conduct the feasibility study. Feasibility study is defined as evaluation or analysis of the potential impact of the proposed project or program. The objective is to determine whether the proposed system is feasible. There are three aspects of feasibility study to which the proposed system is subjected as discussed below.

2.3.1 Technical Feasibility

Technical feasibility assess whether the current technical resources are sufficient for the new system. We can upgrade the level of technology for supporting our website. We check whether the proposed system can be implemented in the present system without supporting the existing hardware.

2.3.2 Economical Feasibility

Economic feasibility determines whether the time and money are available to develop the system. There is no additional hardware used to develop the site, it is inexpensive to build.

2.3.3 Operational Feasibility

Operational feasibility determines if the human resources are available to operate the system once it has been installed. No extra training is needed to use this system. Anyone who has knowledge in internet in english language can easily use the system. The resources that are required to implement or install are already available with the organization.

Chapter 3

3 Software Requirement Specification

3.1 Purpose

The purpose of creating an educational website for differently-abled students is multifaceted and revolves around addressing the unique needs and challenges that these students may face. The educational website for differently-abled students is to create an inclusive, accessible, and supportive online learning environment that empowers individuals with diverse abilities to thrive academically and personally. The project aims to bridge gaps, break down barriers, and contribute to a more inclusive educational landscape.

3.2 Scope

The scope of the "Educational Website for Differently-Abled Students" project is a comprehensive initiative aimed at developing an inclusive and accessible online learning platform. The project will focus on catering to the unique needs of differently-abled students, educators, and administrators, providing a supportive and engaging environment for learning. It also aims to create a holistic and inclusive educational platform that not only meets the immediate needs of its users but also lays the groundwork for ongoing growth and enhancement.

3.3 Overall Description

The project for developing an "Educational Website for Differently-Abled Students" is a visionary initiative with the primary goal of creating an inclusive and accessible online learning platform. This comprehensive website is designed to cater to the diverse educational needs of students with varying abilities, ensuring they have equal access to quality learning resources and a supportive community. At the heart of the project is the creation of an adaptive learning environment. The website will feature a dynamic system that tailors educational content to the individual needs, preferences, and learning styles of differently-abled students.

3.3.1 Product Perspective

The "Educational Website for Differently Abled Students" involves considering how the website fits into the broader context of educational technology and the needs of its users.

3.3.2 Product Functionality

Through this website teachers can upload study materials and quizzes as a part of exam and teacher can see the result of each student. The students get notified

LARKSMITE LAB MANAGEMENT

PROJECT REPORT

Submitted By

NIKHIL ANTO C

Reg. No. CCAVBCA024

For the award of the Degree of

Bachelor of Computer Application (BCA)

(University of Calicut)

under the guidance of

Ms. Soumya P S

Assistant Professor

BACHELOR OF COMPUTER APPLICATION
DEPARTMENT OF COMPUTER SCIENCE
CHRIST COLLEGE
IRINJALAKUDA, KERALA
2021-2024

DEPARTMENT OF COMPUTER SCIENCE

CHRIST COLLEGE (AUTONOMOUS)

IRINJALAKUDA

CERTIFICATE

This is to certify that the project report entitled ***"Larksmite Lab Management"*** *is a bonafide record of the project work done by* ***Nikhil Anto C*** *in partial fulfillment of the requirement for the sixth semester of* ***Bachelor of Computer Application*** *in Department of Computer Science of* ***CHRIST COLLEGE (AUTONOMOUS) IRINJALAKUDA***

.

Ms. Soumya P S
Assistant Professor
Internal Guide

Ms. Sini Thomas
Head of the Department
Computer Science

EXTERNAL EXAMINER

INTERNAL EXAMINER

DECLARATION

I hereby declare that this project work **"LARKSMITE LAB MANAGEMENT"** submitted to Christ College (Autonomous) Irinjalakuda, affiliated to Calicut University in partial fulfillment of the requirement for the award of the Bachelor of Computer Application, is a record of original work done by me, under the guidance of Ms. SOUMYA P S, Assistant Professor, Department of Computer Science.

Place: Irinjalakuda NIKHIL ANTO C

ACKNOWLEDGEMENT

First and foremost we wish to thank Lord almighty for his providence and for being the guiding light throughout the project. We take this opportunity to express our gratitude to our beloved class teacher as well as project guide, Ms. SOUMYA P S, who has been hugely supportive throughout the course of this project. We wish to express our sincere gratitude to our head of department, Ms. SINI THOMAS for giving us all the required facilities for our project. We are thankful for their aspiring guidance and valuable advice during the project work. We would like to take this opportunity to specially thank all other faculty members for their constant and continuous motivation. Finally we would like to thank our family and friends for giving valuable advice and moral support throughout our project.

ABSTRACT

Larksmite Lab Management is an innovative software introduced to manage computer labs. The software is enriched with multiple features, including - status monitoring, process monitoring, screen monitoring, power control, message passing and so on. The client side of the software automatically runs in the background in all computers in the lab. All these features make this software powerful and useful in computer labs.

Contents

Chapter 1

1 Introduction

Larksmite is a sophisticated computer lab management software designed to provide administrators with seamless control and oversight of connected devices. Built using Flutter and Dart, the software offers a bird's eye view of the lab's layout, enabling easy device identification. With features such as screen sharing, power controls, and process monitoring, administrators can efficiently manage resources and troubleshoot issues remotely. The project employs RSA public-key cryptography for secure communication and utilizes SQLite for data storage. Larksmite aims to simplify computer lab administration, offering a centralized solution for monitoring, controlling, and maintaining an optimal computing environment.

1.1 Overview

Larksmite is a streamlined computer lab management solution, prioritizing user-friendly interfaces for efficient device identification. With features such as screen sharing, power controls, and detailed reporting, the platform simplifies administration tasks. Utilizing Flutter and Dart, it ensures responsiveness, while employing RSA public-key cryptography and SQLite for secure communication and data storage. Larksmite's core objective is to centralize control and optimize the management of multiple devices within a network.

Chapter 2

2 System Analysis

2.1 Purpose

The purpose of Larksmite is to streamline computer lab management, offering a centralized solution for efficient device control and monitoring. This project aims to simplify administration tasks, enhance accessibility, and optimize the management of multiple devices within a network.

2.1.1 Existing System

Existing computer lab management systems have limitations in scalability, security, and user interface. Most of them also only work on Microsoft Windows, prompting the development of a cross-platform software. This new system aims to enhance performance, address security concerns, and offer a more user-friendly solution for streamlined device control and monitoring.

2.1.2 Proposed System

Larksmite is envisioned as a revolutionary solution for computer lab management, aiming to address the limitations of the existing system. The proposed system prioritizes a centralized and user-friendly approach to streamline device control and monitoring. Larksmite seeks to enhance efficiency, security, and user experience in computer lab administration, providing administrators with an optimized tool for seamless device management

2.2 Problem Definition

The existing computer lab management system faces challenges in scalability, security, and user interface, hindering efficient device control and monitoring. These limitations necessitate the development of Larksmite.

2.3 Feasibility Study

After the problem is clearly understood and solutions proposed, the next step is to conduct the feasibility study. Feasibility study is defined as evaluation or analysis of the potential impact of the proposed project or program. The objective is to determine whether the proposed system is feasible. There are three aspects of feasibility study to which the proposed system is subjected as discussed below.

2.3.1 Technical Feasibility

Technical feasibility assesses whether the current technical resources are sufficient for the new system. The selected combination of Flutter and Dart offers a robust framework for building a responsive and visually appealing user interface, which is cross-platform by default. This ensures a wide array of compatibility.

2.3.2 Economical Feasibility

Economic feasibility determines whether time and money are available to develop the system. There is no additional hardware needed to develop the software, neither does it need any extra hardware to run.

2.3.3 Operational Feasibility

Operational feasibility determines if the human resources are available to operate the system once it has been installed. No extra training is needed to use this system. Anyone who has basic computer knowledge can easily use the system. The resources that are required to implement or install are already available with the organization.

Chapter 3

3 Software Requirement Specification

3.1 Purpose

The purpose of the software specification is to precisely outline the requirements of Larksmite, serving as a blueprint for its development. This document articulates the specific objectives the software aims to achieve, detailing user interactions, system behaviors, and performance expectations. It is primarily intended for the computer lab administrator for managing and understanding the proposed system.

3.2 Scope

Our project scope encompasses the development of Larksmite, a computer lab management software. It includes features such as computer lab layout visualization, real-time screen monitoring, power controls, and detailed reporting to optimize the efficiency of computer lab administration.

3.3 Overall Description

This section gives an overview of our software - Larksmite. The software offers an intuitive interface, allowing administrators a bird's eye view of the lab layout for streamlined device identification. Incorporating features such as real-time screen monitoring, power controls, and detailed reporting, Larksmite ensures efficient and secure administration.

3.3.1 Product Perspective

LarkSmite is mainly used for managing a computer lab in any institutions. It can be used to monitor each computer screens, orchestrate power options and generate reports.

3.3.2 Product Functionality

Tasks such as device identification, real-time screen monitoring, power controls, process monitoring and detailed reporting are various functions of LarkSmite

3.3.3 Users and Characteristics

There are two types of users within the LarkSmite system: administrators and clients. Administrators, tasked with overseeing computer labs, require access to comprehensive management features, while clients, interacting with the client computer system, remain oblivious to the existence of monitoring tool which always runs in the background.

3.4 Specific Requirements

3.4.1 Hardware Requirements

- System: IBM-Compatible PC
- Processor: Pentium IV or above
- Speed: Above 1GHz
- RAM capacity: 512 MB
- Hard Dsk drive: 40 GB
- Keyboard: Standard
- Mouse: Standard
- Monitor: SVGA Color

3.4.2 Software Requirements

- Operating System: Ubuntu or Gnome DE based Linux
- Languages used: Dart
- Database: SQLite
- Technologies used: Flutter, Shelf/Dart

3.5 Functional Requirements

It contains two main modules.

- Teacher
- Student

Teacher

In LarkSmite, teachers have the functional requirement to use the system as administrators to oversee and manage computer labs. This includes tasks such as device identification, real-time screen monitoring, and utilizing power controls, providing them with the necessary tools for effective supervision and control in an educational setting.

Student

Students in LarkSmite have the functional requirements focused on interacting with their specified client computer, performing their assigned lab task.

3.6 Non Functional Requirements

Non-functional requirements define the overall qualities or attributes of the resulting system. Non-functional system place restrictions on the product being developed, the developed process, and specify external constraints that the product must meet. Examples of non-functional requirements include security, usability, reliability and performance requirements. The principal non - functional constraints which are relevant to critical systems:

- Performance
- Security
- Usability

Performance

Performance requirements concern the speed of operation of a system. Types of performance requirements:

- Response requirements (how quickly the system reacts to a user input).
- Throughput requirements (how much the system can accomplish within a specified amount of time).
- Availability requirements (is the system available for service when requested by end users). The speed of operation of this system is adequate for the requirements.

Reliability

- Reliability is the ability of a system to perform its required function under stated conditions for a specified period of time.
- Constraints on the runtime behavior of the system. This system is reliable because its functionalities can be done on the required conditions.

Safety

Safety requirements refer to the specific criteria, conditions, and constraints that must be met to ensure the safe operation of a software system.

Usability

Usability is the ease with which a user can learn to operate, prepare inputs for, and interpret outputs of system or components. Usability requirements include:

- Understandable error messages.
- Well-formed user interfaces.

3.7 Interface Requirements

3.7.1 Hardware Interfaces

The system must be run in a LAN network, all client devices shall be required to connect to the same LAN network, for example, a WIFI access point or an Ethernet LAN network.

3.7.2 Software Interfaces

Software interface required for the working of the project is the appropriate operating system.

3.7.3 Communication Interfaces

The clients communicate using HTTPS and Secure Websocket.

3.8 Security Requirements

- Only authorized clients are allowed to connect.
- All communication between administrators and client devices is encrypted.
- Unauthorized devices cannot act as a server and clients will not connect to imposter servers

3.9 Platform Used

Ubuntu is a Linux distribution based on Debian and is free and open-source, making it a preferred choice for many institutions. It is developed by Canonical. The default distribution uses the Gnome Desktop Environment, with official support for other desktop environments like KDE, XFCE and Mate. The latest stable version is Ubuntu 23.10 (Mantic Minotaur) and the latest LTS (Long-Term Support) version is Ubuntu 22.04 LTS (Jammy Jellyfish). LarkSmite is designed to operate seamlessly within Linux environments, leveraging the robust features and stability offered by this platform.

3.10 Technologies Used

Dart

Dart serves as the cornerstone programming language in LarkSmite, providing the foundation for implementing the software's logic and functionality. Known for its simplicity and efficiency, Dart facilitates the development of a responsive and intuitive system. With features like strong typing and Just-In-Time (JIT) compilation, Dart enhances the codebase, contributing to the overall reliability and performance of LarkSmite's computer lab management solution.

Flutter

Flutter is an open-source UI (User Interface) software development kit (SDK) created by Google. It allows developers to build native applications for Android, Linux, Windows, iOS as well as macOS platforms using a single codebase. Flutter utilizes the Dart programming language. Flutter offers a hot reload feature, allowing developers to see changes in real-time as they modify the code.

SQLite

LarkSmite has opted for SQLite due to its lightweight and embedded nature. The choice of SQLite aligns with the project's goals of simplicity and seamless integration within the context of computer lab management, providing an efficient and self-contained solution for data storage and retrieval.

Chapter 4

4 Design Document

4.1 Purpose

The purpose of LarkSmite is to revolutionize computer lab management by offering administrators a centralized and user-friendly platform. Through intuitive interfaces and features like real-time screen monitoring, power controls, and detailed reporting, LarkSmite aims to streamline tasks for administrators, teachers, and students. Developed with Flutter in Dart language, the system prioritizes responsiveness and scalability, aiming to enhance the efficiency of computer lab administration within educational environments.

Currently built for Linux environment, LarkSmite seeks to optimize the computer lab management experience. With a focus on security through RSA public-key cryptography and SSL, the system aims to provide a robust, secure, and accessible solution for overseeing and controlling multiple devices within educational settings.

4.2 Scope

Our project, LarkSmite, primarily encompasses the development of a computer lab management software tailored for Linux environments. It includes features for teachers, and students, focusing on efficient device control, real-time screen monitoring, and detailed reporting within the educational context

4.3 Overview

The purpose of this document is to provide a comprehensive guide to LarkSmite, a specialized computer lab management software designed for Linux environments. It outlines the features, functionalities, and the technology stack utilized in the development of the system. By detailing the project's scope, objectives, and key components, this document aims to offer a clear understanding of LarkSmite's role in revolutionizing computer lab administration within educational institutions.

4.4 Data Design

Database is the storehouses of data used in the software systems. The data is stored in tables inside the database. Several tables are created for the manipulation of the data for the system. Two essential settings for a database are

- Primary Key- The field that is unique for all the record occurrence.
- Foreign Key- The field used to set relation between tables.

Normalization

- Normalization is the process of organizing the data in the database.
- Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate the undesirable characteristics like Insertion, Update and Deletion Anomalies.
- Normalization divides the larger table into the smaller table and links them using relationship.
- The normal form is used to reduce redundancy from the database table.

Tables

Server Asymmetric Key

Name	DataType	Constraints	Description
key_id	INTEGER(5)	Primary Key	ID of public-private key pair
public_key	BLOB	Not Null	Public key
private_key	BLOB	Not Null	Private key

Certificates

Name	DataType	Constraints	Description
cert_id	INTEGER(5)	Primary Key	ID of SSL certificate
public_cert	BLOB	Not Null	Public key
private_key	BLOB	Not Null	Private key

IP Address

Name	DataType	Constraints	Description
address	TEXT(20)	Not Null	IP Address of server
cert_id	INTEGER(5)	Foreign Key	Corresponding certificate cert_id

Shared User Key

Name	DataType	Constraints	Description
name	VARCHAR(50)	Not Null	Name of client
shared_key	TEXT(6)	Not Null	Authorization key of client

Chapter 5

5 Development of the System

The software is split into two - server side and client side. In the server side, the project is divided based on major features like networking, datastore, user interface. Each feature is further sub-divided into presentation, domain and data layer to avoid unwanted coupling and increase cohesion. This ensures that each submodule is modular, that is, it can be replaced without affecting the other parts.

In the client side, the project is divided into two major parts - platform dependent and platform independent code. The platform independent code implements features which are not dependent on the platform in which it will be run on. The platform dependent code is dependent on the platform, thus it is different on different operating systems and/or versions. This separation is designed using interfaces, which defines common functions which should be implemented by all platform dependent implementors. By employing such a split, we can achieve optimal code sharing and enhance efficiency.

Chapter 6

6 System Testing

Testing is the penultimate step of software development. An elaborate suite of test data is prepared and the system is tested using this test data. Corrections are made using the results of failed tests and any changes are immediately tested against this test suite, to detect regression. System testing is aimed at ensuring the system works accurately before the live operation commences.

6.1 Test Plan

6.1.1 Scope

This test plan will cover the following testing activities as identified in the testing strategy.

White Box Testing: White box testing is a testing technique, that examines the program structure and derives test data from the program logic/code. In white box testing, the UI is bypassed. Testing is done directly at the code level and the results are compared against specifications. It reveals errors in "hidden" code.
Black Box Testing: Black - box testing is a method of software testing that examines the functionality of an application based on the specifications. Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end-user would.
Unit Testing: The module includes four main components to undergo unit testing. It checks the syntax error, logic error and validity of the program correctness. This test will be performed by the developers.
Integration Testing: After all components pass the unit testing, they get to be tested whether they work correctly when they are running concurrently and communicating to each other. Integration tests considers the entire subsystem and ensure that a set of components play nicely together.
Internal Data Testing: We will test the validity of the data before it enters the database to avoid any problems that may be faced in the database. We will test the encryption of the personal information of all the users along with the admin user names and passwords to ensure maximum security of the admin user.

6.1.2 Software Risk Issues

In this section, the plan is to test the risk involved in critical issues such as:

- Difficulty in setting up the dependencies
- Establishing proper network security

- Ensuring data transmission security

6.1.3 Features to be Tested

- Test whether correct admin user name and password allows you to login.
- Test whether invalid admin user name and password prevents you from login.
- Test whether admin can open valid layout files.
- Test whether server software will not crash when opening invalid layout file
- Test whether server software will start service advertising and start server
- Test whether server software is able to generate SSL certificate and RSA key pair
- Test whether server software is able to perform client functions and display results
- Test whether client software continuosly searches for server in network
- Test whether client connects to server with valid client details
- Test whether client automatically connects to server when details are saved
- Test whether client responds to server requests
- Test whether client detects a fake server and doesn't connect
- Test whether client restarts from potential errors and keep running

6.2 Test Consolidation

6.2.1 Test Item

The items or features to be tested in the test cases are included in the document. Each and every input is tested. The present condition of the system is tested. It is checked to make sure that environment is ready for the application to work.

6.2.2 Input Specifications

These are the inputs required to execute the test case. All required inputs including data elements and values are given below.

Data	Values
Username	A valid username
Password	Strong combination of characters and numbers

Chapter 7

7 System Implementation and Maintenance

7.1 Implementation

System implementation is the conversion of a proposed system into an operational one which involves writing code, training clients and installing softwares. User training is crucial for helping them understand the versatily and helpfulness of the new system.

Software maintenance becomes necessary to ensure the system continues to operate satisfactorily in response to changes in the user's environment. Maintenance activities often involve making minor enhancements or corrections to address issues that may arise during system operations.

In the implementation phase, the team builds the components either from scratch or by composition. Given the architecture document from the design phase and the requirement document from the analysis phase, the team should build exactly what has been requested, though there may be still room for innovation and flexibility. For example, a component may be narrowly designed for this particular system, or the component may be made more general to satisfy a reusability

- Careful planning
- Investigation of system and constraints
- Design the methods to achieve changeover.
- Training the staff in the changed phase.
- Evaluation of changeover method.
- The method of implementation and time scale to be adopted are found out initially.

7.2 Maintenance

This phase occurs as a result of deploying the whole system at the end users organization. They will perform the beta testing and inform to the developers about any required modification to the application. The customer records all the problems that are encountered during the beta testing and reports these to the developer at regular intervals.

7.2.1 Corrective Maintenance

Even with the best quality assurance activities, it is likely that the customer will uncover defects in the software. Corrective maintenance changes the software to correct the defects. Corrective Maintenance activity may consist of repair,

restoration or replacement of equipment. This activity will be result of a regular inspection, which identifies the failure in time for corrective maintenance to be planned and scheduled, then performed during a routine maintenance shutdown.

7.2.2 Adaptive Maintenance

Over time, the original environment (CPU, operating system, business rules, external product characteristics) for which the software was developed is likely to change. Adaptive maintenance results in modification to the software to accommodate changes to its external environment.

7.2.3 Enhanced Maintenance

As software is used, the customer/user will recognize additional functions that will provide added benefits. Perfect maintenance extends the software beyond its original functional requirements.

7.2.4 Preventive Maintenance

Computer software deteriorates due to change, and because of this, preventive maintenance often called software re-engineering, must be conducted to enable the software to serve the needs of its end users. Preventive maintenance makes changes to computer programs so that they can be more easily corrected, adapted and enhanced.

Chapter 8

8 Conclusion and Future Scope

8.1 Conclusion

The objective of LarkSmite is to streamline and enhance the efficiency of managing computer labs in educational institutions or any organization that operates computer labs. It enables the admin/teacher to manage all the computers through a single admin computer. The admin is able to monitor the screens of each computer, view running processes, see basic information about the computer, send messages, perform power controls like sleep, restart, shutdown remotely and generate PDF reports with screenshots. The student's computer behaves the same way, with the client software running in the background.

8.2 Future Scope

- Allow sharing and showing teacher's computer screen on all clients
- Allow remote access similar to VNC, RDP and TeamViewer.
- Add file sharing
- Perform seamless upgrade of software

Appendix

A Activity Diagram

Activity Diagrams are used to illustrate the flow of control in a system and refer to the steps involved in the execution of a use case. It is a type of behavioral diagram and we can depict both sequential processing and concurrent processing of activities using an activity diagram ie an activity diagram focuses on the condition of flow and the sequence in which it happens.
Some Activity Diagram charting forms:

A.1 Initial State

The initial state marks the entry point and the state of the system before the application is opened

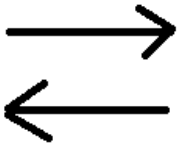
A.2 Final State

The state which the system reaches when a particular process or activity ends is known as a Final State or End State.

A.3 Activity State

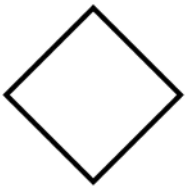
An activity represents execution of an action on objects or by objects. Any action or event that takes place is represented using an activity.

A.4 Control Flow



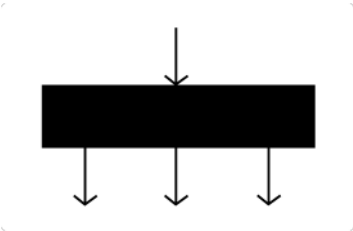
They are used to show the transition from one activity state to another activity state.

A.5 Decision Node



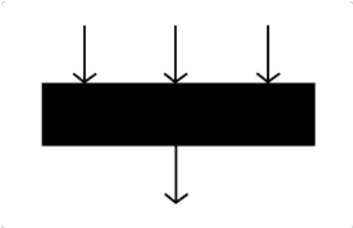
When we need to make a decision before deciding the flow of control, we use the decision node. The outgoing arrows from the decision node can be labelled with conditions.

A.6 Fork



Fork nodes are used to support concurrent activities. We use a fork node when multiple activities get executed concurrently.

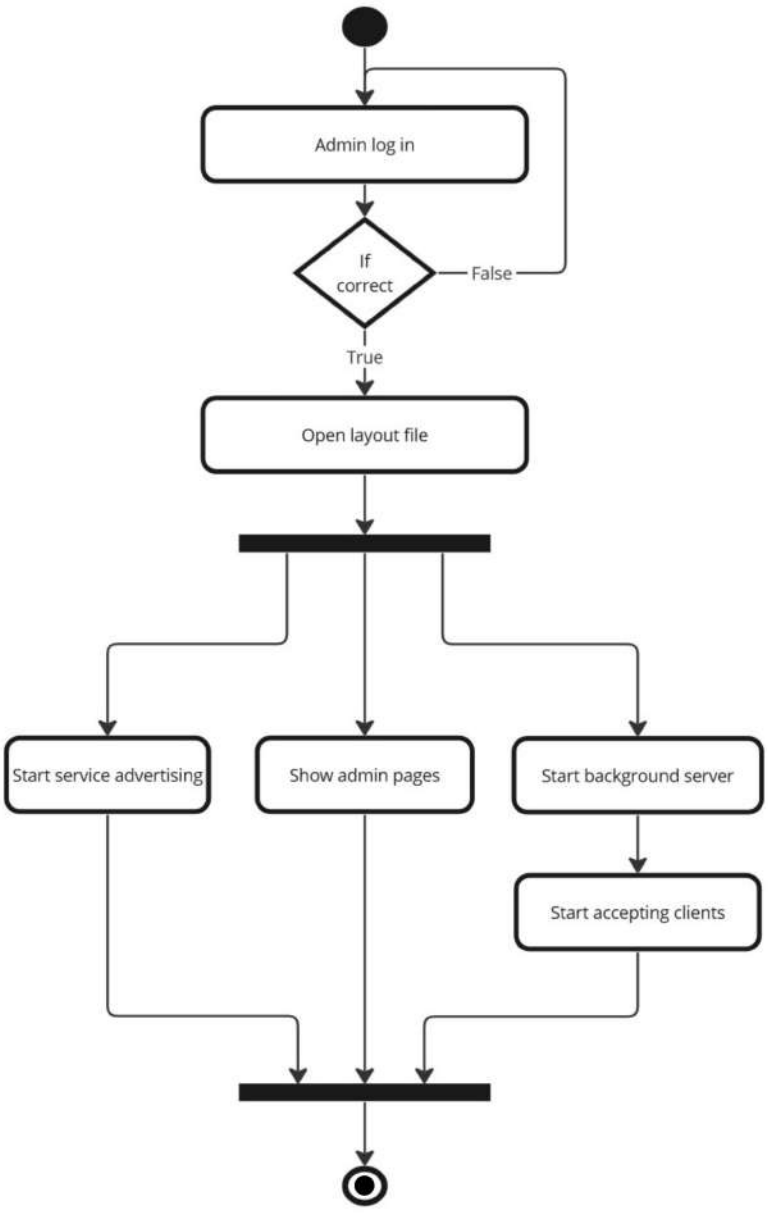
A.7 Join



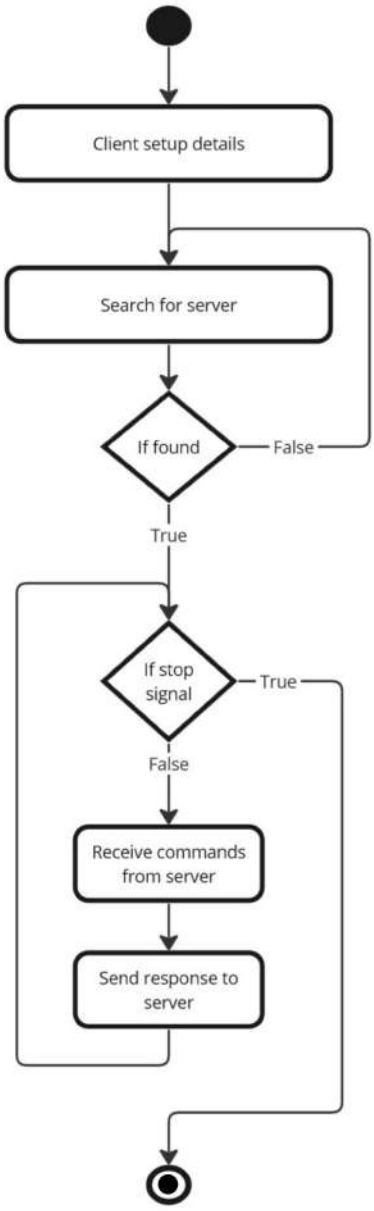
Join nodes are used to support concurrent activities converging into one.

A.8 Activity Diagram

A.8.1 Server Side



A.8.2 Client Side



B Sequence Diagram

Sequence diagrams show elements as they interact over time and they are organized according to object (horizontally) and time (vertically). They capture the interaction between objects in the context of a collaboration. Sequence diagrams are time focused and they show the order of the interaction visually by using the vertical axis of the diagram to represent time.

B.1 Actor

An actor represents a type of role where it interacts with the system and its objects. An actor is always outside the scope of the system. We use actors to depict various roles including human users and other external subjects

B.2 Lifeline

A lifeline is a named element which depicts an individual participant in a sequence diagram. A lifeline always portrays an object internal to the system.

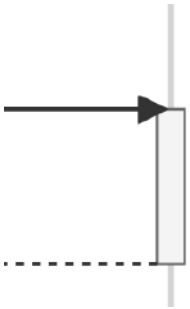
B.3 Messages

Communication between objects is depicted using messages. The messages appear in a sequential order on the lifeline.

B.4 Response Messages

Reply messages are used to show the message being sent from the receiver to the sender.

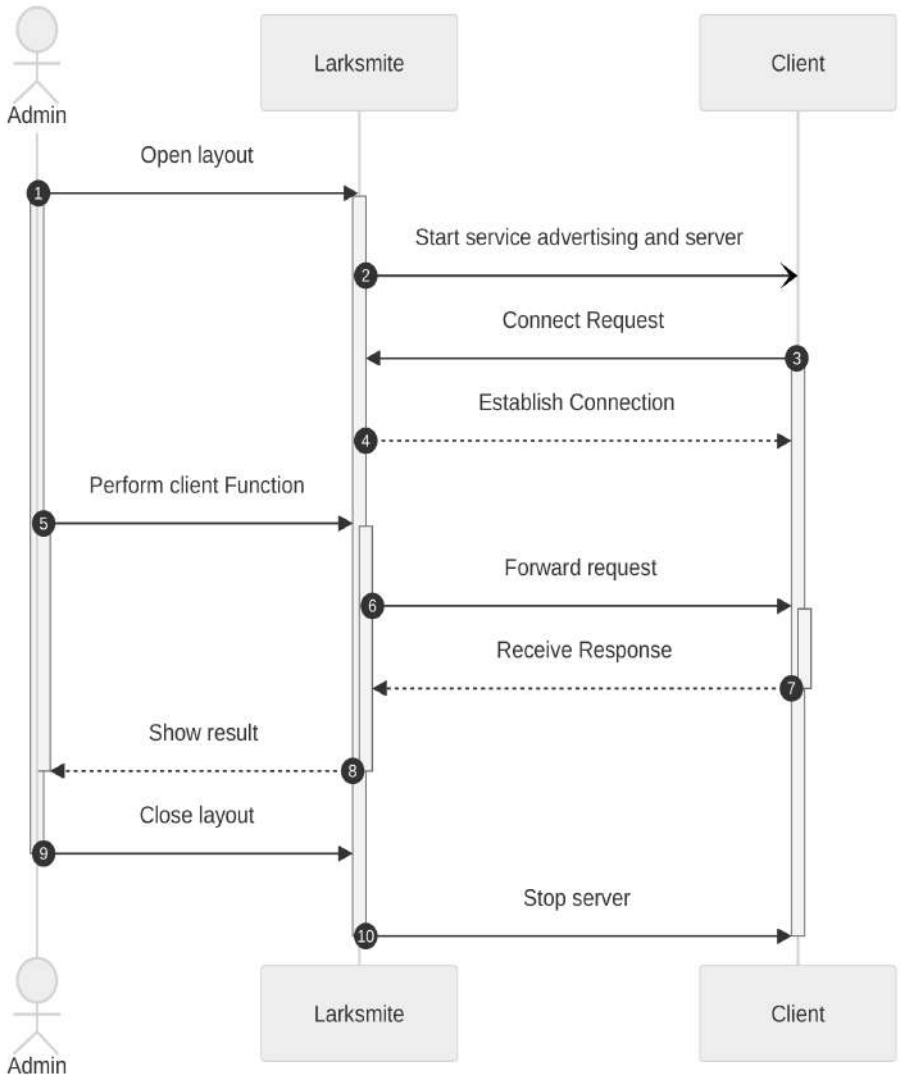
B.5 Activation



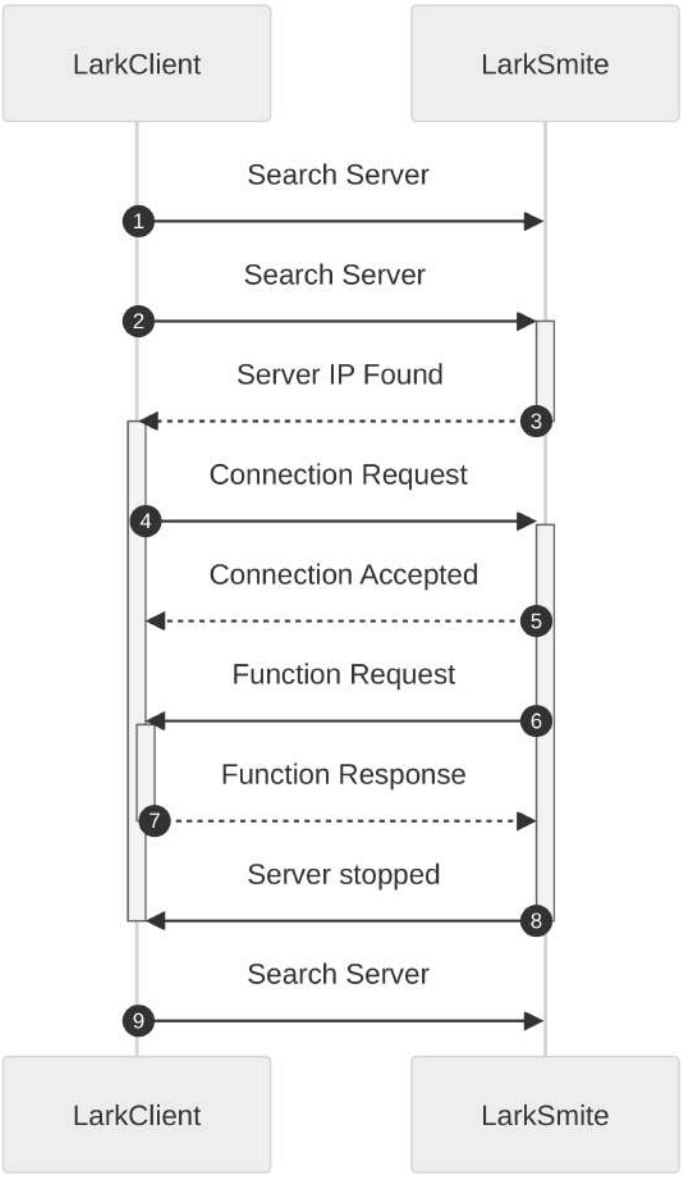
A thin rectangle on a lifeline represents the period during which an element is performing an operation. The top and the bottom of the rectangle is aligned with the initiation and the completion time respectively.

B.6 Sequence Diagram

B.6.1 Server Side

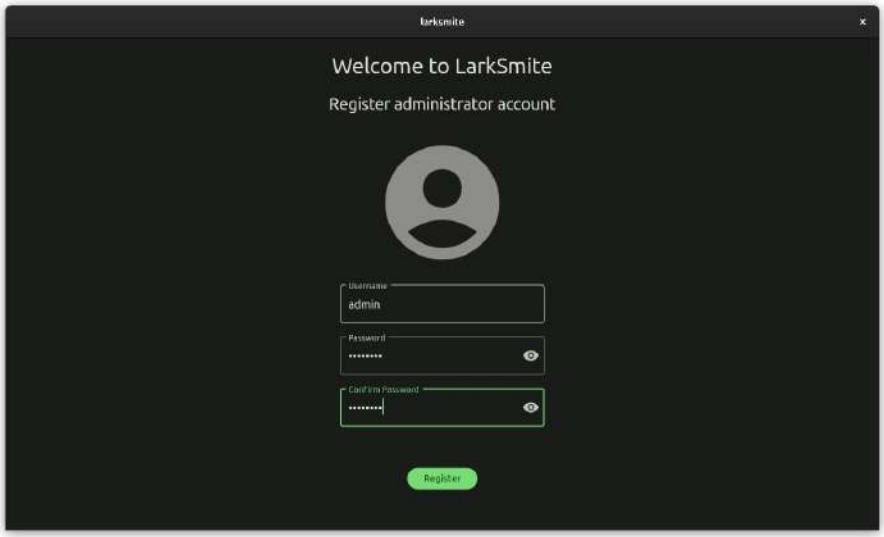


B.6.2 Client Side

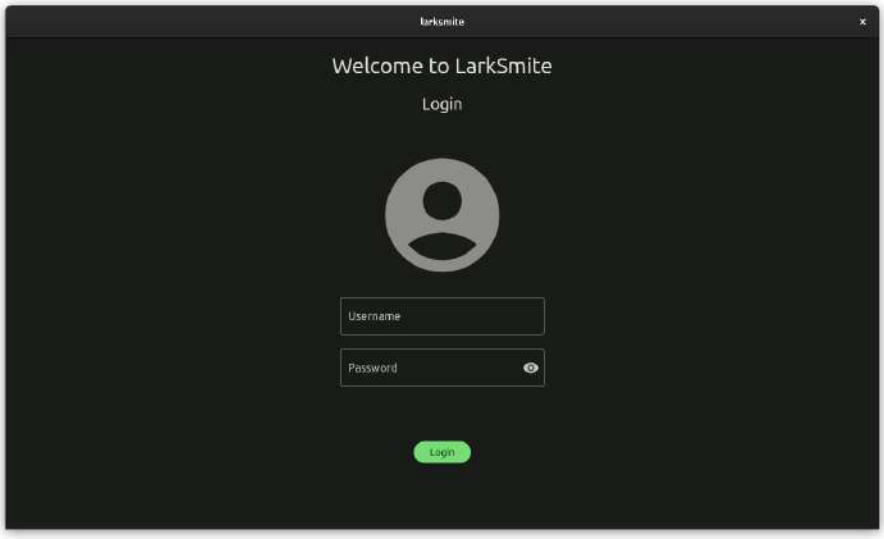


C User Interfaces

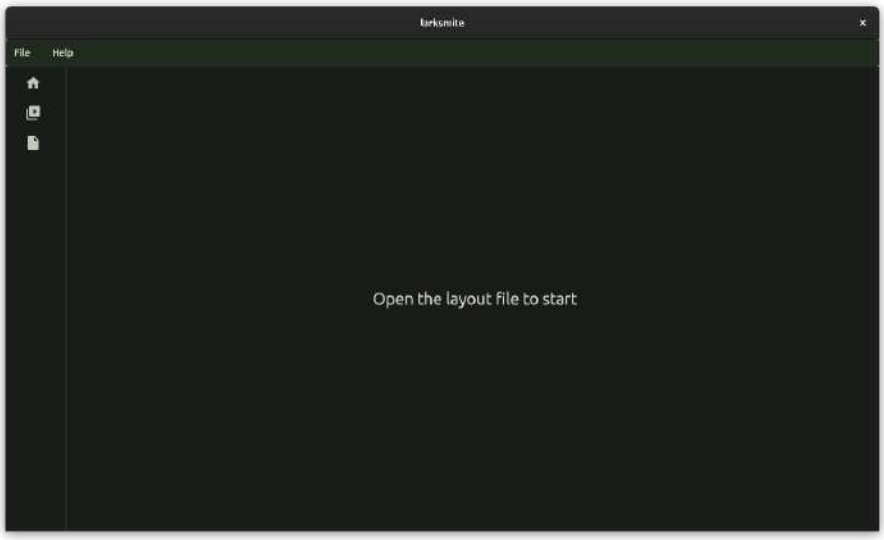
C.1 Admin Registration



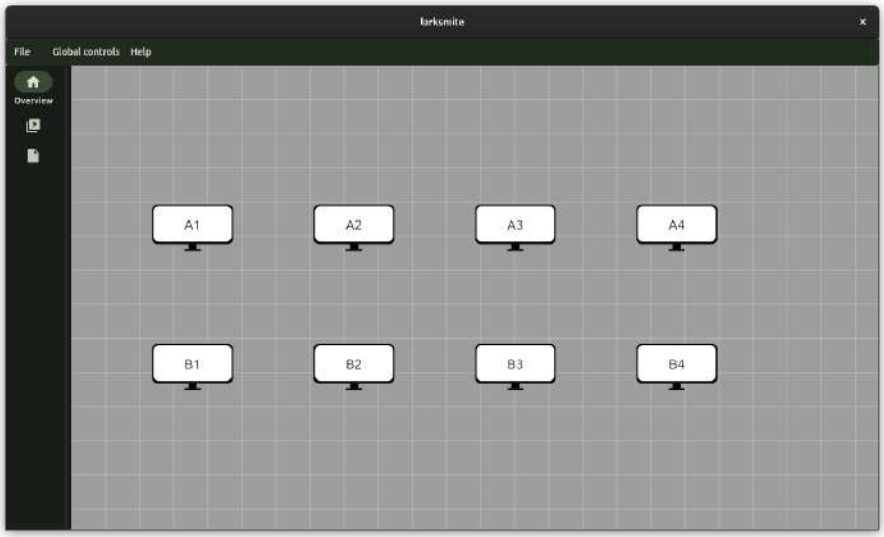
C.2 Admin Login



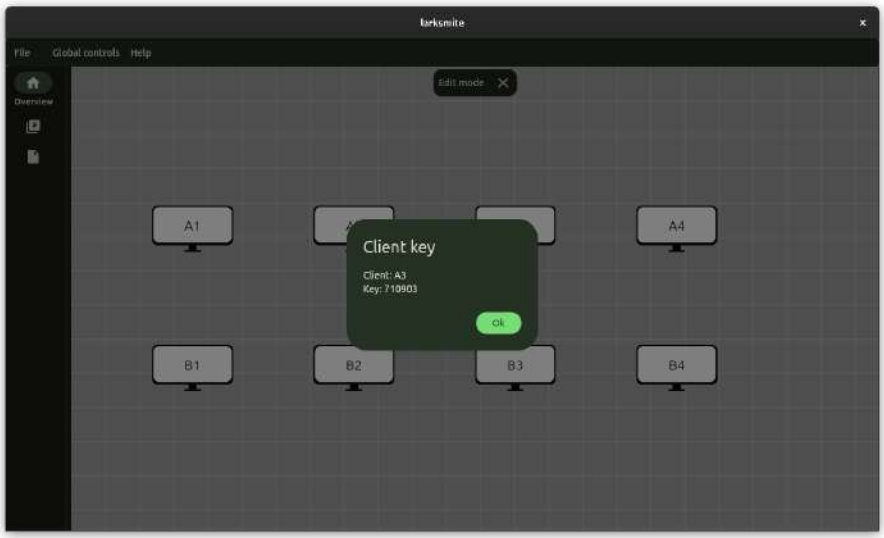
C.3 Home Page



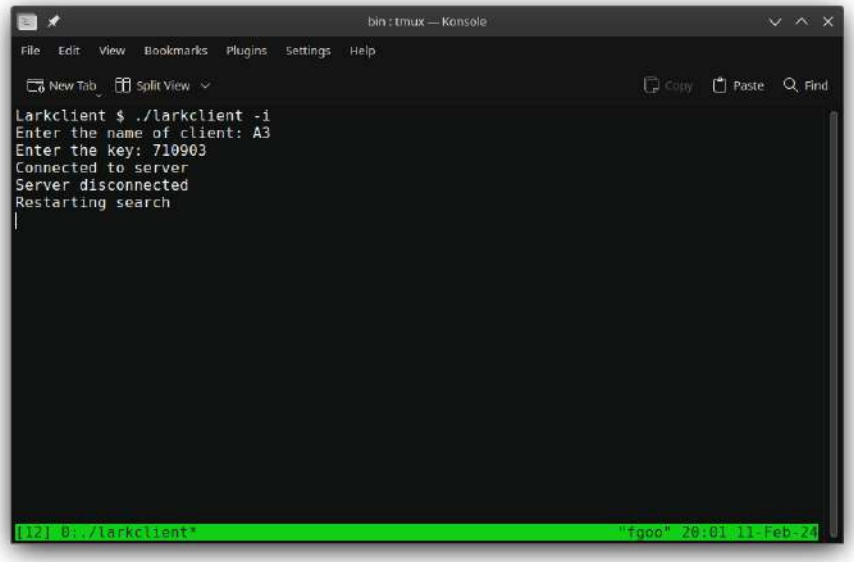
C.4 Layout File Opened



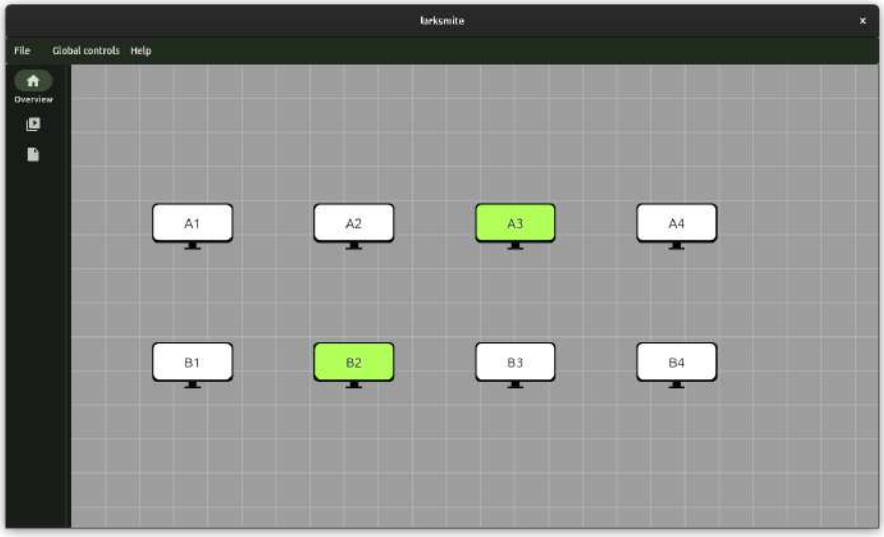
C.5 Adding Clients



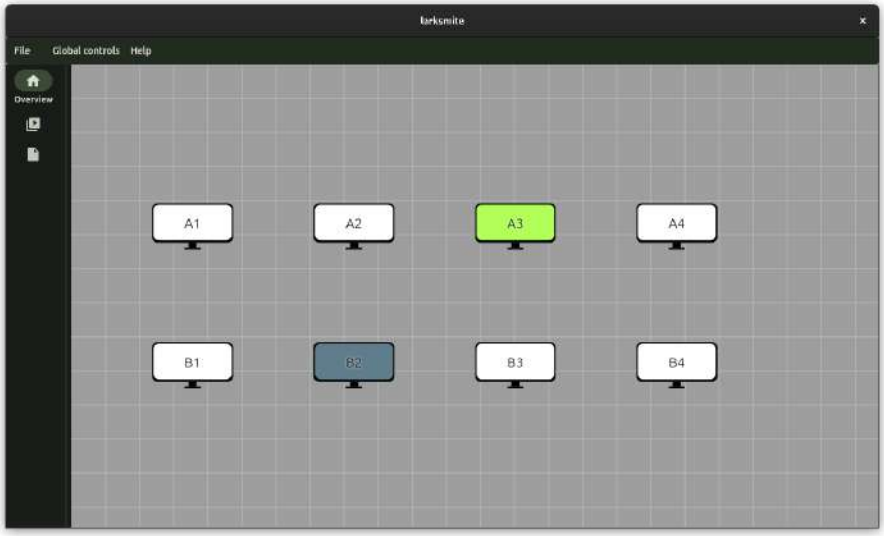
C.6 Settings up Larkclient



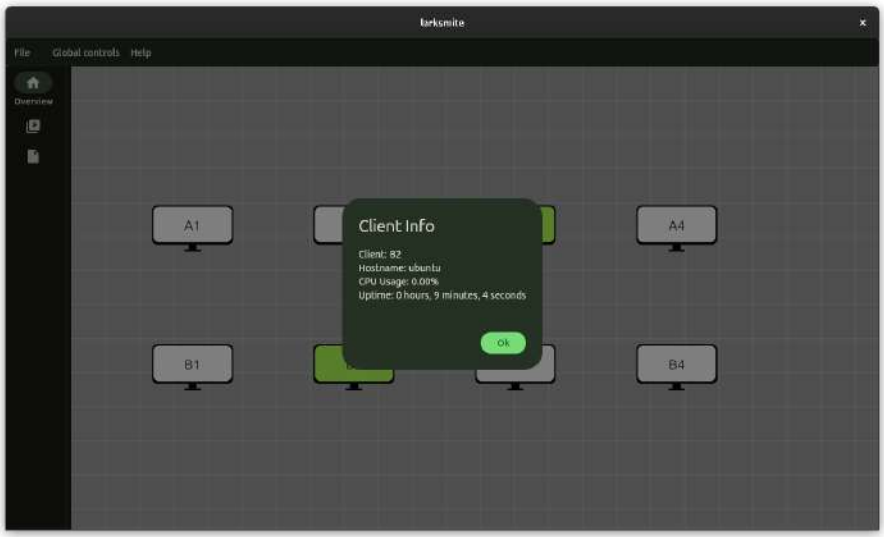
C.7 Client Realtime Status



C.8 Color coded status

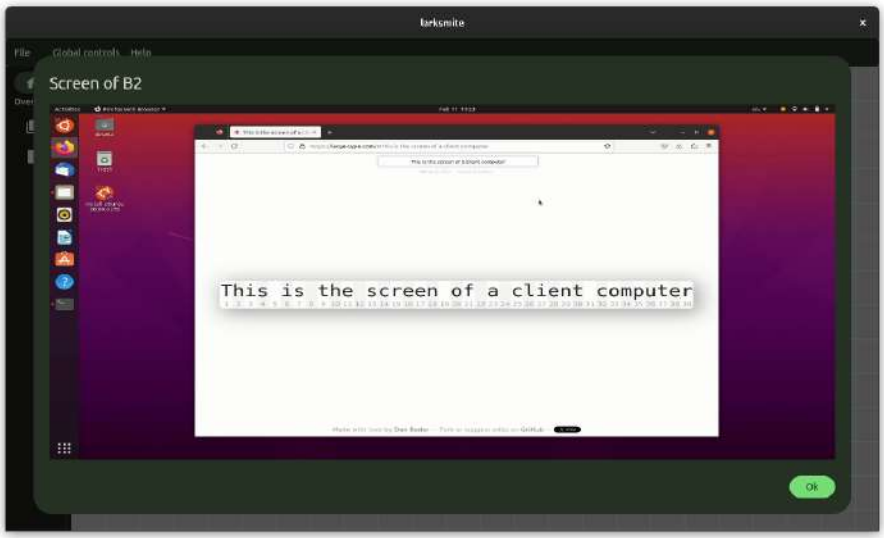


C.9 Client Info

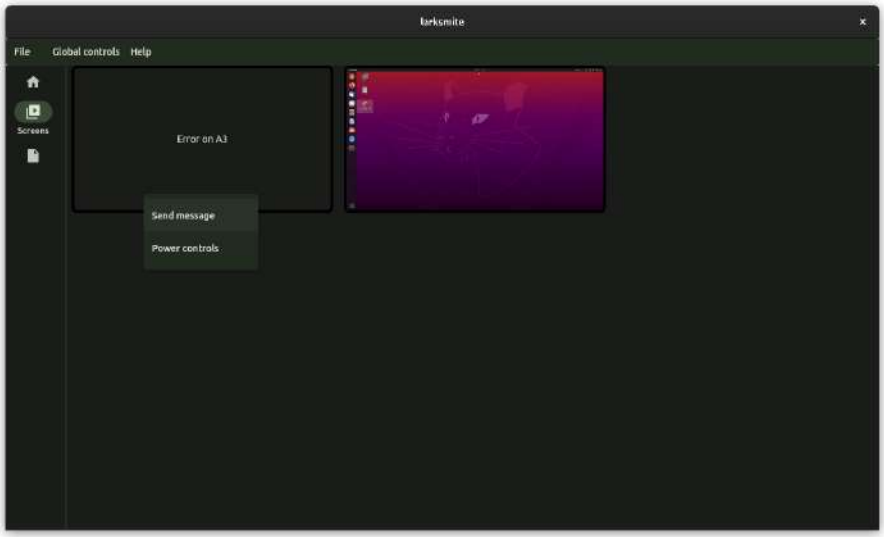


C.10 Client Processes

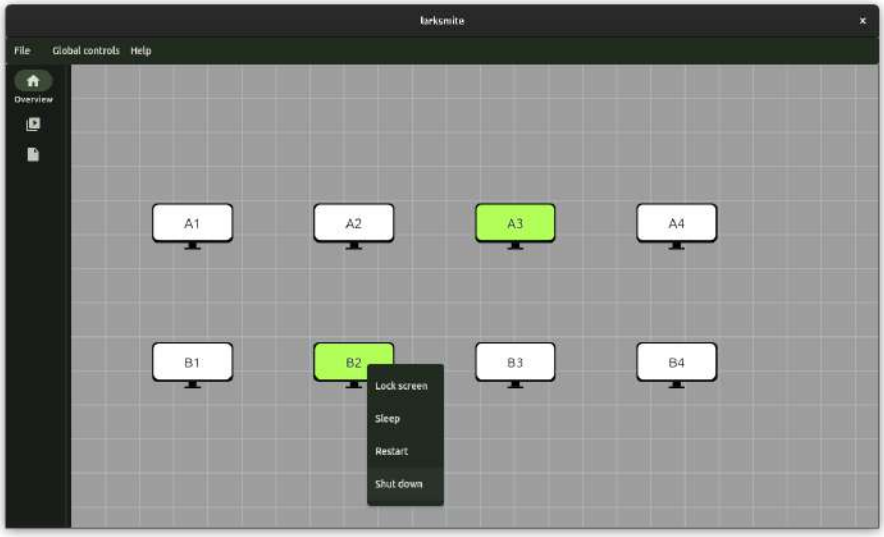
C.11 Single Screen Stream



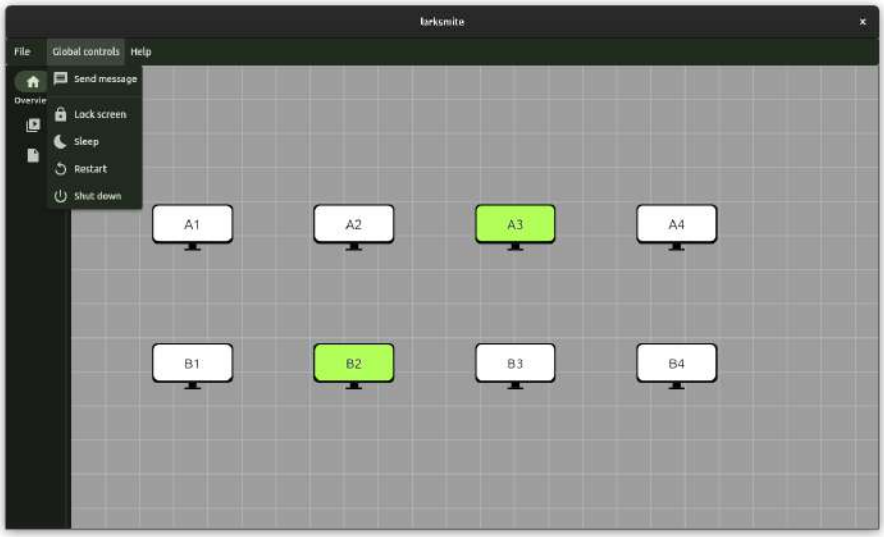
C.12 All Screen Stream



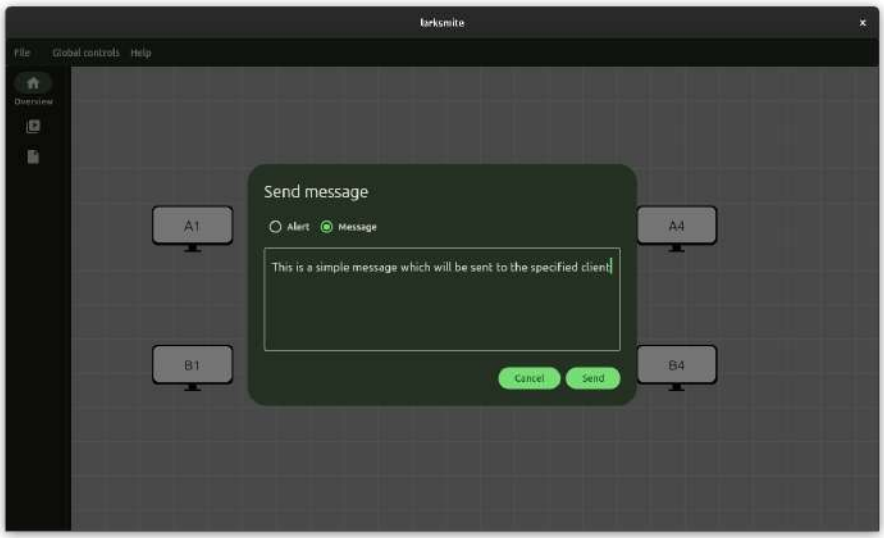
C.13 Power Control



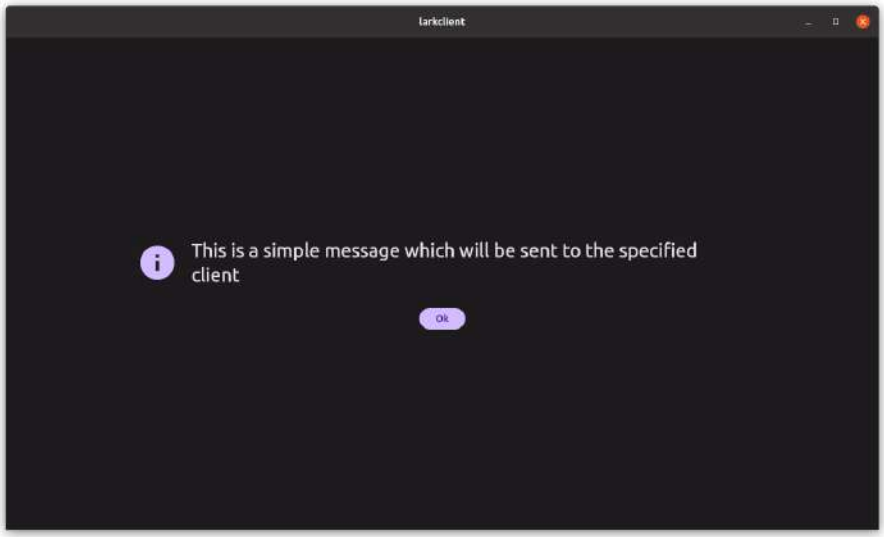
C.14 Global Control



C.15 Sending Message



C.16 Message on Client Side



D Code

main.dart

```
import 'package:flutter_bloc/flutter_bloc.dart';
import 'package:flutter/material.dart';
// import 'package:yaru/yaru.dart';
import 'core/injection_container.dart' as core_di;
import 'features/login/login_injection_container.dart' as login_
    di;
import 'features/dashboard/dashboard_injection_container.dart' as
     dashboard_di;
import 'features/networking/networking_injection_container.dart'
    as networking_di;
import 'features/datastore/datastore_injection_container.dart' as
     datastore_di;
import 'core/cubit/core_data_cubit.dart';
import 'core/router.dart';
import 'core/logging.dart' as logging;

void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await logging.initLogger();
  await core_di.init();
  await datastore_di.init();
  await login_di.init();
  await networking_di.init();
  await dashboard_di.init();
  runApp(const LarkSmiteApp());
}

class LarkSmiteApp extends StatelessWidget {
  const LarkSmiteApp({super.key});
  @override
  Widget build(BuildContext context) {
    return BlocProvider(
      create: (context) => CoreDataCubit(firstStartChecker: core_
          di.sl()),
      child: MaterialApp.router(
        title: 'LarkSmite',
        // theme: yaruLight,
        // darkTheme: yaruDark,
        //
        debugShowCheckedModeBanner: false,
        routerConfig: LarkSmiteRouter.router,
        darkTheme: ThemeData(
```

```
          brightness: Brightness.dark,
          // colorScheme: ColorScheme.fromSeed(seedColor: Colors.
              amber),
          colorSchemeSeed: Colors.green,
          useMaterial3: true,
        ),
      ),
    );
  }
}
```

dashboard.dart

```
import 'package:flutter/material.dart';
import 'package:larksmite/features/dashboard/presentation/cubit/
    app_menu_bar_cubit.dart';
import 'package:larksmite/features/dashboard/presentation/cubit/
    report_view_cubit.dart';
import 'package:larksmite/features/dashboard/presentation/cubit/
    screen_view_cubit.dart';
import 'dashboard_view_frame.dart';
import '../widget/navigation_rail_side.dart';
import '../cubit/birds_eye_view_cubit.dart';
import '../cubit/dashboard_cubit.dart';
import 'package:flutter_bloc/flutter_bloc.dart';
import '../../dashboard_injection_container.dart' as dashboard_di
    ;
import 'package:larksmite/features/networking/networking_
    injection_container.dart'
    as networking_di;
import 'package:larksmite/features/datastore/datastore_injection_
    container.dart'
    as datastore_di;
import 'package:file_picker/file_picker.dart';
import '../widget/app_menu_bar.dart';

class DashboardFrame extends StatefulWidget {
  const DashboardFrame({super.key});

  @override
  State<DashboardFrame> createState() => _DashboardFrameState();
}

class _DashboardFrameState extends State<DashboardFrame> {
  @override
  Widget build(BuildContext context) {
    return MultiBlocProvider(
```

```
providers: [
  BlocProvider(
    create: (context) => DashboardCubit(
      loadLabLayout: dashboard_di.sl(),
      serviceAdvertising: networking_di.sl(),
      manageClientConnections: networking_di.sl(),
    ),
  ),
  BlocProvider(
    create: (context) => BirdsEyeViewCubit(
      manageClientConnections: networking_di.sl(),
      larksmiteDatastore: datastore_di.sl(),
    ),
  ),
  BlocProvider(
    create: (context) => ScreenViewCubit(
      manageClientConnections: networking_di.sl(),
    ),
  ),
  BlocProvider(
    create: (context) => ReportViewCubit(
      manageClientConnections: networking_di.sl(),
    ),
  ),
  BlocProvider(
    create: (context) => AppMenuBarCubit(),
  ),
],
child: BlocListener<DashboardCubit, DashboardState>(
  listener: (context, state) async {
    switch (state) {
      case DashboardShowFilePicker():
        FilePickerResult? filePickerResult =
            await FilePicker.platform.pickFiles(
          initialDirectory: ".",
          dialogTitle: "Open layout file",
          type: FileType.custom,
          allowedExtensions: ["json"],
        );
        if (filePickerResult case FilePickerResult(:final
            files)
            when files.isNotEmpty && files.single.path !=
                null) {
          if (mounted) {
            context.read<DashboardCubit>().loadFromFile(
                  filePath: files.single.path!,
```

```
                );
              }
            } else if (mounted) {
              context.read<DashboardCubit>().layoutNotLoaded();
            }
          case DashboardLayoutLoaded(:final labLayout):
            context.read<AppMenuBarCubit>().layoutLoaded();
            context
                .read<BirdsEyeViewCubit>()
                .gotLabLayout(labLayout: labLayout);
            context
                .read<ScreenViewCubit>()
                .gotLabLayout(labLayout: labLayout);
            context
                .read<ReportViewCubit>()
                .gotLabLayout(labLayout: labLayout);
          default:
            context.read<AppMenuBarCubit>().layoutNotLoaded();
            context.read<BirdsEyeViewCubit>().noLabLayout();
            context.read<ScreenViewCubit>().noLabLayout();
            context.read<ReportViewCubit>().noLabLayout();
            break;
        }
      },
      child: const AppMenuBar(
        body: Scaffold(
          body: Row(
            children: [
              NavigationRailSide(),
              VerticalDivider(),
              DashboardViewFrame()
            ],
          ),
        ),
      ),
    ),
  );
 }
}
```

larkclient_coordinator.dart

```
import 'dart:async';
import 'dart:io';
import 'dart:typed_data';
import 'package:async/async.dart';
import 'package:fpdart/fpdart.dart';
```

```
import 'package:larkclient/core/entity/client_name_and_key.dart';
import 'package:larkclient/core/entity/larksmite_server.dart';
import 'package:larkclient/core/logging.dart';
import 'package:larkclient/core/status_types.dart';
import 'package:larkclient/feature/datastore/domain/usecase/
    server_and_client_details.dart';
import 'package:larkclient/feature/networking/domain/usecase/
    manage_server_conn.dart';
import 'package:larkclient/feature/platform_event_handler/domain/
    repository/platform_event_handler.dart';
import 'package:larksmite_shared/larskmite_shared_entites.dart';

class LarkClientCoordinator {
  final ServerAndClientDetails serverAndClientDetails;
  final ManageServerConnection manageServerConnection;
  HttpClient? _httpClient;
  LarkClientCoordinator({
    required this.serverAndClientDetails,
    required this.manageServerConnection,
    PlatformEventHandler? platformEventHandler,
  }) : _platformEventHandlerSupplied = platformEventHandler;
  final PlatformEventHandler? _platformEventHandlerSupplied;
  PlatformEventHandler get platformEventHandler {
    if (_platformEventHandlerSupplied != null) {
      return _platformEventHandlerSupplied!;
    } else {
      throw UnimplementedError(
        "PlatformEventHandler is not implemented for this platform
            ",
      );
    }
  }

  Stream<Either<Failure, Success>> startEverything({
    ClientNameAndKey? clientNameAndKey,
  }) async* {
    try {
      platformEventHandler;
    } on UnimplementedError catch (error, stacktrace) {
      getLogger().severe(
        "Event handler isn't implemented for this platform",
        error,
        stacktrace,
      );
      yield Either.left(
        NoPlatformEventHandlerAvailable(
```

```
      message: "This platform is not supported",
    ),
  );
  return;
} on Exception catch (error, stacktrace) {
  getLogger().severe(
    "Unexpected error while trying to access
        PlatformEventHandler",
    error,
    stacktrace,
  );
  yield Either.left(
    UnknownUnrecoverableFailure(
      message:
          "Unexpected error while checking if this platform is
              supported",
    ),
  );
  return;
}
final handlerStream = handlePlatformSpecificEvents();

// If the passed clientNameAndKey is null, it (hopefully)
    means its not the first time
// But if it's not null, then it means its the first time
ClientNameAndKey clientDetails;
bool isFirstTime;
if (clientNameAndKey != null) {
  clientDetails = clientNameAndKey;
  isFirstTime = true;
} else {
  final clientNameAndKeyResult =
      await serverAndClientDetails.getSavedClientNameAndKey();
  switch (clientNameAndKeyResult) {
    case Right(:final value):
      clientDetails = value;
      isFirstTime = false;
    case Left(:final value):
      getLogger().severe(
        "Failure while trying to retrieve saved client details
            ",
      );
      yield Either.left(value);
      return;
  }
}
```

```
  final connectionResult = connectToServerSecurely(
    clientNameAndKey: clientDetails,
    isFirstTime: isFirstTime,
  );
  yield* StreamGroup.merge([handlerStream, connectionResult]);
  getLogger().config("All streams finished in startEverything");
  return;
}

/// If [retryTimes] is 0, it will retry until we get successful
     result
/// The result of calling [shouldStopImmediately] with any
    encountered failure
/// is used to determine if we should stop immediately, useful
    in cases where
/// [retryTimes] is inifnite but it should stop when a certain
    condition is met
Future<Either<Failure, T>> _retryTask<T>({
  required Future<Either<Failure, T>> Function() task,
  int retryTimes = 0,
  Duration delayBetweenRetries = const Duration(seconds: 3),
  bool Function(Failure failure)? shouldStopImmediately,
}) async {
  bool retryCondition(int iInside, int retryTimesInside) {
    return (iInside < retryTimesInside || retryTimesInside == 0)
        ;
  }

  Failure? lastFailure;
  shouldStopImmediately ??= (_) => false;
  for (int i = 0; retryCondition(i, retryTimes); i++) {
    final result = await task();
    switch (result) {
      case Right(:final value):
        return Either.right(value);
      case Left(:final value):
        if (shouldStopImmediately(value)) {
          return Either.left(value);
        } else {
          lastFailure = value;
        }
    }
    await Future.delayed(delayBetweenRetries);
  }
  lastFailure ??=
```

```
      NoMoreRetries(message: "No more retries available for task
          ");
  return Either.left(lastFailure);
}

/// This will yield succesful event, if it encounters any
    failure, it will yield
/// that Failure and close immediately, its up to the caller to
     handle it.
/// This version of connection method is more try-and-fail type
     than
/// ask-for-permission type, thus simplifying stuff massively
Stream<Either<Failure, Success>> connectToServerSecurely({
  required ClientNameAndKey clientNameAndKey,
  bool isFirstTime = false,
}) async* {
  getLogger().config(
      "This is${isFirstTime ? '' : ' not'} the first time we're
          starting");
  // Search until we get the server
  LarkSmiteServer searchedServer;

  final searchedServerResult = await _retryTask(
    task: () => manageServerConnection.searchAndGetServer(),
    delayBetweenRetries: const Duration(seconds: 1),
    shouldStopImmediately: (failure) => failure is
        ServerSearchCancelled,
  );
  switch (searchedServerResult) {
    case Right(:final value):
      searchedServer = value;
    case Left(:final value):
      // Since retryTimes is 0 for server searching task, it
          should be able to
      // get the server eventually, but if it's unable to do so,
           it must be
      // the cancelled scenario
      getLogger().severe("Searching server failed", value);
      yield Either.left(value);
      return;
  }
  if (!isFirstTime) {
    getLogger()
        .config("It's not first start, so checking server by
            challenge");
    final isServerTrustedResult = await serverDoChallenge(
```

```
    larkSmiteServer: searchedServer,
  );
  bool isServerTrusted;
  switch (isServerTrustedResult) {
    case Right(:final value):
      isServerTrusted = value;
    case Left(:final value):
      yield Either.left(value);
      return;
  }
  if (!isServerTrusted) {
    yield Either.left(
      ServerChallengeFailed(
        message:
            "${searchedServer.ipAddress} gave unsuccessful
                response for challenge, not trusting it.",
      ),
    );
    return;
  } else {
    getLogger().config("Server completed challenge successfuly
        ");
  }
}

// If we're here, it means the server is trustworthy, or its
    the first time
// which case, we assume its trusted
Uint8List certBinary;

final certBinaryResult = await _retryTask(
  task: () => manageServerConnection.downloadCertificate(
    larkSmiteServer: searchedServer,
  ),
  retryTimes: 5,
);
switch (certBinaryResult) {
  case Right(:final value):
    certBinary = value;
  case Left(:final value):
    getLogger().severe("Unable to download certificate", value
        );
    yield Either.left(value);
    return;
}
final securityContext = SecurityContext();
```

```
securityContext.setTrustedCertificatesBytes(certBinary);
final httpClient = HttpClient(context: securityContext);
_httpClient = httpClient;
final connectionStream = manageServerConnection.
    connectToServer(
  clientNameAndKey: clientNameAndKey,
  httpClient: httpClient,
  ipAddress: searchedServer.ipAddress,
  isFirstStartFlag: isFirstTime,
);
yield* connectionStream.transform(
  StreamTransformer.fromHandlers(
    handleData: (data, sink) async {
      switch (data) {
        case Right(value: final connection):
          if (isFirstTime) {
            getLogger().config(
              "Since it's the first time, "
              "saving public key of server and name + key of
                  client",
            );
            // We don't want any stale data
            await serverAndClientDetails.clearAllData();
            await serverAndClientDetails.saveClientNameAndKey(
              clientNameAndKey: clientNameAndKey,
            );
            await serverAndClientDetails.savePublicKey(
              rsaPublicKey: connection.rsaPublicKeyAbstract,
            );
          }
          sink.add(
            Either.right(
              ConnectServerSuccess(message: "Connected to
                  server"),
            ),
          );
        case Left(:final value):
          sink.add(Either.left(value));
      }
    },
    handleDone: (sink) {
      sink.close();
    },
    handleError: (error, stackTrace, sink) {
      getLogger().severe(
        "Unknown error while transforming in
```

```
            connectToServerSecurely",
          error,
          stackTrace,
        );
        sink.close();
      },
    ),
  );
  return;
}

/// This will handle all events from server. Call this before
    connecting so as to
/// not miss any events
Stream<Either<Failure, Success>> handlePlatformSpecificEvents()
    {
  // Returning as broadcast stream since cancelling this will
      interfere with
  // the cleanup process, but if its a broadcast stream, it will
       continue
  // cleaning up and finish naturally
  return platformEventHandler
      .handleEvents(
        serverCommunicatorDuplex:
            manageServerConnection.serverCommunicatorDuplex,
      )
      .asBroadcastStream();
}

Future<Either<Failure, bool>> serverDoChallenge({
  required LarkSmiteServer larkSmiteServer,
}) async {
  final savedPublicKeyResult =
      await serverAndClientDetails.getPreviouslySavedPublicKey()
          ;
  RSAPublicKeyAbstract rsaPublicKey;
  switch (savedPublicKeyResult) {
    case Right(:final value):
      rsaPublicKey = value;
    case Left(:final value):
      return Either.left(value);
  }
  return manageServerConnection.performChallenge(
    larkSmiteServer: larkSmiteServer,
    rsaPublicKey: rsaPublicKey,
  );
```

```
  }

  Future<void> stopEverything() async {
    await manageServerConnection.stopSearching();
    await manageServerConnection.disconnectFromServer();
    _httpClient?.close(force: true);
    await platformEventHandler.stopHandling();
  }
}
```

manage_server_conn.dart

```
import 'dart:async';
import 'dart:io';
import 'dart:typed_data';
import 'package:fpdart/fpdart.dart';
import 'package:larkclient/core/entity/client_name_and_key.dart';
import 'package:larkclient/core/entity/events_from_server/events_
    from_server.dart';
import 'package:larkclient/core/entity/events_to_server/events_to
    _server.dart';
import 'package:larkclient/core/entity/larksmite_server.dart';
import 'package:larkclient/core/entity/server_communicator_duplex
    .dart';
import 'package:larkclient/core/logging.dart';
import 'package:larkclient/core/status_types.dart';
import 'package:larkclient/feature/networking/domain/repository/
    server_repository.dart';
import 'package:larksmite_shared/larksmite_shared.dart';
import 'package:larksmite_shared/larskmite_shared_entites.dart';
import 'package:uuid/v4.dart';

class ManageServerConnection {
  final ServerRepository serverRepository;
  ServerCommunicatorDuplex get serverCommunicatorDuplex =>
      serverRepository.serverCommunicatorDuplex;
  ManageServerConnection({required this.serverRepository});
  Future<Either<Failure, LarkSmiteServer>> searchAndGetServer()
      async {
    return serverRepository.searchForServer();
  }

  Future<void> stopSearching() async {
    await serverRepository.stopSearchingForServer();
  }

  Future<Either<Failure, RecieveType>> _sendAndWaitForEvent<
```

```
    SendType extends EventsToSendToServer,
    RecieveType extends EventsRecievedFromServer>({
  required SendType sendEvent,
  Duration timeoutDuration = const Duration(seconds: 10),
}) async {
  final result = await sendAndWaitForEvent<SendType, RecieveType
      ,
      ServerErrorResponse, EventsToSendToServer,
          EventsRecievedFromServer>(
    sendEvent: sendEvent,
    sendSink: serverCommunicatorDuplex.eventsToSendToServer,
    recieveStream: serverCommunicatorDuplex.
        eventsRecievedFromServer,
    timeoutDuration: timeoutDuration,
  );
  return result;
}

Future<Either<Failure, ConnectionResponse>> _getConnectResult({
  required ClientNameAndKey clientNameAndKey,
  required bool isFirstStartFlag,
}) async {
  return _sendAndWaitForEvent<ConnectionRequest,
      ConnectionResponse>(
    sendEvent: ConnectionRequest.generate(
      clientNameAndKey: clientNameAndKey,
      isFirstTime: isFirstStartFlag,
    ),
    timeoutDuration: const Duration(seconds: 15),
  );
}

Stream<Either<Failure, ConnectionAccepted>> connectToServer({
  required ClientNameAndKey clientNameAndKey,
  required HttpClient httpClient,
  required InternetAddress ipAddress,
  required bool isFirstStartFlag,
}) {
  final connectToServerStream =
      serverRepository.connectToServerAndListenToEvents(
    httpClient: httpClient,
    ipAddress: ipAddress,
  );
  return connectToServerStream.transform(
    StreamTransformer.fromHandlers(
      handleData: (data, sink) async {
```

```
switch (data) {
  case Right():
    final connectionResult = await _getConnectResult(
      clientNameAndKey: clientNameAndKey,
      isFirstStartFlag: isFirstStartFlag,
    );
    switch (connectionResult) {
      case Right(value: ConnectionAccepted()):
        sink.add(Either.right(
            connectionResult.value as ConnectionAccepted)
                );
      case Right(value: ConnectionDeclined()):
        // Its Right here since, the connection was
            successful and the event was
        // recieved correctly
        sink.add(
          Either.left(
            ClientDetailsAreWrong(
              message:
                  "Server declined connection because
                      client details are wrong",
            ),
          ),
        );
      // Changing the types of [ErrorResponseFailure] and
          [TimeoutFailure]
      // from larksmite_shared to our own failure types
          which extends
      // [RecoverableFailure]
      case Left(
          value: ErrorResponseFailure(
            :final message,
            errorResponse: ServerErrorResponse(),
          )
        ):
        sink.add(Either.left(ServerErrorFailure(message:
            message)));
      case Left(value: TimeoutFailure(:final message)):
        sink.add(Either.left(ServerTimeoutFailure(message
            : message)));
      case Left(:final value):
        sink.add(Either.left(value));
    }
  case Left(:final value):
    sink.add(Either.left(value));
}
```

```
      },
      handleDone: (sink) {
        sink.close();
      },
      handleError: (error, stackTrace, sink) {
        getLogger().severe(
          "Error while transforming events in connectToServer",
          error,
          stackTrace,
        );
        sink.close();
      },
    ),
  );
}

Future<Either<Failure, Uint8List>> downloadCertificate({
  required LarkSmiteServer larkSmiteServer,
}) async {
  return serverRepository.downloadCertificate(
      larkSmiteServer: larkSmiteServer);
}

Future<Either<Failure, bool>> performChallenge({
  required LarkSmiteServer larkSmiteServer,
  required RSAPublicKeyAbstract rsaPublicKey,
}) async {
  final randomString = const Uuid V4().generate();
  final encryptedData = await LarksmiteCryptoRsaPointyCastle().
      encryptString(
    rsaPublicKey: rsaPublicKey as RSAPublicKeyPointyCastle,
    string: randomString,
  );

  final challengeResponse = await serverRepository.
      performChallenge(
    larkSmiteServer: larkSmiteServer,
    serverDoChallengeDecrypt: ServerDoChallengeDecrypt(
      encryptedData: encryptedData,
    ),
  );
  switch (challengeResponse) {
    case Left(value: ServerChallengeFailed()):
      return Either.right(false);
    case Left(:final value):
      return Either.left(value);
```

```
      case Right(:final value):
        // Simple equality check, the real check is if the server
            was able to perform
        // the decryption and send the plain text back
        return Either.right(
          value.decryptedString == randomString,
        );
    }
  }

  Future<Either<Failure, Success>> disconnectFromServer() async {
    return serverRepository.disconnectFromServer();
  }
}
```

linux_event_handler.dart

```
import 'dart:async';
import 'dart:io';
import 'package:fpdart/fpdart.dart';
import 'package:gnome_screenrecord/gnome_screenrecord.dart';
import 'package:larkclient/core/constants.dart';
import 'package:larkclient/core/entity/events_from_server/events_
    from_server.dart';
import 'package:larkclient/core/entity/events_to_server/events_to
    _server.dart';
import 'package:larkclient/core/entity/server_communicator_duplex
    .dart';
import 'package:larkclient/core/logging.dart';
import 'package:larkclient/core/status_types.dart';
import 'package:larkclient/core/useful_helpers.dart';
import 'package:larkclient/feature/platform_event_handler/data/
    datasource/linux_proc_source.dart';
import 'package:larkclient/feature/platform_event_handler/data/
    datasource/linux_run_bash_source.dart';
import 'package:larkclient/feature/platform_event_handler/domain/
    entity/screencast_object.dart';
import 'package:larkclient/feature/platform_event_handler/domain/
    repository/platform_event_handler.dart';
import 'package:larksmite_shared/larskmite_shared_entites.dart';
import 'package:process_monitor/process_monitor.dart';

class LinuxPlatformEventHandler implements PlatformEventHandler {
  final LinuxProcSource linuxProcSource;
  final LinuxRunBashSource linuxRunBashSource;
  ScreencastObj? _screencastObj;
  StreamController<Either<Failure, Success>>? _eventStatusStream;
```

```
LinuxPlatformEventHandler({
  required this.linuxProcSource,
  required this.linuxRunBashSource,
});
Future<ProcessModel> getRootProcess() {
  getLogger().config("Trying to get processes");
  final processMonitor = ProcessMonitor.instance;
  return processMonitor.getProcessTreeRootNode();
}

Future<ClientInfo> getClientInfo() async {
  getLogger().config("Trying to get client info");
  return ClientInfo(
    hostname: Platform.localHostname,
    uptime: await linuxProcSource.getUptime(),
    cpuUsage: await linuxProcSource.getCpuUsage(),
  );
}

Future<void> showMessageGui({
  required String message,
  required MessageType messageType,
}) async {
  getLogger().config("Showing GUI message to user");
  await linuxRunBashSource.showMessageGui(
    message: message,
    messageType: messageType,
  );
}

Future<Uri> getScreencastLink({
  required ScreenCastResolution screenCastResolution,
  required InternetAddress clientIp,
}) async {
  //TODO P3: find a way to make this less kludgy
  getLogger().config("Beginning screencast procedure");
  if (_screencastObj == null) {
    getLogger().config("Getting screen resolution");
    final gnomeRecorder = GnomeScreenRecorder();
    final res = await linuxRunBashSource.getScreenResolution();
    final int width, height;
    width = ((screenCastResolution.resolutionPercentage / 100) *
         res.width)
        .toInt();
    height = ((screenCastResolution.resolutionPercentage / 100)
```

```
    * res.height)
    .toInt();
Stream<DbusRecordStatus> screencastStatusStream;
getLogger().config("About to run gnomeScreenRecord");

screencastStatusStream = gnomeRecorder.startRecording(
  height: height,
  width: width,
  internetAddress: clientIp,
  port: screencastPort,
);

final streamLinkToSend = Completer<Uri>();
final screencastListener = screencastStatusStream.listen(
  // Fortunately we don't need to cancel this stream since
      it finishes when
  // the recording stops, but we should probably find a way
      to make it cleaner
  (event) {
    getLogger().config("Got event $event from gnome screen
        record");
    switch (event) {
      case DbusRecordStarted(
          :final streamLink,
          :final dbusMethodResponse,
        ):
        getLogger().config(
          "Dbus start screen record response: $
              dbusMethodResponse",
        );
        _screencastObj = ScreencastObj(
          streamLinkToSend: streamLink,
          screenRecorder: gnomeRecorder,
        );
        streamLinkToSend.complete(streamLink);
      case DbusRecordWaiting():
        getLogger().config("Screencasting... Waiting for stop
            signal");
      case DbusRecordEnded(:final dbusMethodResponse):
        getLogger().config(
          "Dbus stop screen record response: $
              dbusMethodResponse",
        );
      case DbusError(:final dBusMethodResponseException):
        getLogger().warning(
            "Unable to start recording, response: $
```

```
                dBusMethodResponseException");
            streamLinkToSend.completeError(ScreencastException())
                ;
        }
      },
      cancelOnError: true,
    );
    await streamLinkToSend.future;
    // await screencastListen.cancel();
  } else {
    getLogger()
        .config("Skipping some steps since screencast is already
            running");
  }

  final screencastLink = Uri.parse(
    "tcp://${clientIp.address}:$screencastPort",
  );

  getLogger().config("Sending screen cast link: $screencastLink
      ");
  return screencastLink;
}

Future<void> stopScreencast() async {
  getLogger().config("Stopping screencast");
  if (_screencastObj != null) {
    await _screencastObj!.screenRecorder.stopRecording();
    _screencastObj = null;
  }
  getLogger().config("Stopped screencast");
}

Future<void> handlePowerSignal({
  required PowerSignal powerSignal,
}) async {
  await Future.delayed(powerSignal.performSignalAfter);
  switch (powerSignal.powerSignalType) {
    case PowerSignalType.lockScreen:
      await linuxRunBashSource.lockScreen();
    case PowerSignalType.sleep:
      await linuxRunBashSource.sleepComputer();
    case PowerSignalType.restart:
      await linuxRunBashSource.restartComputer();
    case PowerSignalType.powerOff:
      await linuxRunBashSource.shutdownComputer();
```

```
  }
}

Future<List<int>> getScreenshot() {
  return linuxRunBashSource.getScreenshot();
}

@override
Stream<Either<Failure, Success>> handleEvents({
  required ServerCommunicatorDuplex serverCommunicatorDuplex,
}) async* {
  getLogger().config("Starting the Linux event handler");
  _eventStatusStream = StreamController();

  final serverEventListener =
      serverCommunicatorDuplex.eventsRecievedFromServer.listen(
    (event) async {
      switch (event) {
        case ClientProcessesRequired(:final token):
          try {
            final rootProcess = await getRootProcess();
            serverCommunicatorDuplex.eventsToSendToServer.add(
              ClientProcessesResponse.generateResponseTo(
                responseToToken: token,
                rootProcess: rootProcess,
              ),
            );
            _eventStatusStream?.add(
              Either.right(
                EventHandledSuccessfuly(
                  message: "Handled get process request
                      successfuly",
                  eventToken: token,
                ),
              ),
            );
          } on Exception catch (error, stacktrace) {
            getLogger().warning(
                "Exception while handling ClientProcessRequired",
                error,
                stacktrace);
            serverCommunicatorDuplex.eventsToSendToServer.add(
              ClientErrorResponse.generateResponseTo(
                responseToToken: token,
                message: "Unable to get processes",
                errorString: error.toString(),
```

```
          stacktraceString: stacktrace.toString(),
        ),
      );
      _eventStatusStream?.add(
        Either.left(
          EventHandleFailed(
            message: "Handling of GetProcesses failed",
            eventToken: token,
          ),
        ),
      );
    }
  case ClientInfoRequired(:final token):
    try {
      final clientInfo = await getClientInfo();
      serverCommunicatorDuplex.eventsToSendToServer.add(
        ClientInfoResponse.generateResponseTo(
          responseToToken: token,
          clientInfo: clientInfo,
        ),
      );
      _eventStatusStream?.add(
        Either.right(
          EventHandledSuccessfuly(
            message: "Handled get client info successfuly",
            eventToken: token,
          ),
        ),
      );
    } on Exception catch (error, stacktrace) {
      getLogger().warning("Exception while handling
          ClientInfoRequired",
          error, stacktrace);
      serverCommunicatorDuplex.eventsToSendToServer.add(
        ClientErrorResponse.generateResponseTo(
          responseToToken: token,
          message: "Unable to get info",
          errorString: error.toString(),
          stacktraceString: stacktrace.toString(),
        ),
      );
      _eventStatusStream?.add(
        Either.left(
          EventHandleFailed(
            message: "Handling of GetInfo failed",
            eventToken: token,
```

```
        ),
      ),
    );
  }
case ClientStatusRequired(:final token):
  serverCommunicatorDuplex.eventsToSendToServer.add(
    ClientStatusResponse.generateResponseTo(
      responseToToken: token,
      clientStatus: ClientStatus.ok,
    ),
  );
  _eventStatusStream?.add(
    Either.right(
      EventHandledSuccessfuly(
        message: "Handled get client status successfuly",
        eventToken: token,
      ),
    ),
  );
case PowerSignalRecieved(:final token, :final
    powerSignal):
  try {
    handlePowerSignal(powerSignal: powerSignal);
    serverCommunicatorDuplex.eventsToSendToServer.add(
      ClientAcknowledgeResponse.generateResponseTo(
        responseToToken: token,
      ),
    );
    _eventStatusStream?.add(
      Either.right(
        EventHandledSuccessfuly(
          message: "Handled power signal successfuly",
          eventToken: token,
        ),
      ),
    );
  } on Exception catch (error, stacktrace) {
    getLogger().warning(
        "Exception while handling PowerSignalRecieved",
        error,
        stacktrace);
    serverCommunicatorDuplex.eventsToSendToServer.add(
      ClientErrorResponse.generateResponseTo(
        responseToToken: token,
        message: "Unable to handle power signal",
        errorString: error.toString(),
```

```
          stacktraceString: stacktrace.toString(),
        ),
      );
      _eventStatusStream?.add(
        Either.left(
          EventHandleFailed(
            message: "Handling of power signal failed",
            eventToken: token,
          ),
        ),
      );
    }
  case ClientScreencastRequired(
      :final token,
      :final screenCastResolution,
      :final ipAddressOfClient,
    ):
    try {
      final screencastLink = await getScreencastLink(
        screenCastResolution: screenCastResolution,
        clientIp: ipAddressOfClient,
      );
      serverCommunicatorDuplex.eventsToSendToServer.add(
        ClientScreencastResponse.generateResponseTo(
          responseToToken: token,
          screencastUri: screencastLink,
        ),
      );
      _eventStatusStream?.add(
        Either.right(
          EventHandledSuccessfuly(
            message: "Handled get screencast successfuly",
            eventToken: token,
          ),
        ),
      );
    } on Exception catch (error, stacktrace) {
      getLogger().warning(
          "Exception while handling
              ClientScreencastRequired",
          error,
          stacktrace);
      serverCommunicatorDuplex.eventsToSendToServer.add(
        ClientErrorResponse.generateResponseTo(
          responseToToken: token,
          message: "Unable to get screencast",
```

```
          errorString: error.toString(),
          stacktraceString: stacktrace.toString(),
        ),
      );
      _eventStatusStream?.add(
        Either.left(
          EventHandleFailed(
            message: "Handling of screen cast failed",
            eventToken: token,
          ),
        ),
      );
    }
  case ClientStopScreencast(
      :final token,
    ):
    try {
      await stopScreencast();
      serverCommunicatorDuplex.eventsToSendToServer.add(
        ClientAcknowledgeResponse.generateResponseTo(
          responseToToken: token,
        ),
      );
      _eventStatusStream?.add(
        Either.right(
          EventHandledSuccessfuly(
            message: "Handled stop screencast successfuly",
            eventToken: token,
          ),
        ),
      );
    } on Exception catch (error, stacktrace) {
      getLogger().warning(
          "Exception while handling ClientStopScreencast",
          error,
          stacktrace);
      serverCommunicatorDuplex.eventsToSendToServer.add(
        ClientErrorResponse.generateResponseTo(
          responseToToken: token,
          message: "Unable to stop screencast",
          errorString: error.toString(),
          stacktraceString: stacktrace.toString(),
        ),
      );
      _eventStatusStream?.add(
        Either.left(
```

```
        EventHandleFailed(
          message: "Handling of stop screen cast failed",
          eventToken: token,
        ),
      ),
    );
  }
case ClientScreenshotRequired(
    :final token,
  ):
  try {
    final photoBinary = await getScreenshot();
    serverCommunicatorDuplex.eventsToSendToServer.add(
      ClientScreenshotResponse.generateResponseTo(
        responseToToken: token,
        photoBinary: photoBinary.toList(),
      ),
    );
    _eventStatusStream?.add(
      Either.right(
        EventHandledSuccessfuly(
          message: "Handled get screenshot successfuly",
          eventToken: token,
        ),
      ),
    );
  } on Exception catch (error, stacktrace) {
    getLogger().warning(
        "Exception while handling ClientGetScreenshot",
        error,
        stacktrace);
    serverCommunicatorDuplex.eventsToSendToServer.add(
      ClientErrorResponse.generateResponseTo(
        responseToToken: token,
        message: "Unable to get screenshot",
        errorString: error.toString(),
        stacktraceString: stacktrace.toString(),
      ),
    );
    _eventStatusStream?.add(
      Either.left(
        EventHandleFailed(
          message: "Handling of get screenshot failed",
          eventToken: token,
        ),
      ),
```

```
    );
  }
case ClientShowMessage(
    :final token,
    :final message,
    :final messageType,
  ):
  try {
    await showMessageGui(message: message, messageType:
        messageType);
    serverCommunicatorDuplex.eventsToSendToServer.add(
      ClientAcknowledgeResponse.generateResponseTo(
        responseToToken: token,
      ),
    );
    _eventStatusStream?.add(
      Either.right(
        EventHandledSuccessfuly(
          message: "Handled show message successfuly",
          eventToken: token,
        ),
      ),
    );
  } on Exception catch (error, stacktrace) {
    getLogger().warning("Exception while handling
        ClientShowMessage",
        error, stacktrace);
    serverCommunicatorDuplex.eventsToSendToServer.add(
      ClientErrorResponse.generateResponseTo(
        responseToToken: token,
        message: "Unable to show message",
        errorString: error.toString(),
        stacktraceString: stacktrace.toString(),
      ),
    );
    _eventStatusStream?.add(
      Either.left(
        EventHandleFailed(
          message: "Handling of showing message failed",
          eventToken: token,
        ),
      ),
    );
  }
case ServerAdvertisementRecieved():
case ConnectionAccepted():
```

```
      case ConnectionDeclined():
      case ServerErrorResponse():
        // No need to handle them here, since they're platform
            independent events
        break;
    }
  },
  cancelOnError: true,
);

serverEventListener.onDone(
  () async {
    getLogger().config(
        "Finished listening to events from server in Linux
            event handler");
    _eventStatusStream?.addIfNotClosed(
      Either.left(
        EventsFromServerFinished(
          message: "Finished listening to events in Linux event
              handler",
        ),
      ),
    );
    await _eventStatusStream?.close();
  },
);

serverEventListener.onError(
  (error, stacktrace) async {
    getLogger().severe(
      "Server event listener had an error",
      error,
      stacktrace,
    );
    _eventStatusStream?.addIfNotClosed(
      Either.left(
        EventsFromServerFinished(
          message:
              "Events from server ended with error in Linux
                  event handler",
        ),
      ),
    );
    await _eventStatusStream?.close();
  },
);
```

```
    // Will keep running until it's closed by the top two factors
        or external stop
    getLogger().config("Going to start listening to
        eventStatusStream");
    yield* _eventStatusStream!.stream;
    getLogger().config("Cleaning up Linux event handler");
    await stopScreencast();
    await serverEventListener.cancel();
    _eventStatusStream = null;
    getLogger().config("Finished cleaning up Linux event handler")
        ;
    return;
  }

  @override
  Future<void> stopHandling() async {
    getLogger().config("Stopping linux platform handler");
    await stopScreencast();
    _eventStatusStream?.addIfNotClosed(
      Either.left(
        EventHandlerCancelled(
          message: "Event handler stopped",
        ),
      ),
    );
    getLogger().config("Closing the eventStatus stream");
    await _eventStatusStream?.sink.close();
    await _eventStatusStream?.close();
  }
}
```

BREAST CANCER PREDICTION

PROJECT REPORT

Submitted By

P N MURALI SANKAR

Reg. No. CCAVBCA009

For the award of the Degree of

Bachelor of Computer Application

(BCA)

(University of Calicut)

under the guidance of

Mr. Joju Sebastian

Assistant Professor

BACHELOR OF COMPUTER APPLICATION
DEPARTMENT OF COMPUTER SCIENCE
CHRIST COLLEGE
IRINJALAKUDA, KERALA

2021-2024

DEPARTMENT OF COMPUTER SCIENCE

CHRIST COLLEGE (AUTONOMOUS)

IRINJALAKUDA

CERTIFICATE

This is to certify that the project report entitled ***"BREAST CANCER PREDICTION"*** *is a bonafide record of the project work done by* ***P N MURALI SANKAR*** *in partial fulfillment of the requirement for the sixth semester of* ***Bachelor of Computer Application*** *in Department of Computer Science of* ***CHRIST COLLEGE (AUTONOMOUS) IRINJALAKUDA***

.

Mr.Joju Sebastian
Assistant Professor,CS
Internal Guide

Ms. Sini Thomas
Head of the Department
Computer Science

EXTERNAL EXAMINER **INTERNAL EXAMINER**

DECLARATION

We here by declare that this project work **"BREAST CANCER PREDICTION"** submitted by Christ College (Autonomous)Irinjalakuda,affliated to Calicut University in partial fullfillment of the requirement for the award of the Bachelor of Computer Application, is a record of original work done by us,under the guidance of Mr.JOJU SEBASTIAN,Department of Computer Science.

Place: Irinjalakuda P N MURALI SANKAR

ACKNOWLEDGEMENT

First and foremost we like to thank Lord almighty for his providence and for being the guiding light throughout the project. We wish to express my sincere gratitude to our beloved Department head for giving me all the facilities for our project. We take this opportunity to express my gratitude to the class teacher Ms. SOWMYA PS and head of the department Ms. SINI THOMAS who has been supported us throughout the course of this project. We are thankful for her aspiring guidance and valuable advice during the project work. We express my sincere thanks to my project guide Mr.JOJU SEBASTIAN for supporting and guiding throughout the project.We would take this opportunity to specially thank all other faculty members for their constant and continuous motivation. Finally we would like to thank my family and friends for giving valuable advice and moral support throughout our project.

ABSTRACT

BREAST CANCER PREDICTION is a desktop application that uses CNN algorithm to identify cancer and non cancerous cells in medica lorganization. This system automates the process of predicting cancer by inputting histopathological images and analyzing them to identify cancer and non cancer cells.

Contents

Chapter 1

1 Introduction

"Breast Cancer Prediction" introduces a revolutionary solution for predicting breast cancer, replacing traditional methods. This system incorporates deep learning algorithms to enhance the entire prediction process. The advantages of this project includes heightened accuracy, time efficiency, improved security, and reduced costs. As medical organizations embrace this user-friendly and innovative system, they are propelled into a more streamlined and technologically advanced future of breast cancer prediction.

.

1.1 Overview

The objective of the "Breast Cancer Prediction" project is to develop and implement a cutting-edge solution that automates breast cancer prediction using CNN algorithm. The goal is to replace traditional prediction methods with a more accurate, efficient, and secure solution, ultimately improving outcomes for patients and medical professionals.

Chapter 2

2 System Analysis

2.1 Purpose

Breast Cancer Prediction is a desktop applications that uses CNN algorithm to identify cancer and non-cancer cells in medical organizations. The system automates the process of predicting cancer by inputing histopathological images and analysing them to identify cancer and non-cancer cells. The system is typically composed of a software. The histopathological image of individuals browsed from the system are taken and send to the software, upon clicking predict button the image is analysed and compared with the trained dataset. The dataset contains information about each person's multiple histopathological images cells. The weights of the images are recorded and based on the weights of the new images the result is predicted.

2.1.1 Existing System

The existing system of the breast cancer prediction primarily relies on traditional prediction methods, such as using machine learning algorithms which predicts optimally the cancer and non-cancer cells from the trained histopathological dataset. Here, when a new histopathological image is taken the result accuracy is poor.

2.1.2 Proposed System

Breast Cancer Prediction is a desktop application that is in need for breast cancer identification on lab. Here, Project wise there is a GUI created that takes an image as input and classifies that it's a cancerous or non-cancerous. In this system when a new histopathological image is given here the result is predicted optimally and accuracy is good due to the use of CNN algorithm. Medically speaking Biopsy is taken place here .The histopathological images (Histology – Study of tissues , Pathology – Study of diseases) are taken from the monitor using advanced electron microscope by selecting a sample tissue and then processed and cut into thin sections then stained with Hematoxylin – Eosin (H&E).

2.2 Problem definition

The proposed project aims to tackle the limitation and inefficiency of traditional prediction methods by developing "Breast Cancer Prediction". The existing system predicts the cancer using a machine learning algorithm which only predicts the result based on the image given as input is the trained histopathological images which often lead to inaccuracies. To address these issues, this project will leverage deep learning method to provide a more accurate, secure, and

user-friendly approach to breast cancer prediction. By integrating seamlessly with existing systems and adapting to varying environmental conditions, this system promises to revolutionize breast cancer prediction, delivering enhanced accuracy, efficiency, and security for medical organizations.

2.3 FEASIBILITY STUDY

After the problem is clearly understood and solutions proposed, the next step is to conduct the feasibility study. Feasibility study is defined as evaluation or analysis of the potential impact of the proposed project or program. The objective is to determine whether the proposed system is feasible. There are three aspects of feasibility study to which the proposed system is subjected as discussed below.

2.3.1 Technical Feasibility

The proposed breast cancer prediction system using CNN algorithms is technically feasible with all required tools readily available. The system, implemented in Python, leverages CNN for its speed, efficiency, and low resource requirements. However, the accuracy of these algorithms depends on the number of trained images, necessitating thorough testing and optimization. The system requires a well-curated database of histopathological images, with regular updates and management crucial for accurate prediction. Robust measures for data protection and compliance with privacy regulations are essential due to the sensitive nature of the data involved. In conclusion, the technical feasibility of the system is evident, and with proper planning, testing, and attention to privacy and security, it can be a viable solution for various settings, enhancing operational efficiency.

2.3.2 Economical Feasibility

In this proposed system, all the tools used are free and open source. So that development cost is minimum. The hardware is used to run the system. It is built in our laptop. And also hardware requirement is feasible and not much breast cancer prediction system maintenance is required. The development of the system will not need a huge amount of money. It will be economically feasible. And the money spend for the application will be worth. Hence, the proposed system is economically feasible.

2.3.3 Operational Feasibility

The feasibility of a breast cancer prediction system using CNN algorithms is crucial.CNN's ease of implementation reduces development time and costs, and its low resource requirements allow deployment on various hardware configurations. The system requires a well-maintained dataset of pre-registered faces, regular updates, and management to accommodate user base changes. Addressing privacy and security concerns is critical due to the sensitive nature of

biometric data. User acceptance is a significant factor for successful implementation, requiring effective communication of benefits, addressing concerns, and providing adequate training. With proper planning, thorough testing, and compliance with privacy regulations, a breast cancer prediction system using CNN can predict optimal breast cancer results and enhance operational efficiency.

Chapter 3

3 Software Requirement Specification

3.1 Purpose

The breast cancer diagnosis system aims to leverage deep learning algorithms to accurately analyze histopathological images and automatically predict whether cancer is present. Key goals are achieving real-time cancer screening with minimal errors, extracting insightful image features, training robust models CNN, and obtaining analytics to uncover patterns in cancer data - ultimately enhancing efficiency, reliability and research capabilities compared to manual diagnosis processes. The system requires histopathology image datasets to train models to classify images as normal or malignant based on texture, shape and cell characteristics. It will provide a practical tool set to augment and automate parts of the breast cancer screening process.

3.2 Scope

The scope of this project encompasses the development of a software-based solution for automated breast cancer prediction from histopathological images. It leverages deep learning techniques to analyze cell structure and identify malignant indicators.

3.3 Overall Description

The desktop application caters to medical professionals, offering streamlined analysis of digitized histopathology slide images. Convolutional Neural Networks, it assesses cell morphology, texture, shape, and spatial characteristics to classify images as malignant or benign with confidence scores. The software guides users through preprocessing, feature extraction, model execution, and post-processing steps. Results, metrics, and visual heatmaps highlighting malignant regions are easily accessible. The back-end architecture includes a database, modeling module, processing pipelines, integration logic, and access control layers. Through rigorous training and testing with labeled datasets, the system ensures accuracy, aiding medical decision-making and enhancing patient care.

3.3.1 Product Perspective

The breast cancer prediction system aims to integrate smoothly with current medical systems, empowering healthcare professionals with accurate diagnostics. It aligns with organizational goals of enhancing patient care and operational efficiency, offering a user-friendly interface and scalability to adapt to diverse healthcare settings and technological advances.

3.3.2 Product Functionality

The breast cancer prediction system functions by analyzing histopathological images using CNN algorithm to classify cancerous and non-cancerous cells. Users input images into the system, which then processes them, compares them with a trained dataset, and generates predictions with high accuracy. The system offers a user-friendly interface for seamless interaction, contributes to faster and more accurate cancer detection, and integrates with existing medical infrastructure to enhance overall diagnostic capabilities.

3.3.3 Users and Characteristics

The breast cancer prediction system caters primarily to healthcare professionals involved in cancer diagnosis, such as doctors, radiologists, and pathologists. These users typically possess medical expertise and are familiar with diagnostic processes. They require a system that is intuitive, reliable, and capable of providing accurate predictions to support their clinical decision-making. Additionally, administrators responsible for implementing and maintaining the system within healthcare organizations may also interact with it, requiring a broader understanding of its functionality and integration capabilities.

3.4 Specific Requirements

3.4.1 Hardware Requirements

- System: IBM-Compatible PC
- Processor: Intel Core i3 or above
- Speed: Above 1GHz
- RAM capacity: 4 GB
- Hard Dsk drive: 500 GB
- Keyboard: Standard
- Mouse: Standard
- Monitor: SVGA Color

3.4.2 Software Requirements

- Front End:Python
- Back End: Python
- IDE: Anaconda

3.5 Functional Requirements

It contains three main modules.

- 1.Automated Cancer Prediction
- 2.Real-Time Prediction
- 3.User Interface

Automated Cancer Prediction

The system should automate the process of predicting breast cancer using CNN algorithms based on input histopathological images.

Real-Time Prediction

The system should offer real-time breast cancer prediction capabilities, enabling medical professionals to obtain results promptly.

User Interface

The user interface should be user-friendly and intuitive, allowing medical professionals to input histopathological images easily and view predicted results.

3.6 Non Functional Requirements

Non-functional requirements define the overall qualities or attributes of the resulting system. Non-functional system place restrictions on the product being developed, the developed process, and specify external constraints that the product must meet. Examples of non-functional requirements include safety, security, usability, reliability and performance requirements. Project management issues (costs, time and schedule) are often considered as non-functional requirements. The principal non - functional constraints which are relevant to critical systems :

- performance
- security
- safety
- usability

Performance

Performance requirements concern the speed of operation of a system. Types of performance requirements :

- Response requirements (how quickly the system reacts to a user input).

- Throughput requirements (how much the system can accomplish within a specified amount of time).
- Availability requirements (is the system available for service when requested by end users). The speed of operation of this system is adequate for the requirements.

Reliability

- Reliability is the ability of a system to perform its required function under stated conditions for a specified period of time.
- constraints on the runtime behavior of the system. This system is reliable because its functionalities can be done on the required conditions.

Safety

Safety requirements are not required which exclude unsafe situation from the possible solution of the system.

Usability

Usability is the ease with which a user can learn to operate, prepare inputs for, and interpret outputs of system or components. Usability requirements include :

- information error messages.
- well-formed user interfaces.

3.7 Interface Requirements

3.7.1 Hardware interfaces

The system must run over the internet, all the hardware shall require to connect internet will be hardware interface for the system. As for example modem, WAN-LAN.

3.7.2 Software interfaces

Software interface required for the working of the project is the appropriate operating system.

3.7.3 Communication interfaces

The user of the site communicate using the network connectivity and the data set is accessed

3.8 Security Requirements

- User accesses only their account.
- Validation of input is handled.
- This application containing the computer systems is physically secured against arms or surreptitious entry by intruders.
- Users must be authorized carefully to reduce changes of any such user giving access to an intruder in exchange for a bribe or other favour.

3.9 Platform Used

Windows 11 is the operating system chosen for the "Breast Cancer Prediction" project. Windows 10 provides a user-friendly interface and robust support for various hardware components and software applications. It is widely used in both personal and professional settings, making it a suitable choice for medical organizations implementing the breast cancer prediction system. Additionally, Windows 11 offers security features and regular updates to ensure system stability and reliability.

3.10 Technologies Used

PYTHON

Python is a versatile, high-level programming language known for its simplicity and readability. It offers dynamic typing and automatic memory management, making it suitable for various domains such as web development, data analysis, artificial intelligence, and automation. Python's extensive standard library provides pre-built modules and packages for tasks ranging from file manipulation to network programming. Its interpreted nature allows for rapid development and easy debugging. Python supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Its popularity is fueled by a vibrant community, extensive documentation, and numerous third-party libraries. Overall, Python's simplicity, flexibility, and robust ecosystem make it an ideal choice for beginners and seasoned developers alike

CNN

Convolutional Neural Networks (CNNs) are a class of deep neural networks primarily used for image recognition and computer vision tasks. They consist of multiple layers, including convolutional layers, pooling layers, and fully connected layers. CNNs excel at learning hierarchical representations of features from input data. Convolutional layers apply filters to input images, extracting features such as edges and textures. Pooling layers reduce spatial dimensions while preserving important features. Fully connected layers combine extracted features for classification or regression tasks. CNNs leverage parameter sharing and local connectivity to efficiently learn patterns from large datasets. Their success in image recognition tasks has made them a cornerstone in various fields, including autonomous driving, medical imaging, and facial recognition

Chapter 4

4 Design Document

4.1 Purpose

The purpose of the Design Document for the "Breast Cancer Prediction" project is to provide a concise yet comprehensive overview of the system's architecture, components, functionalities, and interactions. It serves as a roadmap for development, ensuring alignment with project goals and requirements. Additionally, the document facilitates communication and collaboration among team members and stakeholders, guiding informed decision-making and ensuring the successful development and implementation of the breast cancer prediction system.

4.2 Scope

The scope of the "Breast Cancer Prediction" project involves developing an automated breast cancer prediction system using deep learning algorithms. This includes designing a user-friendly interface for inputting histopathological images and implementing backend logic for image processing and prediction generation. Additionally, the project encompasses addressing data security and privacy concerns to ensure compliance with medical regulations and standards. Ultimately, the goal is to deliver a reliable and efficient system that aids medical professionals in accurately predicting breast cancer outcomes.

4.3 Overview

The Design Document for the "Breast Cancer Prediction" project provides an overview of the system architecture, functionalities, and key design considerations. It outlines the scope of the project, which involves developing an automated breast cancer prediction system using and deep learning algorithms. The document addresses the purpose of the project, which is to create a reliable and efficient system that aids medical professionals in accurately predicting breast cancer outcomes. Additionally, it highlights the importance of data security and privacy, as well as adherence to ethical and legal standards. Overall, the Design Document serves as a comprehensive blueprint for the development and implementation of the breast cancer prediction system.

4.4 Data Design

Database are the storehouses of data used in the software systems.The data is stored in tables inside the database.Several tables are created for the manipulation of the data for the system.Two essential settings for a database are

- Primary Key- The field that is unique for all the record occurrence.

- Foreign Key- The field used to set relation between tables.

Normalization

- Normalization is the process of organizing the data in the database.
- Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate the undesirable characteristics like Insertion, Update and Deletion Anomalies.
- Normalization divides the larger table into the smaller table and links them using relationship.
- The normal form is used to reduce redundancy from the database table.

Chapter 5

5 Development of the System

The "Breast Cancer prediction" system was developed using an agile approach, incorporating deep learning algorithms to provide an accurate and user-friendly solution for breast cancer prediction in medical organizations, replacing traditional prediction methods.

Chapter 6

6 System Testing

Testing is the penultimate step of software development.An elaborate testing of the data is prepared and the system is using the test data,while doing testing,errors are noted and correction is made.The users are trained to operate the developed system.Both hardware and software securities are made to run the developed system successfully.System testing is aimed at ensuring the system works accurately before the live operation commences.

6.1 Test Plan

6.1.1 Scope

This test plan will cover the following testing activities as identified in the testing strategy.

- White Box Testing

White box testing is a testing technique, that examines the program structure and derives test data from the program logic/code. In white box testing, the UI is bypassed. Inputs and outputs of ratings are tested directly at the code level and the results are compared against specifications. It Reveals errors in "hidden" code.

- Black Box Testing

Black - box testing is a method of software testing that examines the functionality of an application based on the specifications. Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end - user would.

- Unit Testing

The module includes four main components to undergo unit testing. It checks the syntax error, logic error and validity of the program correctness. This test will be performed by the developers.

- Integration Testing

After all components pass the unit testing, they get to be tested whether they work correctly when they are running concurrently and communicating to each other. Integration tests exercise an entire subsystem and ensure that a set of components play nicely together.

- Internal Data Testing

We will test the validity of the data before it enters the database to avoid any problems that may face us in the database. We will test the encryption of the personal information of all the users along with the user names and passwords to ensure maximum security of the users privacy.

6.1.2 Software risk issues

The "Breast Cancer Prediction" system leverages complex deep learning algorithms to predict breast cancer from histopathological images. While this provides enhanced accuracy compared to traditional methods, it also introduces certain software risks that need to be proactively managed. Key risks include model overfitting on limited training data, algorithmic limitations in capturing real-world variations, integration challenges between system components, performance bottlenecks when dealing with large images, lack of explainability of predictions, and potential security vulnerabilities given the sensitive nature of medical data. Careful software design, extensive testing and validation, performance tuning, security hardening, and adoption of explainable AI techniques are crucial to mitigate these risks. Additionally, bias in training data, software complexity, and obscure failures related to machine learning models require rigorous monitoring and maintenance processes. Overall, the advanced techniques used in Breast Cancer Prediction make software risk management an integral part of developing a robust and reliable solution.

6.1.3 Features to be tested

- Image Upload: Test uploading of histopathology images in different formats, sizes, and resolutions. Verify proper validation and preprocessing.
- User Interface: Test all UI flows and elements like buttons, forms, navigation, and data displays. Check for responsiveness across devices.
- Cancer Prediction: Test prediction accuracy with diverse sample images. Check for correct classification of cancer vs non-cancer.
- Model Training: Validate training process completed without errors. Confirm models are saved properly for prediction.
- Performance: Test system response times and resource usage under different normal and peak load conditions.
- Security: Validate only authorized access to prediction results. Check for data encryption and other security measures.
- Error Handling: Verify proper response for invalid inputs or missing images. Check for fail-safe behavior.
- Integration: Test seamless data flow between UI, preprocessing, and prediction components. Confirm no errors during integration.
- Compatibility: Validate working on different OS versions and hardware configurations per compatibility matrix.
- Functional Flows: Test end-to-end flows like user login, image upload, prediction, and result display for expected behavior.

6.2 Test consolidation

6.2.1 Test item

The items or features to be tested in the test cases are included in the document. Each and every user input is tested. The present condition of the system is tested. It is checked to make sure that environment is ready for the application to work.

6.2.2 Input specifications

These are the inputs required to execute the test case. All required inputs including data elements and values are given below.

Data	Values
Image	histopathological images

Chapter 7

7 System Implementation and Maintenance

7.1 Implementation

System implementation is the conversion of new system into an operating one which involves creating compatible files, training clients and installing hardware. User training is crucial for minimizing resistance to change and giving chance to prove its worth. Training aids user friendly manuals and healthy screens provide the user with a good start. Software maintenance follows conversion to the extent that changes are necessary to maintain satisfactory operations relative to changes in the user's environment. Maintenance often includes minor enhancements or corrections to the problem that surface late in the systems operations. In the implementation phase, the team builds the components either from scratch or by composition. Given the architecture document meant from the design phase and the requirement document from the analysis phase, the team should build exactly what has been requested, though there is still room for innovation and flexibility. For example, a component may be narrowly designed for this particular system, or the component may be made more general to satisfy a reusability

- Careful planning
- Investigation of system and constraints
- Design the methods to achieve changeover.
- Training the staff in the changed phase.
- Evaluation of change over method.
- The method of implementation and time scale to be adopted are found out initially.

7.2 Maintenance

This phase occurs as a result of deploying the whole system at the end users organization. They will perform the beta testing at the end users and inform to the developers about any needed modification to the application. The customer records all the problems that are encountered during the beta testing and reports these to the developer at regular intervals.

7.2.1 Corrective Maintenance

Even with the best quality assurance activities, it is likely that the customer will uncover defects in the software. Corrective maintenance changes the software to correct the defects. Corrective Maintenance activity may consist of repair,

restoration or replacement of equipment. This activity will be result of a regular inspection, which identifies the failure in time for corrective maintenance to be planned and scheduled, then performed during a routine maintenance shutdown.

7.2.2 Adaptive Maintenance

Over time, the original environment(CPU, operating system, business rules, external product characteristics) for which the software was developed is likely to change. Adaptive maintenance results in modification to the software to accommodate changes to its external environment.

7.2.3 Enhanced Maintenance

As software is used, the customer/user will recognize additional functions that will provide the benefit. Perfect maintenance extends the software beyond its original functional requirements.

7.2.4 Preventive Maintenance

Computer software deteriorates due to change, and because of this preventive maintenance often called software re-engineering, must be conducted to enable the software to serve the needs of its end users. Preventive maintenance makes changes to computer programs so that they can be more easily corrected, adapted and enhanced.

Chapter 8

8 Conclusion and Future Scope

8.1 Conclusion

Breast Cancer prediction systems have seen remarkable advancements, providing accurate and efficient solutions for cancer prediction. Looking ahead, the scope for further development is vast, with potential enhancements to enhance accuracy, security, and user experience. Strengthening privacy and security measures are essential aspects to consider. Here, any histopathological images can be given as inputs and optimal result is predicted.vThe integration of AI-based adaptive learning can enhance system accuracy, particularly in challenging environments. Breast cancer prediction systems will continue to be a vital and transformative tool for medical organizations and institutions in the future

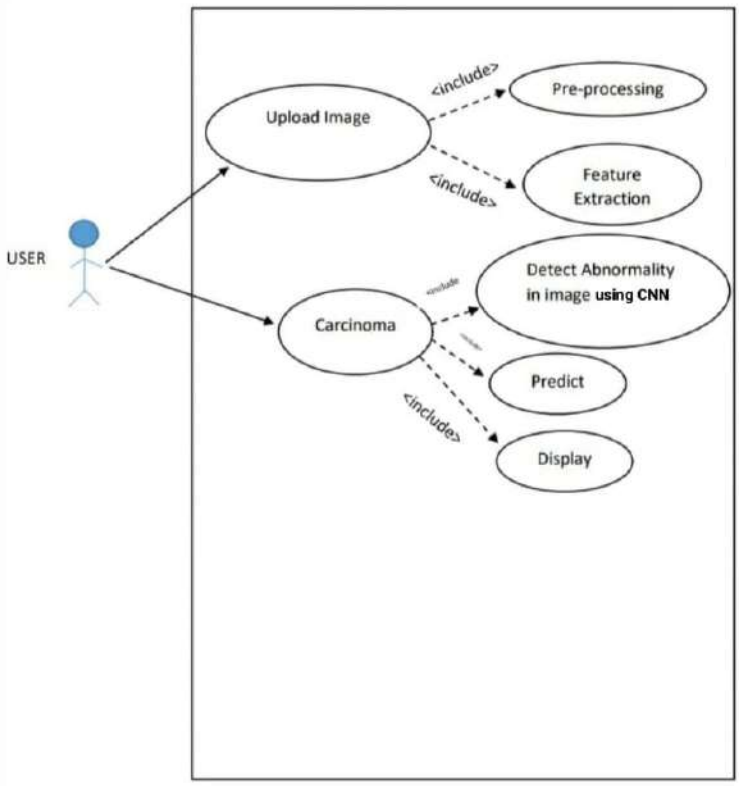
8.2 Future Scope

- Web based platform: Accessible to all the users where they interact with the doctor and results are available online.
- Privacy and Data Protection: Enhancing privacy features and adopting privacy-preserving algorithms will be crucial to safeguarding user data and complying with evolving data protection regulations.
- User Interface and Experience: Investing in user-friendly interfaces and seamless integration with existing systems will encourage greater user acceptance and adoption of the system.
- Real-time Analytics: Developing real-time attendance analytics can provide valuable insights into cancer prediction trends, enabling medical organizations to make data-driven decisions. More image should be trained or the accuracy may vary. Therefore, Several new models should be trained and the more f1 – score giving model should be taken in the future.
- Customization Options: Offering customization options to tailor the system to specific organizational needs and requirements will increase its versatility and adaptability.
- Cross-Platform Compatibility: Ensuring compatibility with various devices and operating systems will make the system accessible and widely applicable in different settings.

Appendix

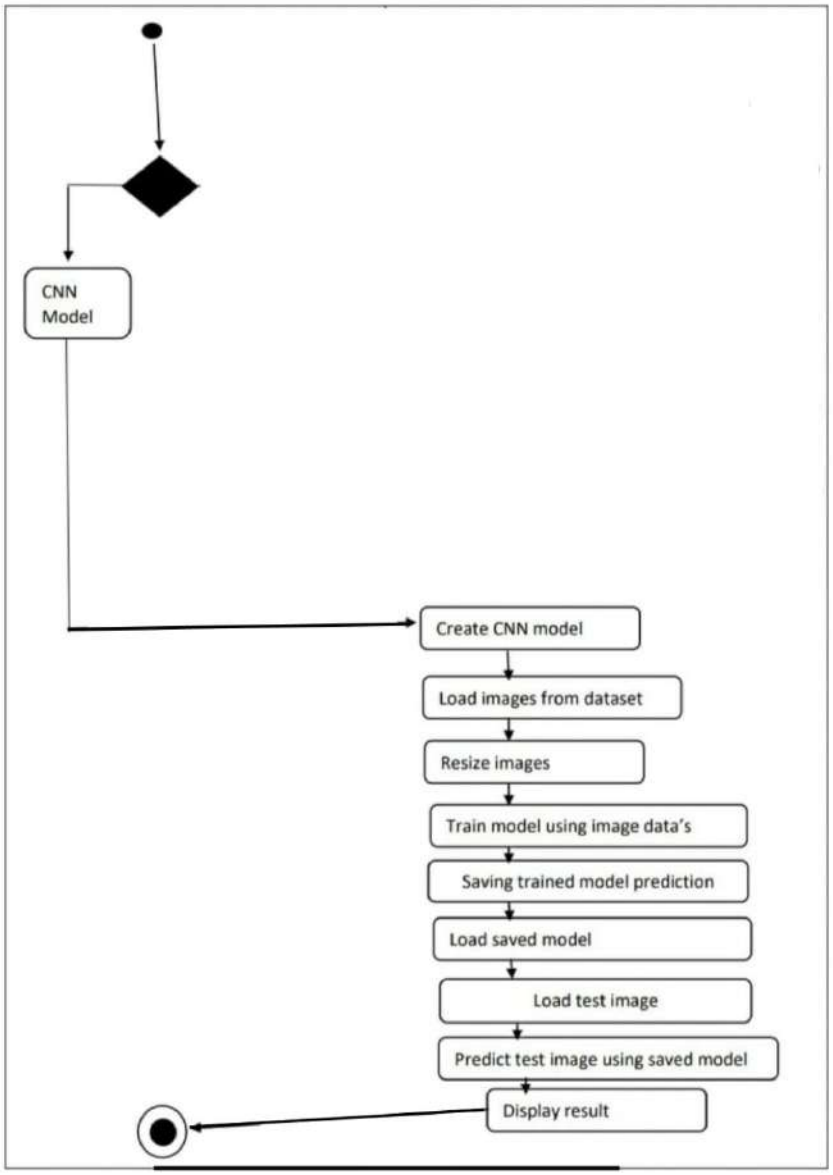
A Usecase Diagram

A.1



B Activity Diagram

B.1



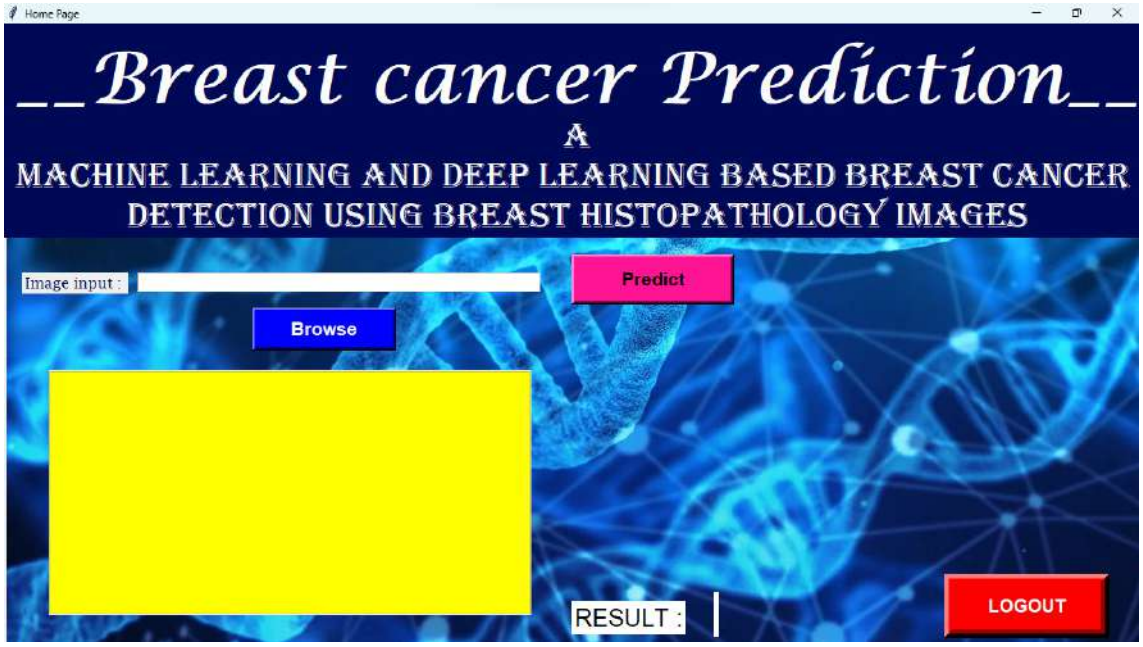
B.2

2. **Breast Cancer**

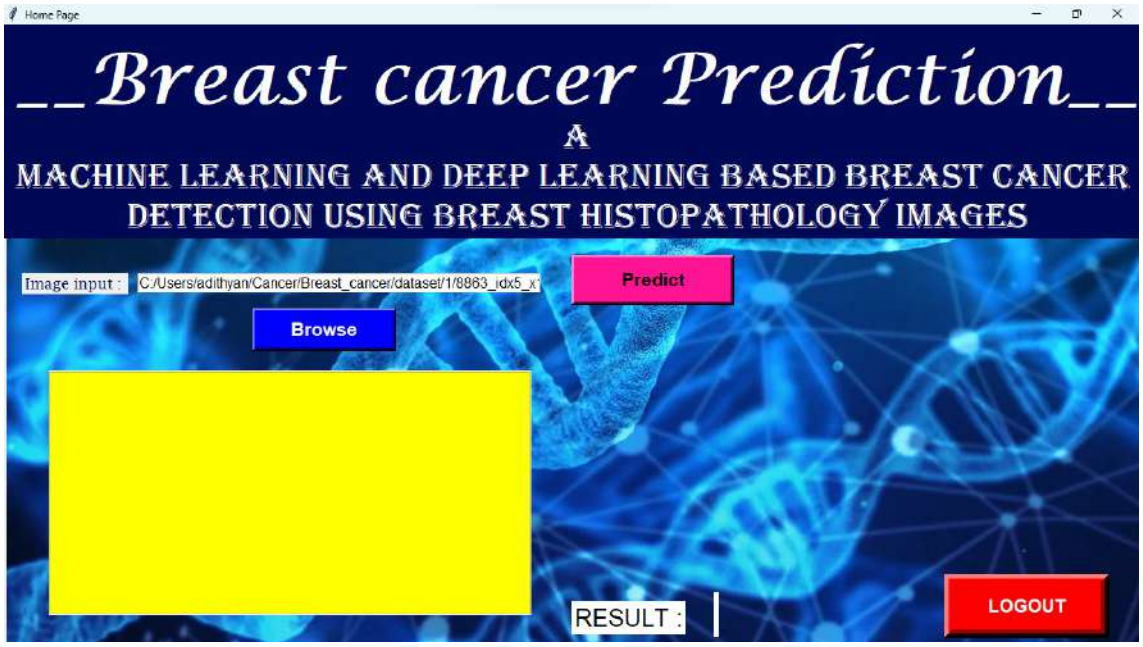
System	**Breast cancer**

C USER INTERFACES

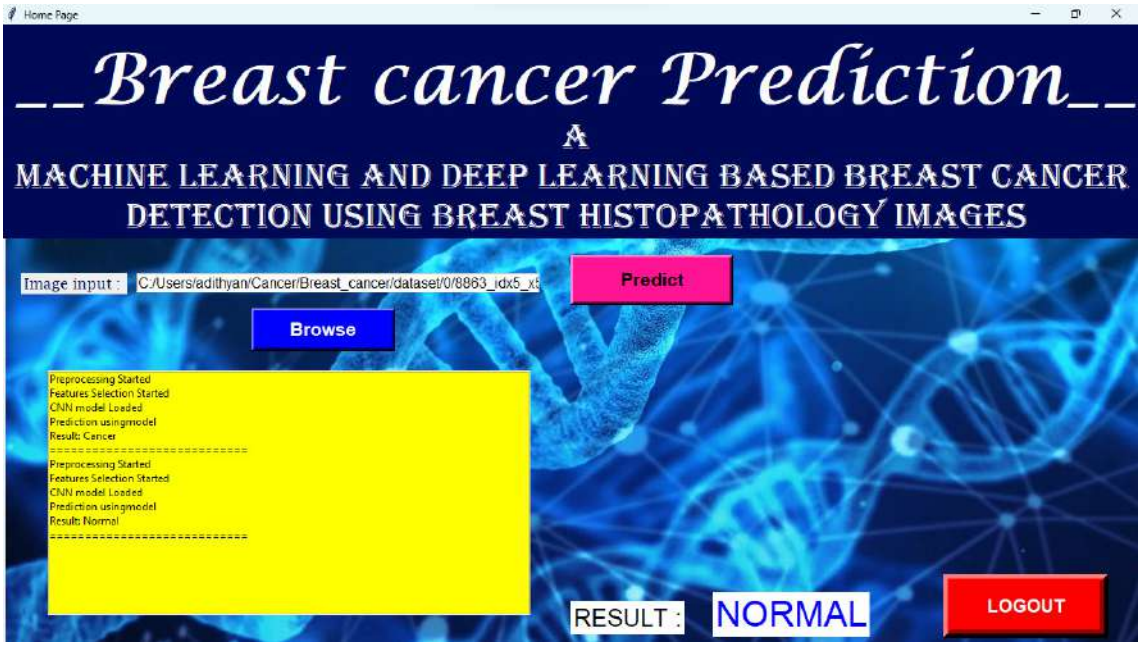
C.1 HOME



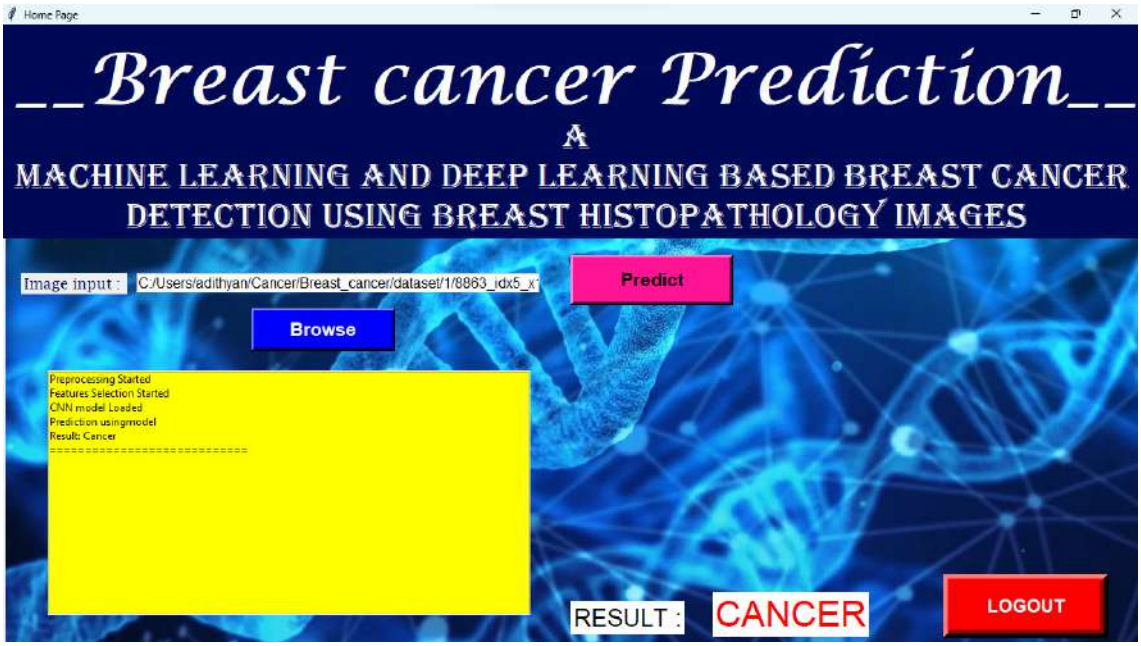
C.2 IMAGE LOCATION



C.3 NORMAL RESULT



C.4 CANCEROUS RESULT



D CODE

preprocess.py

```
import cv2

import numpy as np
import random
from imutils import paths
import os
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
import json
from tensorflow.keras.preprocessing.image import
    img_to_array
from tensorflow.keras.utils import to_categorical
from sklearn import metrics
import pickle

import mahotas as mt
from skimage.filters import gabor

def extract_features(gray):
    textures = mt.features.haralick(gray)
    # cv2.imshow("textures",textures)
    # cv2.waitKey(0)
        # take the mean of it and return it
    ht_mean  = textures.mean(axis=0)
    # print(ht_mean)
    return ht_mean


def build_filters():
        filters = []
        ksize = 31
        for theta in np.arange(0, np.pi, np.pi / 16):
            kern = cv2.getGaborKernel((ksize, ksize),
                4.0, theta, 10.0, 0.5, 0, ktype=cv2.CV_32F
                )
            kern /= 1.5*kern.sum()
            filters.append(kern)
            return filters

def process(img, filters):
        accum = np.zeros_like(img)
```

```
        for kern in filters:
            fimg = cv2.filter2D(img, cv2.CV_8UC3, kern)
            np.maximum(accum, fimg, accum)
            return accum


filters=build_filters()



data = []
label = []
print("[INFO] loading images...")
img_dir=sorted(list(paths.list_images("dataset")))
random.shuffle(img_dir)
print("[INFO]   Preprocessing...")
count=0
for i in img_dir:
    img = cv2.imread(i)
    img=cv2.resize(img, (64,64))
    img1=process(img,filters)
    gray = cv2.cvtColor(img1, cv2.COLOR_BGR2GRAY)
    res2=extract_features(gray)
    # print(res2)
    # print(type(res2))
    # imgdata=img_to_array(res2)
    data.append(res2)
    lb=i.split(os.path.sep)[-2]
    if(lb=="1"):
        label.append(1)
    else:
        label.append(0)
print(len(data))
print(len(label))
```

[breaklines=true]

SvmTrain.py

```
import cv2

import numpy as np
import random
from imutils import paths
import os
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
```

```
import json
from tensorflow.keras.preprocessing.image import
    img_to_array
from tensorflow.keras.utils import to_categorical
from sklearn import metrics
import pickle

import mahotas as mt
from skimage.filters import gabor

def extract_features(gray):
    textures = mt.features.haralick(gray)
    # cv2.imshow("textures",textures)
    # cv2.waitKey(0)
        # take the mean of it and return it
    ht_mean  = textures.mean(axis=0)
    # print(ht_mean)
    return ht_mean


def build_filters():
        filters = []
        ksize = 31
        for theta in np.arange(0, np.pi, np.pi / 16):
            kern = cv2.getGaborKernel((ksize, ksize),
                4.0, theta, 10.0, 0.5, 0, ktype=cv2.CV_32F
                )
            kern /= 1.5*kern.sum()
            filters.append(kern)
            return filters

def process(img, filters):
        accum = np.zeros_like(img)
        for kern in filters:
            fimg = cv2.filter2D(img, cv2.CV_8UC3, kern)
            np.maximum(accum, fimg, accum)
            return accum


filters=build_filters()



data = []
label = []
print("[INFO] loading images...")
```

```
img_dir=sorted(list(paths.list_images("dataset")))
random.shuffle(img_dir)
print("[INFO]   Preprocessing...")
count=0
for i in img_dir:
    img = cv2.imread(i)
    img=cv2.resize(img, (64,64))
    img1=process(img,filters)
    gray = cv2.cvtColor(img1, cv2.COLOR_BGR2GRAY)
    res2=extract_features(gray)
    # print(res2)
    # print(type(res2))
    # imgdata=img_to_array(res2)
    data.append(res2)
    lb=i.split(os.path.sep)[-2]
    if(lb=="1"):
        label.append(1)
    else:
        label.append(0)
print(len(data))
print(len(label))
lenofimage=len(data)
traindata=np.array(data).reshape(lenofimage,-1)
trainlabel=np.array(label)
print(traindata.shape)
print(trainlabel.shape)
X_train, X_test, y_train, y_test = train_test_split(
    traindata, trainlabel, test_size=0.25, random_state
    =42)
print(X_train.shape)
print(y_train.shape)
SVMclassifier = SVC(kernel='linear', random_state=0)
SVMclassifier.fit(X_train, y_train)
# Saving Trained Model
# dbfile=open("SVMmodel","wb")
# pickle.dump(SVMclassifier,dbfile)
# dbfile.close()
y_pred= SVMclassifier.predict(X_test)
print(y_pred)
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))

# 62.38
```

[breaklines=true] **Svmtest.py**

```
import cv2
```

```
import numpy as np
import random
from imutils import paths
import os
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
import json
from tensorflow.keras.preprocessing.image import
    img_to_array
from tensorflow.keras.utils import to_categorical
from sklearn import metrics
import pickle

import mahotas as mt
from skimage.filters import gabor



def extract_features(gray):
    textures = mt.features.haralick(gray)
    # cv2.imshow("textures",textures)
    # cv2.waitKey(0)
        # take the mean of it and return it
    ht_mean  = textures.mean(axis=0)
    # print(ht_mean)
    return ht_mean


def build_filters():
        filters = []
        ksize = 31
        for theta in np.arange(0, np.pi, np.pi / 16):
            kern = cv2.getGaborKernel((ksize, ksize),
                4.0, theta, 10.0, 0.5, 0, ktype=cv2.CV_32F
                )
            kern /= 1.5*kern.sum()
            filters.append(kern)
            return filters

def process(img, filters):
        accum = np.zeros_like(img)
        for kern in filters:
            fimg = cv2.filter2D(img, cv2.CV_8UC3, kern)
            np.maximum(accum, fimg, accum)
```

```
            return accum


# img=cv2.imread("im1.JPG")


filters=build_filters()


img = cv2.imread("31.png")
img=cv2.resize(img, (64,64))
img1=process(img,filters)
gray = cv2.cvtColor(img1, cv2.COLOR_BGR2GRAY)
res2=extract_features(gray)

dbfile=open("SVMmodel","rb")
adamodel=pickle.load(dbfile)
dbfile.close()
# Prediction based on selected model
y_pred = adamodel.predict(res2.reshape(1,-1))
print(y_pred)
```

[breaklines=true] **CNNTrain.py**

```
import cv2

import numpy as np
import random
from imutils import paths
import os
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
import json
from tensorflow.keras.preprocessing.image import
    img_to_array
from tensorflow.keras.utils import to_categorical
from sklearn import metrics
import pickle
import pandas as pd
import numpy as np
import os
from glob import glob
import random
import matplotlib.pylab as plt
from tensorflow.keras.models import model_from_json
```

```
from sklearn.model_selection import train_test_split
import tensorflow as tf
import keras
from keras.utils.np_utils import to_categorical
from keras.models import Sequential
from keras.preprocessing.image import ImageDataGenerator

from tensorflow.keras.optimizers import SGD, RMSprop,
    Adam, Adagrad, Adadelta
from keras.layers import Dense, Dropout, Activation,
    Flatten, BatchNormalization, Conv2D, MaxPool2D,
    MaxPooling2D
from tensorflow.keras.callbacks import EarlyStopping


data = []
label = []
print("[INFO] loading images...")
img_dir=sorted(list(paths.list_images("dataset")))
random.shuffle(img_dir)
print("[INFO]  Preprocessing...")
count=0
for i in img_dir:
    img = cv2.imread(i)
    img=cv2.resize(img, (64,64))
    imgdata=img_to_array(img)
    data.append(imgdata)
    lb=i.split(os.path.sep)[-2]
    if(lb=="1"):
        label.append(1)
    else:
        label.append(0)

traindata=np.array(data)
trainlabel=np.array(label)

X_train, X_test, y_train, y_test = train_test_split(
    traindata, trainlabel, test_size=0.25, random_state
    =42)
print(X_train.shape)
print(y_train.shape)

model = Sequential()

# Convolutional layer and maxpool layer 1
model.add(Conv2D(32,(3,3),activation='relu',input_shape
```

```
    =(64,64,3)))
model.add(MaxPool2D(2,2))

# Convolutional layer and maxpool layer 2
model.add(Conv2D(64,(3,3),activation='relu'))
model.add(MaxPool2D(2,2))

# Convolutional layer and maxpool layer 3
model.add(Conv2D(128,(3,3),activation='relu'))
model.add(MaxPool2D(2,2))

# Convolutional layer and maxpool layer 4
model.add(Conv2D(128,(3,3),activation='relu'))
model.add(MaxPool2D(2,2))

# This layer flattens the resulting image array to 1D
    array
model.add(Flatten())

# Hidden layer with 512 neurons and Rectified Linear Unit
     activation function
model.add(Dense(512,activation='relu'))

# Output layer with single neuron which gives 0 for Cat
    or 1 for Dog
#Here we use sigmoid activation function which makes our
    model output to lie between 0 and 1
model.add(Dense(1,activation='sigmoid'))

# model = Sequential()
# model.add(Conv2D(64, kernel_size=3, activation='relu',
    input_shape=(50,50,3)))
# model.add(Conv2D(32, kernel_size=3, activation='relu'))
# model.add(Flatten())
# model.add(Dense(2, activation="softmax"))
# adam = Adam(learning_rate=0.0001)
model.compile(loss="binary_crossentropy", optimizer="adam
    ", metrics=['accuracy'])

history = model.fit(
    X_train, y_train,
    validation_data=(X_test, y_test),
    epochs= 10,
    batch_size=10
    )
Y_pred = model.predict(X_test)
```

```
print(Y_pred)
#serialize model to JSON
model_json = model.to_json()
with open("CNNmodel.json", "w") as json_file:
        json_file.write(model_json)
# serialize weights to HDF5
model.save_weights("CNNmodelw.h5")
print("Saved model to disk")
#0.9133
```

[breaklines=true]
CNNTest.py

```
from tensorflow.keras.preprocessing.image import
    img_to_array
from tensorflow.keras.models import load_model
import numpy as np
import argparse
import imutils
import cv2
from tensorflow.keras.models import model_from_json


img = cv2.imread("10.png")
img=cv2.resize(img, (64,64))
img = img_to_array(img)
##img = img.reshape(1, 100, 36, 1)

json_file = open('CNNmodel.json', 'r')
loaded_model_json = json_file.read()
json_file.close()
model = model_from_json(loaded_model_json)
# load weights into new model
model.load_weights("CNNmodelw.h5")
print("Loaded model from disk")
img = np.expand_dims(img, axis = 0)
result = model.predict(img)
res1=result[0][0]
if(res1>0.32):
    print("Cancer")
else:
    print("Normal")
print("res=>",result[0][0])
result=np.argmax(result[0][0])
print(result)
```

[breaklines=true] **Resnet_train.py**

```
import numpy as np
import random
import cv2
from glob import glob
import matplotlib.pyplot as plt
from imutils import paths
import os
from PIL import Image
import keras
from tensorflow.keras.preprocessing.image import
    ImageDataGenerator
from tensorflow.keras.applications import ResNet50
from tensorflow.keras.applications.resnet50 import
    preprocess_input
from tensorflow.keras import Model, layers
from tensorflow.keras.models import load_model,
    model_from_json
from tensorflow.keras.preprocessing.image import
    img_to_array
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.models import model_from_json

from sklearn.model_selection import train_test_split



conv_base = ResNet50(
    include_top=False,
    weights='imagenet')

for layer in conv_base.layers:
    layer.trainable = False
x = conv_base.output
x = layers.GlobalAveragePooling2D()(x)
x = layers.Dense(128, activation='relu')(x)
predictions = layers.Dense(2, activation='softmax')(x)
model = Model(conv_base.input, predictions)

model.compile(loss='sparse_categorical_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])


data = []
label = []
```

```
print("[INFO] loading images...")
img_dir=sorted(list(paths.list_images("dataset")))
random.shuffle(img_dir)
print("[INFO]   Preprocessing...")
count=0
for i in img_dir:
    img = cv2.imread(i)
    img=cv2.resize(img,(256,256))
    imgdata=img_to_array(img)
    data.append(imgdata)
    lb=i.split(os.path.sep)[-2]
    if(lb=="0"):
        label.append(0)
    else:
        label.append(1)
print(len(data))
print(len(label))
lenofimage=len(data)
traindata=np.array(data)
trainlabel=np.array(label)
train_labels = to_categorical(trainlabel, num_classes=2)
print(traindata.shape)
print(train_labels.shape)
X_train, X_test, y_train, y_test = train_test_split(
    traindata, trainlabel, test_size=0.25, random_state
    =42)
print(X_train.shape)
print(y_train.shape)
print(X_test.shape)
print(y_test.shape)
```

```
model.fit(X_train, y_train, epochs=3, validation_split
    =0.2, batch_size=5)

model_json = model.to_json()
with open("resnet_model.json", "w") as json_file:
        json_file.write(model_json)
# serialize weights to HDF5
model.save_weights("resnetw_model.h5")
print("[INFO] Saved model to disk")
y=model.predict(X_train)
print(y)
```

[breaklines=true] **Densenettrain.py**

```
import numpy as np
import random
import cv2
from glob import glob
import matplotlib.pyplot as plt
from imutils import paths
import os
from PIL import Image

import keras
from tensorflow.keras.preprocessing.image import
    ImageDataGenerator
from tensorflow.keras.applications.densenet import
    DenseNet121
from keras import Model, layers
from keras.models import load_model, model_from_json
from tensorflow.keras.preprocessing.image import
    img_to_array
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.models import model_from_json
from tensorflow.keras.layers import Dense,
    GlobalAveragePooling2D,Convolution2D,
    BatchNormalization
from tensorflow.keras.layers import Flatten,MaxPooling2D,
    Dropout
from sklearn.model_selection import train_test_split




from tensorflow.keras.layers import Dense, Dropout, Input
    ,Flatten, Conv2D, MaxPool2D, BatchNormalization,
    AveragePooling2D, GlobalAveragePooling2D
from tensorflow.keras.models import Model,Sequential,
    load_model
from tensorflow.keras.optimizers import Adam
def build_densenet():
    densenet = DenseNet121(weights='imagenet',
        include_top=False)

    input = Input(shape=(256, 256, 3))
```

```
    x = Conv2D(3, (3, 3), padding='same')(input)

    x = densenet(x)

    x = GlobalAveragePooling2D()(x)
    x = BatchNormalization()(x)
    x = Dropout(0.5)(x)
    x = Dense(256, activation='relu')(x)
    x = BatchNormalization()(x)
    x = Dropout(0.5)(x)

    # multi output
    output = Dense(15,activation = 'softmax', name='root
        ')(x)


    # model
    model = Model(input,output)

    optimizer = Adam(lr=0.002, beta_1=0.9, beta_2=0.999,
        epsilon=0.1, decay=0.0)
    model.compile(loss='sparse_categorical_crossentropy',
         optimizer=optimizer, metrics=['accuracy'])
    model.summary()

    return model

model = build_densenet()



data = []
label = []
print("[INFO] loading images...")
img_dir=sorted(list(paths.list_images("dataset")))
random.shuffle(img_dir)
print("[INFO]   Preprocessing...")
count=0
for i in img_dir:
    img = cv2.imread(i)
    img=cv2.resize(img,(256,256))
    imgdata=img_to_array(img)
    data.append(imgdata)
    lb=i.split(os.path.sep)[-2]
    if(lb=="0"):
        label.append(0)
```

```
    else:
        label.append(1)
print(len(data))
print(len(label))
lenofimage=len(data)
traindata=np.array(data)
trainlabel=np.array(label)
train_labels = to_categorical(trainlabel, num_classes=2)
print(traindata.shape)
print(train_labels.shape)
X_train, X_test, y_train, y_test = train_test_split(
    traindata, trainlabel, test_size=0.25, random_state
    =42)
print(X_train.shape)
print(y_train.shape)
print(X_test.shape)
print(y_test.shape)


model.fit(X_train, y_train, epochs=10, validation_split
    =0.2, batch_size=5,verbose=1)

# model_json = model.to_json()
# with open("densenet_model.json", "w") as json_file:
#         json_file.write(model_json)
# # serialize weights to HDF5
# model.save_weights("densenetw_model.h5")
# print("[INFO] Saved model to disk")
```

[breaklines=true] **GUI_new.py**

```
from tkinter import *
import time
import re
#Import scikit-learn metrics module for accuracy
    calculation
import pickle
from PIL import Image, ImageTk
import cv2
from tkinter.filedialog import askopenfile
from tensorflow.keras.preprocessing.image import
    img_to_array
from tensorflow.keras.models import load_model
import numpy as np
import argparse
import imutils
```

```
import cv2
from tensorflow.keras.models import model_from_json
from tensorflow.keras.preprocessing.image import
    img_to_array
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.applications.vgg16 import
    preprocess_input
from tensorflow.keras.models import model_from_json

json_file = open('resnet_model.json', 'r')
loaded_model_json = json_file.read()
json_file.close()
modelres = model_from_json(loaded_model_json)
# load weights into new model
modelres.load_weights("resnetw_model.h5")
print("Loaded Resnet model from disk")

json_file = open('densenet_model.json', 'r')
loaded_model_json = json_file.read()
json_file.close()
modeldense = model_from_json(loaded_model_json)
# load weights into new model
modeldense.load_weights("densenet.h5")

print("Loaded Sensenet model from disk")

json_file = open('CNNmodel.json', 'r')
loaded_model_json = json_file.read()
json_file.close()
cnnmodel = model_from_json(loaded_model_json)
# load weights into new model
cnnmodel.load_weights("CNNmodelw.h5")
print("Loaded model from disk")

def pp(a):
    global mylist
    mylist.insert(END, a)



def predict(val):
    print(val)
    img = cv2.imread(val)
    imgr=cv2.resize(img, (64,64))
    imgarr = img_to_array(imgr)
    ##img = img.reshape(1, 100, 36, 1)
```

```
imgarr = np.expand_dims(imgarr, axis = 0)
result = cnnmodel.predict(imgarr)
res1=result[0][0]
print(res1)

if(res1>0.32):
    cnnres=1
else:
    cnnres=0
print("cnnpred==>",cnnres)
imgres=cv2.resize(img,(256,256))
imgdata=img_to_array(imgres)
print(imgdata)
# print(type(imgdata))
# print(imgdata.shape)

train_ds= np.expand_dims(imgdata, axis=0)
# print(train_ds.shape.)

yres=modelres.predict(train_ds)
print(yres)
resres=np.argmax(yres[0])
print("respred==>",resres)

ydense=modeldense.predict(train_ds)
print(ydense)
denseres=np.argmax(ydense[0])
print("densepred==>",denseres)

reslist=[cnnres,resres,denseres]

print("result list==>",reslist)
finalres=max(reslist)
if(finalres==1):
    resc="Cancer"
    print("Cancer")
    root.after(3100, lambda :shrslt.config(text="
        CANCER",fg="red"))

else:
    resc="Normal"
    print("Normal")
    root.after(3100, lambda :shrslt.config(text="
        NORMAL",fg="blue"))
```

```
    root.after(500, lambda : pp("Preprocessing Started ")
        )
    root.after(2000, lambda : pp("Features Selection
        Started "))
    root.after(2200, lambda : pp("CNN model Loaded"))
    root.after(2400, lambda : pp("Prediction usingmodel
        "))
    root.after(2600, lambda : pp("Result: "+resc))
    root.after(2800, lambda : pp
        ("=========================================="))


def browseim():
    global cimg,shrslt,E1
    path = askopenfile()
    n=path.name
    print(path)
    E1.delete(0,"end")
    E1.insert(0, n)

#def upload_file():
 #    global cimg,shrslt,E1
 #    root=Tk()
 #    f_types=[('Png Files','*.png')]
 #    filename=filedialog.askopenfilename(filetypes=
     f_types)
 #   cimg=ImageTk.PhotoImage(file=filename)

def userHome():
    global root, mylist,shrslt,E1
    root = Tk()
    root.geometry("1400x625+0+0")
    root.title("Home Page")

    image = Image.open("dna.png")
    image = image.resize((1500, 725), Image.ANTIALIAS)
    pic = ImageTk.PhotoImage(image)
    lbl_reg=Label(root,image=pic,anchor=CENTER)
    lbl_reg.place(x=0,y=0)

    #----------------------INFO TOP----------------
    lblinfo = Label(root, font=( 'lucida calligraphy'
        ,61, 'bold' ),text="__Breast cancer Prediction__",
        fg="white",bg="#000955",bd=10,anchor='w')
```

```
lblinfo.place(x=0,y=0)

lblinfo2 = Label(root, font=( 'Algerian' ,31 ),text="
    A\n Machine learning and Deep learning based
    Breast cancer \n   detection using Breast
    Histopathology Images   ",fg="white",bg="#000955",
    anchor='w')
lblinfo2.place(x=0,y=100)
lblinfo3 = Label(root, font=( 'Lucida fax' ,0 ),text
    ="Image input : ",fg="#000955",anchor='w')
lblinfo3.place(x=20,y=280)
E1 = Entry(root,width=50,font="Will&Grace")
E1.place(x=150,y=280)
mylist = Listbox(root,width=90, height=17,bg="yellow
    ")

mylist.place( x = 50, y = 390 )
btntrn=Button(root,padx=10,pady=2, bd=4 ,fg="white",
    font=('ariel' ,16,'bold'),width=10, text="Browse",
     bg="blue",command=lambda:browseim())
btntrn.place(x=280, y=320)
# btntrn=Button(root,padx=10,pady=2, bd=4 ,fg="white",
   font=('ariel' ,16,'bold'),width=10, text="image",
   bg="red",command=lambda:upload_file())
#btntrn.place(x=700, y=400)
btnhlp=Button(root,padx=40,pady=6, bd=4 ,fg="black",
    font=('ariel' ,16,'bold'),width=7, text="Predict",
     bg="deep pink",command=lambda:predict(E1.get()))
btnhlp.place(x=640, y=260)



rslt = Label(root, font=( 'aria' ,20, ),text="RESULT
    :",fg="black",bg="white",anchor=W)
rslt.place(x=640,y=650)
shrslt = Label(root, font=( 'aria' ,30, ),text="",fg
    ="blue",bg="white",anchor=W)
shrslt.place(x=800,y=640)

def qexit():
    root.destroy()

btnexit=Button(root,padx=16,pady=8, bd=10 ,fg="white
    ",font=('ariel' ,16,'bold'),width=10, text="LOGOUT
    ", bg="red",command=qexit)
btnexit.place(x=1060, y=620)
```

```
    root.mainloop()

userHome()
```

[breaklines=true]

CAR TRAFFIC SIGN RECOGNIZER

PROJECT REPORT

Submitted By

AKHILASH MURALEEDHARAN

Reg. No. CCAVBCA003

for the award of the Degree of

Bachelor (BCA)

of Computer Application
(University of Calicut)

under the guidance of

Ms.Varsha Ganesh

Assistant Professor

BACHELOR OF COMPUTER APPLICATION
DEPARTMENT OF COMPUTER SCIENCE
CHRIST COLLEGE(AUTONOMOUS)
IRINJALAKUDA, KERALA
INDIA

2021-2024

DEPARTMENT OF COMPUTER SCIENCE

CHRIST COLLEGE (AUTONOMOUS)

IRINJALAKUDA

CERTIFICATE

This is to certify that the project report entitled ***"Car Traffic Sign Recognizer"*** *is a bonfied record of the project work done by* ***Akhilash Muraleedharan*** *in partial fulfillment of the requirement for the sixth semester of* ***Bachelor of Computer Application*** *in Department of Computer Science of* **CHRIST COLLEGE (AUTONOMOUS) IRINJALAKUDA**.

Ms.Varsha Ganesh
Assistant Professor
Internal Guide

Ms.Sini Thomas
Head of the Department
Computer Science

EXTERNAL EXAMINER

INTERNAL EXAMINER

DECLARATION

I hereby declare that this project work **"CAR TRAFFIC SIGN RECOGNIZER"** submitted to Christ College (Autonomous) Irinjalakuda, affiliated to Calicut University in partial fulfillment of the requirement for the award of the Bachelor of Computer Application, is a record of original work done by me, under the guidance of Ms.VARSHA GANESH, Department of Computer Science.

Place: Irinjalakuda AKHILASH MURALEEDHARAN

ACKNOWLEDGEMENT

First and foremost, I would like to thank Lord almighty for his providence and for being the guiding light throughout the project. I wish to express my sincere gratitude to our beloved Department for giving us all the facilities for our project. I take this opportunity to express my gratitude to the class teacher Ms.SOUMYA P.S and the Head of the Department Ms.SINI THOMAS who has supported us throughout the course of this project. I am thankful for their aspiring guidance and valuable advices during the project work. I express my sincere thanks to our project guide Ms.VARSHA GANESH for supporting and guiding us throughout the project. I would take this opportunity to specially thank all other faculty members for their constant and continuous motivation. Finally, I would like to thank my family and friends for giving valuable advices and moral support throughout our project.

ABSTRACT

CAR TRAFFIC SIGN RECOGNIZER is a website designed to helps the driver about recognition of road signs to avoid road accidents. It represents an important feature of advanced driver assistance system, contributing to the safety of the drivers, autonomous vehicles as well and to increase driving comfort. The website has three kind of login facilities :- admin login, user login and driver login. All these features make this website more adaptable and user-friendly.

Contents

Chapter 1

1 Introduction

In today's world road conditions drastically improved as compared with past decades. Obviously, vehicle's speed increased. So, on driver's point of view there might be chances of neglecting mandatory road signs while driving. This project explores the system to helps the driver about recognition of road signs to avoid road accidents. An automatic means of detecting and recognizing traffic signs can make a significant contribution to this goal by providing a fast method of detecting, classifying and logging signs. This method helps to develop the inventory accurately and consistently.

The project is mainly focused on automatic recognition of warning signs placed in local roads captured by image clips. The system classifies the traffic signs present in the image into different categories. With this model, we can read and understand traffic signs which are very important task for autonomous vehicles.

1.1 Overview

The objective of the Car Traffic Sign Recognizer website is to design a simple and adaptable website which helps users to increase transportation efficiency, road safety and to reduce the environmental impact with the use of advanced communication technologies. The road signs are placed on either roadside or above the roads. These signs provide mandatory information regarding to guiding, warning and regulating the behaviors to drivers in order to make driving safer and easier. A system is developed to assist the drivers, based on detection and recognition of signs which corrects the most unsafe driving behaviors.

Chapter 2

2 System Analysis

2.1 Purpose

The purpose of the traffic sign recognition project is to develop a sophisticated system that can automatically detect, interpret, and respond to traffic signs in real-time. This project aims to empower vehicles with the ability to perceive and understand the various types of traffic signs encountered on roadways. Ultimately, the goal of this project is to create a robust and reliable solution for enhancing the awareness and decision-making capabilities of drivers and autonomous vehicles alike.

2.1.1 Existing System

Existing systems of traffic sign recognition utilize a combination of cameras, sensors, and sophisticated image processing algorithms to detect and interpret traffic signs in real-time. By capturing images of traffic signs and analyzing them these systems can recognize various types of signs, including speed limits, stop signs, and lane restrictions. Limitations of existing system : The system may still encounter challenges in accurately detecting and interpreting traffic signs, leading to false detections or missed signs, potentially impacting driver safety.

2.1.2 Proposed System

The proposed system of traffic sign recognition aims to address the current limitations by leveraging advancements in artificial intelligence, particularly deep learning algorithms. By training deep neural networks on large datasets of annotated traffic sign images, the system would learn to accurately detect and interpret various types of signs, including speed limits, stop signs, and lane restrictions, across diverse environmental conditions.

2.2 Problem definition

To know what the problem is and what the needs are before developing it. Here we design Car Traffic Sign Recognizer to detect the traffic signs accurately.

2.3 Feasibility Study

After the problem is clearly understood and solutions proposed, the next step is to conduct the feasibility study. Feasibility study is defined as evaluation or analysis of the potential impact of the proposed project or program. The objective is to determine whether the proposed system is feasible. There are three aspects of feasibility study to which the proposed system is subjected as discussed below.

2.3.1 Technical Feasibility

Technical feasibility assess whether the current technical resources are sufficient for the new system. We can upgrade the level of technology for supporting our website. We check whether the proposed system can be implemented in the present system without supporting the existing hardware.

2.3.2 Economical Feasibility

Economic feasibility determines whether the time and money are available to develop the system. There is no additional hardware used to develop the site, it is inexpensive to build.

2.3.3 Operational Feasibility

Operational feasibility determines if the human resources are available to operate the system once it has been installed. No extra training is needed to use this system. Anyone who has knowledge in internet in english language can easily use the system. The resources that are required to implement or install are already available with the organization.

Chapter 3

3 Software Requirement Specification

3.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the CAR TRAFFIC SIGN RECOGNIZER. It illustrate the purpose and complete description for the development of the system. It explain system constraints, interface and interactions with other external applications.

3.2 Scope

The scope of car traffic sign recognizer extends to improving road safety, enhancing driver assistance technologies, optimizing traffic flow, and advancing the capabilities of autonomous vehicles and smart transportation systems.

3.3 Overall Description

This section give an overview of our website, CAR TRAFFIC SIGN RECOGNIZER. This project aims to develop an intelligent system capable of automatically detecting, recognizing, and interpreting traffic signs from images captured by cameras mounted on vehicles. Upon detection, the system may provide visual or auditory alerts to drivers, contributing to improved road safety and compliance with traffic laws. It plays a vital role in enhancing driver assistance systems, navigation, and autonomous vehicle technologies, ultimately leading to safer and more efficient transportation systems.

3.3.1 Product Perspective

The car traffic sign recognition project aims to develop a robust and user-friendly system that enhances driver safety and convenience. The project can develop a comprehensive solution that enhances road safety, improves driver assistance technologies, and advances the capabilities of autonomous vehicles and smart transportation systems.

3.3.2 Product Functionality

The system provides a comprehensive solution that enhances road safety, improves driver assistance technologies, and advances the capabilities of autonomous vehicles and smart transportation systems by detecting the traffic signs and notifying the driver.

3.3.3 Users and Characteristics

There are three types of users that interact with the site - admin, user and driver. Each of these have different tasks which is performed. Users are able to detect the traffic signs, view drivers and book drivers. Drivers can view the bookings and approve it. Admin is able view all the users and drivers and has the authority to delete the users or drivers.

3.4 Specific Requirements

3.4.1 Hardware Requirements

- System: LAPTOP-TE3E25EG
- Processor: i3 8th Gen or above
- Speed: 2.80 GHz
- RAM capacity: 8 GB
- Hard Dsk drive: 128 GB
- Keyboard: Standard
- Mouse: Standard
- Monitor: 18.5” LED Monitor

3.4.2 Software Requirements

- Operating System: Windows
- Languages used: Python, Django
- Database : SQLite3
- Technologies used: HTML,Javascript,CSS

3.5 Functional Requirements

It contains three main modules.

- 1.Admin
- 2.User
- 3.Driver

Admin

An Admin account is used for editing or managing the website dynamically by Admin panel. The admin can view the users and drivers and has the authority to delete any user or driver.

User

The user can detect the traffic signs. There is a feature providing for booking a driver and viewing the bookings.

Driver

The driver can view the bookings and user details. The driver can approve the booking.

3.6 Non Functional Requirements

Non-functional requirements define the overall qualities or attributes of the resulting system. Non-functional system place restrictions on the product being developed, the developed process, and specify external constraints that the product must meet. Examples of non-functional requirements include safety, security, usability, reliability and performance requirements. Project management issues (costs, time and schedule) are often considered as non-functional requirements. The principle non-functional constraints which are relevant to critical systems :

- performance
- security
- safety
- usability

Performance

Performance requirements concern the speed of operation of a system. Types of performance requirements :

- Response requirements (how quickly the system reacts to a user input).
- Throughput requirements (how much the system can accomplish within a specified amount of time).
- Availability requirements (is the system available for service when requested by end users). The speed of operation of this system is adequate for the requirements.

Reliability

- Reliability is the ability of a system to perform its required function under stated conditions for a specified period of time.
- Constraints on runtime behavior of the system. This system is reliable because its functionalities can be done on the required conditions.

Safety

Safety requirements are not required which exclude unsafe situation from the possible solution of the system.

Usability

Usability is the ease with which a user can learn to operate, prepare inputs for, and interpret outputs of system or components.
Usability requirements include :

- information error messages.
- well-formed user interfaces.

3.7 Interface Requirements

3.7.1 Hardware interfaces

The system must run over the internet, all the hardware shall require to connect internet will be hardware interface for the system. As for example modem, WAN-LAN.

3.7.2 Software interfaces

Software interface required for the working of the project is the appropriate operating system.

3.7.3 Communication interfaces

The user of the site communicate using the network connectivity and the data set is accessed

3.8 Security Requirements

- User accesses only their account.
- Validation of input is handled.
- This application containing the computer systems is physically secured against arms or surreptitious entry by intruders.
- Users must be authorized carefully to reduce changes of any such user giving access to an intruder in exchange for a bribe or other favour.

3.9 Platform Used

Windows 11 is the latest iteration of Microsoft's operating system, introducing a refreshed user interface, enhanced productivity features, and improved performance. With a sleek design aesthetic centered around simplicity and productivity, Windows 11 offers users a more intuitive and personalized computing experience.Windows 11 represents Microsoft's vision for the future of computing, blending familiarity with innovation to empower users to do more. Windows 11 brings performance improvements, with optimizations for better energy efficiency and faster wake times. With support for gaming features like DirectStorage and Auto HDR, along with enhanced security measures, Windows 11 aims to cater to a wide range of users, from casual consumers to power users and professionals, ushering in a new era of computing excellence.

3.10 Technologies Used

Python

Python is a high-level programming language renowned for its simplicity, versatility, and readability. With its clean and concise syntax, Python is accessible to beginners while remaining powerful enough for complex applications in various domains, including web development, data analysis, machine learning, artificial intelligence, and automation.Its extensive standard library and robust ecosystem of third-party packages make it a favorite among developers for rapid prototyping, building scalable applications, and solving a wide range of computational problems. Python's interpreted nature and cross-platform compatibility further contribute to its popularity, enabling developers to write code once and run it on multiple platforms without modification. Overall, Python's ease of use, flexibility, and community support make it a go-to choice for developers worldwide.

SQLite3

SQLite3 is a lightweight, self-contained, serverless relational database engine that is widely used for embedded systems, mobile applications, and small-scale database-driven websites. It is part of the SQLite project, which aims to provide a simple, fast, and reliable database solution with minimal setup and administration. SQLite3 is unique in that it operates as a single disk file and requires no configuration or administration, making it extremely easy to deploy and use. Despite its small footprint, SQLite3 supports many

standard SQL features, including transactions, triggers, and views, making it suitable for a wide range of applications. Overall, SQLite3 is an excellent choice for projects requiring a lightweight and efficient database solution without the overhead of a full-fledged database management system.

Chapter 4

4 Design Document

4.1 Purpose

The purpose of a car traffic sign recognizer project is to enhance road safety, improve driver assistance systems, and facilitate the development of autonomous vehicles. By developing an intelligent system capable of automatically detecting, recognizing, and interpreting traffic signs from images captured by onboard cameras, this project aims to provide valuable information to drivers in real-time. The system can alert drivers about speed limits, stop signs, yield signs, and other relevant traffic regulations, helping them make informed decisions and comply with road rules more effectively. Additionally, integrating traffic sign recognition technology into vehicles contributes to the advancement of autonomous driving technologies by enabling vehicles to perceive and understand the surrounding environment, ultimately leading to safer and more efficient transportation systems. Purpose of this document is to give the detailed description of the architecture and system design for the software. The system design document shows the software will be structured to satisfy the requirement identified in the software requirement specification. It is a translation of requirements into a description of the software structure, software component, interfaces and data necessary for the implementation phase. In a complete software design document, each requirements must be traceable to one or more design entities.

4.2 Scope

Car Traffic Sign Recognizer is a website which helps in detecting the traffic signs and alert the driver to ensure safe and comfortable driving.

4.3 Overview

The purpose of this document is to help the reader to visualize the solution to the project presented.This document verifies how the design meet the requirement stimulated in the SRS document through design viewpoints. The design viewpoints will cover all design elements presented before. This document will provide a direct approach to the development

of this project hence reducing feature creep and ponitedly determine the quality of the design.

4.4 Data Design

Database are the storehouses of data used in the software systems.The data is stored in tables inside the database. Several tables are created for the manipulation of the data for the system.Two essential settings for a database are

- Primary Key- The field that is unique for all the record occurrence.
- Foreign Key- The field used to set relation between tables.

Normalization

- Normalization is the process of organizing the data in the database.
- Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate the undesirable characteristics like Insertion, Update and Deletion Anomalies.
- Normalization divides the larger table into the smaller table and links them using relationship.
- The normal form is used to reduce redundancy from the database table.

Tables

textapp_login

Name	DataType	Constraints	Description
id	int	Primarykey	ID of registered people
username	varchar(100)	Notnull	Name of registered people
password	varchar(100)	Notnull	Password of registered people
usertype	varchar(100)	Notnull	Type of registered people
status	varchar(100)	Notnull	Current status of the booking
driverid_id	bigint	Foreignkey	ID of driver

textapp_user

Name	DataType	Constraints	Description
id	int	Primarykey	ID of users
uname	varchar(100)	Notnull	Name of users
uaddress	varchar(100)	Notnul	Address of users
ucontact	varchar(100)	Notnull	Contact number of the users
uemail	varchar(100)	Notnull	Email id of users
upassword	varchar(100)	Notnull	Password of users

textapp_driver

Name	DataType	Constraints	Description
id	int	Primarykey	ID of users
dname	varchar(100)	Notnull	Name of drivers
daddress	varchar(100)	Notnul	Address of drivers
dcontact	varchar(100)	Notnull	Contact number of the drivers
demail	varchar(100)	Notnull	Email id of drivers
dpassword	varchar(100)	Notnull	Password of drivers
status	varchar(100)	Notnull	Current status of the booking

textapp_booking

Name	DataType	Constraints	Description
id	int	Primarykey	ID of users
date	varchar(100)	Notnull	Date of booking
status	varchar(100)	Notnull	Current status of the booking
did_id	bigint	Foreignkey	ID of driver
uid_id	bigint	Foreignkey	ID of user

Chapter 5

5 Development of the System

This website can be decomposed into a number of submodules.The submodules of car traffic sign recognizer are administrator, user and driver. Each submodule have specific objectives, to avoid unwanted coupling between modules and to increase cohesion, modules are again decomposed into submodules.

Chapter 6

6 System Testing

Testing is the penultimate step of software development.An elaborate testing of the data is prepared and the system is using the test data, while doing testing, errors are noted and correction is made. The users are trained to operate the developed system. Both hardware and software securities are made to run the developed system successfully. System testing is aimed at ensuring the system works accurately before the live operation commences.

6.1 Test Plan

6.1.1 Scope

This test plan will cover the following testing activities as identified in the testing strategy.

- White Box Testing

White box testing is a testing technique, that examines the program structure and derives test data from the program logic/code.In white box testing, the UI is bypassed.Inputs and outputs of ratings are tested directly at the code level and the results are compared against specifications.It reveals errors in "hidden" code.

- Black Box Testing

Black-box testing is a method of software testing that examines the functionality of an application based on the specifications.Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end-user would.

- Unit Testing

The module includes four main components to undergo unit testing.It checks the syntax error, logic error and validity of the program correctness.This test will be performed by the developers.

- Integration Testing

After all components pass the unit testing, they get to be tested whether they work correctly when they are running concurrently and communicating to each other.Integration tests exercise an entire subsystem and ensure that a set of components play nicely together.

- Internal Data Testing

We will test the validity of the data before it enters the database to avoid any problems that may face us in the database. We will test the encryption of the personal information of all the users along with the user names and passwords to ensure maximum security of the users privacy.

6.1.2 Software risk issues

In this section, the plan is to test the risk involved in critical issues such as: Difficult to run on visual studio due to large loading time. Some inherent software risks such as complexity exist; also these issues need to be identified. Proper network connection Working of Mysql database

6.1.3 Features to be tested

- Test whether correct user name and password allows you to login.
- Test whether invalid user name and password prevents you from login.
- Test whether there is any connection problem in the Server.
- Test whether the user and admin details are entered correctly.
- Test whether invalid data entry allows saving data successfully.
- Test whether all pages are loaded correctly.
- Test whether watermark is embedded and extracted properly.
- Test whether the provided security works correctly.

6.2 Test consolidation

6.2.1 Test item

The items or features to be tested in the test cases are included in the document. Each and every user input is tested. The present condition of the system is tested. It is checked to make sure that environment is ready for the application to work.

6.2.2 Input specifications

These are the inputs required to execute the test case. All required inputs including data elements and values are given below.

Data	Values
Email	A valid email ID
Password	Combination of characters and numbers
Phone number	10 Digit number

Chapter 7

7 System Implementation and Maintenance

7.1 Implementation

System implementation is the conversion of new system into an operating one which involves creating compatible files, training clients and installing hardware. User training is crucial for minimizing resistance to change and giving chance to prove its worth. Training aids user friendly manuals and healthy screens provide the user with a good start. Software maintenance follows conversion to the extent that changes are necessary to maintain satisfactory operations relative to changes in the user's environment. Maintenance often includes minor enhancements or corrections to the problem that surface late in the systems operations.In the implementation phase, the team builds the components either from scratch or by composition. Given the architecture document meant from the design phase and the requirement document from the analysis phase, the team should build exactly what has been requested, though there is still room for innovation and flexibility. For example, a component may be narrowly designed for this particular system, or the component may be made more general to satisfy a reusability

- Careful planning
- Investigation of system and constraints
- Design the methods to achieve changeover.
- Training the staff in the changed phase.
- Evaluation of change over method.
- The method of implementation and time scale to be adopted are found out initially.

7.2 Maintenance

This phase occurs as a result of deploying the whole system at the end users organization. They will perform the beta testing at the end users and inform to the developers about any needed modification to the application. The customer records all the problems that are encountered during the beta testing and reports these to the developer at regular intervals.

7.2.1 Corrective Maintenance

Even with the best quality assurance activities, it is likely that the customer will uncover defects in the software.Corrective maintenance changes the software to correct the defects. Corrective Maintenance activity may consist of repair, restoration or replacement of equipment. This activity will be result of a regular inspection, which identifies the failure in time for corrective maintenance to be planned and scheduled, then performed during a routine maintenance shutdown.

7.2.2 Adaptive Maintenance

Over time, the original environment(CPU, operating system, business rules, external product characteristics) for which the software was developed is likely to change. Adaptive maintenance results in modification to the software to accommodate changes to its external environment.

7.2.3 Enhanced Maintenance

As software is used, the customer/user will recognize additional functions that will provide the benefit. Perfect maintenance extends the software beyond its original functional requirements.

7.2.4 Preventive Maintenance

Computer software deteriorates due to change, and because of this preventive maintenance often called software re-engineering, must be conducted to enable the software to serve the needs of its end users. Preventive maintenance makes changes to computer programs so that they can be more easily corrected, adapted and enhanced.

Chapter 8

8 Conclusion and Future Scope

8.1 Conclusion

There are three types of users that interact with the system Admin, User and Driver. Each of these three types has different uses of the system so each of them has their own panel. Admin can manage all the features of website dynamically by login on admin panel. The users can detect the traffic signs and can also book a driver and view the bookings. The driver can view the bookings and user details and can approve them.

8.2 Future Scope

- Increased accuracy using better camera device
- Adaptability to dynamic environments

Appendix

A Data Flow Diagram

Data flow is the one of the best way of documenting the entire functionality of the system. For the system ,which will have data flows in and have some processing inside and then some data flow out from the system can be documented or represented effectively by means of data flow out from the system can be documented or represented effectively by means of data flow diagrams. The data flow diagram are a diagrammatic representation of the system,which has input,process and output. Once any system is represented using a data flow diagram we can identify the following things easily:

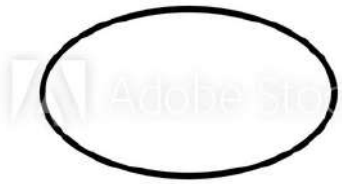
- Various entities interacting with the system are identified
- Flow of data from one entity to another is identified
- The various processes involved in between the interaction of two or more entities in the system are clearly pointed out
- The various data stores which hold the data in between the process,are clearly identified

Some Data Flow Diagram charting forms:

A.1 External source or receiver

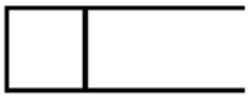
A source or sink is a person or part of organization ,which enters or receives information from the system,but is considered to be outlining the contest of data flow model.

A.2 Transform process



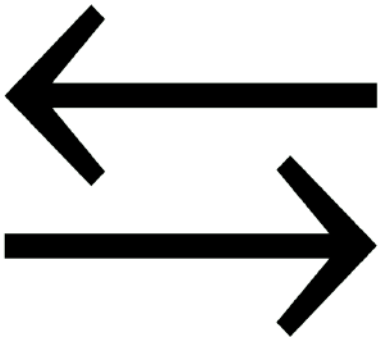
A process represents transformation where incoming data flowsare changed into outgoing data flow

A.3 Data Store



A data store is repository of data that is to be stored for use by one or more process may be as simple as buffer or queue or sophisticated as relational database. They should have clear names.If a process merely uses the contest of store and does not alter it,the arrowhead goes only form the store to the process. If a process alters the details in the store then double-headed arrow is used.

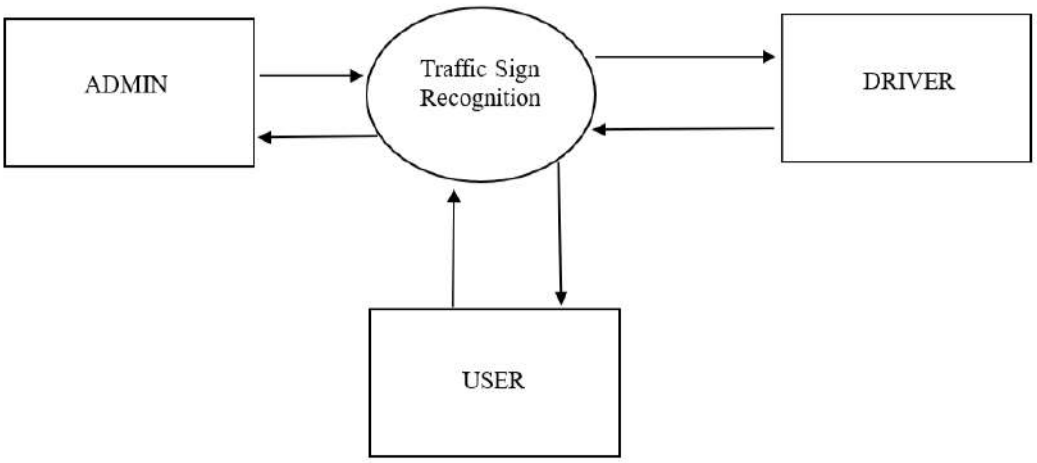
A.4 Data flow



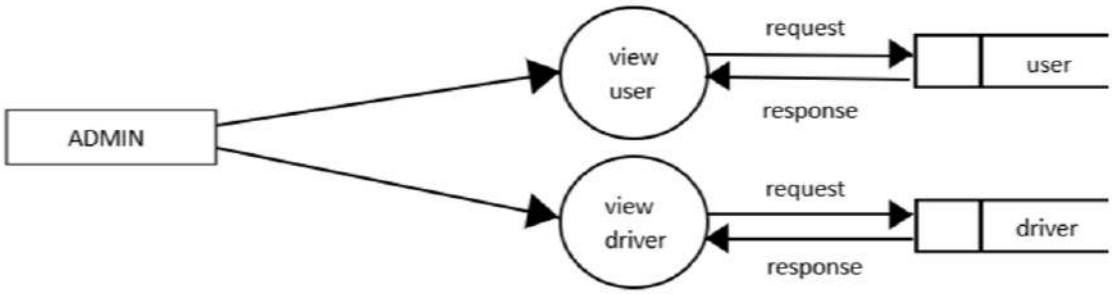
A data flow is a route, which enable packets of data to travel from one point to another.Data may flow,with arrowhead pointing in the direction of the flow

B Data Flow Diagrams

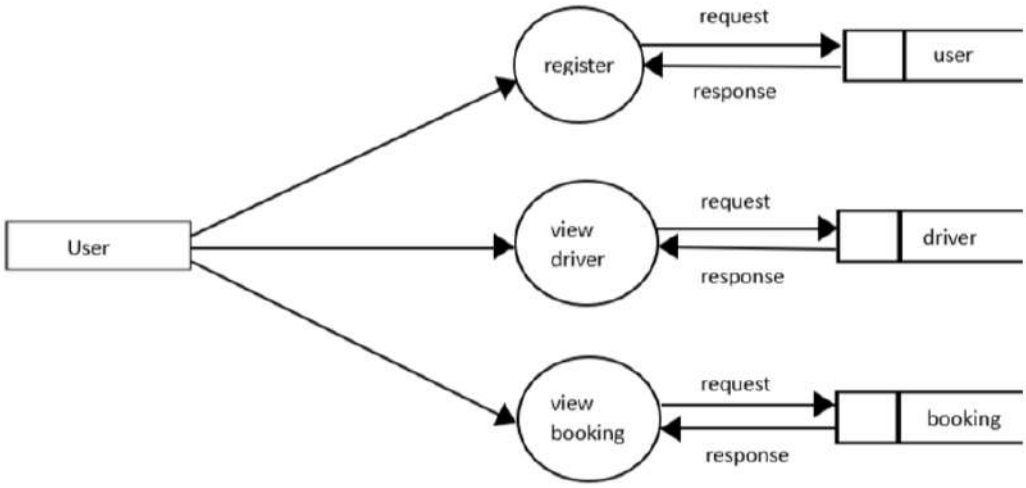
B.1 Level 0



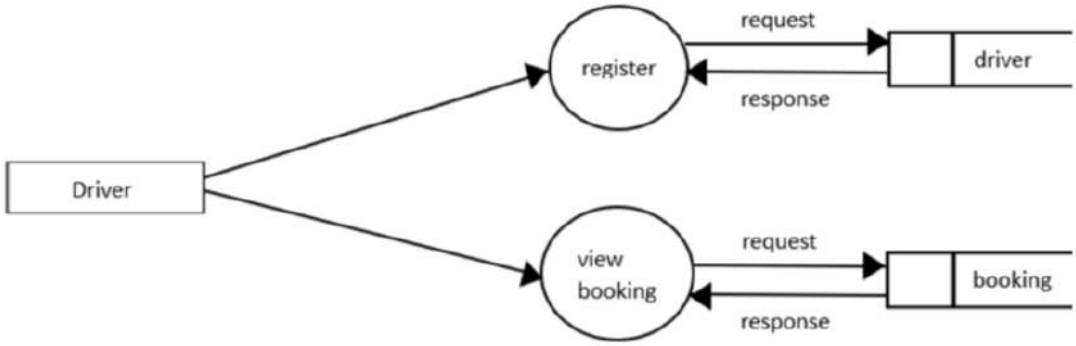
B.2 Level 1.1 - Admin



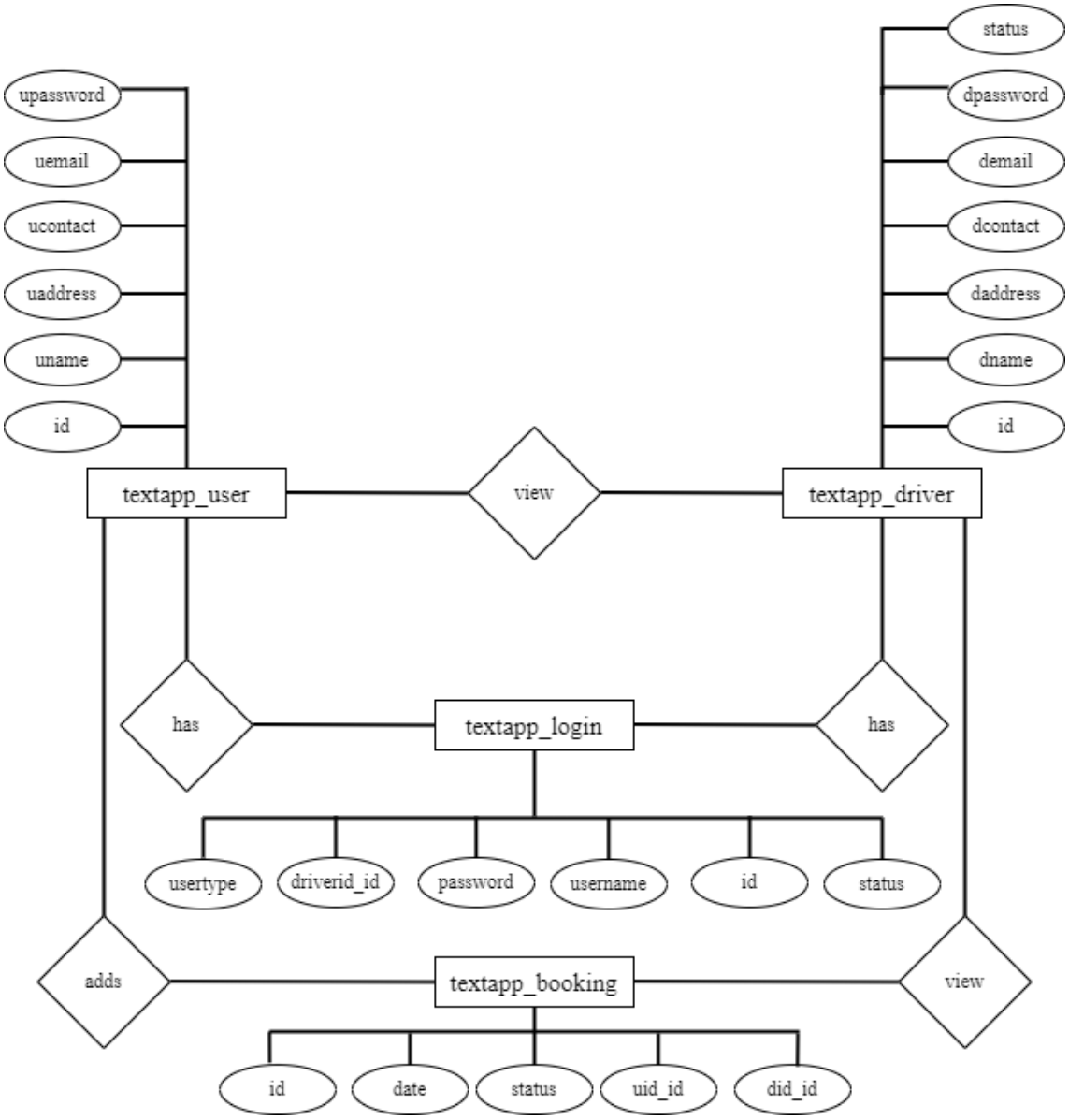
B.3 Level 1.2 - User



B.4 Level 1.3 - Driver

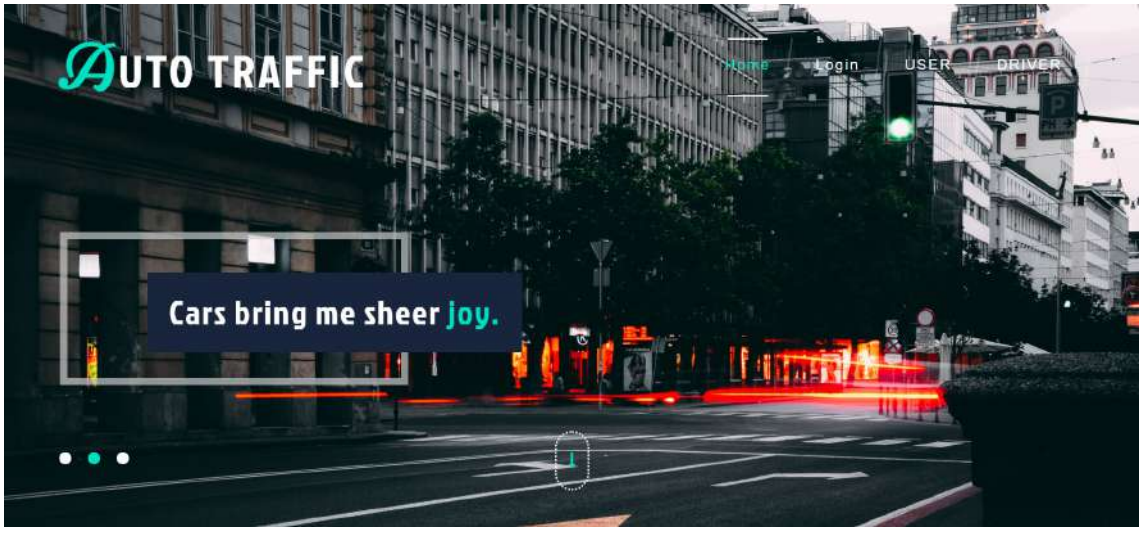


B.5 ER Diagram

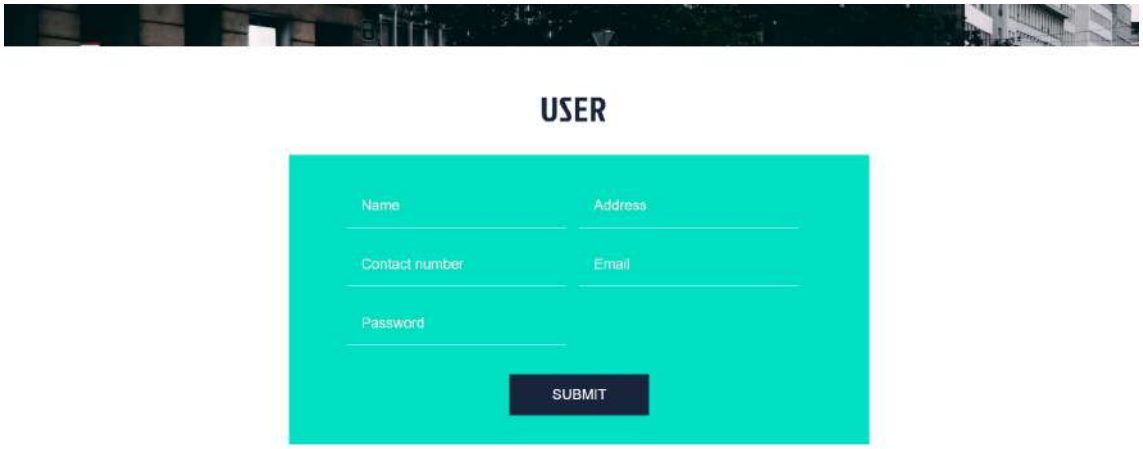


C USER INTERFACES

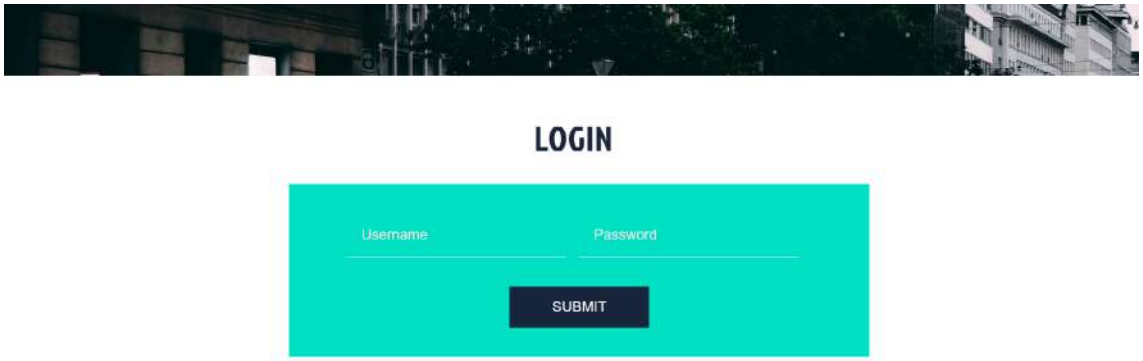
C.1 HOME



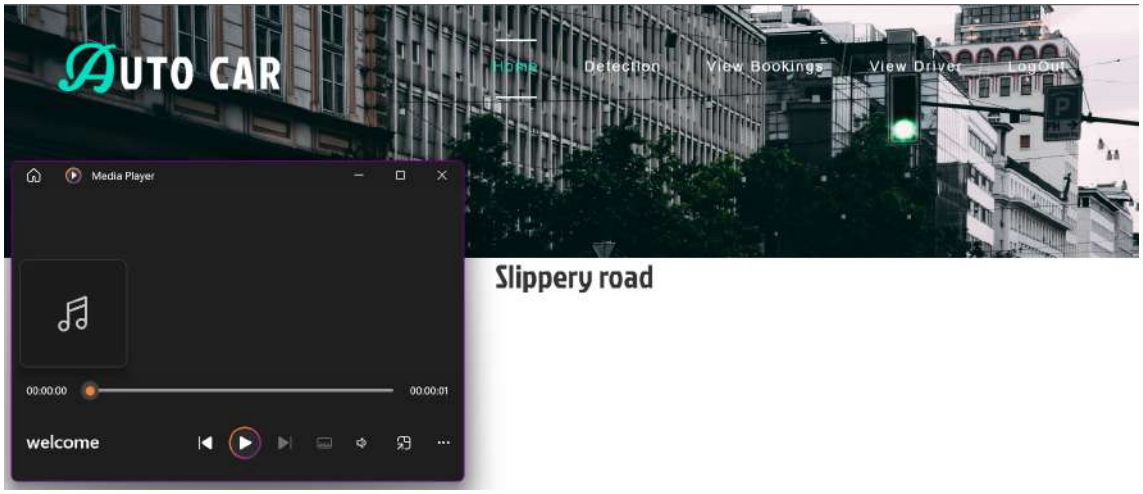
C.2 REGISTRATION



C.3 LOGIN



C.4 DETECTION



C.5 VIEW DRIVERS

NAME	ADDRESS	CONTACT	EMAIL	ACTION
Williams	Kodungalloor	8988713401	williams@gmail.com	Book Now
Simson	Thrissur	9099170787	simson@gmail.com	Book Now
Myna	Chalakkudi	8799136323	myna@gmail.com	Book Now

C.6 VIEW BOOKING

NAME	ADDRESS	CONTACT	EMAIL	STATUS	ACTION
Richard	Chalakkudi	8966139256	richard321@gmail.com		Approve
John	Aluva	8590939355	john@gmail.com		Approve
Philip	Kodungalloor	9890583346	philip@gmail.com		Approve

D Code

urls.py

```
from django.contrib import admin
from django.urls import path
from textapp import views
urlpatterns = [
    path('admin/', admin.site.urls),

    path('index/', views.index),
    path('adminhome/', views.adminhome),
    path('driverhome/', views.driverhome),
    path('userhome/', views.userhome),
    path('', views.index),
    path('login/', views.logins),
    path('userreg/', views.userreg),
    path('driverreg/', views.driverreg),
    path('udp/', views.udp),
    path('adminviewdriver/', views.adminviewdriver),
    path('adminviewuser/', views.adminviewuser),
    path('userviewdriver/', views.userviewdriver),
    path('viewbooking/', views.userviewbooking),
    path('userviewbooking/', views.userviewbooking),
    path('driverviewbooking/', views.driverviewbooking),
    path('ddetection/', views.ddetection),
    path('driverdetection/', views.driverdetection),
    path('imagebyenter/', views.imagebyenter),
    path('prediction/', views.predict),
    path('deletedriver/', views.deletedriver),
    path('deleteuser/', views.deleteuser),
    \# path('udp/', views.udp),

]
```

model.py

```
import os
from tensorflow.keras.preprocessing import image
import matplotlib.pyplot as plt
```

```
import numpy as np
import tensorflow.python.keras.utils as generic\_utils
import random,shutil
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dropout,Conv2D,Flatten,Dense,
MaxPooling2D, BatchNormalization
from tensorflow.keras.models import load\_model


def generator(dir, gen=image.ImageDataGenerator(rescale=1./255), shuffle=True,batch
\_size=1,target\_size=(24,24),class\_mode='categorical' ):

return gen.flow\_from\_directory(dir,batch\_size=batch\_size,shuffle=shuffle,
color\_mode='grayscale',class\_mode=class\_mode,target\_size=target\_size)

BS= 32
TS=(24,24)
train\_batch= generator('data/train',shuffle=True, batch\_size=BS,target\_size=TS)
valid\_batch= generator('data/valid',shuffle=True, batch\_size=BS,target\_size=TS)
SPE= len(train\_batch.classes)//BS
VS = len(valid\_batch.classes)//BS
print(SPE,VS)


\# img,labels= next(train\_batch)
\# print(img.shape)

model = Sequential([
    Conv2D(32, kernel\_size=(3, 3), activation='relu', input\_shape=(24,24,1)),
    MaxPooling2D(pool\_size=(1,1)),
    Conv2D(32,(3,3),activation='relu'),
    MaxPooling2D(pool\_size=(1,1)),
\#32 convolution filters used each of size 3x3
\#again
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D(pool\_size=(1,1)),

\#64 convolution filters used each of size 3x3
\#choose the best features via pooling
```

```
\#randomly turn neurons on and off to improve convergence
    Dropout(0.25),
\#flatten since too many dimensions, we only want a classification output
    Flatten(),
\#fully connected to get all relevant data
    Dense(128, activation='relu'),
\#one more dropout for convergence' sake :)
    Dropout(0.5),
\#output a softmax to squash the matrix into output probabilities
    Dense(2, activation='softmax')
])

model.compile(optimizer='adam',loss='categorical\_crossentropy',metrics=['accuracy'])

model.fit\_generator(train\_batch, validation\_data=valid\_batch,epochs=15,steps\_per

model.save('models/cnnCat2.h5', overwrite=True)
```

models.py

```
[verbetim]

from django.db import models




class Driver(models.Model):
    dname=models.CharField(max\_length=100,blank=True)
    daddress=models.CharField(max\_length=100,blank=True)
    dcontact=models.CharField(max\_length=100,blank=True)
    demail=models.CharField(max\_length=100,blank=True)
    dpassword=models.CharField(max\_length=100,blank=True)
    status=models.CharField(max\_length=100,blank=True)


class Login(models.Model):
    username=models.CharField(max\_length=100,blank=True)
    password=models.CharField(max\_length=100,blank=True)
```

```
    usertype=models.CharField(max\_length=100,blank=True)
    status=models.CharField(max\_length=100,blank=True)
    driverid=models.ForeignKey(Driver,null=True,on\_delete=models.CASCADE)



class User(models.Model):
    uname=models.CharField(max\_length=100,blank=True)
    uaddress=models.CharField(max\_length=100,blank=True)
    ucontact=models.CharField(max\_length=100,blank=True)
    uemail=models.CharField(max\_length=100,blank=True)
    upassword=models.CharField(max\_length=100,blank=True)


class Booking(models.Model):
    uid=models.ForeignKey(User,null=True,on\_delete=models.CASCADE)
    did=models.ForeignKey(Driver,null=True,on\_delete=models.CASCADE)
    date=models.CharField(max\_length=100,blank=True)
    status=models.CharField(max\_length=100,blank=True)
```

views.py

```
\# import email
from http.client import HTTPResponse
from django.shortcuts import render,HttpResponse,HttpResponseRedirect

from .models import *
from django.contrib import messages
from django.db.models import Max, Min,Count,Sum,Avg

import numpy as np
from PIL import Image
import cv2
import tensorflow as tf
import os
from flask\_cors import CORS, cross\_origin
from utils.utils import decodeImage
from predict import traffic
```

```
from django.db.models import Q
import cv2
import tensorflow as tf

def udp(request):
    import os
    from gtts import gTTS
    \# email="admin@gmail.com"
    \# password="admin"

    \# s=Login.objects.create(username=email,password=password,usertype='admin',statu
    \# s.save()
    \# messages.info(request,"already added")
    model\_path = "Traffic.h5"
    loaded\_model = tf.keras.models.load\_model(model\_path)
    videoCaptureObject = cv2.VideoCapture(0)
    print("**************************************************************")
    result = True
    while(result):
        ret,frame = videoCaptureObject.read()
        cv2.imwrite("NewPicture.jpg",frame)
        result = False
    videoCaptureObject.release()
    cv2.destroyAllWindows()
    imagename = "NewPicture.jpg"
    image = cv2.imread(imagename)
    print(image)
    print("**************************************************************")

    image\_fromarray = Image.fromarray(image, 'RGB')
    resize\_image = image\_fromarray.resize((30, 30))
    print("**************************************************************")

    expand\_input = np.expand\_dims(resize\_image,axis=0)
    input\_data = np.array(expand\_input)
    input\_data = input\_data/255
    pred = loaded\_model.predict(input\_data)
    result = pred.argmax()
    print("**************************************************************")
```

```
if result == 0:
    prediction = 'Speed limit (20km/h)'
    print("image",prediction)
elif result == 1:
    prediction = 'Speed limit (30km/h)'
    print("image",prediction)
elif result == 2:
    prediction = 'Speed limit (50km/h)'
    print("image",prediction)
elif result == 3:
    prediction = 'Speed limit (60km/h)'
    print("image",prediction)
elif result == 4:
    prediction = 'Speed limit (70km/h)'
    print("image",prediction)
elif result == 5:
    prediction = 'Speed limit (80km/h)'
    print("image",prediction)
elif result == 6:
    prediction = 'End of speed limit (80km/h)'
    print("image",prediction)
elif result == 7:
    prediction = 'Speed limit (100km/h)'
    print("image",prediction)
elif result == 8:
    prediction = 'Speed limit (120km/h)'
    print("image",prediction)
elif result == 9:
    prediction = 'No passing'
    print("image",prediction)
elif result == 10:
    prediction = 'No passing veh over 3.5 tons'
    print("image",prediction)
elif result == 11:
    prediction = 'Right-of-way at intersection'
    print("image",prediction)
elif result == 12:
    prediction = 'Priority road'
    print("image",prediction)
```

```
elif result == 13:
    prediction = 'Yield'
    print("image",prediction)
elif result == 14:
    prediction = 'Stop'
    print("image",prediction)
elif result == 15:
    prediction = 'No vehicles'
    print("image",prediction)
elif result == 16:
    prediction = 'Veh > 3.5 tons prohibited'
    print("image",prediction)
elif result == 17:
    prediction = 'No entry'
    print("image",prediction)
elif result == 18:
    prediction = 'General caution'
    print("image",prediction)
elif result == 19:
    prediction = 'Dangerous curve left'
    print("image",prediction)
elif result == 20:
    prediction = 'Dangerous curve right'
    print("image",prediction)
elif result == 21:
    prediction = 'Double curve'
    print("image",prediction)
elif result == 22:
    prediction = 'Bumpy road'
    print("image",prediction)
elif result == 23:
    prediction = 'Slippery road'
    print("image",prediction)
elif result == 24:
    prediction = 'Road narrows on the right'
    print("image",prediction)
elif result == 25:
    prediction = 'Road work'
    print("image",prediction)
elif result == 26:
```

```
        prediction = 'Traffic signals'
        print("image",prediction)
    elif result == 27:
        prediction = 'Pedestrians'
        print("image",prediction)
    elif result == 28:
        prediction = 'Children crossing'
        print("image",prediction)
    elif result == 29:
        prediction = 'Bicycles crossing'
        print("image",prediction)
    elif result == 30:
        prediction = 'Beware of ice/snow'
        print("image",prediction)
    elif result == 31:
        prediction = 'Wild animals crossing'
        print("image",prediction)
    elif result == 32:
        prediction = 'End speed + passing limits'
        print("image",prediction)
    elif result == 33:
        prediction = 'Turn right ahead'
        print("image",prediction)
    elif result == 34:
        prediction = 'Turn left ahead'
        print("image",prediction)
    elif result == 35:
        prediction = 'Ahead only'
        print("image",prediction)
    elif result == 36:
        prediction = 'Go straight or right'
        print("image",prediction)
    elif result == 37:
        prediction = 'Go straight or left'
        print("image",prediction)
    elif result == 38:
        prediction = 'Keep right'
        print("image",prediction)
    elif result == 39:
        prediction = 'Keep left'
```

```
        print("image",prediction)
    elif result == 40:
        prediction = 'Roundabout mandatory'
        print("image",prediction)
    elif result == 41:
        prediction = 'End of no passing'
        print("image",prediction)
    elif result == 42:
        prediction = 'End no passing veh > 3.5 tons'
        print("image",prediction)
    language = 'en'
    myobj = gTTS(text=prediction, lang=language, slow=False)
    print("**************************************************")
    print(prediction)
    myobj.save("welcome.mp3")
    os.system("welcome.mp3")
    return render(request,"user/prediction.html",{"prediction":prediction})



    return HttpResponseRedirect("/prediction")


from django.core.files.storage import FileSystemStorage

def imagebyenter(request):
    import os
    from gtts import gTTS
    if request.POST:
        image=request.FILES['image']
        saved\_filename = f"./uploads/{image.name}"
        with open(saved\_filename, "wb") as opened:
            for chunk in image.chunks():
                    opened.write(chunk)

        fs=FileSystemStorage()
        filename=fs.save(image.name,image)
        fileurl=fs.url(filename)

        model\_path = "Traffic.h5"
```

```
loaded\_model = tf.keras.models.load\_model(model\_path)

imagename = image
print(saved\_filename)
image = cv2.imread(saved\_filename)

image\_fromarray = Image.fromarray(image, 'RGB')
resize\_image = image\_fromarray.resize((30, 30))
expand\_input = np.expand\_dims(resize\_image,axis=0)
input\_data = np.array(expand\_input)
input\_data = input\_data/255
pred = loaded\_model.predict(input\_data)
result = pred.argmax()


if result == 0:
    prediction = 'Speed limit (20km/h)'
    print("image",prediction)
elif result == 1:
    prediction = 'Speed limit (30km/h)'
    print("image",prediction)
elif result == 2:
    prediction = 'Speed limit (50km/h)'
    print("image",prediction)
elif result == 3:
    prediction = 'Speed limit (60km/h)'
    print("image",prediction)
elif result == 4:
    prediction = 'Speed limit (70km/h)'
    print("image",prediction)
elif result == 5:
    prediction = 'Speed limit (80km/h)'
    print("image",prediction)
elif result == 6:
    prediction = 'End of speed limit (80km/h)'
    print("image",prediction)
elif result == 7:
    prediction = 'Speed limit (100km/h)'
    print("image",prediction)
elif result == 8:
```

```
        prediction = 'Speed limit (120km/h)'
        print("image",prediction)
    elif result == 9:
        prediction = 'No passing'
        print("image",prediction)
    elif result == 10:
        prediction = 'No passing veh over 3.5 tons'
        print("image",prediction)
    elif result == 11:
        prediction = 'Right-of-way at intersection'
        print("image",prediction)
    elif result == 12:
        prediction = 'Priority road'
        print("image",prediction)
    elif result == 13:
        prediction = 'Yield'
        print("image",prediction)
    elif result == 14:
        prediction = 'Stop'
        print("image",prediction)
    elif result == 15:
        prediction = 'No vehicles'
        print("image",prediction)
    elif result == 16:
        prediction = 'Veh > 3.5 tons prohibited'
        print("image",prediction)
    elif result == 17:
        prediction = 'No entry'
        print("image",prediction)
    elif result == 18:
        prediction = 'General caution'
        print("image",prediction)
    elif result == 19:
        prediction = 'Dangerous curve left'
        print("image",prediction)
    elif result == 20:
        prediction = 'Dangerous curve right'
        print("image",prediction)
    elif result == 21:
        prediction = 'Double curve'
```

```
        print("image",prediction)
    elif result == 22:
        prediction = 'Bumpy road'
        print("image",prediction)
    elif result == 23:
        prediction = 'Slippery road'
        print("image",prediction)
    elif result == 24:
        prediction = 'Road narrows on the right'
        print("image",prediction)
    elif result == 25:
        prediction = 'Road work'
        print("image",prediction)
    elif result == 26:
        prediction = 'Traffic signals'
        print("image",prediction)
    elif result == 27:
        prediction = 'Pedestrians'
        print("image",prediction)
    elif result == 28:
        prediction = 'Children crossing'
        print("image",prediction)
    elif result == 29:
        prediction = 'Bicycles crossing'
        print("image",prediction)
    elif result == 30:
        prediction = 'Beware of ice/snow'
        print("image",prediction)
    elif result == 31:
        prediction = 'Wild animals crossing'
        print("image",prediction)
    elif result == 32:
        prediction = 'End speed + passing limits'
        print("image",prediction)
    elif result == 33:
        prediction = 'Turn right ahead'
        print("image",prediction)
    elif result == 34:
        prediction = 'Turn left ahead'
        print("image",prediction)
```

```
        elif result == 35:
            prediction = 'Ahead only'
            print("image",prediction)
        elif result == 36:
            prediction = 'Go straight or right'
            print("image",prediction)
        elif result == 37:
            prediction = 'Go straight or left'
            print("image",prediction)
        elif result == 38:
            prediction = 'Keep right'
            print("image",prediction)
        elif result == 39:
            prediction = 'Keep left'
            print("image",prediction)
        elif result == 40:
            prediction = 'Roundabout mandatory'
            print("image",prediction)
        elif result == 41:
            prediction = 'End of no passing'
            print("image",prediction)
        elif result == 42:
            prediction = 'End no passing veh > 3.5 tons'
            print("image",prediction)
        language = 'en'
        myobj = gTTS(text=prediction, lang=language, slow=False)
        myobj.save("welcome.mp3")
        os.system("welcome.mp3")
        return render(request,"user/prediction.html",{"prediction":prediction})


    return render(request,"user/enterimage.html")

def predict(request):

    return render(request,"user/prediction.html")
```

```
def index(request):
    return render(request,"common/index.html")

def adminhome(request):
    return render(request,"admin/index.html")

def userhome(request):
    return render(request,"user/index.html")

def driverhome(request):
    return render(request,"driver/index.html")

def driverreg(request):
    if request.POST:
        name=request.POST.get("name")
        address=request.POST.get("address")
        contact=request.POST.get("contact")
        email=request.POST.get("email")
        password=request.POST.get("password")

        s=Driver.objects.create(dname=name,daddress=address,dcontact=contact,demail=e
        s.save()
        did=Driver.objects.aggregate(Max('id'))
        print(did)
        did=did['id\_\_max']
        did=Driver.objects.get(id=did)
        print(did)

    \# print(mark)
        s=Login.objects.create(username=email,password=password,usertype='driver',
        status='requested',driverid=did)
        s.save()
        messages.info(request,"already added")
    return render(request,"common/driverreg.html")



def userreg(request):
```

```
    if request.POST:
        name=request.POST.get("name")
        address=request.POST.get("address")
        contact=request.POST.get("contact")
        email=request.POST.get("email")
        password=request.POST.get("password")

        s=User.objects.create(uname=name,uaddress=address,ucontact=contact,
            uemail=email,upassword=password)
        s.save()
        s=Login.objects.create(username=email,password=password,
             usertype='user',status='requested')
        s.save()
        messages.info(request,"already added")
    return render(request,"common/userreg.html")


def logins(request):
    msg=""
    if request.POST:
        name=request.POST.get("username")
        password=request.POST.get("password")
        print(name)
        print(password)

        s=Login.objects.filter(Q(username=name),Q(password=password))

        try:
            if s[0].usertype=='admin':
                msg="Success"
                return HttpResponseRedirect("/adminhome")
            elif s[0].usertype=='user':
                s=User.objects.get(uemail=name)
                id=s.id
                request.session['uid']=id
                msg="Success"
                return HttpResponseRedirect("/userhome")
            elif s[0].usertype=='driver':
                s=Driver.objects.get(demail=name)
                id=s.id
```

```
                request.session['did']=id
                msg="Success"
                return HttpResponseRedirect("/driverhome")

        except:
            msg="invalid Username Or password"


    \# return HttpResponseRedirect('/login')

    return render(request,"common/login.html",{'msg':msg})




def adminviewdriver(request):
    s=Driver.objects.filter()
    print(s)
    if request.GET:
        id=request.GET.get("id")
        s=Driver.objects.get(id=id)
        username=s.demail
        s=Login.objects.filter(username=username).update(status='approved')
        s=Driver.objects.filter(id=id).update(status='approved')
        return HttpResponseRedirect("/adminviewdriver")
    return render(request,"admin/viewdriver.html",{"data":s})

def deletedriver(request):
    id=request.GET.get('id')
    Driver.objects.filter(id=id).delete()
    return HttpResponseRedirect("/adminviewdriver")


def adminviewuser(request):
    s=User.objects.all()
    print(s)
    return render(request,"admin/viewuser.html",{"data":s})

def deleteuser(request):
    id=request.GET.get('id')
```

```
    User.objects.filter(id=id).delete()
    return HttpResponseRedirect("/adminviewuser")


def userviewdriver(request):
    s=Driver.objects.all()
    uid=request.session['uid']
    if request.GET:
        id=request.GET.get("id")
        driverdetails=Driver.objects.get(id=id)
        userdetails=User.objects.get(id=uid)
        import datetime
        date=datetime.datetime.now()
        s=Booking.objects.create(uid=userdetails,did=driverdetails,date=date,status='
        return HttpResponseRedirect("/userviewdriver")
    return render(request,"user/viewdriver.html",{"data":s})




def userviewbooking(request):
    uid=request.session['uid']
    select=Booking.objects.filter(uid=uid)
    return render(request,"user/viewbooking.html",{"data":select})




def driverviewbooking(request):
    uid=request.session['did']
    print(uid)
    select=Booking.objects.filter(did=uid)
    print(select)
    abcd='approved'
    if request.GET:
        id=request.GET.get("id")
        select=Booking.objects.filter(id=id).update(status='approved')
        return HttpResponseRedirect("/driverhome")

    return render(request,"driver/viewbooking.html",{"data":select,"abcd":abcd})
```

```
def ddetection(request):
    from textapp import cd
    return HttpResponseRedirect("/userhome")



def driverdetection(request):
    from textapp import cd
    return HttpResponseRedirect("/driverhome")
```

predict.py

```
import numpy as np
from PIL import Image
import cv2
import tensorflow as tf

class traffic:
    def \_\_init\_\_(self,filename):
        self.filename =filename


    def trafficsign(self):

        model\_path = "Traffic.h5"
        loaded\_model = tf.keras.models.load\_model(model\_path)

        imagename = self.filename
        image = cv2.imread(imagename)

        image\_fromarray = Image.fromarray(image, 'RGB')
        resize\_image = image\_fromarray.resize((30, 30))
        expand\_input = np.expand\_dims(resize\_image,axis=0)
        input\_data = np.array(expand\_input)
        input\_data = input\_data/255
```

```
pred = loaded\_model.predict(input\_data)
result = pred.argmax()

if result == 0:
    prediction = 'Speed limit (20km/h)'
    return [{"image": prediction}]
elif result == 1:
    prediction = 'Speed limit (30km/h)'
    return [{"image": prediction}]
elif result == 2:
    prediction = 'Speed limit (50km/h)'
    return [{"image": prediction}]
elif result == 3:
    prediction = 'Speed limit (60km/h)'
    return [{"image": prediction}]
elif result == 4:
    prediction = 'Speed limit (70km/h)'
    return [{"image": prediction}]
elif result == 5:
    prediction = 'Speed limit (80km/h)'
    return [{"image": prediction}]
elif result == 6:
    prediction = 'End of speed limit (80km/h)'
    return [{"image": prediction}]
elif result == 7:
    prediction = 'Speed limit (1000km/h)'
    return [{"image": prediction}]
elif result == 8:
    prediction = 'Speed limit (120km/h)'
    return [{"image": prediction}]
elif result == 9:
    prediction = 'No passing'
    return [{"image": prediction}]
elif result == 10:
    prediction = 'No passing veh over 3.5 tons'
    return [{"image": prediction}]
elif result == 11:
    prediction = 'Right-of-way at intersection'
    return [{"image": prediction}]
elif result == 12:
```

```
        prediction = 'Priority road'
        return [{"image": prediction}]
    elif result == 13:
        prediction = 'Yield'
        return [{"image": prediction}]
    elif result == 14:
        prediction = 'Stop'
        return [{"image": prediction}]
    elif result == 15:
        prediction = 'No vehicles'
        return [{"image": prediction}]
    elif result == 16:
        prediction = 'Veh > 3.5 tons prohibited'
        return [{"image": prediction}]
    elif result == 17:
        prediction = 'No entry'
        return [{"image": prediction}]
    elif result == 18:
        prediction = 'General caution'
        return [{"image": prediction}]
    elif result == 19:
        prediction = 'Dangerous curve left'
        return [{"image": prediction}]
    elif result == 20:
        prediction = 'Dangerous curve right'
        return [{"image": prediction}]
    elif result == 21:
        prediction = 'Double curve'
        return [{"image": prediction}]
    elif result == 22:
        prediction = 'Bumpy road'
        return [{"image": prediction}]
    elif result == 23:
        prediction = 'Slippery road'
        return [{"image": prediction}]
    elif result == 24:
        prediction = 'Road narrows on the right'
        return [{"image": prediction}]
    elif result == 25:
        prediction = 'Road work'
```

```
        return [{"image": prediction}]
    elif result == 26:
        prediction = 'Traffic signals'
        return [{"image": prediction}]
    elif result == 27:
        prediction = 'Pedestrians'
        return [{"image": prediction}]
    elif result == 28:
        prediction = 'Children crossing'
        return [{"image": prediction}]
    elif result == 29:
        prediction = 'Bicycles crossing'
        return [{"image": prediction}]
    elif result == 30:
        prediction = 'Beware of ice/snow'
        return [{"image": prediction}]
    elif result == 31:
        prediction = 'Wild animals crossing'
        return [{"image": prediction}]
    elif result == 32:
        prediction = 'End speed + passing limits'
        return [{"image": prediction}]
    elif result == 33:
        prediction = 'Turn right ahead'
        return [{"image": prediction}]
    elif result == 34:
        prediction = 'Turn left ahead'
        return [{"image": prediction}]
    elif result == 35:
        prediction = 'Ahead only'
        return [{"image": prediction}]
    elif result == 36:
        prediction = 'Go straight or right'
        return [{"image": prediction}]
    elif result == 37:
        prediction = 'Go straight or left'
        return [{"image": prediction}]
    elif result == 38:
        prediction = 'Keep right'
        return [{"image": prediction}]
```

```
elif result == 39:
    prediction = 'Keep left'
    return [{"image": prediction}]
elif result == 40:
    prediction = 'Roundabout mandatory'
    return [{"image": prediction}]
elif result == 41:
    prediction = 'End of no passing'
    return [{"image": prediction}]
elif result == 42:
    prediction = 'End no passing veh > 3.5 tons'
    return [{"image": prediction}]
else:
    return [{"ERROR": "Please select another image. !!!"}]
```

CAR TRAFFIC SIGN RECOGNIZER

PROJECT REPORT

Submitted By

JOSHUA JOSE

Reg. No. CCAVBCA039

for the award of the Degree of

Bachelor (BCA)

of Computer Application
(University of Calicut)

under the guidance of

Ms.Varsha Ganesh

Assistant Professor

BACHELOR OF COMPUTER APPLICATION
DEPARTMENT OF COMPUTER SCIENCE
CHRIST COLLEGE(AUTONOMOUS)
IRINJALAKUDA, KERALA
INDIA

2021-2024

DEPARTMENT OF COMPUTER SCIENCE

CHRIST COLLEGE (AUTONOMOUS)

IRINJALAKUDA

CERTIFICATE

This is to certify that the project report entitled ***"Car Traffic Sign Recognizer"*** *is a bonfied record of the project work done by* ***Joshua Jose*** *in partial fulfillment of the requirement for the sixth semester of* ***Bachelor of Computer Application*** *in Department of Computer Science of* **CHRIST COLLEGE (AUTONOMOUS) IRINJALAKUDA**.

Ms.Varsha Ganesh
Assistant Professor
Internal Guide

Ms.Sini Thomas
Head of the Department
Computer Science

EXTERNAL EXAMINER

INTERNAL EXAMINER

DECLARATION

I hereby declare that this project work **"CAR TRAFFIC SIGN RECOGNIZER"** submitted to Christ College (Autonomous) Irinjalakuda, affiliated to Calicut University in partial fulfillment of the requirement for the award of the Bachelor of Computer Application, is a record of original work done by me, under the guidance of Ms.VARSHA GANESH, Department of Computer Science.

Place: Irinjalakuda JOSHUA JOSE

ACKNOWLEDGEMENT

First and foremost, I would like to thank Lord almighty for his providence and for being the guiding light throughout the project. I wish to express my sincere gratitude to our beloved Department for giving us all the facilities for our project. I take this opportunity to express my gratitude to the class teacher Ms.SOUMYA P.S and the Head of the Department Ms.SINI THOMAS who has supported us throughout the course of this project. I am thankful for their aspiring guidance and valuable advices during the project work. I express my sincere thanks to our project guide Ms.VARSHA GANESH for supporting and guiding us throughout the project. I would take this opportunity to specially thank all other faculty members for their constant and continuous motivation. Finally, I would like to thank my family and friends for giving valuable advices and moral support throughout our project.

ABSTRACT

CAR TRAFFIC SIGN RECOGNIZER is a website designed to helps the driver about recognition of road signs to avoid road accidents. It represents an important feature of advanced driver assistance system, contributing to the safety of the drivers, autonomous vehicles as well and to increase driving comfort. The website has three kind of login facilities :- admin login, user login and driver login. All these features make this website more adaptable and user-friendly.

Contents

Chapter 1

1 Introduction

In today's world road conditions drastically improved as compared with past decades. Obviously, vehicle's speed increased. So, on driver's point of view there might be chances of neglecting mandatory road signs while driving. This project explores the system to helps the driver about recognition of road signs to avoid road accidents. An automatic means of detecting and recognizing traffic signs can make a significant contribution to this goal by providing a fast method of detecting, classifying and logging signs. This method helps to develop the inventory accurately and consistently.

The project is mainly focused on automatic recognition of warning signs placed in local roads captured by image clips. The system classifies the traffic signs present in the image into different categories. With this model, we can read and understand traffic signs which are very important task for autonomous vehicles.

1.1 Overview

The objective of the Car Traffic Sign Recognizer website is to design a simple and adaptable website which helps users to increase transportation efficiency, road safety and to reduce the environmental impact with the use of advanced communication technologies. The road signs are placed on either roadside or above the roads. These signs provide mandatory information regarding to guiding, warning and regulating the behaviors to drivers in order to make driving safer and easier. A system is developed to assist the drivers, based on detection and recognition of signs which corrects the most unsafe driving behaviors.

Chapter 2

2 System Analysis

2.1 Purpose

The purpose of the traffic sign recognition project is to develop a sophisticated system that can automatically detect, interpret, and respond to traffic signs in real-time. This project aims to empower vehicles with the ability to perceive and understand the various types of traffic signs encountered on roadways. Ultimately, the goal of this project is to create a robust and reliable solution for enhancing the awareness and decision-making capabilities of drivers and autonomous vehicles alike.

2.1.1 Existing System

Existing systems of traffic sign recognition utilize a combination of cameras, sensors, and sophisticated image processing algorithms to detect and interpret traffic signs in real-time. By capturing images of traffic signs and analyzing them these systems can recognize various types of signs, including speed limits, stop signs, and lane restrictions. Limitations of existing system : The system may still encounter challenges in accurately detecting and interpreting traffic signs, leading to false detections or missed signs, potentially impacting driver safety.

2.1.2 Proposed System

The proposed system of traffic sign recognition aims to address the current limitations by leveraging advancements in artificial intelligence, particularly deep learning algorithms. By training deep neural networks on large datasets of annotated traffic sign images, the system would learn to accurately detect and interpret various types of signs, including speed limits, stop signs, and lane restrictions, across diverse environmental conditions.

2.2 Problem definition

To know what the problem is and what the needs are before developing it. Here we design Car Traffic Sign Recognizer to detect the traffic signs accurately.

2.3 Feasibility Study

After the problem is clearly understood and solutions proposed, the next step is to conduct the feasibility study. Feasibility study is defined as evaluation or analysis of the potential impact of the proposed project or program. The objective is to determine whether the proposed system is feasible. There are three aspects of feasibility study to which the proposed system is subjected as discussed below.

2.3.1 Technical Feasibility

Technical feasibility assess whether the current technical resources are sufficient for the new system. We can upgrade the level of technology for supporting our website. We check whether the proposed system can be implemented in the present system without supporting the existing hardware.

2.3.2 Economical Feasibility

Economic feasibility determines whether the time and money are available to develop the system. There is no additional hardware used to develop the site, it is inexpensive to build.

2.3.3 Operational Feasibility

Operational feasibility determines if the human resources are available to operate the system once it has been installed. No extra training is needed to use this system. Anyone who has knowledge in internet in english language can easily use the system. The resources that are required to implement or install are already available with the organization.

Chapter 3

3 Software Requirement Specification

3.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the CAR TRAFFIC SIGN RECOGNIZER. It illustrate the purpose and complete description for the development of the system. It explain system constraints, interface and interactions with other external applications.

3.2 Scope

The scope of car traffic sign recognizer extends to improving road safety, enhancing driver assistance technologies, optimizing traffic flow, and advancing the capabilities of autonomous vehicles and smart transportation systems.

3.3 Overall Description

This section give an overview of our website, CAR TRAFFIC SIGN RECOGNIZER. This project aims to develop an intelligent system capable of automatically detecting, recognizing, and interpreting traffic signs from images captured by cameras mounted on vehicles. Upon detection, the system may provide visual or auditory alerts to drivers, contributing to improved road safety and compliance with traffic laws. It plays a vital role in enhancing driver assistance systems, navigation, and autonomous vehicle technologies, ultimately leading to safer and more efficient transportation systems.

3.3.1 Product Perspective

The car traffic sign recognition project aims to develop a robust and user-friendly system that enhances driver safety and convenience. The project can develop a comprehensive solution that enhances road safety, improves driver assistance technologies, and advances the capabilities of autonomous vehicles and smart transportation systems.

3.3.2 Product Functionality

The system provides a comprehensive solution that enhances road safety, improves driver assistance technologies, and advances the capabilities of autonomous vehicles and smart transportation systems by detecting the traffic signs and notifying the driver.

3.3.3 Users and Characteristics

There are three types of users that interact with the site - admin, user and driver. Each of these have different tasks which is performed. Users are able to detect the traffic signs, view drivers and book drivers. Drivers can view the bookings and approve it. Admin is able view all the users and drivers and has the authority to delete the users or drivers.

3.4 Specific Requirements

3.4.1 Hardware Requirements

- System: LAPTOP-TE3E25EG
- Processor: i3 8th Gen or above
- Speed: 2.80 GHz
- RAM capacity: 8 GB
- Hard Dsk drive: 128 GB
- Keyboard: Standard
- Mouse: Standard
- Monitor: 18.5” LED Monitor

3.4.2 Software Requirements

- Operating System: Windows
- Languages used: Python, Django
- Database : SQLite3
- Technologies used: HTML,Javascript,CSS

3.5 Functional Requirements

It contains three main modules.

- 1.Admin
- 2.User
- 3.Driver

Admin

An Admin account is used for editing or managing the website dynamically by Admin panel. The admin can view the users and drivers and has the authority to delete any user or driver.

User

The user can detect the traffic signs. There is a feature providing for booking a driver and viewing the bookings.

Driver

The driver can view the bookings and user details. The driver can approve the booking.

3.6 Non Functional Requirements

Non-functional requirements define the overall qualities or attributes of the resulting system. Non-functional system place restrictions on the product being developed, the developed process, and specify external constraints that the product must meet. Examples of non-functional requirements include safety, security, usability, reliability and performance requirements. Project management issues (costs, time and schedule) are often considered as non-functional requirements. The principle non-functional constraints which are relevant to critical systems :

- performance
- security
- safety
- usability

Performance

Performance requirements concern the speed of operation of a system. Types of performance requirements :

- Response requirements (how quickly the system reacts to a user input).
- Throughput requirements (how much the system can accomplish within a specified amount of time).
- Availability requirements (is the system available for service when requested by end users). The speed of operation of this system is adequate for the requirements.

Reliability

- Reliability is the ability of a system to perform its required function under stated conditions for a specified period of time.
- Constraints on runtime behavior of the system. This system is reliable because its functionalities can be done on the required conditions.

Safety

Safety requirements are not required which exclude unsafe situation from the possible solution of the system.

Usability

Usability is the ease with which a user can learn to operate, prepare inputs for, and interpret outputs of system or components.
Usability requirements include :

- information error messages.
- well-formed user interfaces.

3.7 Interface Requirements

3.7.1 Hardware interfaces

The system must run over the internet, all the hardware shall require to connect internet will be hardware interface for the system. As for example modem, WAN-LAN.

3.7.2 Software interfaces

Software interface required for the working of the project is the appropriate operating system.

3.7.3 Communication interfaces

The user of the site communicate using the network connectivity and the data set is accessed

3.8 Security Requirements

- User accesses only their account.
- Validation of input is handled.
- This application containing the computer systems is physically secured against arms or surreptitious entry by intruders.
- Users must be authorized carefully to reduce changes of any such user giving access to an intruder in exchange for a bribe or other favour.

3.9 Platform Used

Windows 11 is the latest iteration of Microsoft's operating system, introducing a refreshed user interface, enhanced productivity features, and improved performance. With a sleek design aesthetic centered around simplicity and productivity, Windows 11 offers users a more intuitive and personalized computing experience.Windows 11 represents Microsoft's vision for the future of computing, blending familiarity with innovation to empower users to do more. Windows 11 brings performance improvements, with optimizations for better energy efficiency and faster wake times. With support for gaming features like DirectStorage and Auto HDR, along with enhanced security measures, Windows 11 aims to cater to a wide range of users, from casual consumers to power users and professionals, ushering in a new era of computing excellence.

3.10 Technologies Used

Python

Python is a high-level programming language renowned for its simplicity, versatility, and readability. With its clean and concise syntax, Python is accessible to beginners while remaining powerful enough for complex applications in various domains, including web development, data analysis, machine learning, artificial intelligence, and automation.Its extensive standard library and robust ecosystem of third-party packages make it a favorite among developers for rapid prototyping, building scalable applications, and solving a wide range of computational problems. Python's interpreted nature and cross-platform compatibility further contribute to its popularity, enabling developers to write code once and run it on multiple platforms without modification. Overall, Python's ease of use, flexibility, and community support make it a go-to choice for developers worldwide.

SQLite3

SQLite3 is a lightweight, self-contained, serverless relational database engine that is widely used for embedded systems, mobile applications, and small-scale database-driven websites. It is part of the SQLite project, which aims to provide a simple, fast, and reliable database solution with minimal setup and administration. SQLite3 is unique in that it operates as a single disk file and requires no configuration or administration, making it extremely easy to deploy and use. Despite its small footprint, SQLite3 supports many

standard SQL features, including transactions, triggers, and views, making it suitable for a wide range of applications. Overall, SQLite3 is an excellent choice for projects requiring a lightweight and efficient database solution without the overhead of a full-fledged database management system.

Chapter 4

4 Design Document

4.1 Purpose

The purpose of a car traffic sign recognizer project is to enhance road safety, improve driver assistance systems, and facilitate the development of autonomous vehicles. By developing an intelligent system capable of automatically detecting, recognizing, and interpreting traffic signs from images captured by onboard cameras, this project aims to provide valuable information to drivers in real-time. The system can alert drivers about speed limits, stop signs, yield signs, and other relevant traffic regulations, helping them make informed decisions and comply with road rules more effectively. Additionally, integrating traffic sign recognition technology into vehicles contributes to the advancement of autonomous driving technologies by enabling vehicles to perceive and understand the surrounding environment, ultimately leading to safer and more efficient transportation systems. Purpose of this document is to give the detailed description of the architecture and system design for the software. The system design document shows the software will be structured to satisfy the requirement identified in the software requirement specification. It is a translation of requirements into a description of the software structure, software component, interfaces and data necessary for the implementation phase. In a complete software design document, each requirements must be traceable to one or more design entities.

4.2 Scope

Car Traffic Sign Recognizer is a website which helps in detecting the traffic signs and alert the driver to ensure safe and comfortable driving.

4.3 Overview

The purpose of this document is to help the reader to visualize the solution to the project presented.This document verifies how the design meet the requirement stimulated in the SRS document through design viewpoints. The design viewpoints will cover all design elements presented before. This document will provide a direct approach to the development

of this project hence reducing feature creep and ponitedly determine the quality of the design.

4.4 Data Design

Database are the storehouses of data used in the software systems.The data is stored in tables inside the database. Several tables are created for the manipulation of the data for the system.Two essential settings for a database are

- Primary Key- The field that is unique for all the record occurrence.
- Foreign Key- The field used to set relation between tables.

Normalization

- Normalization is the process of organizing the data in the database.
- Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate the undesirable characteristics like Insertion, Update and Deletion Anomalies.
- Normalization divides the larger table into the smaller table and links them using relationship.
- The normal form is used to reduce redundancy from the database table.

Tables

textapp_login

Name	DataType	Constraints	Description
id	int	Primarykey	ID of registered people
username	varchar(100)	Notnull	Name of registered people
password	varchar(100)	Notnull	Password of registered people
usertype	varchar(100)	Notnull	Type of registered people
status	varchar(100)	Notnull	Current status of the booking
driverid_id	bigint	Foreignkey	ID of driver

textapp_user

Name	DataType	Constraints	Description
id	int	Primarykey	ID of users
uname	varchar(100)	Notnull	Name of users
uaddress	varchar(100)	Notnul	Address of users
ucontact	varchar(100)	Notnull	Contact number of the users
uemail	varchar(100)	Notnull	Email id of users
upassword	varchar(100)	Notnull	Password of users

textapp_driver

Name	DataType	Constraints	Description
id	int	Primarykey	ID of users
dname	varchar(100)	Notnull	Name of drivers
daddress	varchar(100)	Notnul	Address of drivers
dcontact	varchar(100)	Notnull	Contact number of the drivers
demail	varchar(100)	Notnull	Email id of drivers
dpassword	varchar(100)	Notnull	Password of drivers
status	varchar(100)	Notnull	Current status of the booking

textapp_booking

Name	DataType	Constraints	Description
id	int	Primarykey	ID of users
date	varchar(100)	Notnull	Date of booking
status	varchar(100)	Notnull	Current status of the booking
did_id	bigint	Foreignkey	ID of driver
uid_id	bigint	Foreignkey	ID of user

Chapter 5

5 Development of the System

This website can be decomposed into a number of submodules.The submodules of car traffic sign recognizer are administrator, user and driver. Each submodule have specific objectives, to avoid unwanted coupling between modules and to increase cohesion, modules are again decomposed into submodules.

Chapter 6

6 System Testing

Testing is the penultimate step of software development.An elaborate testing of the data is prepared and the system is using the test data, while doing testing, errors are noted and correction is made. The users are trained to operate the developed system. Both hardware and software securities are made to run the developed system successfully. System testing is aimed at ensuring the system works accurately before the live operation commences.

6.1 Test Plan

6.1.1 Scope

This test plan will cover the following testing activities as identified in the testing strategy.

- White Box Testing

White box testing is a testing technique, that examines the program structure and derives test data from the program logic/code.In white box testing, the UI is bypassed.Inputs and outputs of ratings are tested directly at the code level and the results are compared against specifications.It reveals errors in "hidden" code.

- Black Box Testing

Black-box testing is a method of software testing that examines the functionality of an application based on the specifications.Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end-user would.

- Unit Testing

The module includes four main components to undergo unit testing.It checks the syntax error, logic error and validity of the program correctness.This test will be performed by the developers.

- Integration Testing

After all components pass the unit testing, they get to be tested whether they work correctly when they are running concurrently and communicating to each other.Integration tests exercise an entire subsystem and ensure that a set of components play nicely together.

- Internal Data Testing

We will test the validity of the data before it enters the database to avoid any problems that may face us in the database. We will test the encryption of the personal information of all the users along with the user names and passwords to ensure maximum security of the users privacy.

6.1.2 Software risk issues

In this section, the plan is to test the risk involved in critical issues such as: Difficult to run on visual studio due to large loading time. Some inherent software risks such as complexity exist; also these issues need to be identified. Proper network connection Working of Mysql database

6.1.3 Features to be tested

- Test whether correct user name and password allows you to login.
- Test whether invalid user name and password prevents you from login.
- Test whether there is any connection problem in the Server.
- Test whether the user and admin details are entered correctly.
- Test whether invalid data entry allows saving data successfully.
- Test whether all pages are loaded correctly.
- Test whether watermark is embedded and extracted properly.
- Test whether the provided security works correctly.

6.2 Test consolidation

6.2.1 Test item

The items or features to be tested in the test cases are included in the document. Each and every user input is tested. The present condition of the system is tested. It is checked to make sure that environment is ready for the application to work.

6.2.2 Input specifications

These are the inputs required to execute the test case. All required inputs including data elements and values are given below.

Data	Values
Email	A valid email ID
Password	Combination of characters and numbers
Phone number	10 Digit number

Chapter 7

7 System Implementation and Maintenance

7.1 Implementation

System implementation is the conversion of new system into an operating one which involves creating compatible files, training clients and installing hardware. User training is crucial for minimizing resistance to change and giving chance to prove its worth. Training aids user friendly manuals and healthy screens provide the user with a good start. Software maintenance follows conversion to the extent that changes are necessary to maintain satisfactory operations relative to changes in the user's environment. Maintenance often includes minor enhancements or corrections to the problem that surface late in the systems operations.In the implementation phase, the team builds the components either from scratch or by composition. Given the architecture document meant from the design phase and the requirement document from the analysis phase, the team should build exactly what has been requested, though there is still room for innovation and flexibility. For example, a component may be narrowly designed for this particular system, or the component may be made more general to satisfy a reusability

- Careful planning
- Investigation of system and constraints
- Design the methods to achieve changeover.
- Training the staff in the changed phase.
- Evaluation of change over method.
- The method of implementation and time scale to be adopted are found out initially.

7.2 Maintenance

This phase occurs as a result of deploying the whole system at the end users organization. They will perform the beta testing at the end users and inform to the developers about any needed modification to the application. The customer records all the problems that are encountered during the beta testing and reports these to the developer at regular intervals.

7.2.1 Corrective Maintenance

Even with the best quality assurance activities, it is likely that the customer will uncover defects in the software.Corrective maintenance changes the software to correct the defects. Corrective Maintenance activity may consist of repair, restoration or replacement of equipment. This activity will be result of a regular inspection, which identifies the failure in time for corrective maintenance to be planned and scheduled, then performed during a routine maintenance shutdown.

7.2.2 Adaptive Maintenance

Over time, the original environment(CPU, operating system, business rules, external product characteristics) for which the software was developed is likely to change. Adaptive maintenance results in modification to the software to accommodate changes to its external environment.

7.2.3 Enhanced Maintenance

As software is used, the customer/user will recognize additional functions that will provide the benefit. Perfect maintenance extends the software beyond its original functional requirements.

7.2.4 Preventive Maintenance

Computer software deteriorates due to change, and because of this preventive maintenance often called software re-engineering, must be conducted to enable the software to serve the needs of its end users. Preventive maintenance makes changes to computer programs so that they can be more easily corrected, adapted and enhanced.

Chapter 8

8 Conclusion and Future Scope

8.1 Conclusion

There are three types of users that interact with the system Admin, User and Driver. Each of these three types has different uses of the system so each of them has their own panel. Admin can manage all the features of website dynamically by login on admin panel. The users can detect the traffic signs and can also book a driver and view the bookings. The driver can view the bookings and user details and can approve them.

8.2 Future Scope

- Increased accuracy using better camera device
- Adaptability to dynamic environments

Appendix

A Data Flow Diagram

Data flow is the one of the best way of documenting the entire functionality of the system. For the system ,which will have data flows in and have some processing inside and then some data flow out from the system can be documented or represented effectively by means of data flow out from the system can be documented or represented effectively by means of data flow diagrams. The data flow diagram are a diagrammatic representation of the system,which has input,process and output. Once any system is represented using a data flow diagram we can identify the following things easily:

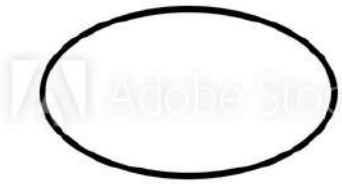
- Various entities interacting with the system are identified
- Flow of data from one entity to another is identified
- The various processes involved in between the interaction of two or more entities in the system are clearly pointed out
- The various data stores which hold the data in between the process,are clearly identified

Some Data Flow Diagram charting forms:

A.1 External source or receiver

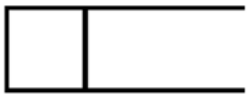
A source or sink is a person or part of organization ,which enters or receives information from the system,but is considered to be outlining the contest of data flow model.

A.2 Transform process



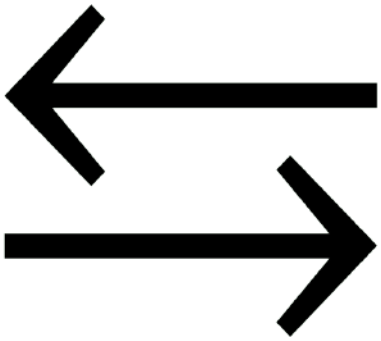
A process represents transformation where incoming data flowsare changed into outgoing data flow

A.3 Data Store



A data store is repository of data that is to be stored for use by one or more process may be as simple as buffer or queue or sophisticated as relational database. They should have clear names.If a process merely uses the contest of store and does not alter it,the arrowhead goes only form the store to the process. If a process alters the details in the store then double-headed arrow is used.

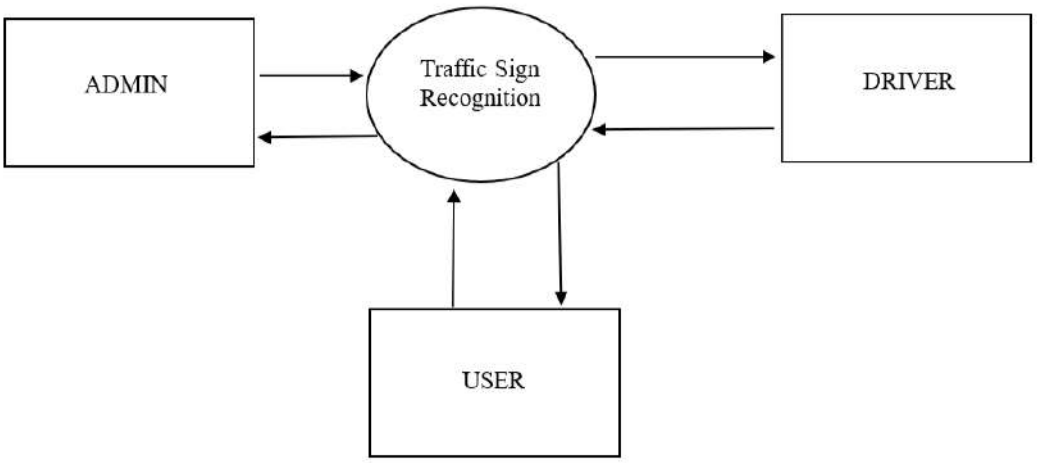
A.4 Data flow



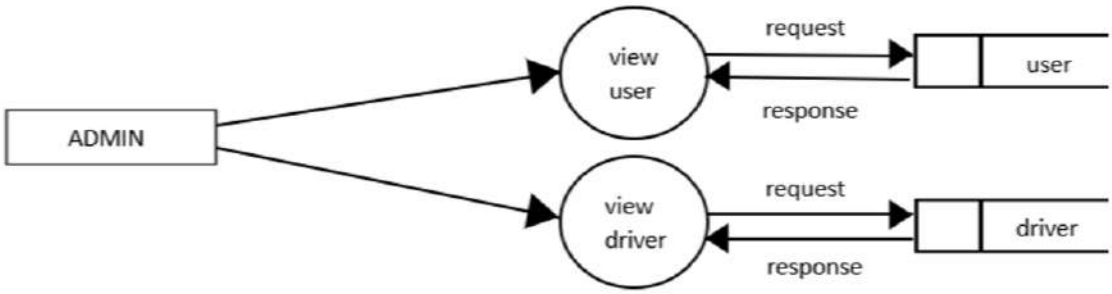
A data flow is a route, which enable packets of data to travel from one point to another.Data may flow,with arrowhead pointing in the direction of the flow

B Data Flow Diagrams

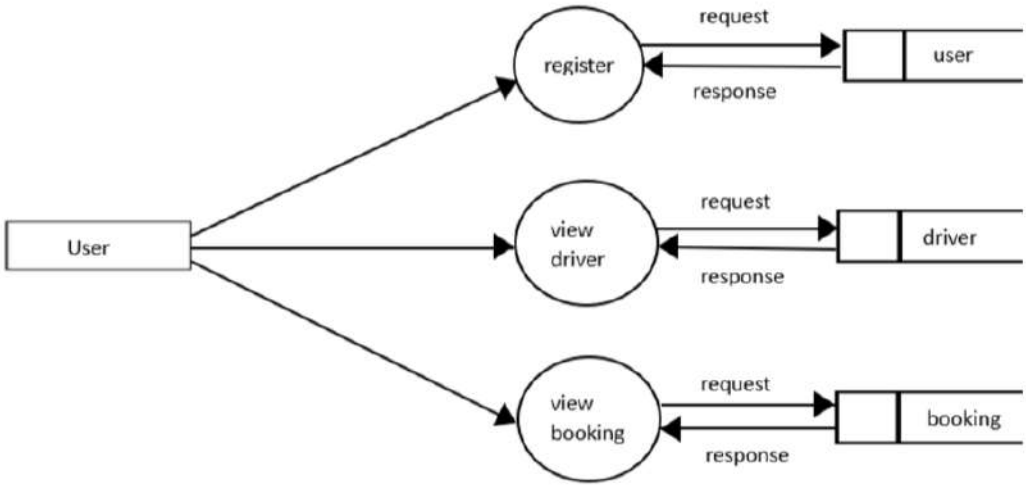
B.1 Level 0



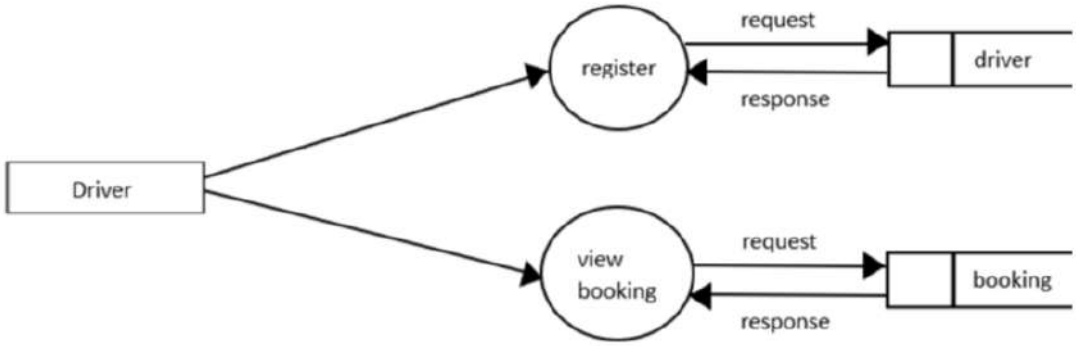
B.2 Level 1.1 - Admin



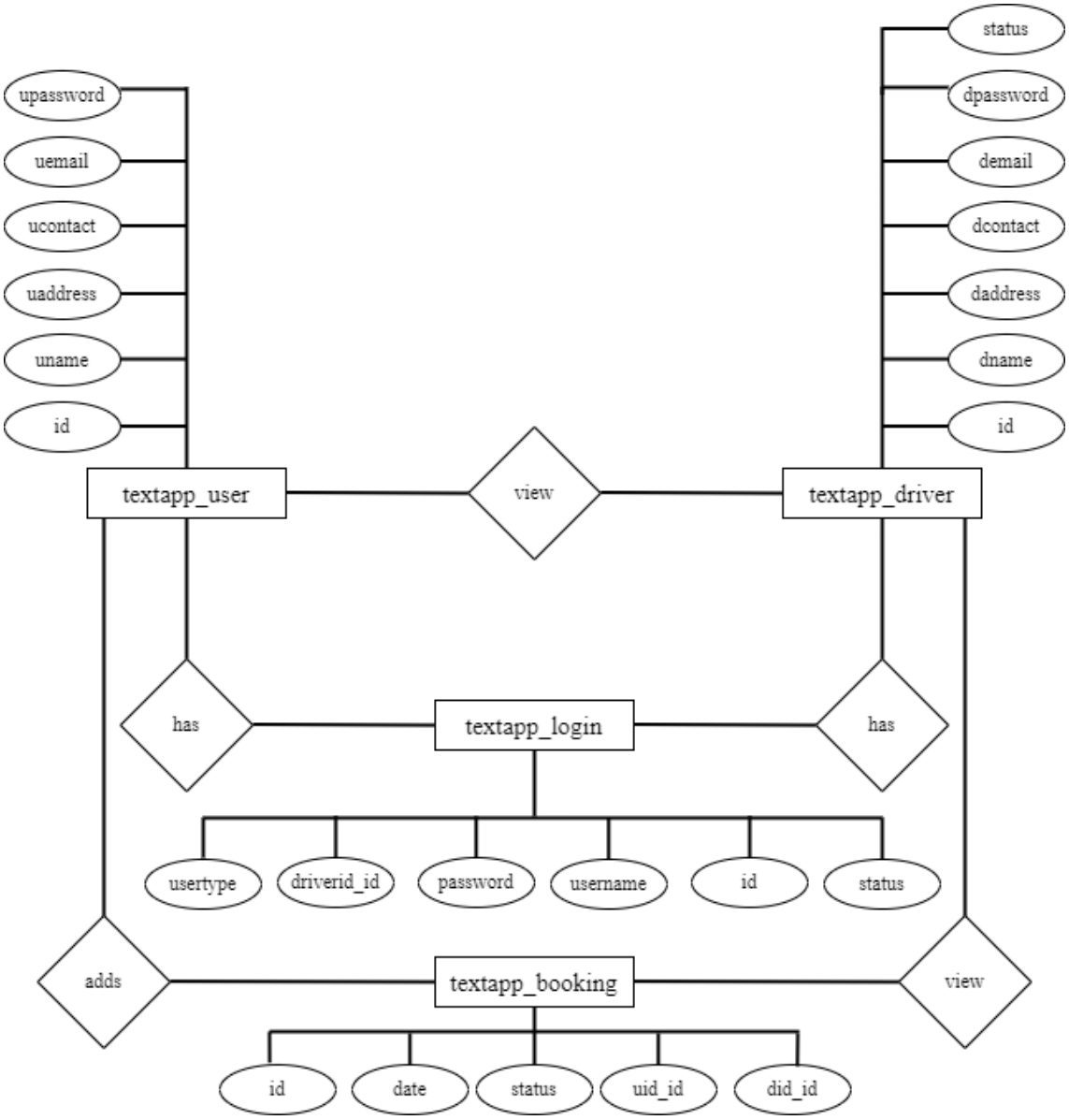
B.3 Level 1.2 - User



B.4 Level 1.3 - Driver

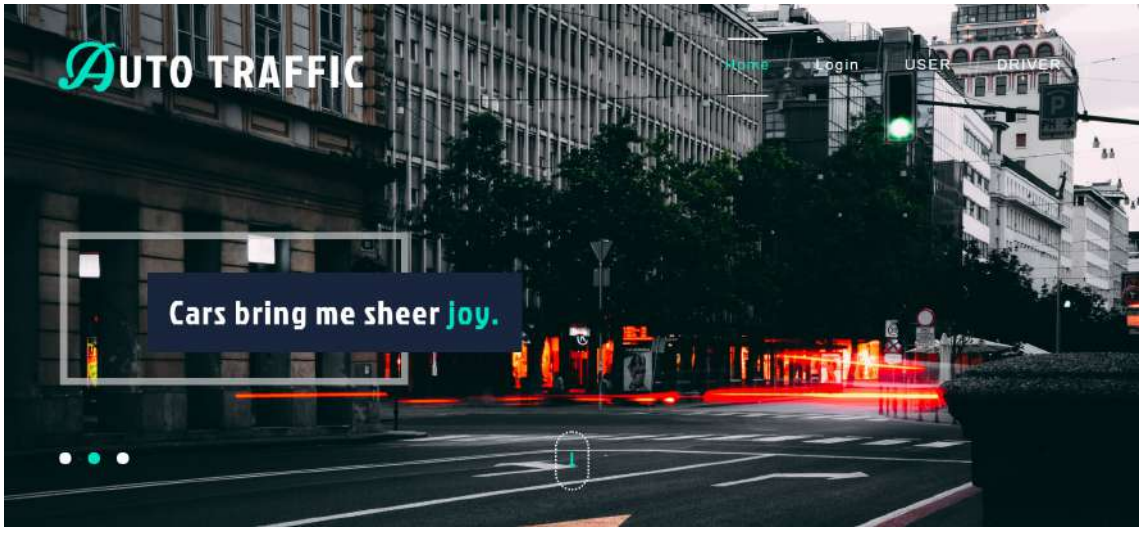


B.5 ER Diagram

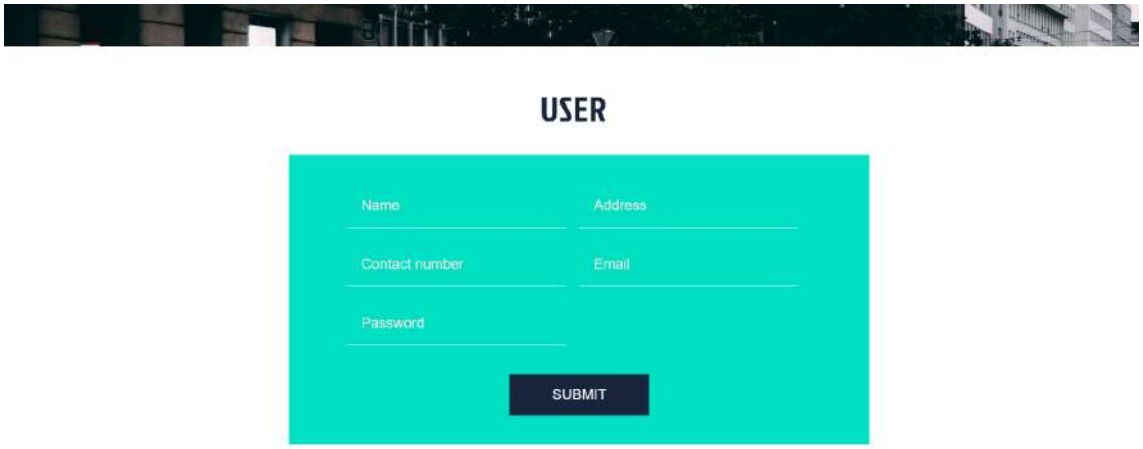


C USER INTERFACES

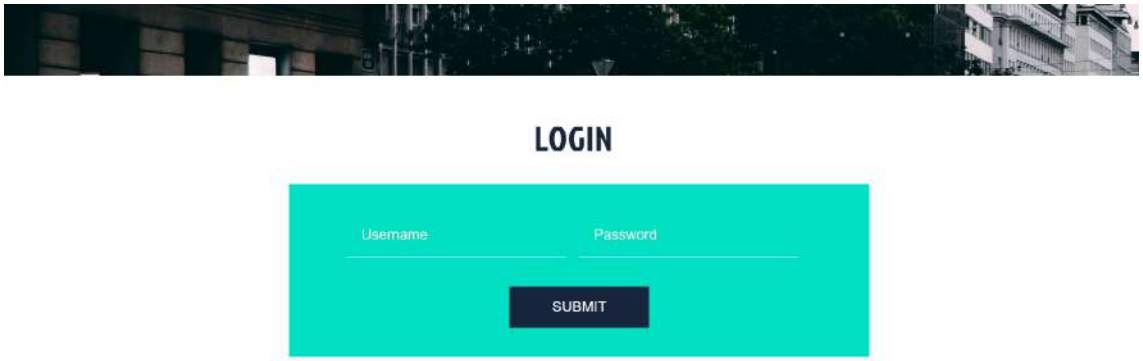
C.1 HOME



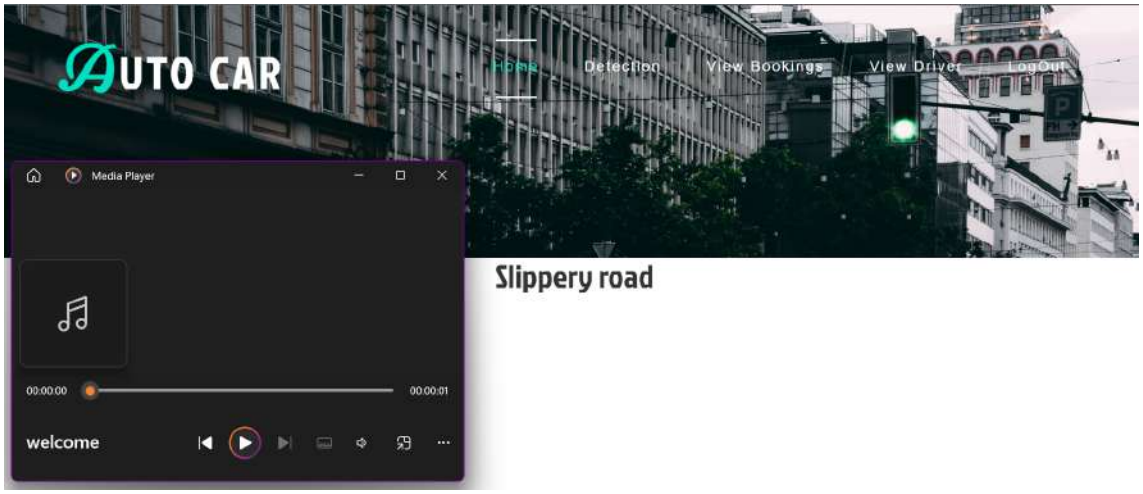
C.2 REGISTRATION



C.3 LOGIN



C.4 DETECTION



C.5 VIEW DRIVERS

NAME	ADDRESS	CONTACT	EMAIL	ACTION
Williams	Kodungalloor	8988713401	williams@gmail.com	Book Now
Simson	Thrissur	9099170787	simson@gmail.com	Book Now
Myna	Chalakkudi	8799136323	myna@gmail.com	Book Now

C.6 VIEW BOOKING

NAME	ADDRESS	CONTACT	EMAIL	STATUS	ACTION
Richard	Chalakkudi	8966139256	richard321@gmail.com		Approve
John	Aluva	8590939355	john@gmail.com		Approve
Philip	Kodungalloor	9890583346	philip@gmail.com		Approve

D Code

urls.py

```
from django.contrib import admin
from django.urls import path
from textapp import views
urlpatterns = [
    path('admin/', admin.site.urls),

    path('index/', views.index),
    path('adminhome/', views.adminhome),
    path('driverhome/', views.driverhome),
    path('userhome/', views.userhome),
    path('', views.index),
    path('login/', views.logins),
    path('userreg/', views.userreg),
    path('driverreg/', views.driverreg),
    path('udp/', views.udp),
    path('adminviewdriver/', views.adminviewdriver),
    path('adminviewuser/', views.adminviewuser),
    path('userviewdriver/', views.userviewdriver),
    path('viewbooking/', views.userviewbooking),
    path('userviewbooking/', views.userviewbooking),
    path('driverviewbooking/', views.driverviewbooking),
    path('ddetection/', views.ddetection),
    path('driverdetection/', views.driverdetection),
    path('imagebyenter/', views.imagebyenter),
    path('prediction/', views.predict),
    path('deletedriver/', views.deletedriver),
    path('deleteuser/', views.deleteuser),
    \# path('udp/', views.udp),

]
```

model.py

```
import os
from tensorflow.keras.preprocessing import image
import matplotlib.pyplot as plt
```

```
import numpy as np
import tensorflow.python.keras.utils as generic\_utils
import random,shutil
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dropout,Conv2D,Flatten,Dense,
MaxPooling2D, BatchNormalization
from tensorflow.keras.models import load\_model


def generator(dir, gen=image.ImageDataGenerator(rescale=1./255), shuffle=True,batch
\_size=1,target\_size=(24,24),class\_mode='categorical' ):

return gen.flow\_from\_directory(dir,batch\_size=batch\_size,shuffle=shuffle,
color\_mode='grayscale',class\_mode=class\_mode,target\_size=target\_size)

BS= 32
TS=(24,24)
train\_batch= generator('data/train',shuffle=True, batch\_size=BS,target\_size=TS)
valid\_batch= generator('data/valid',shuffle=True, batch\_size=BS,target\_size=TS)
SPE= len(train\_batch.classes)//BS
VS = len(valid\_batch.classes)//BS
print(SPE,VS)


\# img,labels= next(train\_batch)
\# print(img.shape)

model = Sequential([
    Conv2D(32, kernel\_size=(3, 3), activation='relu', input\_shape=(24,24,1)),
    MaxPooling2D(pool\_size=(1,1)),
    Conv2D(32,(3,3),activation='relu'),
    MaxPooling2D(pool\_size=(1,1)),
\#32 convolution filters used each of size 3x3
\#again
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D(pool\_size=(1,1)),

\#64 convolution filters used each of size 3x3
\#choose the best features via pooling
```

```
\#randomly turn neurons on and off to improve convergence
    Dropout(0.25),
\#flatten since too many dimensions, we only want a classification output
    Flatten(),
\#fully connected to get all relevant data
    Dense(128, activation='relu'),
\#one more dropout for convergence' sake :)
    Dropout(0.5),
\#output a softmax to squash the matrix into output probabilities
    Dense(2, activation='softmax')
])

model.compile(optimizer='adam',loss='categorical\_crossentropy',metrics=['accuracy'])

model.fit\_generator(train\_batch, validation\_data=valid\_batch,epochs=15,steps\_per

model.save('models/cnnCat2.h5', overwrite=True)
```

models.py

```
[verbetim]

from django.db import models




class Driver(models.Model):
    dname=models.CharField(max\_length=100,blank=True)
    daddress=models.CharField(max\_length=100,blank=True)
    dcontact=models.CharField(max\_length=100,blank=True)
    demail=models.CharField(max\_length=100,blank=True)
    dpassword=models.CharField(max\_length=100,blank=True)
    status=models.CharField(max\_length=100,blank=True)


class Login(models.Model):
    username=models.CharField(max\_length=100,blank=True)
    password=models.CharField(max\_length=100,blank=True)
```

```
    usertype=models.CharField(max\_length=100,blank=True)
    status=models.CharField(max\_length=100,blank=True)
    driverid=models.ForeignKey(Driver,null=True,on\_delete=models.CASCADE)



class User(models.Model):
    uname=models.CharField(max\_length=100,blank=True)
    uaddress=models.CharField(max\_length=100,blank=True)
    ucontact=models.CharField(max\_length=100,blank=True)
    uemail=models.CharField(max\_length=100,blank=True)
    upassword=models.CharField(max\_length=100,blank=True)


class Booking(models.Model):
    uid=models.ForeignKey(User,null=True,on\_delete=models.CASCADE)
    did=models.ForeignKey(Driver,null=True,on\_delete=models.CASCADE)
    date=models.CharField(max\_length=100,blank=True)
    status=models.CharField(max\_length=100,blank=True)
```

views.py

```
\# import email
from http.client import HTTPResponse
from django.shortcuts import render,HttpResponse,HttpResponseRedirect

from .models import *
from django.contrib import messages
from django.db.models import Max, Min,Count,Sum,Avg

import numpy as np
from PIL import Image
import cv2
import tensorflow as tf
import os
from flask\_cors import CORS, cross\_origin
from utils.utils import decodeImage
from predict import traffic
```

```
from django.db.models import Q
import cv2
import tensorflow as tf

def udp(request):
    import os
    from gtts import gTTS
    \# email="admin@gmail.com"
    \# password="admin"

    \# s=Login.objects.create(username=email,password=password,usertype='admin',statu
    \# s.save()
    \# messages.info(request,"already added")
    model\_path = "Traffic.h5"
    loaded\_model = tf.keras.models.load\_model(model\_path)
    videoCaptureObject = cv2.VideoCapture(0)
    print("**************************************************************")
    result = True
    while(result):
        ret,frame = videoCaptureObject.read()
        cv2.imwrite("NewPicture.jpg",frame)
        result = False
    videoCaptureObject.release()
    cv2.destroyAllWindows()
    imagename = "NewPicture.jpg"
    image = cv2.imread(imagename)
    print(image)
    print("**************************************************************")

    image\_fromarray = Image.fromarray(image, 'RGB')
    resize\_image = image\_fromarray.resize((30, 30))
    print("**************************************************************")

    expand\_input = np.expand\_dims(resize\_image,axis=0)
    input\_data = np.array(expand\_input)
    input\_data = input\_data/255
    pred = loaded\_model.predict(input\_data)
    result = pred.argmax()
    print("**************************************************************")
```

```
if result == 0:
    prediction = 'Speed limit (20km/h)'
    print("image",prediction)
elif result == 1:
    prediction = 'Speed limit (30km/h)'
    print("image",prediction)
elif result == 2:
    prediction = 'Speed limit (50km/h)'
    print("image",prediction)
elif result == 3:
    prediction = 'Speed limit (60km/h)'
    print("image",prediction)
elif result == 4:
    prediction = 'Speed limit (70km/h)'
    print("image",prediction)
elif result == 5:
    prediction = 'Speed limit (80km/h)'
    print("image",prediction)
elif result == 6:
    prediction = 'End of speed limit (80km/h)'
    print("image",prediction)
elif result == 7:
    prediction = 'Speed limit (100km/h)'
    print("image",prediction)
elif result == 8:
    prediction = 'Speed limit (120km/h)'
    print("image",prediction)
elif result == 9:
    prediction = 'No passing'
    print("image",prediction)
elif result == 10:
    prediction = 'No passing veh over 3.5 tons'
    print("image",prediction)
elif result == 11:
    prediction = 'Right-of-way at intersection'
    print("image",prediction)
elif result == 12:
    prediction = 'Priority road'
    print("image",prediction)
```

```
elif result == 13:
    prediction = 'Yield'
    print("image",prediction)
elif result == 14:
    prediction = 'Stop'
    print("image",prediction)
elif result == 15:
    prediction = 'No vehicles'
    print("image",prediction)
elif result == 16:
    prediction = 'Veh > 3.5 tons prohibited'
    print("image",prediction)
elif result == 17:
    prediction = 'No entry'
    print("image",prediction)
elif result == 18:
    prediction = 'General caution'
    print("image",prediction)
elif result == 19:
    prediction = 'Dangerous curve left'
    print("image",prediction)
elif result == 20:
    prediction = 'Dangerous curve right'
    print("image",prediction)
elif result == 21:
    prediction = 'Double curve'
    print("image",prediction)
elif result == 22:
    prediction = 'Bumpy road'
    print("image",prediction)
elif result == 23:
    prediction = 'Slippery road'
    print("image",prediction)
elif result == 24:
    prediction = 'Road narrows on the right'
    print("image",prediction)
elif result == 25:
    prediction = 'Road work'
    print("image",prediction)
elif result == 26:
```

```
        prediction = 'Traffic signals'
        print("image",prediction)
    elif result == 27:
        prediction = 'Pedestrians'
        print("image",prediction)
    elif result == 28:
        prediction = 'Children crossing'
        print("image",prediction)
    elif result == 29:
        prediction = 'Bicycles crossing'
        print("image",prediction)
    elif result == 30:
        prediction = 'Beware of ice/snow'
        print("image",prediction)
    elif result == 31:
        prediction = 'Wild animals crossing'
        print("image",prediction)
    elif result == 32:
        prediction = 'End speed + passing limits'
        print("image",prediction)
    elif result == 33:
        prediction = 'Turn right ahead'
        print("image",prediction)
    elif result == 34:
        prediction = 'Turn left ahead'
        print("image",prediction)
    elif result == 35:
        prediction = 'Ahead only'
        print("image",prediction)
    elif result == 36:
        prediction = 'Go straight or right'
        print("image",prediction)
    elif result == 37:
        prediction = 'Go straight or left'
        print("image",prediction)
    elif result == 38:
        prediction = 'Keep right'
        print("image",prediction)
    elif result == 39:
        prediction = 'Keep left'
```

```
        print("image",prediction)
    elif result == 40:
        prediction = 'Roundabout mandatory'
        print("image",prediction)
    elif result == 41:
        prediction = 'End of no passing'
        print("image",prediction)
    elif result == 42:
        prediction = 'End no passing veh > 3.5 tons'
        print("image",prediction)
    language = 'en'
    myobj = gTTS(text=prediction, lang=language, slow=False)
    print("**************************************************")
    print(prediction)
    myobj.save("welcome.mp3")
    os.system("welcome.mp3")
    return render(request,"user/prediction.html",{"prediction":prediction})



    return HttpResponseRedirect("/prediction")


from django.core.files.storage import FileSystemStorage

def imagebyenter(request):
    import os
    from gtts import gTTS
    if request.POST:
        image=request.FILES['image']
        saved\_filename = f"./uploads/{image.name}"
        with open(saved\_filename, "wb") as opened:
            for chunk in image.chunks():
                    opened.write(chunk)

        fs=FileSystemStorage()
        filename=fs.save(image.name,image)
        fileurl=fs.url(filename)

        model\_path = "Traffic.h5"
```

```
loaded\_model = tf.keras.models.load\_model(model\_path)

imagename = image
print(saved\_filename)
image = cv2.imread(saved\_filename)

image\_fromarray = Image.fromarray(image, 'RGB')
resize\_image = image\_fromarray.resize((30, 30))
expand\_input = np.expand\_dims(resize\_image,axis=0)
input\_data = np.array(expand\_input)
input\_data = input\_data/255
pred = loaded\_model.predict(input\_data)
result = pred.argmax()


if result == 0:
    prediction = 'Speed limit (20km/h)'
    print("image",prediction)
elif result == 1:
    prediction = 'Speed limit (30km/h)'
    print("image",prediction)
elif result == 2:
    prediction = 'Speed limit (50km/h)'
    print("image",prediction)
elif result == 3:
    prediction = 'Speed limit (60km/h)'
    print("image",prediction)
elif result == 4:
    prediction = 'Speed limit (70km/h)'
    print("image",prediction)
elif result == 5:
    prediction = 'Speed limit (80km/h)'
    print("image",prediction)
elif result == 6:
    prediction = 'End of speed limit (80km/h)'
    print("image",prediction)
elif result == 7:
    prediction = 'Speed limit (100km/h)'
    print("image",prediction)
elif result == 8:
```

```
        prediction = 'Speed limit (120km/h)'
        print("image",prediction)
    elif result == 9:
        prediction = 'No passing'
        print("image",prediction)
    elif result == 10:
        prediction = 'No passing veh over 3.5 tons'
        print("image",prediction)
    elif result == 11:
        prediction = 'Right-of-way at intersection'
        print("image",prediction)
    elif result == 12:
        prediction = 'Priority road'
        print("image",prediction)
    elif result == 13:
        prediction = 'Yield'
        print("image",prediction)
    elif result == 14:
        prediction = 'Stop'
        print("image",prediction)
    elif result == 15:
        prediction = 'No vehicles'
        print("image",prediction)
    elif result == 16:
        prediction = 'Veh > 3.5 tons prohibited'
        print("image",prediction)
    elif result == 17:
        prediction = 'No entry'
        print("image",prediction)
    elif result == 18:
        prediction = 'General caution'
        print("image",prediction)
    elif result == 19:
        prediction = 'Dangerous curve left'
        print("image",prediction)
    elif result == 20:
        prediction = 'Dangerous curve right'
        print("image",prediction)
    elif result == 21:
        prediction = 'Double curve'
```

```
        print("image",prediction)
    elif result == 22:
        prediction = 'Bumpy road'
        print("image",prediction)
    elif result == 23:
        prediction = 'Slippery road'
        print("image",prediction)
    elif result == 24:
        prediction = 'Road narrows on the right'
        print("image",prediction)
    elif result == 25:
        prediction = 'Road work'
        print("image",prediction)
    elif result == 26:
        prediction = 'Traffic signals'
        print("image",prediction)
    elif result == 27:
        prediction = 'Pedestrians'
        print("image",prediction)
    elif result == 28:
        prediction = 'Children crossing'
        print("image",prediction)
    elif result == 29:
        prediction = 'Bicycles crossing'
        print("image",prediction)
    elif result == 30:
        prediction = 'Beware of ice/snow'
        print("image",prediction)
    elif result == 31:
        prediction = 'Wild animals crossing'
        print("image",prediction)
    elif result == 32:
        prediction = 'End speed + passing limits'
        print("image",prediction)
    elif result == 33:
        prediction = 'Turn right ahead'
        print("image",prediction)
    elif result == 34:
        prediction = 'Turn left ahead'
        print("image",prediction)
```

```
        elif result == 35:
            prediction = 'Ahead only'
            print("image",prediction)
        elif result == 36:
            prediction = 'Go straight or right'
            print("image",prediction)
        elif result == 37:
            prediction = 'Go straight or left'
            print("image",prediction)
        elif result == 38:
            prediction = 'Keep right'
            print("image",prediction)
        elif result == 39:
            prediction = 'Keep left'
            print("image",prediction)
        elif result == 40:
            prediction = 'Roundabout mandatory'
            print("image",prediction)
        elif result == 41:
            prediction = 'End of no passing'
            print("image",prediction)
        elif result == 42:
            prediction = 'End no passing veh > 3.5 tons'
            print("image",prediction)
        language = 'en'
        myobj = gTTS(text=prediction, lang=language, slow=False)
        myobj.save("welcome.mp3")
        os.system("welcome.mp3")
        return render(request,"user/prediction.html",{"prediction":prediction})


    return render(request,"user/enterimage.html")

def predict(request):

    return render(request,"user/prediction.html")
```

```
def index(request):
    return render(request,"common/index.html")

def adminhome(request):
    return render(request,"admin/index.html")

def userhome(request):
    return render(request,"user/index.html")

def driverhome(request):
    return render(request,"driver/index.html")

def driverreg(request):
    if request.POST:
        name=request.POST.get("name")
        address=request.POST.get("address")
        contact=request.POST.get("contact")
        email=request.POST.get("email")
        password=request.POST.get("password")

        s=Driver.objects.create(dname=name,daddress=address,dcontact=contact,demail=e
        s.save()
        did=Driver.objects.aggregate(Max('id'))
        print(did)
        did=did['id\_\_max']
        did=Driver.objects.get(id=did)
        print(did)

    \# print(mark)
        s=Login.objects.create(username=email,password=password,usertype='driver',
        status='requested',driverid=did)
        s.save()
        messages.info(request,"already added")
    return render(request,"common/driverreg.html")



def userreg(request):
```

```
    if request.POST:
        name=request.POST.get("name")
        address=request.POST.get("address")
        contact=request.POST.get("contact")
        email=request.POST.get("email")
        password=request.POST.get("password")

        s=User.objects.create(uname=name,uaddress=address,ucontact=contact,
            uemail=email,upassword=password)
        s.save()
        s=Login.objects.create(username=email,password=password,
             usertype='user',status='requested')
        s.save()
        messages.info(request,"already added")
    return render(request,"common/userreg.html")


def logins(request):
    msg=""
    if request.POST:
        name=request.POST.get("username")
        password=request.POST.get("password")
        print(name)
        print(password)

        s=Login.objects.filter(Q(username=name),Q(password=password))

        try:
            if s[0].usertype=='admin':
                msg="Success"
                return HttpResponseRedirect("/adminhome")
            elif s[0].usertype=='user':
                s=User.objects.get(uemail=name)
                id=s.id
                request.session['uid']=id
                msg="Success"
                return HttpResponseRedirect("/userhome")
            elif s[0].usertype=='driver':
                s=Driver.objects.get(demail=name)
                id=s.id
```

```
                request.session['did']=id
                msg="Success"
                return HttpResponseRedirect("/driverhome")

        except:
            msg="invalid Username Or password"


    \# return HttpResponseRedirect('/login')

    return render(request,"common/login.html",{'msg':msg})




def adminviewdriver(request):
    s=Driver.objects.filter()
    print(s)
    if request.GET:
        id=request.GET.get("id")
        s=Driver.objects.get(id=id)
        username=s.demail
        s=Login.objects.filter(username=username).update(status='approved')
        s=Driver.objects.filter(id=id).update(status='approved')
        return HttpResponseRedirect("/adminviewdriver")
    return render(request,"admin/viewdriver.html",{"data":s})

def deletedriver(request):
    id=request.GET.get('id')
    Driver.objects.filter(id=id).delete()
    return HttpResponseRedirect("/adminviewdriver")


def adminviewuser(request):
    s=User.objects.all()
    print(s)
    return render(request,"admin/viewuser.html",{"data":s})

def deleteuser(request):
    id=request.GET.get('id')
```

```
    User.objects.filter(id=id).delete()
    return HttpResponseRedirect("/adminviewuser")


def userviewdriver(request):
    s=Driver.objects.all()
    uid=request.session['uid']
    if request.GET:
        id=request.GET.get("id")
        driverdetails=Driver.objects.get(id=id)
        userdetails=User.objects.get(id=uid)
        import datetime
        date=datetime.datetime.now()
        s=Booking.objects.create(uid=userdetails,did=driverdetails,date=date,status='
        return HttpResponseRedirect("/userviewdriver")
    return render(request,"user/viewdriver.html",{"data":s})




def userviewbooking(request):
    uid=request.session['uid']
    select=Booking.objects.filter(uid=uid)
    return render(request,"user/viewbooking.html",{"data":select})




def driverviewbooking(request):
    uid=request.session['did']
    print(uid)
    select=Booking.objects.filter(did=uid)
    print(select)
    abcd='approved'
    if request.GET:
        id=request.GET.get("id")
        select=Booking.objects.filter(id=id).update(status='approved')
        return HttpResponseRedirect("/driverhome")

    return render(request,"driver/viewbooking.html",{"data":select,"abcd":abcd})
```

```
def ddetection(request):
    from textapp import cd
    return HttpResponseRedirect("/userhome")



def driverdetection(request):
    from textapp import cd
    return HttpResponseRedirect("/driverhome")
```

predict.py

```
import numpy as np
from PIL import Image
import cv2
import tensorflow as tf

class traffic:
    def \_\_init\_\_(self,filename):
        self.filename =filename


    def trafficsign(self):

        model\_path = "Traffic.h5"
        loaded\_model = tf.keras.models.load\_model(model\_path)

        imagename = self.filename
        image = cv2.imread(imagename)

        image\_fromarray = Image.fromarray(image, 'RGB')
        resize\_image = image\_fromarray.resize((30, 30))
        expand\_input = np.expand\_dims(resize\_image,axis=0)
        input\_data = np.array(expand\_input)
        input\_data = input\_data/255
```

```
pred = loaded\_model.predict(input\_data)
result = pred.argmax()

if result == 0:
    prediction = 'Speed limit (20km/h)'
    return [{"image": prediction}]
elif result == 1:
    prediction = 'Speed limit (30km/h)'
    return [{"image": prediction}]
elif result == 2:
    prediction = 'Speed limit (50km/h)'
    return [{"image": prediction}]
elif result == 3:
    prediction = 'Speed limit (60km/h)'
    return [{"image": prediction}]
elif result == 4:
    prediction = 'Speed limit (70km/h)'
    return [{"image": prediction}]
elif result == 5:
    prediction = 'Speed limit (80km/h)'
    return [{"image": prediction}]
elif result == 6:
    prediction = 'End of speed limit (80km/h)'
    return [{"image": prediction}]
elif result == 7:
    prediction = 'Speed limit (1000km/h)'
    return [{"image": prediction}]
elif result == 8:
    prediction = 'Speed limit (120km/h)'
    return [{"image": prediction}]
elif result == 9:
    prediction = 'No passing'
    return [{"image": prediction}]
elif result == 10:
    prediction = 'No passing veh over 3.5 tons'
    return [{"image": prediction}]
elif result == 11:
    prediction = 'Right-of-way at intersection'
    return [{"image": prediction}]
elif result == 12:
```

```
        prediction = 'Priority road'
        return [{"image": prediction}]
    elif result == 13:
        prediction = 'Yield'
        return [{"image": prediction}]
    elif result == 14:
        prediction = 'Stop'
        return [{"image": prediction}]
    elif result == 15:
        prediction = 'No vehicles'
        return [{"image": prediction}]
    elif result == 16:
        prediction = 'Veh > 3.5 tons prohibited'
        return [{"image": prediction}]
    elif result == 17:
        prediction = 'No entry'
        return [{"image": prediction}]
    elif result == 18:
        prediction = 'General caution'
        return [{"image": prediction}]
    elif result == 19:
        prediction = 'Dangerous curve left'
        return [{"image": prediction}]
    elif result == 20:
        prediction = 'Dangerous curve right'
        return [{"image": prediction}]
    elif result == 21:
        prediction = 'Double curve'
        return [{"image": prediction}]
    elif result == 22:
        prediction = 'Bumpy road'
        return [{"image": prediction}]
    elif result == 23:
        prediction = 'Slippery road'
        return [{"image": prediction}]
    elif result == 24:
        prediction = 'Road narrows on the right'
        return [{"image": prediction}]
    elif result == 25:
        prediction = 'Road work'
```

```
        return [{"image": prediction}]
    elif result == 26:
        prediction = 'Traffic signals'
        return [{"image": prediction}]
    elif result == 27:
        prediction = 'Pedestrians'
        return [{"image": prediction}]
    elif result == 28:
        prediction = 'Children crossing'
        return [{"image": prediction}]
    elif result == 29:
        prediction = 'Bicycles crossing'
        return [{"image": prediction}]
    elif result == 30:
        prediction = 'Beware of ice/snow'
        return [{"image": prediction}]
    elif result == 31:
        prediction = 'Wild animals crossing'
        return [{"image": prediction}]
    elif result == 32:
        prediction = 'End speed + passing limits'
        return [{"image": prediction}]
    elif result == 33:
        prediction = 'Turn right ahead'
        return [{"image": prediction}]
    elif result == 34:
        prediction = 'Turn left ahead'
        return [{"image": prediction}]
    elif result == 35:
        prediction = 'Ahead only'
        return [{"image": prediction}]
    elif result == 36:
        prediction = 'Go straight or right'
        return [{"image": prediction}]
    elif result == 37:
        prediction = 'Go straight or left'
        return [{"image": prediction}]
    elif result == 38:
        prediction = 'Keep right'
        return [{"image": prediction}]
```

```
elif result == 39:
    prediction = 'Keep left'
    return [{"image": prediction}]
elif result == 40:
    prediction = 'Roundabout mandatory'
    return [{"image": prediction}]
elif result == 41:
    prediction = 'End of no passing'
    return [{"image": prediction}]
elif result == 42:
    prediction = 'End no passing veh > 3.5 tons'
    return [{"image": prediction}]
else:
    return [{"ERROR": "Please select another image. !!!"}]
```

VENTURE VISTA

PROJECT REPORT

Submitted By
SHAMNAS A.P
Reg. No. CCAVBCA013

for the award of the Degree of

Bachelor of Computer Application (B.C.A.)

(University of Calicut)

under the guidance of

Ms. Soumya P.S
Assistant Professor

Bachelor of Computer Application
DEPARTMENT OF COMPUTER SCIENCE
CHRIST COLLEGE(Autonomous)
IRINJALAKUDA, KERALA
INDIA

2021-24

DEPARTMENT OF COMPUTER SCIENCE

CHRIST COLLEGE (AUTONOMOUS)

IRINJALAKUDA

CERTIFICATE

This is to certify that the project report entitled ***"VENTURE VISTA"*** *is a bonafide record of the project work done by* ***SHAMNAS AP*** *in partial fulfillment of the requirement for the sixth semester of* ***Bachelor of Computer Application*** *in Department of Computer Science of* ***CHRIST COLLEGE (AUTONOMOUS) IRINJALAKUDA***

.

Ms. SOUMYA P.S
Assistant Professor,CS
Internal Guide

Ms. SINI THOMAS
Head of the Department
Computer Science

EXTERNAL EXAMINER

INTERNAL EXAMINER

DECLARATION

We hereby declare that this project work **"VENTURE VISTA"** submitted to Christ College (Autonomous) Irinjalakuda, affiliated to Calicut University in partial fulfillment of the requirement for the award of the Bachelor of Computer Application, is a record of original work done by us, under the guidance of Ms. SOUMYA P.S, Department of computer Science.

Place: Irinjalakuda SHAMNAS AP

ACKNOWLEDGEMENT

First and foremost I like to thank Lord almighty for his providence and for being the guiding light throughout the project. I wish to express my sincere gratitude to my beloved Department head for giving me all the facilities for my project. I take this opportunity to express my gratitude to the class teacher Ms. SOUMYA P.S and head of the department Ms. SINI THOMAS who has been supported me throughout the course of this project. I are thankful for her aspiring guidance and valuable advice during the project work. I express my sincere thanks to my project guide Ms. SOUMYA P.S for supporting and guiding throughout the project.I would take this opportunity to specially thank all other faculty members for their constant and continuous motivation. Finally I would like to thank my family and friends for giving valuable advice and moral support throughout my project.

ABSTRACT

VENTURE VISTA Our travel guide website aims to inspire and assist travelers in exploring new destinations worldwide. From detailed city guides to off-the-beaten-path adventures, we provide a wealth of information to help users plan unforgettable trips. With curated recommendations for accommodations, dining, activities, and transportation, our platform caters to all types of travelers, security scores,live money exchange rates. By combining expert insights with user reviews and ratings, we strive to be a one-stop resource for discovering and planning your next adventure.

Contents

Chapter 1

1 Introduction

This is an web application that helps the users to know more about the major attractions near to them. The application provides all the details about tourist places or major attractions like their opening time, closing time, history and other details. The app also provides all the details of the hotels available for the users. The app can also provide the best route to the places you have selected. The app will come in handy for those who travel a lot, or like to travel to new places.

1.1 Overview

The objective of the TRAVEL GUIDE APPLICATION is to design an simple and adaptable application that helps the users to know more about the major attractions near to them. The web application includes many interesting features such as providing place suggestions to visit which best suites for the preferences you have given.It is a User friendly application.

Chapter 2

2 System Analysis

2.1 Purpose

A travel guide application serves as a comprehensive resource for travelers, offering a wide range of information and features to enhance their travel experiences. Its primary purpose is to provide users with detailed insights into various destinations, including popular attractions, historical sites, cultural experiences, dining options, accommodations, and local services.

By leveraging the app, users can efficiently plan their trips, exploring different destinations based on their interests and preferences. They can access detailed descriptions, reviews, and ratings for various attractions and services, helping them make informed decisions about their travel itinerary.

Furthermore, a travel guide application often includes features such as offline maps, itinerary planners, currency converters, and language translators, which can be invaluable for travelers navigating unfamiliar environments. These features can help users navigate their destinations more effectively, communicate with locals, and manage their travel budgets.

Overall, a travel guide application aims to simplify the travel planning process, inspire users to explore new destinations, and enhance their overall travel experiences by providing them with relevant and reliable information.
.

2.1.1 Existing System

The existing system are apps that provides data about tour packages and so on. Apps like this, which provide all details of the places are not available. Limitations of existing system :We cannot get all the details about the places we are looking for from a single application.

2.1.2 Proposed System

The proposed system is an web application which provides all the details related to the places you are looking for your next trip. We also provide a feature that gives you suggestions of the places near you to visit, based on the preferences you have provided. Advantages of proposed system :The application gives all the details about the places you are looking for.It also

provide place suggestions to visit which best suites for the preferences you have given.

2.2 Problem definition

To know what the problem is and what the needs are before developing it. Here we design an application for travel guiding.

2.3 FEASIBILITY STUDY

After the problem is clearly understood and solutions proposed, the next step is to conduct the feasibility study. Feasibility study is defined as evaluation or analysis of the potential impact of the proposed project or program. The objective is to determine whether the proposed system is feasible. There are three aspects of feasibility study to which the proposed system is subjected as discussed below.

2.3.1 Technical Feasibility

Technical feasibility assess whether the current technical resources are sufficient for the new system. We can upgrade the level of technology for supporting our website. We check whether the proposed system can be implemented in the present system without supporting the existing hardware.

2.3.2 Economical Feasibility

Economic feasibility determines whether the time and money are available to develop the system. There is no additional hardware used to develop the site, it is inexpensive to build.

2.3.3 Operational Feasibility

Operational feasibility determines if the human resources are available to operate the system once it has been installed. No extra training is needed to use this system. Anyone who has knowledge in internet in English language can easily use the system. The resources that are required to implement or install are already available with the organization.

Chapter 3

3 Software Requirement Specification

3.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the TRAVEL GUIDE APPLICATION. It illustrate the purpose and complete description for the development of the system. It explain system constraints, interface and interactions with other external applications. This document is primarily intended to be proposed to provide all the details related to the places you are looking for your next trip.

3.2 Scope

The application gives all the details about the places you are looking for.It also provide place suggestions to visit which best suites for the preferences you have given.

3.3 Overall Description

This section give an overview of our application, TRAVEL GUIDE APPLICATION. This is an web application that helps the users to know more about the major attractions near to them. The application provides all the details about tourist places or major attractions like their opening time, closing time, history and other details. The app also provides all the details of the hotels available for the users. The app can also provide the best route to the places you have selected. The app will come in handy for those who travel a lot, or like to travel to new places.

3.3.1 Product Perspective

TRAVEL GUIDE APPLICATION is mainly used to know the details about your next trip. It also provide features that gives you suggestions to visit according to the preferences you have given.

3.3.2 Product Functionality

Through this system admin can upload various data. User can view the major attractions and their history and other details available in an area.

Users can also view all the information about the hotels available in a specific area.

3.3.3 Users and Characteristics

There are two types of users that interact with the application, people who wish to travel and the guide. Each of these have different tasks which is performed. Admin is able to upload details of places and can view the feedback.

3.4 Specific Requirements

3.4.1 Hardware Requirements

- System: IBM-Compatible PC
- Processor: Intel Core i3
- Speed: Above 1GHz
- RAM capacity: 4 GB
- Hard Dsk drive: 500 GB

3.4.2 Software Requirements

- Front End : React
- Back End : Djanko Python
- Database : MySql

3.5 Functional Requirements

It contains three main modules.

- 1.Admin
- 2.User

Admin

The admin will upload the details of the places. All the details that are provided in the app is uploaded by the admin. The admin can view user's feedbacks and can take suitable decisions based on the user's decisions.

User

The users can view the major attractions and their history and other details available in an area. Users can also view all the information about the hotels available in a specific area. The users can get the best route available to the selected place from their current location, which is calculated by the app and shown in the map. Users can also get notifications on the places that are suggested by the app based on the preferences they have given.

3.6 Non Functional Requirements

Non-functional requirements define the overall qualities or attributes of the resulting system. Non-functional system place restrictions on the product being developed, the developed process, and specify external constraints that the product must meet. Examples of non-functional requirements include safety, security, usability, reliability and performance requirements. Project management issues (costs, time and schedule) are often considered as non-functional requirements. The principal non - functional constraints which are relevant to critical systems :

- performance
- security
- safety
- usability

Performance

Performance requirements concern the speed of operation of a system. Types of performance requirements :

- Response requirements (how quickly the system reacts to a user input).
- Throughput requirements (how much the system can accomplish within a specified amount of time).
- Availability requirements (is the system available for service when requested by end users). The speed of operation of this system is adequate for the requirements.

Reliability

- Reliability is the ability of a system to perform its required function under stated conditions for a specified period of time.

- constraints on the runtime behavior of the system. This system is reliable because its functionalities can be done on the required conditions.

Safety

Safety requirements are not required which exclude unsafe situation from the possible solution of the system.

Usability

Usability is the ease with which a user can learn to operate, prepare inputs for, and interpret outputs of system or components. Usability requirements include :

- information error messages.
- well-formed user interfaces.

3.7 Interface Requirements

3.7.1 Hardware interfaces

The system must run over the internet, all the hardware shall require to connect internet will be hardware interface for the system. As for example modem, WAN-LAN.

3.7.2 Software interfaces

Software interface required for the working of the project is the appropriate operating system.

3.7.3 Communication interfaces

The user of the site communicate using the network connectivity and the data set is accessed

3.8 Security Requirements

- User accesses only their account.
- Validation of input is handled.
- This application containing the computer systems is physically secured against arms or surreptitious entry by intruders.
- Users must be authorized carefully to reduce changes of any such user giving access to an intruder in exchange for a bribe or other favour.

3.9 Platform Used

Windows 10 is a major version of the Microsoft Windows operating system that was released on July 29, 2015.It is built on the Windows NT kernel and follows windows 8.Part of the reason Microsoft decided to name the 2015 release "Windows 10"(and skipped "windows 9")is because the operating system is designed to be a new direction for Microsoft. One of the primary aims of windows 10 is to Unify the windows experience across multiple devices, such desktop computers, tablets, and smartphones. As part of this effort, Microsoft developed Windows10 Mobile alongside Windows 10 to replaces Windows Phone - Microsoft's previous mobile OS. Windows 10 also integrates other Microsoft services, such as Xbox Live and the Cortana voice recognition assistant.

3.10 Technologies Used

Djanko

Django is a free and open-source web framework, written in Python, that follows the model-template-views (MTV) architectural pattern. It is used to build database-driven websites. Django takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel.

React

React.js is an open-source JavaScript library used to build user interfaces (UIs) for single-page applications. React manages the view layer and is used for the development of both web and mobile applications.React is a component-based library, which means that UIs are built by combining reusable components. Components are self-contained pieces of code that can be used to render HTML, CSS, and JavaScript.React uses a virtual DOM, which is a lightweight representation of the real DOM. The virtual DOM is used to efficiently update the UI when data changes.React is a popular choice for building UIs because it is fast, scalable, and easy to learn.

MySQL

MySQL, pronounced either "My S-Q-L" or "My Squel, " is an open source relational database management system. It is based on the structure query

language (SQL), which is used for adding, removing and modifying information in the database. Standard SQL commands, such as ADD, DROP, INSERT,and UPDATE can be used with MySQL.

What distinguishes PHP from something like client - side Javascript is that the code is executed on the server, generating HTML which is then sent to the client. The client would receive the results of running that script, but would not know what the underlying code was. You can even configure your web server to process all your HTML files with PHP, and then there's really no way that users can tell what you have up your sleeve. The best things in using PHP are that it is extremely simple for a newcomer, but offers many advanced features for a professional programmer.

Chapter 4

4 Design Document

4.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the TRAVEL GUIDE APPLICATION.. It illustrate the purpose and complete description for the development of the system. It explain system constraints, interface and interactions with other external applications. This document is primarily intended help the users to know more about the major attractions near to them.This section give an overview of our system, "9 to 5" APPLICATION. The application provides all the details about tourist places or major attractions like their opening time, closing time, history and other details. The app also provides all the details of the hotels available for the users. The app can also provide the best route to the places you have selected. The app will come in handy for those who travel a lot, or like to travel to new places.

4.2 Scope

The application gives all the details about the places you are looking for.It also provide place suggestions to visit which best suites for the preferences you have given.

4.3 Overview

The purpose of this document is to help the reader to visualize the solution to the project presented.This document verifies how the design meet the requirement stimulated in the SRS document through design viewpoints.The design viewpoints will cover all design elements presented before. This document will provide a direct approach to the development of this project hence reducing feature creep and ponitedly determine the quality of the design.

4.4 Data Design

Database are the storehouses of data used in the software systems.The data is stored in tables inside the database.Several tables are created for the manipulation of the data for the system.Two essential settings for a database are

- Primary Key- The field that is unique for all the record occurrence.
- Foreign Key- The field used to set relation between tables.

Normalization

- Normalization is the process of organizing the data in the database.
- Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate the undesirable characteristics like Insertion, Update and Deletion Anomalies.
- Normalization divides the larger table into the smaller table and links them using relationship.
- The normal form is used to reduce redundancy from the database table.

Chapter 5

5 Development of the System

The system is an web application which provides all the details related to the places you are looking for your next trip. We also provide a feature that gives you suggestions of the places near you to visit, based on the preferences you have provided.

Chapter 6

6 System Testing

Testing is the penultimate step of software development.An elaborate testing of the data is prepared and the system is using the test data,while doing testing,errors are noted and correction is made.The users are trained to operate the developed system.Both hardware and software securities are made to run the developed system successfully.System testing is aimed at ensuring the system works accurately before the live operation commences.

6.1 Test Plan

6.1.1 Scope

This test plan will cover the following testing activities as identified in the testing strategy.

- White Box Testing

White box testing is a testing technique, that examines the program structure and derives test data from the program logic/code. In white box testing, the UI is bypassed. Inputs and outputs of ratings are tested directly at the code level and the results are compared against specifications. It Reveals errors in "hidden" code.

- Black Box Testing

Black - box testing is a method of software testing that examines the functionality of an application based on the specifications. Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end - user would.

- Unit Testing

The module includes four main components to undergo unit testing. It checks the syntax error, logic error and validity of the program correctness. This test will be performed by the developers.

- Integration Testing

After all components pass the unit testing, they get to be tested whether they work correctly when they are running concurrently and communicating to each other. Integration tests exercise an entire subsystem and ensure that a set of components play nicely together.

- Internal Data Testing

We will test the validity of the data before it enters the database to avoid any problems that may face us in the database. We will test the encryption of the personal information of all the users along with the user names and passwords to ensure maximum security of the users privacy.

6.1.2 Software risk issues

In this section, the plan is to test the risk involved in critical issues such as: Difficult to run on visual studio due to large loading time. Some inherent software risks such as complexity exist; also these issues need to be identified. Proper network connection Working of Mysql database

6.1.3 Features to be tested

- Test whether correct user name and password allows you to login.
- Test whether invalid user name and password prevents you from login.
- Test whether there is any connection problem in the Server.
- Test whether the student and admin details are entered correctly.
- Test whether invalid data entry allows saving data successfully.
- Test whether all pages are loaded correctly.
- Test whether watermark is embedded and extracted properly.
- Test whether the provided security works correctly.

6.2 Test consolidation

6.2.1 Test item

The items or features to be tested in the test cases are included in the document. Each and every user input is tested. The present condition of the system is tested. It is checked to make sure that environment is ready for the application to work.

6.2.2 Input specifications

These are the inputs required to execute the test case. All required inputs including data elements and values are given below.

Data	Values
Email	A valid email ID
Password	Combination of characters and numbers
Phone number	10 Digit number

Chapter 7

7 System Implementation and Maintenance

7.1 Implementation

System implementation is the conversion of new system into an operating one which involves creating compatible files, training clients and installing hardware. User training is crucial for minimizing resistance to change and giving chance to prove its worth. Training aids user friendly manuals and healthy screens provide the user with a good start. Software maintenance follows conversion to the extent that changes are necessary to maintain satisfactory operations relative to changes in the user's environment. Maintenance often includes minor enhancements or corrections to the problem that surface late in the systems operations. In the implementation phase, the team builds the components either from scratch or by composition. Given the architecture document meant from the design phase and the requirement document from the analysis phase, the team should build exactly what has been requested, though there is still room for innovation and flexibility. For example, a component may be narrowly designed for this particular system, or the component may be made more general to satisfy a reusability

- Careful planning
- Investigation of system and constraints
- Design the methods to achieve changeover.
- Training the staff in the changed phase.
- Evaluation of change over method.
- The method of implementation and time scale to be adopted are found out initially.

7.2 Maintenance

This phase occurs as a result of deploying the whole system at the end users organization. They will perform the beta testing at the end users and inform to the developers about any needed modification to the application. The customer records all the problems that are encountered during the beta testing and reports these to the developer at regular intervals.

7.2.1 Corrective Maintenance

Even with the best quality assurance activities, it is likely that the customer will uncover defects in the software. Corrective maintenance changes the software to correct the defects. Corrective Maintenance activity may consist of repair, restoration or replacement of equipment. This activity will be result of a regular inspection, which identifies the failure in time for corrective maintenance to be planned and scheduled, then performed during a routine maintenance shutdown.

7.2.2 Adaptive Maintenance

Over time, the original environment(CPU, operating system, business rules, external product characteristics) for which the software was developed is likely to change. Adaptive maintenance results in modification to the software to accommodate changes to its external environment.

7.2.3 Enhanced Maintenance

As software is used, the customer/user will recognize additional functions that will provide the benefit. Perfect maintenance extends the software beyond its original functional requirements.

7.2.4 Preventive Maintenance

Computer software deteriorates due to change, and because of this preventive maintenance often called software re-engineering, must be conducted to enable the software to serve the needs of its end users. Preventive maintenance makes changes to computer programs so that they can be more easily corrected, adapted and enhanced.

Chapter 8

8 Conclusion and Future Scope

8.1 Conclusion

There are two types of users that interact with the system Admin, and the traveler.Each of these two types has different uses of the system so each of them has their own panel.Admin can manage all the features of application dynamically by login on admin panel. The users can view the major attractions and their history and other details available in an area. Users can also view all the information about the hotels available in a specific area.

8.2 Future Scope

- Gives all the details about the places you are looking for.
- Provide place suggestions to visit which best suites for the preferences you have given.

Appendix

A Data Flow Diagram

Data flow is the one of the best way of documenting the entire functionality of the system.For the system ,which will have data flows in and have some processing inside and then some data flow out from the system can be documented or represented effectively by means of data flow out from the system can be documented or represented effectively by means of data flow diagrams. The data flow diagram are a diagrammatic representation of the system,which has input,process and output.Once any system is represented using a data flow diagram we can identify the following things easily:

- Various entities interacting with the system are identified
- Flow of data from one entity to another is identified
- The various processes involved in between the interaction of two or more entities in the system are clearly pointed out
- The various data stores which hold the data in between the process,are clearly identified

Some Data Flow Diagram charting forms:

A.1 External source or receiver

A source or sink is a person or part of organization ,which enters or receives information from the system,but is considered to be outlining the contest of data flow model.

A.2 Transform process

A process represents transformation where incoming data flowsare changed into outgoing data flow

A.3 Data Store

A data store is repository of data that is to be stored for use by one or more process may be as simple as buffer or queue or sophisticated as relational database. They should have clear names.If a process merely uses the contest of store and does not alter it,the arrowhead goes only form the store to the

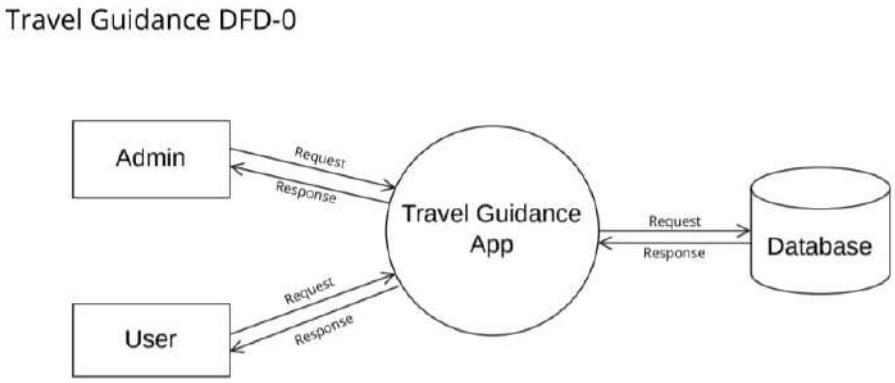
process. If a process alters the details in the store then double-headed arrow is used.

A.4 Data flow

A data flow is a route, which enable packets of data to travel from one point to another.Data may flow,with arrowhead pointing in the direction of the flow

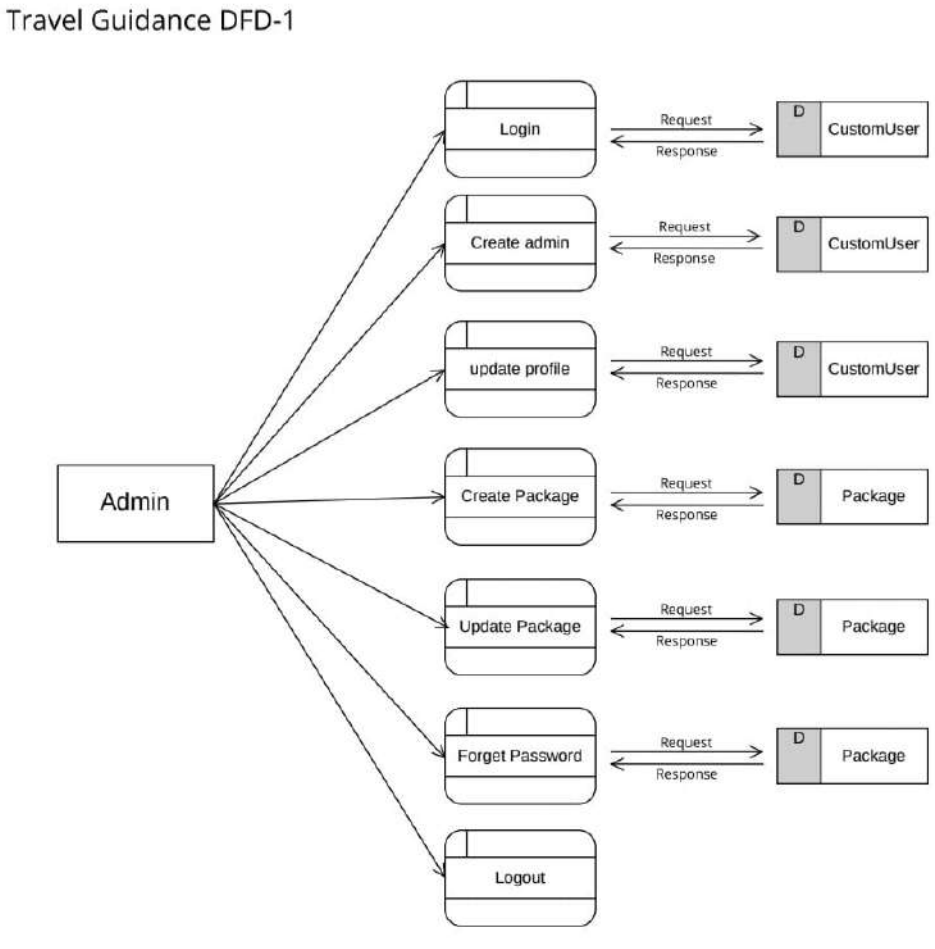
B dfd0

B.1 Level 0



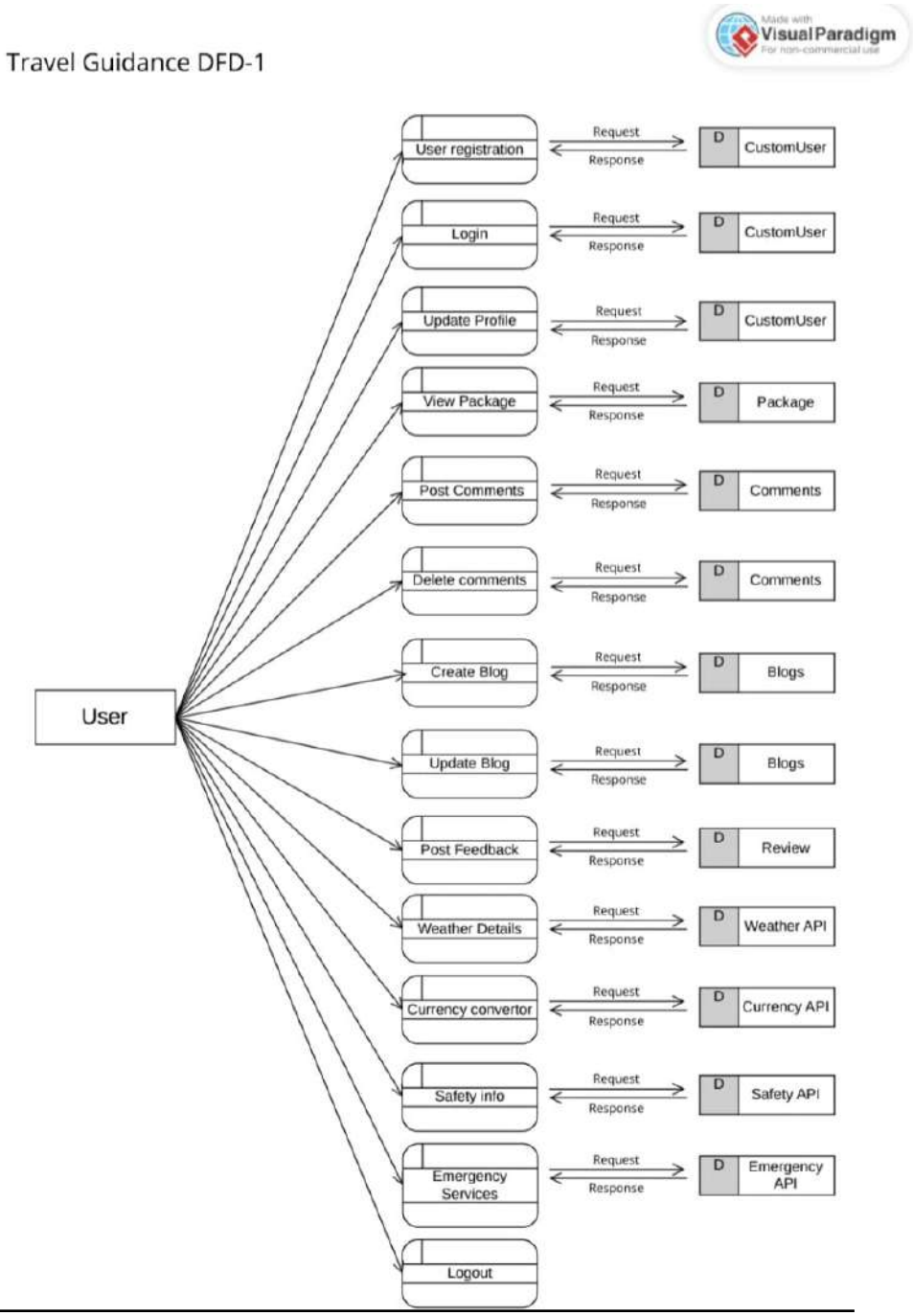
C Data Flow Diagram of Admin

C.1 Level 0

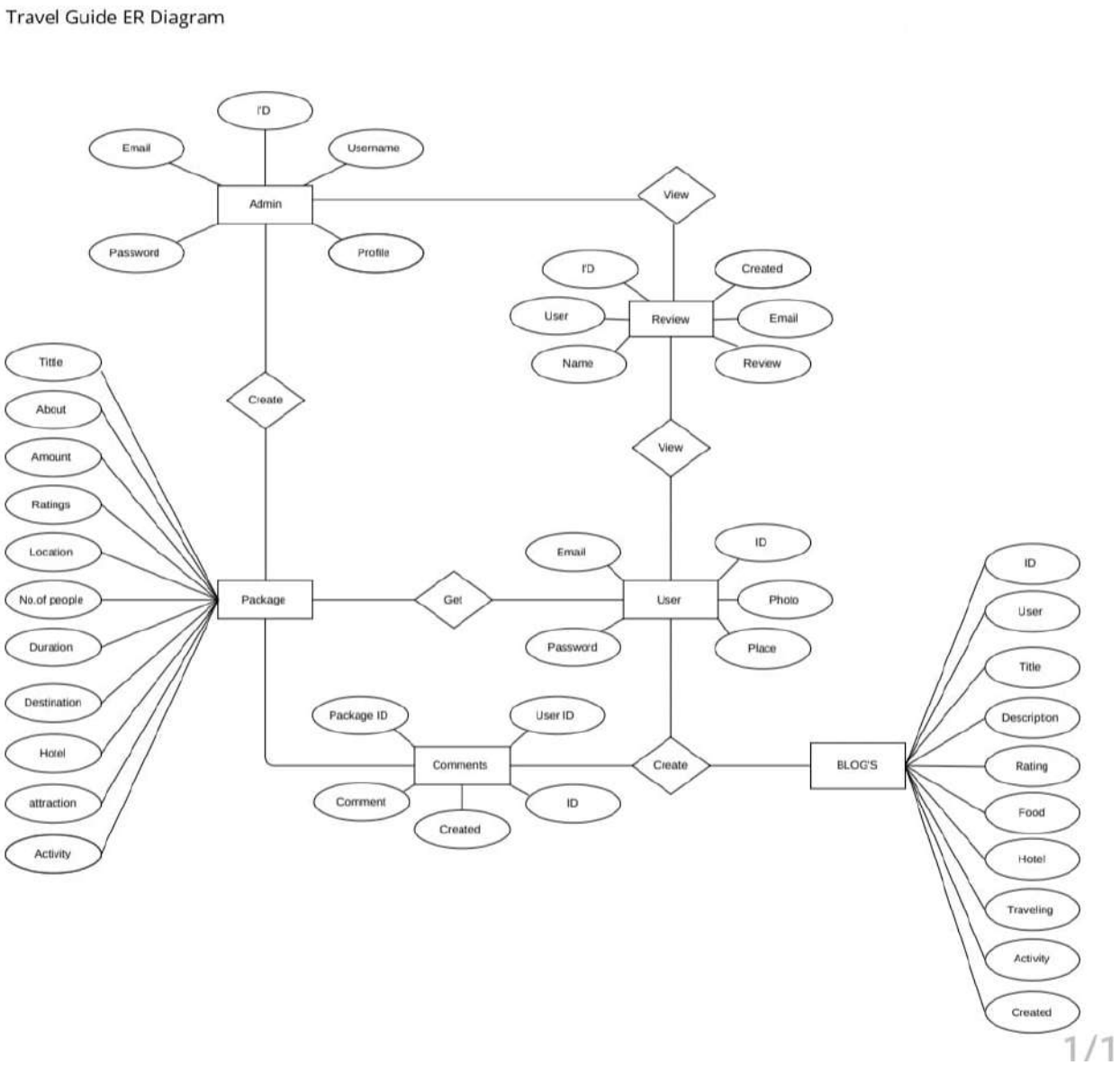


D Data Flow Diagram of User

D.1 Level 0

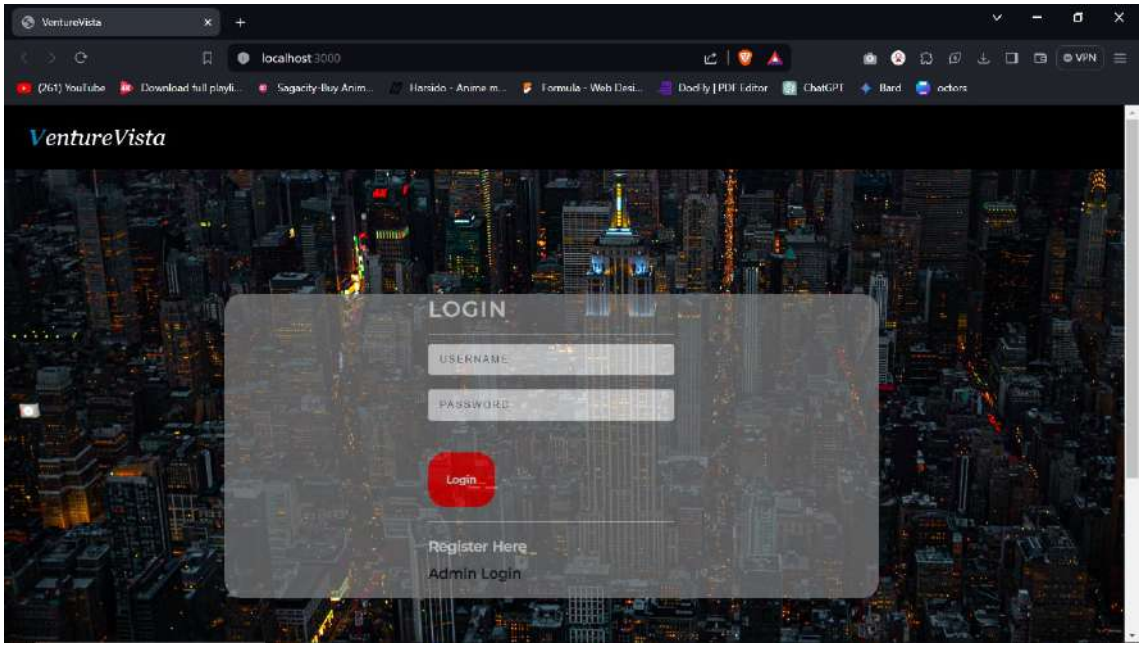


E E-R Diagram

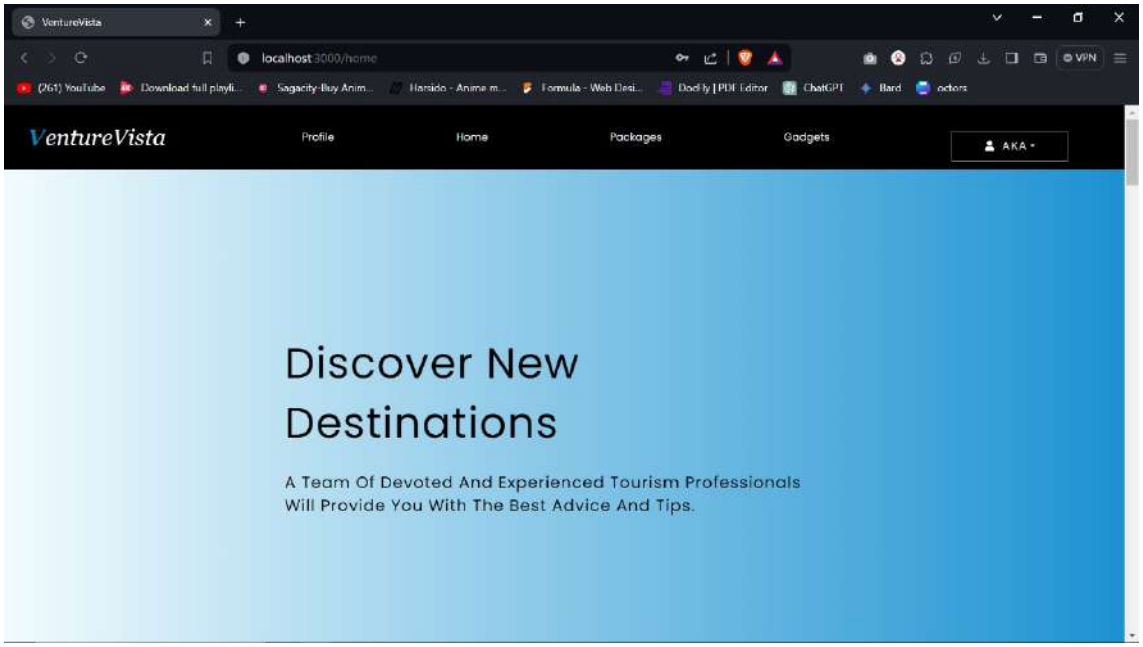

Travel Guide ER Diagram
I'D
Email
Username
Admin
Password
Profile
View
I'D
Created
User
Review
Email
Name
Review
Tittle
About
Amount
Ratings
Location
No.of people
Duration
Destination
Hotel
attraction
Activity
Create
View
Email
ID
Package
Get
User
Photo
Password
Place
Package ID
User ID
Comments
Create
BLOG'S
Comment
ID
Created
ID
User
Title
Description
Rating
Food
Hotel
Traveling
Activity
Created
1/1

F USER INTERFACES

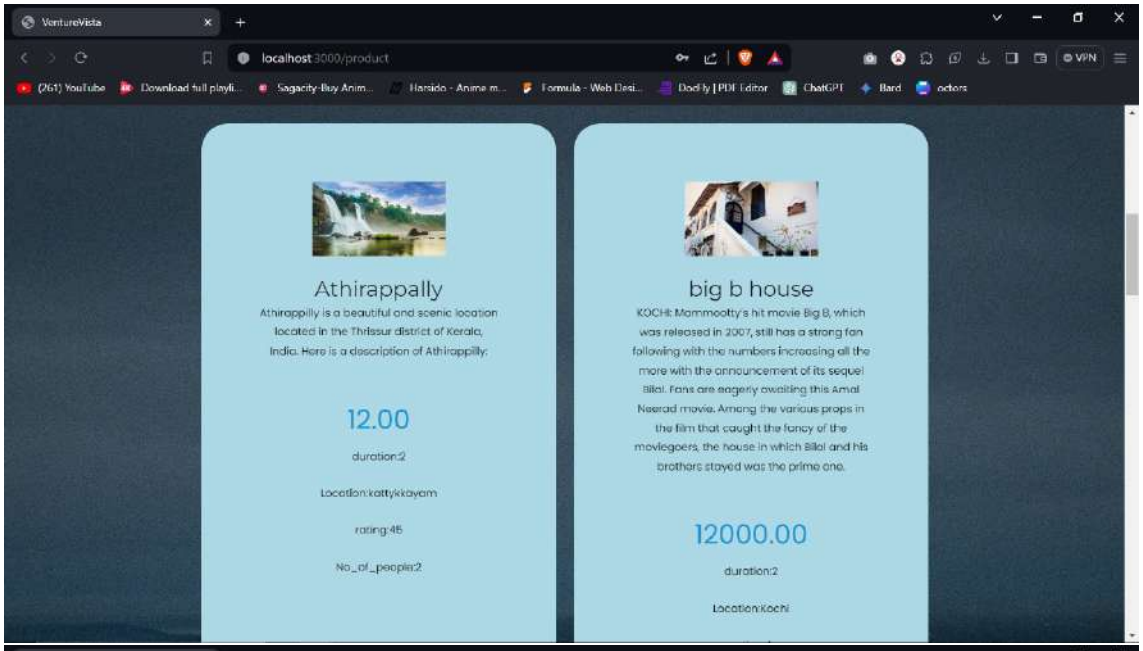
F.1 LOGIN

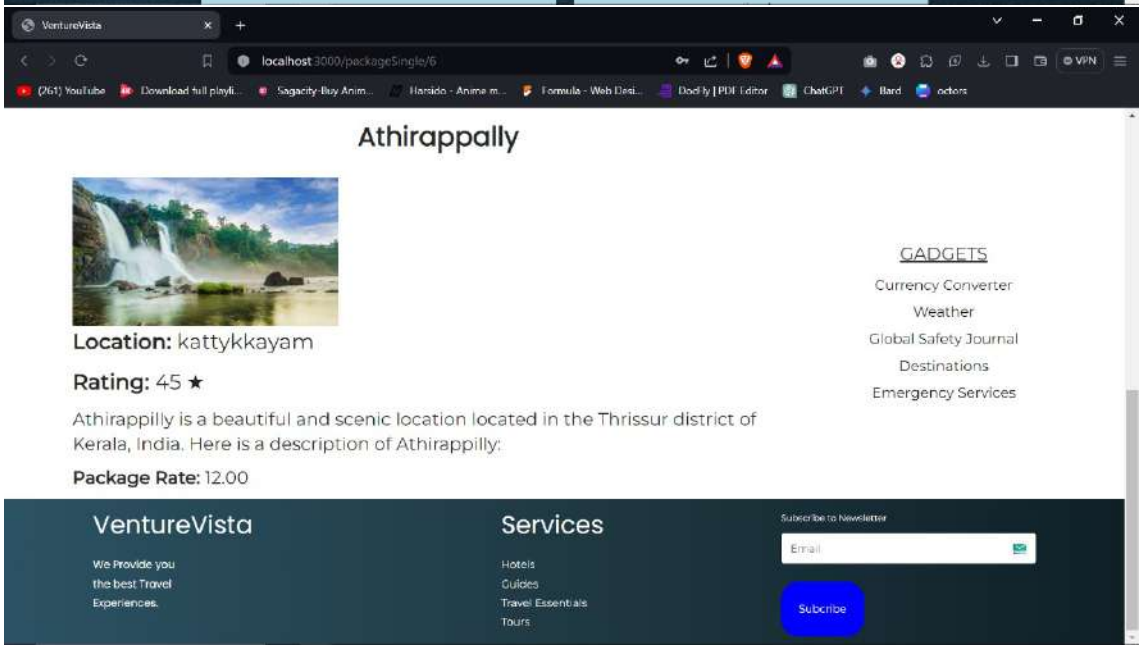


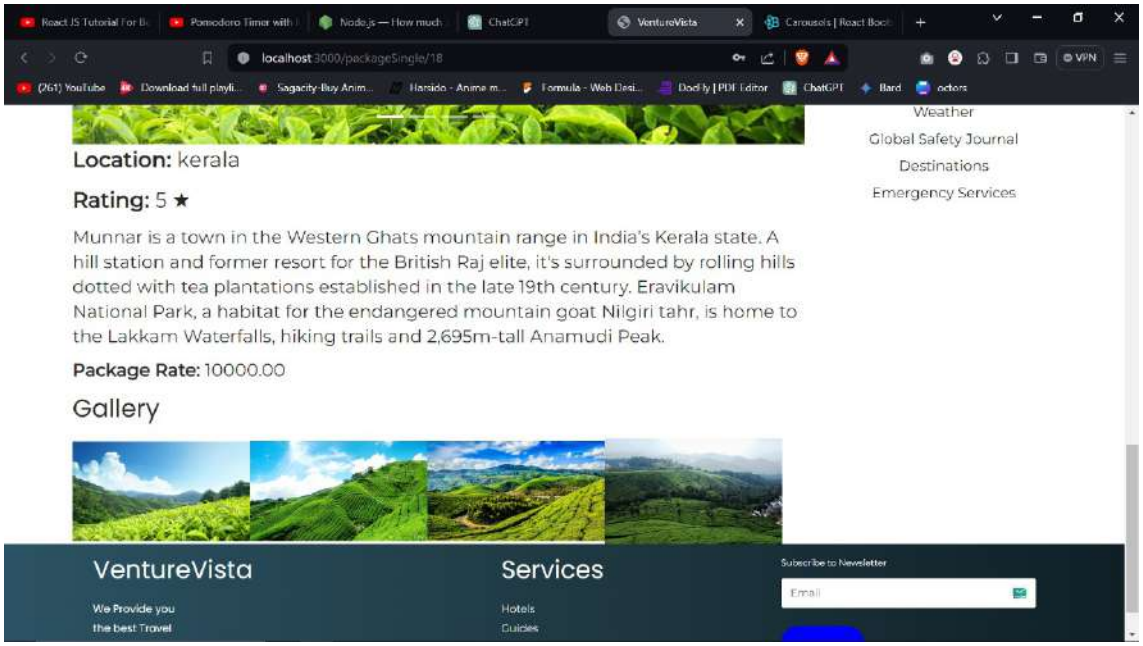
F.2 HOME


VentureVista
localhost:3000/home
VentureVista
Profile
Home
Packages
Gadgets
AKA
Discover New Destinations
A Team Of Devoted And Experienced Tourism Professionals
Will Provide You With The Best Advice And Tips.

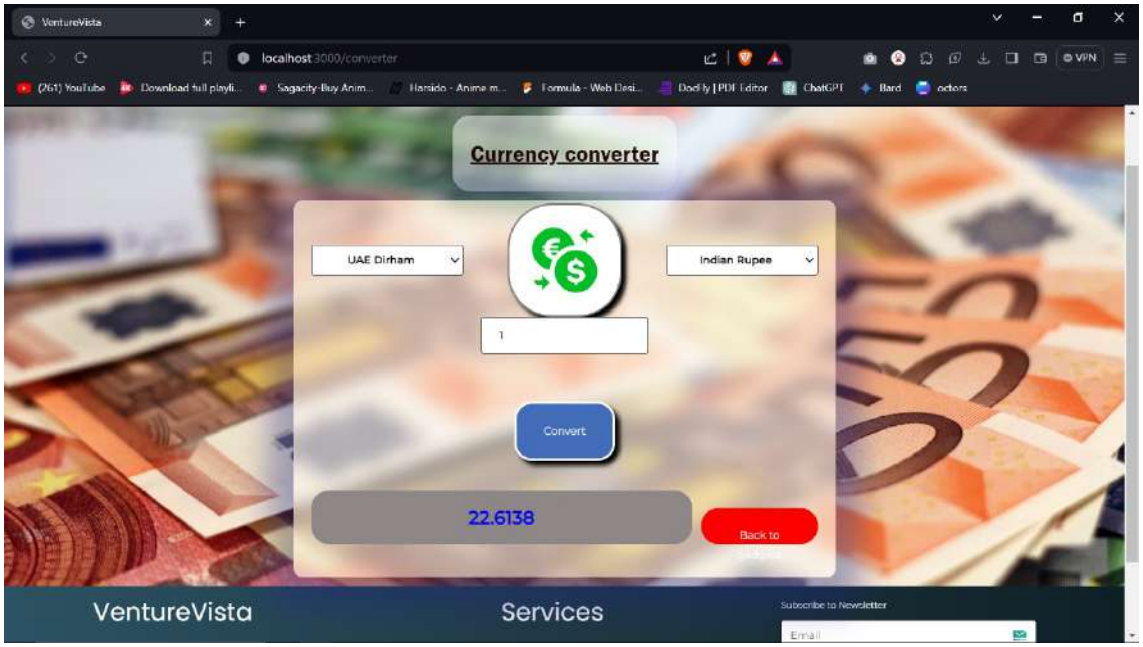
F.3 PACKAGES


VentureVista
localhost:3000/product
Athirappally
Athirappilly is a beautiful and scenic location located in the Thrissur district of Kerala, India. Here is a description of Athirappilly:
12.00
duration:2
Location:kattykkayam
rating:45
No_of_people:2
big b house
KOCHI: Mammootty's hit movie Big B, which was released in 2007, still has a strong fan following with the numbers increasing all the more with the announcement of its sequel Bilal. Fans are eagerly awaiting this Amal Neerad movie. Among the various props in the film that caught the fancy of the moviegoers, the house in which Bilal and his brothers stayed was the prime one.
12000.00
duration:2
Location:Kochi

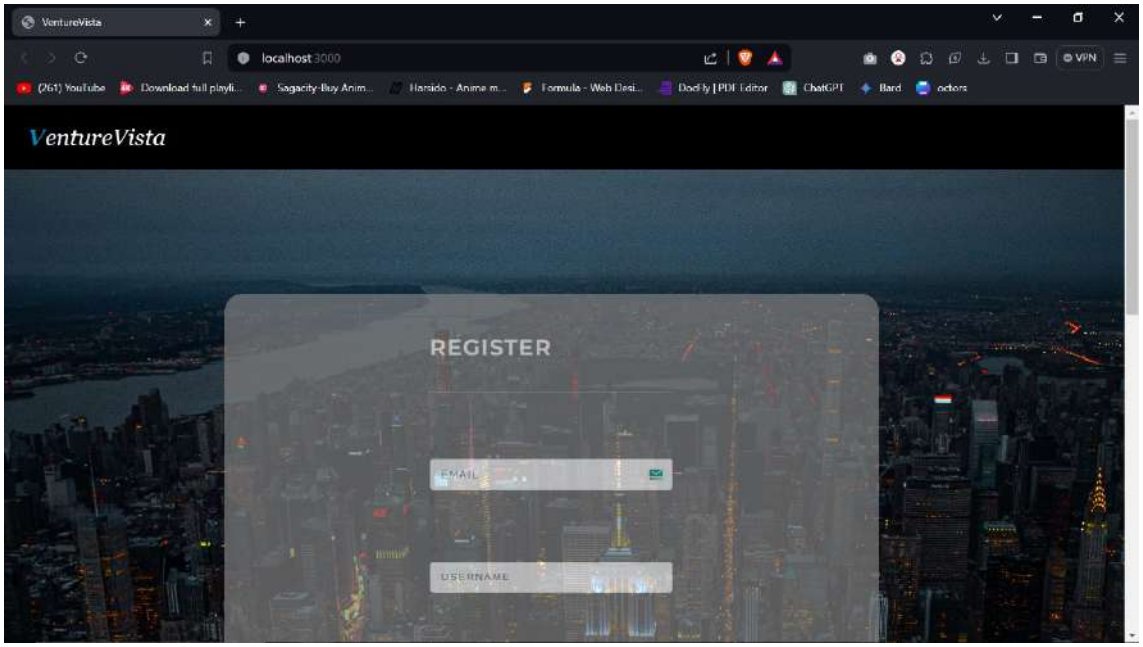

VentureVista
localhost:3000/packageSingle/6
Athirappally
Location: kattykkayam
Rating: 45 ★
Athirappilly is a beautiful and scenic location located in the Thrissur district of Kerala, India. Here is a description of Athirappilly:
Package Rate: 12.00
GADGETS
Currency Converter
Weather
Global Safety Journal
Destinations
Emergency Services
VentureVista
We Provide you the best Travel Experiences.
Services
Hotels
Guides
Travel Essentials
Tours
Subscribe to Newsletter
Email
Subcribe


localhost:3000/packageSingle/18
Weather
Global Safety Journal
Destinations
Emergency Services
Location: kerala
Rating: 5 ★
Munnar is a town in the Western Ghats mountain range in India's Kerala state. A hill station and former resort for the British Raj elite, it's surrounded by rolling hills dotted with tea plantations established in the late 19th century. Eravikulam National Park, a habitat for the endangered mountain goat Nilgiri tahr, is home to the Lakkam Waterfalls, hiking trails and 2,695m-tall Anamudi Peak.
Package Rate: 10000.00
Gallery
VentureVista
We Provide you the best Travel
Services
Hotels
Guides
Subscribe to Newsletter
Email

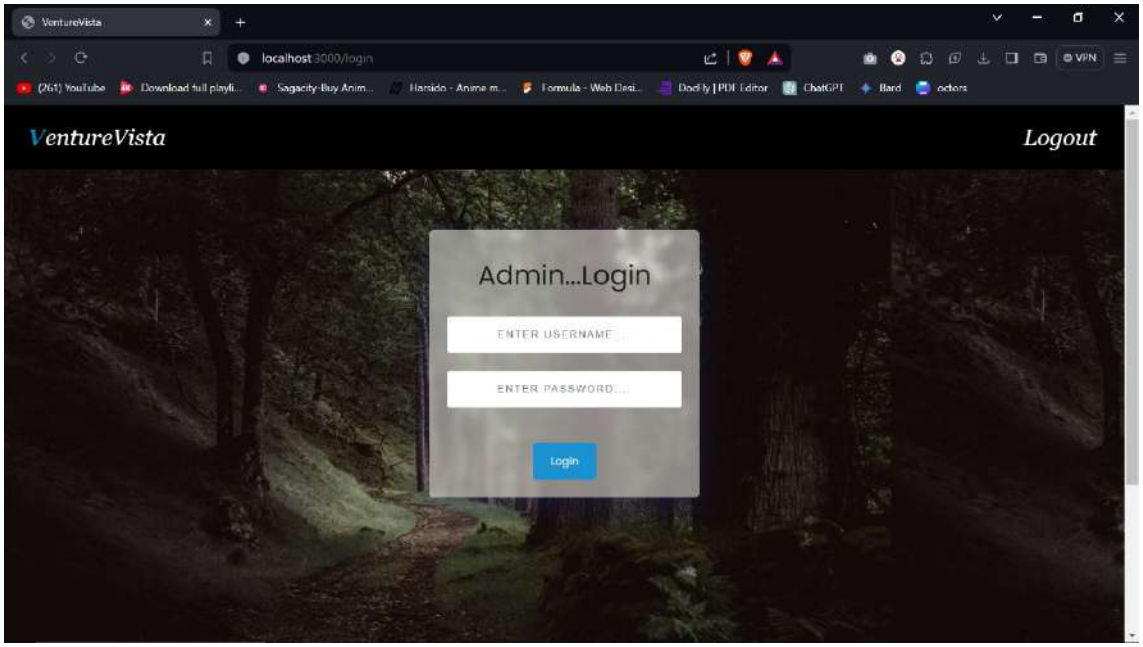
F.4 CURRENCY CONVERTER



F.5 REGISTER


VentureVista
localhost:3000
VentureVista
REGISTER
EMAIL
USERNAME

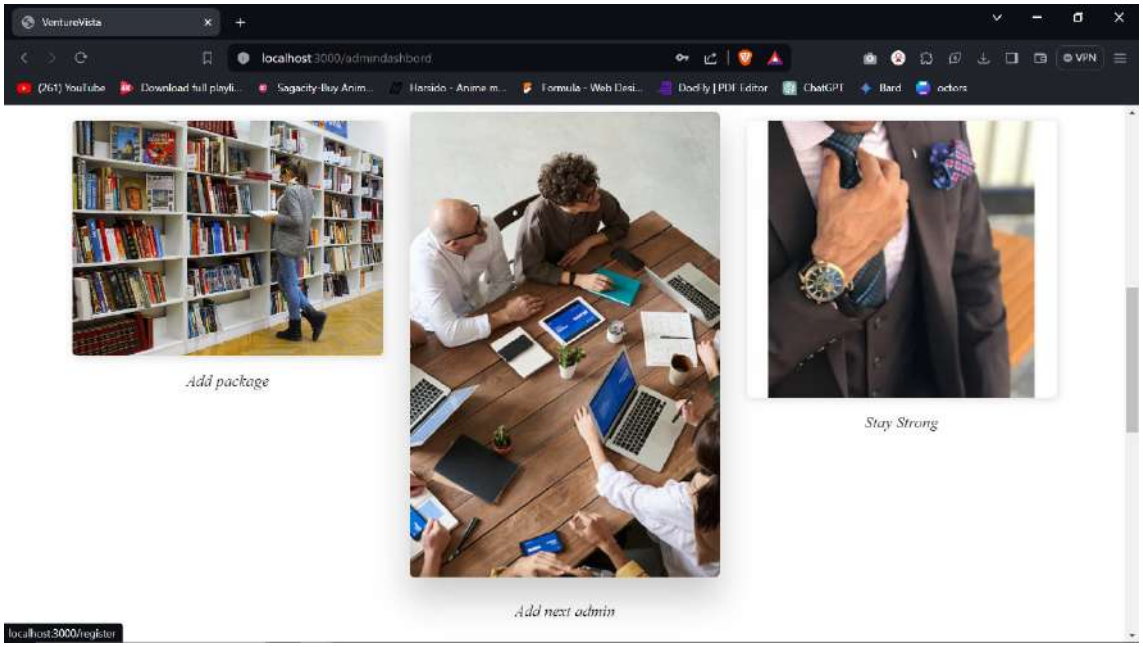
F.6 ADMIN LOGIN



F.7 ADMIN


VentureVista
localhost:3000/admindashboard
VentureVista
Logout
WELCOME MR.ADMIN
THE MAGICAL POWERES OF
YOU ARE....

F.8 ADMIN OPTIONS



G CODE

FRONTEND

```
//
// Source code recreated from a .class
file by
IntelliJ IDEA
// (powered by Fernflower decompiler)
//

import React, { useState, useEffect }
from
"react";
import { Modal, Button, Carousel } from
"react-bootstrap";
import { BrowserRouter, Route, Link }
from
"react-router-dom";
import Image from
"../images/athirappalli.jpeg";
import "./product.css";
function Product({ product }) {
const [show, setShow] = useState(false);
const handleClose = () =>
setShow(false);
const handleShow = () => setShow(true);  console.log(product)

  return (
    <>
      <div>
        <div className="product--card">
<img src={product.attraction}
alt="Product
Shoes" className="product--img" />
          <h4>{product.title}</h4>
          <p>
            {product.about}
          </p>
          <h2>{product.amount}</h2>
```

```
<p>duration:{product.duration}</p><p>Location:{product.location}
</p><p>rating:{product.rating}</p><p>
No_of_people:{product.no_of_people}</p>
        </div>
      </div>
    </>
  );
}

export default Product;
import React from 'react'
import './gadgets.css'
function AdminPowers() {
  return (
<body data-spy="scroll"
data-target=".onpage-navigation"
data-offset="60">
      <main>

<section class="bg-dark-30
showcase-page-header module parallax-bg"id='re' data-background="">
          <div class="titan-caption">
<div class="caption-content"><div class="font-alt mb-30
titan-title-size-1">Our best gadgets to
assits
you</div>
<div class="font-alt mb-40
titan-title-size-4">feel free to rush
trough
them</div>
            </div>
          </div>
        </section>
<div class="main showcase-page"><section class="module-medium"
id="demos"> <div class="container">
<div class="row multi-columns-row"><div
class="col-md-4 col-sm-6 col-xs-12"><a
class="content-box" href="/converter">
<div class="content-box-image"><img
src="https://blog.cdn.cmarix.com/blog/wp-content
```

```
/uploads/2023/07/develop-a-currency-converter-mobile-app.webp"
alt="Currency Converter" /></div>
<h3 class="content-box-title
font-serif">Currency
Converter</h3></a></div>
<div class="col-md-4 col-sm-6
col-xs-12"><a
class="content-box" href="Wheather">
<div class="content-box-image"><img
src="https://play-lh.googleusercontent.com/znc7FD1B
qyFR92b8n6hyROZrAR3FsTyV_ThO79hyuQG-Nb45z2qHFDTnf1HXH50DYg"
alt="Agency" /></div>
<h3 class="content-box-title
font-serif">wheather app</h3></a></div>
<div class="col-md-4 col-sm-6
col-xs-12"><a
class="content-box" href="/journal">
<div class="content-box-image"><img
src="https://s.yimg.com/ny/api/res/1.2/rx7Bjb7IghNsT35t13jufQ--
/YXBwaWQ9aGlnaGxhbmRlcjt3PTY0MDtoPTQ4MA--
/https://s.yimg.com/os/creatr-images/2020-05/45815870-9238-11ea-8fff-
1180915abec1"
alt="Portfolio" /></div>
<h3 class="content-box-title
font-serif">Global Safety
Journal</h3></a></div>
<div class="col-md-4 col-sm-6
col-xs-12"><a
class="content-box"
href="/destinations">
<div class="content-box-image"><img
src="https://images.pexels.com/photos/7235910/pexels-photo-
7235910.jpeg?auto=compress&cs=tinysrgb&w=300"
alt="Restaurant" /></div>
<h3 class="content-box-title
font-serif">Destinations</h3></a></div>
<div class="col-md-4 col-sm-6
col-xs-12"><a
class="content-box" href="/emergency">
<div class="content-box-image"><img
```

```
src="https://cdn.dribbble.com/userupload/7319839/
file/original-86f171450ede48437f8b1cfd8ac352f1.png?
crop=0x0-2880x2160&resize=400x300&vertical=center"
alt="Finance" /></div>
<h3 class="content-box-title
font-serif">Emergency
services</h3></a></div>
{/* <div class="col-md-4 col-sm-6
col-xs-12"><a class="content-box"
href="/product">
<div class="content-box-image"><img
src="https://cdn.dribbble.com/userupload/10920096/file/

original-4b166c456ae22ae068a4c961d6053a74.png?resize=400x0"
alt="Landing"/></div>
<h3 class="content-box-title
font-serif">Travel
packages</h3></a></div> */}
              </div>
            </div>
          </section>
        </div>
<div class="scroll-up"><a
href="#totop"><i
class="fa
fa-angle-double-up"></i></a></div>
      </main>
<script
src="assets/lib/jquery/dist/jquery.js"></script>
<script
src="assets/lib/bootstrap/dist/js/bootstrap.min.js"></script>
<script
src="assets/lib/wow/dist/wow.js"></script>
<script
src="assets/lib/jquery.mb.ytplayer/dist/jquery.mb.YTPlayer.js"></script>
<script
src="assets/lib/isotope/dist/isotope.pkgd.js"></script>
<script
src="assets/lib/imagesloaded/imagesloaded.pkgd.js"></script>
<script
```

```
src="assets/lib/flexslider/jquery.flexslider.js"></script>
<script
src="assets/lib/owl.carousel/dist/owl.carousel.min.js"></script>
<script
src="assets/lib/smoothscroll.js"></script>
<script
src="assets/lib/magnific-popup/dist/jquery.magnific-popup.js"></script>
<script
src="assets/lib/simple-text-rotator/jquery.simple-text-rotator.min.js"></script>
<script
src="assets/js/plugins.js"></script><script
src="assets/js/main.js"></script>
</body>

  )
}

export default AdminPowers
import React, { useState, useEffect }
from
"react";
import { useNavigate } from
"react-router-dom";
import axios from "axios";
import Loader from
"../components/Loader";
import Error from "../components/Error";import Success from
"../components/Success";
import "./login.css";
import { Link } from "react-router-dom";// import Loginimg from
"../images/Login.svg";function
LoginScreen() {

const [FormData, setFormData] =
useState({

  });
const navigate=useNavigate()
const [loading, setLoading] =
useState(false); const [error, setError]
```

```
= useState("");
const [success, setSuccess] =
useState("");
const handlechange=(e)=>{
  setFormData({
    ...FormData,
    [e.target.name]:e.target.value
  })
}
const Login=async(e)=>{
const {username,password}=FormData

await axios.post('admin-login/',{
  username,
  password
}).then((res)=>{
  if(res.status===200){
    console.log(res.status)
    if(res.status==200){
localStorage.setItem('adminkey',res.data.access)
localStorage.setItem('isadmin',res.data.is_superuser)
alert("admin login
successfull..welcome..."+res.data.username)
navigate('/admindashbord')
    }
    else{
alert("detected unautherized entry...")    }
    setLoading(false);
  }

}).catch(()=>{
alert("detected unauthorized entry..")  })
}

  return (
    <div className="login--screen">
{/* <img src={Loginimg} alt="login img"
className="login--img" /> */}
      {loading && <Loader></Loader>}
```

```
<div className="row
justify-content-center"> <div
className="col-md-5 mt-5">
{error.length > 0 && <Error
msg={error}></Error>}
          <div className="bs">
            <h2>Admin...Login</h2>

            <input
              type="text"
              className="form-control"
              name="username"
placeholder="Enter username...."
onChange={handlechange}
              style={{
                textAlign: "center",
                padding: "20px",
                marginBottom: "20px",
              }}
            />
            <input
              type="password"
              className="form-control"
placeholder="enter Password...."
name="password"
              onChange={handlechange}
              style={{
                textAlign: "center",
                padding: "20px",
                marginBottom: "20px",
              }}
            />
            {loading ? (
<div>Login...Please Wait...</div> ) : (
<button className="login--btn"
onClick={Login}>
                Login
              </button>
            )}
          </div>
```

```
        </div>
      </div>
    </div>
  );
}

export default LoginScreen;
```

BACKEND

```
//
// Source code recreated from a .class
file by
IntelliJ IDEA
// (powered by Fernflower decompiler)
//
from django.db import models

# Create your models here.


from django.contrib.auth.models import
AbstractUser
from django.db import models

class CustomUser(AbstractUser):

phone = models.CharField(max_length=15,
blank=True)
place = models.CharField(max_length=255,blank=True)
nickname =
models.CharField(max_length=30,
blank=True)
color = models.CharField(max_length=30,
blank=True)
image =
models.ImageField(upload_to='user_images/',
blank=True, null=True)
```

```
    def _str_(self):
        return self.username



class Package(models.Model):
admin = models.ForeignKey(CustomUser,
on_delete=models.CASCADE)
title = models.CharField(max_length=255)    about = models.TextField()
amount =
models.DecimalField(max_digits=10,
decimal_places=2)
    rating = models.IntegerField()
location =
models.CharField(max_length=255)duration
=
models.CharField(max_length=50)no_of_people
= models.CharField(max_length=50)hotel =
models.ImageField(upload_to='package_photos/',
blank=True, null=True)
destination =
models.ImageField(upload_to='package_photos/',
blank=True, null=True)
activity =
models.ImageField(upload_to='package_photos/',
blank=True, null=True)
attraction =
models.ImageField(upload_to='package_photos/',
blank=True, null=True)

    def _str_(self):
        return self.title


class Comments(models.Model):
user = models.ForeignKey(CustomUser,
on_delete=models.CASCADE)
package = models.ForeignKey(Package,
on_delete=models.CASCADE)
```

```
    comment = models.TextField()
created =
models.DateField(auto_now_add=True)
    def _str_(self):
        return self.comment


class Blogs(models.Model):
user = models.ForeignKey(CustomUser,
on_delete=models.CASCADE)
title = models.CharField(max_length=255)    description = models.TextField()
    rating = models.IntegerField()
food =
models.ImageField(upload_to='blog_photos/',
blank=True, null=True)
hotel =
models.ImageField(upload_to='blog_photos/',
blank=True, null=True)
travelling =
models.ImageField(upload_to='blog_photos/',
blank=True, null=True)
activity =
models.ImageField(upload_to='blog_photos/',
blank=True, null=True)
created =
models.DateField(auto_now_add=True)
    def _str_(self):
        return self.title

class Review(models.Model):

name =
models.CharField(max_length=255,blank=True,null=True)
email=models.CharField(max_length=255,blank=True,null=True)
    review = models.TextField()
created =
models.DateField(auto_now_add=True)
    def _str_(self):
        return self.review
"""
```

```
URL configuration for travel_guide
project.

The urlpatterns list routes URLs to
views. For
more information please see:
https://docs.djangoproject.com/en/5.0/topics/http/urls/
Examples:
Function views
1. Add an import: from my_app import
views2. Add a URL to urlpatterns:
path('',
views.home, name='home')
Class-based views
1. Add an import: from other_app.views
import
Home
2. Add a URL to urlpatterns: path('',
Home.as_view(), name='home')
Including another URLconf
1. Import the include() function: from
django.urls import include, path
2. Add a URL to urlpatterns:
path('blog/',
include('blog.urls'))
"""
from django.contrib import admin
from django.urls import path
from . import views
from rest_framework_simplejwt import
views as
jwt_views

urlpatterns = [
path('',views.UserRegistrationView.as_view(),
name='user-registration'),
path('admin-register/',
views.SuperuserRegistrationView.as_view(),
name='superuser-registration'),
path('user-login/',
```

```
views.UserloginView.as_view(),
name='user-token_obtain_pair'),
path('admin-login/',
views.AdminloginView.as_view(),
name='admin-token_obtain_pair'),
path('api/token/refresh/',
jwt_views.TokenRefreshView.as_view(),
name='token_refresh'),
path('user-details/',
views.UserDetailsView.as_view(),
name='user-details'),
path('update-user/',
views.UpdateUserDetailsView.as_view(),
name='update-user-details'),

path('send-otp/',
views.PasswordResetOTPSendView.as_view(),
name='update-user-details'),
path('otp-validation/',
views.OTPValidationView.as_view(),
name='otp-validation'),
path('change-password/',
views.ChangePasswordView.as_view(),
name='new-password'),

path('package-crud/',
views.PackageCRUDView.as_view()),
path('package-crud/<int:pk>/',
views.PackageCRUDView.as_view(),),

path('list-packages/',
views.ListPackagesView.as_view(),
name='list-packages'),
path('package-detail/<int:pk>/',
views.PackageDetailView.as_view(),
name='package-detail-view'),

path('create-comment/<int:pk>/',
views.CreateCommentView.as_view(),
name='create-comment'),
```

```
path('list-comments/<int:pk>/',
views.ListCommentsView.as_view(),
name='list-comments'),
path('delete-comment/<int:pk>/',
views.DeleteCommentView.as_view(),
name='delete-comment'),


path('create-blog/',
views.CreateBlogView.as_view(),
name='create-blog'),
path('list-blogs/',
views.ListBlogsView.as_view(),
name='list-blogs'),
path('blog-detail/<int:pk>/',
views.BlogDetailView.as_view(),
name='blog-detail'),

path('list-user-blogs/',
views.ListUserBlogsView.as_view(),
name='list-user-blogs'),
path('update-blog/<int:pk>/',
views.UpdateBlogView.as_view(),
name='update-blog'),
path('delete-blog/<int:pk>/',
views.DeleteBlogView.as_view(),
name='delete-blog'),


path('create-review/',
views.CreatReviewView.as_view(),),
path('list-reviews/',
views.ListReviewView.as_view(),),
path('review-detail/<int:pk>/',
views.ReviewDetailView.as_view(),),


]
from rest_framework import serializers
from .models import
```

```
CustomUser,Package,Comments,Blogs,Reviewfrom
rest_framework_simplejwt.serializers
import TokenObtainPairSerializer
from django.core.exceptions import
ObjectDoesNotExist


class
UserTokenObtainPairSerializer(TokenObtainPairSerializer):
    @classmethod
    def get_token(cls, user):
        token = super().get_token(user)
        # adding custom claims
token['username'] = user.usernametoken['is_superuser'] =
user.is_superuser return token

    def validate(self, attrs):
        data = super().validate(attrs)
        user = self.user
        if user.is_superuser==False:
data["is_superuser"] = user.is_superuser
data["username"] = user.username
            return data
        else:
raise serializers.ValidationError("Only
common
users are allowed to log in here.")


class
AdminTokenObtainPairSerializer(TokenObtainPairSerializer):
    @classmethod
    def get_token(cls, user):
        token = super().get_token(user)
        # adding custom claims
token['username'] = user.usernametoken['is_superuser'] =
user.is_superuser return token

    def validate(self, attrs):
        data = super().validate(attrs)
```

```
        user = self.user
        if user.is_superuser==True:
data["is_superuser"] =
user.is_superuserdata["username"] =
user.username return data
        else:
raise serializers.ValidationError(" Onlyadministrators are allowed to log in
here.")



class
CustomUserSerializer(serializers.ModelSerializer):
password_confirmation =
serializers.CharField(write_only=True)
email =
serializers.EmailField(required=True)
image=serializers.ImageField(max_length=None,use_url=True,required=False)
    class Meta:
        model = CustomUser
fields = ('id', 'username', 'email',
'phone',
'place', 'image',
'password','password_confirmation')
extra_kwargs = {'password':
{'write_only':
True}}

    def validate(self, data):
        try:
CustomUser.objects.get(email=data['email'])raise
serializers.ValidationError({'email':'Email
already exists.'})
        except ObjectDoesNotExist:
            pass

if data['password'] !=
data['password_confirmation']:
raise
serializers.ValidationError({'password_confirmation':"Passwords
```

```
do not match."})
        return data

    def create(self, validated_data):
validated_data.pop('password_confirmation',
None)
user =
CustomUser.objects.create_user(**validated_data)
        return user

class
CustomUserupdateSerializer(serializers.ModelSerializer):
image=serializers.ImageField(max_length=None,use_url=True,required=False)
    class Meta:
        model = CustomUser
fields =
['username','image','phone','place','email']

    def validate_email(self, value):
if
CustomUser.objects.filter(email__iexact=value)
.exclude(pk=self.instance.pk).exists():
raise serializers.ValidationError("This
email
address is already in use.")
        return value


class
PackageSerializer(serializers.ModelSerializer):
    class Meta:
        model = Package
        exclude = ['admin']

class
PackagelistSerializer(serializers.ModelSerializer):
    class Meta:
        model = Package
        fields = "_all_"
```

```
class
CommentSerializer(serializers.ModelSerializer):
    class Meta:
        model = Comments
        fields = ['comment']

class
CommentlistSerializer(serializers.ModelSerializer):
    class Meta:
        model = Comments
        fields = "_all_"

class
BlogSerializer(serializers.ModelSerializer):
    class Meta:
        model = Blogs
        exclude = ['user']

class
BloglistSerializer(serializers.ModelSerializer):
    class Meta:
        model = Blogs
        fields = "_all_"


class
ReviewSerializer(serializers.ModelSerializer):
    class Meta:
        model = Review
        fields = "_all_"

class
ReviewlistSerializer(serializers.ModelSerializer):
    class Meta:
        model = Review
        fields = "_all_"
from django.contrib import admin

# Register your models here
from .models import
```

```
CustomUser,Package,Comments,Blogs,Review# admin.site.register(CustomUser)
class
CustomUseryadmin(admin.ModelAdmin):
list_display =
['username','is_superuser']admin.site.register(CustomUser,CustomUseryadmin)

class Packageadmin(admin.ModelAdmin):
list_display =
['admin','title','amount','location','duration']
admin.site.register(Package,Packageadmin)

class Commentsadmin(admin.ModelAdmin):
list_display =
['user','comment','package']admin.site.register(Comments,Commentsadmin)

class Blogsadmin(admin.ModelAdmin):
    list_display = ['user','title']
admin.site.register(Blogs,Blogsadmin)



class Reviewadmin(admin.ModelAdmin):
    list_display = ['name','review']
admin.site.register(Review,Reviewadmin)
```

FOGGY ROAD ALERT SYSTEM

PROJECT REPORT

Submitted By

SREEJISHNA K

Reg. No. CCAVBCA014

For the award of the Degree of

Bachelor of Computer Application (BCA)

(University of Calicut)

under the guidance of

Ms. Sharon Joy C

Assistant Professor

BACHELOR OF COMPUTER APPLICATION
DEPARTMENT OF COMPUTER SCIENCE
CHRIST COLLEGE(AUTONOMOUS)
IRINJALAKUDA, KERALA
INDIA

2021-2024

DEPARTMENT OF COMPUTER SCIENCE

CHRIST COLLEGE (AUTONOMOUS)

IRINJALAKUDA

CERTIFICATE

This is to certify that the project report entitled ***"Foggy Road Alert System"*** *is a bonafide record of the project work done by* ***Sreejishna K*** *in partial fulfillment of the requirement for the sixth semester of* ***Bachelor of Computer Application*** *in Department of Computer Science of* ***CHRIST COLLEGE (AUTONOMOUS) IRINJALAKUDA***

.

Ms. Sharon Joy C
Assistant Professor,CS
Internal Guide

Ms. Sini Thomas
Head of the Department
Computer Science

EXTERNAL EXAMINER **INTERNAL EXAMINER**

DECLARATION

We hereby declare that this project work **"FOGGY ROAD ALERT SYSTEM"** submitted to Christ College (Autonomous) Irinjalakuda, affiliated to Calicut University in partial fulfillment of the requirement for the award of the Bachelor of Computer Application, is a record of original work done by us, under the guidance of Ms. SHARON JOY C, Department of Computer Science.

Place: Irinjalakuda SREEJISHNA K

ACKNOWLEDGEMENT

First and foremost we like to thank Lord almighty for his providence and for being the guiding light throughout the project. We wish to express my sincere gratitude to our beloved Department head for giving me all the facilities for our project. We take this opportunity to express my gratitude to the class teacher Ms. SOUMYA P.S and head of the department Ms. SINI THOMAS who has been supported us throughout the course of this project. We are thankful for her aspiring guidance and valuable advice during the project work. We express my sincere thanks to my project guide Ms. SHARON JOY C for supporting and guiding throughout the project.We would take this opportunity to specially thank all other faculty members for their constant and continuous motivation. Finally we would like to thank my family and friends for giving valuable advice and moral support throughout our project.

ABSTRACT

FOGGY ROAD ALERT SYSTEM is an innovative application of computer vision and deep learning technologies, aims to significantly enhance road safety by providing real-time visibility information during adverse weather conditions. Leveraging Python, OpenCV, TensorFlow, and Keras, the system employs a Convolutional Neural Network (CNN)-based fog detection model trained on a diverse dataset to analyse images captured by strategically placed cameras. Real-time monitoring ensures prompt responses to changing fog conditions, and an alerting mechanism notifies authorities when intensity thresholds are exceeded. The user-friendly interface allows for visualizing real-time fog data and historical trends. The system's successful deployment in fog-prone areas demonstrates its potential to reduce accidents and improve traffic management. Recommendations for future enhancements include exploring advanced sensors and dynamic threshold adjustments, further solidifying its role in proactive road safety measures. Additionally, the system seamlessly integrates with Django, offering features such as real-time fog identification, continuous monitoring, user authentication, adaptability for model retraining, and scalable deployment. The comprehensive documentation ensures ease of use, fostering continuous improvement through advancements in deep learning techniques and user feedback. This integration harmonizes cutting-edge capabilities and an intuitive web application, effectively addressing the challenges posed by foggy environmental conditions.

Contents

Chapter 1

1 Introduction

Fog is a major cause of road accidents worldwide. To improve road safety and traffic management, we propose a foggy road alert system that uses a web application, a fog detection model, and an email notification system. The web application allows users to upload videos of foggy roads and view the results of fog detection and alert generation. The fog detection model uses deep learning to classify each video frame as foggy or non-foggy, and calculate the average fog level. The email notification system sends an alert to the nearest traffic police station if the fog level is high.

1.1 Overview

The FOGGY ROAD ALERT SYSTEM aims to develop a web application that warns authorities about foggy roads using video analysis and email notification. The web application allows users to upload videos of foggy roads and view the fog detection results. The fog detection model uses deep learning to classify each video frame as foggy or non-foggy, and calculate the average fog level. The email notification system sends an alert to the nearest traffic police station if the fog level is high.It is a User friendly application.

Chapter 2

2 System Analysis

2.1 Purpose

The main purpose of this application owned by Krishnapriya K ,Aljo Poulose and Sreejishna K is to warn authorities about foggy roads using video analysis and email notification to prevent accidents and road blocks.

2.1.1 Existing System

The existing systems are using sensors placed on the roads.These sensors detect the fog and generates the warning messages to authorities.Limitationss of this sytems are sensosrs are costly,requires periodic maintenance,less accuracy

2.1.2 Proposed System

The proposed system is a web application which allows users to upload videos of foggy roads and view the fog detection results. The fog detection model uses deep learning to classify each video frame as foggy or non-foggy, and calculate the average fog level. The email notification system sends an alert to the nearest traffic police station's email address if the fog level is high. The proposed system aims to improve road safety and traffic management by providing timely and accurate information about the fog conditions on the roads.

2.2 Problem definition

To understand the problem and the needs before developing it, we are designing a foggy road alert system to detect fog on roads and generate warning messages for the traffic police.

2.3 FEASIBILITY STUDY

After the problem is clearly understood and solutions proposed, the next step is to conduct the feasibility study. Feasibility study is defined as evaluation or analysis of the potential impact of the proposed project or program. The objective is to determine whether the proposed system is feasible. There are three aspects of feasibility study to which the proposed system is subjected as discussed below.

2.3.1 Technical Feasibility

Technical feasibility assess whether the current technical resources are sufficient for the new system. We can upgrade the level of technology for supporting our

website. We check whether the proposed system can be implemented in the present system without supporting the existing hardware.

2.3.2 Economical Feasibility

Economic feasibility determines whether the time and money are available to develop the system. There is no additional hardware used to develop the site, it is inexpensive to build.

2.3.3 Operational Feasibility

Operational feasibility determines if the human resources are available to operate the system once it has been installed. No extra training is needed to use this system. Anyone who has knowledge in internet in English language can easily use the system. The resources that are required to implement or install are already available with the organization.

Chapter 3

3 Software Requirement Specification

3.1 Purpose

The purpose of this document is to outline the software requirements for the Foggy Road Alert System. It provides a comprehensive overview of the functionalities, constraints, and interactions necessary for the development of the system. This document serves as a blueprint for designing and implementing a robust solution aimed at enhancing road safety during foggy conditions. It delineates the system's capabilities to detect fog, generate timely alerts to traffic authorities.This document aims to facilitate the creation of an efficient and effective foggy road alert system that mitigates risks and improves overall road safety.

3.2 Scope

The Foggy Road Alert System aims to detect foggy conditions in real-time and provide timely alerts to drivers and authorities, enhancing road safety during low visibility conditions.

3.3 Overall Description

This section provides an overview of our Foggy Road Alert System. The project entails detecting the fog percentage in uploaded videos, triggering a warning email to the nearest traffic police authorities if the percentage surpasses a predefined threshold. Upon receiving the alert, authorities can take necessary measures to manage traffic on foggy roads, thereby reducing accidents associated with fog in various locations.The main aim of our project is to ensure the safety of people travelling through these foggy roads.

3.3.1 Product Perspective

The Foggy Road Alert System operates within the broader context of road safety and transportation management systems. From a product perspective, it serves as a critical component in enhancing road safety by specifically addressing the challenges posed by foggy conditions.The system is a standalone applocation that integrates CNN-based image processing for fog detection.It interacts with users through a Django based web interface

3.3.2 Product Functionality

The Foggy Road Alert System is equipped with several key functionalities aimed at bolstering road safety during foggy conditions. Leveraging sophisticated algorithms, the system automates the process of fog detection in uploaded videos,

ensuring prompt alerts when hazardous conditions arise. Robust access controls are implemented to guarantee that only authorized users can access the fog detection features, thereby enhancing system security. Powered by Django, the system's web application provides users with an intuitive interface, facilitating seamless interaction and ease of use. Additionally, the system delivers real-time alerts to users and traffic authorities, enabling swift responses to hazardous foggy conditions and ultimately enhancing overall road safety.

3.3.3 Users and Characteristics

The primary users of the system are traffic police authorities responsible for monitoring road conditions and managing traffic flow. They utilize the system to receive timely fog alerts and coordinate responses to ensure road safety.Secondary users include drivers who rely on the system's alerts and recommendations to navigate safely during foggy conditions.The characteristics include technical proficiency,time sensitivty,reliability and accesibilty.Then there are administators who oversee the overall functioning and maintenance of the system.Administrators can monitor and manage the generation and dissemination of fog alerts.

3.4 Specific Requirements

3.4.1 Hardware Requirements

- System: IBM-Compatible PC
- Processor: Intel Core i3
- Speed: Above 1GHz
- RAM capacity: 4 GB
- Hard Dsk drive: 500 GB

3.4.2 Software Requirements

- Front End : HTML,CSS,Bootstrap
- Back End : Python,Django
- IDE : PyCharm Community Edition 2023.1.3

3.5 Functional Requirements

It contains three main modules.

- 1.Fog Detection
- 2.Alert Generation
- 3.User Interface

Fog Detection

This part detects fog in videos or camera feeds.It uses algorithms to analyze images and determine fog density. This module is responsible for the core functionality of detecting foggy conditions.

Alert Generation

This module handles the generation and dissemination of fog alerts to traffic authorities.Develop algorithms for generating fog alerts based on detected fog density and predefined thresholds.

User Interface

This module encompasses the web interface and user interactions with the Foggy Road Alert System.Design and develop user-friendly interfaces for interacting with fog detection features and alerts.

3.6 Non Functional Requirements

Non-functional requirements define the overall qualities or attributes of the resulting system. Non-functional system place restrictions on the product being developed, the developed process, and specify external constraints that the product must meet. Examples of non-functional requirements include safety, security, usability, reliability and performance requirements. Project management issues (costs, time and schedule) are often considered as non-functional requirements. The principal non - functional constraints which are relevant to critical systems :

- performance
- security
- safety
- usability

Performance

Performance requirements concern the speed of operation of a system. Types of performance requirements :

- Response requirements (how quickly the system reacts to a user input).
- Throughput requirements (how much the system can accomplish within a specified amount of time).
- Availability requirements (is the system available for service when requested by end users). The speed of operation of this system is adequate for the requirements.

Reliability

- Reliability is the ability of a system to perform its required function under stated conditions for a specified period of time.
- constraints on the runtime behavior of the system. This system is reliable because its functionalities can be done on the required conditions.

Safety

- Safety requirements are not required which exclude unsafe situation from the possible solution of the system.

Usability

Usability is the ease with which a user can learn to operate, prepare inputs for,and interpret outputs of system or components. Usability requirements include :

- information error messages.
- well-formed user interfaces.

3.7 Interface Requirements

3.7.1 Hardware interfaces

The system must run over the internet, all the hardware shall require to connect internet will be hardware interface for the system. As for example modem, WAN-LAN.

3.7.2 Software interfaces

Software interface required for the working of the project is the appropriate operating system.

3.7.3 Communication interfaces

The user of the site communicate using the network connectivity and the data set is accessed

3.8 Security Requirements

- Validation of input is handled.
- This application containing the computer systems is physically secured against arms or surreptitious entry by intruders.
- Users must be authorized carefully to reduce changes of any such user giving access to an intruder in exchange for a bribe or other favour.

3.9 Platform Used

The platform used for developing and running the project is Windows 11, the latest version of Microsoft's operating system. Windows 11 offers a redesigned and refreshed user interface, new tools and features, and improved security and performance. Windows 11 supports various web development frameworks and languages, such as HTML, CSS, JavaScript, PHP, Python, and more. Windows 11 also enables the use of machine learning models, such as the one used for fog detection, through libraries and tools like TensorFlow, PyTorch, OpenCV, and others. Windows 11 is compatible with most devices that meet the minimum system requirements.

3.10 Technologies Used

HTML

HTML stands for HyperText Markup Language. It is the standard markup language for creating web pages. HTML uses elements, which are defined by tags, to structure and label the content of a web page. HTML elements can have attributes, which provide additional information or functionality. HTML can also include other technologies, such as CSS and JavaScript, to enhance the appearance and behavior of web pages.

CSS

CSS stands for Cascading Style Sheets. It is a language that allows you to style and layout web pages by applying rules to HTML elements. CSS can control the appearance, position, size, color, font, animation, and more of the web content. CSS can also adapt to different devices, screen sizes, and orientations. CSS is one of the core technologies of the web, along with HTML and JavaScript.

PYTHON

Python is a versatile, high-level programming language known for its simplicity and readability. It offers dynamic typing and automatic memory management, making it suitable for various domains such as web development, data analysis, artificial intelligence, and automation. Python's extensive standard library provides pre-built modules and packages for tasks ranging from file manipulation to network programming. Its interpreted nature allows for rapid development and easy debugging. Python supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Its popularity is fueled by a vibrant community, extensive documentation, and numerous third-party libraries. Overall, Python's simplicity, flexibility, and robust ecosystem make it an ideal choice for beginners and seasoned developers alike.

DJANGO

Django is a high-level Python web framework that simplifies web development by emphasizing reusability, rapid development, and the principle of DRY (Don't Repeat Yourself). It follows the Model-View-Controller (MVC) architectural pattern, with models defining data structures, views handling user interface logic, and templates rendering HTML. Django's built-in features include an ORM (Object-Relational Mapping) for interacting with databases, a secure authentication system, URL routing, and a powerful admin interface for managing content. Its scalability and extensibility are further enhanced by a vast ecosystem of third-party packages. Django's emphasis on convention over configuration and its adherence to best practices make it a popular choice for building robust and maintainable web applications.

CNN

Convolutional Neural Networks (CNNs) are a class of deep neural networks primarily used for image recognition and computer vision tasks. They consist of multiple layers, including convolutional layers, pooling layers, and fully connected layers. CNNs excel at learning hierarchical representations of features from input data. Convolutional layers apply filters to input images, extracting features such as edges and textures. Pooling layers reduce spatial dimensions while preserving important features. Fully connected layers combine extracted features for classification or regression tasks. CNNs leverage parameter sharing and local connectivity to efficiently learn patterns from large datasets. Their success in image recognition tasks has made them a cornerstone in various fields, including autonomous driving, medical imaging, and facial recognition.

Chapter 4

4 Design Document

4.1 Purpose

The purpose of the foggy road alert system project is to address the safety concerns posed by reduced visibility due to foggy conditions on roads. By leveraging a combination of web application interface and a trained convolutional neural network (CNN) model, the system aims to detect and quantify the level of fog in video frames. Through real-time analysis, it provides timely alerts to the nearest traffic police stations via email when the fog level exceeds a predefined threshold. This proactive approach intends to enhance road safety by enabling authorities to take precautionary measures promptly, such as issuing advisories, implementing speed restrictions, or deploying resources to manage traffic effectively, thereby reducing the likelihood of accidents and ensuring the well-being of commuters.

4.2 Scope

The project aims to create a web application using a trained CNN model to detect fog levels in uploaded videos, issuing alerts to traffic police stations for hazardous conditions.

4.3 Overview

The purpose of this document is to help the reader to visualize the solution to the project presented.This document verifies how the design meet the requirement stimulated in the SRS document through design viewpoints.The design viewpoints will cover all design elements presented before. This document will provide a direct approach to the development of this project hence reducing feature creep and ponitedly determine the quality of the design.

4.4 Data Design

Database are the storehouses of data used in the software systems.The data is stored in tables inside the database.Several tables are created for the manipulation of the data for the system.Two essential settings for a database are

- Primary Key- The field that is unique for all the record occurrence.
- Foreign Key- The field used to set relation between tables.

Normalization

- Normalization is the process of organizing the data in the database.

- Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate the undesirable characteristics like Insertion, Update and Deletion Anomalies.
- Normalization divides the larger table into the smaller table and links them using relationship.
- The normal form is used to reduce redundancy from the database table.

Chapter 5

5 Development of the System

The development phase of the Foggy Road Alert System using CNN with Django involves the practical implementation of the proposed project. It encompasses several key steps and components that contribute to the seamless integration of Convolutional Neural Networks (CNN) for fire detection with the Django web framework

Chapter 6

6 System Testing

Testing is the penultimate step of software development.An elaborate testing of the data is prepared and the system is using the test data,while doing testing,errors are noted and correction is made.The users are trained to operate the developed system.Both hardware and software securities are made to run the developed system successfully.System testing is aimed at ensuring the system works accurately before the live operation commences.

6.1 Test Plan

6.1.1 Scope

This test plan will cover the following testing activities as identified in the testing strategy.

- White Box Testing

White box testing is a testing technique, that examines the program structure and derives test data from the program logic/code. In white box testing, the UI is bypassed. Inputs and outputs of ratings are tested directly at the code level and the results are compared against specifications. It Reveals errors in "hidden" code.

- Black Box Testing

Black - box testing is a method of software testing that examines the functionality of an application based on the specifications. Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end - user would.

- Unit Testing

The module includes four main components to undergo unit testing. It checks the syntax error, logic error and validity of the program correctness. This test will be performed by the developers.

- Integration Testing

After all components pass the unit testing, they get to be tested whether they work correctly when they are running concurrently and communicating to each other. Integration tests exercise an entire subsystem and ensure that a set of components play nicely together.

- Internal Data Testing

We will test the validity of the data before it enters the database to avoid any problems that may face us in the database. We will test the encryption of the personal information of all the users along with the user names and passwords to ensure maximum security of the users privacy.

6.1.2 Software risk issues

In this section, the plan is to test the risk involved in critical issues such as: Difficult to run on visual studio due to large loading time. Some inherent software risks such as complexity exist; also these issues need to be identified. Proper network connection Working of Mysql database

6.1.3 Features to be tested

- Test whether correct user name and password allows you to login.
- Test whether invalid user name and password prevents you from login.
- Test whether there is any connection problem in the Server.
- Test whether the student and admin details are entered correctly.
- Test whether invalid data entry allows saving data successfully.
- Test whether all pages are loaded correctly.
- Test whether watermark is embedded and extracted properly.
- Test whether the provided security works correctly.

6.2 Test consolidation

6.2.1 Test item

The items or features to be tested in the test cases are included in the document. Each and every user input is tested. The present condition of the system is tested. It is checked to make sure that environment is ready for the application to work.

6.2.2 Input specifications

The input specifications for testing the Foggy Road Alert System invlove varoius type of data to assess system's perfomance comprehensively.The main user input to be tested is uploaded videos.The model trained using CNN also will be tested

Chapter 7

7 System Implementation and Maintenance

7.1 Implementation

System implementation is the conversion of new system into an operating one which involves creating compatible files, training clients and installing hardware. User training is crucial for minimizing resistance to change and giving chance to prove its worth. Training aids user friendly manuals and healthy screens provide the user with a good start. Software maintenance follows conversion to the extent that changes are necessary to maintain satisfactory operations relative to changes in the user's environment. Maintenance often includes minor enhancements or corrections to the problem that surface late in the systems operations. In the implementation phase, the team builds the components either from scratch or by composition. Given the architecture document meant from the design phase and the requirement document from the analysis phase, the team should build exactly what has been requested, though there is still room for innovation and flexibility. For example, a component may be narrowly designed for this particular system, or the component may be made more general to satisfy a reusability

- Careful planning
- Investigation of system and constraints
- Design the methods to achieve changeover.
- Training the staff in the changed phase.
- Evaluation of change over method.
- The method of implementation and time scale to be adopted are found out initially.

7.2 Maintenance

This phase occurs as a result of deploying the whole system at the end users organization. They will perform the beta testing at the end users and inform to the developers about any needed modification to the application. The customer records all the problems that are encountered during the beta testing and reports these to the developer at regular intervals.

7.2.1 Corrective Maintenance

Even with the best quality assurance activities, it is likely that the customer will uncover defects in the software. Corrective maintenance changes the software to correct the defects. Corrective Maintenance activity may consist of repair,

restoration or replacement of equipment. This activity will be result of a regular inspection, which identifies the failure in time for corrective maintenance to be planned and scheduled, then performed during a routine maintenance shutdown.

7.2.2 Adaptive Maintenance

Over time, the original environment(CPU, operating system, business rules, external product characteristics) for which the software was developed is likely to change. Adaptive maintenance results in modification to the software to accommodate changes to its external environment.

7.2.3 Enhanced Maintenance

As software is used, the customer/user will recognize additional functions that will provide the benefit. Perfect maintenance extends the software beyond its original functional requirements.

7.2.4 Preventive Maintenance

Computer software deteriorates due to change, and because of this preventive maintenance often called software re-engineering, must be conducted to enable the software to serve the needs of its end users. Preventive maintenance makes changes to computer programs so that they can be more easily corrected, adapted and enhanced.

Chapter 8

8 Conclusion and Future Scope

8.1 Conclusion

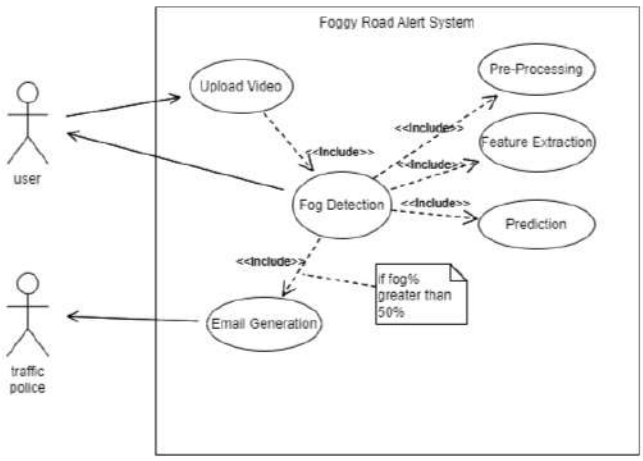
In conclusion, the Foggy Road Alert System serves two primary stakeholders: administrators and traffic authorities. Administrators utilize the dynamic admin panel to manage system features effectively, ensuring smooth operation and optimal performance. Traffic authorities play a critical role in receiving timely alerts generated by the system, enabling them to implement necessary traffic control measures and ensure road safety during foggy conditions. By providing administrators and traffic authorities with essential tools and insights, the Foggy Road Alert System contributes to enhanced road safety and improved traffic management practices.

8.2 Future Scope

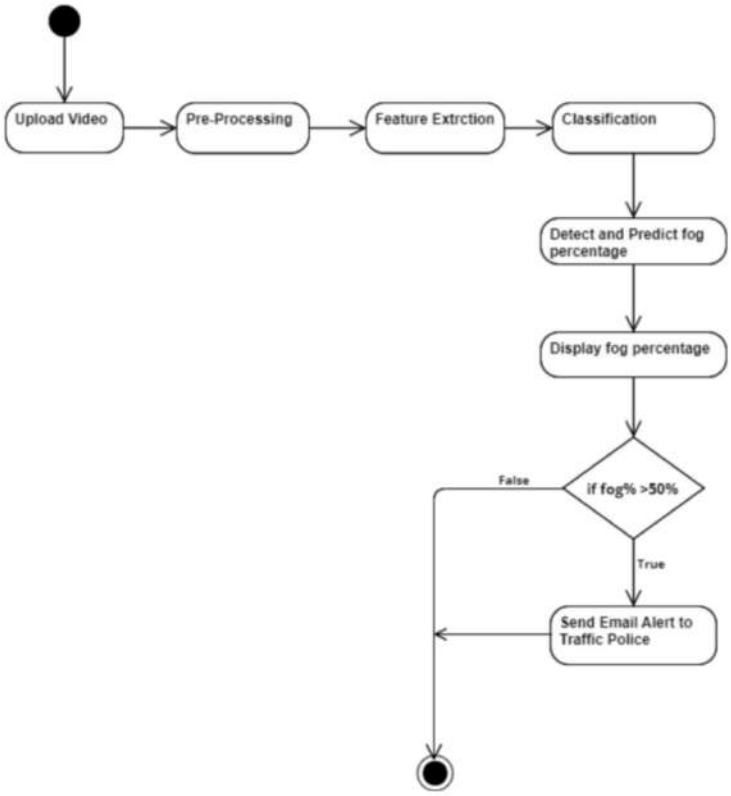
- Implement predictive analytics for proactive measures.
- Collaborate with navigation systems for safer routes.
- Engage users with crowdsourced fog reports.
- Ensure scalable and reliable system architecture.

Appendix

A Usecase Diagram

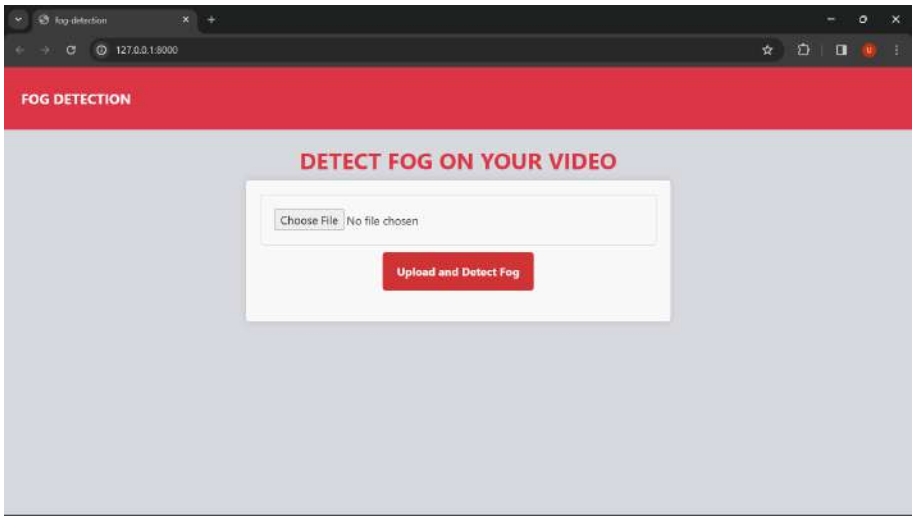


B Activity Diagram

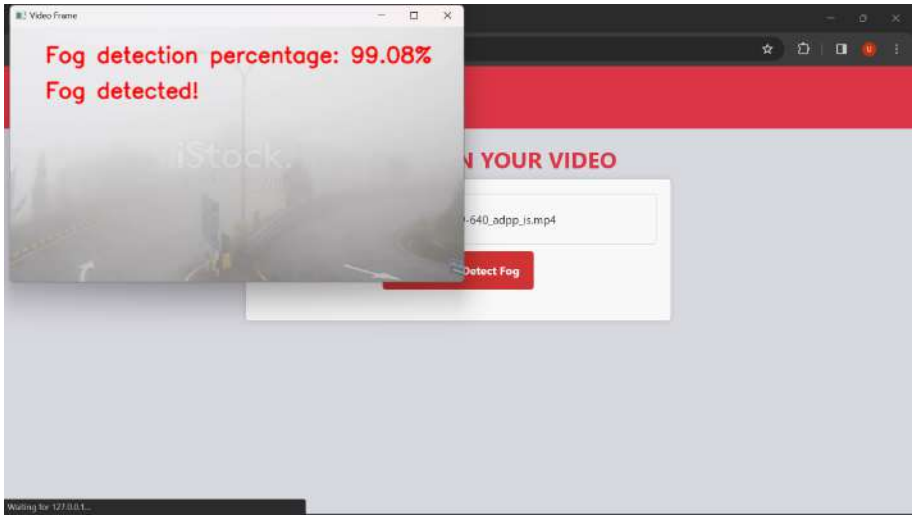


C USER INTERFACES

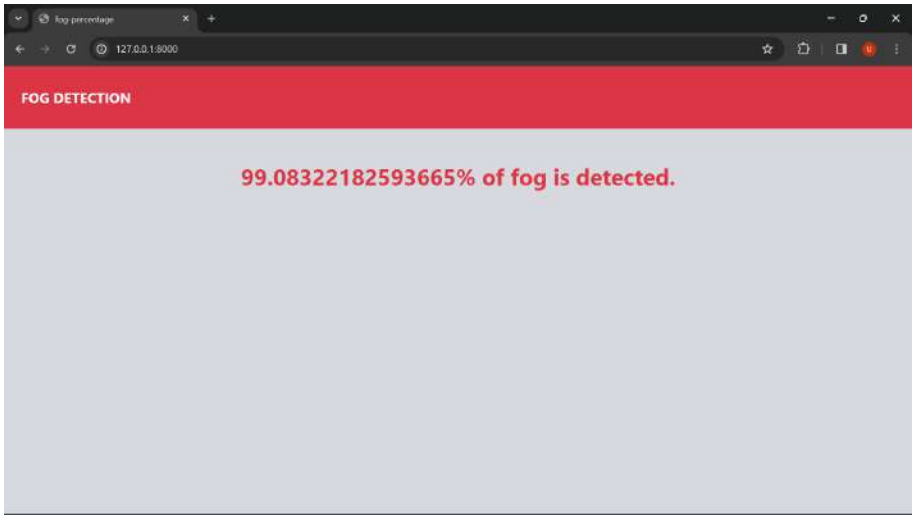
C.1



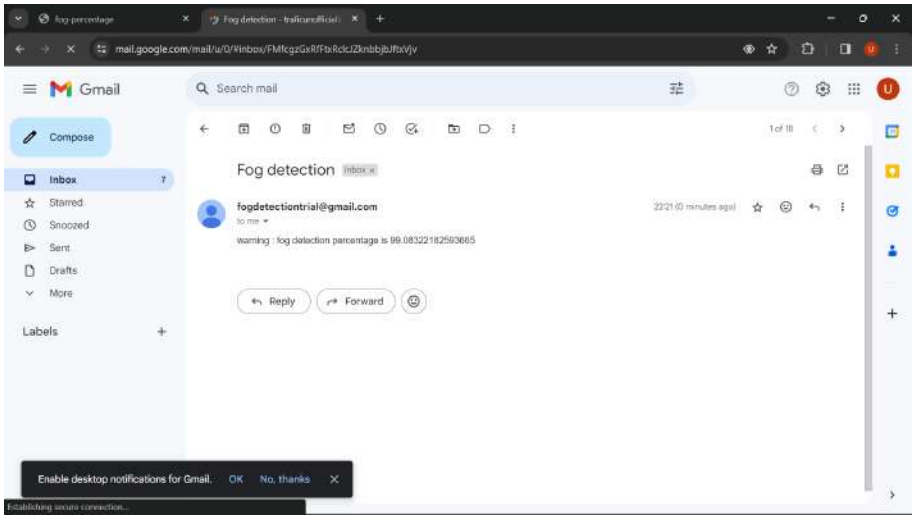
C.2 FOG DETECTION



C.3 FOG PERCENTAGE CALCULATION



C.4 ALERT


Gmail
Search mail
Compose
Inbox
Starred
Snoozed
Sent
Drafts
More
Labels
Fog detection
fogdetectiontrial@gmail.com
warning : fog detection percentage is 99.08322182593665
Reply
Forward
Enable desktop notifications for Gmail.
OK
No, thanks

D CODE

views.py

```
from django.shortcuts import render
import cv2
import numpy as np
from django.core.files.storage import FileSystemStorage
import os
from django.core.mail import send_mail
from keras.models import load_model

def home(request):
    if request.method=="POST" and 'video_file' in request
        .FILES:
        videofile =request.FILES['video_file']
        file_storage=FileSystemStorage()
        filename=file_storage.save(videofile.name,
            videofile)

        video_path =os.path.join(file_storage.location,
            filename)
        if not os.path.exists(video_path):
            error_message = "Video file does not exist."
            return render(request, 'detection_error.html
                ', {'error': error_message})

        cap= cv2.VideoCapture(video_path)
        import tensorflow as tf
        print("TensorFlow version:", tf.__version__)

        model = load_model('C:/Users/USER/OneDrive/
            Desktop/pfog/fog_detection-1/fog_detection/
            fog_detection_project/keras_model.h5')

        total_fog_percetage = []

        if not cap.isOpened():
            error_message="Error opening video file."
            return render(request,'detection_error.html
                ',{'error':error_message})

        while True:

            ret, frame = cap.read()
```

```
        if not ret:
            break

        resized_frame = cv2.resize(frame, (224, 224))

        fog_probability = model.predict(np.
            expand_dims(resized_frame, axis=0))[0, 0]

        threshold = 0.5

        if fog_probability > threshold:

            total_fog_percetage.append(
                fog_probability*100)
            percentage_text = f"Fog detection
                percentage: {fog_probability * 100:.2f
                }%"
            cv2.putText(frame, percentage_text, (50,
                50), cv2.FONT_HERSHEY_SIMPLEX, 1, (0,
                0, 255), 2, cv2.LINE_AA)

            cv2.putText(frame, "Fog detected!", (50,
                100), cv2.FONT_HERSHEY_SIMPLEX, 1, (0,
                 0, 255), 2, cv2.LINE_AA)
        else:

            no_fog_percentage = 0

            cv2.putText(frame, "No fog detected",
                (50, 100), cv2.FONT_HERSHEY_SIMPLEX,
                1, (0, 255, 0), 2, cv2.LINE_AA)

        cv2.imshow('Video Frame', frame)

        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

cap.release()
cv2.destroyAllWindows()

try:
    average_fog_percentage = sum(
        total_fog_percetage) / len(
        total_fog_percetage)
```

```
            if  average_fog_percentage >= 50 :
                subject = 'Fog detection'
                mail_content =f'warning : fog detection
                    percentage is {average_fog_percentage
                    }'
                from_mail = 'fogdetectiontrial@gmail.com'
                recipient_list = ['traficunofficial@gmail
                    .com']
                send_mail(subject, mail_content,
                    from_mail, recipient_list)

                return render(request,'fog_percentage.
                    html',{'f': average_fog_percentage })

        except ZeroDivisionError:
            print("Error: Division by zero. Unable to
                calculate the average.")

            return render(request,'fog_percentage.html
                ',{'f':no_fog_percentage})
    else:
        return render(request,'home.html')
```

fogdetection(2-01-2024).ipynb

```
import os
import cv2
import numpy as np
import tensorflow as tf
from tensorflow.keras.preprocessing import image
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing.image import
    ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D,
     Flatten, Dense

train_dir = "C:/Users/my/OneDrive/Desktop/project/
    fog_detection/fog_dataset"

train_datagen = ImageDataGenerator(rescale=1./255,
    validation_split=0.2)

train_generator = train_datagen.flow_from_directory(
    train_dir,
```

```
    target_size=(150, 150),
    batch_size=32,
    class_mode='binary',
    subset='training'
)

validation_generator = train_datagen.flow_from_directory(
    train_dir,
    target_size=(150, 150),
    batch_size=32,
    class_mode='binary',
    subset='validation'
)

model = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(32, (3, 3), activation='relu',
        input_shape=(150, 150, 3)),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Conv2D(64, (3, 3), activation='relu')
        ,
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Conv2D(128, (3, 3), activation='relu
        '),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(512, activation='relu'),
    tf.keras.layers.Dense(1, activation='sigmoid')
])

model.compile(optimizer='adam', loss='binary_crossentropy
    ', metrics=['accuracy'])
model.fit(train_generator, validation_data=
    validation_generator, batch_size=32, epochs=20,
    validation_steps=len(validation_generator))
model.save("fog_clear_model_2.h5")
model = load_model("keras.model.h5")
```

home.html

```
<!DOCTYPE html>
{% load static %}
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
        initial-scale=1.0">
```

```
    <title>fog-detection</title>
    <link rel="stylesheet" href="https://maxcdn.
        bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min
        .css">
 <link rel="stylesheet" href="{% static 'home.css' %}">
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.
    com/bootstrap/4.5.2/css/bootstrap.min.css">

  </head>
<body style="background-color: rgb(216, 216, 223);">

  <nav class="navbar-nav navbar-expand-lg navbar-light bg
      -danger p-4">
  <a class="navbar-brand" href="{% url 'fog_detection:
      home' %}" style="color: white;font-weight: bold;">
      FOG DETECTION</a>

  </nav>

  <div class="pt-4">
  <h2 style="font-weight:bold ;" class="text-danger">
      DETECT FOG ON YOUR VIDEO </h2>
<div class="form-container">
<form method="POST" enctype="multipart/form-data">
  {% csrf_token %}
  <div class="text-center">
  <input type="file" name="video_file">

  <input  type="submit" value="Upload and Detect Fog">
</div>

</form>
</div>
</div>


</body>
</html>
```

fogpercentage.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
        initial-scale=1.0">
```

```
    <title>fog-percentage</title>
    <link rel="stylesheet" href="https://maxcdn.
        bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min
        .css">

</head>
<body style="background-color: rgb(216, 216, 223);">
    <nav class="navbar-nav navbar-expand-lg navbar-light
        bg-danger p-4">
        <a class="navbar-brand" href="{% url '
            fog_detection:home' %}" style="color: white;
            font-weight: bold;">FOG DETECTION</a>

        </nav>
    <div class="form-container pt-5">

        <h2 class="text-danger font-weight-bold text-
            center">{{f}}% of fog is detected.</h2>

    </div>

</body>
</html>
```

settings.py

```
from pathlib import Path
import os

BASE_DIR = Path(__file__).resolve().parent.parent

SECRET_KEY = 'django-insecure-uqy^qc_gs6#n6a1juaeyc^x2qy!
    l$@18xom0b3*&g$b7%!j@68'

DEBUG = True

ALLOWED_HOSTS = []

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'fog_detection',
]
```

```
MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware
        ',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.
        AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware
        ',
    'django.middleware.clickjacking.
        XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'fog_detection_project.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.
            DjangoTemplates',
        'DIRS': [ BASE_DIR / 'templates'],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug
                    ',
                'django.template.context_processors.
                    request',
                'django.contrib.auth.context_processors.
                    auth',
                'django.contrib.messages.
                    context_processors.messages',
            ],
        },
    },
]

WSGI_APPLICATION = 'fog_detection_project.wsgi.
    application'

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
    }
```

```
}

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.
            UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.
            MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.
            CommonPasswordValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.
            NumericPasswordValidator',
    },
]

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_TZ = True

STATIC_URL = 'static/'
STATICFILES_DIRS = [BASE_DIR / 'static']

DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
MEDIA_URL = '/media/'
MEDIA_ROOT = os.path.join(BASE_DIR , 'media')

EMAIL_BACKEND = 'django.core.mail.backends.smtp.
    EmailBackend'
EMAIL_HOST = 'smtp.gmail.com'
EMAIL_USE_TLS = True
EMAIL_PORT = 587
EMAIL_HOST_USER = 'fogdetectiontrial@gmail.com'
EMAIL_HOST_PASSWORD = 'anmh saxu exmh pfyr'
```

GET SCHEME

PROJECT REPORT

Submitted By

JISNA JOHNY

Reg. No. CCAVBCA038

for the award of the Degree of

Bachelor of Computer Application

(University of Calicut)

under the guidance of

Ms. Rasmi P.M

Assistant Professor

BACHELOR OF COMPUTER APPLICATION
DEPARTMENT OF COMPUTER SCIENCE
CHRIST COLLEGE(Autonomous)
IRINJALAKUDA, KERALA
INDIA

2021-2024

DEPARTMENT OF COMPUTER SCIENCE

CHRIST COLLEGE (AUTONOMOUS)

IRINJALAKUDA

CERTIFICATE

This is to certify that the project report entitled ***"Get Scheme"*** *is a bonfied record of the project work done by* ***Jisna Johny*** *in partial fulfillment of the requirement for the sixth semester of* ***Bachelor of Computer Application*** *in Department of Computer Science of* **CHRIST COLLEGE (AUTONOMOUS) IRINJALAKUDA**

Ms. Rasmi P.M
Assistant Professor,CS
Internal Guide

Ms. Sini Thomas
Head of the Department
Computer Science

EXTERNAL EXAMINER

INTERNAL EXAMINER

DECLARATION

I hereby declare that this project work **"GET SCHEME"** submitted to Christ College (Autonomous) Irinjalakuda, affiliated to Calicut University in partial fulfillment of the requirement for the award of the Bachelor of Computer Application, is a record of original work done by me, under the guidance of Ms. RASMI P.M, Department of computer Science.

Place: Irinjalakuda JISNA JOHNY

ACKNOWLEDGEMENT

First and foremost i like to thank Lord almighty for his providence and for being the guiding light throughout the project. I wish to express my sincere gratitude to our beloved Department head for giving me all the facilities for our project. I take this opportunity to express my sincere gratitude to the class teacher Ms. SOUMYA P.S and head of the department Ms. SINI THOMAS who has been supported me throughout the course of this project. I am thankful for her aspiring guidance and valuable advice during the project work. I express my sincere thanks to my project guide Ms. RASMI P.M for supporting and guiding throughout the project.I would take this opportunity to specially thank all other faculty members for their constant and continuous motivation. Finally i would like to thank my family and friends for giving valuable advice and moral support throughout our project.

ABSTRACT

GET SCHEME is a mobile application designed to streamline access to government schemes for citizens . The app enables different categories of people to access various welfare programs offered by governments, categorized by sectors and demographics. Users can easily get schemes based on their eligibility criteria through the app, improving their access to social welfare opportunities and promoting inclusive governance.

Contents

Chapter 1

1 Introduction

Introducing our groundbreaking app: Get Scheme. Say goodbye to endless searches and confusion when it comes to accessing government assistance programs. Our app is your go-to resource for effortlessly discovering and accessing a multitude of government schemes tailored to your specific needs. Whether you're seeking financial aid, educational grants, healthcare benefits, or housing support, Get Scheme simplifies the process by providing a comprehensive database of available programs. With user-friendly navigation and personalized recommendations, we ensure that you never miss out on valuable opportunities for assistance. Take control of your access to government schemes today with our intuitive and empowering app.

.

1.1 Overview

Introducing our groundbreaking app: Get Scheme. Say goodbye to endless searches and confusion when it comes to accessing government assistance programs. Our app is your go-to resource for effortlessly discovering and accessing a multitude of government schemes tailored to your specific needs. Whether you're seeking financial aid, educational grants, healthcare benefits, or housing support, Get Scheme simplifies the process by providing a comprehensive database of available programs. With user-friendly navigation and personalized recommendations, we ensure that you never miss out on valuable opportunities for assistance. Take control of your access to government schemes today with our intuitive and empowering app.

Chapter 2

2 System Analysis

2.1 Purpose

The purpose of the Get Scheme app is to facilitate easy access to various government assistance programs and schemes, simplifying the process of discovering, managing, and applying for financial aid, educational grants, healthcare benefits, housing support, and more

2.1.1 Existing System

Currently, there is no existing application that comprehensively addresses the diverse needs of individuals seeking government assistance programs. While there are some platforms and websites that provide information about specific schemes or sectors, these offerings are often limited in scope and accessibility. Users may struggle to find relevant programs that match their unique circumstances, leading to frustration and disenchantment with the system.

2.1.2 Proposed System

The proposed system is a mobile application designed to simplify access to government assistance programs for individuals from various backgrounds and circumstances. Users will be able to register for an account, input personal details, and receive personalized recommendations for relevant schemes based on their eligibility criteria. The system will maintain a centralized database of government assistance programs across different sectors, providing real-time updates and deadlines. With a user-friendly interface, robust security measures, and community engagement initiatives, the platform aims to empower users to navigate the complexities of bureaucracy effectively and access the support they need to improve their lives.

2.2 Problem definition

The problem definition of the Get Scheme app revolves around the fragmented and often confusing process of accessing government assistance programs. Users currently face challenges in navigating multiple sources of information and eligibility criteria, leading to inefficiencies and missed opportunities for valuable assistance. The app aims to address these issues by providing a unified platform that simplifies the discovery, management, and application process for government schemes

2.3 Feasibility Study

After the problem is clearly understood and solutions proposed, the next step is to conduct the feasibility study. Feasibility study is defined as evaluation or analysis of the potential impact of the proposed project or program. The objective is to determine whether the proposed system is feasible. There are three aspects of feasibility study to which the proposed system is subjected as discussed below.

2.3.1 Technical Feasibility

Technical feasibility assess whether the current technical resources are sufficient for the new system. We can upgrade the level of technology for supporting our website. We check whether the proposed system can be implemented in the present system without supporting the existing hardware.

2.3.2 Economical Feasibility

Economic feasibility determines whether the time and money are available to develop the system. There is no additional hardware used to develop the site, it is inexpensive to build.

2.3.3 Operational Feasibility

Operational feasibility determines if the human resources are available to operate the system once it has been installed. No extra training is needed to use this system. Anyone who has knowledge in internet in english language can easily use the system. The resources that are required to implement or install are already available with the organization.

Chapter 3

3 Software Requirement Specification

3.1 Purpose

The purpose of defining software requirements for the Get Scheme app is to outline the specific functionalities, features, and performance criteria necessary to meet user needs effectively. This ensures that the development team understands the project objectives clearly and can design, implement, and test the software accordingly, ultimately delivering a robust and user-friendly application that fulfills its intended purpose of simplifying access to government schemes and enhancing financial well-being.

3.2 Scope

The scope of the Get Scheme project involves developing a user-friendly mobile application that consolidates information on government schemes, offers personalized recommendations, and guides users through the application process. It aims to streamline access to assistance programs, enhance financial literacy, and improve the overall user experience in navigating government initiatives.

3.3 Overall Description

The software requirements for our project aim to address the needs of individuals seeking access to government assistance programs in a user-friendly and efficient manner.The software requirements encompass the entire development lifecycle, from gathering user requirements to design, implementation, testing, deployment, and maintenance. The goal is to create a robust and user-centric platform that democratizes access to government assistance programs, promoting inclusivity, transparency, and efficiency.

3.3.1 Product Perspective

Our project operates within the broader context of government assistance programs and aims to streamline the process of accessing these programs for individuals.

3.3.2 Product Functionality

Our platform offers a range of functionalities aimed at simplifying the process of accessing government assistance programs and providing users with personalized support.

3.3.3 Users and Characteristics

Admin is able to add new schemes to the database.other category of users like farmers,students,government employees,women,military,sports can retrive the schemes based in their criteria of eligibilty.

3.4 Specific Requirements

3.4.1 Hardware Requirements

- System:Android Device
- Processor: Mediatek Helio g85
- Speed: Above 1GHz
- RAM capacity: 512 MB
- Hard Dsk drive: 2 GB

3.4.2 Software Requirements

- Operating System:Android
- Languages used: Python Django,Flutter
- Database : MySql
- Technologies used: HTML,Javascript,CSS

3.5 Functional Requirements

It contains three main modules.

- 1.Admin
- 2.User

Admin

An Admin account is used for adding new schemes. The new schemes which are added by the admin will be notified to the users through their registered email id.

User

The user can register or login the app.The new schemes based on the eligilbility criteria provided by the user will be displayed in the my scheme section.The new schemes which are added recently will be displayed in the new schemes section.The my profile section is used to edit the eligibilty criteria provided by the user.

3.6 Non Functional Requirements

Non-functional requirements define the overall qualities or attributes of the resulting system. Non-functional system place restrictions on the product being developed, the developed process, and specify external constraints that the product must meet. Examples of non-functional requirements include safety, security, usability, reliability and performance requirements. Project management issues (costs, time and schedule) are often considered as non-functional requirements. The principal non - functional constraints which are relevant to critical systems :

- Performance
- Security
- Safety
- Usability

Performance

Performance requirements concern the speed of operation of a system. Types of performance requirements :

- Response requirements (how quickly the system reacts to a user input).
- Throughput requirements (how much the system can accomplish within a specified amount of time).
- Availability requirements (is the system available for service when requested by end users). The speed of operation of this system is adequate for the requirements.

Reliability

- Reliability is the ability of a system to perform its required function under stated conditions for a specified period of time.
- Constraints on the runtime behavior of the system. This system is reliable because its functionalities can be done on the required conditions.

Safety

Safety requirements are not required which exclude unsafe situation from the possible solution of the system.

Usability

Usability is the ease with which a user can learn to operate, prepare inputs for, and interpret outputs of system or components. Usability requirements include :

- Information error messages.
- Well-formed user interfaces.

3.7 Interface Requirements

3.7.1 Hardware interfaces

The system must run over the internet, all the hardware shall require to connect internet will be hardware interface for the system. As for example modem, WAN-LAN.

3.7.2 Software interfaces

Software interface required for the working of the project is the appropriate operating system.

3.7.3 Communication interfaces

The user of the site communicate using the network connectivity and the data set is accessed

3.8 Security Requirements

- User accesses only their account.
- Validation of input is handled.
- This application containing the computer systems is physically secured against arms or surreptitious entry by intruders.
- Users must be authorized carefully to reduce changes of any such user giving access to an intruder in exchange for a bribe or other favour.

3.9 Platform Used

For the Get Scheme app project, the Android operating system serves as the foundation for delivering a seamless and accessible user experience. Leveraging the Android OS allows the development team to tap into a vast ecosystem of tools, libraries, and resources provided by Google, facilitating efficient app development and optimization for a wide range of devices. By adhering to Android's design guidelines and platform standards, the app ensures consistency and familiarity for users across different devices and versions of the OS. Moreover,

Android's robust security features enable the implementation of stringent measures to safeguard user data and privacy, instilling trust and confidence among users. Overall, the utilization of the Android OS in this project enables the delivery of a high-quality and feature-rich app that effectively serves its purpose of simplifying access to government schemes and enhancing financial well-being for users.

3.10 Technologies Used

Python Django

Python Django is a high-level web framework that simplifies the development of complex web applications by providing a clean and pragmatic design. It follows the "batteries-included" philosophy, offering built-in features for common web development tasks such as URL routing, database management, form handling, and user authentication. Django's emphasis on DRY (Don't Repeat Yourself) principle and its robust security features make it a popular choice for developers seeking efficiency, scalability, and rapid development. With its rich ecosystem of third-party packages and active community support, Django empowers developers to build sophisticated web applications with ease.

Flutter

Flutter is an open-source UI software development kit (SDK) created by Google for building natively compiled applications for mobile, web, and desktop from a single codebase. It uses the Dart programming language and offers a rich set of pre-designed widgets that enable developers to create visually stunning and highly performant user interfaces. Flutter's hot reload feature allows for rapid iteration and experimentation during development, resulting in faster development cycles. With its cross-platform capabilities and expressive UI framework, Flutter has gained popularity among developers for building beautiful and responsive applications across various platforms.

MySQL

MySQL, pronounced either "My S-Q-L" or "My Squel, " is an open source relational database management system. It is based on the structure query language (SQL), which is used for adding, removing and modifying information in the database. Standard SQL commands, such as ADD, DROP, INSERT,and UPDATE can be used with MySQL.

What distinguishes PHP from something like client - side Javascript is that the code is executed on the server, generating HTML which is then sent to the client. The client would receive the results of running that script, but would not know what the underlying code was. You can even configure your web server to process all your HTML files with PHP, and then there's really no way that users can tell what you have up your sleeve. The best things in using PHP are

that it is extremely simple for a newcomer, but offers many advanced features for a professional programmer.

Chapter 4

4 Design Document

4.1 Purpose

The purpose of the design document is to provide a comprehensive blueprint for the development team, outlining the architectural design, system components, and technical specifications required to build the software solution. This document serves as a roadmap that guides the development process from conception to implementation, ensuring alignment with project goals and requirements. By detailing the system's structure, interfaces, data flow, and functionality, the design document facilitates effective communication among stakeholders, fosters collaboration within the development team, and enables informed decision-making throughout the software development lifecycle. Additionally, the design document serves as a reference for future maintenance and updates, providing a clear understanding of the system's design rationale and facilitating efficient troubleshooting and modification as needed. Overall, the design document plays a crucial role in translating requirements into a coherent and actionable plan, ultimately leading to the successful delivery of a high-quality software solution that meets the needs of its users.

4.2 Scope

In the design document for the Get Scheme app project, the scope outlines the specific functionalities and features that the app will encompass. This includes defining the categories of government schemes to be covered, such as finance, education, healthcare, and housing. Additionally, the scope may detail the user interface design, including the layout of screens, navigation flow, and interactive elements.

4.3 Overview

the design document serves as a comprehensive guide for implementing the software solution, ensuring that all aspects of system design and architecture are properly documented and aligned with project requirements. It provides a roadmap for developers, designers, and testers to follow during the development process, facilitating efficient collaboration and successful project delivery.

4.4 Data Design

Database are the storehouses of data used in the software systems.The data is stored in tables inside the database.Several tables are created for the manipulation of the data for the system.Two essential settings for a database are

- Primary Key- The field that is unique for all the record occurrence.

- Foreign Key- The field used to set relation between tables.

Normalization

- Normalization is the process of organizing the data in the database.
- Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate the undesirable characteristics like Insertion, Update and Deletion Anomalies.
- Normalization divides the larger table into the smaller table and links them using relationship.
- The normal form is used to reduce redundancy from the database table.

Tables

User

Name	DataType	Constraints	Description
First name	char	Primarykey	Name
Lastname	char(100)	Notnull	Name
gender	char(250)	Notnul	Gender
Email	varchar(250)	Notnul	MailID
Password	varchar(250)	Notnul	Password

Scheme

Name	DataType	Constraints	Description
Scheme app	char	Primarykey	Name
Type	char	Notnull	Types
Description	varchar(250)	Notnul	Description
Start Age	int	Notnul	Age
End Age	int	Notnul	Age
Link	varchar	Notnul	Link

Chapter 5

5 Development of the System

The development of the Get Scheme system involves a systematic approach encompassing various stages. It begins with thorough requirement gathering, where project goals, functionalities, and user expectations are identified and documented. Following this, the design phase commences, where system architecture, database schema, and user interface designs are meticulously planned to ensure alignment with project requirements and user needs. Subsequently, developers begin coding the system, utilizing technologies like Django for backend development and Flutter for frontend implementation. Rigorous testing is then conducted to identify and rectify any issues, ensuring the system's reliability, functionality, and performance. Once testing is complete, the system is deployed to production environments, and ongoing maintenance and updates are performed to address evolving requirements and enhance user experience. Collaboration among developers, designers, testers, and stakeholders is crucial throughout the development process to ensure the successful delivery of the Get Scheme app.

Chapter 6

6 System Testing

Testing is the penultimate step of software development.An elaborate testing of the data is prepared and the system is using the test data,while doing testing,errors are noted and correction is made.The users are trained to operate the developed system.Both hardware and software securities are made to run the developed system successfully.System testing is aimed at ensuring the system works accurately before the live operation commences.

6.1 Test Plan

6.1.1 Scope

This test plan will cover the following testing activities as identified in the testing strategy.

- White Box Testing

White box testing is a testing technique, that examines the program structure and derives test data from the program logic/code. In white box testing, the UI is bypassed. Inputs and outputs of ratings are tested directly at the code level and the results are compared against specifications. It Reveals errors in "hidden" code.

- Black Box Testing

Black - box testing is a method of software testing that examines the functionality of an application based on the specifications. Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end - user would.

- Unit Testing

The module includes four main components to undergo unit testing. It checks the syntax error, logic error and validity of the program correctness. This test will be performed by the developers.

- Integration Testing

After all components pass the unit testing, they get to be tested whether they work correctly when they are running concurrently and communicating to each other. Integration tests exercise an entire subsystem and ensure that a set of components play nicely together.

- Internal Data Testing

We will test the validity of the data before it enters the database to avoid any problems that may face us in the database. We will test the encryption of the personal information of all the users along with the user names and passwords to ensure maximum security of the users privacy.

6.1.2 Software risk issues

The software risks of the Get Scheme app include potential security vulnerabilities, such as data breaches or unauthorized access to user information, which could compromise user privacy and trust. Additionally, software bugs or glitches may lead to system malfunctions or unexpected behavior, impacting user experience and satisfaction. Moreover, compatibility issues with different devices, operating systems, or third-party libraries could pose challenges in ensuring consistent functionality and performance across various platforms.

6.1.3 Features to be tested

- Test whether correct user name and password allows you to login.
- Test whether invalid user name and password prevents you from login.
- Test whether there is any connection problem in the Server.
- Test whether the user and admin details are entered correctly.
- Test whether invalid data entry allows saving data successfully.
- Test whether all pages are loaded correctly.
- Test whether the provided security works correctly.

6.2 Test consolidation

6.2.1 Test item

The items or features to be tested in the test cases are included in the document. Each and every user input is tested. The present condition of the system is tested. It is checked to make sure that environment is ready for the application to work.

6.2.2 Input specifications

These are the inputs required to execute the test case. All required inputs including data elements and values are given below.

Data	Values
Email	A valid email ID
Password	Combination of characters and numbers
Phone number	10 Digit number

Chapter 7

7 System Implementation and Maintenance

7.1 Implementation

System implementation is the conversion of new system into an operating one which involves creating compatible files, training clients and installing hardware. User training is crucial for minimizing resistance to change and giving chance to prove its worth. Training aids user friendly manuals and healthy screens provide the user with a good start. Software maintenance follows conversion to the extent that changes are necessary to maintain satisfactory operations relative to changes in the user's environment. Maintenance often includes minor enhancements or corrections to the problem that surface late in the systems operations. In the implementation phase, the team builds the components either from scratch or by composition. Given the architecture document meant from the design phase and the requirement document from the analysis phase, the team should build exactly what has been requested, though there is still room for innovation and flexibility. For example, a component may be narrowly designed for this particular system, or the component may be made more general to satisfy a reusability

- Careful planning
- Investigation of system and constraints
- Design the methods to achieve changeover.
- Training the staff in the changed phase.
- Evaluation of change over method.
- The method of implementation and time scale to be adopted are found out initially.

7.2 Maintenance

This phase occurs as a result of deploying the whole system at the end users organization. They will perform the beta testing at the end users and inform to the developers about any needed modification to the application. The customer records all the problems that are encountered during the beta testing and reports these to the developer at regular intervals.

7.2.1 Corrective Maintenance

Even with the best quality assurance activities, it is likely that the customer will uncover defects in the software. Corrective maintenance changes the software to correct the defects. Corrective Maintenance activity may consist of repair,

restoration or replacement of equipment. This activity will be result of a regular inspection, which identifies the failure in time for corrective maintenance to be planned and scheduled, then performed during a routine maintenance shutdown.

7.2.2 Adaptive Maintenance

Over time, the original environment(CPU, operating system, business rules, external product characteristics) for which the software was developed is likely to change. Adaptive maintenance results in modification to the software to accommodate changes to its external environment.

7.2.3 Enhanced Maintenance

As software is used, the customer/user will recognize additional functions that will provide the benefit. Perfect maintenance extends the software beyond its original functional requirements.

7.2.4 Preventive Maintenance

Computer software deteriorates due to change, and because of this preventive maintenance often called software re-engineering, must be conducted to enable the software to serve the needs of its end users. Preventive maintenance makes changes to computer programs so that they can be more easily corrected, adapted and enhanced.

Chapter 8

8 Conclusion and Future Scope

8.1 Conclusion

This app facilitate access to government schemes presents a significant opportunity to enhance citizen engagement and promote inclusivity in accessing essential services. By providing a user-friendly interface, timely updates, and personalized recommendations, the app can empower individuals to make informed decisions and avail themselves of relevant benefits. Moreover, such a platform can foster transparency, efficiency, and accountability within the government system, ultimately contributing to socio-economic development and welfare.

8.2 Future Scope

The future scope refers to the potential opportunities, advancements, and possibilities for growth, development, and expansion within a particular field, project, or concept. It encompasses various aspects, including technological advancements, market trends, policy changes, and societal needs, that can influence and shape future outcomes. Future scope entails exploring new avenues for innovation, addressing emerging challenges, and seizing opportunities for improvement and progress. It involves strategic planning, research, and forecasting to anticipate and capitalize on future trends and opportunities, thereby ensuring sustainable growth and success.

- Integration with Other Services: Integrating the app with other government services and platforms to create a comprehensive ecosystem for citizens to access various benefits seamlessly.
- Feedback Mechanisms: Establishing robust feedback mechanisms to gather insights from users, identify pain points, and continuously improve the app's usability and effectiveness.
- Partnerships and Collaborations: Collaborating with NGOs, community organizations, and private sector partners to amplify the app's impact, reach underserved populations, and leverage additional resources for implementation and promotion.
- Personalization and AI Recommendations: Utilizing artificial intelligence to provide personalized recommendations based on users' demographics, preferences, and past interactions with the app.

Appendix

A Data Flow Diagram

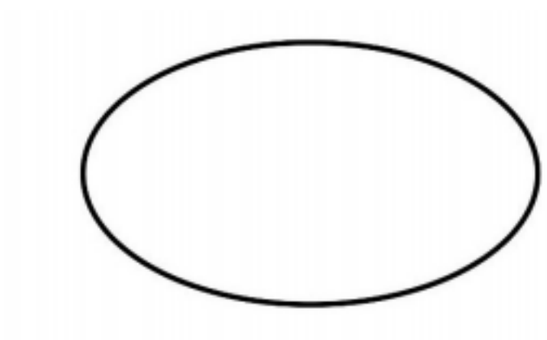
Data flow is the one of the best way of documenting the entire functionality of the system.For the system ,which will have data flows in and have some processing inside and then some data flow out from the system can be documented or represented effectively by means of data flow out from the system can be documented or represented effectively by means of data flow diagrams. The data flow diagram are a diagrammatic representation of the system,which has input,process and output.Once any system is represented using a data flow diagram we can identify the following things easily:

- Various entities interacting with the system are identified
- Flow of data from one entity to another is identified
- The various processes involved in between the interaction of two or more entities in the system are clearly pointed out
- The various data stores which hold the data in between the process,are clearly identified

Some Data Flow Diagram charting forms:

A.1 External source or receiver

A.2 Transform process

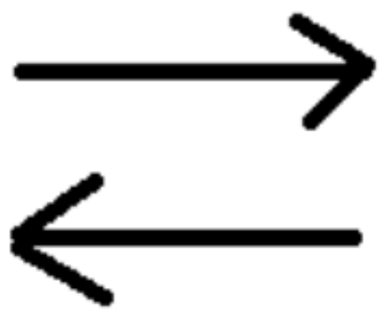


A process represents transformation where incoming data flowsare changed into outgoing data flow

A.3 Data Store

A data store is repository of data that is to be stored for use by one or more process may be as simple as buffer or queue or sophisticated as relational database. They should have clear names.If a process merely uses the contest of store and does not alter it,the arrowhead goes only form the store to the process. If a process alters the details in the store then double-headed arrow is used.

A.4 Data flow

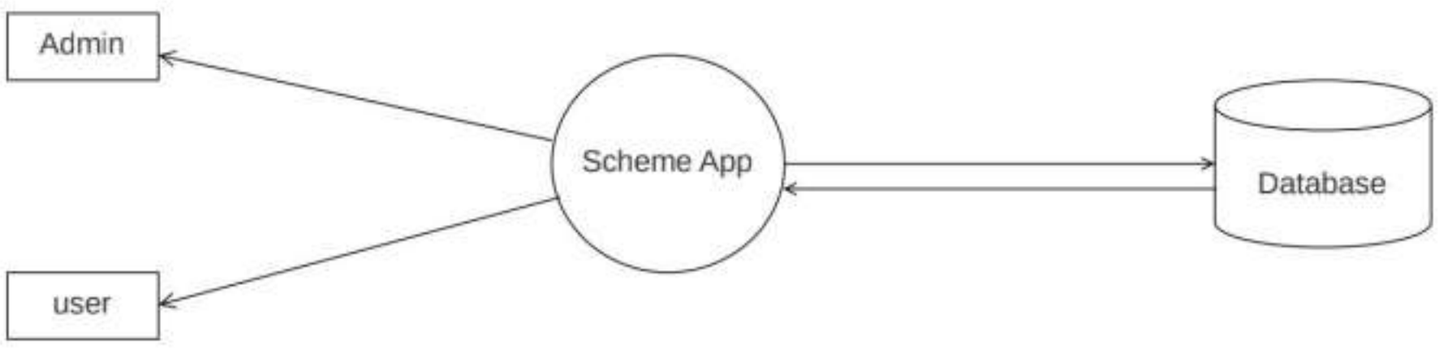


A data flow is a route, which enable packets of data to travel from one point to another.Data may flow,with arrowhead pointing in the direction of the flow

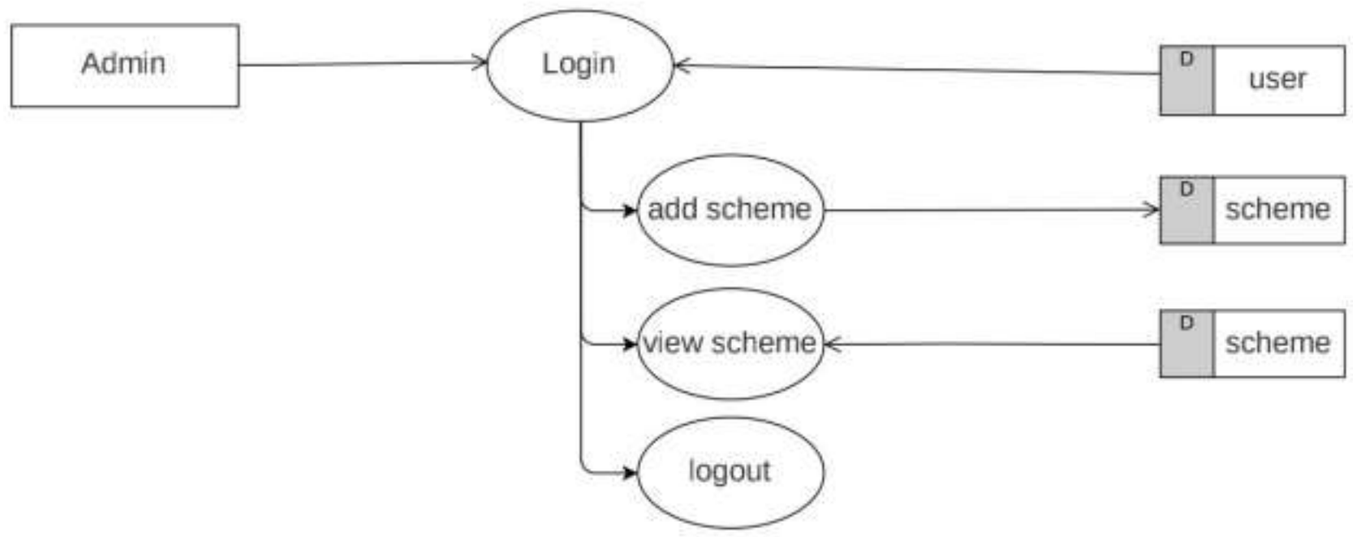
B Data Flow Diagrams

B.1 DFD1

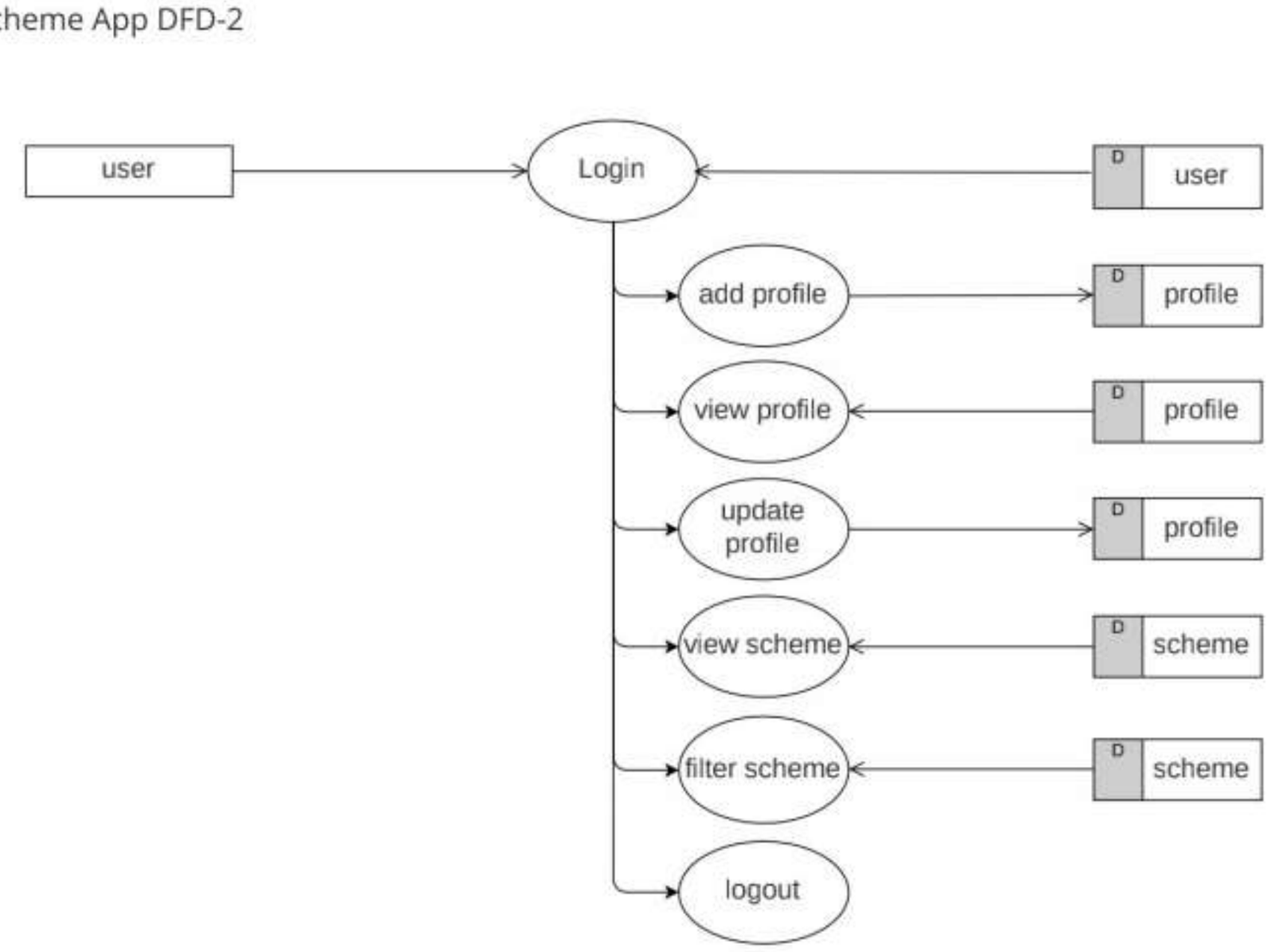
Scheme App DFD-0



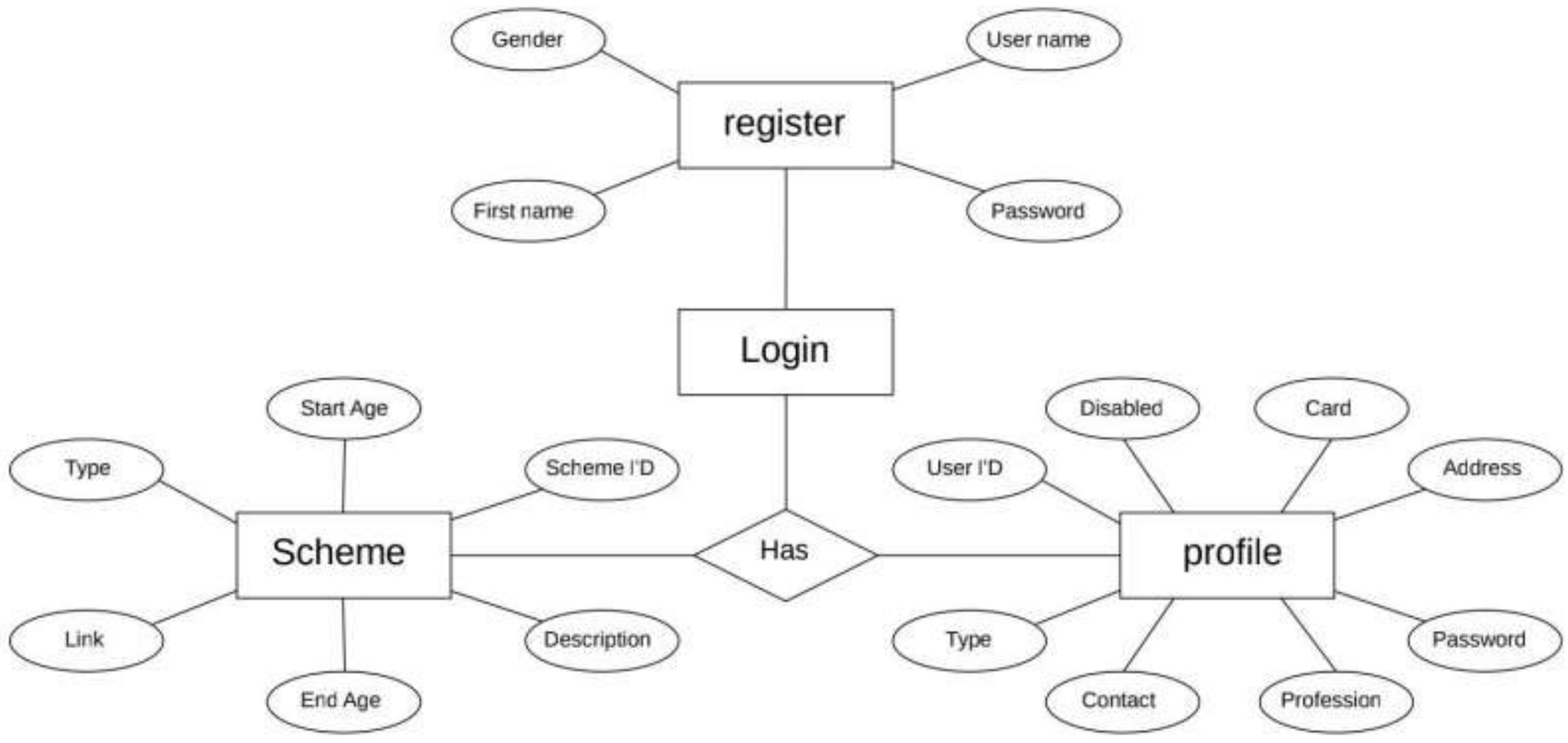
Scheme App DFD-1



B.2 DFD2

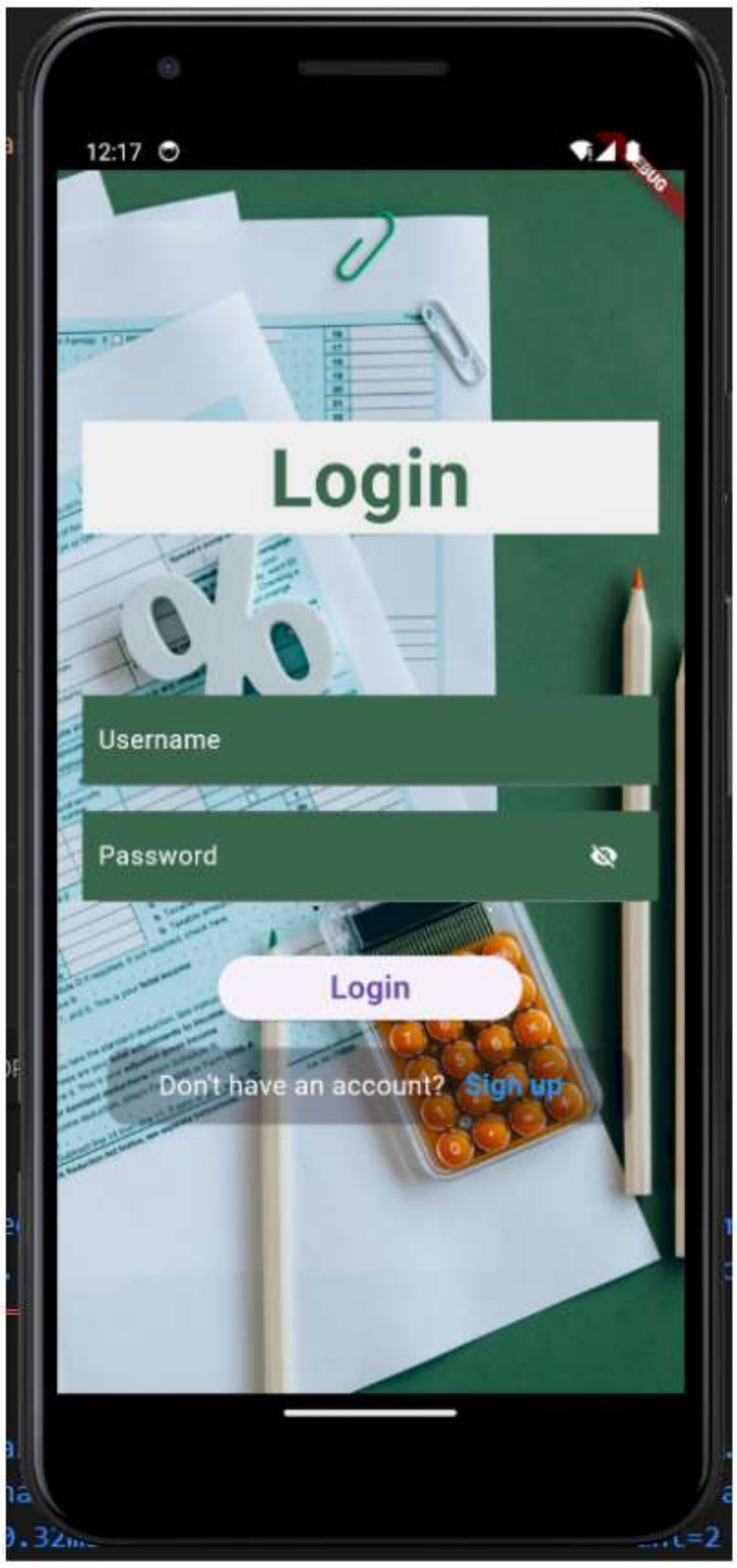

Scheme App DFD-2
user
Login
D
user
add profile
D
profile
view profile
D
profile
update profile
D
profile
view scheme
D
scheme
filter scheme
D
scheme
logout

B.3 ER Diagram

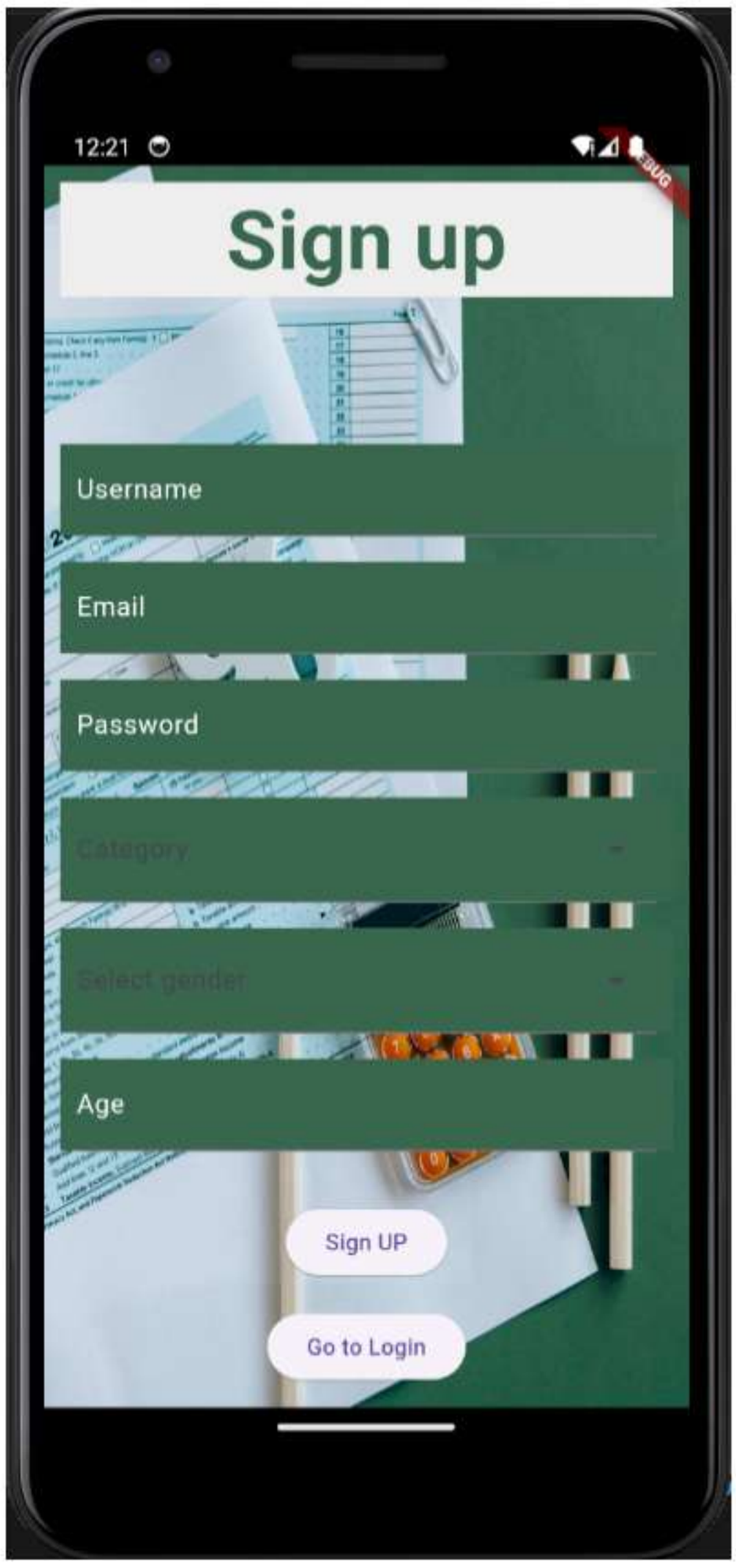


C USER INTERFACES

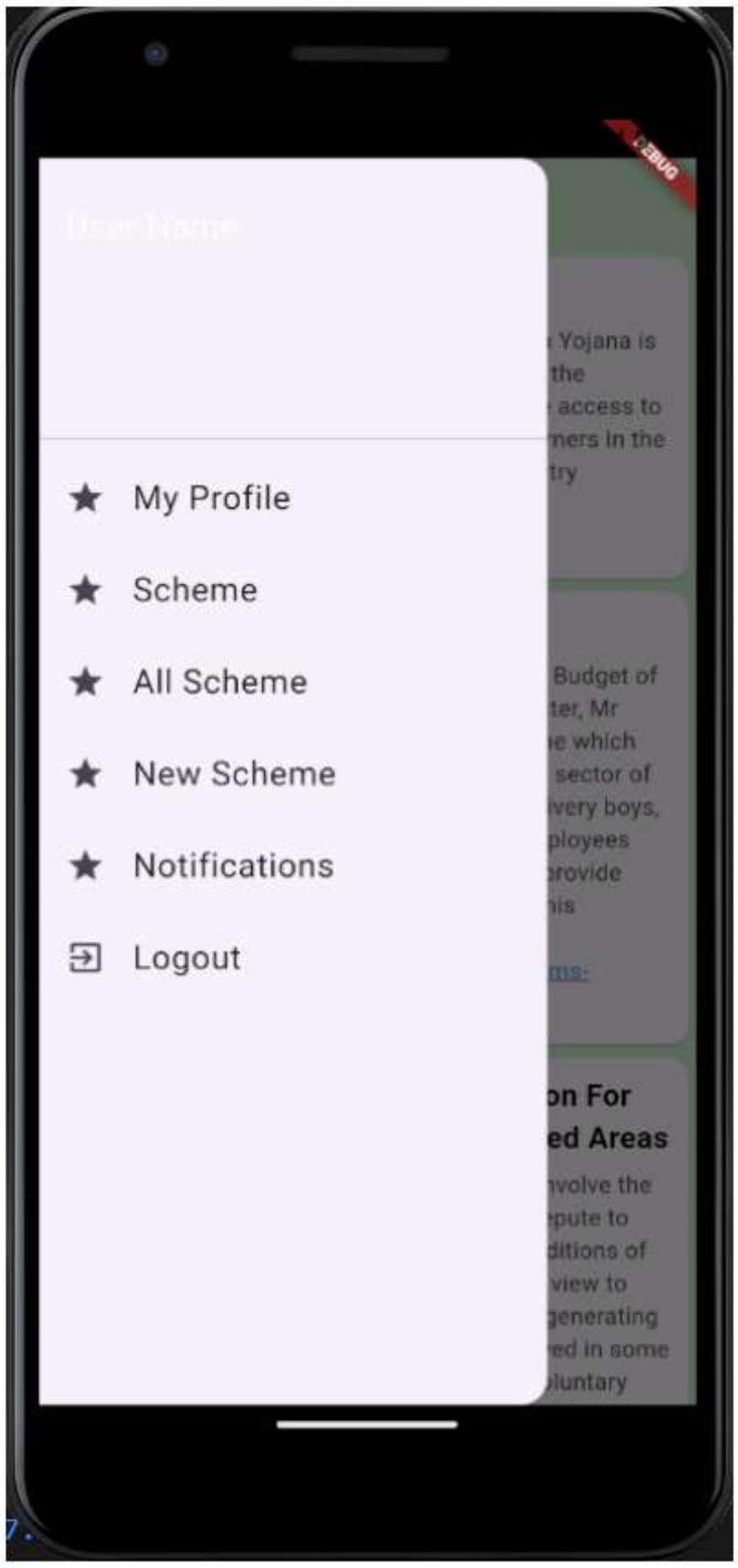
C.1 LOGIN



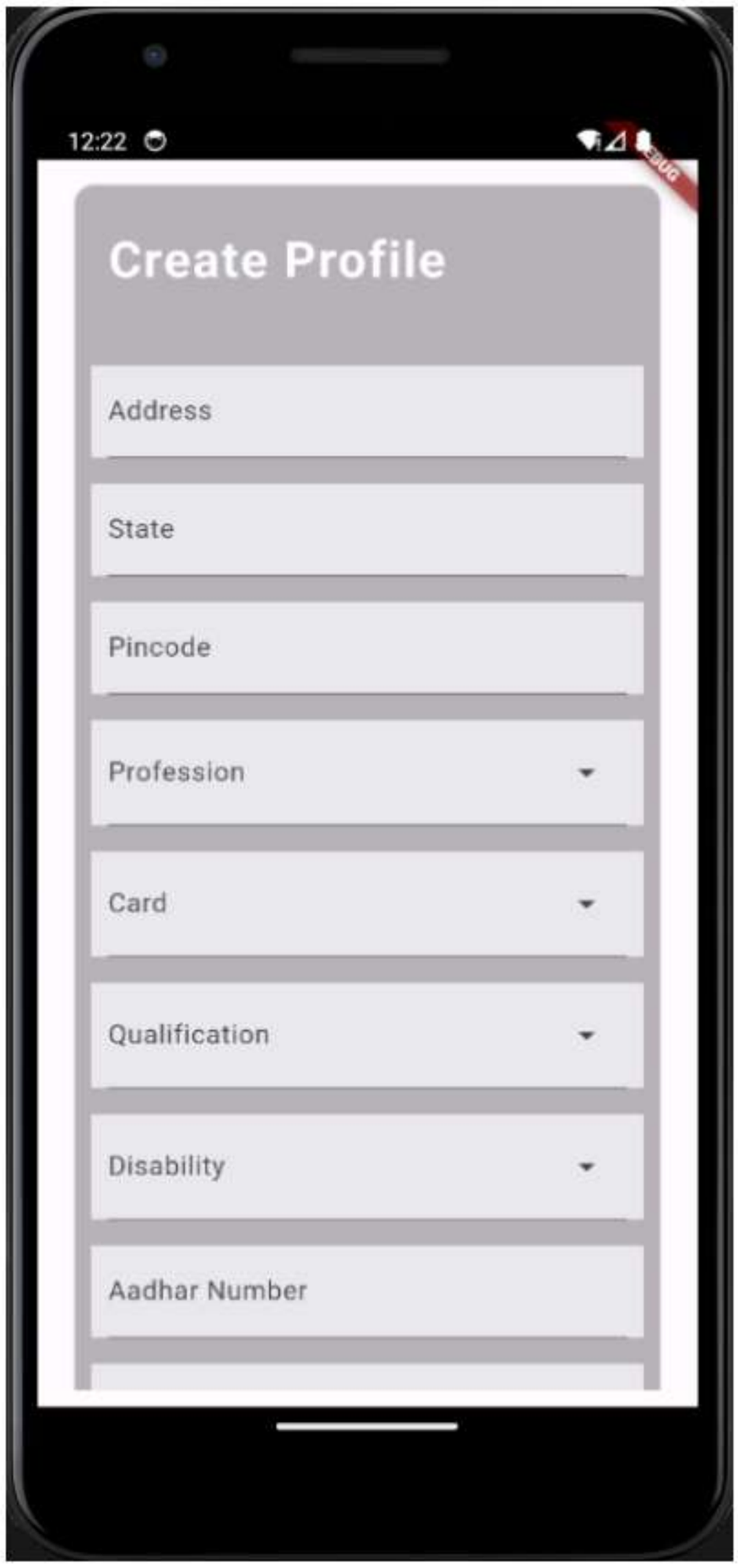
C.2 SIGNUP


12:21
DEBUG
Sign up
Username
Email
Password
Category
Select gender
Age
Sign UP
Go to Login

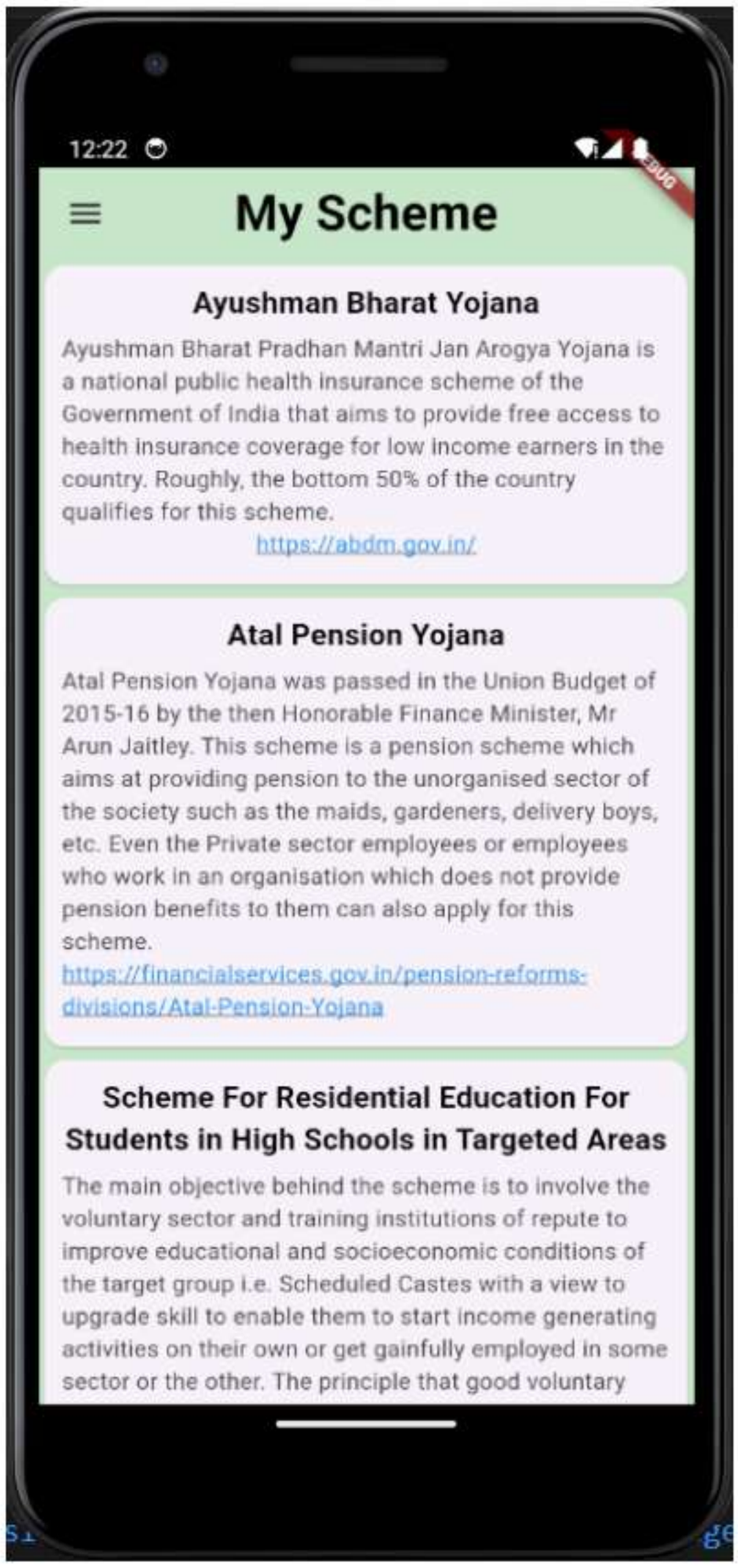
C.3 SIDE MENU


DEBUG
User Name
My Profile
Scheme
All Scheme
New Scheme
Notifications
Logout
Yojana is
the
access to
mers in the
try
Budget of
ter, Mr
e which
sector of
very boys,
ployees
provide
his
ms-
on For
ed Areas
nvolve the
pute to
ditions of
view to
generating
ed in some
luntary

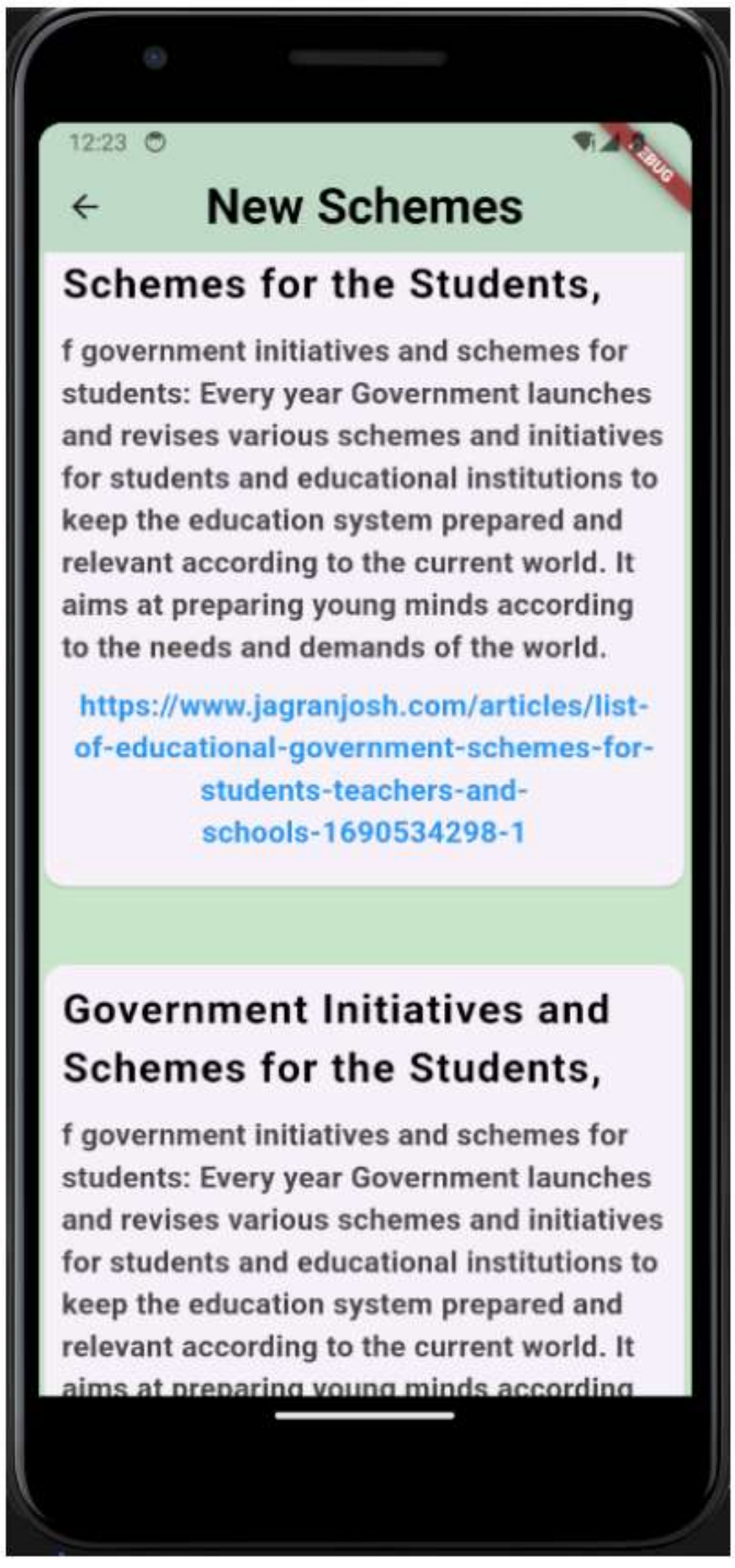
C.4 CREATE PROFILE



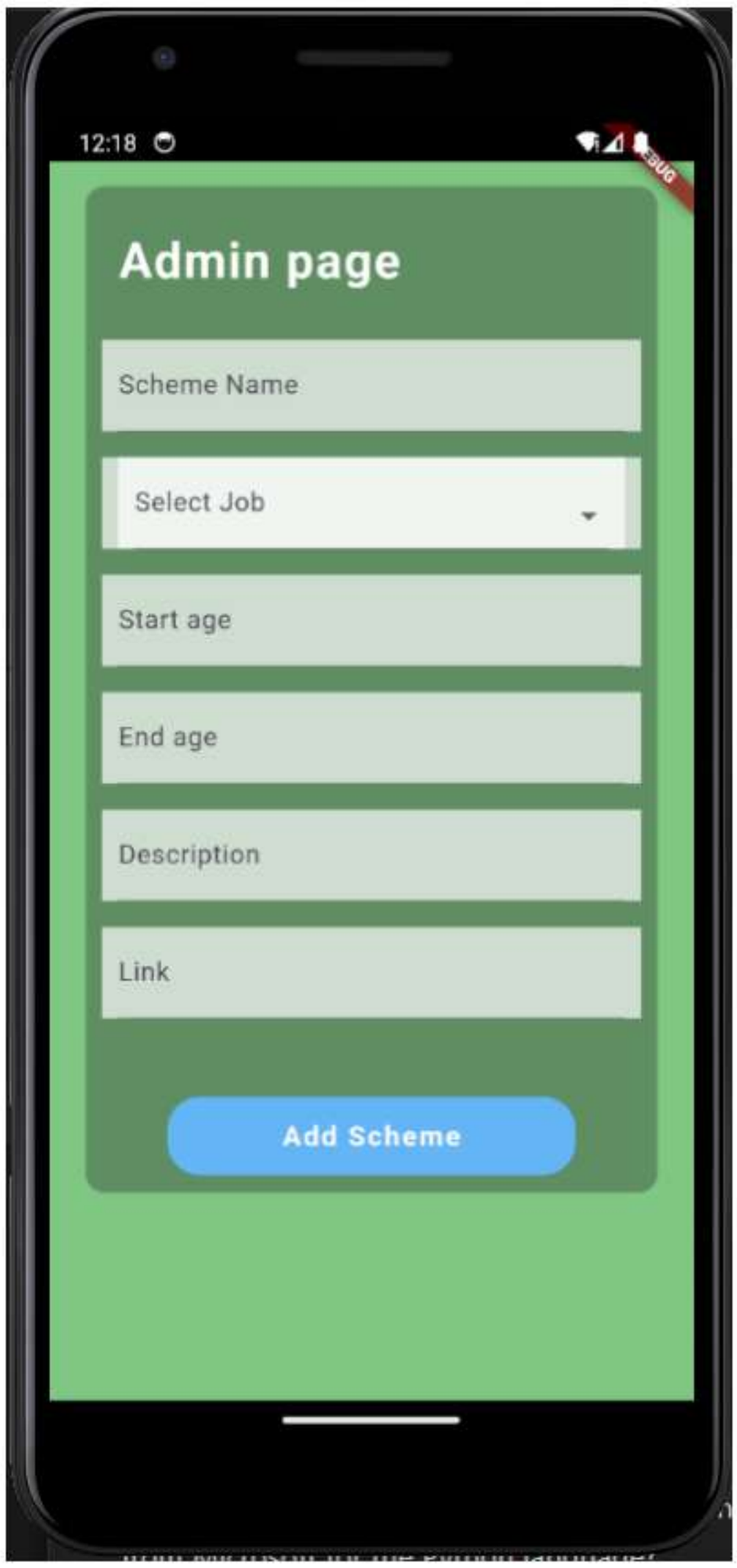
C.5 MY SCHEME


12:22
My Scheme
Ayushman Bharat Yojana
Ayushman Bharat Pradhan Mantri Jan Arogya Yojana is a national public health insurance scheme of the Government of India that aims to provide free access to health insurance coverage for low income earners in the country. Roughly, the bottom 50% of the country qualifies for this scheme.
https://abdm.gov.in/
Atal Pension Yojana
Atal Pension Yojana was passed in the Union Budget of 2015-16 by the then Honorable Finance Minister, Mr Arun Jaitley. This scheme is a pension scheme which aims at providing pension to the unorganised sector of the society such as the maids, gardeners, delivery boys, etc. Even the Private sector employees or employees who work in an organisation which does not provide pension benefits to them can also apply for this scheme.
https://financialservices.gov.in/pension-reforms-divisions/Atal-Pension-Yojana
Scheme For Residential Education For Students in High Schools in Targeted Areas
The main objective behind the scheme is to involve the voluntary sector and training institutions of repute to improve educational and socioeconomic conditions of the target group i.e. Scheduled Castes with a view to upgrade skill to enable them to start income generating activities on their own or get gainfully employed in some sector or the other. The principle that good voluntary

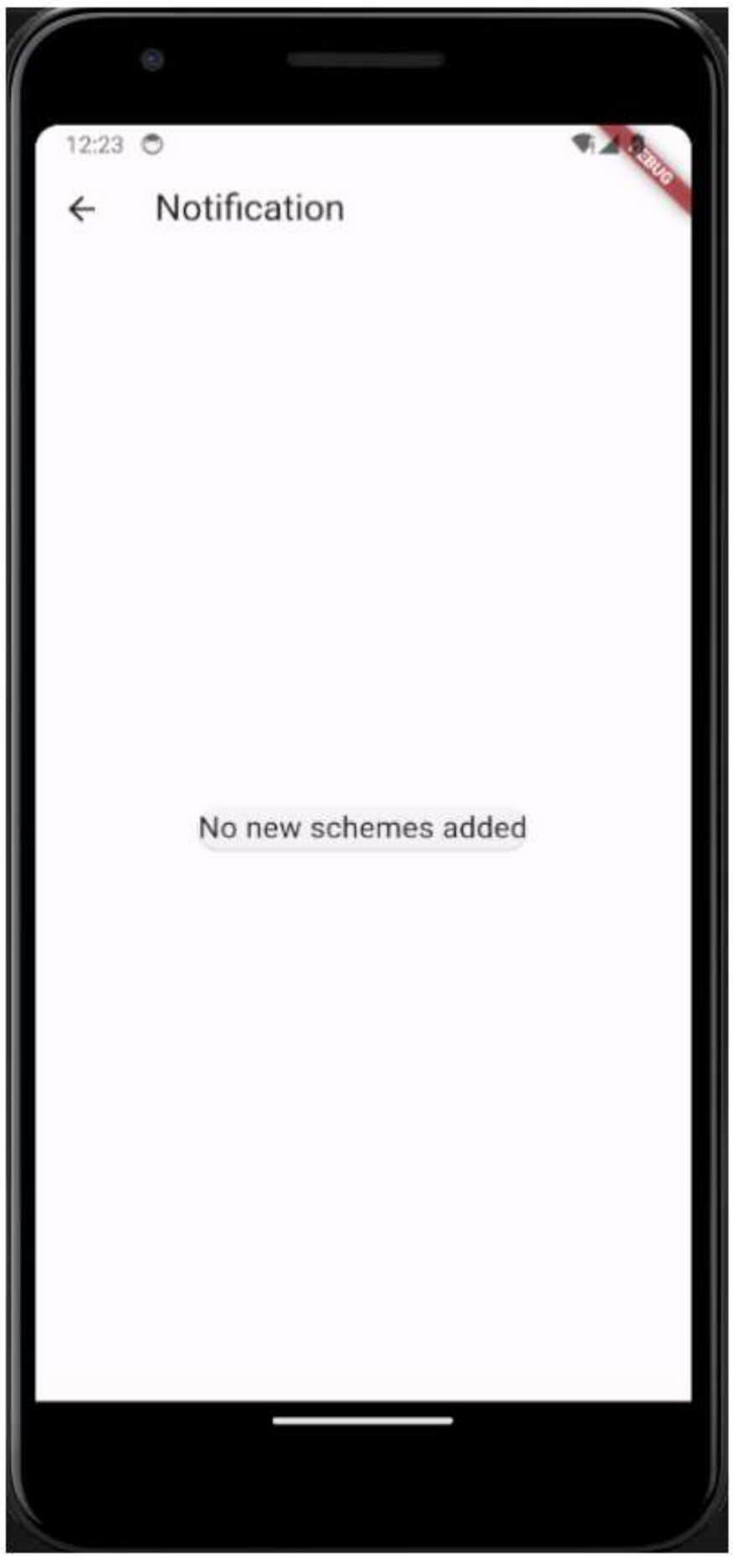
C.6 NEW SCHEME



C.7 ADMIN PAGE



C.8 NOTIFICATION



D CODE

admin.py

```
from django.contrib import admin
from .models import *
# Register your models here.
admin.site.register(CustomUser)
admin.site.register(ProfileDB)
admin.site.register(State)
```

apps.py

```
from django.apps import AppConfig
class AdminuiConfig(AppConfig):
    default_auto_field = 'django.db.models.BigAutoField'
    name = 'AdminUI'
```

models.py

```
from django.db import models
from django.contrib.auth.models import User,AbstractUser

class CustomUser(AbstractUser):
    GENDER_CHOICES = [
        ('Male', 'Male'),
        ('Female', 'Female'),
        ('Others', 'Others')
    ]
    gender = models.CharField(max_length=100, choices=
        GENDER_CHOICES, null=True)

    PROFESSION_CHOICES = [
        ('Farmer', 'Farmer'),
        ('Student', 'Student'),
        ('Disabled', 'Disabled'),
        ('Women', 'Women'),
        ('Teacher', 'Teacher'),
        ('Sports', 'Sports'),
        ('Military', 'Military'),
        ('Government Employee', 'Government Employee'),

    ]
    profession = models.CharField(max_length=100, choices
        =PROFESSION_CHOICES, default='Farmer')
    age = models.IntegerField(null=True)

class State(models.Model):
```

```
    name=models.CharField(max_length=100)

    class Meta:
        ordering =('name',)
        verbose_name_plural='States'

    def __str__(self):
            return self.name

# Create your models here.
class ProfileDB(models.Model):
    user = models.OneToOneField(CustomUser, on_delete=
        models.CASCADE, unique=True)
    Address = models.CharField(max_length=1000, null=True
        , blank=True)
    State = models.CharField(max_length=100,null=True)
    Pincode = models.IntegerField(null=True, blank=True)
    Aadhar_Number = models.BigIntegerField(null=True,
        blank=True)
    Contact = models.BigIntegerField(null=True, blank=
        True)
    options=(
        ('Farmer', 'Farmer'),
        ('Student', 'Student'),
        ('Disabled', 'Disabled'),
        ('Women', 'Women'),
        ('Teacher', 'Teacher'),
        ('Sports', 'Sports'),
        ('Military', 'Military'),
        ('Government Employee', 'Government Employee'),

    )
    profession = models.CharField(max_length=100,choices=
        options,default='Farmer')
    options=(
        ('APL','APL'),
        ('BPL','BPL'),
        ('AAY','AAY')
    )
    card=models.CharField(max_length=100,choices=options,
        null=True)
    options=(
        ('SSLC','SSLC'),
        ('PLUS TWO','PLUS TWO'),
        ('UG','UG'),
        ('PG','PG')
```

```
)
qual=models.CharField(max_length=100,choices=options,
    null=True)
options=(
    ('Mobility Impairment','Mobility Impairment'), #
        any full
    ('Visual Impairment','Visual Impairment'),  #
        audio
    ('Hearing Impairment','Hearing Impairment'), #
        image
    ('Learning Disability','Learning Disability'), #
        any
    ('Autism Spectrum Disorder','Autism Spectrum
        Disorder'), #any
    ('Speech Impairment','Speech Impairment'), #any
    ('Intellectual Disability','Intellectual
        Disability'), #any
    ('None','None')
)
disable=models.CharField(max_length=100,choices=
    options,null=True)
# Profile_Image = models.ImageField(upload_to='users/
    profile_images', null=True, blank=True)
```

serializers.py

```
from rest_framework import serializers
from rest_framework.validators import UniqueValidator
from .models import *
from rest_framework_simplejwt.serializers import
    TokenObtainPairSerializer
import json
# from AdminUI.models import ProfileDB



class UserSerializers(serializers.ModelSerializer):
    email = serializers.EmailField(
        validators=[UniqueValidator(queryset=CustomUser.
            objects.all())]
    )
    password=serializers.CharField(write_only=True)
    class Meta:
    model = CustomUser
    fields = (
        'username',
        'email',
```

```
            'profession',
            'gender',
            'age',
            'password'
        )
        extra_kwargs = {
            'password': {'write_only': True}  # Ensure
                password isn't included in responses
        }

    def create(self, validated_data):
        return CustomUser.objects.create_user(**
            validated_data)



class ProfileSerializer(serializers.ModelSerializer):
    class Meta:
        model = ProfileDB
        fields = (
            'user',
            'Address',
            'State',
            'Pincode',
            'profession',
            'card',
            'qual',
            'disable',
            'Aadhar_Number',
            'Contact',

        )


class LoginSerializer(serializers.Serializer):
    username = serializers.CharField()
    password = serializers.CharField(write_only=True)


class PasswordChangeSerializer(serializers.Serializer):
    old_password = serializers.CharField(required=True)
    new_password = serializers.CharField(required=True)

from rest_framework_simplejwt.serializers import
    TokenObtainPairSerializer
```

```
class MyTokenObtainPairSerializer(
    TokenObtainPairSerializer):
     @classmethod
     def get_token(cls, user):
          token = super().get_token(user)

          # Add custom claims to the token
          token['user_id'] = user.id
          token['username'] = user.username
          token['email'] = user.email
          token['is_superuser'] = user.is_superuser
          token['age'] = user.age
          token['gender'] = user.gender
          token['profession'] = user.profession
          # Add more custom claims as needed

          return token

     def validate(self, attrs):
          data = super().validate(attrs)

          # Include additional user details in the response
          data['user_id'] = self.user.id
          data['username'] = self.user.username
          data['email'] = self.user.email
          data['is_superuser'] = self.user.is_superuser
          data['age'] = self.user.age
          data['gender'] = self.user.gender
          data['profession'] = self.user.profession
          # Add more user details as needed

          return data
```

tests.py

```
from django.test import TestCase

# Create your tests here.
```

urls.py

```
from django.urls import path
from AdminUI import views
from .views import  *
from django.conf import settings
from django.conf.urls.static import static
```

```
urlpatterns = [
    path('register/',views.register_request.as_view(),
        name="register"), # registration
    path('profile/',ProfileView.as_view(),name='pro'),#
        profikle get put
    path('changepassword/',ChangePasswordView.as_view(),
        name="changeps"),#changepassword url
    path('logins/',LoginView.as_view(),name='log')

] + static(settings.MEDIA_URL, document_root=settings.
    MEDIA_ROOT)
```

views.py

```
from rest_framework import permissions
from django.shortcuts import render,redirect
from rest_framework import status,mixins,generics
from rest_framework.response import Response
from rest_framework.views import APIView
from rest_framework.viewsets import ModelViewSet
from rest_framework_simplejwt import authentication
from rest_framework.renderers import TemplateHTMLRenderer
    ,JSONRenderer
from django.contrib.auth import authenticate
from rest_framework.authtoken.models import Token
from rest_framework_simplejwt.views import
    TokenObtainPairView
from django.contrib.auth import authenticate, login
from Scheme.models import *
from Scheme.serializers import *
from django.contrib.auth import update_session_auth_hash
from rest_framework.renderers import MultiPartRenderer
from rest_framework.parsers import MultiPartParser
from .models import *
from .serializers import *
from django.contrib.auth.hashers import check_password

from rest_framework_simplejwt.views import
    TokenObtainPairView
from rest_framework.response import Response


class MyTokenObtainPairView(TokenObtainPairView):
    serializer_class = MyTokenObtainPairSerializer

    def post(self, request, *args, **kwargs):
```

```
        response = super().post(request, *args, **kwargs)
        serializer = MyTokenObtainPairSerializer(data=
            request.data)
        serializer.is_valid(raise_exception=True)

        # Include serialized user data in the response
        response.data.update(serializer.validated_data)
        return response


class register_request(mixins.ListModelMixin,mixins.
    CreateModelMixin,generics.GenericAPIView):
    queryset=CustomUser.objects.all()
    serializer_class=UserSerializers

    # def get(self,request):
    #     return self.list(request)

    def post(self,request,*args,**kwargs):
        return self.create(request)


class LoginView(APIView):
    serializer_class = LoginSerializer

    def post(self, request):
        serializer = self.serializer_class(data=request.
            data)

        if serializer.is_valid():
            username = serializer.validated_data['
                username']
            password = serializer.validated_data['
                password']
            user = authenticate(request, username=
                username, password=password)
            user = CustomUser.objects.get(username=
                username)

            if user is not None and check_password(
                password, user.password):
                login(request,user)
                return Response({'message': 'Login
                    successful'}, status=status.
                    HTTP_200_OK)
```

```
            else:
                return Response({'message': 'Invalid
                    credentials'}, status=status.
                    HTTP_401_UNAUTHORIZED)

        return Response(serializer.errors, status=status.
            HTTP_400_BAD_REQUEST)
```

```
class ProfileView(APIView):
    # permission_classes = [permissions.IsAuthenticated]
    # authentication_classes = [authentication.
        JWTAuthentication]
    serializer_class=ProfileSerializer


    def get(self,req,*args,**kwargs):
        prr=req.user.id
        print(prr)
        try:

            pr=ProfileDB.objects.get(user=req.user.id)
            print(pr)
            dser=ProfileSerializer(pr)
            return Response(data=dser.data)
        except:
            return Response({"msg":"Profile is not
                created"},status=status.
                HTTP_400_BAD_REQUEST)
    def post(self,req,*args,**kwargs):
        user=req.user.id
        ser=ProfileSerializer(data=req.POST)
        if ser.is_valid():
            ser.save()
            return Response({"Msg":"Profile Added"},
                status=status.HTTP_201_CREATED)
        else:
            return Response({"Msg": ser.errors},
                status=status.HTTP_400_BAD_REQUEST)

    def put(self,req,*args,**kwargs):
        try:
            pr=ProfileDB.objects.get(user=req.user.id)
            print(req.user.id)
            ser=ProfileSerializer(data=req.data,instance=
                pr)
            if ser.is_valid():
```

```
            ser.save()
            return Response({"msg":"Updated"})
        else:
            return Response({"msg":ser.errors},status
                =status.HTTP_422_UNPROCESSABLE_ENTITY)
    except:
        return Response({"msg":"Invalid ID"},status=
            status.HTTP_400_BAD_REQUEST)


class ChangePasswordView(APIView):
    # permission_classes = [permissions.IsAuthenticated]

    def post(self, request):
        user = request.user
        serializer = PasswordChangeSerializer(data=
            request.data)

        if serializer.is_valid():
            old_password = serializer.data.get("
                old_password")
            new_password = serializer.data.get("
                new_password")

            if not user.check_password(old_password):
                return Response({"old_password": ["Wrong
                    password."]}, status=status.
                    HTTP_400_BAD_REQUEST)

            user.set_password(new_password)
            user.save()
            update_session_auth_hash(request, user)
            return Response({"message": "Password updated
                successfully."}, status=status.
                HTTP_200_OK)

        return Response(serializer.errors, status=status.
            HTTP_400_BAD_REQUEST)
```

asgi.py

```
"""
ASGI config for GetScheme project.

It exposes the ASGI callable as a module-level variable
    named ``application``.
```

```
For more information on this file, see
https://docs.djangoproject.com/en/5.0/howto/deployment/
    asgi/
"""

import os

from django.core.asgi import get_asgi_application

os.environ.setdefault('DJANGO_SETTINGS_MODULE', '
    GetScheme.settings')

application = get_asgi_application()
```

settings.py

```
"""
Django settings for GetScheme project.

Generated by 'django-admin startproject' using Django
    4.1.4.

For more information on this file, see
https://docs.djangoproject.com/en/4.1/topics/settings/

For the full list of settings and their values, see
https://docs.djangoproject.com/en/4.1/ref/settings/
"""

from pathlib import Path
import os.path
import crispy_forms

# Build paths inside the project like this: BASE_DIR / '
    subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent


# Quick-start development settings - unsuitable for
    production
# See https://docs.djangoproject.com/en/4.1/howto/
    deployment/checklist/

# SECURITY WARNING: keep the secret key used in
    production secret!
SECRET_KEY = 'django-insecure-!w0yrs!6_bkr!qcez740ze&q*rz
    &_syfidoc^@iid3url9*6+o'
```

```
# SECURITY WARNING: don't run with debug turned on in
    production!
DEBUG = True

ALLOWED_HOSTS = ['10.0.2.2']
# ALLOWED_HOSTS = []


# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'AdminUI',
    'Scheme',
    'crispy_forms',
    'rest_framework',
    # 'rest_framework.authtoken',
    'rest_framework_simplejwt',
    'corsheaders'
]

# CRISPY_TEMPLATE_PACK = 'bootstrap4'

MIDDLEWARE = [
    'corsheaders.middleware.CorsMiddleware',
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware
        ',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.
       AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware
        ',
    'django.middleware.clickjacking.
        XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'GetScheme.urls'
```

```
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.
            DjangoTemplates',
        'DIRS': [os.path.join(BASE_DIR, 'templates')],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug
                    ',
                'django.template.context_processors.
                    request',
                'django.contrib.auth.context_processors.
                    auth',
                'django.contrib.messages.
                    context_processors.messages',
            ],
        },
    },
]

WSGI_APPLICATION = 'GetScheme.wsgi.application'


# Database
# https://docs.djangoproject.com/en/4.1/ref/settings/#
    databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'GETSCHEME',
        'USER': 'root',
        'PASSWORD': 'root',
        'HOST':'localhost',
        'PORT':'3306',
    }
}


# Password validation
# https://docs.djangoproject.com/en/4.1/ref/settings/#
    auth-password-validators
```

```
AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.
            UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.
            MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.
            CommonPasswordValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.
            NumericPasswordValidator',
    },
]


# Internationalization
# https://docs.djangoproject.com/en/4.1/topics/i18n/

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'GMT'

USE_I18N = True

USE_TZ = True


# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/4.1/howto/static-
    files/

STATIC_URL = 'static/'
MEDIA_URL = '/media/'
MEDIA_ROOT = os.path.join(BASE_DIR,'media/')

# Default primary key field type
# https://docs.djangoproject.com/en/4.1/ref/settings/#
    default-auto-field

DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
```

```
REST_FRAMEWORK = {
    'DEFAULT_AUTHENTICATION_CLASSES': [
        'rest_framework_simplejwt.authentication.
            JWTAuthentication',
    ],
}

from datetime import timedelta


SIMPLE_JWT = {
    'ACCESS_TOKEN_LIFETIME': timedelta(hours=10),
    'SLIDING_TOKEN_REFRESH_LIFETIME': timedelta(days=1),
    'SLIDING_TOKEN_LIFETIME': timedelta(days=1),
    'SLIDING_TOKEN_REFRESH_LIFETIME': timedelta(days=30),
}

AUTH_USER_MODEL = 'AdminUI.CustomUser'

CORS_ALLOW_METHODS = (
    "DELETE",
    "GET",
    "OPTIONS",
    "PATCH",
    "POST",
    "PUT",
)

EMAIL_BACKEND = 'django.core.mail.backends.smtp.
    EmailBackend'
EMAIL_HOST = 'smtp.gmail.com'  # Replace with your SMTP
    host
EMAIL_PORT = 587  # Replace with your SMTP port
EMAIL_USE_TLS = True  # Use TLS for secure connection
EMAIL_HOST_USER = 'hibindixon123123@gmail.com'  # Replace
     with your email
EMAIL_HOST_PASSWORD = 'altujgkcglvsttop'  # Replace with
    your email password
```

urls.py

```
"""GetScheme URL Configuration

The `urlpatterns` list routes URLs to views. For more
```

```
    information please see:
     https://docs.djangoproject.com/en/4.1/topics/http/
        urls/
Examples:
Function views
    1. Add an import:  from my_app import views
    2. Add a URL to urlpatterns:  path('', views.home,
        name='home')
Class-based views
    1. Add an import:  from other_app.views import Home
    2. Add a URL to urlpatterns:  path('', Home.as_view()
        , name='home')
Including another URLconf
    1. Import the include() function: from django.urls
        import include, path
    2. Add a URL to urlpatterns:  path('blog/', include('
        blog.urls'))
"""
from django.contrib import admin
from django.contrib.staticfiles.urls import
    staticfiles_urlpatterns, static
from django.urls import path, include
from rest_framework_simplejwt.views import (
    TokenObtainPairView,
    TokenRefreshView,
)

import AdminUI.urls
import Scheme.urls
from . import settings
from  rest_framework.authtoken import views
from AdminUI.views import *
from rest_framework.decorators import api_view,
    permission_classes
from rest_framework.permissions import AllowAny
from rest_framework_simplejwt.tokens import RefreshToken

@api_view(['GET'])
@permission_classes([AllowAny])
def get_tokens_for_user(request):

    # find the user base in params
    user = CustomUser.objects.first()

    refresh = RefreshToken.for_user(user)
```

```
    return Response({
        'refresh': str(refresh),
        'access': str(refresh.access_token),
    })

urlpatterns = [
    path('admin/', admin.site.urls),
    path('AdminUI/', include("AdminUI.urls")),
    path('Scheme/', include(Scheme.urls)),
    path('login/',MyTokenObtainPairView.as_view(),name='
        login'),
    path('token/', TokenObtainPairView.as_view(), name='
        token_obtain_pair'),
    path('token/refresh/', TokenRefreshView.as_view(),
        name='token_refresh'),
]
urlpatterns += staticfiles_urlpatterns()
urlpatterns += static(settings.MEDIA_URL, document_root=
    settings.MEDIA_ROOT)
```

wsgi.py

```
"""
WSGI config for GetScheme project.

It exposes the WSGI callable as a module-level variable
    named ``application``.

For more information on this file, see
https://docs.djangoproject.com/en/5.0/howto/deployment/
    wsgi/
"""

import os

from django.core.wsgi import get_wsgi_application

os.environ.setdefault('DJANGO_SETTINGS_MODULE', '
    GetScheme.settings')

application = get_wsgi_application()
```

admin.py

```
from django.contrib import admin

from Scheme.models import SchemesDB
```

```
# Register your models here.
admin.site.register(SchemesDB)
```

apps.py

```
from django.apps import AppConfig


class SchemeConfig(AppConfig):
    default_auto_field = 'django.db.models.BigAutoField'
    name = 'Scheme'
```

models.py

```
  from django.db import models
from django.utils import timezone
# Create your models here.

class SchemesDB(models.Model):
    Scheme_Name = models.CharField(max_length=100,null=
        True)
    options=(
        ('Disabled','Disabled'),
        ('Farmer','Farmer'),
        ('Age','Age'),
        ('Student','Student'),
        ('Government Employ','Government Employ'),
        ('General','General'),
        ('Women','Women'),
        ('Teacher','Teacher'),
        ('Sports','Sports'),
        ('Military','Military')
)
type = models.CharField(max_length=100,null=True,
    choices=options)
start_age=models.IntegerField(null=True)
end_age=models.IntegerField(null=True)
Description = models.CharField(max_length=1000,null=
    True)
Link = models.CharField(max_length=1000,null=True)
timestamp=models.DateTimeField(auto_now_add=True,null
    =True)


def contains_age(self, age_to_check):
    return self.start_age <= age_to_check <= self.
        end_age
```

serializers.py

```
from django.contrib.auth.models import User
from rest_framework import serializers

from Scheme.models import SchemesDB


class SchemeSerializer(serializers.ModelSerializer):
    class Meta:
        model = SchemesDB
        fields = (
            'Scheme_Name',
            'type',
            'start_age',
            'end_age',
            'Description',
            'Link'
        )
```

tests.py

```
from django.test import TestCase

# Create your tests here.
```

urls.py

```
from django.urls import path
from .views import *


urlpatterns = [
    path('scheme/',SchemeView.as_view(),name='scheme'),
    path('new_elements/',NewElementsAPIView.as_view(),
        name='elements'),
    path('myscheme/',FilterAPIView.as_view(),name='my'),
    path('notification/',NotificationsAPIView.as_view(),
        name='ntv')

]
```

views.py

```
from django.shortcuts import render
from rest_framework.views import APIView
from rest_framework_simplejwt import authentication
from rest_framework import permissions
from .models import *
from .serializers import *
from rest_framework.response import Response
```

```
from rest_framework import status
from django.utils import timezone
from AdminUI.models import ProfileDB
from django.http import JsonResponse
from AdminUI.models import CustomUser
from django.core.mail import send_mail
from django.conf import settings
# Create your views here.

class SchemeView(APIView):
    # permission_classes = [permissions.IsAuthenticated]
    # authentication_classes = [authentication.
      JWTAuthentication]

    def get(self,req,*args,**kwargs):
        try:
            pr=SchemesDB.objects.all()
            dser=SchemeSerializer(pr,many=True)
            return Response(data=dser.data)
        except:
            return Response({"msg":"Invalid ID"},status=
                status.HTTP_400_BAD_REQUEST)
    def post(self,req):
            ser=SchemeSerializer(data=req.POST)
            if ser.is_valid():
scheme_instance = ser.save()
users = CustomUser.objects.all()  #
    Replace with your actual User model
subject = 'New Scheme Created'
message = f'A new scheme has been created
    : '  # Adjust based on your Scheme
    model fields
from_email = 'hibindixon123123@gmail.com'
for user in users:
    to_email = [user.email]
    send_mail(subject, message,
        from_email, to_email,
        fail_silently=False)


return Response({"Msg":"Scheme Added"},
    status=status.HTTP_201_CREATED)
            else:
                return Response({"Msg": ser.errors},
                    status=status.HTTP_400_BAD_REQUEST)
```

```
import datetime

class NewElementsAPIView(APIView):
    def get(self, request):
        since_date = datetime.datetime.now() - datetime.
            timedelta(days=10)
        # current_time = timezone.now()
        if since_date:
            new_elements = SchemesDB.objects.filter(
                timestamp__gt=since_date)
        else:
            new_elements = SchemesDB.objects.all()

        serialized_data = SchemeSerializer(new_elements,
            many=True)
        return Response(serialized_data.data)


class NotificationsAPIView(APIView):
    def get(self, request):
        since_date = datetime.datetime.now() - datetime.
            timedelta(hours=5,minutes=41)
        # current_time = timezone.now()
        new_elements = SchemesDB.objects.filter(
            timestamp__gt=since_date).order_by("-timestamp
            ")
        if new_elements:
            msg="New schemes added check it..."
            response={"message":msg}
            return JsonResponse(response)
        else :
            msg="No new schemes added"
            response={"message":msg}
            return JsonResponse(response)


        # serialized_data = SchemeSerializer(new_elements
            , many=True)
        # return Response(data=serialized_data.data)


from django.shortcuts import get_object_or_404
class FilterAPIView(APIView):
    def get(self, request):

        id=request.user.id
```

```
pr = get_object_or_404(ProfileDB, user=id)
print(pr.id)
user_profession = pr.profession if hasattr(pr, '
    profession') else None
user_age = request.user.age if hasattr(request.
    user, 'age') else None
print(user_profession)
print(request.user.profession)
schemes = SchemesDB.objects.filter(type=
    user_profession,start_age__lte=user_age,
    end_age__gte=user_age)
filtered_schemes = [scheme for scheme in schemes
    if scheme.contains_age(user_age)]
serialized_data = SchemeSerializer(
    filtered_schemes, many=True)
return Response(serialized_data.data)
```

manage.py

```
#!/usr/bin/env python
"""Django's command-line utility for administrative tasks
    ."""
import os
import sys


def main():
    """Run administrative tasks."""
    os.environ.setdefault('DJANGO_SETTINGS_MODULE', '
        GetScheme.settings')
    try:
        from django.core.management import
            execute_from_command_line
    except ImportError as exc:
        raise ImportError(
            "Couldn't import Django. Are you sure it's
                installed and "
            "available on your PYTHONPATH environment
                variable? Did you "
            "forget to activate a virtual environment?"
        ) from exc
    execute_from_command_line(sys.argv)


if __name__ == '__main__':
    main()
```

control.dart

```
import 'package:get/get.dart';
import 'package:schemeapp/service/service.dart';

class SchemeController extends GetxController {
  var list = [].obs;

  @override
  void onInit() {
    getScheme();
    super.onInit();
  }

  void getScheme() async {
    try {
      var data = await HttpScheme.fetchscheme();
      if (data != null) {
        list.value = data;
      }
    } catch (e) {
      print(e);
    }
  }
}
```

filter_scheme_controller.dart

```
import 'package:get/get.dart';
import 'package:schemeapp/service/filter_scheme_serrvice.
    dart';
import 'package:schemeapp/service/service.dart';

class FilterSchemeController extends GetxController {
  var flist = [].obs;

  @override
  void onInit() {
    getfilterScheme();
    super.onInit();
  }

  void getfilterScheme() async {
    try {
      var data = await HttpFilterScheme.filterscheme();
      if (data != null) {
        flist.value = data;
      }
    } catch (e) {
```

```
        print(e);
      }
    }
}
```

filterscheme.dart

```
// To parse this JSON data, do
//
//     final filterScheme = filterSchemeFromJson(
    jsonString);

import 'dart:convert';

List<FilterScheme> filterSchemeFromJson(String str) =>
    List<FilterScheme>.from(json.decode(str).map((x) =>
    FilterScheme.fromJson(x)));

String filterSchemeToJson(List<FilterScheme> data) =>
    json.encode(List<dynamic>.from(data.map((x) => x.
    toJson())));

class FilterScheme {
    String? schemeName;
    String? type;
    int? startAge;
    int? endAge;
    String? description;
    String? link;

    FilterScheme({
        this.schemeName,
        this.type,
        this.startAge,
        this.endAge,
        this.description,
        this.link,
    });

    factory FilterScheme.fromJson(Map<String, dynamic>
    json) => FilterScheme(
        schemeName: json["Scheme_Name"],
        type: json["type"],
        startAge: json["start_age"],
        endAge: json["end_age"],
        description: json["Description"],
        link: json["Link"],
```

```
);

Map<String, dynamic> toJson() => {
        "Scheme_Name": schemeName,
        "type": type,
        "start_age": startAge,
        "end_age": endAge,
        "Description": description,
        "Link": link,
    };
}
```

modell.dart

```
// To parse this JSON data, do
//
//     final schmeModel = schmeModelFromJson(jsonString);

import 'dart:convert';

List<SchmeModel> schmeModelFromJson(String str) => List<
    SchmeModel>.from(json.decode(str).map((x) =>
    SchmeModel.fromJson(x)));

String schmeModelToJson(List<SchmeModel> data) => json.
    encode(List<dynamic>.from(data.map((x) => x.toJson())
    ));

class SchmeModel {
    String? schemeName;
    String? schmeModelNew;
    String? description;
    String? link;

    SchmeModel({
        this.schemeName,
        this.schmeModelNew,
        this.description,
        this.link,
    });

    factory SchmeModel.fromJson(Map<String, dynamic> json
        ) => SchmeModel(
    schemeName: json["Scheme_Name"],
    schmeModelNew: json["new"],
    description: json["Description"],
    link: json["Link"],
```

```
);

Map<String, dynamic> toJson() => {
      "Scheme_Name": schemeName,
      "new": schmeModelNew,
      "Description": description,
      "Link": link,
    };
}
```

adminl.dart

```
import 'package:flutter/material.dart';
import 'package:url_launcher/url_launcher.dart';

void main() {
  runApp(MaterialApp(home: Admin()));
}

class Admin extends StatefulWidget {
  @override
  State<Admin> createState() => _AdminState();
}

class _AdminState extends State<Admin> {
  TextEditingController linkController =
    TextEditingController();

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text('Upload Page'),
    ),
    body: Padding(
      padding: const EdgeInsets.all(16.0),
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          TextField(
            controller: linkController,
            decoration: InputDecoration(labelText: '
                Enter Link'),
    ),
    SizedBox(height: 20),
    ElevatedButton(
      onPressed: () {
```

```
                String uploadedLink = linkController.text
                    ;
                _launchURL(uploadedLink);
              },
              child: Text('Upload Link'),
            ),
          ],
            ),
          ),
        );
    }

    _launchURL(String link) async {
      if (await canLaunch(link)) {
        await launch(link);
      } else {
        throw 'Could not launch $link';
      }
    }
}
```

edit_profile.dart

```
import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;

void main() {
  runApp(MyHomePage());
}

class MyHomePage extends StatefulWidget {
  @override
  _MyHomePageState createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {
  TextEditingController usernameController =
      TextEditingController();
  // TextEditingController emailController =
      TextEditingController();
  TextEditingController stateController =
      TextEditingController();
  TextEditingController pinController =
      TextEditingController();
  TextEditingController addressController =
      TextEditingController();
  TextEditingController contactController =
```

```
    TextEditingController();
TextEditingController aadharController =
    TextEditingController();

Future<void> postData() async {
  final url = 'http://10.0.2.2:8000/AdminUI/profile/';

final response = await http.post(
  Uri.parse(url),
  body: {
    'user': usernameController.text,
    'Address': addressController.text,
    'State': stateController.text,
    'Pincode': pinController.text,
    'Aadhar_Number': addressController.text,
    'Contact': contactController.text,
    'profession': aadharController.text,
    'card': aadharController.text,
    'qual': aadharController.text,
    'disable': aadharController.text,
  },
);

if (response.statusCode == 201) {
  print('Data posted successfully');
  } else {
    print('Failed to post data. Status code: ${response
        .statusCode}');
  }
}

String profesion = 'Farmer';
String card = 'APL';
String qualification = 'SSLC';
String disable = "None";

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text('HTTP Post Example'),
    ),
body:
      // Padding(
      //   padding: const EdgeInsets.all(16.0),
      //   child: Column(
```

```
//      mainAxisAlignment: MainAxisAlignment.
    center,
//      children: [
//        TextField(
//          controller: usernameController,
//          decoration: InputDecoration(
    labelText: 'Username'),
//        ),
//        TextField(
//          controller: emailController,
//          decoration: InputDecoration(
    labelText: 'Email'),
//        ),
//        TextField(
//          controller: stateController,
//          decoration: InputDecoration(
    labelText: 'State'),
//        ),
//        TextField(
//          controller: pinController,
//          decoration: InputDecoration(
    labelText: 'Pin'),
//        ),
//        TextField(
//          controller: addressController,
//          decoration: InputDecoration(
    labelText: 'Address'),
//        ),
//        TextField(
//          controller: contactController,
//          decoration: InputDecoration(
    labelText: 'Contact'),
//        ),
//        TextField(
//          controller: aadharController,
//          decoration: InputDecoration(
    labelText: 'Aadhar'),
//        ),
//        SizedBox(height: 20),
//        ElevatedButton(
//          onPressed: postData,
//          child: Text('Post Data'),
//        ),
//      ],
//    ),
// ),
```

```
    SingleChildScrollView(
  child: Padding(
    padding: const EdgeInsets.all(16.0),
    child: Column(
      children: [
        // username
  TextFormField(
    controller: usernameController,
    decoration: InputDecoration(labelText: '
        Username'),
  ),

  // Address
  TextFormField(
    controller: addressController,
    decoration: InputDecoration(labelText: '
        Address'),
  ),

  // state
  TextFormField(
    controller: stateController,
    decoration: InputDecoration(labelText: '
        stste'),
  ),

  // Pincod
  TextFormField(
    controller: pinController,
    decoration: InputDecoration(labelText: '
        Pincode'),
  ),

  // Adhar number
  TextFormField(
    controller: aadharController,
    decoration: InputDecoration(labelText: '
        Adhar number'),
  ),

  // Contact
  TextFormField(
    controller: contactController,
    decoration: InputDecoration(labelText: '
        Contact'),
```

```
),

SizedBox(height: 20.0),

// proffesion
DropdownButtonFormField<String>(
  value: profesion,
  onChanged: (String? newValue) {
    setState(() {
      profesion = newValue!;
    });
  },
  items: <String>[
    'Farmer',
    'Student',
    'Disability',
    'Government Employ'
  ].map<DropdownMenuItem<String>>((String
      value) {
    return DropdownMenuItem<String>(
      value: value,
      child: Text(value),
    );
  }).toList(),
  decoration: InputDecoration(labelText: '
      Profession'),
),
SizedBox(height: 20.0),

// card
DropdownButtonFormField<String>(
  value: card,
  onChanged: (String? newValue) {
    setState(() {
      card = newValue!;
    });
  },
  items: <String>[
    'APL',
    'BPL',
    'AAY',
  ].map<DropdownMenuItem<String>>((String
      value) {
    return DropdownMenuItem<String>(
      value: value,
      child: Text(value),
```

```
        );
      }).toList(),
      decoration: InputDecoration(labelText: '
          Card'),
    ),
    SizedBox(height: 20.0),

    // Qualifivcation
    DropdownButtonFormField<String>(
      value: qualification,
      onChanged: (String? newValue) {
        setState(() {
          qualification = newValue!;
        });
      },
      items: <String>['SSLC', 'PLUS TWO', 'UG',
          'PG']
          .map<DropdownMenuItem<String>>((
              String value) {
        return DropdownMenuItem<String>(
          value: value,
          child: Text(value),
        );
      }).toList(),
      decoration: InputDecoration(labelText: '
          Qualification'),
    ),
    SizedBox(height: 20.0),

    // Disable
    DropdownButtonFormField<String>(
      value: disable,
      onChanged: (String? newValue) {
        setState(() {
          disable = newValue!;
        });
      },
      items: <String>[
        "Mobility Impairment",
        "Visual Impairment",
        "Hearing Impairment",
        "Learning Disability",
        "Autism Spectrum Disorder",
        "Speech Impairment",
        "Intellectual Disability",
        "None",
```

```
].map<DropdownMenuItem<String>>((String
    value) {
  return DropdownMenuItem<String>(
    value: value,
    child: Text(value),
  );
}).toList(),
                    decoration: InputDecoration(labelText: '
                        Disable'),
                  ),
                  SizedBox(height: 20.0),

                  ElevatedButton(
                    onPressed: () {
                      //
                    },
                    child: Text('Save'),
                  ),
                ],
              ),
            ),
          ),
        ),
      );
  }
}
```

home.dart

```
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'package:schemeapp/controller/control.dart';
import 'package:schemeapp/model/model.dart';
import 'package:schemeapp/screens/user/loginpage.dart';
import 'package:schemeapp/views/Mysche.dart';
import 'package:schemeapp/views/uu.dart';
import 'package:url_launcher/url_launcher.dart';

class SchemePage extends StatelessWidget {
  final SchemeController schemeController = Get.put(
      SchemeController());

  @override
  Widget build(BuildContext context) {
    return SafeArea(
      child: Scaffold(
        drawer: Drawer(
          child: ListView(
```

```
padding: EdgeInsets.zero,
children: <Widget>[
  SizedBox(
    height: 10,
  ),
  DrawerHeader(
    decoration: BoxDecoration(
  image: DecorationImage(
    image: NetworkImage(
      "https://qph.cf2.quoracdn.net/main-
          qimg-89
          dc4d0e21c42c7f0778582ee45c7440-
          pjlq",
    ),
    fit: BoxFit.fill,
  ),
),
child: Text(
  'User Name', // Replace with the actual
      user name
  style: TextStyle(
    color: Colors.white,
    fontSize: 20,
  ),
),
),
ListTile(
  leading: Icon(Icons.star),
  title: Text('My Profile', style:
      TextStyle(fontSize: 20)),
  onTap: () {
    Navigator.of(context).push(
      // MaterialPageRoute(builder: (
          context) => EditProfile()),
      MaterialPageRoute(builder: (context)
          => Profile()),
    );
  },
),
ListTile(
  leading: Icon(Icons.star),
  title: Text('Scheme', style: TextStyle(
    fontSize: 20)),
onTap: () {
  Navigator.of(context).push(
    MaterialPageRoute(builder: (context)
```

```
                => FilterSchemePage()),
          );
        },
      ),
      ListTile(
        leading: Icon(Icons.star),
        title: Text('All Scheme', style:
            TextStyle(fontSize: 20)),
        onTap: () {
          Navigator.of(context).push(
            MaterialPageRoute(builder: (context)
                => SchemePage()),
          );
        },
      ),
      ListTile(
        leading: Icon(Icons.exit_to_app),
        title: Text('Logout', style: TextStyle(
            fontSize: 20)),
        onTap: () {
          Navigator.of(context).push(
            MaterialPageRoute(builder: (context)
                => LoginPage()),
          );
        },
          ),
        ],
      ),
    ),
    backgroundColor: Colors.green[100],
    appBar: AppBar(
      backgroundColor: Colors.green[100],
      centerTitle: true,
      title: Text(
        "My Scheme",
        style: TextStyle(
          fontSize: 30.0,
          fontWeight: FontWeight.bold,
          color: Colors.black,
        ),
      ),
    ),
    body: Obx(
      () => ListView.builder(
        itemCount: schemeController.list.length,
        itemBuilder: (context, index) {
```

```
SchmeModel scheme = schemeController.list[
    index];

return Card(
  child: Container(
    padding: EdgeInsets.all(10),
    child: Column(
children: [
  Text(
    '${scheme.schemeName}',
    textAlign: TextAlign.center,
    style: TextStyle(
      fontSize: 18,
      fontWeight: FontWeight.bold,
      color: Colors.green,
    ),
  ),
  SizedBox(height: 5),
  Container(
    child: Text('${scheme.description
        }'),
  ),
  GestureDetector(
    onTap: () {
      launchURL(scheme.link);
    },
    child: Text(
      '${scheme.link}',
      style: TextStyle(
        color: Colors.blue,
        decoration: TextDecoration.
            underline,
      ),
    ),
  ),
),
SizedBox(height: 5),
// Container(
//   width: 80,
//   height: 35,
//   child: Card(
//     shape:
    RoundedRectangleBorder(
//       borderRadius: BorderRadius
    .circular(20),
//     ),
//     shadowColor: const Color.
```

```
        fromARGB(255, 246, 16, 0),
//        child: Center(
//            child: Text('${scheme.
    schmeModelNew ?? ''}')),
//    ),
// ),
// SizedBox(height: 5),
                    ],
                  ),
                ),
              ),
            ),
          ),
        ),
      );
}

void launchURL(String? link) async {
  if (await canLaunch(link!)) {
    await launch(link,
        forceSafariVC: false, forceWebView: false,
            enableJavaScript: true);
  } else {
    throw 'Could not launch $link';
  }
}
```

loginpage.dart

```
import 'dart:convert';
import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;
import 'package:schemeapp/screens/user/Siginpage.dart';
import 'package:schemeapp/screens/user/home.dart';
import 'package:schemeapp/screens/user/homepage.dart';
import 'package:schemeapp/service/filter_scheme_serrvice.
    dart';
import 'package:schemeapp/views/scheme.dart';

class LoginPage extends StatefulWidget {
  @override
  _LoginPageState createState() => _LoginPageState();
}

class _LoginPageState extends State<LoginPage> {
```

```
final _usernameController = TextEditingController();
final _passwordController = TextEditingController();

void login() async {
  final response = await http.post(
    Uri.parse('http://10.0.2.2:8000/login/'),
    body: {
      'username': _usernameController.text,
      'password': _passwordController.text,
    },
  );

  if (response.statusCode == 200) {
    var responseMap = jsonDecode(response.body);
    String accessToken = responseMap['access'];

    print("accesssssstoken is :$accessToken");
    print(response.body);

    Navigator.push(
      context,
      MaterialPageRoute(
        builder: (context) => SchemePage(),
      ),
    );
  } else {
    ScaffoldMessenger.of(context).showSnackBar(
      SnackBar(
        content: Text('Invalid credentials. Please try
            again.'),
      ),
    );
  }
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    body: Container(
      decoration: BoxDecoration(
        image: DecorationImage(
          image: NetworkImage(
"https://images.unsplash.com/photo
    -1613125700782-8394bec3e89d?q=80&w
    =1887&auto=format&fit=crop&ixlib=rb
    -4.0.3&ixid=
```

```
                    M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA
                    %3D%3D") ,
            fit: BoxFit.cover,
        ),
    ),
    child: Padding(
      padding: const EdgeInsets.all(16.0),
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          Text(
            'Login',
            style: TextStyle(
              fontSize: 24,
              fontWeight: FontWeight.bold,
              color: Colors.white,
            ),
          ),
          TextField(
            controller: _usernameController,
            decoration: InputDecoration(labelText: '
                Username'),
          ),
          SizedBox(height: 16),
          TextField(
            controller: _passwordController,
            obscureText: true,
            decoration: InputDecoration(labelText: '
                Password'),
    ),
    SizedBox(height: 24),
    ElevatedButton(
      onPressed: login,
      child: Text('Login'),
    ),
    TextButton(
      onPressed: () {
        Navigator.of(context).push(
            MaterialPageRoute(
          builder: (context) => Siginup(),
        ));
      },
      child: Text(
        "Don't have an account? Sign in",
        style: TextStyle(color: Colors.white,
            fontSize: 15),
```

```
                    ) ,
                ) ,
            ] ,
        ) ,
    ) ,
) ,
) ;
  }
}
```

siginpage.dart

```
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'package:http/http.dart' as http;
import 'dart:convert';

import 'package:schemeapp/screens/user/loginpage.dart';

void main() {
  runApp(MaterialApp(
    home: Siginup(),
  ));
}

class Siginup extends StatefulWidget {
  @override
  _MyHomePageState createState() => _MyHomePageState();
}

class _MyHomePageState extends State<Siginup> {
  final TextEditingController usernameController =
      TextEditingController();
  final TextEditingController emailController =
      TextEditingController();
  final TextEditingController passwordController =
      TextEditingController();
  final TextEditingController jobController =
      TextEditingController();
  final TextEditingController ageController =
      TextEditingController();
  final TextEditingController genderController =
      TextEditingController();

  List<String> jobList = ['Farmer', 'Student', 'Disabled
      ', 'Other'];
  // List<String> ageList = ['18-25', '26-35', '36-50',
```

```
    '51+'];
List<String> genderlist = ["Male", "Female"];

Future<void> postData() async {
  final String url =
      'http://10.0.2.2:8000/AdminUI/register/'; //
         Replace with your API endpoint

  // final Map<String, String> data = {
  //   'username': usernameController.text,
  //   'email': emailController.text,
  //   'password': passwordController.text,
  //   'profession': jobController.text,
  //   'age': ageController.text,
  // };

  final response = await http.post(
    Uri.parse(url),
    headers: {
      'Content-Type':
          'application/x-www-form-urlencoded', //
             Adjust the content type if needed
    },
    body: {
      'username': usernameController.text.trim(),
      'email': emailController.text.trim(),
      'password': passwordController.text.trim(),
      'age': ageController.text.trim(),
      'profession': jobController.text.trim(),
      'gender': genderController.text.trim()
    },
  );
  if (response.statusCode == 201) {
    Navigator.push(
        context,
        MaterialPageRoute(
          builder: (context) => LoginPage(),
        ));
    print("success");
  } else {
    print(response.statusCode);
    print("Error");
  }
}

@override
```

```
Widget build(BuildContext context) {
  return SafeArea(
    child: Scaffold(
      body: Container(
        height: double.infinity,
        decoration: BoxDecoration(
          image: DecorationImage(
            image: NetworkImage(
              "https://images.unsplash.com/photo
                  -1613125700782-8394bec3e89d?q=80&w
                  =1887&auto=format&fit=crop&ixlib=rb
                  -4.0.3&ixid=
                  M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA
                  %3D%3D",
            ),
            fit: BoxFit.fill,
          ),
        ),
        child: Padding(
          padding: const EdgeInsets.symmetric(vertical:
              10, horizontal: 10),
          child: SingleChildScrollView(
            scrollDirection: Axis.vertical,
            child: Column(
              mainAxisAlignment: MainAxisAlignment.
                spaceBetween,
              children: [
                Container(
                  alignment: Alignment.center,
                  child: Text(
                    "Sign in",
                    style: TextStyle(
                        fontSize: 30,
                        color: Colors.white,
                        fontWeight: FontWeight.bold),
                  ),
                  height: 50,
                  color: Colors.transparent,
                  width: double.maxFinite,
                ),
                SizedBox(
                  height: 100,
                ),
                TextField(
                  controller: usernameController,
                  decoration: InputDecoration(
```

```
            labelText: 'Username',
            labelStyle: TextStyle(color:
                Colors.black)),
    ),
    SizedBox(height: 16),
    TextField(
      controller: emailController,
      decoration: InputDecoration(
          labelText: 'Email',
          labelStyle: TextStyle(color:
              Colors.black)),
    ),
    SizedBox(height: 16),
    TextField(
      controller: passwordController,
      decoration: InputDecoration(
          labelText: 'Password',
          labelStyle: TextStyle(color:
              Colors.black)),
      obscureText: true,
    ),
    SizedBox(height: 16),
    DropdownTextField(
      controller: jobController,
      items: jobList,
      hintText: 'Select Job',
    ),
    SizedBox(height: 16),
    DropdownTextField(
      controller: genderController,
      items: genderlist,
      hintText: 'Select gender',
    ),
    SizedBox(height: 16),
    TextField(
      controller: ageController,
      decoration: InputDecoration(
          labelText: 'Age',
          labelStyle: TextStyle(color:
              Colors.black)),
      obscureText: false,
    ),
    SizedBox(height: 32),
    ElevatedButton(
      onPressed: () {
        postData();
```

```
                    },
                    child: Text('Post Data'),
                  ),
                ],
              ),
            ),
          ),
        ),
      );
    }
  }

  class DropdownTextField extends StatelessWidget {
    final TextEditingController controller;
    final List<String> items;
    final String hintText;

    const DropdownTextField({
      Key? key,
      required this.controller,
      required this.items,
      required this.hintText,
    }) : super(key: key);

    @override
    Widget build(BuildContext context) {
      return TextField(
        controller: controller,
        decoration: InputDecoration(
          labelText: hintText,
          suffixIcon: PopupMenuButton<String>(
            icon: const Icon(Icons.arrow_drop_down),
            onSelected: (value) {
              controller.text = value;
            },
            itemBuilder: (BuildContext context) {
              return items.map<PopupMenuItem<String>>((
                  String value) {
                return PopupMenuItem<String>(
                  value: value,
                  child: Text(value),
                );
              }).toList();
            },
          ),
```

```
        ),
      );
    }
}
```

splash.dart

```
import 'dart:async';
import 'package:flutter/material.dart';
import 'package:schemeapp/screens/user/loginpage.dart';

class SplashPage extends StatefulWidget {
  @override
  _SplashPageState createState() => _SplashPageState();
}

class _SplashPageState extends State<SplashPage> {
  @override
  void initState() {
    super.initState();

    Timer(
      Duration(seconds: 6),
      () => Navigator.pushReplacement(
        context,
        MaterialPageRoute(builder: (context) => LoginPage
            ()),
      ),
    );
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    body: Center(
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
         Image.network("https://www.logopeople.in/wp-
             content/uploads/2013/01/government-of-india
             .jpg")

        ],
      ),
    ),
  );
}
```

```
}

void main() {
  runApp(MaterialApp(
    home: SplashPage(),
    debugShowCheckedModeBanner: false,
  ));
}
```

adminpage.dart

```
import 'package:flutter/material.dart';

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Admin(),
    );
  }
}

class Admin extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Form Page'),
      ),
      body: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.stretch,
          children: [
            TextFormField(
              decoration: InputDecoration(labelText: '
                  Main Heading'),
            ),
            SizedBox(height: 16.0),
            TextFormField(
              decoration: InputDecoration(labelText: 'Age
                  '),
              keyboardType: TextInputType.number,
```

```
),
SizedBox(height: 16.0),
TextFormField(
  decoration: InputDecoration(labelText: '
      Description'),
),
SizedBox(height: 16.0),
                TextFormField(
                  decoration: InputDecoration(labelText: '
                      Category'),
                ),
                SizedBox(height: 32.0),
                ElevatedButton(
                  onPressed: () {

                  },
                  child: Text('Submit'),
                ),
              ],
            ),
          ),
        );
    }
}
```

filter_scheme_servicce.dart

```
import 'package:http/http.dart' as https;
import 'package:schemeapp/model/filterscheme.dart';
import 'package:schemeapp/screens/user/loginpage.dart';

// class HttpFilterScheme {
//   static String authToken =
//       ''; // Static variable to store the
    authentication token

//   static void setAuthToken(String token) {
//     authToken = token;
//   }

class HttpFilterScheme {
  static Future<dynamic> filterscheme() async {
    // var token = LoginPage.authToken;
    var response = await https.get(
      Uri.parse("http://10.0.2.2:8000/Scheme/myscheme/"),
      headers: {
        "Authorization":
```

```
"Bearer "eyJhbGciOiJI" "UzI1NiIsInR5cCI6Ik" "
    pXVCJ9.eyJ0b2tlbl9" "0eXBlIjoiYWNjZXNz" "
    IiwiZXhwIjoxN" "zA1MDc2Njc" "2LCJpYXQiOjE"
     "3MDUwNDA2N" "zYsImp0aSI6ImE1" "
    MTYxMDY4MTEz" "YzRkMzZiN2E" "
    zMTRhYWE5Y2NlZD" "VkIiwidXNlcl" "9
    pZCI6MSwidX" "Nlcm5hbWUiO" "
        iJhZG1pbiIsInBhc3" "N3b3JkIjoic" "
        GJrZGYyX3NoYTI" "1NiQ2MDAwM" "
        DAkQ2JMZUowam" "NROGxhVWJET1lu" "
        dHdlTSR0aEk4Y" "k9MZzNKWGw3" "
        Q0Z5eDZJV1JBVUJ" "ydm8rZHRPRlUvN" "
        y9GaytHRnE0PSIsIm" "F1dGgiOnR" "ydWV9.99
        hg34" "HC6zTC9nOu1q2_" "9-
        Vh4S6_J36cWilDV28AFHs"
    },
  );
  if (response.statusCode == 200) {
    var data = response.body;
    print(response.body);
    return filterSchemeFromJson(data);
  }
 }
}
```

service.dart

```
import 'package:http/http.dart' as https;
import 'package:schemeapp/model/model.dart';

class HttpScheme {
  static Future<dynamic> fetchscheme() async {
    var response = await https
        .get(Uri.parse("http://10.0.2.2:8000/Scheme/
            scheme/?format=json"),
        // headers: {"auth":"Bearer rtdyfgujoikl"}
        );
    if (response.statusCode == 200) {
      var data = response.body;
      print(response.body);
      return schmeModelFromJson(data);
    }
  }
}
```

Mysche.dart

```
import 'package:flutter/material.dart';
```

```
import 'package:get/get.dart';
import 'package:schemeapp/controller/
    filter_scheme_controller.dart';
import 'package:schemeapp/model/filterscheme.dart';
import 'package:url_launcher/url_launcher.dart';

class FilterSchemePage extends StatelessWidget {
  final FilterSchemeController filterSchemeController =
      Get.put(FilterSchemeController());

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.green[100],
      appBar: AppBar(
        backgroundColor: Colors.green[100],
        centerTitle: true,
        title: Text(
          "Filter Scheme",
          style: TextStyle(
            fontSize: 30.0,
            fontWeight: FontWeight.bold,
            color: Colors.black,
          ),
        ),
      ),
      body: Obx(
        () => ListView.builder(
          itemCount: filterSchemeController.flist.length,
          itemBuilder: (context, index) {
            FilterScheme filterScheme =
                filterSchemeController.flist[index];

            return Padding(
              padding: const EdgeInsets.symmetric(
                  vertical: 20),
              child: Card(
                child: Container(
                  // height: 70,
                  alignment: Alignment.center,
                  padding: EdgeInsets.all(10),
                  child: Column(
                    crossAxisAlignment:
                        CrossAxisAlignment.start,
                    mainAxisAlignment: MainAxisAlignment.
                        center,
```

```
children: [
  Text(
    '${filterScheme.schemeName}',
    textAlign: TextAlign.left,
    style: TextStyle(
      fontSize: 25,
      letterSpacing: 1.1,
      fontWeight: FontWeight.bold,
      color: Colors.black,
    ),
),
SizedBox(height: 10),
Text(
  '${filterScheme.description}',
  textAlign: TextAlign.left,
  style: TextStyle(
    fontSize: 18,
    fontWeight: FontWeight.bold,
    color: Colors.grey[800],
  ),
),
SizedBox(height: 10),
InkWell(
  onTap: () => launchURL(
      filterScheme.link),
  child: Text(
    '${filterScheme.link}',
    textAlign: TextAlign.center,
    style: TextStyle(
      fontSize: 18,
      fontWeight: FontWeight.bold,
      color: Colors.blue,
    ),
  ),
),
SizedBox(height: 5),
// Container(
//   child: Text('${scheme.
    description}'),
// ),
// GestureDetector(
//   onTap: () {
//     launchURL(scheme.link);
//   },
//   child: Text(
//     '${scheme.link}',
```

```
//       style: TextStyle(
//         color: Colors.blue,
//         decoration: TextDecoration
    .underline,
//       ),
//     ),
// ),
// SizedBox(height: 5),
// Container(
//   width: 80,
//   height: 35,
//   child: Card(
//     shape:
    RoundedRectangleBorder(
//       borderRadius: BorderRadius
    .circular(20),
//       ),
//       shadowColor: const Color.
    fromARGB(255, 246, 16, 0),
//       child: Center(child: Text('$
    {scheme.schmeModelNew}')),
//   ),
// ),

SizedBox(height: 5),
                    ],
                  ),
                ),
              );
            },
          ),
        ),
      ),
    );
}

void launchURL(String? link) async {
  if (await canLaunch(link!)) {
    await launch(link,
        forceSafariVC: false, forceWebView: false,
            enableJavaScript: true);
  } else {
    throw 'Could not launch $link';
  }
}
```

uu.dart

```
import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;

import 'package:schemeapp/screens/user/loginpage.dart';

void main() {
  runApp(MaterialApp(
    home: Profile(),
  ));
}

class Profile extends StatefulWidget {
  @override
  _ProfilePageState createState() => _ProfilePageState();
}

class _ProfilePageState extends State<Profile> {
  final TextEditingController usernameController =
      TextEditingController();
  final TextEditingController addressController =
      TextEditingController();
  final TextEditingController stateController =
      TextEditingController();
  final TextEditingController pinController =
      TextEditingController();
  final TextEditingController aadharController =
      TextEditingController();
  final TextEditingController contactController =
      TextEditingController();
  final TextEditingController professionController =
      TextEditingController();
  final TextEditingController cardController =
      TextEditingController();
  final TextEditingController qualController =
      TextEditingController();
  final TextEditingController disableController =
      TextEditingController();

  List<String> profession = ['Farmer', 'Student', '
      Disabled', 'Other'];
  // List<String> ageList = ['18-25', '26-35', '36-50',
      '51+'];
  List<String> card = ["Apl", "Bpl", "Aay"];
  List<String> qual = ["sslc", "Plus tow", "ug", "pg"];
```

```
List<String> disable = [
  "Mobility impairment",
  "visual impairment",
  "Hearing impairment",
  "Learing Disabilty",
  "Autism Spectrum Disoder",
  "Speach Imaariment",
  "Inrellectual Disability"
];

Future<void> postData() async {
  final String url =
      'http://10.0.2.2:8000/AdminUI/profile/'; //
         Replace with your API endpoint

  // final Map<String, String> data = {
  //   'username': usernameController.text,
  //   'email': emailController.text,
  //   'password': passwordController.text,
  //   'profession': jobController.text,
  //   'age': ageController.text,
  // };

  final response = await http.post(
    Uri.parse(url),
headers: {
  'Content-Type':
      'application/x-www-form-urlencoded', //
         Adjust the content type if needed
},
body: {
  // 'username': usernameController.text.trim(),
  // 'email': emailController.text.trim(),
  // 'password': passwordController.text.trim(),
  // 'age': ageController.text.trim(),
  // 'profession': jobController.text.trim(),
  // 'gender': genderController.text.trim()

  'user': usernameController.text.trim(),
  'Address': addressController.text.trim(),
  'State': stateController.text.trim(),
  'Pincode': pinController.text.trim(),
  'Aadhar_Number': aadharController.text.trim(),
  'Contact': contactController.text.trim(),
  'profession': professionController.text.trim(),
  'card': cardController.text.trim(),
```

```
        'qual': qualController.text.trim(),
        'disable': disableController.text.trim(),
      },
    );
    if (response.statusCode == 201) {
      Navigator.push(
          context,
          MaterialPageRoute(
            builder: (context) => LoginPage(),
          ));
      print("success");
    } else {
      print(response.statusCode);
      print("Error");
    }
  }

  @override
  Widget build(BuildContext context) {
    return SafeArea(
      child: Scaffold(
        body: Container(
          height: double.infinity,
          decoration: BoxDecoration(),
          child: SingleChildScrollView(
            child: Padding(
              padding: const EdgeInsets.symmetric(
                  horizontal: 10),
              child: SingleChildScrollView(
                scrollDirection: Axis.vertical,
                child: Column(
                  mainAxisAlignment: MainAxisAlignment.
                      spaceBetween,
                  children: [
                    Container(
                      alignment: Alignment.center,
                      child: Text(
                        "Profile",
                        style: TextStyle(
                            fontSize: 30,
                            color: Colors.black,
                            fontWeight: FontWeight.bold),
                      ),
                      height: 50,
                      color: Colors.transparent,
                      width: double.maxFinite,
```

```
),
// SizedBox(
//   height: 100,
// ),
TextField(
  controller: usernameController,
  decoration: InputDecoration(
      labelText: 'Username',
      labelStyle: TextStyle(color:
          Colors.black)),
),
SizedBox(height: 16),
TextField(
  controller: addressController,
  decoration: InputDecoration(
      labelText: 'address',
      labelStyle: TextStyle(color:
          Colors.black)),
),
SizedBox(height: 16),
TextField(
  controller: stateController,
  decoration: InputDecoration(
      labelText: 'state',
      labelStyle: TextStyle(color:
          Colors.black)),
  obscureText: true,
),
SizedBox(height: 16),
TextField(
  controller: pinController,
  decoration: InputDecoration(
      labelText: 'pincode',
      labelStyle: TextStyle(color:
          Colors.black)),
),
SizedBox(height: 16),
TextField(
  controller: aadharController,
  decoration: InputDecoration(
      labelText: 'aadhar no',
      labelStyle: TextStyle(color:
          Colors.black)),
),
SizedBox(height: 16),
TextField(
```

```
            controller: professionController,
            decoration: InputDecoration(
                labelText: 'job',
                labelStyle: TextStyle(color:
                    Colors.black)),
        ),
        SizedBox(height: 16),
        DropdownTextField(
          controller: cardController,
          items: card,
          hintText: 'Ration card',
        ),
        SizedBox(height: 16),
        TextField(
          controller: qualController,
          decoration: InputDecoration(
              labelText: 'qualificcation',
              labelStyle: TextStyle(color:
                  Colors.black)),
        ),
        SizedBox(height: 16),
        TextField(
          controller: disableController,
          decoration: InputDecoration(
              labelText: 'category',
              labelStyle: TextStyle(color:
                  Colors.black)),
        ),
        SizedBox(height: 16),
        TextField(
          controller: contactController,
          decoration: InputDecoration(
              labelText: 'contact',
              labelStyle: TextStyle(color:
                  Colors.black)),
          obscureText: false,
        ),
        SizedBox(height: 32),
        ElevatedButton(
          onPressed: () {
            postData();
          },
            child: Text('Post Data'),
          ),
        ],
      ),
```

```
                    ) ,
                ) ,
            ) ,
        ) ,
    ) ,
  ) ;
}
```

```
class DropdownTextField extends StatelessWidget {
  final TextEditingController controller;
  final List<String> items;
  final String hintText;

  const DropdownTextField({
    Key? key,
    required this.controller,
    required this.items,
    required this.hintText,
  }) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return TextField(
      controller: controller,
      decoration: InputDecoration(
        labelText: hintText,
        suffixIcon: PopupMenuButton<String>(
          icon: const Icon(Icons.arrow_drop_down),
          onSelected: (value) {
            controller.text = value;
          },
          itemBuilder: (BuildContext context) {
            return items.map<PopupMenuItem<String>>((
                String value) {
              return PopupMenuItem<String>(
                value: value,
                child: Text(value),
              );
            }).toList();
          },
        ),
      );
  }
}}
```

main.dart

```
import 'package:flutter/material.dart';
import 'package:schemeapp/screens/user/Siginpage.dart';
import 'package:schemeapp/screens/user/admin.dart';
import 'package:schemeapp/screens/user/homepage.dart';
import 'package:schemeapp/screens/user/loginpage.dart';
import 'package:schemeapp/screens/user/splash.dart';
import 'package:schemeapp/views/Mysche.dart';

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: SplashPage(),
    );
  }
}
```

FOGGY ROAD ALERT SYSTEM

PROJECT REPORT

Submitted By

KRISHNAPRIYA K

Reg. No. CCAVBCA041

For the award of the Degree of

Bachelor of Computer Application (BCA)

(University of Calicut)

under the guidance of

Ms. Sharon Joy C

Assistant Professor

BACHELOR OF COMPUTER APPLICATION
DEPARTMENT OF COMPUTER SCIENCE
CHRIST COLLEGE(AUTONOMOUS)
IRINJALAKUDA, KERALA
INDIA

2021-2024

DEPARTMENT OF COMPUTER SCIENCE

CHRIST COLLEGE (AUTONOMOUS)

IRINJALAKUDA

CERTIFICATE

This is to certify that the project report entitled ***"Foggy Road Alert System"*** *is a bonafide record of the project work done by* ***Krishnapriya K*** *in partial fulfillment of the requirement for the sixth semester of* ***Bachelor of Computer Application*** *in Department of Computer Science of* ***CHRIST COLLEGE (AUTONOMOUS) IRINJALAKUDA***

.

Ms. Sharon Joy C
Assistant Professor,CS
Internal Guide

Ms. Sini Thomas
Head of the Department
Computer Science

EXTERNAL EXAMINER **INTERNAL EXAMINER**

DECLARATION

We hereby declare that this project work **"FOGGY ROAD ALERT SYSTEM"** submitted to Christ College (Autonomous) Irinjalakuda, affiliated to Calicut University in partial fulfillment of the requirement for the award of the Bachelor of Computer Application, is a record of original work done by us, under the guidance of Ms. SHARON JOY C, Department of Computer Science.

Place: Irinjalakuda KRISHNAPRIYA K

ACKNOWLEDGEMENT

First and foremost we like to thank Lord almighty for his providence and for being the guiding light throughout the project. We wish to express my sincere gratitude to our beloved Department head for giving me all the facilities for our project. We take this opportunity to express my gratitude to the class teacher Ms. SOUMYA P.S and head of the department Ms. SINI THOMAS who has been supported us throughout the course of this project. We are thankful for her aspiring guidance and valuable advice during the project work. We express my sincere thanks to my project guide Ms. SHARON JOY C for supporting and guiding throughout the project.We would take this opportunity to specially thank all other faculty members for their constant and continuous motivation. Finally we would like to thank my family and friends for giving valuable advice and moral support throughout our project.

ABSTRACT

FOGGY ROAD ALERT SYSTEM is an innovative application of computer vision and deep learning technologies, aims to significantly enhance road safety by providing real-time visibility information during adverse weather conditions. Leveraging Python, OpenCV, TensorFlow, and Keras, the system employs a Convolutional Neural Network (CNN)-based fog detection model trained on a diverse dataset to analyse images captured by strategically placed cameras. Real-time monitoring ensures prompt responses to changing fog conditions, and an alerting mechanism notifies authorities when intensity thresholds are exceeded. The user-friendly interface allows for visualizing real-time fog data and historical trends. The system's successful deployment in fog-prone areas demonstrates its potential to reduce accidents and improve traffic management. Recommendations for future enhancements include exploring advanced sensors and dynamic threshold adjustments, further solidifying its role in proactive road safety measures. Additionally, the system seamlessly integrates with Django, offering features such as real-time fog identification, continuous monitoring, user authentication, adaptability for model retraining, and scalable deployment. The comprehensive documentation ensures ease of use, fostering continuous improvement through advancements in deep learning techniques and user feedback. This integration harmonizes cutting-edge capabilities and an intuitive web application, effectively addressing the challenges posed by foggy environmental conditions.

Contents

Chapter 1

1 Introduction

Fog is a major cause of road accidents worldwide. To improve road safety and traffic management, we propose a foggy road alert system that uses a web application, a fog detection model, and an email notification system. The web application allows users to upload videos of foggy roads and view the results of fog detection and alert generation. The fog detection model uses deep learning to classify each video frame as foggy or non-foggy, and calculate the average fog level. The email notification system sends an alert to the nearest traffic police station if the fog level is high.

1.1 Overview

The FOGGY ROAD ALERT SYSTEM aims to develop a web application that warns authorities about foggy roads using video analysis and email notification. The web application allows users to upload videos of foggy roads and view the fog detection results. The fog detection model uses deep learning to classify each video frame as foggy or non-foggy, and calculate the average fog level. The email notification system sends an alert to the nearest traffic police station if the fog level is high.It is a User friendly application.

Chapter 2

2 System Analysis

2.1 Purpose

The main purpose of this application owned by Krishnapriya K ,Aljo Poulose and Sreejishna K is to warn authorities about foggy roads using video analysis and email notification to prevent accidents and road blocks.

2.1.1 Existing System

The existing systems are using sensors placed on the roads.These sensors detect the fog and generates the warning messages to authorities.Limitationss of this sytems are sensosrs are costly,requires periodic maintenance,less accuracy

2.1.2 Proposed System

The proposed system is a web application which allows users to upload videos of foggy roads and view the fog detection results. The fog detection model uses deep learning to classify each video frame as foggy or non-foggy, and calculate the average fog level. The email notification system sends an alert to the nearest traffic police station's email address if the fog level is high. The proposed system aims to improve road safety and traffic management by providing timely and accurate information about the fog conditions on the roads.

2.2 Problem definition

To understand the problem and the needs before developing it, we are designing a foggy road alert system to detect fog on roads and generate warning messages for the traffic police.

2.3 FEASIBILITY STUDY

After the problem is clearly understood and solutions proposed, the next step is to conduct the feasibility study. Feasibility study is defined as evaluation or analysis of the potential impact of the proposed project or program. The objective is to determine whether the proposed system is feasible. There are three aspects of feasibility study to which the proposed system is subjected as discussed below.

2.3.1 Technical Feasibility

Technical feasibility assess whether the current technical resources are sufficient for the new system. We can upgrade the level of technology for supporting our

website. We check whether the proposed system can be implemented in the present system without supporting the existing hardware.

2.3.2 Economical Feasibility

Economic feasibility determines whether the time and money are available to develop the system. There is no additional hardware used to develop the site, it is inexpensive to build.

2.3.3 Operational Feasibility

Operational feasibility determines if the human resources are available to operate the system once it has been installed. No extra training is needed to use this system. Anyone who has knowledge in internet in English language can easily use the system. The resources that are required to implement or install are already available with the organization.

Chapter 3

3 Software Requirement Specification

3.1 Purpose

The purpose of this document is to outline the software requirements for the Foggy Road Alert System. It provides a comprehensive overview of the functionalities, constraints, and interactions necessary for the development of the system. This document serves as a blueprint for designing and implementing a robust solution aimed at enhancing road safety during foggy conditions. It delineates the system's capabilities to detect fog, generate timely alerts to traffic authorities.This document aims to facilitate the creation of an efficient and effective foggy road alert system that mitigates risks and improves overall road safety.

3.2 Scope

The Foggy Road Alert System aims to detect foggy conditions in real-time and provide timely alerts to drivers and authorities, enhancing road safety during low visibility conditions.

3.3 Overall Description

This section provides an overview of our Foggy Road Alert System. The project entails detecting the fog percentage in uploaded videos, triggering a warning email to the nearest traffic police authorities if the percentage surpasses a predefined threshold. Upon receiving the alert, authorities can take necessary measures to manage traffic on foggy roads, thereby reducing accidents associated with fog in various locations.The main aim of our project is to ensure the safety of people travelling through these foggy roads.

3.3.1 Product Perspective

The Foggy Road Alert System operates within the broader context of road safety and transportation management systems. From a product perspective, it serves as a critical component in enhancing road safety by specifically addressing the challenges posed by foggy conditions.The system is a standalone applocation that integrates CNN-based image processing for fog detection.It interacts with users through a Django based web interface

3.3.2 Product Functionality

The Foggy Road Alert System is equipped with several key functionalities aimed at bolstering road safety during foggy conditions. Leveraging sophisticated algorithms, the system automates the process of fog detection in uploaded videos,

ensuring prompt alerts when hazardous conditions arise. Robust access controls are implemented to guarantee that only authorized users can access the fog detection features, thereby enhancing system security. Powered by Django, the system's web application provides users with an intuitive interface, facilitating seamless interaction and ease of use. Additionally, the system delivers real-time alerts to users and traffic authorities, enabling swift responses to hazardous foggy conditions and ultimately enhancing overall road safety.

3.3.3 Users and Characteristics

The primary users of the system are traffic police authorities responsible for monitoring road conditions and managing traffic flow. They utilize the system to receive timely fog alerts and coordinate responses to ensure road safety.Secondary users include drivers who rely on the system's alerts and recommendations to navigate safely during foggy conditions.The characteristics include technical proficiency,time sensitivty,reliability and accesibilty.Then there are administators who oversee the overall functioning and maintenance of the system.Administrators can monitor and manage the generation and dissemination of fog alerts.

3.4 Specific Requirements

3.4.1 Hardware Requirements

- System: IBM-Compatible PC
- Processor: Intel Core i3
- Speed: Above 1GHz
- RAM capacity: 4 GB
- Hard Dsk drive: 500 GB

3.4.2 Software Requirements

- Front End : HTML,CSS,Bootstrap
- Back End : Python,Django
- IDE : PyCharm Community Edition 2023.1.3

3.5 Functional Requirements

It contains three main modules.

- 1.Fog Detection
- 2.Alert Generation
- 3.User Interface

Fog Detection

This part detects fog in videos or camera feeds.It uses algorithms to analyze images and determine fog density. This module is responsible for the core functionality of detecting foggy conditions.

Alert Generation

This module handles the generation and dissemination of fog alerts to traffic authorities.Develop algorithms for generating fog alerts based on detected fog density and predefined thresholds.

User Interface

This module encompasses the web interface and user interactions with the Foggy Road Alert System.Design and develop user-friendly interfaces for interacting with fog detection features and alerts.

3.6 Non Functional Requirements

Non-functional requirements define the overall qualities or attributes of the resulting system. Non-functional system place restrictions on the product being developed, the developed process, and specify external constraints that the product must meet. Examples of non-functional requirements include safety, security, usability, reliability and performance requirements. Project management issues (costs, time and schedule) are often considered as non-functional requirements. The principal non - functional constraints which are relevant to critical systems :

- performance
- security
- safety
- usability

Performance

Performance requirements concern the speed of operation of a system. Types of performance requirements :

- Response requirements (how quickly the system reacts to a user input).
- Throughput requirements (how much the system can accomplish within a specified amount of time).
- Availability requirements (is the system available for service when requested by end users). The speed of operation of this system is adequate for the requirements.

Reliability

- Reliability is the ability of a system to perform its required function under stated conditions for a specified period of time.
- constraints on the runtime behavior of the system. This system is reliable because its functionalities can be done on the required conditions.

Safety

- Safety requirements are not required which exclude unsafe situation from the possible solution of the system.

Usability

Usability is the ease with which a user can learn to operate, prepare inputs for,and interpret outputs of system or components. Usability requirements include :

- information error messages.
- well-formed user interfaces.

3.7 Interface Requirements

3.7.1 Hardware interfaces

The system must run over the internet, all the hardware shall require to connect internet will be hardware interface for the system. As for example modem, WAN-LAN.

3.7.2 Software interfaces

Software interface required for the working of the project is the appropriate operating system.

3.7.3 Communication interfaces

The user of the site communicate using the network connectivity and the data set is accessed

3.8 Security Requirements

- Validation of input is handled.
- This application containing the computer systems is physically secured against arms or surreptitious entry by intruders.
- Users must be authorized carefully to reduce changes of any such user giving access to an intruder in exchange for a bribe or other favour.

3.9 Platform Used

The platform used for developing and running the project is Windows 11, the latest version of Microsoft's operating system. Windows 11 offers a redesigned and refreshed user interface, new tools and features, and improved security and performance. Windows 11 supports various web development frameworks and languages, such as HTML, CSS, JavaScript, PHP, Python, and more. Windows 11 also enables the use of machine learning models, such as the one used for fog detection, through libraries and tools like TensorFlow, PyTorch, OpenCV, and others. Windows 11 is compatible with most devices that meet the minimum system requirements.

3.10 Technologies Used

HTML

HTML stands for HyperText Markup Language. It is the standard markup language for creating web pages. HTML uses elements, which are defined by tags, to structure and label the content of a web page. HTML elements can have attributes, which provide additional information or functionality. HTML can also include other technologies, such as CSS and JavaScript, to enhance the appearance and behavior of web pages.

CSS

CSS stands for Cascading Style Sheets. It is a language that allows you to style and layout web pages by applying rules to HTML elements. CSS can control the appearance, position, size, color, font, animation, and more of the web content. CSS can also adapt to different devices, screen sizes, and orientations. CSS is one of the core technologies of the web, along with HTML and JavaScript.

PYTHON

Python is a versatile, high-level programming language known for its simplicity and readability. It offers dynamic typing and automatic memory management, making it suitable for various domains such as web development, data analysis, artificial intelligence, and automation. Python's extensive standard library provides pre-built modules and packages for tasks ranging from file manipulation to network programming. Its interpreted nature allows for rapid development and easy debugging. Python supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Its popularity is fueled by a vibrant community, extensive documentation, and numerous third-party libraries. Overall, Python's simplicity, flexibility, and robust ecosystem make it an ideal choice for beginners and seasoned developers alike.

DJANGO

Django is a high-level Python web framework that simplifies web development by emphasizing reusability, rapid development, and the principle of DRY (Don't Repeat Yourself). It follows the Model-View-Controller (MVC) architectural pattern, with models defining data structures, views handling user interface logic, and templates rendering HTML. Django's built-in features include an ORM (Object-Relational Mapping) for interacting with databases, a secure authentication system, URL routing, and a powerful admin interface for managing content. Its scalability and extensibility are further enhanced by a vast ecosystem of third-party packages. Django's emphasis on convention over configuration and its adherence to best practices make it a popular choice for building robust and maintainable web applications.

CNN

Convolutional Neural Networks (CNNs) are a class of deep neural networks primarily used for image recognition and computer vision tasks. They consist of multiple layers, including convolutional layers, pooling layers, and fully connected layers. CNNs excel at learning hierarchical representations of features from input data. Convolutional layers apply filters to input images, extracting features such as edges and textures. Pooling layers reduce spatial dimensions while preserving important features. Fully connected layers combine extracted features for classification or regression tasks. CNNs leverage parameter sharing and local connectivity to efficiently learn patterns from large datasets. Their success in image recognition tasks has made them a cornerstone in various fields, including autonomous driving, medical imaging, and facial recognition.

Chapter 4

4 Design Document

4.1 Purpose

The purpose of the foggy road alert system project is to address the safety concerns posed by reduced visibility due to foggy conditions on roads. By leveraging a combination of web application interface and a trained convolutional neural network (CNN) model, the system aims to detect and quantify the level of fog in video frames. Through real-time analysis, it provides timely alerts to the nearest traffic police stations via email when the fog level exceeds a predefined threshold. This proactive approach intends to enhance road safety by enabling authorities to take precautionary measures promptly, such as issuing advisories, implementing speed restrictions, or deploying resources to manage traffic effectively, thereby reducing the likelihood of accidents and ensuring the well-being of commuters.

4.2 Scope

The project aims to create a web application using a trained CNN model to detect fog levels in uploaded videos, issuing alerts to traffic police stations for hazardous conditions.

4.3 Overview

The purpose of this document is to help the reader to visualize the solution to the project presented.This document verifies how the design meet the requirement stimulated in the SRS document through design viewpoints.The design viewpoints will cover all design elements presented before. This document will provide a direct approach to the development of this project hence reducing feature creep and ponitedly determine the quality of the design.

4.4 Data Design

Database are the storehouses of data used in the software systems.The data is stored in tables inside the database.Several tables are created for the manipulation of the data for the system.Two essential settings for a database are

- Primary Key- The field that is unique for all the record occurrence.
- Foreign Key- The field used to set relation between tables.

Normalization

- Normalization is the process of organizing the data in the database.

- Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate the undesirable characteristics like Insertion, Update and Deletion Anomalies.
- Normalization divides the larger table into the smaller table and links them using relationship.
- The normal form is used to reduce redundancy from the database table.

Chapter 5

5 Development of the System

The development phase of the Foggy Road Alert System using CNN with Django involves the practical implementation of the proposed project. It encompasses several key steps and components that contribute to the seamless integration of Convolutional Neural Networks (CNN) for fire detection with the Django web framework

Chapter 6

6 System Testing

Testing is the penultimate step of software development.An elaborate testing of the data is prepared and the system is using the test data,while doing testing,errors are noted and correction is made.The users are trained to operate the developed system.Both hardware and software securities are made to run the developed system successfully.System testing is aimed at ensuring the system works accurately before the live operation commences.

6.1 Test Plan

6.1.1 Scope

This test plan will cover the following testing activities as identified in the testing strategy.

- White Box Testing

White box testing is a testing technique, that examines the program structure and derives test data from the program logic/code. In white box testing, the UI is bypassed. Inputs and outputs of ratings are tested directly at the code level and the results are compared against specifications. It Reveals errors in "hidden" code.

- Black Box Testing

Black - box testing is a method of software testing that examines the functionality of an application based on the specifications. Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end - user would.

- Unit Testing

The module includes four main components to undergo unit testing. It checks the syntax error, logic error and validity of the program correctness. This test will be performed by the developers.

- Integration Testing

After all components pass the unit testing, they get to be tested whether they work correctly when they are running concurrently and communicating to each other. Integration tests exercise an entire subsystem and ensure that a set of components play nicely together.

- Internal Data Testing

We will test the validity of the data before it enters the database to avoid any problems that may face us in the database. We will test the encryption of the personal information of all the users along with the user names and passwords to ensure maximum security of the users privacy.

6.1.2 Software risk issues

In this section, the plan is to test the risk involved in critical issues such as: Difficult to run on visual studio due to large loading time. Some inherent software risks such as complexity exist; also these issues need to be identified. Proper network connection Working of Mysql database

6.1.3 Features to be tested

- Test whether correct user name and password allows you to login.
- Test whether invalid user name and password prevents you from login.
- Test whether there is any connection problem in the Server.
- Test whether the student and admin details are entered correctly.
- Test whether invalid data entry allows saving data successfully.
- Test whether all pages are loaded correctly.
- Test whether watermark is embedded and extracted properly.
- Test whether the provided security works correctly.

6.2 Test consolidation

6.2.1 Test item

The items or features to be tested in the test cases are included in the document. Each and every user input is tested. The present condition of the system is tested. It is checked to make sure that environment is ready for the application to work.

6.2.2 Input specifications

The input specifications for testing the Foggy Road Alert System invlove varoius type of data to assess system's perfomance comprehensively.The main user input to be tested is uploaded videos.The model trained using CNN also will be tested

Chapter 7

7 System Implementation and Maintenance

7.1 Implementation

System implementation is the conversion of new system into an operating one which involves creating compatible files, training clients and installing hardware. User training is crucial for minimizing resistance to change and giving chance to prove its worth. Training aids user friendly manuals and healthy screens provide the user with a good start. Software maintenance follows conversion to the extent that changes are necessary to maintain satisfactory operations relative to changes in the user's environment. Maintenance often includes minor enhancements or corrections to the problem that surface late in the systems operations. In the implementation phase, the team builds the components either from scratch or by composition. Given the architecture document meant from the design phase and the requirement document from the analysis phase, the team should build exactly what has been requested, though there is still room for innovation and flexibility. For example, a component may be narrowly designed for this particular system, or the component may be made more general to satisfy a reusability

- Careful planning
- Investigation of system and constraints
- Design the methods to achieve changeover.
- Training the staff in the changed phase.
- Evaluation of change over method.
- The method of implementation and time scale to be adopted are found out initially.

7.2 Maintenance

This phase occurs as a result of deploying the whole system at the end users organization. They will perform the beta testing at the end users and inform to the developers about any needed modification to the application. The customer records all the problems that are encountered during the beta testing and reports these to the developer at regular intervals.

7.2.1 Corrective Maintenance

Even with the best quality assurance activities, it is likely that the customer will uncover defects in the software. Corrective maintenance changes the software to correct the defects. Corrective Maintenance activity may consist of repair,

restoration or replacement of equipment. This activity will be result of a regular inspection, which identifies the failure in time for corrective maintenance to be planned and scheduled, then performed during a routine maintenance shutdown.

7.2.2 Adaptive Maintenance

Over time, the original environment(CPU, operating system, business rules, external product characteristics) for which the software was developed is likely to change. Adaptive maintenance results in modification to the software to accommodate changes to its external environment.

7.2.3 Enhanced Maintenance

As software is used, the customer/user will recognize additional functions that will provide the benefit. Perfect maintenance extends the software beyond its original functional requirements.

7.2.4 Preventive Maintenance

Computer software deteriorates due to change, and because of this preventive maintenance often called software re-engineering, must be conducted to enable the software to serve the needs of its end users. Preventive maintenance makes changes to computer programs so that they can be more easily corrected, adapted and enhanced.

Chapter 8

8 Conclusion and Future Scope

8.1 Conclusion

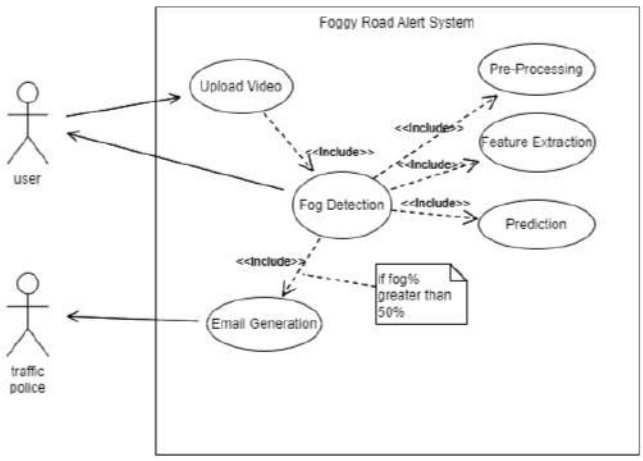
In conclusion, the Foggy Road Alert System serves two primary stakeholders: administrators and traffic authorities. Administrators utilize the dynamic admin panel to manage system features effectively, ensuring smooth operation and optimal performance. Traffic authorities play a critical role in receiving timely alerts generated by the system, enabling them to implement necessary traffic control measures and ensure road safety during foggy conditions. By providing administrators and traffic authorities with essential tools and insights, the Foggy Road Alert System contributes to enhanced road safety and improved traffic management practices.

8.2 Future Scope

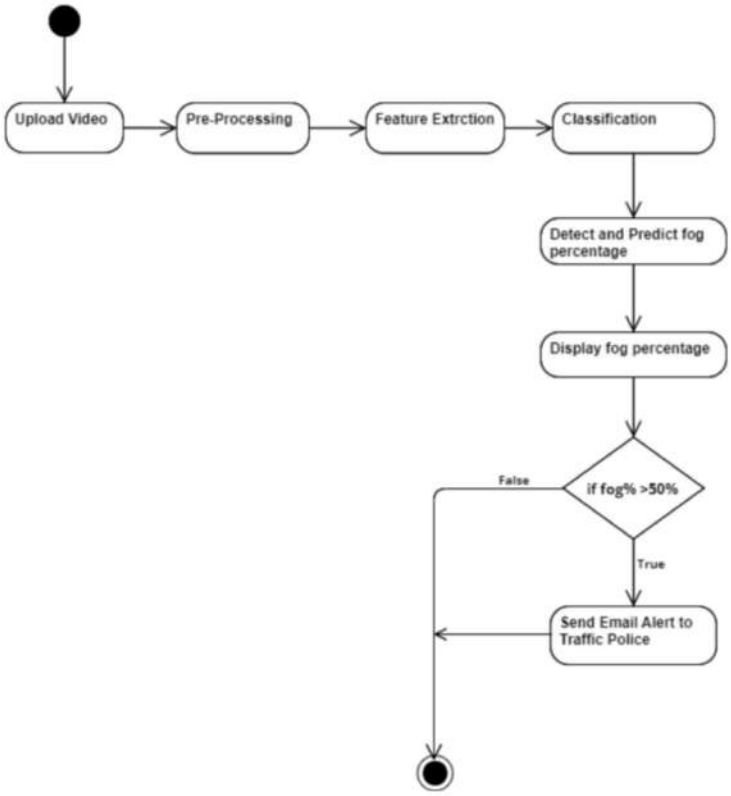
- Implement predictive analytics for proactive measures.
- Collaborate with navigation systems for safer routes.
- Engage users with crowdsourced fog reports.
- Ensure scalable and reliable system architecture.

Appendix

A Usecase Diagram

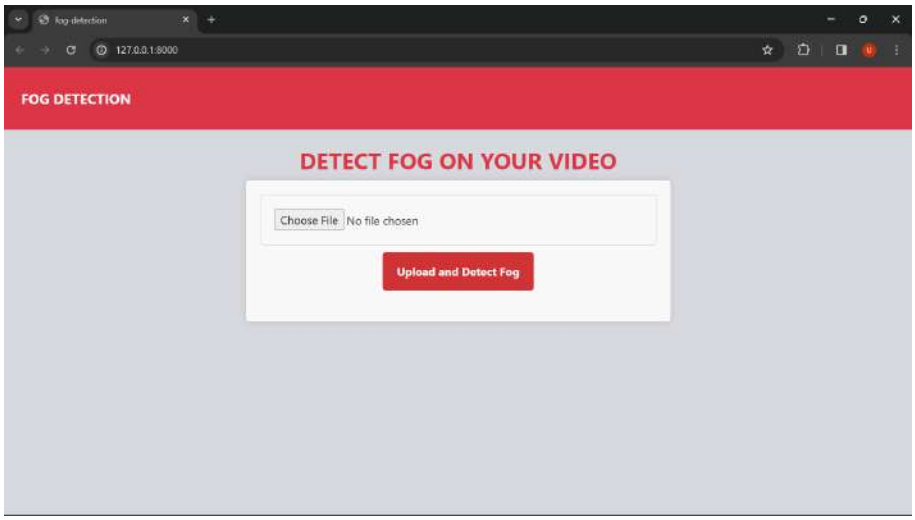


B Activity Diagram

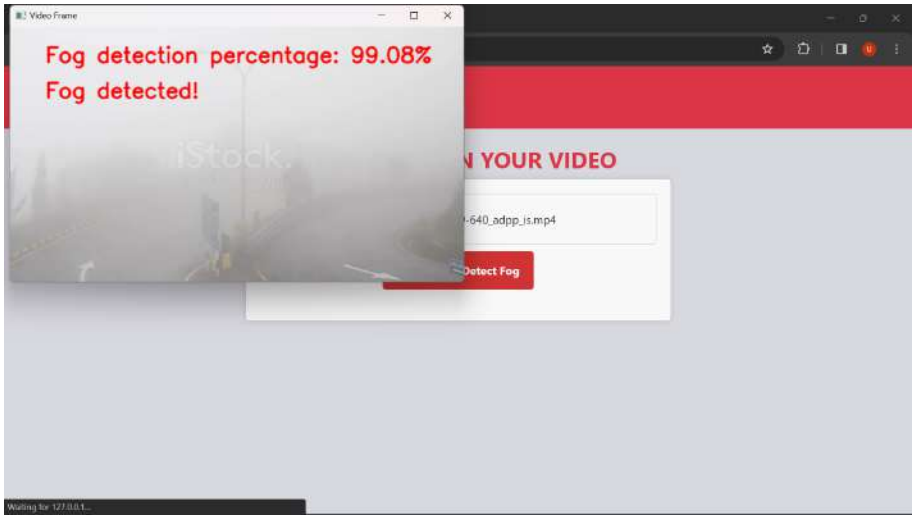


C USER INTERFACES

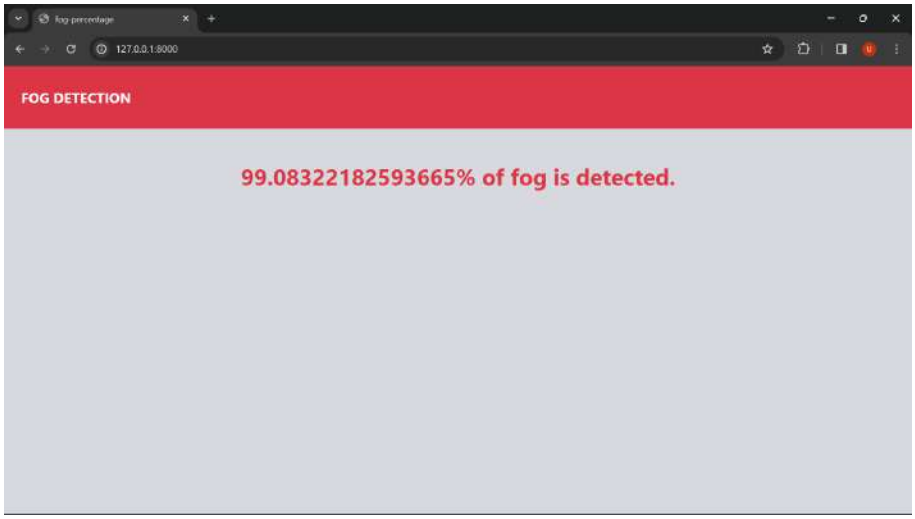
C.1



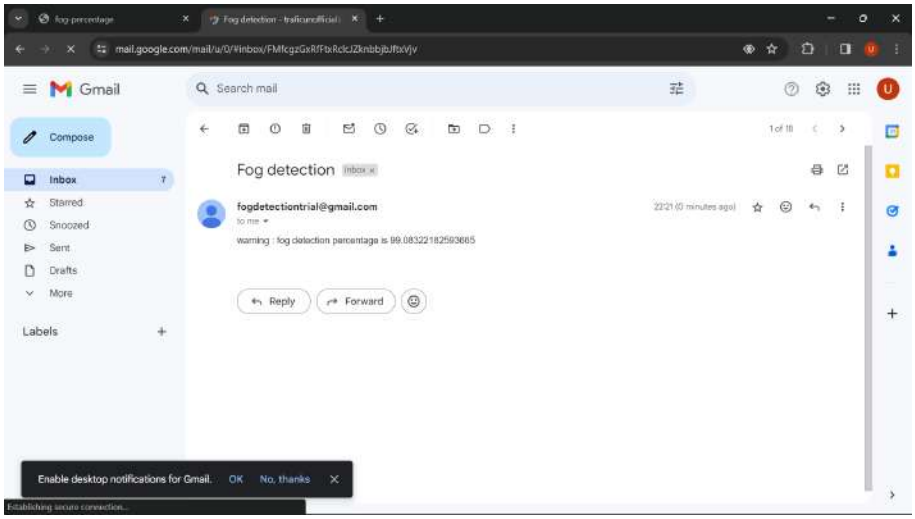
C.2 FOG DETECTION



C.3 FOG PERCENTAGE CALCULATION



C.4 ALERT


Gmail
Search mail
Compose
Inbox
Starred
Snoozed
Sent
Drafts
More
Labels
Fog detection
fogdetectiontrial@gmail.com
warning : fog detection percentage is 99.08322182593665
Reply
Forward
Enable desktop notifications for Gmail.
OK
No, thanks

D CODE

views.py

```
from django.shortcuts import render
import cv2
import numpy as np
from django.core.files.storage import FileSystemStorage
import os
from django.core.mail import send_mail
from keras.models import load_model

def home(request):
    if request.method=="POST" and 'video_file' in request
        .FILES:
        videofile =request.FILES['video_file']
        file_storage=FileSystemStorage()
        filename=file_storage.save(videofile.name,
            videofile)

        video_path =os.path.join(file_storage.location,
            filename)
        if not os.path.exists(video_path):
            error_message = "Video file does not exist."
            return render(request, 'detection_error.html
                ', {'error': error_message})

        cap= cv2.VideoCapture(video_path)
        import tensorflow as tf
        print("TensorFlow version:", tf.__version__)

        model = load_model('C:/Users/USER/OneDrive/
            Desktop/pfog/fog_detection-1/fog_detection/
            fog_detection_project/keras_model.h5')

        total_fog_percetage = []

        if not cap.isOpened():
            error_message="Error opening video file."
            return render(request,'detection_error.html
                ',{'error':error_message})

        while True:

            ret, frame = cap.read()
```

```
        if not ret:
            break

        resized_frame = cv2.resize(frame, (224, 224))

        fog_probability = model.predict(np.
            expand_dims(resized_frame, axis=0))[0, 0]

        threshold = 0.5

        if fog_probability > threshold:

            total_fog_percetage.append(
                fog_probability*100)
            percentage_text = f"Fog detection
                percentage: {fog_probability * 100:.2f
                }%"
            cv2.putText(frame, percentage_text, (50,
                50), cv2.FONT_HERSHEY_SIMPLEX, 1, (0,
                0, 255), 2, cv2.LINE_AA)

            cv2.putText(frame, "Fog detected!", (50,
                100), cv2.FONT_HERSHEY_SIMPLEX, 1, (0,
                 0, 255), 2, cv2.LINE_AA)
        else:

            no_fog_percentage = 0

            cv2.putText(frame, "No fog detected",
                (50, 100), cv2.FONT_HERSHEY_SIMPLEX,
                1, (0, 255, 0), 2, cv2.LINE_AA)

        cv2.imshow('Video Frame', frame)

        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

cap.release()
cv2.destroyAllWindows()

try:
    average_fog_percentage = sum(
        total_fog_percetage) / len(
        total_fog_percetage)
```

```
            if  average_fog_percentage >= 50 :
                subject = 'Fog detection'
                mail_content =f'warning : fog detection
                    percentage is {average_fog_percentage
                    }'
                from_mail = 'fogdetectiontrial@gmail.com'
                recipient_list = ['traficunofficial@gmail
                    .com']
                send_mail(subject, mail_content,
                    from_mail, recipient_list)

                return render(request,'fog_percentage.
                    html',{'f': average_fog_percentage })

        except ZeroDivisionError:
            print("Error: Division by zero. Unable to
                calculate the average.")

            return render(request,'fog_percentage.html
                ',{'f':no_fog_percentage})
    else:
        return render(request,'home.html')
```

fogdetection(2-01-2024).ipynb

```
import os
import cv2
import numpy as np
import tensorflow as tf
from tensorflow.keras.preprocessing import image
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing.image import
    ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D,
     Flatten, Dense

train_dir = "C:/Users/my/OneDrive/Desktop/project/
    fog_detection/fog_dataset"

train_datagen = ImageDataGenerator(rescale=1./255,
    validation_split=0.2)

train_generator = train_datagen.flow_from_directory(
    train_dir,
```

```
    target_size=(150, 150),
    batch_size=32,
    class_mode='binary',
    subset='training'
)

validation_generator = train_datagen.flow_from_directory(
    train_dir,
    target_size=(150, 150),
    batch_size=32,
    class_mode='binary',
    subset='validation'
)

model = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(32, (3, 3), activation='relu',
        input_shape=(150, 150, 3)),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Conv2D(64, (3, 3), activation='relu')
        ,
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Conv2D(128, (3, 3), activation='relu
        '),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(512, activation='relu'),
    tf.keras.layers.Dense(1, activation='sigmoid')
])

model.compile(optimizer='adam', loss='binary_crossentropy
    ', metrics=['accuracy'])
model.fit(train_generator, validation_data=
    validation_generator, batch_size=32, epochs=20,
    validation_steps=len(validation_generator))
model.save("fog_clear_model_2.h5")
model = load_model("keras.model.h5")
```

home.html

```
<!DOCTYPE html>
{% load static %}
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
        initial-scale=1.0">
```

```
    <title>fog-detection</title>
    <link rel="stylesheet" href="https://maxcdn.
        bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min
        .css">
 <link rel="stylesheet" href="{% static 'home.css' %}">
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.
    com/bootstrap/4.5.2/css/bootstrap.min.css">

  </head>
<body style="background-color: rgb(216, 216, 223);">

  <nav class="navbar-nav navbar-expand-lg navbar-light bg
      -danger p-4">
  <a class="navbar-brand" href="{% url 'fog_detection:
      home' %}" style="color: white;font-weight: bold;">
      FOG DETECTION</a>

  </nav>

  <div class="pt-4">
  <h2 style="font-weight:bold ;" class="text-danger">
      DETECT FOG ON YOUR VIDEO </h2>
<div class="form-container">
<form method="POST" enctype="multipart/form-data">
  {% csrf_token %}
  <div class="text-center">
  <input type="file" name="video_file">

  <input  type="submit" value="Upload and Detect Fog">
</div>

</form>
</div>
</div>


</body>
</html>
```

fogpercentage.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
        initial-scale=1.0">
```

```
    <title>fog-percentage</title>
    <link rel="stylesheet" href="https://maxcdn.
        bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min
        .css">

</head>
<body style="background-color: rgb(216, 216, 223);">
    <nav class="navbar-nav navbar-expand-lg navbar-light
        bg-danger p-4">
        <a class="navbar-brand" href="{% url '
            fog_detection:home' %}" style="color: white;
            font-weight: bold;">FOG DETECTION</a>

        </nav>
    <div class="form-container pt-5">

        <h2 class="text-danger font-weight-bold text-
            center">{{f}}% of fog is detected.</h2>

    </div>

</body>
</html>
```

settings.py

```
from pathlib import Path
import os

BASE_DIR = Path(__file__).resolve().parent.parent

SECRET_KEY = 'django-insecure-uqy^qc_gs6#n6a1juaeyc^x2qy!
    l$@18xom0b3*&g$b7%!j@68'

DEBUG = True

ALLOWED_HOSTS = []

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'fog_detection',
]
```

```
MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware
        ',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.
        AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware
        ',
    'django.middleware.clickjacking.
        XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'fog_detection_project.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.
            DjangoTemplates',
        'DIRS': [ BASE_DIR / 'templates'],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug
                    ',
                'django.template.context_processors.
                    request',
                'django.contrib.auth.context_processors.
                    auth',
                'django.contrib.messages.
                    context_processors.messages',
            ],
        },
    },
]

WSGI_APPLICATION = 'fog_detection_project.wsgi.
    application'

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
    }
```

```
}

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.
            UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.
            MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.
            CommonPasswordValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.
            NumericPasswordValidator',
    },
]

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_TZ = True

STATIC_URL = 'static/'
STATICFILES_DIRS = [BASE_DIR / 'static']

DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
MEDIA_URL = '/media/'
MEDIA_ROOT = os.path.join(BASE_DIR , 'media')

EMAIL_BACKEND = 'django.core.mail.backends.smtp.
    EmailBackend'
EMAIL_HOST = 'smtp.gmail.com'
EMAIL_USE_TLS = True
EMAIL_PORT = 587
EMAIL_HOST_USER = 'fogdetectiontrial@gmail.com'
EMAIL_HOST_PASSWORD = 'anmh saxu exmh pfyr'
```

VENTURE VISTA

PROJECT REPORT

Submitted By

MARIA MERRYL

Reg. No. CCAVBCA006

for the award of the Degree of

Bachelor of Computer Application (B.C.A.)

(University of Calicut)

under the guidance of

Ms. Soumya P.S

Assistant Professor

Bachelor of Computer Application
DEPARTMENT OF COMPUTER SCIENCE
CHRIST COLLEGE(Autonomous)
IRINJALAKUDA, KERALA
INDIA

2021-24

DEPARTMENT OF COMPUTER SCIENCE

CHRIST COLLEGE (AUTONOMOUS)

IRINJALAKUDA

CERTIFICATE

This is to certify that the project report entitled ***"VENTURE VISTA"*** *is a bonafide record of the project work done by* ***MARIA MERRYL*** *in partial fulfillment of the requirement for the sixth semester of* ***Bachelor of Computer Application*** *in Department of Computer Science of* ***CHRIST COLLEGE (AUTONOMOUS) IRINJALAKUDA***

.

Ms. SOUMYA P.S
Assistant Professor,CS
Internal Guide

Ms. SINI THOMAS
Head of the Department
Computer Science

EXTERNAL EXAMINER

INTERNAL EXAMINER

DECLARATION

We hereby declare that this project work **"VENTURE VISTA"** submitted to Christ College (Autonomous) Irinjalakuda, affiliated to Calicut University in partial fulfillment of the requirement for the award of the Bachelor of Computer Application, is a record of original work done by us, under the guidance of Ms. SOUMYA P.S, Department of computer Science.

Place: Irinjalakuda MARIA MERRYL

ACKNOWLEDGEMENT

First and foremost I like to thank Lord almighty for his providence and for being the guiding light throughout the project. I wish to express my sincere gratitude to my beloved Department head for giving me all the facilities for my project. I take this opportunity to express my gratitude to the class teacher Ms. SOUMYA P.S and head of the department Ms. SINI THOMAS who has been supported us throughout the course of this project. I are thankful for her aspiring guidance and valuable advice during the project work. I express my sincere thanks to my project guide Ms. SOUMYA P.S for supporting and guiding throughout the project.I would take this opportunity to specially thank all other faculty members for their constant and continuous motivation. Finally I would like to thank my family and friends for giving valuable advice and moral support throughout my project.

ABSTRACT

VENTURE VISTA Our travel guide website aims to inspire and assist travelers in exploring new destinations worldwide. From detailed city guides to off-the-beaten-path adventures, we provide a wealth of information to help users plan unforgettable trips. With curated recommendations for accommodations, dining, activities, and transportation, our platform caters to all types of travelers, security scores,live money exchange rates. By combining expert insights with user reviews and ratings, we strive to be a one-stop resource for discovering and planning your next adventure.

Contents

Chapter 1

1 Introduction

This is an web application that helps the users to know more about the major attractions near to them. The application provides all the details about tourist places or major attractions like their opening time, closing time, history and other details. The app also provides all the details of the hotels available for the users. The app can also provide the best route to the places you have selected. The app will come in handy for those who travel a lot, or like to travel to new places.

1.1 Overview

The objective of the TRAVEL GUIDE APPLICATION is to design an simple and adaptable application that helps the users to know more about the major attractions near to them. The web application includes many interesting features such as providing place suggestions to visit which best suites for the preferences you have given.It is a User friendly application.

Chapter 2

2 System Analysis

2.1 Purpose

A travel guide application serves as a comprehensive resource for travelers, offering a wide range of information and features to enhance their travel experiences. Its primary purpose is to provide users with detailed insights into various destinations, including popular attractions, historical sites, cultural experiences, dining options, accommodations, and local services.

By leveraging the app, users can efficiently plan their trips, exploring different destinations based on their interests and preferences. They can access detailed descriptions, reviews, and ratings for various attractions and services, helping them make informed decisions about their travel itinerary.

Furthermore, a travel guide application often includes features such as offline maps, itinerary planners, currency converters, and language translators, which can be invaluable for travelers navigating unfamiliar environments. These features can help users navigate their destinations more effectively, communicate with locals, and manage their travel budgets.

Overall, a travel guide application aims to simplify the travel planning process, inspire users to explore new destinations, and enhance their overall travel experiences by providing them with relevant and reliable information.
.

2.1.1 Existing System

The existing system are apps that provides data about tour packages and so on. Apps like this, which provide all details of the places are not available. Limitations of existing system :We cannot get all the details about the places we are looking for from a single application.

2.1.2 Proposed System

The proposed system is an web application which provides all the details related to the places you are looking for your next trip. We also provide a feature that gives you suggestions of the places near you to visit, based on the preferences you have provided. Advantages of proposed system :The application gives all the details about the places you are looking for.It also

provide place suggestions to visit which best suites for the preferences you have given.

2.2 Problem definition

To know what the problem is and what the needs are before developing it. Here we design an application for travel guiding.

2.3 FEASIBILITY STUDY

After the problem is clearly understood and solutions proposed, the next step is to conduct the feasibility study. Feasibility study is defined as evaluation or analysis of the potential impact of the proposed project or program. The objective is to determine whether the proposed system is feasible. There are three aspects of feasibility study to which the proposed system is subjected as discussed below.

2.3.1 Technical Feasibility

Technical feasibility assess whether the current technical resources are sufficient for the new system. We can upgrade the level of technology for supporting our website. We check whether the proposed system can be implemented in the present system without supporting the existing hardware.

2.3.2 Economical Feasibility

Economic feasibility determines whether the time and money are available to develop the system. There is no additional hardware used to develop the site, it is inexpensive to build.

2.3.3 Operational Feasibility

Operational feasibility determines if the human resources are available to operate the system once it has been installed. No extra training is needed to use this system. Anyone who has knowledge in internet in English language can easily use the system. The resources that are required to implement or install are already available with the organization.

Chapter 3

3 Software Requirement Specification

3.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the TRAVEL GUIDE APPLICATION. It illustrate the purpose and complete description for the development of the system. It explain system constraints, interface and interactions with other external applications. This document is primarily intended to be proposed to provide all the details related to the places you are looking for your next trip.

3.2 Scope

The application gives all the details about the places you are looking for.It also provide place suggestions to visit which best suites for the preferences you have given.

3.3 Overall Description

This section give an overview of our application, TRAVEL GUIDE APPLICATION. This is an web application that helps the users to know more about the major attractions near to them. The application provides all the details about tourist places or major attractions like their opening time, closing time, history and other details. The app also provides all the details of the hotels available for the users. The app can also provide the best route to the places you have selected. The app will come in handy for those who travel a lot, or like to travel to new places.

3.3.1 Product Perspective

TRAVEL GUIDE APPLICATION is mainly used to know the details about your next trip. It also provide features that gives you suggestions to visit according to the preferences you have given.

3.3.2 Product Functionality

Through this system admin can upload various data. User can view the major attractions and their history and other details available in an area.

Users can also view all the information about the hotels available in a specific area.

3.3.3 Users and Characteristics

There are two types of users that interact with the application, people who wish to travel and the guide. Each of these have different tasks which is performed. Admin is able to upload details of places and can view the feedback.

3.4 Specific Requirements

3.4.1 Hardware Requirements

- System: IBM-Compatible PC
- Processor: Intel Core i3
- Speed: Above 1GHz
- RAM capacity: 4 GB
- Hard Dsk drive: 500 GB

3.4.2 Software Requirements

- Front End : React
- Back End : Djanko Python
- Database : MySql

3.5 Functional Requirements

It contains three main modules.

- 1.Admin
- 2.User

Admin

The admin will upload the details of the places. All the details that are provided in the app is uploaded by the admin. The admin can view user's feedbacks and can take suitable decisions based on the user's decisions.

User

The users can view the major attractions and their history and other details available in an area. Users can also view all the information about the hotels available in a specific area. The users can get the best route available to the selected place from their current location, which is calculated by the app and shown in the map. Users can also get notifications on the places that are suggested by the app based on the preferences they have given.

3.6 Non Functional Requirements

Non-functional requirements define the overall qualities or attributes of the resulting system. Non-functional system place restrictions on the product being developed, the developed process, and specify external constraints that the product must meet. Examples of non-functional requirements include safety, security, usability, reliability and performance requirements. Project management issues (costs, time and schedule) are often considered as non-functional requirements. The principal non - functional constraints which are relevant to critical systems :

- performance
- security
- safety
- usability

Performance

Performance requirements concern the speed of operation of a system. Types of performance requirements :

- Response requirements (how quickly the system reacts to a user input).
- Throughput requirements (how much the system can accomplish within a specified amount of time).
- Availability requirements (is the system available for service when requested by end users). The speed of operation of this system is adequate for the requirements.

Reliability

- Reliability is the ability of a system to perform its required function under stated conditions for a specified period of time.

- constraints on the runtime behavior of the system. This system is reliable because its functionalities can be done on the required conditions.

Safety

Safety requirements are not required which exclude unsafe situation from the possible solution of the system.

Usability

Usability is the ease with which a user can learn to operate, prepare inputs for, and interpret outputs of system or components. Usability requirements include :

- information error messages.
- well-formed user interfaces.

3.7 Interface Requirements

3.7.1 Hardware interfaces

The system must run over the internet, all the hardware shall require to connect internet will be hardware interface for the system. As for example modem, WAN-LAN.

3.7.2 Software interfaces

Software interface required for the working of the project is the appropriate operating system.

3.7.3 Communication interfaces

The user of the site communicate using the network connectivity and the data set is accessed

3.8 Security Requirements

- User accesses only their account.
- Validation of input is handled.
- This application containing the computer systems is physically secured against arms or surreptitious entry by intruders.
- Users must be authorized carefully to reduce changes of any such user giving access to an intruder in exchange for a bribe or other favour.

3.9 Platform Used

Windows 10 is a major version of the Microsoft Windows operating system that was released on July 29, 2015.It is built on the Windows NT kernel and follows windows 8.Part of the reason Microsoft decided to name the 2015 release "Windows 10"(and skipped "windows 9")is because the operating system is designed to be a new direction for Microsoft. One of the primary aims of windows 10 is to Unify the windows experience across multiple devices, such desktop computers, tablets, and smartphones. As part of this effort, Microsoft developed Windows10 Mobile alongside Windows 10 to replaces Windows Phone - Microsoft's previous mobile OS. Windows 10 also integrates other Microsoft services, such as Xbox Live and the Cortana voice recognition assistant.

3.10 Technologies Used

Djanko

Django is a free and open-source web framework, written in Python, that follows the model-template-views (MTV) architectural pattern. It is used to build database-driven websites. Django takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel.

React

React.js is an open-source JavaScript library used to build user interfaces (UIs) for single-page applications. React manages the view layer and is used for the development of both web and mobile applications.React is a component-based library, which means that UIs are built by combining reusable components. Components are self-contained pieces of code that can be used to render HTML, CSS, and JavaScript.React uses a virtual DOM, which is a lightweight representation of the real DOM. The virtual DOM is used to efficiently update the UI when data changes.React is a popular choice for building UIs because it is fast, scalable, and easy to learn.

MySQL

MySQL, pronounced either "My S-Q-L" or "My Squel, " is an open source relational database management system. It is based on the structure query

language (SQL), which is used for adding, removing and modifying information in the database. Standard SQL commands, such as ADD, DROP, INSERT,and UPDATE can be used with MySQL.

What distinguishes PHP from something like client - side Javascript is that the code is executed on the server, generating HTML which is then sent to the client. The client would receive the results of running that script, but would not know what the underlying code was. You can even configure your web server to process all your HTML files with PHP, and then there's really no way that users can tell what you have up your sleeve. The best things in using PHP are that it is extremely simple for a newcomer, but offers many advanced features for a professional programmer.

Chapter 4

4 Design Document

4.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the TRAVEL GUIDE APPLICATION.. It illustrate the purpose and complete description for the development of the system. It explain system constraints, interface and interactions with other external applications. This document is primarily intended help the users to know more about the major attractions near to them.This section give an overview of our system, "9 to 5" APPLICATION. The application provides all the details about tourist places or major attractions like their opening time, closing time, history and other details. The app also provides all the details of the hotels available for the users. The app can also provide the best route to the places you have selected. The app will come in handy for those who travel a lot, or like to travel to new places.

4.2 Scope

The application gives all the details about the places you are looking for.It also provide place suggestions to visit which best suites for the preferences you have given.

4.3 Overview

The purpose of this document is to help the reader to visualize the solution to the project presented.This document verifies how the design meet the requirement stimulated in the SRS document through design viewpoints.The design viewpoints will cover all design elements presented before. This document will provide a direct approach to the development of this project hence reducing feature creep and ponitedly determine the quality of the design.

4.4 Data Design

Database are the storehouses of data used in the software systems.The data is stored in tables inside the database.Several tables are created for the manipulation of the data for the system.Two essential settings for a database are

- Primary Key- The field that is unique for all the record occurrence.
- Foreign Key- The field used to set relation between tables.

Normalization

- Normalization is the process of organizing the data in the database.
- Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate the undesirable characteristics like Insertion, Update and Deletion Anomalies.
- Normalization divides the larger table into the smaller table and links them using relationship.
- The normal form is used to reduce redundancy from the database table.

Chapter 5

5 Development of the System

The system is an web application which provides all the details related to the places you are looking for your next trip. We also provide a feature that gives you suggestions of the places near you to visit, based on the preferences you have provided.

Chapter 6

6 System Testing

Testing is the penultimate step of software development.An elaborate testing of the data is prepared and the system is using the test data,while doing testing,errors are noted and correction is made.The users are trained to operate the developed system.Both hardware and software securities are made to run the developed system successfully.System testing is aimed at ensuring the system works accurately before the live operation commences.

6.1 Test Plan

6.1.1 Scope

This test plan will cover the following testing activities as identified in the testing strategy.

- White Box Testing

White box testing is a testing technique, that examines the program structure and derives test data from the program logic/code. In white box testing, the UI is bypassed. Inputs and outputs of ratings are tested directly at the code level and the results are compared against specifications. It Reveals errors in "hidden" code.

- Black Box Testing

Black - box testing is a method of software testing that examines the functionality of an application based on the specifications. Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end - user would.

- Unit Testing

The module includes four main components to undergo unit testing. It checks the syntax error, logic error and validity of the program correctness. This test will be performed by the developers.

- Integration Testing

After all components pass the unit testing, they get to be tested whether they work correctly when they are running concurrently and communicating to each other. Integration tests exercise an entire subsystem and ensure that a set of components play nicely together.

- Internal Data Testing

We will test the validity of the data before it enters the database to avoid any problems that may face us in the database. We will test the encryption of the personal information of all the users along with the user names and passwords to ensure maximum security of the users privacy.

6.1.2 Software risk issues

In this section, the plan is to test the risk involved in critical issues such as: Difficult to run on visual studio due to large loading time. Some inherent software risks such as complexity exist; also these issues need to be identified. Proper network connection Working of Mysql database

6.1.3 Features to be tested

- Test whether correct user name and password allows you to login.
- Test whether invalid user name and password prevents you from login.
- Test whether there is any connection problem in the Server.
- Test whether the student and admin details are entered correctly.
- Test whether invalid data entry allows saving data successfully.
- Test whether all pages are loaded correctly.
- Test whether watermark is embedded and extracted properly.
- Test whether the provided security works correctly.

6.2 Test consolidation

6.2.1 Test item

The items or features to be tested in the test cases are included in the document. Each and every user input is tested. The present condition of the system is tested. It is checked to make sure that environment is ready for the application to work.

6.2.2 Input specifications

These are the inputs required to execute the test case. All required inputs including data elements and values are given below.

Data	Values
Email	A valid email ID
Password	Combination of characters and numbers
Phone number	10 Digit number

Chapter 7

7 System Implementation and Maintenance

7.1 Implementation

System implementation is the conversion of new system into an operating one which involves creating compatible files, training clients and installing hardware. User training is crucial for minimizing resistance to change and giving chance to prove its worth. Training aids user friendly manuals and healthy screens provide the user with a good start. Software maintenance follows conversion to the extent that changes are necessary to maintain satisfactory operations relative to changes in the user's environment. Maintenance often includes minor enhancements or corrections to the problem that surface late in the systems operations. In the implementation phase, the team builds the components either from scratch or by composition. Given the architecture document meant from the design phase and the requirement document from the analysis phase, the team should build exactly what has been requested, though there is still room for innovation and flexibility. For example, a component may be narrowly designed for this particular system, or the component may be made more general to satisfy a reusability

- Careful planning
- Investigation of system and constraints
- Design the methods to achieve changeover.
- Training the staff in the changed phase.
- Evaluation of change over method.
- The method of implementation and time scale to be adopted are found out initially.

7.2 Maintenance

This phase occurs as a result of deploying the whole system at the end users organization. They will perform the beta testing at the end users and inform to the developers about any needed modification to the application. The customer records all the problems that are encountered during the beta testing and reports these to the developer at regular intervals.

7.2.1 Corrective Maintenance

Even with the best quality assurance activities, it is likely that the customer will uncover defects in the software. Corrective maintenance changes the software to correct the defects. Corrective Maintenance activity may consist of repair, restoration or replacement of equipment. This activity will be result of a regular inspection, which identifies the failure in time for corrective maintenance to be planned and scheduled, then performed during a routine maintenance shutdown.

7.2.2 Adaptive Maintenance

Over time, the original environment(CPU, operating system, business rules, external product characteristics) for which the software was developed is likely to change. Adaptive maintenance results in modification to the software to accommodate changes to its external environment.

7.2.3 Enhanced Maintenance

As software is used, the customer/user will recognize additional functions that will provide the benefit. Perfect maintenance extends the software beyond its original functional requirements.

7.2.4 Preventive Maintenance

Computer software deteriorates due to change, and because of this preventive maintenance often called software re-engineering, must be conducted to enable the software to serve the needs of its end users. Preventive maintenance makes changes to computer programs so that they can be more easily corrected, adapted and enhanced.

Chapter 8

8 Conclusion and Future Scope

8.1 Conclusion

There are two types of users that interact with the system Admin, and the traveler.Each of these two types has different uses of the system so each of them has their own panel.Admin can manage all the features of application dynamically by login on admin panel. The users can view the major attractions and their history and other details available in an area. Users can also view all the information about the hotels available in a specific area.

8.2 Future Scope

- Gives all the details about the places you are looking for.
- Provide place suggestions to visit which best suites for the preferences you have given.

Appendix

A Data Flow Diagram

Data flow is the one of the best way of documenting the entire functionality of the system.For the system ,which will have data flows in and have some processing inside and then some data flow out from the system can be documented or represented effectively by means of data flow out from the system can be documented or represented effectively by means of data flow diagrams. The data flow diagram are a diagrammatic representation of the system,which has input,process and output.Once any system is represented using a data flow diagram we can identify the following things easily:

- Various entities interacting with the system are identified
- Flow of data from one entity to another is identified
- The various processes involved in between the interaction of two or more entities in the system are clearly pointed out
- The various data stores which hold the data in between the process,are clearly identified

Some Data Flow Diagram charting forms:

A.1 External source or receiver

A source or sink is a person or part of organization ,which enters or receives information from the system,but is considered to be outlining the contest of data flow model.

A.2 Transform process

A process represents transformation where incoming data flowsare changed into outgoing data flow

A.3 Data Store

A data store is repository of data that is to be stored for use by one or more process may be as simple as buffer or queue or sophisticated as relational database. They should have clear names.If a process merely uses the contest of store and does not alter it,the arrowhead goes only form the store to the

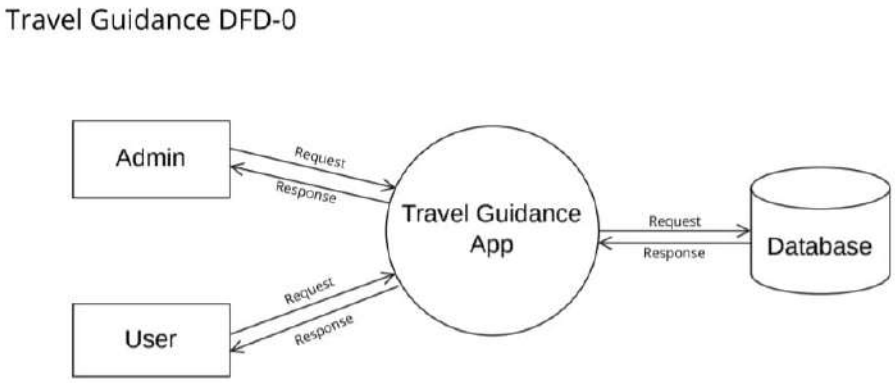
process. If a process alters the details in the store then double-headed arrow is used.

A.4 Data flow

A data flow is a route, which enable packets of data to travel from one point to another.Data may flow,with arrowhead pointing in the direction of the flow

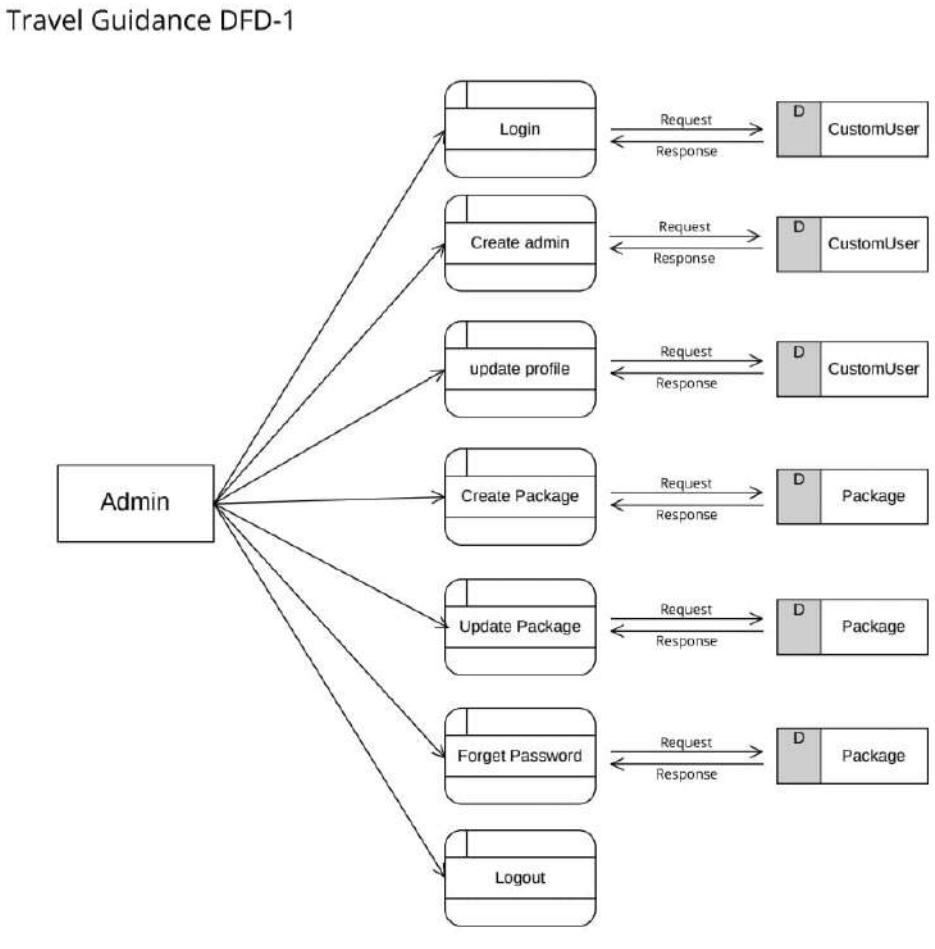
B dfd0

B.1 Level 0



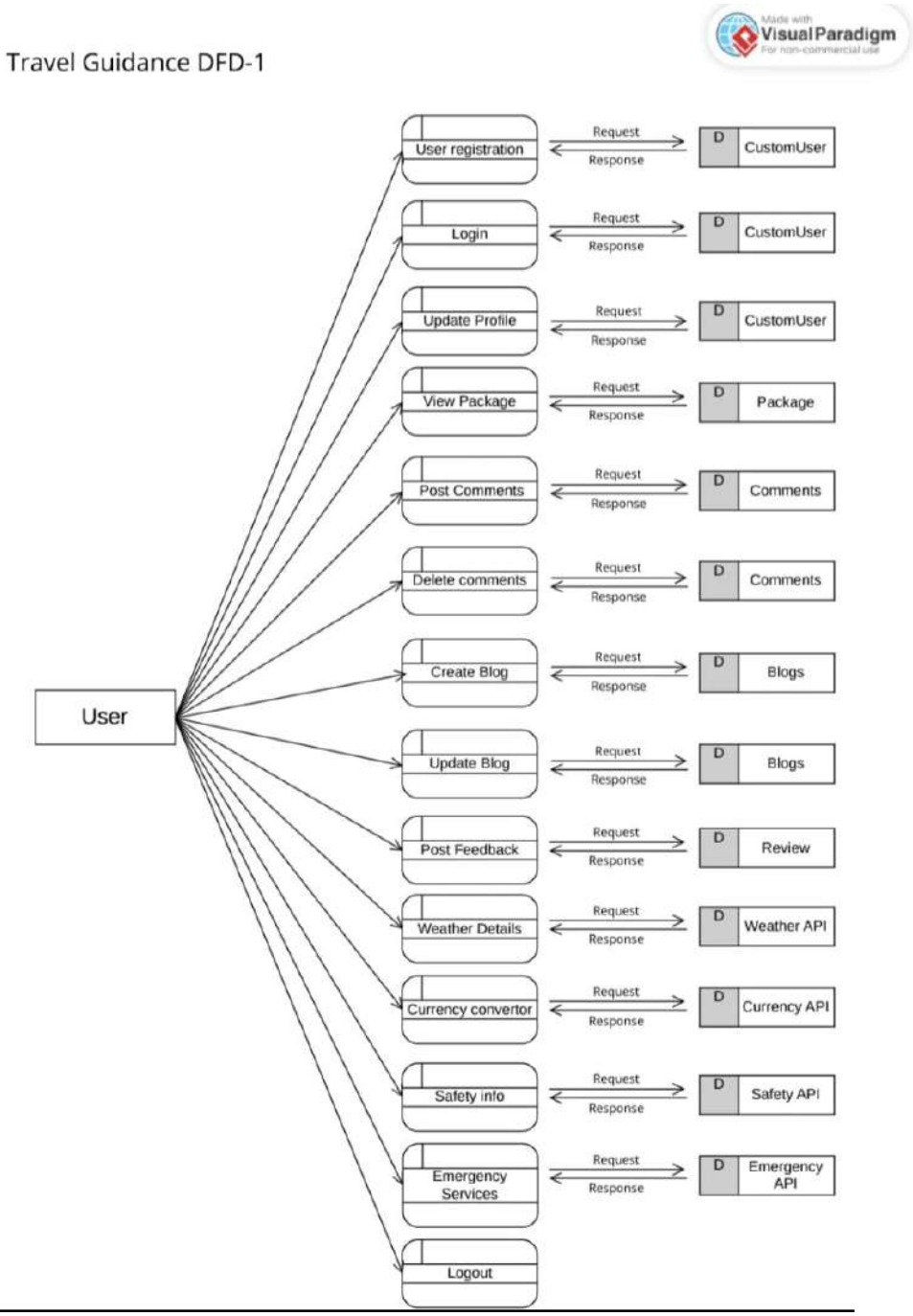
C Data Flow Diagram of Admin

C.1 Level 0

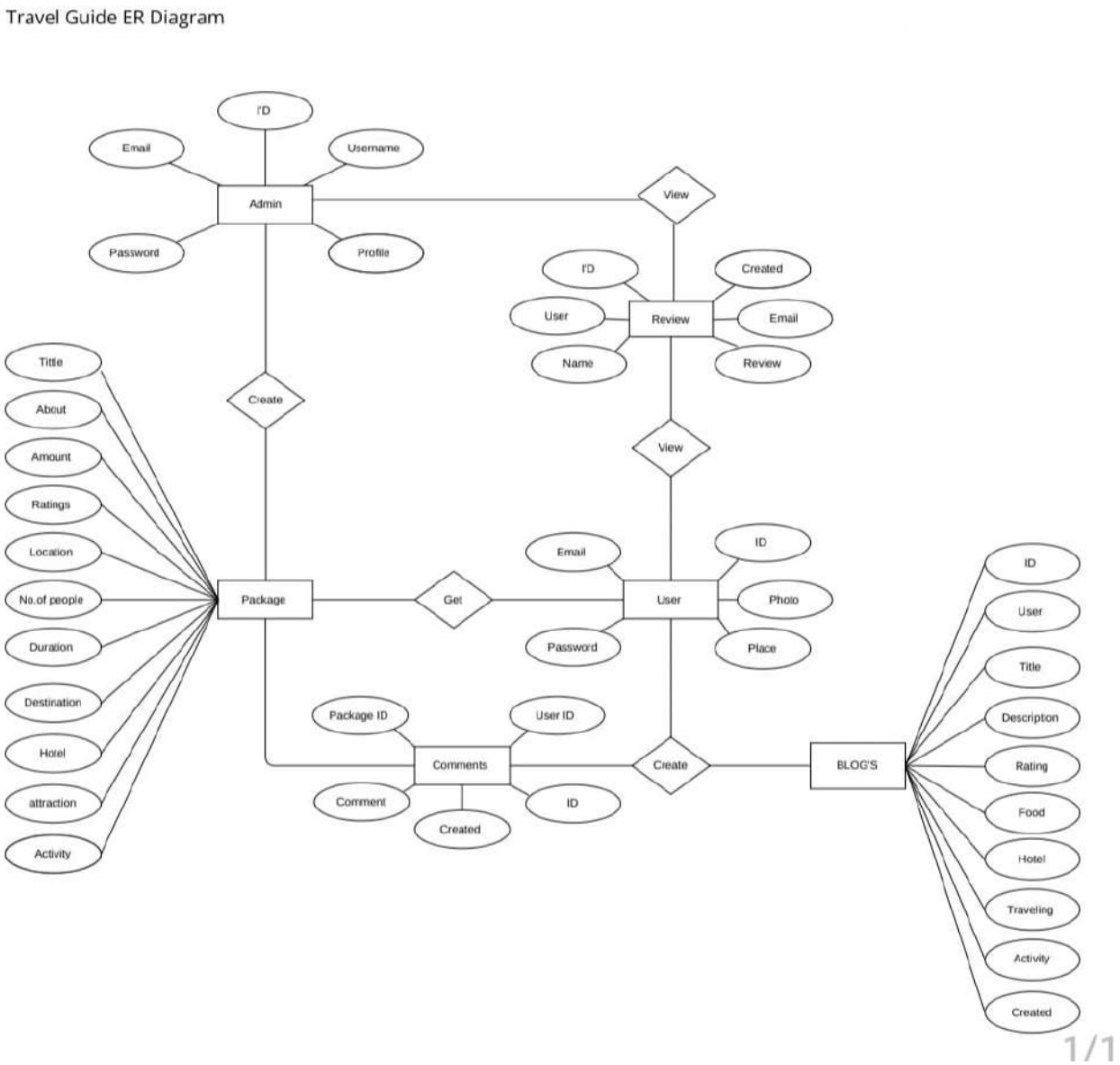


D Data Flow Diagram of User

D.1 Level 0

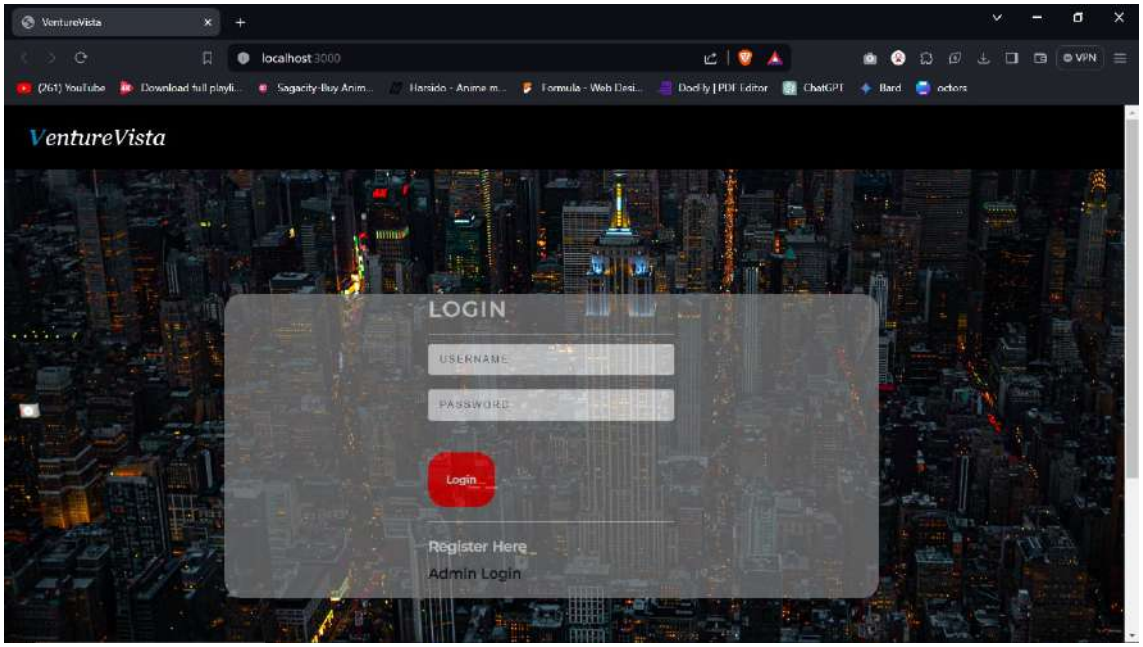


E E-R Diagram

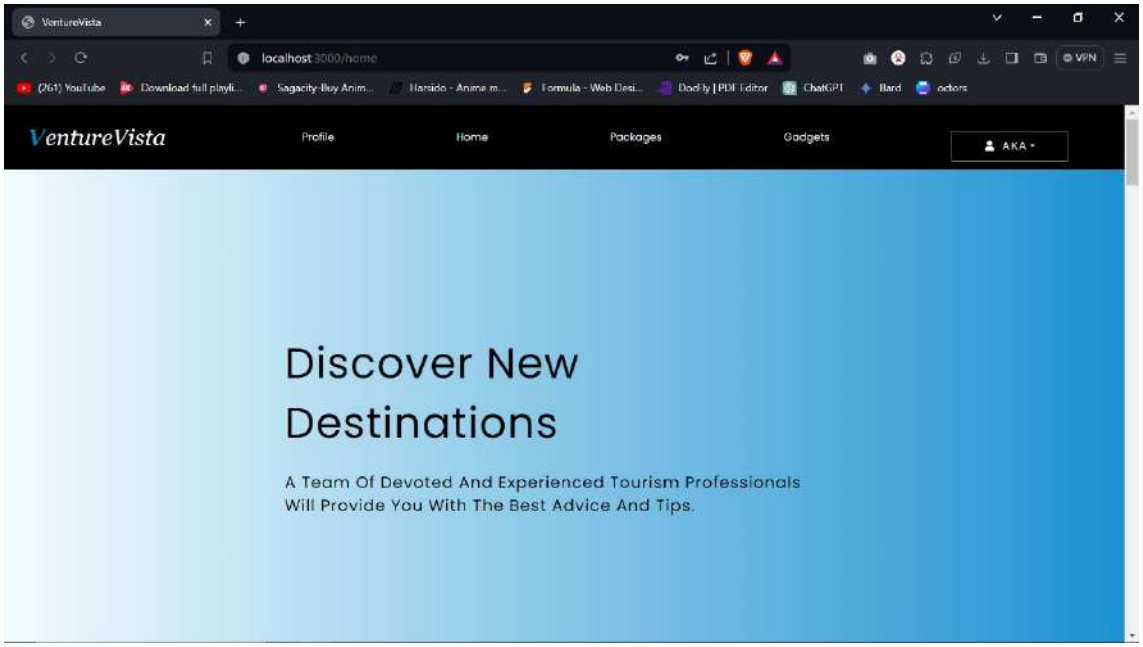

Travel Guide ER Diagram
I'D
Email
Username
Admin
Password
Profile
View
I'D
Created
User
Review
Email
Name
Review
Tittle
About
Amount
Ratings
Location
No.of people
Duration
Destination
Hotel
attraction
Activity
Create
View
Email
ID
Package
Get
User
Photo
Password
Place
Package ID
User ID
Comments
Create
BLOG'S
Comment
ID
Created
ID
User
Title
Description
Rating
Food
Hotel
Traveling
Activity
Created
1/1

F USER INTERFACES

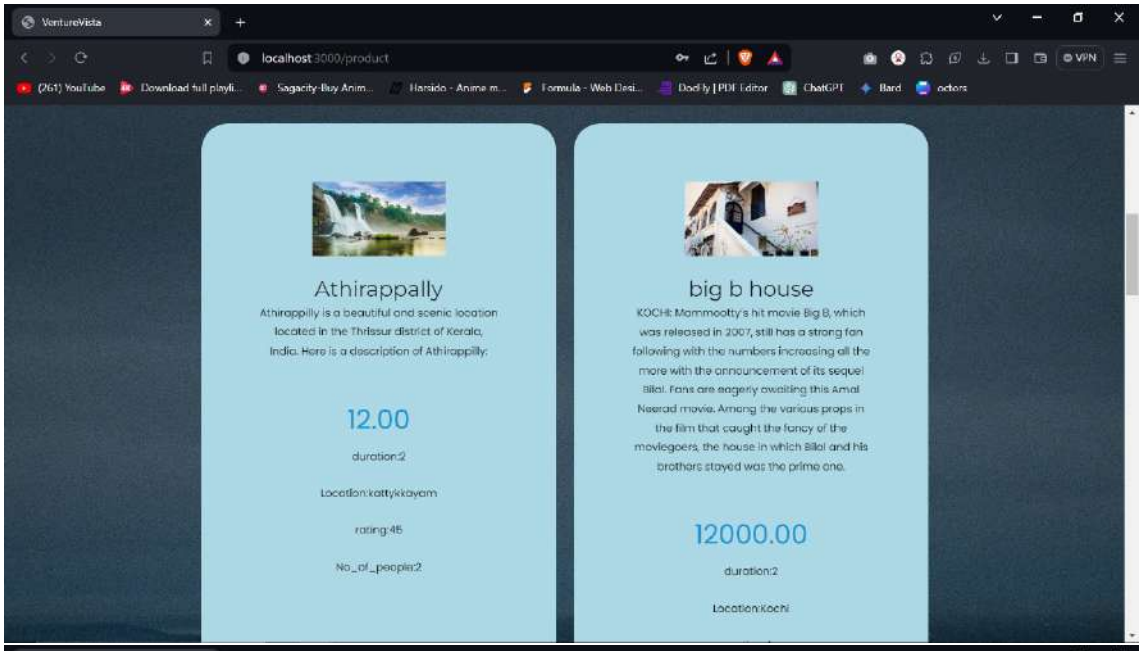
F.1 LOGIN

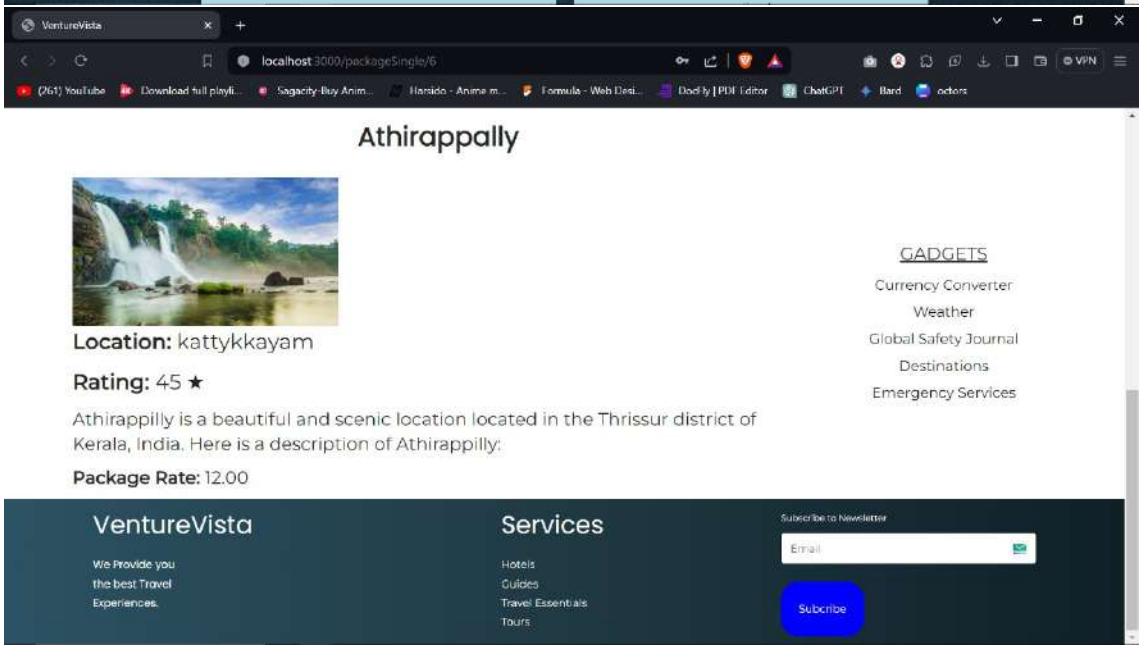


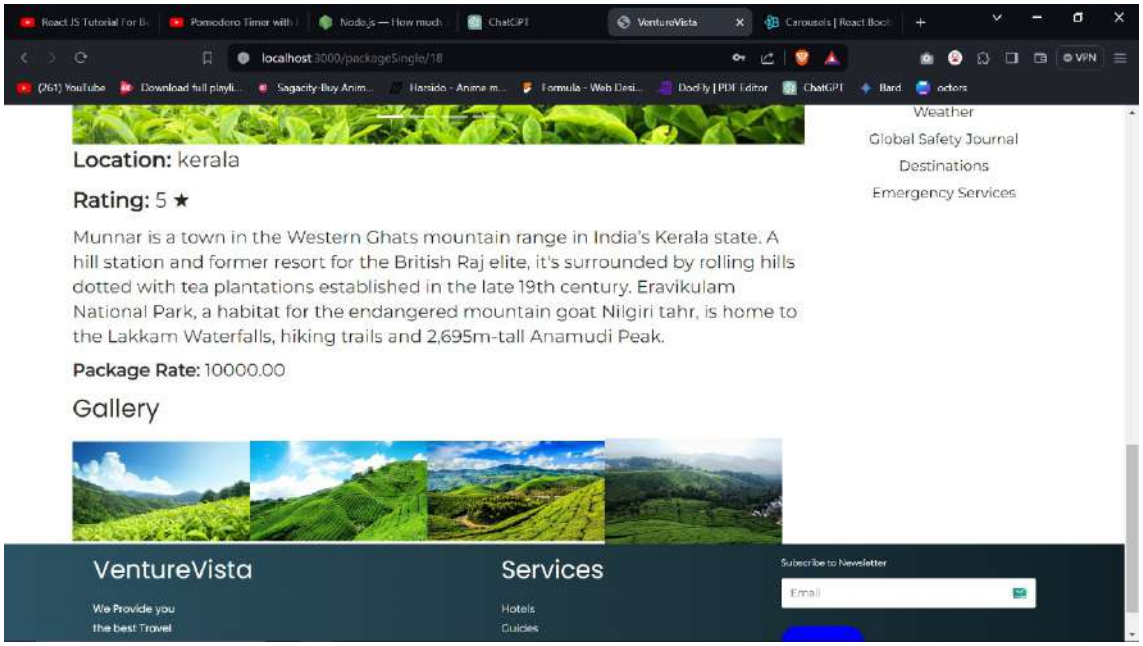
F.2 HOME


VentureVista
localhost:3000/home
VentureVista
Profile
Home
Packages
Gadgets
AKA
Discover New Destinations
A Team Of Devoted And Experienced Tourism Professionals
Will Provide You With The Best Advice And Tips.

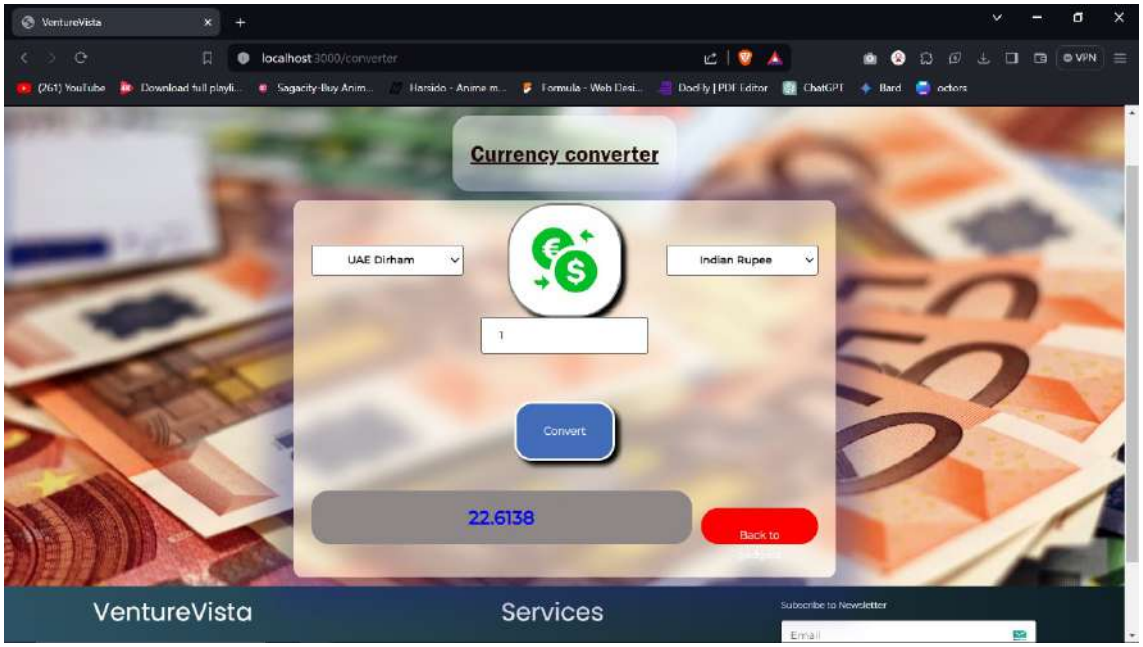
F.3 PACKAGES


VentureVista
localhost:3000/product
Athirappally
Athirappilly is a beautiful and scenic location located in the Thrissur district of Kerala, India. Here is a description of Athirappilly:
12.00
duration:2
Location:kattykkayam
rating:45
No_of_people:2
big b house
KOCHI: Mammootty's hit movie Big B, which was released in 2007, still has a strong fan following with the numbers increasing all the more with the announcement of its sequel Bilal. Fans are eagerly awaiting this Amal Neerad movie. Among the various props in the film that caught the fancy of the moviegoers, the house in which Bilal and his brothers stayed was the prime one.
12000.00
duration:2
Location:Kochi

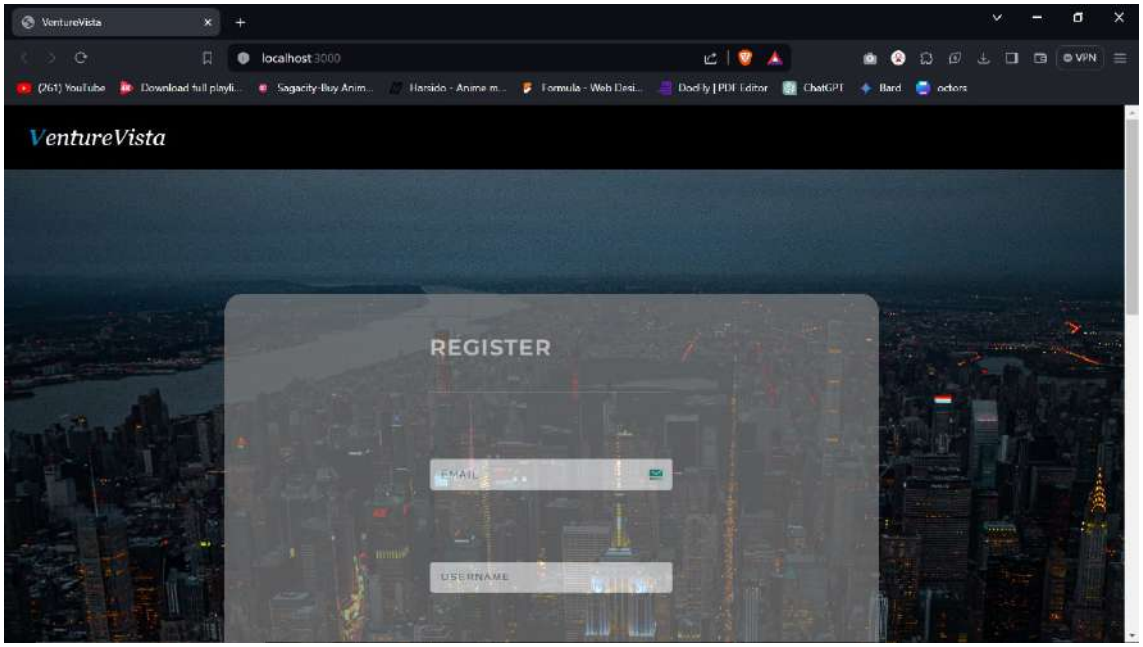

VentureVista
localhost:3000/packageSingle/6
Athirappally
Location: kattykkayam
Rating: 45 ★
Athirappilly is a beautiful and scenic location located in the Thrissur district of Kerala, India. Here is a description of Athirappilly:
Package Rate: 12.00
GADGETS
Currency Converter
Weather
Global Safety Journal
Destinations
Emergency Services
VentureVista
We Provide you the best Travel Experiences.
Services
Hotels
Guides
Travel Essentials
Tours
Subscribe to Newsletter
Email
Subcribe


localhost:3000/packageSingle/18
Weather
Global Safety Journal
Destinations
Emergency Services
Location: kerala
Rating: 5 ★
Munnar is a town in the Western Ghats mountain range in India's Kerala state. A hill station and former resort for the British Raj elite, it's surrounded by rolling hills dotted with tea plantations established in the late 19th century. Eravikulam National Park, a habitat for the endangered mountain goat Nilgiri tahr, is home to the Lakkam Waterfalls, hiking trails and 2,695m-tall Anamudi Peak.
Package Rate: 10000.00
Gallery
VentureVista
We Provide you the best Travel
Services
Hotels
Guides
Subscribe to Newsletter
Email

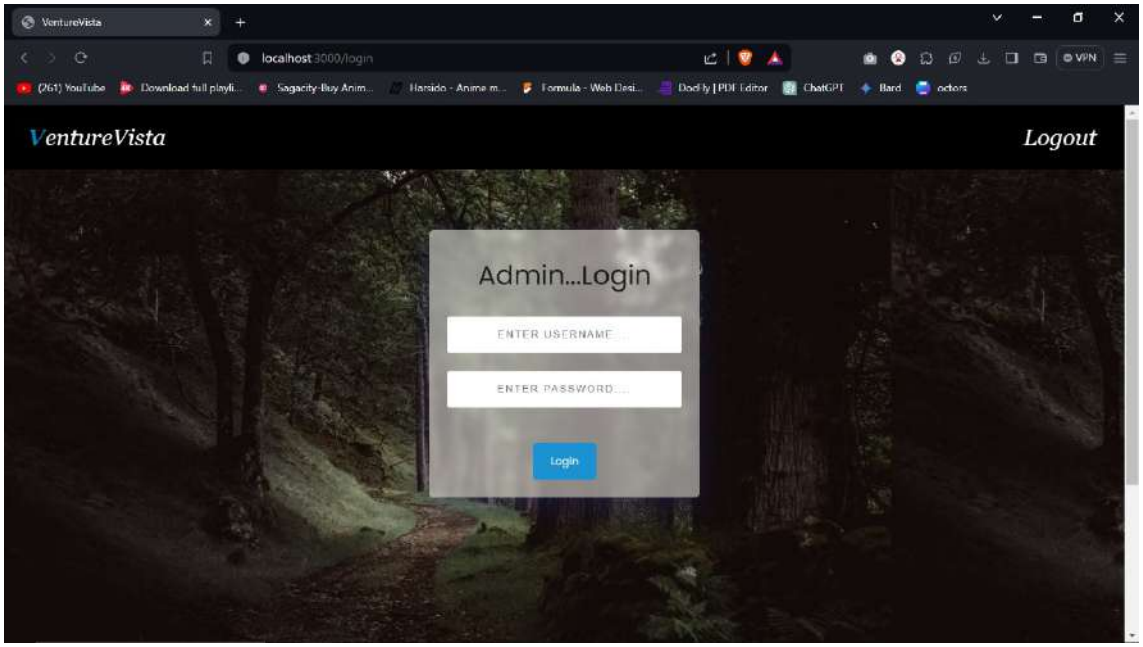
F.4 CURRENCY CONVERTER



F.5 REGISTER


VentureVista
localhost:3000
VentureVista
REGISTER
EMAIL
USERNAME

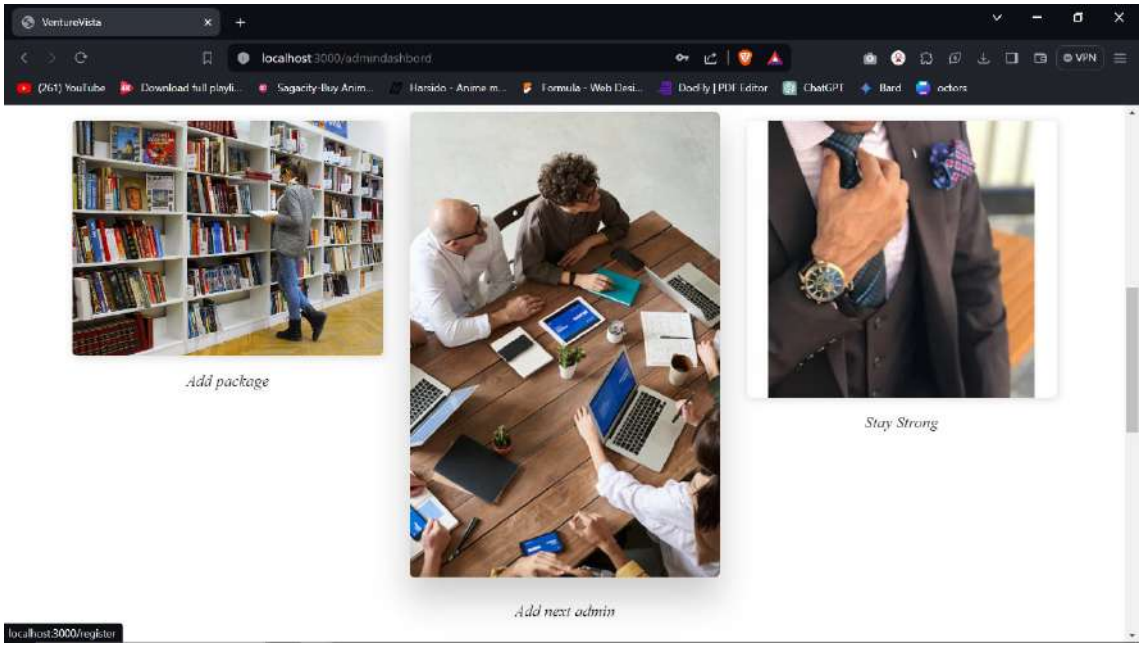
F.6 ADMIN LOGIN



F.7 ADMIN


VentureVista
localhost:3000/admindashboard
VentureVista
Logout
WELCOME MR.ADMIN
THE MAGICAL POWERES OF
YOU ARE....

F.8 ADMIN OPTIONS



G CODE

FRONTEND

```
//
// Source code recreated from a .class
file by
IntelliJ IDEA
// (powered by Fernflower decompiler)
//

import React, { useState, useEffect }
from
"react";
import { Modal, Button, Carousel } from
"react-bootstrap";
import { BrowserRouter, Route, Link }
from
"react-router-dom";
import Image from
"../images/athirappalli.jpeg";
import "./product.css";
function Product({ product }) {
const [show, setShow] = useState(false);
const handleClose = () =>
setShow(false);
const handleShow = () => setShow(true);  console.log(product)

  return (
    <>
      <div>
        <div className="product--card">
<img src={product.attraction}
alt="Product
Shoes" className="product--img" />
          <h4>{product.title}</h4>
          <p>
            {product.about}
          </p>
          <h2>{product.amount}</h2>
```

```
<p>duration:{product.duration}</p><p>Location:{product.location}
</p><p>rating:{product.rating}</p><p>
No_of_people:{product.no_of_people}</p>
        </div>
      </div>
    </>
  );
}

export default Product;
import React from 'react'
import './gadgets.css'
function AdminPowers() {
  return (
<body data-spy="scroll"
data-target=".onpage-navigation"
data-offset="60">
      <main>

<section class="bg-dark-30
showcase-page-header module parallax-bg"id='re' data-background="">
          <div class="titan-caption">
<div class="caption-content"><div class="font-alt mb-30
titan-title-size-1">Our best gadgets to
assits
you</div>
<div class="font-alt mb-40
titan-title-size-4">feel free to rush
trough
them</div>
            </div>
          </div>
        </section>
<div class="main showcase-page"><section class="module-medium"
id="demos"> <div class="container">
<div class="row multi-columns-row"><div
class="col-md-4 col-sm-6 col-xs-12"><a
class="content-box" href="/converter">
<div class="content-box-image"><img
src="https://blog.cdn.cmarix.com/blog/wp-content
```

```
/uploads/2023/07/develop-a-currency-converter-mobile-app.webp"
alt="Currency Converter" /></div>
<h3 class="content-box-title
font-serif">Currency
Converter</h3></a></div>
<div class="col-md-4 col-sm-6
col-xs-12"><a
class="content-box" href="Wheather">
<div class="content-box-image"><img
src="https://play-lh.googleusercontent.com/znc7FD1B
qyFR92b8n6hyROZrAR3FsTyV_ThO79hyuQG-Nb45z2qHFDTnf1HXH50DYg"
alt="Agency" /></div>
<h3 class="content-box-title
font-serif">wheather app</h3></a></div>
<div class="col-md-4 col-sm-6
col-xs-12"><a
class="content-box" href="/journal">
<div class="content-box-image"><img
src="https://s.yimg.com/ny/api/res/1.2/rx7Bjb7IghNsT35t13jufQ--
/YXBwaWQ9aGlnaGxhbmRlcjt3PTY0MDtoPTQ4MA--
/https://s.yimg.com/os/creatr-images/2020-05/45815870-9238-11ea-8fff-
1180915abec1"
alt="Portfolio" /></div>
<h3 class="content-box-title
font-serif">Global Safety
Journal</h3></a></div>
<div class="col-md-4 col-sm-6
col-xs-12"><a
class="content-box"
href="/destinations">
<div class="content-box-image"><img
src="https://images.pexels.com/photos/7235910/pexels-photo-
7235910.jpeg?auto=compress&cs=tinysrgb&w=300"
alt="Restaurant" /></div>
<h3 class="content-box-title
font-serif">Destinations</h3></a></div>
<div class="col-md-4 col-sm-6
col-xs-12"><a
class="content-box" href="/emergency">
<div class="content-box-image"><img
```

```
src="https://cdn.dribbble.com/userupload/7319839/
file/original-86f171450ede48437f8b1cfd8ac352f1.png?
crop=0x0-2880x2160&resize=400x300&vertical=center"
alt="Finance" /></div>
<h3 class="content-box-title
font-serif">Emergency
services</h3></a></div>
{/* <div class="col-md-4 col-sm-6
col-xs-12"><a class="content-box"
href="/product">
<div class="content-box-image"><img
src="https://cdn.dribbble.com/userupload/10920096/file/

original-4b166c456ae22ae068a4c961d6053a74.png?resize=400x0"
alt="Landing"/></div>
<h3 class="content-box-title
font-serif">Travel
packages</h3></a></div> */}
              </div>
            </div>
          </section>
        </div>
<div class="scroll-up"><a
href="#totop"><i
class="fa
fa-angle-double-up"></i></a></div>
      </main>
<script
src="assets/lib/jquery/dist/jquery.js"></script>
<script
src="assets/lib/bootstrap/dist/js/bootstrap.min.js"></script>
<script
src="assets/lib/wow/dist/wow.js"></script>
<script
src="assets/lib/jquery.mb.ytplayer/dist/jquery.mb.YTPlayer.js"></script>
<script
src="assets/lib/isotope/dist/isotope.pkgd.js"></script>
<script
src="assets/lib/imagesloaded/imagesloaded.pkgd.js"></script>
<script
```

```
src="assets/lib/flexslider/jquery.flexslider.js"></script>
<script
src="assets/lib/owl.carousel/dist/owl.carousel.min.js"></script>
<script
src="assets/lib/smoothscroll.js"></script>
<script
src="assets/lib/magnific-popup/dist/jquery.magnific-popup.js"></script>
<script
src="assets/lib/simple-text-rotator/jquery.simple-text-rotator.min.js"></script>
<script
src="assets/js/plugins.js"></script><script
src="assets/js/main.js"></script>
</body>

  )
}

export default AdminPowers
import React, { useState, useEffect }
from
"react";
import { useNavigate } from
"react-router-dom";
import axios from "axios";
import Loader from
"../components/Loader";
import Error from "../components/Error";import Success from
"../components/Success";
import "./login.css";
import { Link } from "react-router-dom";// import Loginimg from
"../images/Login.svg";function
LoginScreen() {

const [FormData, setFormData] =
useState({

  });
const navigate=useNavigate()
const [loading, setLoading] =
useState(false); const [error, setError]
```

```
= useState("");
const [success, setSuccess] =
useState("");
const handlechange=(e)=>{
  setFormData({
    ...FormData,
    [e.target.name]:e.target.value
  })
}
const Login=async(e)=>{
const {username,password}=FormData

await axios.post('admin-login/',{
  username,
  password
}).then((res)=>{
  if(res.status===200){
    console.log(res.status)
    if(res.status==200){
localStorage.setItem('adminkey',res.data.access)
localStorage.setItem('isadmin',res.data.is_superuser)
alert("admin login
successfull..welcome..."+res.data.username)
navigate('/admindashbord')
    }
    else{
alert("detected unautherized entry...")    }
    setLoading(false);
  }

}).catch(()=>{
alert("detected unauthorized entry..")  })
}

  return (
    <div className="login--screen">
{/* <img src={Loginimg} alt="login img"
className="login--img" /> */}
      {loading && <Loader></Loader>}
```

```
<div className="row
justify-content-center"> <div
className="col-md-5 mt-5">
{error.length > 0 && <Error
msg={error}></Error>}
          <div className="bs">
            <h2>Admin...Login</h2>

            <input
              type="text"
              className="form-control"
              name="username"
placeholder="Enter username...."
onChange={handlechange}
              style={{
                textAlign: "center",
                padding: "20px",
                marginBottom: "20px",
              }}
            />
            <input
              type="password"
              className="form-control"
placeholder="enter Password...."
name="password"
              onChange={handlechange}
              style={{
                textAlign: "center",
                padding: "20px",
                marginBottom: "20px",
              }}
            />
            {loading ? (
<div>Login...Please Wait...</div> ) : (
<button className="login--btn"
onClick={Login}>
                Login
              </button>
            )}
          </div>
```

```
        </div>
      </div>
    </div>
  );
}

export default LoginScreen;
```

BACKEND

```
//
// Source code recreated from a .class
file by
IntelliJ IDEA
// (powered by Fernflower decompiler)
//
from django.db import models

# Create your models here.


from django.contrib.auth.models import
AbstractUser
from django.db import models

class CustomUser(AbstractUser):

phone = models.CharField(max_length=15,
blank=True)
place = models.CharField(max_length=255,blank=True)
nickname =
models.CharField(max_length=30,
blank=True)
color = models.CharField(max_length=30,
blank=True)
image =
models.ImageField(upload_to='user_images/',
blank=True, null=True)
```

```
    def _str_(self):
        return self.username



class Package(models.Model):
admin = models.ForeignKey(CustomUser,
on_delete=models.CASCADE)
title = models.CharField(max_length=255)    about = models.TextField()
amount =
models.DecimalField(max_digits=10,
decimal_places=2)
    rating = models.IntegerField()
location =
models.CharField(max_length=255)duration
=
models.CharField(max_length=50)no_of_people
= models.CharField(max_length=50)hotel =
models.ImageField(upload_to='package_photos/',
blank=True, null=True)
destination =
models.ImageField(upload_to='package_photos/',
blank=True, null=True)
activity =
models.ImageField(upload_to='package_photos/',
blank=True, null=True)
attraction =
models.ImageField(upload_to='package_photos/',
blank=True, null=True)

    def _str_(self):
        return self.title


class Comments(models.Model):
user = models.ForeignKey(CustomUser,
on_delete=models.CASCADE)
package = models.ForeignKey(Package,
on_delete=models.CASCADE)
```

```
    comment = models.TextField()
created =
models.DateField(auto_now_add=True)
    def _str_(self):
        return self.comment


class Blogs(models.Model):
user = models.ForeignKey(CustomUser,
on_delete=models.CASCADE)
title = models.CharField(max_length=255)    description = models.TextField()
    rating = models.IntegerField()
food =
models.ImageField(upload_to='blog_photos/',
blank=True, null=True)
hotel =
models.ImageField(upload_to='blog_photos/',
blank=True, null=True)
travelling =
models.ImageField(upload_to='blog_photos/',
blank=True, null=True)
activity =
models.ImageField(upload_to='blog_photos/',
blank=True, null=True)
created =
models.DateField(auto_now_add=True)
    def _str_(self):
        return self.title

class Review(models.Model):

name =
models.CharField(max_length=255,blank=True,null=True)
email=models.CharField(max_length=255,blank=True,null=True)
    review = models.TextField()
created =
models.DateField(auto_now_add=True)
    def _str_(self):
        return self.review
"""
```

```
URL configuration for travel_guide
project.

The urlpatterns list routes URLs to
views. For
more information please see:
https://docs.djangoproject.com/en/5.0/topics/http/urls/
Examples:
Function views
1. Add an import: from my_app import
views2. Add a URL to urlpatterns:
path('',
views.home, name='home')
Class-based views
1. Add an import: from other_app.views
import
Home
2. Add a URL to urlpatterns: path('',
Home.as_view(), name='home')
Including another URLconf
1. Import the include() function: from
django.urls import include, path
2. Add a URL to urlpatterns:
path('blog/',
include('blog.urls'))
"""
from django.contrib import admin
from django.urls import path
from . import views
from rest_framework_simplejwt import
views as
jwt_views

urlpatterns = [
path('',views.UserRegistrationView.as_view(),
name='user-registration'),
path('admin-register/',
views.SuperuserRegistrationView.as_view(),
name='superuser-registration'),
path('user-login/',
```

```
views.UserloginView.as_view(),
name='user-token_obtain_pair'),
path('admin-login/',
views.AdminloginView.as_view(),
name='admin-token_obtain_pair'),
path('api/token/refresh/',
jwt_views.TokenRefreshView.as_view(),
name='token_refresh'),
path('user-details/',
views.UserDetailsView.as_view(),
name='user-details'),
path('update-user/',
views.UpdateUserDetailsView.as_view(),
name='update-user-details'),

path('send-otp/',
views.PasswordResetOTPSendView.as_view(),
name='update-user-details'),
path('otp-validation/',
views.OTPValidationView.as_view(),
name='otp-validation'),
path('change-password/',
views.ChangePasswordView.as_view(),
name='new-password'),

path('package-crud/',
views.PackageCRUDView.as_view()),
path('package-crud/<int:pk>/',
views.PackageCRUDView.as_view(),),

path('list-packages/',
views.ListPackagesView.as_view(),
name='list-packages'),
path('package-detail/<int:pk>/',
views.PackageDetailView.as_view(),
name='package-detail-view'),

path('create-comment/<int:pk>/',
views.CreateCommentView.as_view(),
name='create-comment'),
```

```
path('list-comments/<int:pk>/',
views.ListCommentsView.as_view(),
name='list-comments'),
path('delete-comment/<int:pk>/',
views.DeleteCommentView.as_view(),
name='delete-comment'),


path('create-blog/',
views.CreateBlogView.as_view(),
name='create-blog'),
path('list-blogs/',
views.ListBlogsView.as_view(),
name='list-blogs'),
path('blog-detail/<int:pk>/',
views.BlogDetailView.as_view(),
name='blog-detail'),

path('list-user-blogs/',
views.ListUserBlogsView.as_view(),
name='list-user-blogs'),
path('update-blog/<int:pk>/',
views.UpdateBlogView.as_view(),
name='update-blog'),
path('delete-blog/<int:pk>/',
views.DeleteBlogView.as_view(),
name='delete-blog'),


path('create-review/',
views.CreatReviewView.as_view(),),
path('list-reviews/',
views.ListReviewView.as_view(),),
path('review-detail/<int:pk>/',
views.ReviewDetailView.as_view(),),


]
from rest_framework import serializers
from .models import
```

```
CustomUser,Package,Comments,Blogs,Reviewfrom
rest_framework_simplejwt.serializers
import TokenObtainPairSerializer
from django.core.exceptions import
ObjectDoesNotExist


class
UserTokenObtainPairSerializer(TokenObtainPairSerializer):
    @classmethod
    def get_token(cls, user):
        token = super().get_token(user)
        # adding custom claims
token['username'] = user.usernametoken['is_superuser'] =
user.is_superuser return token

    def validate(self, attrs):
        data = super().validate(attrs)
        user = self.user
        if user.is_superuser==False:
data["is_superuser"] = user.is_superuser
data["username"] = user.username
            return data
        else:
raise serializers.ValidationError("Only
common
users are allowed to log in here.")


class
AdminTokenObtainPairSerializer(TokenObtainPairSerializer):
    @classmethod
    def get_token(cls, user):
        token = super().get_token(user)
        # adding custom claims
token['username'] = user.usernametoken['is_superuser'] =
user.is_superuser return token

    def validate(self, attrs):
        data = super().validate(attrs)
```

```
        user = self.user
        if user.is_superuser==True:
data["is_superuser"] =
user.is_superuserdata["username"] =
user.username return data
        else:
raise serializers.ValidationError(" Onlyadministrators are allowed to log in
here.")



class
CustomUserSerializer(serializers.ModelSerializer):
password_confirmation =
serializers.CharField(write_only=True)
email =
serializers.EmailField(required=True)
image=serializers.ImageField(max_length=None,use_url=True,required=False)
    class Meta:
        model = CustomUser
fields = ('id', 'username', 'email',
'phone',
'place', 'image',
'password','password_confirmation')
extra_kwargs = {'password':
{'write_only':
True}}

    def validate(self, data):
        try:
CustomUser.objects.get(email=data['email'])raise
serializers.ValidationError({'email':'Email
already exists.'})
        except ObjectDoesNotExist:
            pass

if data['password'] !=
data['password_confirmation']:
raise
serializers.ValidationError({'password_confirmation':"Passwords
```

```
do not match."})
        return data

    def create(self, validated_data):
validated_data.pop('password_confirmation',
None)
user =
CustomUser.objects.create_user(**validated_data)
        return user

class
CustomUserupdateSerializer(serializers.ModelSerializer):
image=serializers.ImageField(max_length=None,use_url=True,required=False)
    class Meta:
        model = CustomUser
fields =
['username','image','phone','place','email']

    def validate_email(self, value):
if
CustomUser.objects.filter(email__iexact=value)
.exclude(pk=self.instance.pk).exists():
raise serializers.ValidationError("This
email
address is already in use.")
        return value


class
PackageSerializer(serializers.ModelSerializer):
    class Meta:
        model = Package
        exclude = ['admin']

class
PackagelistSerializer(serializers.ModelSerializer):
    class Meta:
        model = Package
        fields = "_all_"
```

```
class
CommentSerializer(serializers.ModelSerializer):
    class Meta:
        model = Comments
        fields = ['comment']

class
CommentlistSerializer(serializers.ModelSerializer):
    class Meta:
        model = Comments
        fields = "_all_"

class
BlogSerializer(serializers.ModelSerializer):
    class Meta:
        model = Blogs
        exclude = ['user']

class
BloglistSerializer(serializers.ModelSerializer):
    class Meta:
        model = Blogs
        fields = "_all_"


class
ReviewSerializer(serializers.ModelSerializer):
    class Meta:
        model = Review
        fields = "_all_"

class
ReviewlistSerializer(serializers.ModelSerializer):
    class Meta:
        model = Review
        fields = "_all_"
from django.contrib import admin

# Register your models here
from .models import
```

```
CustomUser,Package,Comments,Blogs,Review# admin.site.register(CustomUser)
class
CustomUseryadmin(admin.ModelAdmin):
list_display =
['username','is_superuser']admin.site.register(CustomUser,CustomUseryadmin)

class Packageadmin(admin.ModelAdmin):
list_display =
['admin','title','amount','location','duration']
admin.site.register(Package,Packageadmin)

class Commentsadmin(admin.ModelAdmin):
list_display =
['user','comment','package']admin.site.register(Comments,Commentsadmin)

class Blogsadmin(admin.ModelAdmin):
    list_display = ['user','title']
admin.site.register(Blogs,Blogsadmin)



class Reviewadmin(admin.ModelAdmin):
    list_display = ['name','review']
admin.site.register(Review,Reviewadmin)
```

Dr.BABY
VACCINATION APPLICATION

PROJECT REPORT

Submitted By

NAVEED NIHAN K M

Reg. No. CCAVBCA045

Bachelors

in Computer Application
(University of Calicut)

under the guidance of

Mr.Thoufeeq Ansari

Assistant Professor

Bachelors in Computer Application
DEPARTMENT OF COMPUTER SCIENCE
CHRIST COLLEGE(Autonomous)
IRINJALAKUDA, KERALA
INDIA

2021 - 2024

DECLARATION

I here by declare that this project work **"Dr.BABY - VACCINATION APPLICATION"** submitted by Christ College (Autonomous)Irinjalakuda ,affiliated to Calicut University in partial fullfillment of the requirement for the award of the Bachelor of Computer Application, is a record of original work done by me,under the guidance of Mr.THOUFEEQ ANSARI,Department of Computer Science.

Place : Irinjalakuda NAVEED NIHAN K M

DEPARTMENT OF COMPUTER SCIENCE

CHRIST COLLEGE (AUTONOMOUS)

IRINJALAKUDA

CERTIFICATE

This is to certify that the project report entitled ***"Dr. Baby - Vaccination Application"*** *is a bonfied record of the project work done by* ***Naveed Nihan*** *in partial fulfillment of the requirement for the sixth semester of* ***Bachelor of Computer Application*** *in Department of Computer Science of* **CHRIST COLLEGE (AUTONOMOUS) , IRINJALAKUDA**

Ms. Thoufeeq Ansari
Assistant Professor,CS
Internal Guide

Ms. Sini Thomas
Head of the Department
Computer Science

EXTERNAL EXAMINER

INTERNAL EXAMINER

ACKNOWLEDGEMENT

First and foremost I would like to thank Lord almighty for his providence and for being the guiding light throughout the project. I wish to express my sincere gratitude to our beloved Department head for giving me all the facilities for our project. I take this opportunity to express my gratitude to the class teacher Ms. SOWMYA P.S and head of the department Ms. SINI THOMAS who has been a great support for me throughout the course of this project. I am very thankful for her aspiring guidance and valuable advice during the project work. I would like to express my sincere thanks to our project guide Mr. THOUFEEQ ANSARI for supporting and guiding throughout the project. I take this opportunity to specially thank all other faculty members for their constant and continuous motivation. Finally I would like to thank my family and friends for giving valuable advice and moral support throughout our project.

ABSTRACT

Dr. BABY is a comprehensive mobile application designed to simplify and streamline the process of managing baby vaccinations. With a user-friendly interface and intuitive features, Dr.Baby aims to alleviate the burden on parents and caregivers by providing timely reminders, educational resources, and appointment scheduling tools. The application utilizes personalized vaccination schedules based on the baby's age, ensuring adherence to recommended immunization timelines. Additionally, Dr.Baby offers real-time updates on vaccine availability and information on local healthcare providers and clinics. By promoting vaccination compliance and empowering parents with valuable knowledge, Dr.Baby strives to contribute to improved public health outcomes and safeguarding the well-being of infants worldwide.

Contents

Chapter 1

1 Introduction

We have developed an application, suitable for both iOS and androind user. Our application mainly focuses on the vaccinations mandatory for the new born babies. The application contains a calender , booking system and a chatbot. It gives out a notification when is the time for the baby's vaccination schedule. This app is designed to assist the parents of the new born with vaccination tracking.

1.1 Overview

The objective of our Dr. BABY - VACCINATION APPLICATION is to ensure that all the new born's do not miss out on the mandatory vaccinations that can be life treatening. The application includes many features that is required to assist the parents with the vaccinations, such as a chatbot, calender scheduling. It is a User friendly application.

Chapter 2

2 System Analysis

2.1 Purpose

The main purpose of this application is to make the vaccination journey of the new born's easier and effortless for the parents.

2.1.1 Existing System

There are many applications for women to keep track of periods, vaccination apps developed during the covid era and so on, but an app for the parents of the new born are still not available. Limitations of existing system :We cannot the information as well as a schedule for the vaccination for the baby.

2.1.2 Proposed System

The proposed system is an flutter - based application that is developed to help parents of the new born to track all the mandatory vaccinations. We also provide a lot of other health related services along with the vaccination scheduling.

2.2 Problem definition

To know what the problem is and what the needs are before developing it. Here we designed an application for vaccinations specially for the new-born babies.

2.3 FEASIBILITY STUDY

After the problem is clearly understood and solutions proposed, the next step is to conduct the feasibility study. Feasibility study is defined as evaluation or analysis of the potential impact of the proposed project or program. The objective is to determine whether the proposed system is feasible. There are three aspects of feasibility study to which the proposed system is subjected as discussed below.

2.3.1 Technical Feasibility

Technical feasibility assess whether the current technical resources are sufficient for the new system. We can upgrade the level of technology for supporting our application. We checked whether the proposed system can be implemented in the present system without supporting the existing hardware.

2.3.2 Economical Feasibility

Economic feasibility determines whether the time and money are available to develop the system. There is no additional hardware used to develop the site,so it is inexpensive to build.

2.3.3 Operational Feasibility

Operational feasibility determines if the human resources are available to operate the system once it has been installed. No extra training is needed to use this system. Anyone who has knowledge in internet in English language can easily use the system. The resources that are required to implement or install are already available with the organization.

Chapter 3

3 Software Requirement Specification

3.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the Dr.BABY - VACCINATION APPLICATION. It illustrate the purpose and complete description for the development of the system. It explain system constraints, interface and interactions with other external applications. This document is primarily intended to be proposed to help with vaccination journey and bookings of the vaccination trp

3.2 Scope

The application gives all the details about the vaccinaton dates and booking system.It also provides a chatbot that helps with any doubts arising to the parents.

3.3 Overall Description

Dr.BABY is a comprehensive mobile application designed to streamline and enhance the vaccination experience for parents and caregivers of infants. The app is dedicated to ensuring that babies receive timely and appropriate vaccinations, helping to protect them from various preventable diseases. It serves as a one-stop solution for managing vaccination schedules, accessing reliable information, and facilitating communication with healthcare providers.

3.3.1 Product Perspective

Dr.BABY APPLICATION is mainly used to know the details about your next vaccination trip. It also provide features that gives you a booking system as well as a chatbot that helps with all the users doubts.

3.3.2 Product Functionality

Through this system, admin can upload various data based on the vaccination help services, health advices etc. User will be able to view their vaccine schedule in a calender and will also be directed to a booking page to book the vaccine appointememt.

3.3.3 Users and Characteristics

There are two users - admin and parent. The admin can regulate everything in the system and they can add new hospitals and booking slots. The users/parent

will be able to schedule their vaccination trips as well as book vaccination slots at the nearest hospitals.

3.4 Specific Requirements

3.4.1 Hardware Requirements

- System: LAPTOP-S2BJ4TFP
- Processor: Intel Core i7-1065G7 CPU
- Speed: 1.30 GHz - 1.50GHz
- RAM capacity: 12 GB
- Hard Dsk drive: 952 GB

3.4.2 Software Requirements

- Front End : Flutter
- Back End : Python Django
- Database : MySql
- IDE : Visual Studio Code

3.5 Functional Requirements

It contains two main modules :

- 1.Admin
- 2.Parent

Admin

The admin will be able to handle the database in person. It can also add on more and more hospital location, booking area's etc. The admin also has access to user database that enable the admin to remove any malicious users.

Parent

The users/parent will be able to view the calender that has all the vaccine schedules that has been already set up according to the DOB of the child from child profile. They also have access to various other supporting services. Chatbot and booking system is few of the services that assists the user/parent for a smooth vaccinaton scheduling, booking etc.

3.6 Non Functional Requirements

Non-functional requirements define the overall qualities or attributes of the resulting system. Non-functional system place restrictions on the product being developed, the developed process, and specify external constraints that the product must meet. Examples of non-functional requirements include safety, security, usability, reliability and performance requirements. Project management issues (costs, time and schedule) are often considered as non-functional requirements. The principal non - functional constraints which are relevant to critical systems :

- performance
- security
- safety
- usability

Performance

Performance requirements concern the speed of operation of a system. Types of performance requirements :

- Response requirements (how quickly the system reacts to a user input).
- Throughput requirements (how much the system can accomplish within a specified amount of time).
- Availability requirements (is the system available for service when requested by end users). The speed of operation of this system is adequate for the requirements.

Reliability

- Reliability is the ability of a system to perform its required function under stated conditions for a specified period of time.
- constraints on the runtime behavior of the system. This system is reliable because its functionalities can be done on the required conditions.

Safety

Safety requirements are not required which exclude unsafe situation from the possible solution of the system.

Usability

Usability is the ease with which a user can learn to operate, prepare inputs for, and interpret outputs of system or components.

Usability requirements include :

- information error messages.
- well-formed user interfaces.

3.7 Interface Requirements

3.7.1 Hardware interfaces

The system must run over the internet, all the hardware shall require to connect internet will be hardware interface for the system. As for example modem, WAN-LAN.

3.7.2 Software interfaces

Software interface required for the working of the project is the appropriate operating system.

3.7.3 Communication interfaces

The user of the site communicate using the network connectivity and the data set is accessed

3.8 Security Requirements

- User accesses only their account.
- Validation of input is handled.
- This application containing the computer systems is physically secured against arms or surreptitious entry by intruders.
- Users must be authorized carefully to reduce changes of any such user giving access to an intruder in exchange for a bribe or other favour.

3.9 Platform Used

Windows 11, the latest operating system from Microsoft, introduces a refreshed and modernized user interface with a centered Start menu, rounded corners, and enhanced Snap layouts for improved multitasking. The taskbar has been redesigned, featuring a simplified and more streamlined look. Windows 11 also brings advancements in gaming with DirectStorage and Auto HDR, providing a more immersive gaming experience. The Microsoft Store has undergone a significant overhaul, offering a wider range of apps and a more user-friendly interface. Additionally, Windows 11 focuses on productivity enhancements, such as the integration of Microsoft Teams directly into the taskbar for seamless communication. With its fresh design and enhanced features, Windows 11 aims to deliver a more intuitive and enjoyable computing experience for users.

3.10 Technologies Used

Flutter

Flutter is an open-source UI software development toolkit created by Google, providing a comprehensive framework for building natively compiled applications across mobile, web, and desktop from a single codebase. Launched in 2017, Flutter has gained significant popularity among developers due to its fast development cycles, expressive and flexible UI, and native performance. Using Dart as its programming language, Flutter enables developers to create visually appealing and responsive applications that can run seamlessly on different platforms, fostering a streamlined and efficient development process. With a rich set of pre-designed widgets, a strong community support, and continuous updates, Flutter empowers developers to build high-quality, cross-platform applications that deliver a consistent user experience across various devices.

SQLite 3

SQLite is a lightweight, self-contained, and serverless relational database management system (RDBMS) that is widely used for embedded systems, mobile applications, and small to medium-sized projects. Developed as a C library, SQLite offers a simple and efficient solution for storing and retrieving data without the need for a separate server process. It operates on a single ordinary disk file and supports standard SQL syntax, providing users with a familiar relational database experience. SQLite is known for its ease of use, portability, and minimal setup requirements, making it a popular choice for developers seeking a straightforward and compact database solution for various applications.

Python Django

Django is a high-level web framework for building robust and scalable web applications using the Python programming language. Developed with the principles of simplicity, flexibility, and maintainability in mind, Django follows the Model-View-Controller (MVC) architectural pattern, known as Model-View-Template (MVT) in Django's terminology. It comes equipped with a wide range of built-in features, including an Object-Relational Mapping (ORM) system for database management, a powerful templating engine, and a comprehensive administration interface. Django encourages the implementation of clean, reusable code through its "Don't Repeat Yourself" (DRY) philosophy, enabling developers to focus on application logic rather than boilerplate code. With a vibrant and supportive community, Django has become a popular choice for web developers, offering a quick and efficient way to create feature-rich web applications.

Chapter 4

4 Design Document

4.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the Dr.BABY APPLICATION. It illustrate the purpose and complete description for the development of the system. It explain system constraints, interface and interactions with other external applications. This document is primarily intended to help the users to schedule see their upcoming vaccines. . The application provides all the details about the vaccinations, date on which the vaccines are to be taken, booking system and other services. The app also provides all the details of the vaccination to be taken.

4.2 Scope

The scope of this application can be comprehensive, addressing various aspects related to the vaccination process, healthcare, and information management for infants and toddlers.

4.3 Overview

The purpose of this document is to help the reader to visualize the solution to the project presented.This document verifies how the design meet the requirement stimulated in the SRS document through design viewpoints.The design viewpoints will cover all design elements presented before. This document will provide a direct approach to the development of this project hence reducing feature creep and ponitedly determine the quality of the design.

4.4 Data Design

Database are the storehouses of data used in the software systems.The data is stored in tables inside the database.Several tables are created for the manipulation of the data for the system.Two essential settings for a database are

- Primary Key- The field that is unique for all the record occurrence.
- Foreign Key- The field used to set relation between tables.

Normalization

- Normalization is the process of organizing the data in the database.
- Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate the undesirable characteristics like Insertion, Update and Deletion Anomalies.

- Normalization divides the larger table into the smaller table and links them using relationship.
- The normal form is used to reduce redundancy from the database table.

Tables

Name	DataType	Constraints	Description
UserID	int(11)	Primary key	ID of users
username	varchar(50)	Not null	Name of users
password	varchar(250)	Not null	Password of users
email	varchar(100)	Not null	Email of the users
FirstName	varchar(200)	Not null	First name of user
LastName	varchar(20)	Not null	Last name of user
Phone	varchar(50)	Not null	Contact number
Role	varchar(250)	Not null	Relation to baby

Table 1: User Account

Name	DataType	Constraints	Description
BabyID	int(11)	Primarykey	ID of baby
UserID	int(9)	Foreignkey	ID from User table
FirstName	varchar(200)	Notnull	First name of baby
LastName	varchar(20)	Notnull	Last name of baby
Date of Birth	date()	Notnull	Babies date of birth
Gender	char(1)	Notnull	Gender of the baby
Location	varcher(5)	Notnull	Place of baby

Table 2: Babies Profile

Name	DataType	Constraints	Description
VaccineID	int(11)	Foreignkey	id from vaccine name table
VaccineName	varchar(50)	Notnull	Name of vaccines
Vaccine_Available	varchar(50)	Notnull	Available Vaccines

Table 3: Vaccine Table

Name	DataType	Constraints	Description
Vaccine_Status_ID	int(11)	Primarykey	id for vaccine status table
VaccineName	varchar(50)	Foreignkey	Reference to vaccine name table
BabyID	int(11)	Foreignkey	Id of the baby,refernce to baby profile

Table 4: Vaccine Status

Name	DataType	Constraints	Description
Vaccine_Booking_ID	int(11)	Primarykey	Id for vaccine booking table
ParentName	varchar(50)	Foreignkey	Reference to user profile table
Parent_Email	varchar(50)	Notnull	Email for vaccine booking
Hospital	varchar(50)	Foreignkey	Rreference to hospital table
Vaccine_Available	varchar(50)	Foreignkey	Reference to vaccine table
Booking_Date	date()	Notnull	Date of vaccine

Table 5: Vaccine Booking

Name	DataType	Constraints	Description
Hospital_ID	int(11)	Primarykey	Id for hospital table
Hospital	varchar(50)	Notnull	Hospital name
Slots_Available	varchar(50)	Notnull	Vaccine Slots
Vaccine_Available	varchar(50)	Foreignkey	Reference to vaccine table

Table 6: Hospital Table

Chapter 5

5 Development of the System

The development of the baby vaccination app involves a comprehensive approach to ensure the health and well-being of infants. The app is designed to streamline and simplify the vaccination process, providing parents with a user-friendly platform to keep track of their baby's immunization schedule. It includes features such as personalized vaccination reminders, informative content about each vaccine, and a secure digital record of the child's vaccination history.Development considerations encompass data security, user accessibility, and collaboration with healthcare professionals to ensure accurate and reliable information.

Chapter 6

6 System Testing

Testing is the penultimate step of software development.An elaborate testing of the data is prepared and the system is using the test data,while doing testing,errors are noted and correction is made.The users are trained to operate the developed system.Both hardware and software securities are made to run the developed system successfully.System testing is aimed at ensuring the system works accurately before the live operation commences.

6.1 Test Plan

6.1.1 Scope

This test plan will cover the following testing activities as identified in the testing strategy.

- White Box Testing

White box testing is a testing technique, that examines the program structure and derives test data from the program logic/code. In white box testing, the UI is bypassed. Inputs and outputs of ratings are tested directly at the code level and the results are compared against specifications. It Reveals errors in "hidden" code.

- Black Box Testing

Black - box testing is a method of software testing that examines the functionality of an application based on the specifications. Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end - user would.

- Unit Testing

The module includes four main components to undergo unit testing. It checks the syntax error, logic error and validity of the program correctness. This test will be performed by the developers.

- Integration Testing

After all components pass the unit testing, they get to be tested whether they work correctly when they are running concurrently and communicating to each other. Integration tests exercise an entire subsystem and ensure that a set of components play nicely together.

- Internal Data Testing

We will test the validity of the data before it enters the database to avoid any problems that may face us in the database. We will test the encryption of the personal information of all the users along with the user names and passwords to ensure maximum security of the users privacy.

6.1.2 Software risk issues

In this section, the plan is to test the risk involved in critical issues such as: Difficult to run on visual studio due to large loading time. Some inherent software risks such as complexity exist, Proper network connection,Working of SQLite 3 database.

6.1.3 Features to be tested

- Test whether correct user name and password allows you to login.
- Test whether invalid user name and password prevents you from login.
- Test whether there is any connection problem in the Server.
- Test whether the parent and admin details are entered correctly.
- Test whether invalid data entry allows saving data successfully.
- Test whether all pages are loaded correctly.
- Test whether the provided security works correctly.

6.2 Test consolidation

6.2.1 Test item

The items or features to be tested in the test cases are included in the document. Each and every user input is tested. The present condition of the system is tested. It is checked to make sure that environment is ready for the application to work.

6.2.2 Input specifications

These are the inputs required to execute the test case. All required inputs including data elements and values are given below.

Data	Values
Email	A valid email ID
Password	Combination of characters and numbers
Phone number	10 Digit number

Chapter 7

7 System Implementation and Maintenance

7.1 Implementation

System implementation is the conversion of new system into an operating one which involves creating compatible files, training clients and installing hardware. User training is crucial for minimizing resistance to change and giving chance to prove its worth. Training aids user friendly manuals and healthy screens provide the user with a good start. Software maintenance follows conversion to the extent that changes are necessary to maintain satisfactory operations relative to changes in the user's environment. Maintenance often includes minor enhancements or corrections to the problem that surface late in the systems operations. In the implementation phase, the team builds the components either from scratch or by composition. Given the architecture document meant from the design phase and the requirement document from the analysis phase, the team should build exactly what has been requested, though there is still room for innovation and flexibility. For example, a component may be narrowly designed for this particular system, or the component may be made more general to satisfy a reusability

- Careful planning
- Investigation of system and constraints
- Design the methods to achieve changeover.
- Training the staff in the changed phase.
- Evaluation of change over method.
- The method of implementation and time scale to be adopted are found out initially.

7.2 Maintenance

This phase occurs as a result of deploying the whole system at the end users organization. They will perform the beta testing at the end users and inform to the developers about any needed modification to the application. The customer records all the problems that are encountered during the beta testing and reports these to the developer at regular intervals.

7.2.1 Corrective Maintenance

Even with the best quality assurance activities, it is likely that the customer will uncover defects in the software. Corrective maintenance changes the software to correct the defects. Corrective Maintenance activity may consist of repair,

restoration or replacement of equipment. This activity will be result of a regular inspection, which identifies the failure in time for corrective maintenance to be planned and scheduled, then performed during a routine maintenance shutdown.

7.2.2 Adaptive Maintenance

Over time, the original environment(CPU, operating system, business rules, external product characteristics) for which the software was developed is likely to change. Adaptive maintenance results in modification to the software to accommodate changes to its external environment.

7.2.3 Enhanced Maintenance

As software is used, the parent/user will recognize additional functions that will provide the benefit. Perfect maintenance extends the software beyond its original functional requirements.

7.2.4 Preventive Maintenance

Computer software deteriorates due to change, and because of this preventive maintenance often called software re-engineering, must be conducted to enable the software to serve the needs of its end users. Preventive maintenance makes changes to computer programs so that they can be more easily corrected, adapted and enhanced.

Chapter 8

8 Conclusion and Future Scope

8.1 Conclusion

In conclusion, the baby vaccination app represents a pivotal tool in the realm of child healthcare, offering a streamlined and technologically advanced solution for parents and healthcare providers. As the app continues to evolve, its future scope extends beyond the confines of vaccination tracking, encompassing personalized health records, AI-driven predictive analysis, telemedicine integration and etc. By fostering a sense of community, providing real-time health monitoring, and adhering to robust security measures, the app stands poised to contribute significantly to global immunization efforts and enhance the overall well-being of children. As we navigate the ever-changing landscape of healthcare technology, the baby vaccination app serves as a beacon of innovation and empowerment, bridging the gap between parents and healthcare, ultimately ensuring a healthier and brighter future for the youngest members of our society.

8.2 Future Scope

Here are some potential future developments:

- Personalized Health Records:

The app could expand to include a broader range of health information, allowing parents to maintain a comprehensive digital health record for their child. This could include growth charts, developmental milestones, and other relevant healthcare data.

- AI Integration for Predictive Analysis:

Incorporating artificial intelligence (AI) could enable predictive analysis based on health data, helping parents and healthcare providers anticipate potential health issues and take preventive measures.

- Telemedicine Integration:

Facilitating telemedicine consultations within the app could allow parents to connect with healthcare professionals for advice or consultations without physically visiting a clinic, especially in remote or underserved areas.

- Global Immunization Tracking:

Collaborating with healthcare organizations and governments to contribute to global efforts in tracking and managing vaccination coverage, aiding in the prevention of the spread of vaccine-preventable diseases.

Appendix

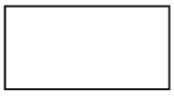
A Data Flow Diagram

Data flow is the one of the best way of documenting the entire functionality of the system.For the system ,which will have data flows in and have some processing inside and then some data flow out from the system can be documented or represented effectively by means of data flow out from the system can be documented or represented effectively by means of data flow diagrams. The data flow diagram are a diagrammatic representation of the system,which has input,process and output.Once any system is represented using a data flow diagram we can identify the following things easily:

- Various entities interacting with the system are identified
- Flow of data from one entity to another is identified
- The various processes involved in between the interaction of two or more entities in the system are clearly pointed out
- The various data stores which hold the data in between the process,are clearly identified

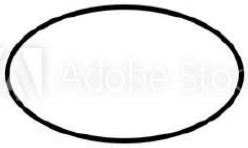
Some Data Flow Diagram charting forms:

A.1 External source or receiver



A source or sink is a person or part of organization ,which enters or receives information from the system,but is considered to be outlining the contest of data flow model.

A.2 Transform process

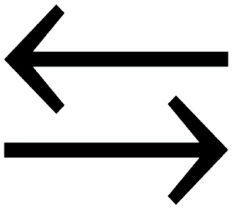


A process represents transformation where incoming data flowsare changed into outgoing data flow

A.3 Data Store

A data store is repository of data that is to be stored for use by one or more process may be as simple as buffer or queue or sophisticated as relational database. They should have clear names.If a process merely uses the contest of store and does not alter it,the arrowhead goes only form the store to the process. If a process alters the details in the store then double-headed arrow is used.

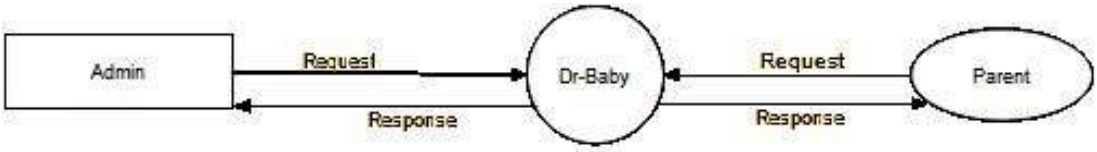
A.4 Data flow



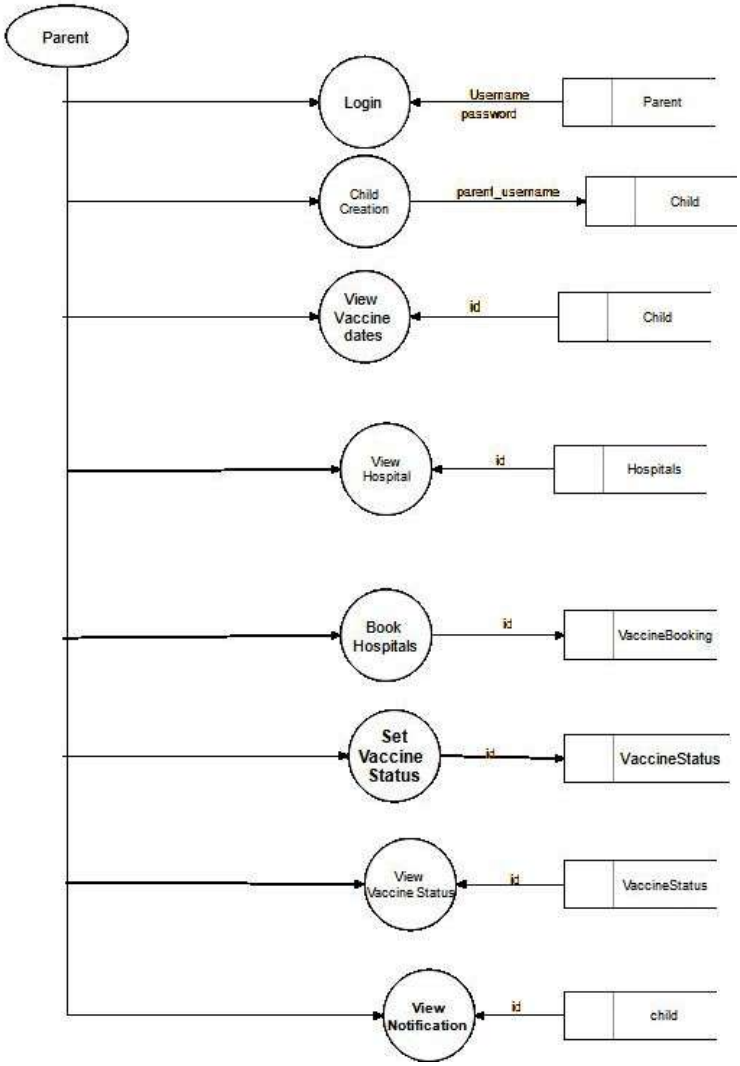
A data flow is a route, which enable packets of data to travel from one point to another.Data may flow,with arrowhead pointing in the direction of the flow

B Data Flow Diagrams

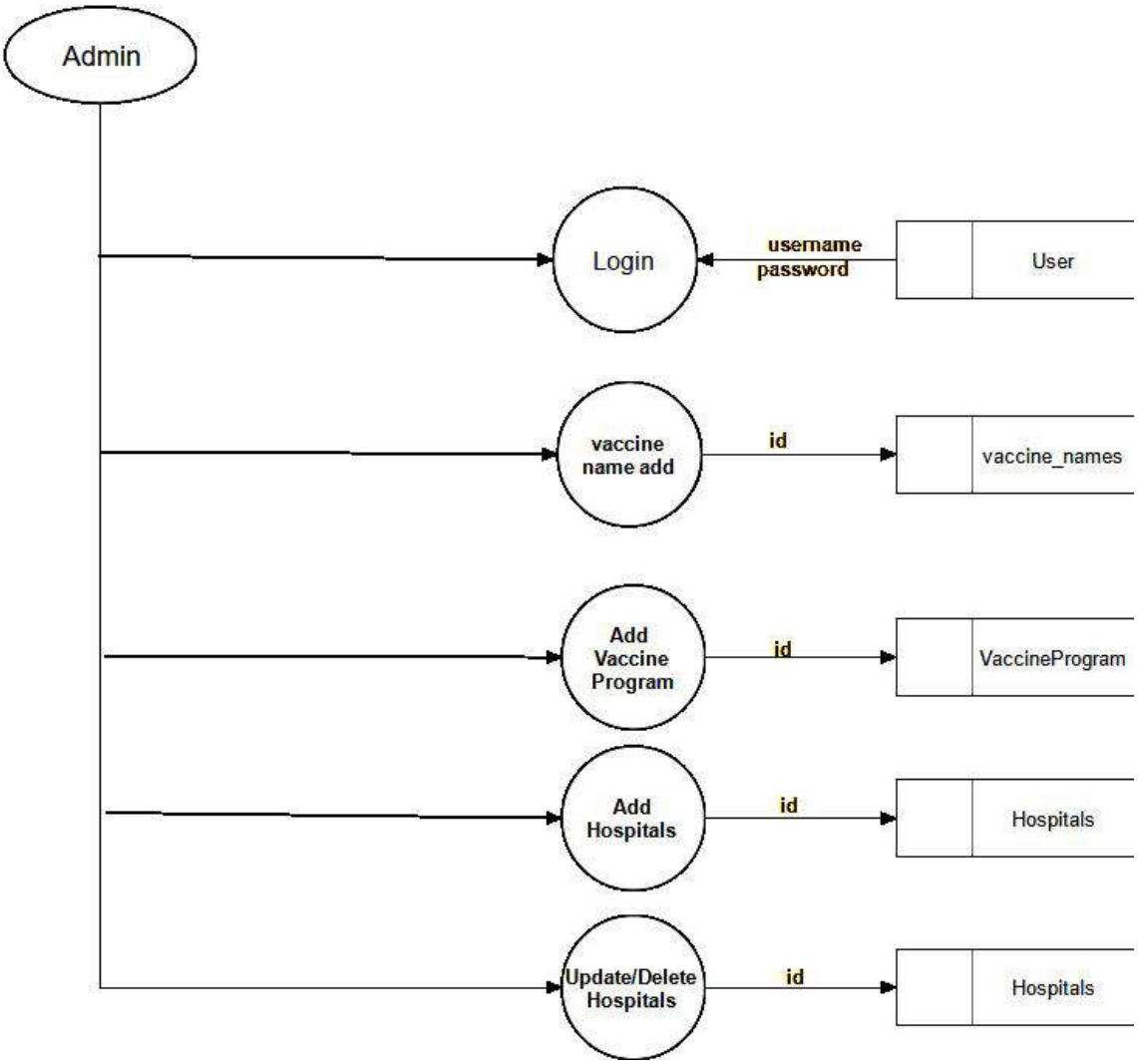
B.1 Level 0



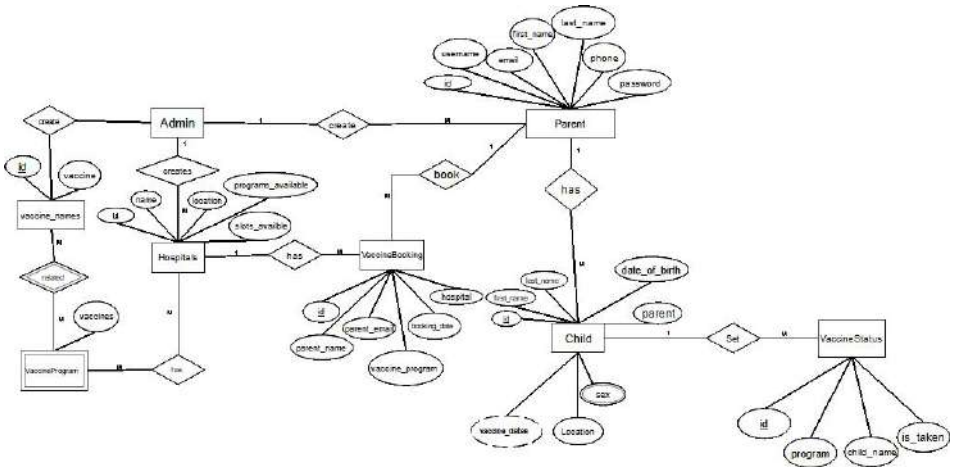
B.2 Level 1



B.3 Level 2

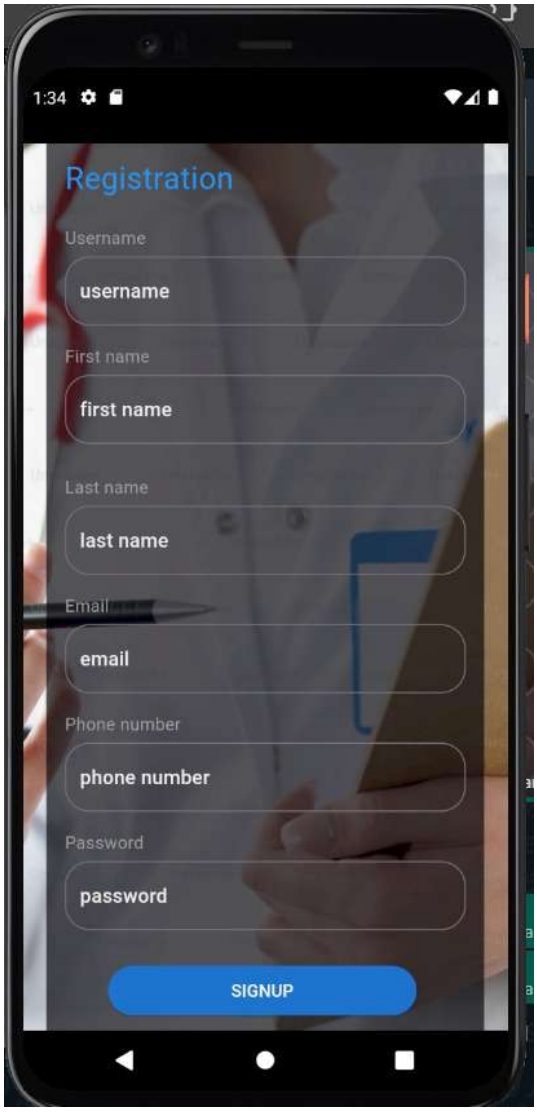

Admin
Login
username
password
User
vaccine
name add
id
vaccine_names
Add
Vaccine
Program
id
VaccineProgram
Add
Hospitals
id
Hospitals
Update/Delete
Hospitals
id
Hospitals

B.4 E-R Diagram

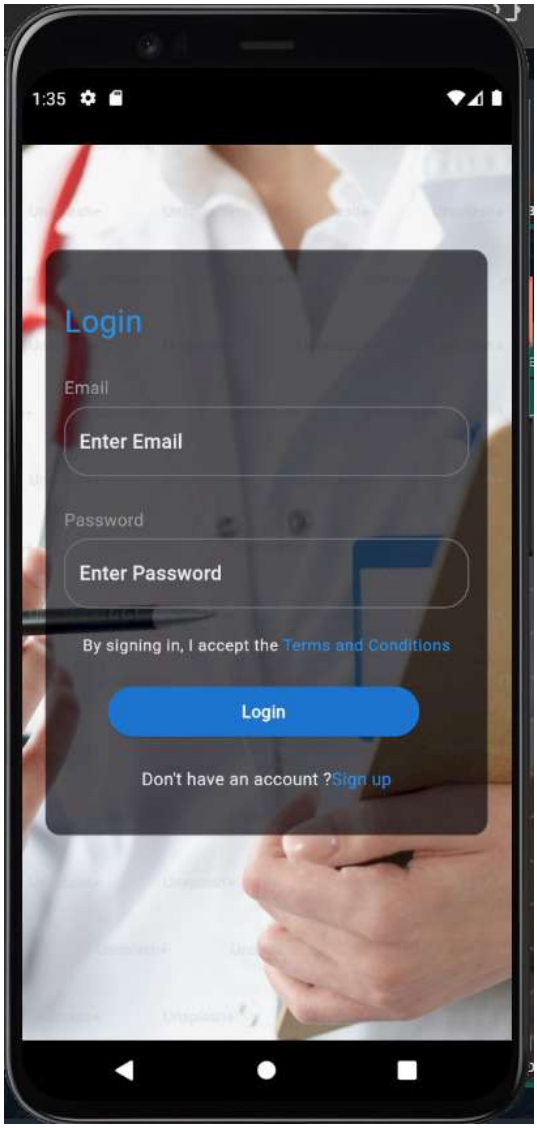


C Interface

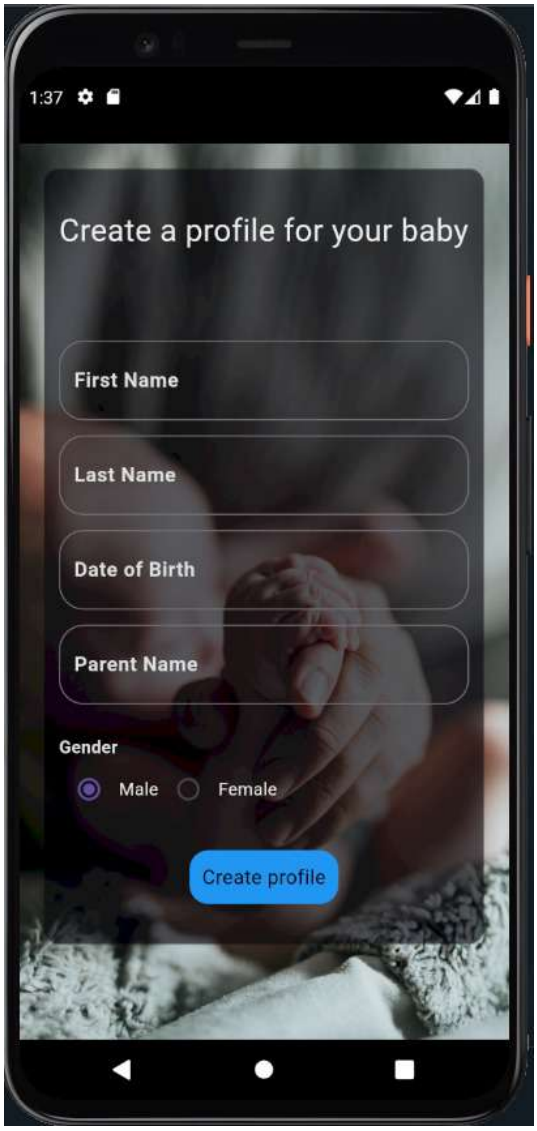
C.1 Registration



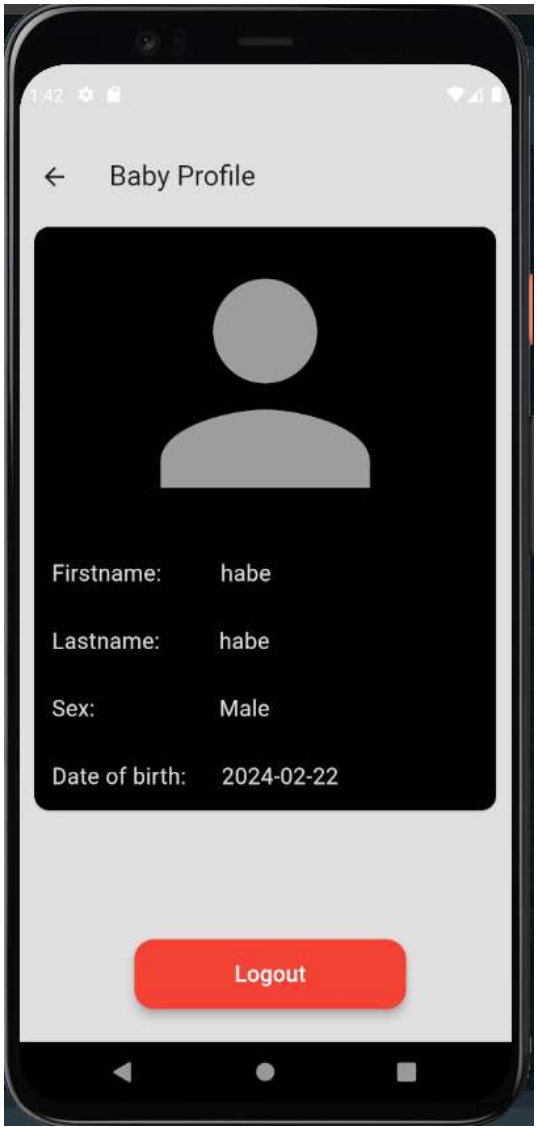
C.2 Login


1:35
Login
Email
Enter Email
Password
Enter Password
By signing in, I accept the Terms and Conditions
Login
Don't have an account ?Sign up

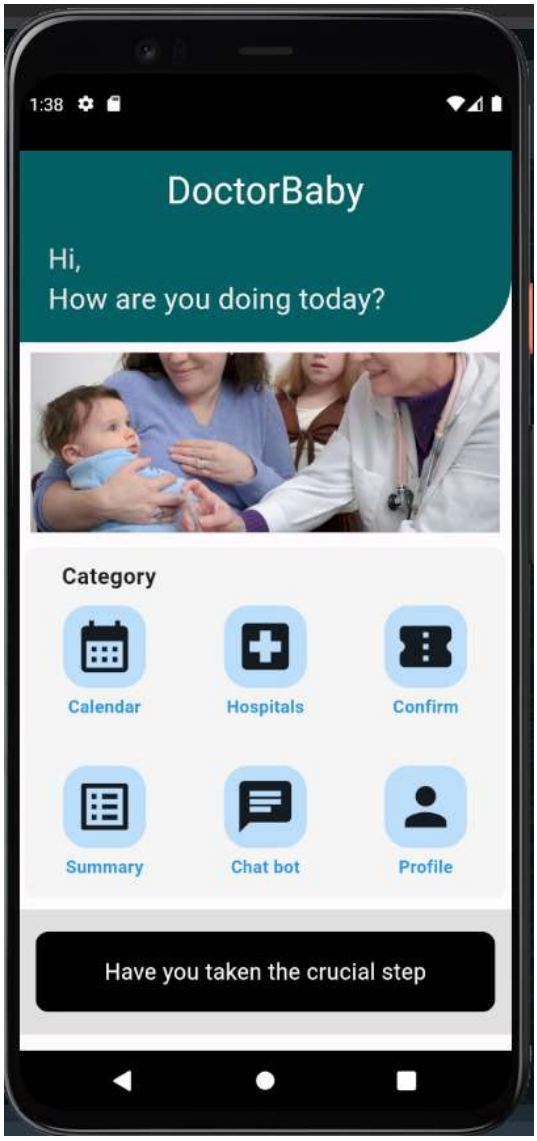
C.3 Profile


1:37
Create a profile for your baby
First Name
Last Name
Date of Birth
Parent Name
Gender
Male
Female
Create profile

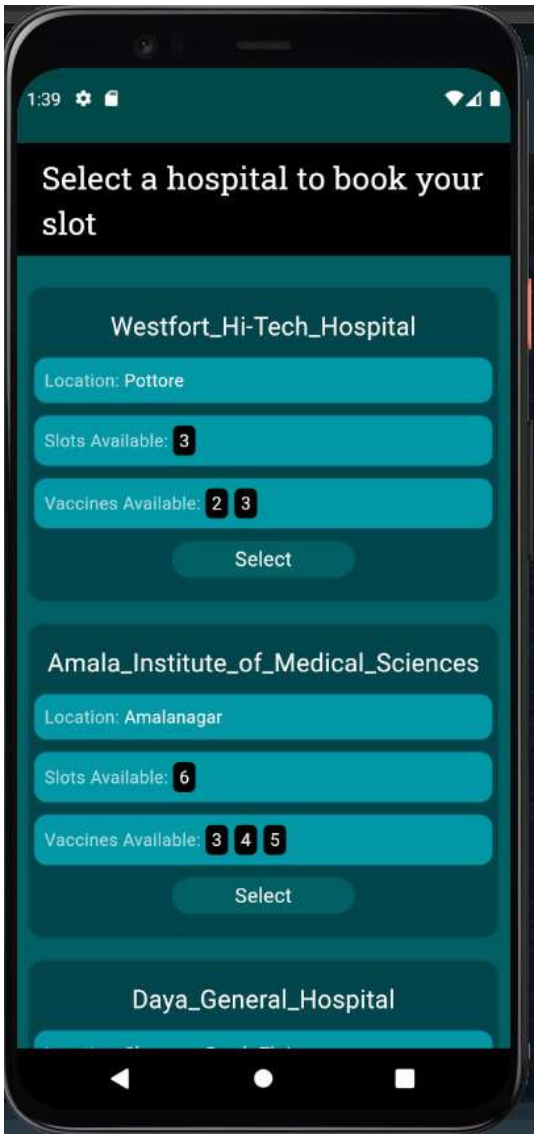
C.4 User Profile



C.5 Home


1:38
DoctorBaby
Hi,
How are you doing today?
Category
Calendar
Hospitals
Confirm
Summary
Chat bot
Profile
Have you taken the crucial step

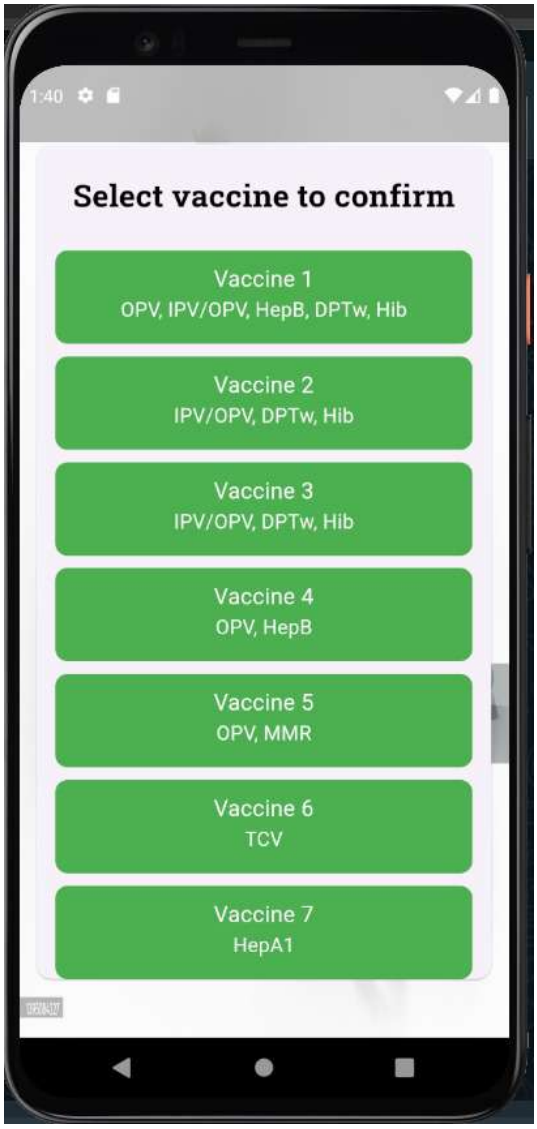
C.6 Hospital


1:39
Select a hospital to book your slot
Westfort_Hi-Tech_Hospital
Location: Pottore
Slots Available: 3
Vaccines Available: 2 3
Select
Amala_Institute_of_Medical_Sciences
Location: Amalanagar
Slots Available: 6
Vaccines Available: 3 4 5
Select
Daya_General_Hospital

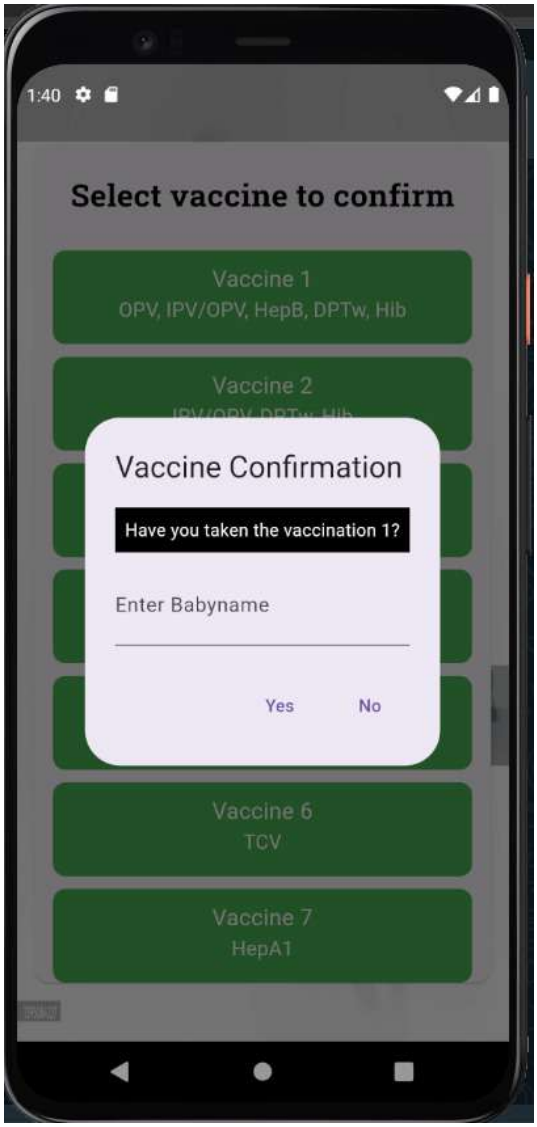
C.7 Calendar


1:38
Calendar
February 2024
2 weeks
Sun Mon Tue Wed Thu Fri Sat
28 29 30 31 1 2 3
4 5 6 7 8 9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 1 2
Vaccine Dates:
2/4/2024
29/4/2024
2/5/2024
21/5/2024

C.8 Vaccine Confirm

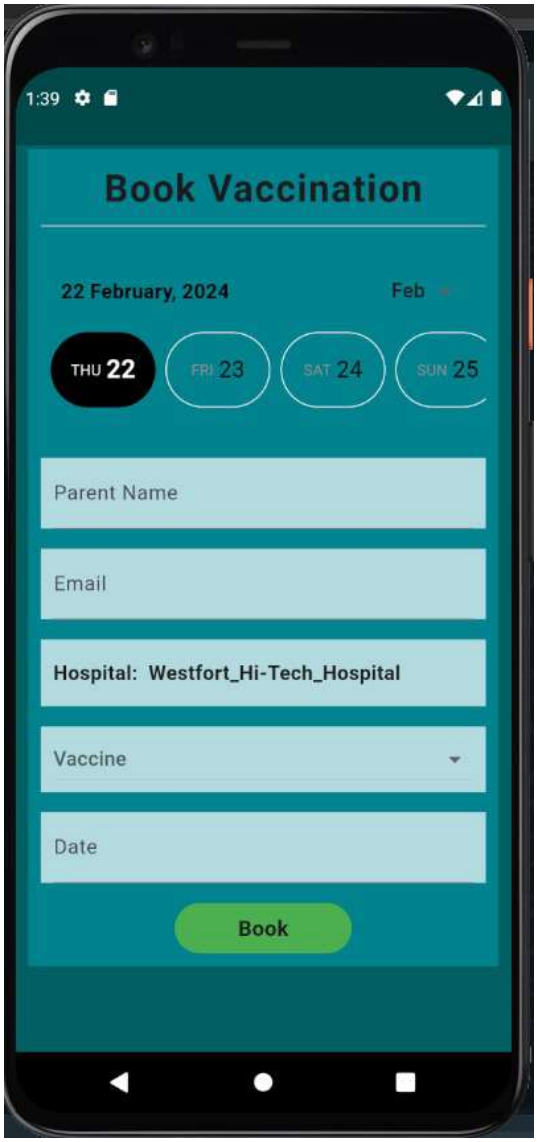


C.9 Confirmation



C.10 Chatbot

C.11 Vaccine Booking



D CODE

tasks.py

```
from celery import shared_task
from django.contrib.auth.models import User
from django.utils import timezone
import datetime
from datetime import timedelta
from django.core.mail import send_mail
from babyvaccinepro import settings
from .models import *

from datetime import datetime, timedelta
from django.utils import timezone




@shared_task(bind=True)
def send_mail_based_on_dates(self):
    recently_registered_user = User.objects.order_by
    ('-date_joined').first()

    if recently_registered_user:
        recent_user_email = recently_registered_user.email
        children = Child.objects.filter
        (parent=recently_registered_user)

        for child in children:
            # Map health review dates to corresponding program
               IDs based on child's date of birth
            health_review_program_mapping = {
                (child.date_of_birth + timedelta(days=40)): [1],
                (child.date_of_birth + timedelta(days=67)): [2],
                (child.date_of_birth + timedelta(days=70)): [3],
                (child.date_of_birth + timedelta(days=89)): [4],
                (child.date_of_birth + timedelta(days=180)): [5],
                (child.date_of_birth + timedelta(days=304)): [6],
                (child.date_of_birth + timedelta(days=363)): [7],

            }

            for rev_date, program_ids in health_review_program
   _mapping.items():
```

```
                if rev_date == timezone.localtime(timezone.
  now()).date():
                    user = child.parent
                    mail_subject = "Health Review Date Reminder"

                    # Filter programs based on the list of
      program IDs
                    programs = VaccinePrograms.objects.filter(
  id__in=program_ids)

                    # Construct the message with vaccines and
                        their names
                    message = f"Hi {user.username}, This is
   a reminder for a health review appointment today
      for {child.first_name}.\n"

                    if programs.exists():
                        for program in programs:
                            message += f"\nVaccines Are\n"

    # Use all() to get all related vaccines
                            for vaccine_info in program.vaccines.all():
                                        vaccine_name = vaccine_
    info.vaccine
                                        message += f"{vaccine_name}\n"

                    else:
                        message += "No relevant vaccination
       programs found for this appointment."

                    to_email = recent_user_email
                    send_mail(
                        subject=mail_subject,
                        message=message,
                        from_email=settings.EMAIL_HOST_USER,
                        recipient_list=[to_email],
                        fail_silently=True,
                    )
                    return "done"
```

views.py

```
from django.contrib.auth import authenticate, login, logout
from django.shortcuts import render
from rest_framework.views import APIView
from .serializer import *
from rest_framework.response import Response
```

```
from rest_framework import generics
from django.db import transaction

from rest_framework import status

from rest_framework.authentication import BasicAuthentication,
TokenAuthentication

from rest_framework.authtoken.models import Token
from django.contrib.auth.hashers import check_password
from django.http import HttpResponse

from django.core.mail import send_mail
from babyvaccinepro.settings import EMAIL_HOST_USER

from django.shortcuts import get_object_or_404
from django.http import JsonResponse

from chatterbot import ChatBot
from chatterbot.trainers import ListTrainer
from PyPDF2 import PdfReader
from langchain.embeddings.openai import OpenAIEmbeddings
from langchain.text_splitter import CharacterTextSplitter
from langchain.vectorstores import FAISS
from langchain.chains.question_answering import load_qa_chain
from langchain.llms import OpenAI
import os



#register parent
class Registeruser(APIView):
    def post(self, request):
        serializer = UserSerializer(data=request.data)
        if serializer.is_valid():
            user = serializer.save()
            # Create a token for the registered user
            token, created = Token.objects.get_or_create(user=user)

            # Get the email of the registered user
            user_email = user.email

            # Your email sending logic using the user's email
            subject = 'Welcome to Dr.baby'
            message = f'Congratulations,\n' \
```

```
                    f'You have successfully registered with our
            website.\n' \
                    f'username: {user.email}\n' \
                    f'WELCOME'

            send_mail(subject, message, settings.EMAIL_HOST_USER,
   [user_email], fail_silently=False)

            # Assuming token creation or any other
               response data you want to send back
            return Response({'data': serializer.data, 'token':
    token.key}, status=status.HTTP_201_CREATED)

        return Response(serializer.errors, status=status.
        HTTP_400_BAD_REQUEST)


#user login

class LoginView(APIView):
    serializer_class = loginserializer

    def post(self, request):
        serializer = self.serializer_class(data=request.data)

        if serializer.is_valid():
            email = serializer.validated_data['email']
            password = serializer.validated_data['password']
            user = authenticate(request, email=email, password
   =password)
            user = User.objects.get(email=email)

            if user is not None and check_password(password,
   user.password):
                login(request,user)
                return Response({'message': 'Login successful'},
                status=status.HTTP_200_OK)
            else:
                return Response({'message': 'Invalid credentials'},
                status=status.HTTP_401_UNAUTHORIZED)

        return Response(serializer.errors, status=status.
        HTTP_400_BAD_REQUEST)


class logoutview(APIView):
```

```
    def post(self,request):
        logout(request)
        return Response({'msg':'logout successfully'})


#child details get and post
class ChildListCreateView(generics.ListCreateAPIView):
    queryset = Child.objects.all()
    serializer_class = ChildSerializer

#child details delete and update

class ChildDetailView(generics.RetrieveUpdateDestroyAPIView):
    queryset = Child.objects.all()
    serializer_class = ChildSerializer


class VaxNameListCreateView(generics.ListCreateAPIView):
    queryset = VaxName.objects.all()
    serializer_class = VaxNameSerializer

class VaxNameDetailView(generics.RetrieveUpdateDestroyAPIView):
    queryset = VaxName.objects.all()
    serializer_class = VaxNameSerializer
```

```
from .tasks import *
from django_celery_beat.models import PeriodicTask,CrontabSchedule
import json


#celery function to send mail

def send_mail_to_parent(request):
    send_mail_based_on_dates.delay()
    return JsonResponse({"message": "Email sending initiated."},
    status=200)

# class SendMailToParentView(APIView):
#     def post(self, request):
```

```
#         send_mail_based_on_dates.delay()
#         return Response({"message": "Email sending initiated."},
           status=status.HTTP_200_OK)


# class SendMailToParentView(APIView):
#     def get(self, request):
#         send_mail_based_on_dates.delay()
#         return Response({"message": "Email sending initiated."}
           , status=status.HTTP_200_OK)
```

```
class VaxCycleAPIView(generics.ListCreateAPIView):
   queryset = Vax_Cycle.objects.all()
   serializer_class = VaxCycleSerializer


class VaxCycleDelete_Update(generics.RetrieveUpdateDestroyAPIView):
    queryset = Vax_Cycle.objects.all()
    serializer_class = VaxCycleSerializer

class VaxAPIView(generics.ListCreateAPIView):
   queryset=Vax.objects.all()
   serializer_class=VaxSerializer

class VaxDelete_Update(generics.RetrieveUpdateDestroyAPIView):
   queryset=Vax.objects.all()
   serializer_class=VaxSerializer
```

```
#vaccine dates


def vaccination_dates_view(request, child_id):
    child = get_object_or_404(Child, pk=child_id)
    vaccination_dates = child.get_vaccination_dates()
```

```
    # Example: Convert vaccination dates to strings for
       JSON response
    vaccination_dates_str = [str(date) for date in vaccination
    _dates]

    return JsonResponse({'vaccination_dates': vaccination_
    dates_str})



class ChatbotAPI(APIView):
    def __init__(self):
        super().__init__()
        # Read PDF and initialize necessary components
        os.environ["OPENAI_API_KEY"] = "sk-y5f7syPdWNeTUUSrDZteT
        3BlbkFJW1Xes6rz21Zmo0hwDxxl"
        pdfreader = PdfReader(r"C:\flutter\backend2\dr-babyvaccine
        \Dr.baby.pdf")
        raw_text = ''
        for page in pdfreader.pages:
            content = page.extract_text()

            if content:
                raw_text += content

        text_splitter = CharacterTextSplitter(
            separator="\n",
            chunk_size=800,
            chunk_overlap=200,
            length_function=len,
        )
        texts = text_splitter.split_text(raw_text)

        embeddings = OpenAIEmbeddings()
        self.document_search = FAISS.from_texts(texts, embeddings)

        self.chain = load_qa_chain(OpenAI(), chain_type="stuff")

    def post(self, request):
        user_input = request.data.get('user_input')

        if user_input.lower() == 'exit':
            return Response({"response": "Goodbye!"},
            status=status.HTTP_200_OK)
```

```
        # Your existing response logic
        bot_response = self.get_response(user_input)
        return Response({"response": bot_response}, status=
        status.HTTP_200_OK)

    def get_response(self, user_input):
        if user_input.lower() in ["hi", "hello", "hey", "hy", "hai"]:
            return "Hello, welcome to Dr Baby. How can I
            assist you today!"
        elif user_input.lower() in ["bye", "by", "thank you",
      "thanks"]:
            return "bye"
        else:
            docs = self.document_search.similarity_search
           (user_input)
            return self.chain.run(input_documents=docs, question
   =user_input)


#vaccine names

class VaccineListView(APIView):
    def post(self,request):
        a=VaccineNameSerializer(data=request.data)
        if a.is_valid():
            a.save()
        return Response(a.data)
    def get(self,request):
        qs=vaccine_names.objects.all()
        a=VaccineNameSerializer(qs,many=True)
        return Response(a.data)



class VaccineProgramsListCreateView(generics.ListCreateAPIView):
    queryset = VaccinePrograms.objects.all()
    serializer_class = VaccineProgramSerializer

class VaccineProgramsDetailView(generics.Retrieve
UpdateDestroyAPIView):
    queryset = VaccinePrograms.objects.all()
    serializer_class = VaccineProgramSerializer
```

```
class HospitalsAPIView(APIView):
    def get(self, request, *args, **kwargs):
        # Filter hospitals with available slots
        hospitals = Hospitals.objects.filter(slots_available__gt=0)
        serializer = HospitalsSerializer(hospitals, many=True)
        return Response(serializer.data)

    def post(self, request, *args, **kwargs):
        serializer = HospitalsSerializer(data=request.data)

        if serializer.is_valid():
            # Save Hospitals instance
            hospital_instance = serializer.save()

            # If vaccine_names data is present, add it to the
               hospital_instance
            vaccine_names_data = request.data.get('programs_
            available', [])
            if vaccine_names_data:
                # Retrieve existing VaccinePrograms instances based
                   on provided IDs
                programs_instances = VaccinePrograms.objects.filter
                (id__in=vaccine_names_data)

                # Add the existing VaccinePrograms instances to
                   hospital_instance
                hospital_instance.programs_available.set(
                programs_instances)

                # Get the vaccine names in the response
                vaccine_names_response = [vaccine.vaccine for
                program in programs_instances for vaccine in
                program.vaccines.all()]
                serializer.data['vaccine_names'] = vaccine_names_
                response

            return Response(serializer.data, status=status.
            HTTP_201_CREATED)

        return Response(serializer.errors, status=status.
        HTTP_400_BAD_REQUEST)
```

```
class VaccineBookingList(APIView):
    def get(self, request):
        bookings = VaccineBooking.objects.all()
        serializer = VaccineBookingSerializer(bookings, many=True)
        return Response(serializer.data)

    def post(self, request):
        serializer = VaccineBookingSerializer(data=request.data)

        if serializer.is_valid():
            hospital_id = request.data.get('hospital')
            program_id = request.data.get('vaccine_program')

            try:
                hospital = Hospitals.objects.get(pk=hospital_id)
                program = VaccinePrograms.objects.get(pk=program_id)
            except (Hospitals.DoesNotExist, VaccinePrograms.
   DoesNotExist):
                return Response({'message': 'Hospital or vaccine
                program not found'}, status=status.
        HTTP_404_NOT_FOUND)

            if hospital.slots_available > 0 and program in
            hospital.programs_available.all():
                with transaction.atomic():
                    hospital.slots_available -= 1
                    hospital.save()

                    serializer.validated_data['hospital'] = hospital
                    serializer.validated_data['vaccine_program']
  = program
                    vaccine_booking = serializer.save()

                    send_booking_confirmation_mail(vaccine_
                     booking.parent_email, vaccine_booking.
            hospital.name,
                     vaccine_booking.vaccine_
                     program.id)

                return Response({'message': 'Booking successful'},
                status=status.HTTP_201_CREATED)
            else:
                return Response({'message':
                'Invalid booking request'},
                status=status.HTTP_400_BAD_REQUEST)
```

```
        return Response(serializer.errors, status=
        status.HTTP_400_BAD_REQUEST)

def send_booking_confirmation_mail(parent_email, hospital_name,
 vaccine_program_id):
    vaccine_program = VaccinePrograms.objects.get(
    pk=vaccine_program_id)

    subject = 'Vaccine Booking Confirmation'
    message = f'Thank you for booking the vaccine program at
    {hospital_name}!\n'
    message += 'Vaccines included:\n'
    for vaccine in vaccine_program.vaccines.all():
        message += f'- {vaccine}\n'

    from_email = 'amrithababy142@gmail.com'
    # Replace with your email
    recipient_list = [parent_email]

    send_mail(subject, message, from_email, recipient_list,
    fail_silently=False)

    return Response({'message': 'Booking successful'},
   status=status.HTTP_201_CREATED)




from django.http import Http404

class VaccineProgramsAPI(APIView):
    def get_object(self, pk):
        try:
            return VaccinePrograms.objects.get(pk=pk)
        except VaccinePrograms.DoesNotExist:
            raise Http404

    def get(self, request, pk=None, format=None):
        if pk:
            # Get a specific vaccine program and its status
            vaccine_program = self.get_object(pk)
            statuses = VaccineStatus.objects.filter(
             program=vaccine_program)

            data = []
```

```
            for status in statuses:
                serializer_status = VaccineStatusSerializer(status)
                program_data = {
                    'program': VaccineProgramSerializer
                    (vaccine_program).data,
                    'statuses': [
                        {
                            'status': {
                                'id': status.id,
                                'program': status.program.id,
                                'child_name': status.
                                 child_name.first_name,
                                'is_taken': status.is_taken
                            }
                        }
                    ]
                }
                data.append(program_data)

            if not data:
                return Response({'is_taken': False})

            return Response(data)
        else:
            # Get status for all vaccine programs
            vaccine_programs = VaccinePrograms.objects.all()
            data = []

            for program in vaccine_programs:
                statuses = VaccineStatus.objects.filter(program
                =program)
                program_data = {
                    'program': VaccineProgramSerializer(program)
                    .data,
                    'statuses': []
                }

                for status in statuses:
                    status_data = {
                        'status': {
                            'id': status.id,
                            'program': status.program.id,
                            'child_name': status.child_name.
                            first_name,
                            'is_taken': status.is_taken
                        }
```

```
                }
                program_data['statuses'].append(status_data)

            data.append(program_data)

        return Response(data)

def post(self, request, pk=None, format=None):
    if pk:
        # Update the status of a specific vaccine program
        vaccine_program = self.get_object(pk)
        child_name = request.data.get('child_name')
      # Retrieve the child name from the request data

        try:
            child = Child.objects.get(first_name=child_name)
         # Retrieve the Child instance using the name
        except Child.DoesNotExist:
            return Response({'error': f'Child with name
            {child_name} not found'}, status=status
            .HTTP_404_NOT_FOUND)

        status, created = VaccineStatus.objects.get_or_create
        (program=vaccine_program, child_name=child)

        # Use 'is_taken' as the field name, and ensure
            it's a boolean
        is_taken = request.data.get('is_taken', False)
        is_taken = is_taken.lower() == 'true'
        # Convert to boolean if needed

        status.is_taken = is_taken
        status.save()

        # Check if the vaccine is taken and return
           the appropriate status
        result_status = 'Taken' if status.is_taken else
        'Pending'

        return Response({'status': result_status})
    else:
        # Create a new vaccine status with child details
        return Response({'error':'Invalid request'})
```

```
class HospitalDetailAPIView(APIView):
    def get_object(self, pk):
        try:
            return Hospitals.objects.get(pk=pk)
        except Hospitals.DoesNotExist:
            raise Http404

    def get(self, request, pk, format=None):
        hospital = self.get_object(pk)
        serializer = HospitalsSerializer(hospital)
        return Response(serializer.data)

    def put(self, request, pk, format=None):
        hospital = self.get_object(pk)
        serializer = HospitalsSerializer(hospital,
        data=request.data)
        if serializer.is_valid():
            serializer.save()
            return Response(serializer.data)
        return Response(serializer.errors, status=status.
        HTTP_400_BAD_REQUEST)

    def delete(self, request, pk, format=None):
        hospital = self.get_object(pk)
        hospital.delete()
        return Response(status=status.HTTP_204_NO_CONTENT)



class ChildVaccineStatusAPI(APIView):
    def get(self, request, child_name, format=None):
        try:
            child = Child.objects.get(first_name=child_name)
        except Child.DoesNotExist:
            return Response({'error': f'Child with name
           {child_name} not found'}, status=status.HTTP_404_NOT_FOUND)

        # Get all VaccineStatus entries for the given child
        statuses = VaccineStatus.objects.filter(child_name=child)

        # Get all available VaccinePrograms
        all_programs = VaccinePrograms.objects.all()

        data = []

        for program in all_programs:
```

```
            # Check if the child has a status for this program
            status_for_program = statuses.filter(program=program)
            .first()

            # If the child has a status, include it in the response
            if status_for_program:
                program_data = {
                    # 'program': VaccineProgramSerializer(status
                    _for_program.program).data,
                    'status': {
                        'id': status_for_program.id,
                        'program': status_for_program.program.id,
                        'child_name': status_for_program.
                child_name.first_name,
                        'is_taken': status_for_program.is_taken
                    }
                }
            else:
                # If the child doesn't have a status, create a
                   placeholder entry with 'is_taken' set to False
                program_data = {
                    # 'program': VaccineProgramSerializer
                        (program).data,
                    'status': {
                        'id': None,
                        'program': program.id,
                        'child_name': child.first_name,
                        'is_taken': False
                    }
                }

            data.append(program_data)

        return Response(data)
```

dates_controller.dart

```
import 'dart:convert';

import 'package:doctor_baby/model/dates_model.dart';
import 'package:doctor_baby/services/dates_service.dart';
import 'package:get/get.dart';

class DatesController extends GetxController{

  var dates = <DateTime>[].obs;
```

```
  @override
  void onInit() {
    getDates();
    super.onInit();
  }

  void getDates() async{
    try {
      var data = await Httpdates.fetchDates();
      var jsondata = json.decode(data);
      VaccinationDates vaccinationDates = VaccinationDates.from
      Json(jsondata);
      dates.assignAll(vaccinationDates.vaccinationDates ?? []);
      print(dates);
    } catch (e) {
      print(e);
    }
  }
}
```

dates_model.dart

```
import 'dart:convert';

VaccinationDates vaccinationDatesFromJson(String str) =>
VaccinationDates.fromJson(json.decode(str));

String vaccinationDatesToJson(VaccinationDates data) =>
 json.encode(data.toJson());

class VaccinationDates {
    List<DateTime>? vaccinationDates;

    VaccinationDates({
        this.vaccinationDates,
    });

    factory VaccinationDates.fromJson(Map<String, dynamic> json) =>
VaccinationDates(
        vaccinationDates: json["vaccination_dates"] == null ? [] :
List<DateTime>.from(json["vaccination_dates"]!.map((x) =>
 DateTime.parse(x))),
    );

    Map<String, dynamic> toJson() => {
        "vaccination_dates": vaccinationDates == null ? [] :
```

```
List<dynamic>.from(vaccinationDates!.map((x) => "${x.year
.toString().padLeft(4, '0')}
-${x.month.toString().padLeft(2, '0')}-${x.day.toString().
padLeft(2, '0')}")),
    };
}
```

dates_service.dart

```
import 'package:doctor_baby/view/profile.dart';
import 'package:http/http.dart' as https;

class Httpdates{
  static Future<dynamic> fetchDates() async{
    var url = "http://10.0.2.2:8000/babyapp/childcreate/
${ProfilePage.userId}/vaccination-dates/";
    print("URL: $url");
    var response = await https.get(Uri.parse(url));
    if(response.statusCode==200){
      return response.body;
    }else{
      throw Exception();
    }
  }
}
```

calender.dart

```
import 'package:doctor_baby/controller/dates_controller.dart';
import 'package:doctor_baby/view/chat.dart';
import 'package:doctor_baby/view/mail/mail.dart';
import 'package:doctor_baby/view/program_status_view.dart';
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'package:get/get_core/src/get_main.dart';
import 'package:table_calendar/table_calendar.dart';
import 'package:intl/intl.dart';


class VaccineCalendar extends StatefulWidget {

  final DatesController datesController = Get.put(DatesController());

  @override
  _VaccineCalendarState createState() => _VaccineCalendarState();
  }
```

```
class _VaccineCalendarState extends State<VaccineCalendar> {
  CalendarFormat _calendarFormat = CalendarFormat.month;
  DateTime _focusedDay = DateTime.now();
  DateTime? _selectedDay;
  Map<DateTime, List<String>> _events = {};
  DateTime? _selectedListViewDate;


  @override
  void initState() {
    super.initState();

  sendMailToParent();

    datesController.dates.forEach((date) {
      _events[date] = ['Vaccine Date'];
    });

    DateTime birthDate = DateTime.now();
 }

  List<String> _getEventsForDay(DateTime date) {
    return _events[date] ?? [];
  }

  void _showEventDetails(DateTime selectedDay) {
    List<String> events = _events[selectedDay] ?? [];
    if (events.isNotEmpty) {
      showDialog(
        context: context,
        builder: (BuildContext context) {
          return AlertDialog(
            title: Text('Events on ${selectedDay.toString().
            split(' ')[0]}'),
            content: Column(
              crossAxisAlignment: CrossAxisAlignment.start,
              mainAxisSize: MainAxisSize.min,
              children: events.map((event) {
                return Text('- $event');
              }).toList(),
            ),
            actions: <Widget>[
              TextButton(
                onPressed: () {
                  Navigator.of(context).pop();
                },
```

```
              child: Text('Close'),
            ),
          ],
        );
      },
    );
  }
}

final DatesController datesController = Get.put(
DatesController());

@override
Widget build(BuildContext context) {
  return SafeArea(
    child: Scaffold(

      // floatingActionButton: FloatingActionButton(onPressed: (){
      //   Navigator.of(context).push(MaterialPageRoute(builder:
            (context)=>ChatBot()));
      // },child: Icon(Icons.message),),

      backgroundColor: Colors.white,
      appBar: AppBar(
        title: Text("Calendar", style: TextStyle(fontWeight:
        FontWeight.bold),),
      ),
      // drawer: Drawer(
      // child: Column(
      //   children: [
      //     TextButton(onPressed: () => Get.to(Programsview()),
            child: Text("Confirm"))
      //   ],
      // ),
    // )
      body: Expanded(
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            // Container(
            //   alignment: Alignment.center,
            //   child: Text("Doctor baby",
            //   style: TextStyle(
            //     fontSize: 25,
            //     color: Colors.black,
            //     fontWeight: FontWeight.bold
```

```
//   ),)),

TableCalendar(
  firstDay: DateTime.utc(2021, 1, 1),
  lastDay: DateTime.utc(2025, 12, 31),
  focusedDay: _focusedDay,
  calendarFormat: _calendarFormat,
  eventLoader: _getEventsForDay,
  headerStyle: HeaderStyle(
    formatButtonVisible: true,
  formatButtonDecoration: BoxDecoration
  (color: Colors.grey),
  leftChevronIcon: Icon(Icons.chevron_left,
  color: Colors.black),
  rightChevronIcon: Icon(Icons.chevron_right,
  color: Colors.black),
    titleTextStyle: TextStyle(color: Colors.black),
  ),
  calendarStyle: CalendarStyle(
    selectedDecoration: BoxDecoration(
      color: Colors.red,
      shape: BoxShape.circle,
    ),
    todayDecoration: BoxDecoration(
      color: Colors.green,
      shape: BoxShape.circle,
    ),
    defaultTextStyle: TextStyle(color: Colors.black),
  ),
  onFormatChanged: (format) {
    setState(() {
      _calendarFormat = format;
    });
  },
  selectedDayPredicate: (day) {
    return isSameDay(_selectedDay, day);
  },
  onDaySelected: (selectedDay, focusedDay) {
    setState(() {
      _selectedDay = selectedDay;
      _focusedDay = focusedDay;
    });
    _showEventDetails(selectedDay);
  },
  calendarBuilders: CalendarBuilders(
    markerBuilder: (context, date, events) {
```

```
        final markers = <Widget>[];

        if (_events[date] != null && _events[date]!.
        isNotEmpty) {
          markers.add(
            Positioned(
              bottom: 1,
              child: Container(
                height: 6,
                width: 6,
                decoration: BoxDecoration(
                  color: Colors.yellow,
                  shape: BoxShape.circle,
                ),
              ),
            ),
          );
        }

        return Row(children: markers);
      },
    ),
  ),
  // SizedBox(height: 10),
  Padding(
    padding: const EdgeInsets.all(8.0),
    child: Text(
      'Vaccine Dates:',
      style: TextStyle(letterSpacing: 1.2,fontSize:
    20, fontWeight: FontWeight.bold, color: Colors.black),
    ),
  ),
  Expanded(
    child: Container(
      // padding: EdgeInsets.all(8),
      decoration: BoxDecoration(
        color: Colors.grey[400],
        borderRadius: BorderRadius.only(
          topLeft: Radius.circular(25),
          topRight: Radius.circular(25),
        ),
      ),
      child:
      Obx(() =>
      ListView.builder(
        itemCount: datesController.dates.length,
```

```
                    itemBuilder: (context, index) {
                      DateTime date = datesController.dates[index];
                      return Card(
                        margin: EdgeInsets.symmetric(vertical: 5,
                        horizontal: 10),
                        color: Colors.grey[900],
                        child: ListTile(
                          title: Row(mainAxisAlignment:
                          MainAxisAlignment.center,
                            children: [
                              Text(
                                '${date.day}/${date.month}/
                                 ${date.year}',
                                style: TextStyle(
                                  color: Colors.white,
                                  letterSpacing: 1.2,
                                  fontSize: 17
                                ),
                              ),
                            ],
                          ),
                          onTap: () {
                            setState(() {
                              _selectedListViewDate = date;
                              _selectedDay = date;
                              _focusedDay = date;
                            });
                            _showEventDetails(date);
                          },
                        ),
                      );
                    },
                  ),
                ),
              ),)
            ],
          ),
        ),
      ),
    );
  }
}
```

hospital_controller.dart

```
import 'package:doctor_baby/services/hospital_service.dart';
import 'package:get/get.dart';
```

```
class HospitalsController extends GetxController{

  var hospitalsList = [].obs;

  @override
  void onInit() {
    getHospitals();
    super.onInit();
  }

  void getHospitals() async{
    try {
      var datas = await HospitalService.fetchHopitals();
      if(datas!=null){
        hospitalsList.value=datas;
      }
    } catch (e) {
      print(e);
    }
  }

}
```

hospital_model.dart

```
// To parse this JSON data, do
//
//     final hospitals = hospitalsFromJson(jsonString);

import 'dart:convert';

List<Hospitals> hospitalsFromJson(String str) =>
List<Hospitals>.from(json.decode(str).map((x) =>
Hospitals.fromJson(x)));

String hospitalsToJson(List<Hospitals> data) =>
json.encode(List<dynamic>.from(data.map((x) => x.toJson())));

class Hospitals {
    int? id;
    String? name;
    String? location;
    int? slotsAvailable;
    List<int>? programsAvailable;
    List<ProgramsDetail>? programsDetails;
```

```
    Hospitals({
        this.id,
        this.name,
        this.location,
        this.slotsAvailable,
        this.programsAvailable,
        this.programsDetails,
    });

    factory Hospitals.fromJson(Map<String, dynamic> json) =>
    Hospitals(
        id: json["id"],
        name: json["name"],
        location: json["location"],
        slotsAvailable: json["slots_available"],
        programsAvailable: json["programs_available"] == null ?
 [] : List<int>.from(json["programs_available"]!.map((x) => x)),
        programsDetails: json["programs_details"] == null ?
 [] : List<ProgramsDetail>.from(json["programs_details"]!.map((x) =>
ProgramsDetail.fromJson(x))),
    );

    Map<String, dynamic> toJson() => {
        "id": id,
        "name": name,
        "location": location,
        "slots_available": slotsAvailable,
        "programs_available": programsAvailable == null ?
          [] : List<dynamic>.from(programsAvailable!.map((x) => x)),
        "programs_details": programsDetails == null ?
          [] : List<dynamic>.from(programsDetails!.map((x) =>

 x.toJson())),
    };
}

class ProgramsDetail {
    List<Vaccine>? vaccines;

    ProgramsDetail({
        this.vaccines,
    });

    factory ProgramsDetail.fromJson(Map<String, dynamic> json)
    => ProgramsDetail(
        vaccines: json["vaccines"] == null ? [] : List<Vaccine>.
```

```
        from(json["vaccines"]!.map((x) => vaccineValues.map[x]!)),
    );

    Map<String, dynamic> toJson() => {
        "vaccines": vaccines == null ? [] : List<dynamic>
        .from(vaccines!.map((x) => vaccineValues.reverse[x])),
    };
}

enum Vaccine {
    DP_TW,
    HEP_B,
    HIB,
    IPV_OPV,
    MMR,
    OPV,
    TCV
}

final vaccineValues = EnumValues({
    "DPTw": Vaccine.DP_TW,
    "HepB": Vaccine.HEP_B,
    "Hib": Vaccine.HIB,
    "IPV/OPV": Vaccine.IPV_OPV,
    "MMR": Vaccine.MMR,
    "OPV": Vaccine.OPV,
    "TCV": Vaccine.TCV
});

class EnumValues<T> {
    Map<String, T> map;
    late Map<T, String> reverseMap;

    EnumValues(this.map);

    Map<T, String> get reverse {
        reverseMap = map.map((k, v) => MapEntry(v, k));
        return reverseMap;
    }
}
```

hospital_service.dart

```
import 'package:doctor_baby/model/hospitals_model.dart';
import 'package:http/http.dart' as https;

class HospitalService{
```

```
  static Future<dynamic> fetchHopitals()async{
    var response = await https.get(Uri.parse(
    "http://10.0.2.2:8000/babyapp/hospitals/"));
    if(response.statusCode == 200){
      var data = response.body;
      return hospitalsFromJson(data);
    }else{
      throw Exception();
    }
  }
}
```

hospital_view.dart

```
import 'package:doctor_baby/controller/hospital_controller.dart';
import 'package:doctor_baby/model/hospitals_model.dart';
import 'package:doctor_baby/model/program_status_model.dart';
import 'package:doctor_baby/view/booking_screen.dart';
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'package:google_fonts/google_fonts.dart';

void main(){
  runApp(GetMaterialApp(home: Hospitalsview(),));
}

class Hospitalsview extends StatelessWidget {

  final HospitalsController hospitalsController =
 Get.put(HospitalsController());

  Hospitalsview({super.key});

  @override
  Widget build(BuildContext context) {
    return Scaffold(
        backgroundColor: Colors.cyan[900],
      body: Column(
        children: [

          SizedBox(height: 60),

          Container(padding: EdgeInsets.symmetric(horizontal: 20),
            alignment: Alignment.centerLeft,
            height: 90,
            decoration: BoxDecoration(
              color:Colors.black,
```

```
      ),
    child: Text("Select a hospital to book your slot",
   style: GoogleFonts.robotoSlab(
    fontSize: 25,
    color: Colors.white,
  ),)
  ),

SizedBox(height: 15),

  Expanded(
    child: Obx(() =>
      ListView.builder(padding: EdgeInsets.symmetric(
      horizontal: 5),
        itemCount: hospitalsController.hospitalsList.length,
        itemBuilder: (context, index) {
        Hospitals hospitals = hospitalsController.
       hospitalsList[index];
        return Padding(
          padding: const EdgeInsets.symmetric(vertical: 5),
          child: InkWell(
            onTap: ()=> Get.to(Booking(hospitalname:
            hospitals.name!, hospitalId: hospitals.id!,)),
            child: Card(
              color:Colors.transparent,
              child: Container(
                alignment: Alignment.center,
                padding: EdgeInsets.symmetric(horizontal: 5),
                height: 250,
                child: Column(
                  mainAxisAlignment: MainAxisAlignment.center,
                  children: [
                    Container(
                      alignment: Alignment.center,
                      child: Text(hospitals.name!,
                      style: TextStyle(
                        color: Colors.white,
                        fontSize: 20),)),

    SizedBox(height: 10,),

     Column(
     mainAxisAlignment: MainAxisAlignment.spaceEvenly,
      children: [
      Container(
      decoration: BoxDecoration(
```

```
                color: Colors.cyan[700],
                borderRadius: BorderRadius.circular(10)),
                padding: EdgeInsets.all(8),
                // alignment: Alignment.centerLeft,
                 child: Row(
                  children: [
                  Text("Location: ",
                  style: TextStyle(color: Colors.white70),
                   ),
                  Text(hospitals.location!,
                   style: TextStyle(color: Colors.white),
                     ),
                      ],
                      )
                     ),

               SizedBox(height: 10,),

                Container(
                decoration: BoxDecoration(
                color: Colors.cyan[700],
                borderRadius: BorderRadius.circular(10)),
                padding: EdgeInsets.all(8),
                // alignment: Alignment.centerLeft,
                child: Row(
                children: [
                          Text("Slots Available: ",
                          style: TextStyle(color: Colors.white70),),
                           Container(
                           decoration: BoxDecoration(
                            color: Colors.black,
                             borderRadius: BorderRadius.circular(5)
                             ),
                            padding: EdgeInsets.symmetric(horizontal: 5,
                            vertical: 2),
                            child: Text("${hospitals.slotsAvailable!}",
                            style: TextStyle(color: Colors.white),),
                                        ),
                                      ],
                                    )
                                    ),

                                    SizedBox(height: 10,),

Container(
decoration: BoxDecoration(
```

```
    color: Colors.cyan[700],
    borderRadius: BorderRadius.circular(10),
  ),
  padding: EdgeInsets.all(8),
  child: Row(
    children: [
      Text("Vaccines Available: ", style: TextStyle(
      color: Colors.white70),),
      Row(
        children: hospitals.programsAvailable!.map((program) {
          return Padding(
            padding: EdgeInsets.only(right: 5),
            child: Container(
              padding: EdgeInsets.symmetric(horizontal: 5,
              vertical: 2),
              decoration: BoxDecoration(
                color: Colors.black,
                borderRadius: BorderRadius.circular(5),
              ),
              child: Text(
                "$program",
                style: TextStyle(color:
                Colors.white),
              ),
            ),
          );
        }).toList(),
      ),
    ],
  ),
)
```

```
                          ],
                        ),
                        SizedBox(height: 10,),

                        Container(
                          decoration: BoxDecoration(
                          color: Colors.cyan[900],
                          borderRadius: BorderRadius.circular(20)),
                          padding: EdgeInsets.symmetric(
                          horizontal: 50, vertical: 3),
                          child: Text("Select", style:
                          TextStyle(fontSize: 16,
                          color: Colors.white),),
```

```
                          )
                        ],
                      ),
                    ),
                  ),
                ),
              );
            },),
          ),
        ),
      ],
    ),
  );
}
```

program_status_controller.dart

```
import 'package:doctor_baby/services/hospital_service.dart';
import 'package:doctor_baby/services/program_status_service.dart';
import 'package:get/get.dart';

class ProgramController extends GetxController{

  var programlist = [].obs;

  @override
  void onInit() {
    getprograms();
    super.onInit();
  }

  void getprograms() async{
    try {
      var datas = await ProgramService.fetchPrograms();
      if(datas!=null){
        programlist.value=datas;
      }
    } catch (e) {
      print(e);
    }
  }

}
```

program_status_model.dart

```
// To parse this JSON data, do
```

```
//
//     final programStatus = programStatusFromJson(jsonString);

import 'dart:convert';

List<ProgramStatus> programStatusFromJson(String str) =>
List<ProgramStatus>.from(json.decode(str).map((x) =>

ProgramStatus.fromJson(x)));

String programStatusToJson(List<ProgramStatus> data) =>
 json.encode(List<dynamic>.from(data.map((x) => x.toJson())));

class ProgramStatus {
    Program? program;
    List<StatusElement>? statuses;

    ProgramStatus({
        this.program,
        this.statuses,
    });

    factory ProgramStatus.fromJson(Map<String, dynamic> json) =>
 ProgramStatus(
        program: json["program"] == null ? null : Program.fromJson
        (json["program"]),
        statuses: json["statuses"] == null ? [] :
        List<StatusElement>.
        from(json["statuses"]!.map((x) =>
       StatusElement.fromJson(x))),
    );

    Map<String, dynamic> toJson() => {
        "program": program?.toJson(),
        "statuses": statuses == null ? [] : List<dynamic>.
         from(statuses!.map((x) => x.toJson())),
    };
}

class Program {
    int? id;
    List<Vaccine>? vaccines;

    Program({
        this.id,
        this.vaccines,
```

```
    });

    factory Program.fromJson(Map<String, dynamic> json) => Program(
        id: json["id"],
        vaccines: json["vaccines"] == null ? [] : List<Vaccine>.
        from(json["vaccines"]!.map((x) => Vaccine.fromJson(x))),
    );

    Map<String, dynamic> toJson() => {
        "id": id,
        "vaccines": vaccines == null ? [] :
         List<dynamic>.from(vaccines!.map((x) => x.toJson())),
    };
}

class Vaccine {
    int? id;
    String? vaccine;

    Vaccine({
        this.id,
        this.vaccine,
    });

    factory Vaccine.fromJson(Map<String, dynamic> json) => Vaccine(
        id: json["id"],
        vaccine: json["vaccine"],
    );

    Map<String, dynamic> toJson() => {
        "id": id,
        "vaccine": vaccine,
    };
}

class StatusElement {
    StatusStatus? status;

    StatusElement({
        this.status,
    });

    factory StatusElement.fromJson(Map<String, dynamic> json) =>
    StatusElement(
     status: json["status"] == null ? null : StatusStatus.
    fromJson(json["status"]),
```

```
    );

    Map<String, dynamic> toJson() => {
        "status": status?.toJson(),
    };
}

class StatusStatus {
    int? id;
    int? program;
    String? childName;
    bool? isTaken;

    StatusStatus({
        this.id,
        this.program,
        this.childName,
        this.isTaken,
    });

    factory StatusStatus.fromJson(Map<String, dynamic> json) =>
   StatusStatus(
        id: json["id"],
        program: json["program"],
        childName: json["child_name"],
        isTaken: json["is_taken"],
    );

    Map<String, dynamic> toJson() => {
        "id": id,
        "program": program,
        "child_name": childName,
        "is_taken": isTaken,
    };
}
```

program_status_service.dart

```
import 'package:doctor_baby/model/program_status_model.dart';
import 'package:http/http.dart' as https;

class ProgramService{
  static Future<dynamic> fetchPrograms()async{
    var response = await https.get(Uri.parse(
   "http://10.0.2.2:8000/babyapp/vaccinestatus/"));
    if(response.statusCode == 200){
      var data = response.body;
```

```
      return programStatusFromJson(data);
    }else{
      throw Exception();
    }
  }
}
```

program_status_view.dart

```
import 'dart:convert';

import 'package:doctor_baby/controller/program
%20_status_controller.dart';
import 'package:doctor_baby/model/program_status_model.dart';
import 'package:doctor_baby/view/calendar.dart';
import 'package:doctor_baby/view/home.dart';
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:http/http.dart' as http;

void main(){
  runApp(GetMaterialApp(home: Programsview(),));
}

class VaccineSelectionController extends GetxController {
  RxString selectedVaccineId = "".obs;

  void setSelectedVaccineId(String id) {
    selectedVaccineId.value = id;
  }
}

final VaccineSelectionController vaccineSelectionController =
Get.put(VaccineSelectionController());


class Programsview extends StatelessWidget {

  final ProgramController programController = Get.put(
ProgramController());
  final TextEditingController dropController=
TextEditingController();
  TextEditingController nameController = TextEditingController();

  static String enteredName = "";
```

```
Programsview({super.key});

@override
Widget build(BuildContext context) {
  return Scaffold(
    // appBar: AppBar(
      // backgroundColor: Colors.black,
    body: Container(
      height: double.infinity,
        decoration: BoxDecoration(
          image: DecorationImage(
            image: NetworkImage(
              "https://media.istockphoto.com/id/1395084227/photo/
             covid-19-vaccine.jpg?s=2048x2048&w=is&k=20&c=
             kiPagHTCx5T6itBE_ek4AtZA8QeiXrUIaSt2VuNom1E=",
            ),
            fit: BoxFit.fill,
          ),
        ),
      // decoration: BoxDecoration(image: DecorationImage
      (image: AssetImage("assets/baby.png"))),
      child: SingleChildScrollView(
        child: Column(
          children: [

          SizedBox(height: 100),

            Padding(
              padding: const EdgeInsets.all(10.0),
              child: Card(
                child: Container(alignment: Alignment.center,
                  height: 550,
                  color: Colors.transparent,
                  child: Column(
                    children: [

                Container(
                  padding: EdgeInsets.symmetric(horizontal: 20),
                  alignment: Alignment.center,
                  height: 90,
                  decoration: BoxDecoration(
                    color:Colors.transparent,
                    borderRadius: BorderRadius.only(
                    bottomLeft: Radius.circular(20),
                    bottomRight: Radius.circular(20)
```

```
                    )),
                  child: Text("Select vaccine to confirm",
                 style: GoogleFonts.robotoSlab(
                  fontSize: 25,
                  color: Colors.black,
                  fontWeight: FontWeight.bold
                ),)
                ),

                        Expanded(
                          child: Obx(() =>
                            ListView.builder(padding:
                            EdgeInsets.symmetric(horizontal: 5),
                              itemCount: programController.
                              programlist.length,
                              itemBuilder: (context, index) {
                              ProgramStatus program =
                              programController.programlist[index];
                              return Padding(
                                padding: const EdgeInsets.
                                symmetric(vertical: 5, horizontal: 5),
                                child: InkWell(
                                  onTap: () {
   // vaccineSelectionController.setSelectedVaccineId(program.
program!.id.toString());
  String selectedVaccineId = program.program!.id.toString();
  showDialog(
           context: context,
           builder: (BuildContext context) {
           return Container(
           child: Padding(
            padding: const EdgeInsets.symmetric(vertical: 50),
            child: AlertDialog(
            title: Text("Vaccine Confirmation"),
            content: SingleChildScrollView(
            child: Column(
            children: [
             Container(
             alignment: Alignment.center,
             width: double.infinity,
             padding: EdgeInsets.all(8),
             color: Colors.black,
            child: Text("Have you taken the vaccination
            ${program.program!.id.toString()}?", style:
            TextStyle(color: Colors.white),)),
```

```
SizedBox(height: 10,),

TextField(
controller: nameController,
decoration: InputDecoration(labelText: 'Enter Babyname'),
                                                                                ),
],
),
),
actions: [
TextButton(
onPressed: () async {

enteredName = nameController.text.trim();

String username = nameController.text.trim();

String yes = "True";

await post(selectedVaccineId, username, yes);
// Get.to(VaccineCalendar());
},
child: Text("Yes"),
),
TextButton(
onPressed: () {
Navigator.of(context).pop();
},
child: Text("No"),
),
],
),
),
);
},
);
} ,
child: Padding(
padding: const EdgeInsets.symmetric(horizontal: 20),
child: Container(
decoration: BoxDecoration(
color:Colors.green,
borderRadius: BorderRadius.circular(10)),
child: Container(
alignment: Alignment.center,
padding: EdgeInsets.symmetric(horizontal: 5,vertical: 10),
```

```
height: 50,
child: Row(
mainAxisAlignment: MainAxisAlignment.spaceAround,
children: [
Container(
alignment: Alignment.center,
\child: Text("Vaccine ${program.program!.id.toString()}",
style: TextStyle(
color: Colors.white,
fontSize: 20),)),
                                          ],
                                        ),
                                      ),
                                    ),
                                  ),
                                ),
                              ),
                            );
                          },),
                        ),
                      ),
                    ],
                  ),
                ),
              ),
            ],
          ),
        ),
      ),
    );
  }
}

Future<void> post(String vaccineId, String username,
 String yes) async {
    final url = Uri.parse('http://10.0.2.2:8000/babyapp/
    setvaccinestatus/$vaccineId/');
    final response = await http.post(
      url,
      headers: {'Content-Type': 'application/json'},
      body:
      jsonEncode(
        {
        'is_taken': yes,
        'child_name': username
        }
```

```
        ),
    );

    if (response.statusCode == 200) {
      print('Confirmation successful');
      Get.snackbar("Vaccination", "Confirm");
      Get.to(Home());
    } else {
      print('Error: ${response.statusCode}');
    }
  }
```

summary_controller.dart

```
import 'package:doctor_baby/services/summary_service.dart';
import 'package:doctor_baby/services/vaccine_service.dart';
import 'package:get/get.dart';

class SummaryController extends GetxController{

  var summaryList = [].obs;

  @override
  void onInit() {
    getSummary();
    super.onInit();
  }

  void getSummary() async{
    try {
      var datas = await VaccineSummary.fetchSummary();
      if(datas!=null){
        summaryList.value=datas;
      }
    } catch (e) {
      print(e);
    }
  }

}
```

summary_model.dart

```
// To parse this JSON data, do
//
//     final summary = summaryFromJson(jsonString);
```

```
import 'dart:convert';

List<Summary> summaryFromJson(String str) =>
List<Summary>.from(json.decode(str).map((x) => Summary.fromJson(x)));

String summaryToJson(List<Summary> data) =>
json.encode(List<dynamic>.from(data.map((x) => x.toJson())));

class Summary {
    Status? status;

    Summary({
        this.status,
    });

    factory Summary.fromJson(Map<String, dynamic> json) => Summary(
        status: json["status"] == null ? null :
        Status.fromJson(json["status"]),
    );

    Map<String, dynamic> toJson() => {
        "status": status?.toJson(),
    };
}

class Status {
    int? id;
    int? program;
    String? childName;
    bool? isTaken;

    Status({
        this.id,
        this.program,
        this.childName,
        this.isTaken,
    });

    factory Status.fromJson(Map<String, dynamic> json) => Status(
        id: json["id"],
        program: json["program"],
        childName: json["child_name"],
        isTaken: json["is_taken"],
    );

    Map<String, dynamic> toJson() => {
```

```
        "id": id,
        "program": program,
        "child_name": childName,
        "is_taken": isTaken,
    };
}
```

summary_service.dart

```
import 'package:doctor_baby/model/vaccines_model.dart';
import 'package:doctor_baby/view/program_status_view.dart';
import 'package:http/http.dart' as https;

class VaccineSummary{
  static Future<dynamic> fetchSummary()async{
    var response = await https.get(Uri.parse(
    "http://10.0.2.2:8000/babyapp/child_vaccine_status/nazrin/"));
    if(response.statusCode == 200){
      var data = response.body;
      return vaccinesFromJson(data);
    }else{
      throw Exception();
    }
  }
}
```

booking_summary.dart

```
import 'package:doctor_baby/model/summary_model.dart';
import 'package:doctor_baby/view/profile.dart';
import 'package:doctor_baby/view/program_status_view.dart';
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'package:http/http.dart' as http;


class SummaryScreen extends StatefulWidget {
  @override
  _SummaryScreenState createState() => _SummaryScreenState();
}

class _SummaryScreenState extends State<SummaryScreen> {
  final _summaryController = Get.put(SummaryController());

  @override
  void initState() {
    super.initState();
```

```
    _summaryController.fetchSummaryData();
  }

  @override
  Widget build(BuildContext context) {
    return SafeArea(
      child: Scaffold(
        backgroundColor: Colors.grey,

        body: Padding(
          padding: const EdgeInsets.symmetric(vertical: 25),
          child: Card(color: Colors.black54,
            child: Container(
              padding: EdgeInsets.symmetric(vertical: 10),
              // height: 600,
              child: Column(mainAxisAlignment: MainAxisAlignment.center,
                children: [

                  Text("Vaccine Summary", style: TextStyle(color:
                   Colors.grey[200], fontSize: 25, fontWeight:
                   FontWeight.bold),),
                  SizedBox(height: 20),

                  Expanded(
                    child: Obx(
                      () {
                        if (_summaryController.isLoading.value) {
                          return Center(child:
                          CircularProgressIndicator());
                        } else {
                          return ListView.builder(
                            itemCount: _summaryController.
                            summaryList.length,
                            itemBuilder: (context, index) {
                              var summary = _summaryController.
                              summaryList[index];

                              String isTakenDisplay = summary.
                              status?.isTaken == true ?
                              'Completed' : 'Pending';
                              var isTakenColor =
                            summary.status?.isTaken == true ?
                            Colors.green : Colors.red;

return
Padding(
```

```
padding: const EdgeInsets.symmetric(vertical: 5, horizontal: 10),
child: Container(
decoration: BoxDecoration(
color: Colors.black,
borderRadius: BorderRadius.circular(10)),
padding: EdgeInsets.symmetric(vertical: 12,horizontal: 10),
width: double.infinity,
child: Row(
children: [
Container(
width: 150,
padding: EdgeInsets.all(10),
child: Text('Vaccine ${summary.status?.program}'
,style: TextStyle(fontSize: 20, color: Colors.white)),
),
Expanded(
child: Container(
decoration: BoxDecoration(
color: isTakenColor,
borderRadius: BorderRadius.circular(20)),
alignment: Alignment.center,
padding: EdgeInsets.all(10),
child: Text('${isTakenDisplay}',style: TextStyle(fontSize: 20),)),
                                        )
                                      ],
                                    ),
                                  ),
                                );
ListTile(
title: Text('Vaccine ${summary.status?.program}'),
// subtitle: Text('Is Taken: ${summary.status?.isTaken}'),
subtitle: Text('Taken: ${isTakenDisplay}'),
                                );
                              },
                            );
                          }
                        },
                      ),
                    ),
                  ],
                ),
              ),
            ),
          ),
        ),
      ),
    );
```

```
  }
}

class SummaryController extends GetxController {
  RxBool isLoading = true.obs;
  RxList<Summary> summaryList = <Summary>[].obs;

  Future<void> fetchSummaryData() async {
    try {

      String? firstName = await Util.getFirstName();

      final response = await http.get(Uri.parse(
     'http://10.0.2.2:8000/babyapp/child_vaccine_status/$firstName/'));
      if (response.statusCode == 200) {
        final List<Summary> summaries = summaryFromJson(response.body);
        summaryList.assignAll(summaries);
      } else {
        throw Exception('Failed to load data');
      }
    } catch (e) {
      print('Error: $e');
    } finally {
      isLoading.value = false;
    }
  }
}
```

vaccine_comtroller.dart

```
import 'package:doctor_baby/services/vaccine_service.dart';
import 'package:get/get.dart';

class VaccineController extends GetxController{

  var vaccineList = [].obs;

  @override
  void onInit() {
    getVaccines();
    super.onInit();
  }

  void getVaccines() async{
    try {
      var datas = await VaccineService.fetchVaccine();
      if(datas!=null){
```

```
        vaccineList.value=datas;
      }
    } catch (e) {
      print(e);
    }
  }

}
```

vaccine_model.dart

```
// To parse this JSON data, do
//
//     final vaccines = vaccinesFromJson(jsonString);

import 'dart:convert';

List<Vaccines> vaccinesFromJson(String str) =>
 List<Vaccines>.from(json.decode(str).map((x) => Vaccines.fromJson(x)));

String vaccinesToJson(List<Vaccines> data) =>
 json.encode(List<dynamic>.from(data.map((x) => x.toJson())));

class Vaccines {
    int? id;
    List<Vaccine>? vaccines;

    Vaccines({
        this.id,
        this.vaccines,
    });

    factory Vaccines.fromJson(Map<String, dynamic> json) => Vaccines(
        id: json["id"],
        vaccines: json["vaccines"] == null ? [] : List<Vaccine>.
        from(json["vaccines"]!.map((x) => Vaccine.fromJson(x))),
    );

    Map<String, dynamic> toJson() => {
        "id": id,
        "vaccines": vaccines == null ? [] : List<dynamic>.
         from(vaccines!.map((x) => x.toJson())),
    };
}

class Vaccine {
    int? id;
```

```
    String? vaccine;

    Vaccine({
        this.id,
        this.vaccine,
    });

    factory Vaccine.fromJson(Map<String, dynamic> json) => Vaccine(
        id: json["id"],
        vaccine: json["vaccine"],
    );

    Map<String, dynamic> toJson() => {
        "id": id,
        "vaccine": vaccine,
    };
}
```

vaccine_service.dart

```
import 'package:doctor_baby/model/vaccines_model.dart';
import 'package:http/http.dart' as https;

class VaccineService{
  static Future<dynamic> fetchVaccine()async{
    var response = await https.get(Uri.parse(
    "http://10.0.2.2:8000/babyapp/vaccine_programs/"));
    if(response.statusCode == 200){
      var data = response.body;
      return vaccinesFromJson(data);
    }else{
      throw Exception();
    }
  }
}
```

chat.dart

```
import 'dart:convert';
import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;
import 'package:get/get.dart';

class ChatController extends GetxController {
  RxList<Map<String, String>> chat = <Map<String, String>>[].obs;

  TextEditingController textController = TextEditingController();
```

```
  RxString text = ''.obs;
  RxString prediction = ''.obs;

  Future<void> postData(String text) async {
    var url = "http://10.0.2.2:8000/babyapp/chatbot/";
    chat.add({
      "responeby": "user",
      "text": text,
    });
    try {
      final jsonResponse = await http.post(
        Uri.parse(url),
        body: {'user_input': text.trim()},
      );

      if (jsonResponse.statusCode == 200) {
        Map<String, dynamic> parsedResponse = jsonDecode
        (jsonResponse.body);
        String responseValue = await parsedResponse['response'];
        chat.add({
          "responeby": "bot",
          "text": responseValue,
        });
        print(responseValue);
      } else {
        print("Error: ${jsonResponse.statusCode}");
      }
    } catch (error) {
      print("Error: $error");
    }
  }
}

class ChatBot extends StatelessWidget {
  final ChatController chatController = Get.put(ChatController());

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        centerTitle: true,
        backgroundColor: Colors.cyan[900],
        title: Text(
          "Baby Helper",
          style: TextStyle(fontWeight: FontWeight.bold, color:
          Colors.white),
```

```
        ),
      ),
      body: SingleChildScrollView(
        child: Column(
          children: [
Obx(() => ListView.builder(
shrinkWrap: true,
physics: NeverScrollableScrollPhysics(),
itemCount: chatController.chat.length,
itemBuilder: (context, index) {
String chatText = chatController.chat[index]["text"]!;
return Align(
alignment: chatController.chat[index]["responeby"] == "bot"
? Alignment.centerLeft
: Alignment.centerRight,
child: Container(
child: Padding(
padding: const EdgeInsets.all(8.0),
child: Container(
decoration: BoxDecoration(
color: chatController.chat[index]["responeby"] == "bot"
? Colors.blue
: Colors.grey,
borderRadius: chatController.chat[index]["responeby"] ==
"bot"
? BorderRadius.only(
bottomLeft: Radius.circular(10),
bottomRight: Radius.circular(10),
topRight: Radius.circular(10))
: BorderRadius.only(
bottomLeft: Radius.circular(10),
bottomRight: Radius.circular(10),
topLeft: Radius.circular(10)),
                            ),
                            child: Padding(
                              padding: const EdgeInsets.only(
                                  top: 8, bottom: 8, left: 15,
                                  right: 15),
                              child: Text(
                                chatText,
                                style: TextStyle(
                                    fontSize: 18,
                                    color: Colors.white,
                                    fontWeight: FontWeight.bold),
                              ),
                            ),
```

```
                    ),
                  ),
                ),
              );
            },
          )),
      Padding(
        padding: EdgeInsets.only(
          bottom: MediaQuery.of(context).viewInsets.bottom,
        ),
      ),
    ],
  ),
),
bottomNavigationBar: Padding(
  padding: EdgeInsets.only(
    bottom: MediaQuery.of(context).viewInsets.bottom,
  ),
  child: Container(
    height: 60,
    decoration: BoxDecoration(
      color: Colors.grey[200],
      borderRadius: BorderRadius.circular(20),
    ),
    child: Padding(
      padding: const EdgeInsets.symmetric(horizontal: 20),
      child: Row(
        children: [
          Expanded(
            child: TextFormField(
              controller: chatController.textController,
              onChanged: (value) => chatController.text.
              value = value,
              decoration: InputDecoration(
                border: InputBorder.none,
                hintText: "Enter here ..",
              ),
            ),
          ),
          SizedBox(width: 10),
          InkWell(
            onTap: () {
              chatController.postData(chatController.
              text.value);
              chatController.textController.text = "";
            },
```

```
                  child: CircleAvatar(
                    radius: 18,
                    backgroundColor: Colors.blue,
                    child: Center(child: Icon(Icons.send)),
                  ),
                ),
              ],
            ),
          ),
        ),
      ),
    );
  }
}

void main() {
  runApp(GetMaterialApp(home: ChatBot()));
}
```

home.dart

```
import 'package:doctor_baby/view/booking_summary.dart';
import 'package:doctor_baby/view/calendar.dart';
import 'package:doctor_baby/view/chat.dart';
import 'package:doctor_baby/view/components/carousals.dart';
import 'package:doctor_baby/view/hospitals_view.dart';
import 'package:doctor_baby/view/program_status_view.dart';
import 'package:flutter/material.dart';
import 'package:get/get.dart';

void main(){
  runApp(GetMaterialApp(home: Home(),debugShowCheckedModeBanner:
  false,));
}

class Home extends StatefulWidget {
  const Home({super.key});

  @override
  State<Home> createState() => _HomeState();
}

class _HomeState extends State<Home> {

  var name = [
    "Calendar",
    "Hospitals",
```

```
    "Confirm",
    "Summary",
    "Chat bot",
    "Profile",
  ];

  var icon = [
    Icons.calendar_month,
    Icons.local_hospital,
    Icons.confirmation_num_rounded,
    Icons.list_alt,
    Icons.chat,
    Icons.person,
  ];

  var color = [
    Colors.blue[100],
    Colors.red[300],
    Colors.green[100],
    Colors.pink[100],
    Colors.purple[200],
    Colors.cyan[100],
  ];

  var pages = [
    VaccineCalendar(),
    Hospitalsview(),
    Programsview(),
    SummaryScreen(),
    ChatBot(),
    ChatBot(),
  ];


  @override
  Widget build(BuildContext context) {
    return SafeArea(
      child: Scaffold(
        body: SingleChildScrollView(
          child: Column(
          children: [

            Container(
              decoration: BoxDecoration(
                color: Colors.cyan[900],
                borderRadius: BorderRadius.only(bottomRight:
```

```
               Radius.circular(50))
              ),
              child: Column(
                children: [
                  SizedBox(height: 10,),

                  Container(
                    alignment: Alignment.center,
                    width: double.infinity,
                    child: Text("DoctorBaby", style: TextStyle(

color: Colors.white, fontSize: 30),),
                    ),
            Container(
              width: double.infinity,
              padding: EdgeInsets.symmetric(horizontal: 15,
              vertical: 10),
              height: 100,
              child: Padding(
                padding: const EdgeInsets.all(8.0),
                child: Column(
                  mainAxisAlignment: MainAxisAlignment.center,
                  crossAxisAlignment: CrossAxisAlignment.start,
                  children: [
                    Text("Hi User, ",style: TextStyle(color:
                    Colors.grey[300], fontSize: 22),),
                    Text("How are you doing today?",style:
                    TextStyle(color: Colors.grey[200], fontSize: 23))
                  ],
                ),
              ),
            ),
                ],
              ),
            ),


            Carousal(),

            Padding(
              padding: const EdgeInsets.all(4.0),
              child: Container(
                decoration: BoxDecoration(
                color: Colors.grey[100],
                  borderRadius: BorderRadius.circular(10)),
                height:300,
```

```
  child: Column(
    children: [
      Padding(
        padding: const EdgeInsets.symmetric(
       horizontal: 30, vertical: 10),
        child: Container(
          alignment: Alignment.centerLeft,
          height: 25,
          width: double.infinity,
          child: Text("Category", style: TextStyle(
          fontSize: 18, fontWeight: FontWeight.bold),),
        ),
      ),
      SizedBox(height: 10,),
      Expanded(
child: GridView.builder(
  physics: NeverScrollableScrollPhysics(),
  itemCount: 6,
  gridDelegate:
  SliverGridDelegateWithFixedCrossAxisCount(crossAxisCount: 3),
  itemBuilder: (context, index){
    return Padding(
      padding: const EdgeInsets.all(1.0),
      child: Column(
        children: [
          InkWell(onTap: () => Get.to(pages[index]),
            child: Container(
              decoration: BoxDecoration(
              color: Colors.blue[100],
                borderRadius: BorderRadius.circular(20)
              ),
              padding: EdgeInsets.all(8),
              child: Icon(icon[index], size: 50),
            ),
          ),
          SizedBox(height: 5,),
          Text(name[index], style: TextStyle(
          color: Colors.blue, fontWeight:
          FontWeight.bold),)
        ],
      ),
    );
  }),
      )
    ],
  ),
```

```
  ),
),

Padding(
  padding: const EdgeInsets.symmetric(
  horizontal: 10, vertical: 5),
  child: Container(
    decoration: BoxDecoration(
    color: Colors.grey[300],
      borderRadius: BorderRadius.circular(10)
    ),
    width: double.infinity,
    child: Expanded(
      child: Padding(
        padding: const EdgeInsets.all(2),
        child: Row(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Card(
              // color: Colors.grey[900],
              child: Container(
                width: 140,
                padding: EdgeInsets.all(10),
                child: Column(
                  mainAxisAlignment: MainAxisAlignment.center,
                  children: [

                  Text("Quick and",style:
                   TextStyle(color: Colors.black)),
                   // SizedBox(height: 3,),
                   Text("easy",style:
                    TextStyle(color: Colors.black)),
                   // SizedBox(height: 3,),
                   Text("appointments",style:
                   TextStyle(color: Colors.black)),
                ],),
              ),
            ),
            SizedBox(width: 30),
            Card(
              // color: Colors.grey[900],
              child: Container(
                width: 140,
                padding: EdgeInsets.all(10),
                child: Column(
                  crossAxisAlignment: CrossAxisAlignment.center,
```

```
                              mainAxisAlignment: MainAxisAlignment.center,
                              children: [

                              Text("Safe and", style:
                              TextStyle(color: Colors.black),),
                              // SizedBox(height: 3,),
                              Text("effective",style:
                              TextStyle(color: Colors.black)),
                              // SizedBox(height: 3,),
                              Text("vaccines",style:
                               TextStyle(color: Colors.black)),
                            ],),
                          ),
                        ),
                      ],
                    ),
                  )),
              ),
            )

          ],
                ),
        )
      ),
    );

  }
}
```

profile.dart

```
import 'package:doctor_baby/view/bottom_nav.dart';
import 'package:doctor_baby/view/calendar.dart';
import 'package:doctor_baby/view/home.dart';
import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;
import 'dart:convert';
import 'dart:io';
import 'package:image_picker/image_picker.dart';
import 'package:shared_preferences/shared_preferences.dart';

class ProfilePage extends StatefulWidget {

  static int? userId;

  @override
  _ProfilePageState createState() => _ProfilePageState();
```

```
}

class _ProfilePageState extends State<ProfilePage> {
  final _firstnameController = TextEditingController();
  final _lastnameController = TextEditingController();
  final _dobController = TextEditingController();
  final _parentNameController = TextEditingController();
  File? _profileImage;
  String _selectedGender = 'Male';



  Future<void> _saveProfile() async {
    if (_validateForm()) {
      final url = 'http://10.0.2.2:8000/babyapp/childcreate/';

      final requestData = {
        'first_name': _firstnameController.text,
        'last_name': _lastnameController.text,
        'date_of_birth': _dobController.text,
        'sex': _selectedGender,
        'parent_username': _parentNameController.text,
      };

      try {
        final response = await http.post(
          Uri.parse(url),
          body: json.encode(requestData),
          headers: {
            'Content-Type': 'application/json',
          },
        );

        if (response.statusCode == 201) {
          print(response.body);
          _showSnackBar('Profile created successfully!');
          final Map<String, dynamic> responseData =
          json.decode(response.body);

          // userId = responseData['id'];
          ProfilePage.userId = responseData["id"];


          await saveFirstName(_firstnameController.text);
```

```
      print('Retrieved ID: ${ProfilePage.userId}');
        Navigator.of(context).push(MaterialPageRoute(builder:
         (context) => Home()));
      } else {
        _showSnackBar('Failed to create profile. Please try again.');
      }
    } catch (e) {
      print('Exception: $e');
      _showSnackBar('Failed to create profile. Please try again.');
    }
  }
}


Future<void> saveFirstName(String firstName) async {
  SharedPreferences prefs = await SharedPreferences.getInstance();
  prefs.setString('firstName', firstName);
}



bool _validateForm() {
  if (_firstnameController.text.isEmpty ||
      _lastnameController.text.isEmpty ||
      _dobController.text.isEmpty ||
      _parentNameController.text.isEmpty ||
      _selectedGender.isEmpty) {
    _showSnackBar('Please fill in all fields.');
    return false;
  }
  return true;
}

void _showSnackBar(String message) {
  ScaffoldMessenger.of(context).showSnackBar(
    SnackBar(content: Text(message)),
  );
}

 Future<void> _selectDate(BuildContext context) async {
  final DateTime? picked = await showDatePicker(
    context: context,
    initialDate: DateTime.now(),
    firstDate: DateTime(1900),
    lastDate: DateTime.now(),
  );
```

```
    if (picked != null && picked != DateTime.now()) {
      setState(() {
        _dobController.text = picked.toLocal().toString().

split(' ')[0];
      });
    }
  }

  Future<void> _pickImage() async {
    final picker = ImagePicker();
    final pickedFile = await picker.pickImage(source:
   ImageSource.gallery);

    if (pickedFile != null) {
      setState(() {
        _profileImage = File(pickedFile.path);
      });
    }
  }
  @override
  Widget build(BuildContext context) {
    return SafeArea(
      child: Scaffold(
        body: Container(
          decoration: BoxDecoration(image: DecorationImage
         (image: NetworkImage("https://images.unsplash.com/
          photo-1582486225644-aeacf6aa0b1bq=80&w=
         1887&auto=format&fit=crop&ixlib=rb-4.0.3&ixid=

M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA
%3D%3D"), fit: BoxFit.cover)),
          height: double.infinity,
          child: Padding(
            padding: const EdgeInsets.all(16.0),
            child: SingleChildScrollView(
              child: Card(
                color: Colors.black45,
                child: Padding(
                  padding: const EdgeInsets.all(12.0),
                  child: Form(
                    child: Column(
                      crossAxisAlignment: CrossAxisAlignment.start,
                      children: [
```

```
  SizedBox(height: 20),
Text("Create a profile for your baby",
style: TextStyle(color: Colors.grey[200],
fontSize: 25),),

  SizedBox(height: 70),

  TextFormField(style: TextStyle(color:
  Colors.grey[100]),
    controller: _firstnameController,
    decoration: InputDecoration(labelText:
    'First Name',
      labelStyle: TextStyle(fontWeight:
     FontWeight.bold, color: Colors.grey[300]),
      border: OutlineInputBorder
    (borderRadius: BorderRadius
          .circular(20) ),),

    validator: (value) {
      if (value == null || value.isEmpty) {
        return 'Please enter your first name.';
      }
      return null;
    },
  ),
  SizedBox(height: 12),
  TextFormField(style: TextStyle(color:
   Colors.grey[100]),
    controller: _lastnameController,
    decoration: InputDecoration(labelText:
   'Last Name',
      labelStyle: TextStyle(fontWeight:
      FontWeight.bold, color: Colors.grey[300]),
      border: OutlineInputBorder(
      borderRadius: BorderRadius
          .circular(20) ),),

    validator: (value) {
      if (value == null || value.isEmpty) {
        return 'Please enter your last name.';
      }
      return null;
    },
  ),
  SizedBox(height: 12),
  TextFormField(style: TextStyle(
```

```
                        color: Colors.grey[100]),
                          controller: _dobController,
                          decoration: InputDecoration(
                            labelText: 'Date of Birth',  labelStyle:
                          TextStyle(fontWeight: FontWeight.bold,
                           color: Colors.grey[300]),
                            border: OutlineInputBorder(
                            borderRadius: BorderRadius
                                .circular(20) ),),
                          onTap: () => _selectDate(context),
                          readOnly: true,
                        ),
                        SizedBox(height: 12),
                        TextFormField(style: TextStyle(color:
                          Colors.grey[100]),
                          controller: _parentNameController,
                          decoration: InputDecoration(labelText:
                         'Parent Name',  labelStyle: TextStyle(fontWeight:
                         FontWeight.bold,color: Colors.grey[300]),
                            border: OutlineInputBorder(
                          borderRadius: BorderRadius
                                .circular(20) ),),
                          validator: (value) {
                            if (value == null || value.isEmpty) {
                              return 'Please enter the parent name.';
                            }
                            return null;
                          },
                        ),
                        SizedBox(height: 24),

                        Text("Gender", style: TextStyle(fontWeight:

FontWeight.bold, color: Colors.grey[300])),
                        Row(
                          children: [
                            Radio(
                              value: 'Male',
                              groupValue: _selectedGender,
                              onChanged: (value) {
                                setState(() {
                                  _selectedGender = value.toString();
                                });
                              },
                            ),
                            Text('Male', style: TextStyle(
```

```
                            color: Colors.grey[200]),),
                            Radio(
                              value: 'Female',
                              groupValue: _selectedGender,
                              onChanged: (value) {
                                setState(() {
                                  _selectedGender = value.toString();
                                });
                              },
                            ),
                            Text('Female', style: TextStyle(

color: Colors.grey[200]),),
                          ],
                        ),


                        SizedBox(height: 25),

                        Center(
                          child: GestureDetector(
                            onTap: _saveProfile,
                            child: Container(
                              padding: EdgeInsets.all(10),
                              child: Text(
                                "Create profile",
                                style: TextStyle(fontSize: 16,
                                color: Colors.black),
                              ),
                              decoration: BoxDecoration(
                                borderRadius: BorderRadius.circular(15),
                                color: Colors.blue,
                              ),
                            ),
                          ),
                        ),
                        SizedBox(height: 20),
                      ],
                    ),
                  ),
                ),
              ),
            ),
          ),
        ),
      ),
```

```
    );
  }
}


  class Util {
  static Future<String?> getFirstName() async {
    SharedPreferences prefs = await SharedPreferences.getInstance();
    return prefs.getString('firstName');
  }
}


void main() {
  runApp(MaterialApp(
    home: ProfilePage(),
  ));
}
```

mail.dart

```
import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;

void sendMailToParent() async {
  var url = 'http://10.0.2.2:8000/babyapp/send_mail_date/';

  try {
    var response = await http.get(
      Uri.parse(url),
      headers: <String, String>{
        'Content-Type': 'application/json; charset=UTF-8',
      },
    );

    if (response.statusCode == 200) {
      print('Email sending initiated.');
    } else {
      print('Failed to send email. Status code: ${response.statusCode}');
    }
  } catch (e) {
    print('Error sending email: $e');
  }
}
```

CAR TRAFFIC SIGN RECOGNIZER

PROJECT REPORT

Submitted By

REVATHI RAJESH

Reg. No. CCAVBCA010

for the award of the Degree of

Bachelor (BCA)

of Computer Application
(University of Calicut)

under the guidance of

Ms.Varsha Ganesh

Assistant Professor

BACHELOR OF COMPUTER APPLICATION
DEPARTMENT OF COMPUTER SCIENCE
CHRIST COLLEGE(AUTONOMOUS)
IRINJALAKUDA, KERALA
INDIA

2021-2024

DEPARTMENT OF COMPUTER SCIENCE

CHRIST COLLEGE (AUTONOMOUS)

IRINJALAKUDA

CERTIFICATE

This is to certify that the project report entitled ***"Car Traffic Sign Recognizer"*** *is a bonfied record of the project work done by* ***Revathi Rajesh*** *in partial fulfillment of the requirement for the sixth semester of* ***Bachelor of Computer Application*** *in Department of Computer Science of* **CHRIST COLLEGE (AUTONOMOUS) IRINJALAKUDA**.

Ms.Varsha Ganesh
Assistant Professor
Internal Guide

Ms.Sini Thomas
Head of the Department
Computer Science

EXTERNAL EXAMINER

INTERNAL EXAMINER

DECLARATION

I hereby declare that this project work **"CAR TRAFFIC SIGN RECOGNIZER"** submitted to Christ College (Autonomous) Irinjalakuda, affiliated to Calicut University in partial fulfillment of the requirement for the award of the Bachelor of Computer Application, is a record of original work done by me, under the guidance of Ms.VARSHA GANESH, Department of Computer Science.

Place: Irinjalakuda REVATHI RAJESH

ACKNOWLEDGEMENT

First and foremost, I would like to thank Lord almighty for his providence and for being the guiding light throughout the project. I wish to express my sincere gratitude to our beloved Department for giving us all the facilities for our project. I take this opportunity to express my gratitude to the class teacher Ms.SOUMYA P.S and the Head of the Department Ms.SINI THOMAS who has supported us throughout the course of this project. I am thankful for their aspiring guidance and valuable advices during the project work. I express my sincere thanks to our project guide Ms.VARSHA GANESH for supporting and guiding us throughout the project. I would take this opportunity to specially thank all other faculty members for their constant and continuous motivation. Finally, I would like to thank my family and friends for giving valuable advices and moral support throughout our project.

ABSTRACT

CAR TRAFFIC SIGN RECOGNIZER is a website designed to helps the driver about recognition of road signs to avoid road accidents. It represents an important feature of advanced driver assistance system, contributing to the safety of the drivers, autonomous vehicles as well and to increase driving comfort. The website has three kind of login facilities :- admin login, user login and driver login. All these features make this website more adaptable and user-friendly.

Contents

Chapter 1

1 Introduction

In today's world road conditions drastically improved as compared with past decades. Obviously, vehicle's speed increased. So, on driver's point of view there might be chances of neglecting mandatory road signs while driving. This project explores the system to helps the driver about recognition of road signs to avoid road accidents. An automatic means of detecting and recognizing traffic signs can make a significant contribution to this goal by providing a fast method of detecting, classifying and logging signs. This method helps to develop the inventory accurately and consistently.

The project is mainly focused on automatic recognition of warning signs placed in local roads captured by image clips. The system classifies the traffic signs present in the image into different categories. With this model, we can read and understand traffic signs which are very important task for autonomous vehicles.

1.1 Overview

The objective of the Car Traffic Sign Recognizer website is to design a simple and adaptable website which helps users to increase transportation efficiency, road safety and to reduce the environmental impact with the use of advanced communication technologies. The road signs are placed on either roadside or above the roads. These signs provide mandatory information regarding to guiding, warning and regulating the behaviors to drivers in order to make driving safer and easier. A system is developed to assist the drivers, based on detection and recognition of signs which corrects the most unsafe driving behaviors.

Chapter 2

2 System Analysis

2.1 Purpose

The purpose of the traffic sign recognition project is to develop a sophisticated system that can automatically detect, interpret, and respond to traffic signs in real-time. This project aims to empower vehicles with the ability to perceive and understand the various types of traffic signs encountered on roadways. Ultimately, the goal of this project is to create a robust and reliable solution for enhancing the awareness and decision-making capabilities of drivers and autonomous vehicles alike.

2.1.1 Existing System

Existing systems of traffic sign recognition utilize a combination of cameras, sensors, and sophisticated image processing algorithms to detect and interpret traffic signs in real-time. By capturing images of traffic signs and analyzing them these systems can recognize various types of signs, including speed limits, stop signs, and lane restrictions. Limitations of existing system : The system may still encounter challenges in accurately detecting and interpreting traffic signs, leading to false detections or missed signs, potentially impacting driver safety.

2.1.2 Proposed System

The proposed system of traffic sign recognition aims to address the current limitations by leveraging advancements in artificial intelligence, particularly deep learning algorithms. By training deep neural networks on large datasets of annotated traffic sign images, the system would learn to accurately detect and interpret various types of signs, including speed limits, stop signs, and lane restrictions, across diverse environmental conditions.

2.2 Problem definition

To know what the problem is and what the needs are before developing it. Here we design Car Traffic Sign Recognizer to detect the traffic signs accurately.

2.3 Feasibility Study

After the problem is clearly understood and solutions proposed, the next step is to conduct the feasibility study. Feasibility study is defined as evaluation or analysis of the potential impact of the proposed project or program. The objective is to determine whether the proposed system is feasible. There are three aspects of feasibility study to which the proposed system is subjected as discussed below.

2.3.1 Technical Feasibility

Technical feasibility assess whether the current technical resources are sufficient for the new system. We can upgrade the level of technology for supporting our website. We check whether the proposed system can be implemented in the present system without supporting the existing hardware.

2.3.2 Economical Feasibility

Economic feasibility determines whether the time and money are available to develop the system. There is no additional hardware used to develop the site, it is inexpensive to build.

2.3.3 Operational Feasibility

Operational feasibility determines if the human resources are available to operate the system once it has been installed. No extra training is needed to use this system. Anyone who has knowledge in internet in english language can easily use the system. The resources that are required to implement or install are already available with the organization.

Chapter 3

3 Software Requirement Specification

3.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the CAR TRAFFIC SIGN RECOGNIZER. It illustrate the purpose and complete description for the development of the system. It explain system constraints, interface and interactions with other external applications.

3.2 Scope

The scope of car traffic sign recognizer extends to improving road safety, enhancing driver assistance technologies, optimizing traffic flow, and advancing the capabilities of autonomous vehicles and smart transportation systems.

3.3 Overall Description

This section give an overview of our website, CAR TRAFFIC SIGN RECOGNIZER. This project aims to develop an intelligent system capable of automatically detecting, recognizing, and interpreting traffic signs from images captured by cameras mounted on vehicles. Upon detection, the system may provide visual or auditory alerts to drivers, contributing to improved road safety and compliance with traffic laws. It plays a vital role in enhancing driver assistance systems, navigation, and autonomous vehicle technologies, ultimately leading to safer and more efficient transportation systems.

3.3.1 Product Perspective

The car traffic sign recognition project aims to develop a robust and user-friendly system that enhances driver safety and convenience. The project can develop a comprehensive solution that enhances road safety, improves driver assistance technologies, and advances the capabilities of autonomous vehicles and smart transportation systems.

3.3.2 Product Functionality

The system provides a comprehensive solution that enhances road safety, improves driver assistance technologies, and advances the capabilities of autonomous vehicles and smart transportation systems by detecting the traffic signs and notifying the driver.

3.3.3 Users and Characteristics

There are three types of users that interact with the site - admin, user and driver. Each of these have different tasks which is performed. Users are able to detect the traffic signs, view drivers and book drivers. Drivers can view the bookings and approve it. Admin is able view all the users and drivers and has the authority to delete the users or drivers.

3.4 Specific Requirements

3.4.1 Hardware Requirements

- System: LAPTOP-TE3E25EG
- Processor: i3 8th Gen or above
- Speed: 2.80 GHz
- RAM capacity: 8 GB
- Hard Dsk drive: 128 GB
- Keyboard: Standard
- Mouse: Standard
- Monitor: 18.5” LED Monitor

3.4.2 Software Requirements

- Operating System: Windows
- Languages used: Python, Django
- Database : SQLite3
- Technologies used: HTML,Javascript,CSS

3.5 Functional Requirements

It contains three main modules.

- 1.Admin
- 2.User
- 3.Driver

Admin

An Admin account is used for editing or managing the website dynamically by Admin panel. The admin can view the users and drivers and has the authority to delete any user or driver.

User

The user can detect the traffic signs. There is a feature providing for booking a driver and viewing the bookings.

Driver

The driver can view the bookings and user details. The driver can approve the booking.

3.6 Non Functional Requirements

Non-functional requirements define the overall qualities or attributes of the resulting system. Non-functional system place restrictions on the product being developed, the developed process, and specify external constraints that the product must meet. Examples of non-functional requirements include safety, security, usability, reliability and performance requirements. Project management issues (costs, time and schedule) are often considered as non-functional requirements. The principle non-functional constraints which are relevant to critical systems :

- performance
- security
- safety
- usability

Performance

Performance requirements concern the speed of operation of a system. Types of performance requirements :

- Response requirements (how quickly the system reacts to a user input).
- Throughput requirements (how much the system can accomplish within a specified amount of time).
- Availability requirements (is the system available for service when requested by end users). The speed of operation of this system is adequate for the requirements.

Reliability

- Reliability is the ability of a system to perform its required function under stated conditions for a specified period of time.
- Constraints on runtime behavior of the system. This system is reliable because its functionalities can be done on the required conditions.

Safety

Safety requirements are not required which exclude unsafe situation from the possible solution of the system.

Usability

Usability is the ease with which a user can learn to operate, prepare inputs for, and interpret outputs of system or components.
Usability requirements include :

- information error messages.
- well-formed user interfaces.

3.7 Interface Requirements

3.7.1 Hardware interfaces

The system must run over the internet, all the hardware shall require to connect internet will be hardware interface for the system. As for example modem, WAN-LAN.

3.7.2 Software interfaces

Software interface required for the working of the project is the appropriate operating system.

3.7.3 Communication interfaces

The user of the site communicate using the network connectivity and the data set is accessed

3.8 Security Requirements

- User accesses only their account.
- Validation of input is handled.
- This application containing the computer systems is physically secured against arms or surreptitious entry by intruders.
- Users must be authorized carefully to reduce changes of any such user giving access to an intruder in exchange for a bribe or other favour.

3.9 Platform Used

Windows 11 is the latest iteration of Microsoft's operating system, introducing a refreshed user interface, enhanced productivity features, and improved performance. With a sleek design aesthetic centered around simplicity and productivity, Windows 11 offers users a more intuitive and personalized computing experience.Windows 11 represents Microsoft's vision for the future of computing, blending familiarity with innovation to empower users to do more. Windows 11 brings performance improvements, with optimizations for better energy efficiency and faster wake times. With support for gaming features like DirectStorage and Auto HDR, along with enhanced security measures, Windows 11 aims to cater to a wide range of users, from casual consumers to power users and professionals, ushering in a new era of computing excellence.

3.10 Technologies Used

Python

Python is a high-level programming language renowned for its simplicity, versatility, and readability. With its clean and concise syntax, Python is accessible to beginners while remaining powerful enough for complex applications in various domains, including web development, data analysis, machine learning, artificial intelligence, and automation.Its extensive standard library and robust ecosystem of third-party packages make it a favorite among developers for rapid prototyping, building scalable applications, and solving a wide range of computational problems. Python's interpreted nature and cross-platform compatibility further contribute to its popularity, enabling developers to write code once and run it on multiple platforms without modification. Overall, Python's ease of use, flexibility, and community support make it a go-to choice for developers worldwide.

SQLite3

SQLite3 is a lightweight, self-contained, serverless relational database engine that is widely used for embedded systems, mobile applications, and small-scale database-driven websites. It is part of the SQLite project, which aims to provide a simple, fast, and reliable database solution with minimal setup and administration. SQLite3 is unique in that it operates as a single disk file and requires no configuration or administration, making it extremely easy to deploy and use. Despite its small footprint, SQLite3 supports many

standard SQL features, including transactions, triggers, and views, making it suitable for a wide range of applications. Overall, SQLite3 is an excellent choice for projects requiring a lightweight and efficient database solution without the overhead of a full-fledged database management system.

Chapter 4

4 Design Document

4.1 Purpose

The purpose of a car traffic sign recognizer project is to enhance road safety, improve driver assistance systems, and facilitate the development of autonomous vehicles. By developing an intelligent system capable of automatically detecting, recognizing, and interpreting traffic signs from images captured by onboard cameras, this project aims to provide valuable information to drivers in real-time. The system can alert drivers about speed limits, stop signs, yield signs, and other relevant traffic regulations, helping them make informed decisions and comply with road rules more effectively. Additionally, integrating traffic sign recognition technology into vehicles contributes to the advancement of autonomous driving technologies by enabling vehicles to perceive and understand the surrounding environment, ultimately leading to safer and more efficient transportation systems. Purpose of this document is to give the detailed description of the architecture and system design for the software. The system design document shows the software will be structured to satisfy the requirement identified in the software requirement specification. It is a translation of requirements into a description of the software structure, software component, interfaces and data necessary for the implementation phase. In a complete software design document, each requirements must be traceable to one or more design entities.

4.2 Scope

Car Traffic Sign Recognizer is a website which helps in detecting the traffic signs and alert the driver to ensure safe and comfortable driving.

4.3 Overview

The purpose of this document is to help the reader to visualize the solution to the project presented.This document verifies how the design meet the requirement stimulated in the SRS document through design viewpoints. The design viewpoints will cover all design elements presented before. This document will provide a direct approach to the development

of this project hence reducing feature creep and ponitedly determine the quality of the design.

4.4 Data Design

Database are the storehouses of data used in the software systems.The data is stored in tables inside the database. Several tables are created for the manipulation of the data for the system.Two essential settings for a database are

- Primary Key- The field that is unique for all the record occurrence.
- Foreign Key- The field used to set relation between tables.

Normalization

- Normalization is the process of organizing the data in the database.
- Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate the undesirable characteristics like Insertion, Update and Deletion Anomalies.
- Normalization divides the larger table into the smaller table and links them using relationship.
- The normal form is used to reduce redundancy from the database table.

Tables

textapp_login

Name	DataType	Constraints	Description
id	int	Primarykey	ID of registered people
username	varchar(100)	Notnull	Name of registered people
password	varchar(100)	Notnull	Password of registered people
usertype	varchar(100)	Notnull	Type of registered people
status	varchar(100)	Notnull	Current status of the booking
driverid_id	bigint	Foreignkey	ID of driver

textapp_user

Name	DataType	Constraints	Description
id	int	Primarykey	ID of users
uname	varchar(100)	Notnull	Name of users
uaddress	varchar(100)	Notnul	Address of users
ucontact	varchar(100)	Notnull	Contact number of the users
uemail	varchar(100)	Notnull	Email id of users
upassword	varchar(100)	Notnull	Password of users

textapp_driver

Name	DataType	Constraints	Description
id	int	Primarykey	ID of users
dname	varchar(100)	Notnull	Name of drivers
daddress	varchar(100)	Notnul	Address of drivers
dcontact	varchar(100)	Notnull	Contact number of the drivers
demail	varchar(100)	Notnull	Email id of drivers
dpassword	varchar(100)	Notnull	Password of drivers
status	varchar(100)	Notnull	Current status of the booking

textapp_booking

Name	DataType	Constraints	Description
id	int	Primarykey	ID of users
date	varchar(100)	Notnull	Date of booking
status	varchar(100)	Notnull	Current status of the booking
did_id	bigint	Foreignkey	ID of driver
uid_id	bigint	Foreignkey	ID of user

Chapter 5

5 Development of the System

This website can be decomposed into a number of submodules.The submodules of car traffic sign recognizer are administrator, user and driver. Each submodule have specific objectives, to avoid unwanted coupling between modules and to increase cohesion, modules are again decomposed into submodules.

Chapter 6

6 System Testing

Testing is the penultimate step of software development.An elaborate testing of the data is prepared and the system is using the test data, while doing testing, errors are noted and correction is made. The users are trained to operate the developed system. Both hardware and software securities are made to run the developed system successfully. System testing is aimed at ensuring the system works accurately before the live operation commences.

6.1 Test Plan

6.1.1 Scope

This test plan will cover the following testing activities as identified in the testing strategy.

- White Box Testing

White box testing is a testing technique, that examines the program structure and derives test data from the program logic/code.In white box testing, the UI is bypassed.Inputs and outputs of ratings are tested directly at the code level and the results are compared against specifications.It reveals errors in "hidden" code.

- Black Box Testing

Black-box testing is a method of software testing that examines the functionality of an application based on the specifications.Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end-user would.

- Unit Testing

The module includes four main components to undergo unit testing.It checks the syntax error, logic error and validity of the program correctness.This test will be performed by the developers.

- Integration Testing

After all components pass the unit testing, they get to be tested whether they work correctly when they are running concurrently and communicating to each other.Integration tests exercise an entire subsystem and ensure that a set of components play nicely together.

- Internal Data Testing

We will test the validity of the data before it enters the database to avoid any problems that may face us in the database. We will test the encryption of the personal information of all the users along with the user names and passwords to ensure maximum security of the users privacy.

6.1.2 Software risk issues

In this section, the plan is to test the risk involved in critical issues such as: Difficult to run on visual studio due to large loading time. Some inherent software risks such as complexity exist; also these issues need to be identified. Proper network connection Working of Mysql database

6.1.3 Features to be tested

- Test whether correct user name and password allows you to login.
- Test whether invalid user name and password prevents you from login.
- Test whether there is any connection problem in the Server.
- Test whether the user and admin details are entered correctly.
- Test whether invalid data entry allows saving data successfully.
- Test whether all pages are loaded correctly.
- Test whether watermark is embedded and extracted properly.
- Test whether the provided security works correctly.

6.2 Test consolidation

6.2.1 Test item

The items or features to be tested in the test cases are included in the document. Each and every user input is tested. The present condition of the system is tested. It is checked to make sure that environment is ready for the application to work.

6.2.2 Input specifications

These are the inputs required to execute the test case. All required inputs including data elements and values are given below.

Data	Values
Email	A valid email ID
Password	Combination of characters and numbers
Phone number	10 Digit number

Chapter 7

7 System Implementation and Maintenance

7.1 Implementation

System implementation is the conversion of new system into an operating one which involves creating compatible files, training clients and installing hardware. User training is crucial for minimizing resistance to change and giving chance to prove its worth. Training aids user friendly manuals and healthy screens provide the user with a good start. Software maintenance follows conversion to the extent that changes are necessary to maintain satisfactory operations relative to changes in the user's environment. Maintenance often includes minor enhancements or corrections to the problem that surface late in the systems operations.In the implementation phase, the team builds the components either from scratch or by composition. Given the architecture document meant from the design phase and the requirement document from the analysis phase, the team should build exactly what has been requested, though there is still room for innovation and flexibility. For example, a component may be narrowly designed for this particular system, or the component may be made more general to satisfy a reusability

- Careful planning
- Investigation of system and constraints
- Design the methods to achieve changeover.
- Training the staff in the changed phase.
- Evaluation of change over method.
- The method of implementation and time scale to be adopted are found out initially.

7.2 Maintenance

This phase occurs as a result of deploying the whole system at the end users organization. They will perform the beta testing at the end users and inform to the developers about any needed modification to the application. The customer records all the problems that are encountered during the beta testing and reports these to the developer at regular intervals.

7.2.1 Corrective Maintenance

Even with the best quality assurance activities, it is likely that the customer will uncover defects in the software.Corrective maintenance changes the software to correct the defects. Corrective Maintenance activity may consist of repair, restoration or replacement of equipment. This activity will be result of a regular inspection, which identifies the failure in time for corrective maintenance to be planned and scheduled, then performed during a routine maintenance shutdown.

7.2.2 Adaptive Maintenance

Over time, the original environment(CPU, operating system, business rules, external product characteristics) for which the software was developed is likely to change. Adaptive maintenance results in modification to the software to accommodate changes to its external environment.

7.2.3 Enhanced Maintenance

As software is used, the customer/user will recognize additional functions that will provide the benefit. Perfect maintenance extends the software beyond its original functional requirements.

7.2.4 Preventive Maintenance

Computer software deteriorates due to change, and because of this preventive maintenance often called software re-engineering, must be conducted to enable the software to serve the needs of its end users. Preventive maintenance makes changes to computer programs so that they can be more easily corrected, adapted and enhanced.

Chapter 8

8 Conclusion and Future Scope

8.1 Conclusion

There are three types of users that interact with the system Admin, User and Driver. Each of these three types has different uses of the system so each of them has their own panel. Admin can manage all the features of website dynamically by login on admin panel. The users can detect the traffic signs and can also book a driver and view the bookings. The driver can view the bookings and user details and can approve them.

8.2 Future Scope

- Increased accuracy using better camera device
- Adaptability to dynamic environments

Appendix

A Data Flow Diagram

Data flow is the one of the best way of documenting the entire functionality of the system. For the system ,which will have data flows in and have some processing inside and then some data flow out from the system can be documented or represented effectively by means of data flow out from the system can be documented or represented effectively by means of data flow diagrams. The data flow diagram are a diagrammatic representation of the system,which has input,process and output. Once any system is represented using a data flow diagram we can identify the following things easily:

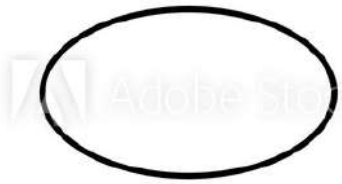
- Various entities interacting with the system are identified
- Flow of data from one entity to another is identified
- The various processes involved in between the interaction of two or more entities in the system are clearly pointed out
- The various data stores which hold the data in between the process,are clearly identified

Some Data Flow Diagram charting forms:

A.1 External source or receiver

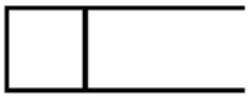
A source or sink is a person or part of organization ,which enters or receives information from the system,but is considered to be outlining the contest of data flow model.

A.2 Transform process



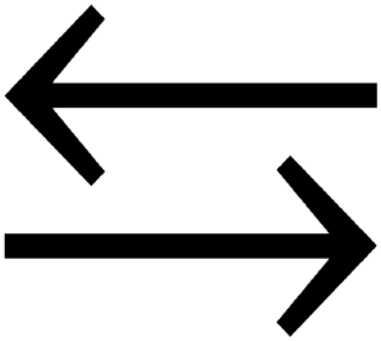
A process represents transformation where incoming data flowsare changed into outgoing data flow

A.3 Data Store



A data store is repository of data that is to be stored for use by one or more process may be as simple as buffer or queue or sophisticated as relational database. They should have clear names.If a process merely uses the contest of store and does not alter it,the arrowhead goes only form the store to the process. If a process alters the details in the store then double-headed arrow is used.

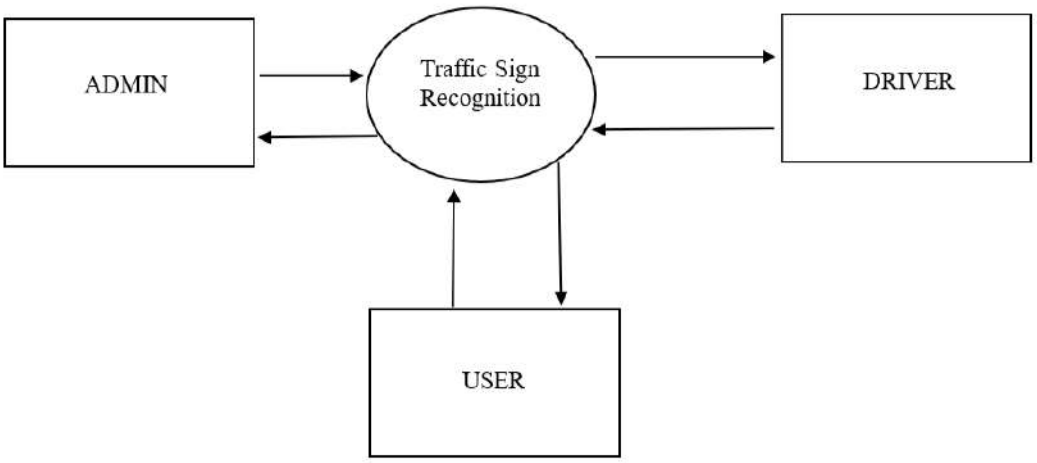
A.4 Data flow



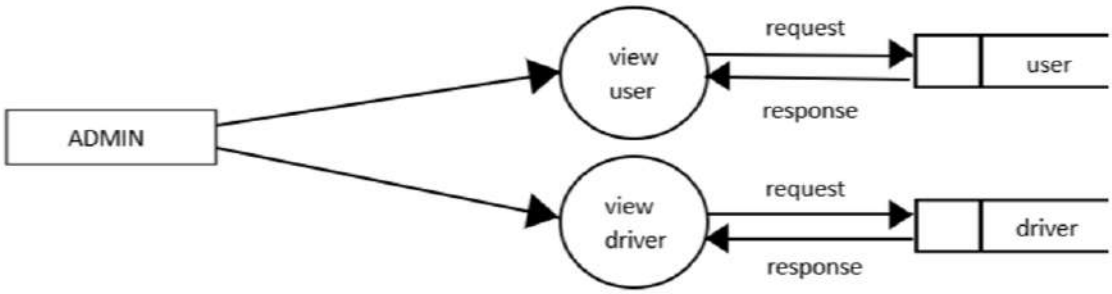
A data flow is a route, which enable packets of data to travel from one point to another.Data may flow,with arrowhead pointing in the direction of the flow

B Data Flow Diagrams

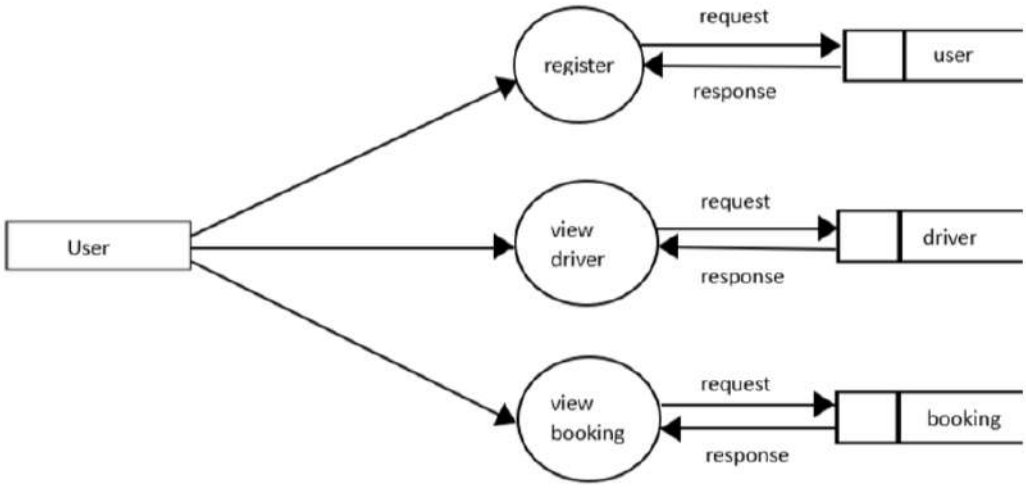
B.1 Level 0



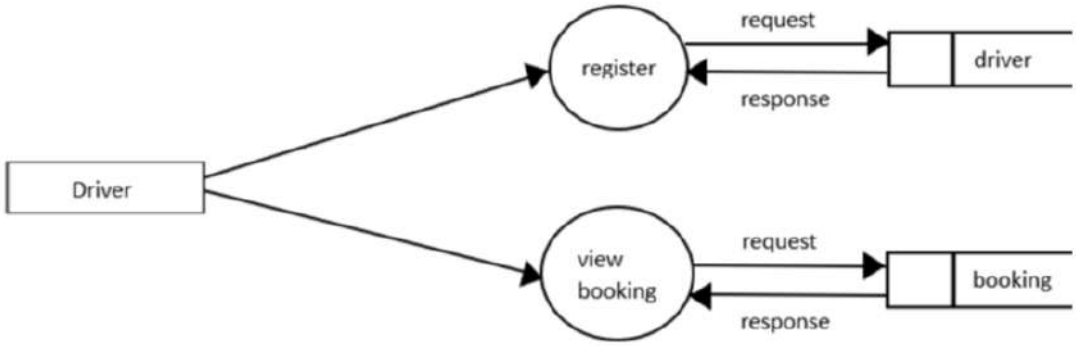
B.2 Level 1.1 - Admin



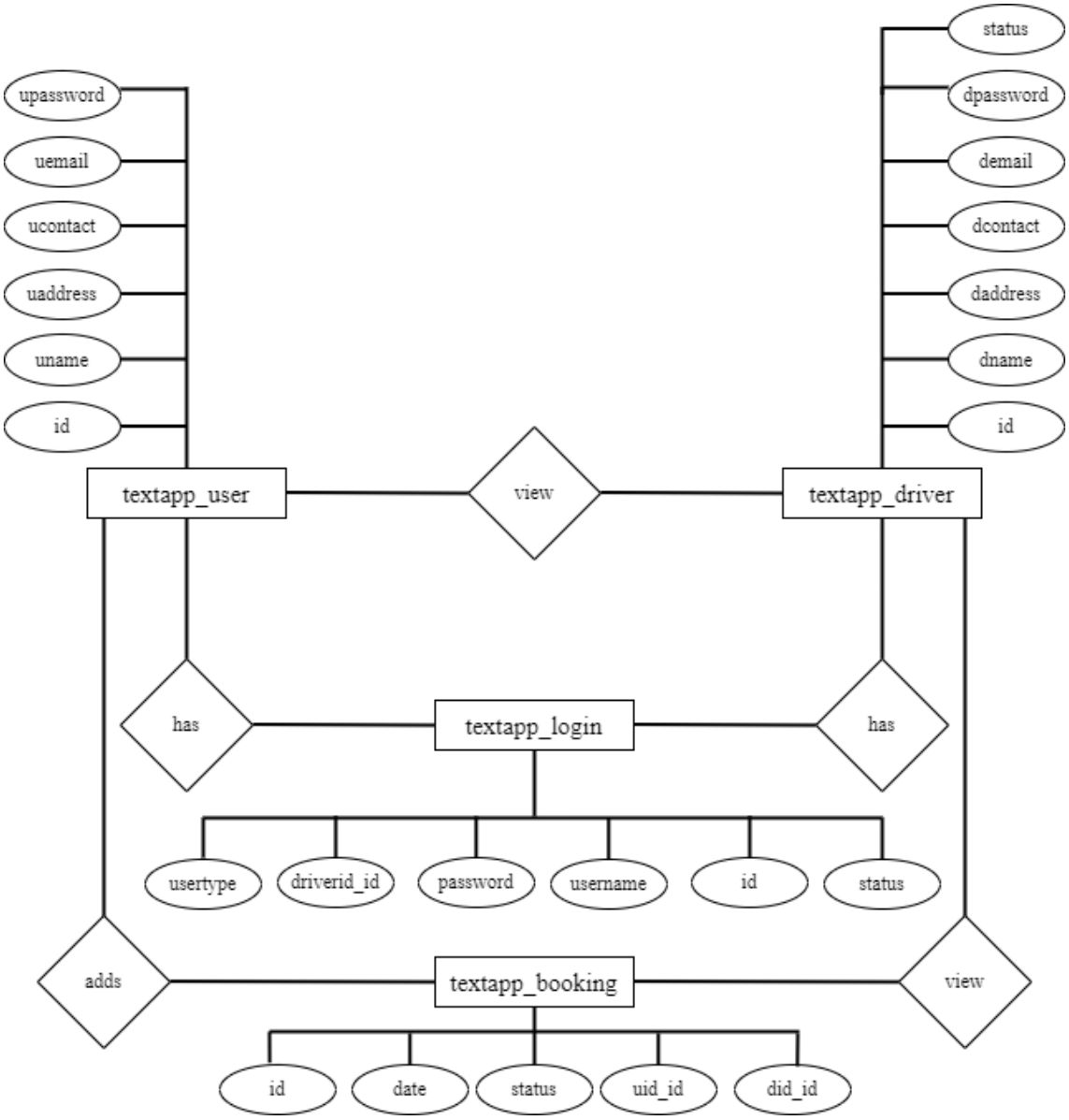
B.3 Level 1.2 - User



B.4 Level 1.3 - Driver

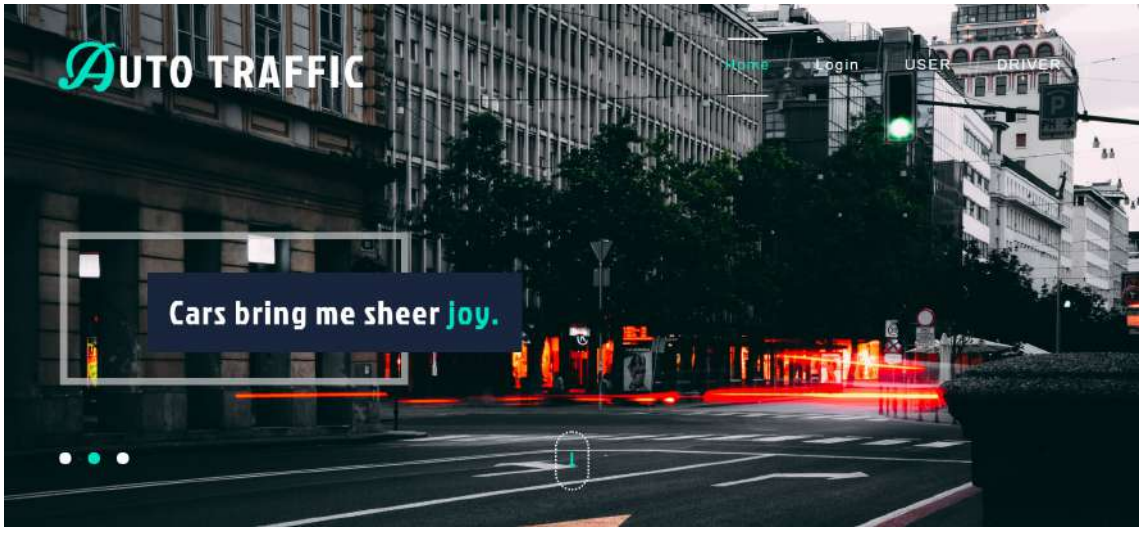


B.5 ER Diagram

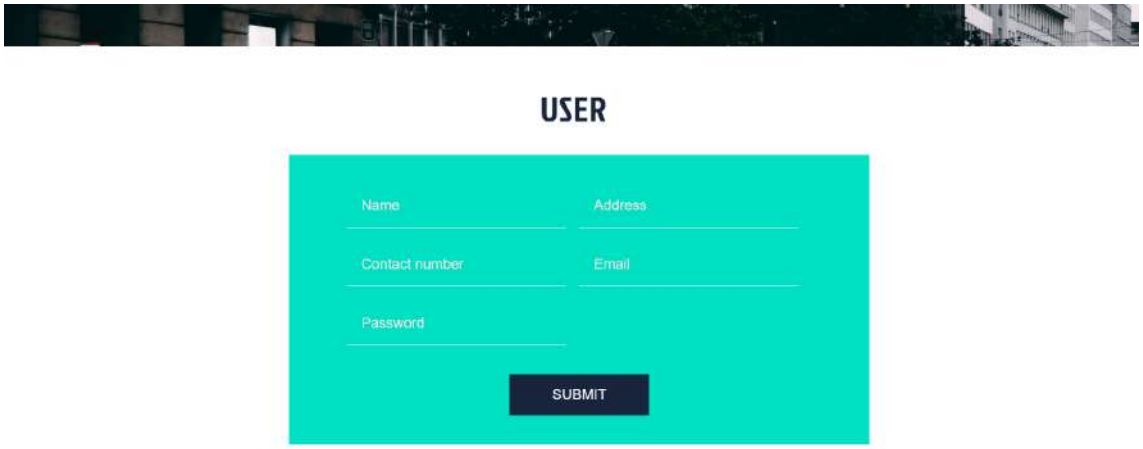


C USER INTERFACES

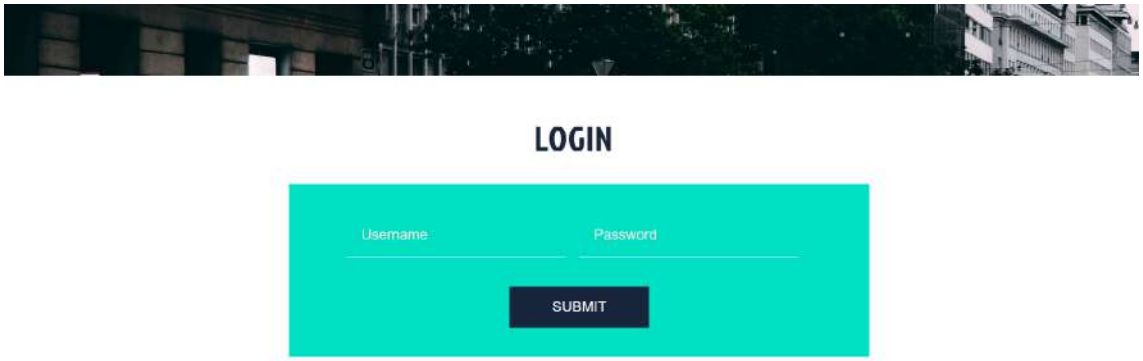
C.1 HOME



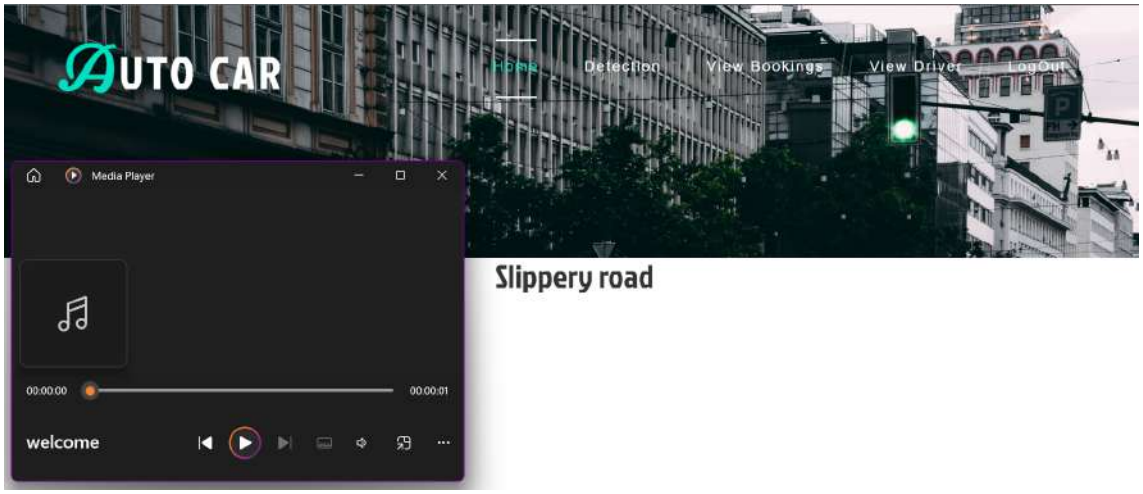
C.2 REGISTRATION



C.3 LOGIN



C.4 DETECTION



C.5 VIEW DRIVERS

NAME	ADDRESS	CONTACT	EMAIL	ACTION
Williams	Kodungalloor	8988713401	williams@gmail.com	Book Now
Simson	Thrissur	9099170787	simson@gmail.com	Book Now
Myna	Chalakkudi	8799136323	myna@gmail.com	Book Now

C.6 VIEW BOOKING

NAME	ADDRESS	CONTACT	EMAIL	STATUS	ACTION
Richard	Chalakkudi	8966139256	richard321@gmail.com		Approve
John	Aluva	8590939355	john@gmail.com		Approve
Philip	Kodungalloor	9890583346	philip@gmail.com		Approve

D Code

urls.py

```
from django.contrib import admin
from django.urls import path
from textapp import views
urlpatterns = [
    path('admin/', admin.site.urls),

    path('index/', views.index),
    path('adminhome/', views.adminhome),
    path('driverhome/', views.driverhome),
    path('userhome/', views.userhome),
    path('', views.index),
    path('login/', views.logins),
    path('userreg/', views.userreg),
    path('driverreg/', views.driverreg),
    path('udp/', views.udp),
    path('adminviewdriver/', views.adminviewdriver),
    path('adminviewuser/', views.adminviewuser),
    path('userviewdriver/', views.userviewdriver),
    path('viewbooking/', views.userviewbooking),
    path('userviewbooking/', views.userviewbooking),
    path('driverviewbooking/', views.driverviewbooking),
    path('ddetection/', views.ddetection),
    path('driverdetection/', views.driverdetection),
    path('imagebyenter/', views.imagebyenter),
    path('prediction/', views.predict),
    path('deletedriver/', views.deletedriver),
    path('deleteuser/', views.deleteuser),
    \# path('udp/', views.udp),

]
```

model.py

```
import os
from tensorflow.keras.preprocessing import image
import matplotlib.pyplot as plt
```

```
import numpy as np
import tensorflow.python.keras.utils as generic\_utils
import random,shutil
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dropout,Conv2D,Flatten,Dense,
MaxPooling2D, BatchNormalization
from tensorflow.keras.models import load\_model


def generator(dir, gen=image.ImageDataGenerator(rescale=1./255), shuffle=True,batch
\_size=1,target\_size=(24,24),class\_mode='categorical' ):

return gen.flow\_from\_directory(dir,batch\_size=batch\_size,shuffle=shuffle,
color\_mode='grayscale',class\_mode=class\_mode,target\_size=target\_size)

BS= 32
TS=(24,24)
train\_batch= generator('data/train',shuffle=True, batch\_size=BS,target\_size=TS)
valid\_batch= generator('data/valid',shuffle=True, batch\_size=BS,target\_size=TS)
SPE= len(train\_batch.classes)//BS
VS = len(valid\_batch.classes)//BS
print(SPE,VS)


\# img,labels= next(train\_batch)
\# print(img.shape)

model = Sequential([
    Conv2D(32, kernel\_size=(3, 3), activation='relu', input\_shape=(24,24,1)),
    MaxPooling2D(pool\_size=(1,1)),
    Conv2D(32,(3,3),activation='relu'),
    MaxPooling2D(pool\_size=(1,1)),
\#32 convolution filters used each of size 3x3
\#again
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D(pool\_size=(1,1)),

\#64 convolution filters used each of size 3x3
\#choose the best features via pooling
```

```
\#randomly turn neurons on and off to improve convergence
    Dropout(0.25),
\#flatten since too many dimensions, we only want a classification output
    Flatten(),
\#fully connected to get all relevant data
    Dense(128, activation='relu'),
\#one more dropout for convergence' sake :)
    Dropout(0.5),
\#output a softmax to squash the matrix into output probabilities
    Dense(2, activation='softmax')
])

model.compile(optimizer='adam',loss='categorical\_crossentropy',metrics=['accuracy'])

model.fit\_generator(train\_batch, validation\_data=valid\_batch,epochs=15,steps\_per

model.save('models/cnnCat2.h5', overwrite=True)
```

models.py

```
[verbetim]

from django.db import models




class Driver(models.Model):
    dname=models.CharField(max\_length=100,blank=True)
    daddress=models.CharField(max\_length=100,blank=True)
    dcontact=models.CharField(max\_length=100,blank=True)
    demail=models.CharField(max\_length=100,blank=True)
    dpassword=models.CharField(max\_length=100,blank=True)
    status=models.CharField(max\_length=100,blank=True)


class Login(models.Model):
    username=models.CharField(max\_length=100,blank=True)
    password=models.CharField(max\_length=100,blank=True)
```

```
    usertype=models.CharField(max\_length=100,blank=True)
    status=models.CharField(max\_length=100,blank=True)
    driverid=models.ForeignKey(Driver,null=True,on\_delete=models.CASCADE)



class User(models.Model):
    uname=models.CharField(max\_length=100,blank=True)
    uaddress=models.CharField(max\_length=100,blank=True)
    ucontact=models.CharField(max\_length=100,blank=True)
    uemail=models.CharField(max\_length=100,blank=True)
    upassword=models.CharField(max\_length=100,blank=True)


class Booking(models.Model):
    uid=models.ForeignKey(User,null=True,on\_delete=models.CASCADE)
    did=models.ForeignKey(Driver,null=True,on\_delete=models.CASCADE)
    date=models.CharField(max\_length=100,blank=True)
    status=models.CharField(max\_length=100,blank=True)
```

views.py

```
\# import email
from http.client import HTTPResponse
from django.shortcuts import render,HttpResponse,HttpResponseRedirect

from .models import *
from django.contrib import messages
from django.db.models import Max, Min,Count,Sum,Avg

import numpy as np
from PIL import Image
import cv2
import tensorflow as tf
import os
from flask\_cors import CORS, cross\_origin
from utils.utils import decodeImage
from predict import traffic
```

```
from django.db.models import Q
import cv2
import tensorflow as tf

def udp(request):
    import os
    from gtts import gTTS
    \# email="admin@gmail.com"
    \# password="admin"

    \# s=Login.objects.create(username=email,password=password,usertype='admin',statu
    \# s.save()
    \# messages.info(request,"already added")
    model\_path = "Traffic.h5"
    loaded\_model = tf.keras.models.load\_model(model\_path)
    videoCaptureObject = cv2.VideoCapture(0)
    print("**************************************************************")
    result = True
    while(result):
        ret,frame = videoCaptureObject.read()
        cv2.imwrite("NewPicture.jpg",frame)
        result = False
    videoCaptureObject.release()
    cv2.destroyAllWindows()
    imagename = "NewPicture.jpg"
    image = cv2.imread(imagename)
    print(image)
    print("**************************************************************")

    image\_fromarray = Image.fromarray(image, 'RGB')
    resize\_image = image\_fromarray.resize((30, 30))
    print("**************************************************************")

    expand\_input = np.expand\_dims(resize\_image,axis=0)
    input\_data = np.array(expand\_input)
    input\_data = input\_data/255
    pred = loaded\_model.predict(input\_data)
    result = pred.argmax()
    print("**************************************************************")
```

```
if result == 0:
    prediction = 'Speed limit (20km/h)'
    print("image",prediction)
elif result == 1:
    prediction = 'Speed limit (30km/h)'
    print("image",prediction)
elif result == 2:
    prediction = 'Speed limit (50km/h)'
    print("image",prediction)
elif result == 3:
    prediction = 'Speed limit (60km/h)'
    print("image",prediction)
elif result == 4:
    prediction = 'Speed limit (70km/h)'
    print("image",prediction)
elif result == 5:
    prediction = 'Speed limit (80km/h)'
    print("image",prediction)
elif result == 6:
    prediction = 'End of speed limit (80km/h)'
    print("image",prediction)
elif result == 7:
    prediction = 'Speed limit (100km/h)'
    print("image",prediction)
elif result == 8:
    prediction = 'Speed limit (120km/h)'
    print("image",prediction)
elif result == 9:
    prediction = 'No passing'
    print("image",prediction)
elif result == 10:
    prediction = 'No passing veh over 3.5 tons'
    print("image",prediction)
elif result == 11:
    prediction = 'Right-of-way at intersection'
    print("image",prediction)
elif result == 12:
    prediction = 'Priority road'
    print("image",prediction)
```

```
elif result == 13:
    prediction = 'Yield'
    print("image",prediction)
elif result == 14:
    prediction = 'Stop'
    print("image",prediction)
elif result == 15:
    prediction = 'No vehicles'
    print("image",prediction)
elif result == 16:
    prediction = 'Veh > 3.5 tons prohibited'
    print("image",prediction)
elif result == 17:
    prediction = 'No entry'
    print("image",prediction)
elif result == 18:
    prediction = 'General caution'
    print("image",prediction)
elif result == 19:
    prediction = 'Dangerous curve left'
    print("image",prediction)
elif result == 20:
    prediction = 'Dangerous curve right'
    print("image",prediction)
elif result == 21:
    prediction = 'Double curve'
    print("image",prediction)
elif result == 22:
    prediction = 'Bumpy road'
    print("image",prediction)
elif result == 23:
    prediction = 'Slippery road'
    print("image",prediction)
elif result == 24:
    prediction = 'Road narrows on the right'
    print("image",prediction)
elif result == 25:
    prediction = 'Road work'
    print("image",prediction)
elif result == 26:
```

```
        prediction = 'Traffic signals'
        print("image",prediction)
    elif result == 27:
        prediction = 'Pedestrians'
        print("image",prediction)
    elif result == 28:
        prediction = 'Children crossing'
        print("image",prediction)
    elif result == 29:
        prediction = 'Bicycles crossing'
        print("image",prediction)
    elif result == 30:
        prediction = 'Beware of ice/snow'
        print("image",prediction)
    elif result == 31:
        prediction = 'Wild animals crossing'
        print("image",prediction)
    elif result == 32:
        prediction = 'End speed + passing limits'
        print("image",prediction)
    elif result == 33:
        prediction = 'Turn right ahead'
        print("image",prediction)
    elif result == 34:
        prediction = 'Turn left ahead'
        print("image",prediction)
    elif result == 35:
        prediction = 'Ahead only'
        print("image",prediction)
    elif result == 36:
        prediction = 'Go straight or right'
        print("image",prediction)
    elif result == 37:
        prediction = 'Go straight or left'
        print("image",prediction)
    elif result == 38:
        prediction = 'Keep right'
        print("image",prediction)
    elif result == 39:
        prediction = 'Keep left'
```

```
        print("image",prediction)
    elif result == 40:
        prediction = 'Roundabout mandatory'
        print("image",prediction)
    elif result == 41:
        prediction = 'End of no passing'
        print("image",prediction)
    elif result == 42:
        prediction = 'End no passing veh > 3.5 tons'
        print("image",prediction)
    language = 'en'
    myobj = gTTS(text=prediction, lang=language, slow=False)
    print("**************************************************")
    print(prediction)
    myobj.save("welcome.mp3")
    os.system("welcome.mp3")
    return render(request,"user/prediction.html",{"prediction":prediction})



    return HttpResponseRedirect("/prediction")


from django.core.files.storage import FileSystemStorage

def imagebyenter(request):
    import os
    from gtts import gTTS
    if request.POST:
        image=request.FILES['image']
        saved\_filename = f"./uploads/{image.name}"
        with open(saved\_filename, "wb") as opened:
            for chunk in image.chunks():
                    opened.write(chunk)

        fs=FileSystemStorage()
        filename=fs.save(image.name,image)
        fileurl=fs.url(filename)

        model\_path = "Traffic.h5"
```

```
loaded\_model = tf.keras.models.load\_model(model\_path)

imagename = image
print(saved\_filename)
image = cv2.imread(saved\_filename)

image\_fromarray = Image.fromarray(image, 'RGB')
resize\_image = image\_fromarray.resize((30, 30))
expand\_input = np.expand\_dims(resize\_image,axis=0)
input\_data = np.array(expand\_input)
input\_data = input\_data/255
pred = loaded\_model.predict(input\_data)
result = pred.argmax()


if result == 0:
    prediction = 'Speed limit (20km/h)'
    print("image",prediction)
elif result == 1:
    prediction = 'Speed limit (30km/h)'
    print("image",prediction)
elif result == 2:
    prediction = 'Speed limit (50km/h)'
    print("image",prediction)
elif result == 3:
    prediction = 'Speed limit (60km/h)'
    print("image",prediction)
elif result == 4:
    prediction = 'Speed limit (70km/h)'
    print("image",prediction)
elif result == 5:
    prediction = 'Speed limit (80km/h)'
    print("image",prediction)
elif result == 6:
    prediction = 'End of speed limit (80km/h)'
    print("image",prediction)
elif result == 7:
    prediction = 'Speed limit (100km/h)'
    print("image",prediction)
elif result == 8:
```

```
        prediction = 'Speed limit (120km/h)'
        print("image",prediction)
    elif result == 9:
        prediction = 'No passing'
        print("image",prediction)
    elif result == 10:
        prediction = 'No passing veh over 3.5 tons'
        print("image",prediction)
    elif result == 11:
        prediction = 'Right-of-way at intersection'
        print("image",prediction)
    elif result == 12:
        prediction = 'Priority road'
        print("image",prediction)
    elif result == 13:
        prediction = 'Yield'
        print("image",prediction)
    elif result == 14:
        prediction = 'Stop'
        print("image",prediction)
    elif result == 15:
        prediction = 'No vehicles'
        print("image",prediction)
    elif result == 16:
        prediction = 'Veh > 3.5 tons prohibited'
        print("image",prediction)
    elif result == 17:
        prediction = 'No entry'
        print("image",prediction)
    elif result == 18:
        prediction = 'General caution'
        print("image",prediction)
    elif result == 19:
        prediction = 'Dangerous curve left'
        print("image",prediction)
    elif result == 20:
        prediction = 'Dangerous curve right'
        print("image",prediction)
    elif result == 21:
        prediction = 'Double curve'
```

```
        print("image",prediction)
    elif result == 22:
        prediction = 'Bumpy road'
        print("image",prediction)
    elif result == 23:
        prediction = 'Slippery road'
        print("image",prediction)
    elif result == 24:
        prediction = 'Road narrows on the right'
        print("image",prediction)
    elif result == 25:
        prediction = 'Road work'
        print("image",prediction)
    elif result == 26:
        prediction = 'Traffic signals'
        print("image",prediction)
    elif result == 27:
        prediction = 'Pedestrians'
        print("image",prediction)
    elif result == 28:
        prediction = 'Children crossing'
        print("image",prediction)
    elif result == 29:
        prediction = 'Bicycles crossing'
        print("image",prediction)
    elif result == 30:
        prediction = 'Beware of ice/snow'
        print("image",prediction)
    elif result == 31:
        prediction = 'Wild animals crossing'
        print("image",prediction)
    elif result == 32:
        prediction = 'End speed + passing limits'
        print("image",prediction)
    elif result == 33:
        prediction = 'Turn right ahead'
        print("image",prediction)
    elif result == 34:
        prediction = 'Turn left ahead'
        print("image",prediction)
```

```
        elif result == 35:
            prediction = 'Ahead only'
            print("image",prediction)
        elif result == 36:
            prediction = 'Go straight or right'
            print("image",prediction)
        elif result == 37:
            prediction = 'Go straight or left'
            print("image",prediction)
        elif result == 38:
            prediction = 'Keep right'
            print("image",prediction)
        elif result == 39:
            prediction = 'Keep left'
            print("image",prediction)
        elif result == 40:
            prediction = 'Roundabout mandatory'
            print("image",prediction)
        elif result == 41:
            prediction = 'End of no passing'
            print("image",prediction)
        elif result == 42:
            prediction = 'End no passing veh > 3.5 tons'
            print("image",prediction)
        language = 'en'
        myobj = gTTS(text=prediction, lang=language, slow=False)
        myobj.save("welcome.mp3")
        os.system("welcome.mp3")
        return render(request,"user/prediction.html",{"prediction":prediction})


    return render(request,"user/enterimage.html")

def predict(request):

    return render(request,"user/prediction.html")
```

```
def index(request):
    return render(request,"common/index.html")

def adminhome(request):
    return render(request,"admin/index.html")

def userhome(request):
    return render(request,"user/index.html")

def driverhome(request):
    return render(request,"driver/index.html")

def driverreg(request):
    if request.POST:
        name=request.POST.get("name")
        address=request.POST.get("address")
        contact=request.POST.get("contact")
        email=request.POST.get("email")
        password=request.POST.get("password")

        s=Driver.objects.create(dname=name,daddress=address,dcontact=contact,demail=e
        s.save()
        did=Driver.objects.aggregate(Max('id'))
        print(did)
        did=did['id\_\_max']
        did=Driver.objects.get(id=did)
        print(did)

    \# print(mark)
        s=Login.objects.create(username=email,password=password,usertype='driver',
        status='requested',driverid=did)
        s.save()
        messages.info(request,"already added")
    return render(request,"common/driverreg.html")



def userreg(request):
```

```
    if request.POST:
        name=request.POST.get("name")
        address=request.POST.get("address")
        contact=request.POST.get("contact")
        email=request.POST.get("email")
        password=request.POST.get("password")

        s=User.objects.create(uname=name,uaddress=address,ucontact=contact,
            uemail=email,upassword=password)
        s.save()
        s=Login.objects.create(username=email,password=password,
             usertype='user',status='requested')
        s.save()
        messages.info(request,"already added")
    return render(request,"common/userreg.html")


def logins(request):
    msg=""
    if request.POST:
        name=request.POST.get("username")
        password=request.POST.get("password")
        print(name)
        print(password)

        s=Login.objects.filter(Q(username=name),Q(password=password))

        try:
            if s[0].usertype=='admin':
                msg="Success"
                return HttpResponseRedirect("/adminhome")
            elif s[0].usertype=='user':
                s=User.objects.get(uemail=name)
                id=s.id
                request.session['uid']=id
                msg="Success"
                return HttpResponseRedirect("/userhome")
            elif s[0].usertype=='driver':
                s=Driver.objects.get(demail=name)
                id=s.id
```

```
                request.session['did']=id
                msg="Success"
                return HttpResponseRedirect("/driverhome")

        except:
            msg="invalid Username Or password"


    \# return HttpResponseRedirect('/login')

    return render(request,"common/login.html",{'msg':msg})




def adminviewdriver(request):
    s=Driver.objects.filter()
    print(s)
    if request.GET:
        id=request.GET.get("id")
        s=Driver.objects.get(id=id)
        username=s.demail
        s=Login.objects.filter(username=username).update(status='approved')
        s=Driver.objects.filter(id=id).update(status='approved')
        return HttpResponseRedirect("/adminviewdriver")
    return render(request,"admin/viewdriver.html",{"data":s})

def deletedriver(request):
    id=request.GET.get('id')
    Driver.objects.filter(id=id).delete()
    return HttpResponseRedirect("/adminviewdriver")


def adminviewuser(request):
    s=User.objects.all()
    print(s)
    return render(request,"admin/viewuser.html",{"data":s})

def deleteuser(request):
    id=request.GET.get('id')
```

```
    User.objects.filter(id=id).delete()
    return HttpResponseRedirect("/adminviewuser")


def userviewdriver(request):
    s=Driver.objects.all()
    uid=request.session['uid']
    if request.GET:
        id=request.GET.get("id")
        driverdetails=Driver.objects.get(id=id)
        userdetails=User.objects.get(id=uid)
        import datetime
        date=datetime.datetime.now()
        s=Booking.objects.create(uid=userdetails,did=driverdetails,date=date,status='
        return HttpResponseRedirect("/userviewdriver")
    return render(request,"user/viewdriver.html",{"data":s})




def userviewbooking(request):
    uid=request.session['uid']
    select=Booking.objects.filter(uid=uid)
    return render(request,"user/viewbooking.html",{"data":select})




def driverviewbooking(request):
    uid=request.session['did']
    print(uid)
    select=Booking.objects.filter(did=uid)
    print(select)
    abcd='approved'
    if request.GET:
        id=request.GET.get("id")
        select=Booking.objects.filter(id=id).update(status='approved')
        return HttpResponseRedirect("/driverhome")

    return render(request,"driver/viewbooking.html",{"data":select,"abcd":abcd})
```

```
def ddetection(request):
    from textapp import cd
    return HttpResponseRedirect("/userhome")



def driverdetection(request):
    from textapp import cd
    return HttpResponseRedirect("/driverhome")
```

predict.py

```
import numpy as np
from PIL import Image
import cv2
import tensorflow as tf

class traffic:
    def \_\_init\_\_(self,filename):
        self.filename =filename


    def trafficsign(self):

        model\_path = "Traffic.h5"
        loaded\_model = tf.keras.models.load\_model(model\_path)

        imagename = self.filename
        image = cv2.imread(imagename)

        image\_fromarray = Image.fromarray(image, 'RGB')
        resize\_image = image\_fromarray.resize((30, 30))
        expand\_input = np.expand\_dims(resize\_image,axis=0)
        input\_data = np.array(expand\_input)
        input\_data = input\_data/255
```

```
pred = loaded\_model.predict(input\_data)
result = pred.argmax()

if result == 0:
    prediction = 'Speed limit (20km/h)'
    return [{"image": prediction}]
elif result == 1:
    prediction = 'Speed limit (30km/h)'
    return [{"image": prediction}]
elif result == 2:
    prediction = 'Speed limit (50km/h)'
    return [{"image": prediction}]
elif result == 3:
    prediction = 'Speed limit (60km/h)'
    return [{"image": prediction}]
elif result == 4:
    prediction = 'Speed limit (70km/h)'
    return [{"image": prediction}]
elif result == 5:
    prediction = 'Speed limit (80km/h)'
    return [{"image": prediction}]
elif result == 6:
    prediction = 'End of speed limit (80km/h)'
    return [{"image": prediction}]
elif result == 7:
    prediction = 'Speed limit (1000km/h)'
    return [{"image": prediction}]
elif result == 8:
    prediction = 'Speed limit (120km/h)'
    return [{"image": prediction}]
elif result == 9:
    prediction = 'No passing'
    return [{"image": prediction}]
elif result == 10:
    prediction = 'No passing veh over 3.5 tons'
    return [{"image": prediction}]
elif result == 11:
    prediction = 'Right-of-way at intersection'
    return [{"image": prediction}]
elif result == 12:
```

```
        prediction = 'Priority road'
        return [{"image": prediction}]
    elif result == 13:
        prediction = 'Yield'
        return [{"image": prediction}]
    elif result == 14:
        prediction = 'Stop'
        return [{"image": prediction}]
    elif result == 15:
        prediction = 'No vehicles'
        return [{"image": prediction}]
    elif result == 16:
        prediction = 'Veh > 3.5 tons prohibited'
        return [{"image": prediction}]
    elif result == 17:
        prediction = 'No entry'
        return [{"image": prediction}]
    elif result == 18:
        prediction = 'General caution'
        return [{"image": prediction}]
    elif result == 19:
        prediction = 'Dangerous curve left'
        return [{"image": prediction}]
    elif result == 20:
        prediction = 'Dangerous curve right'
        return [{"image": prediction}]
    elif result == 21:
        prediction = 'Double curve'
        return [{"image": prediction}]
    elif result == 22:
        prediction = 'Bumpy road'
        return [{"image": prediction}]
    elif result == 23:
        prediction = 'Slippery road'
        return [{"image": prediction}]
    elif result == 24:
        prediction = 'Road narrows on the right'
        return [{"image": prediction}]
    elif result == 25:
        prediction = 'Road work'
```

```
        return [{"image": prediction}]
    elif result == 26:
        prediction = 'Traffic signals'
        return [{"image": prediction}]
    elif result == 27:
        prediction = 'Pedestrians'
        return [{"image": prediction}]
    elif result == 28:
        prediction = 'Children crossing'
        return [{"image": prediction}]
    elif result == 29:
        prediction = 'Bicycles crossing'
        return [{"image": prediction}]
    elif result == 30:
        prediction = 'Beware of ice/snow'
        return [{"image": prediction}]
    elif result == 31:
        prediction = 'Wild animals crossing'
        return [{"image": prediction}]
    elif result == 32:
        prediction = 'End speed + passing limits'
        return [{"image": prediction}]
    elif result == 33:
        prediction = 'Turn right ahead'
        return [{"image": prediction}]
    elif result == 34:
        prediction = 'Turn left ahead'
        return [{"image": prediction}]
    elif result == 35:
        prediction = 'Ahead only'
        return [{"image": prediction}]
    elif result == 36:
        prediction = 'Go straight or right'
        return [{"image": prediction}]
    elif result == 37:
        prediction = 'Go straight or left'
        return [{"image": prediction}]
    elif result == 38:
        prediction = 'Keep right'
        return [{"image": prediction}]
```

```
elif result == 39:
    prediction = 'Keep left'
    return [{"image": prediction}]
elif result == 40:
    prediction = 'Roundabout mandatory'
    return [{"image": prediction}]
elif result == 41:
    prediction = 'End of no passing'
    return [{"image": prediction}]
elif result == 42:
    prediction = 'End no passing veh > 3.5 tons'
    return [{"image": prediction}]
else:
    return [{"ERROR": "Please select another image. !!!"}]
```

DEPARTMENT OF COMPUTER SCIENCE

CHRIST COLLEGE (AUTONOMOUS)

IRINJALAKUDA

CERTIFICATE

This is to certify that the project report entitled ***"Air Quality Index"*** *is a bonfied record of the project work done by* ***Abhay S Nair*** *in partial fulfillment of the requirement for the sixth semester of* ***Bachelor of Computer Application*** *in Department of Computer Science of* **CHRIST COLLEGE (AUTONOMOUS) IRINJALAKUDA**

Ms. Viji Viswanathan
Assistant Professor,CS
Internal Guide

28/03/24

EXTERNAL EXAMINER

Christ College (Autonomous) * Department of Computer Science * Irinjalakuda Pin - 680 125

Ms. Sini Thomas
Head of the Department
Computer Science

25/3/24

INTERNAL EXAMINER

DECLARATION

We hereby declare that this project work **"AIR QUALITY INDEX"** submitted to Christ College (Autonomous) Irinjalakuda, affiliated to Calicut University in partial fulfillment of the requirement for the award of the Bachelor of Computer Application, is a record of original work done by us, under the guidance of Ms. VIJI VISWANATHAN, Department of computer Science.

Place: Irinjalakuda

ABHAY S NAIR

ACKNOWLEDGEMENT

First and foremost we like to thank Lord almighty for his providence and for being the guiding light throughout the project. We wish to express my sincere gratitude to our beloved Department head for giving me all the facilities for our project. We take this opportunity to express my gratitude to the class teacher Ms. SOUMYA P.S and head of the department Ms. SINI THOMAS who has been supported us throughout the course of this project. We are thankful for her aspiring guidance and valuable advice during the project work. We express my sincere thanks to my project guide Ms. VIJI VISWANATHAN for supporting and guiding throughout the project.We would take this opportunity to specially thank all other faculty members for their constant and continuous motivation. Finally we would like to thank my family and friends for giving valuable advice and moral support throughout our project.

ABSTRACT

Air Quality Index is a innovative web application that takes on paramount importance as it addresses the need for timely and accurate information about air quality. The web application is enriched with a login form ,a registratin form and a prediction window where we enter the prediction attributes.This application gives accurate air qualty and a graphical visualization for the same. All these features make this website more adaptable and user-friendly.

Contents

Chapter 1

1 Introduction

Air quality is a critical component of environmental health management and its significance cannot be overstated.Air pollution has become a most pressing concern and it has affected not only the environment but also the health of billions of people. Air quality indexing takes on importance as it addresses the timely and accurate information about the air quality.The purpose of the system is to create a robust and reliable system that not only monitors air quality but also predicts its future trends.This step represents a critial step forward in environmental management and how we safeguard the air we breathe.

1.1 Overview

The core objective of the project is to develop an Air quality indexing and prediction system that harness the power of data and predictive analytics to provide real-time insights to air quality. The development of this system can lead to a healthier and more sustainable future.

Chapter 2

2 System Analysis

2.1 Purpose

The purpose of this project is to develop an Air quality indexing and prediction system that harness the power of data and predictive analytics to provide real-time insights to air quality.

2.1.1 Existing System

Presently all the processes gives very simple outlook. It is provided with login and registeration pages. The web application is developed using HTML,CSS and Bootstrap in the front end. Backend is developed using Python and the framework being django. Limitations of existing system :No Graph provided,No login and registeration pages provided,

2.1.2 Proposed System

As a software developer, we are expected to design and develop any program that works efficiently without any delays.For Convinence we provider a login and registeration pages to make it a web application type project . And with the addition of a Graph to make the project more understandable to the users.

2.2 Problem definition

To know what the problem is and what the needs are before developing it. Here we design a web application for detecting the air quality

2.3 FEASIBILITY STUDY

After the problem is clearly understood and solutions proposed, the next step is to conduct the feasibility study. Feasibility study is defined as evaluation or analysis of the potential impact of the proposed project or program. The objective is to determine whether the proposed system is feasible. There are three aspects of feasibility study to which the proposed system is subjected as discussed below.

2.3.1 Technical Feasibility

Technical feasibility assess whether the current technical resources are sufficient for the new system. We can upgrade the level of technology for supporting our web application. We check whether the proposed system can be implemented in the present system without supporting the existing hardware.

Chapter 3

3 Software Requirement Specification

3.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the Project Air quality detection system. It illustrate the purpose and complete description for the development of the system. It explain system constraints, interface and interactions with other external applications. This document is primarily intended to harness the power of data and predictive analytics to provide real-time insights to air quality

3.2 Scope

Our project has made it easier to detect the air quality in real-time . We can get all information about the quality of air wheather it is good or moderate or poor etc with a suitable graph included .

3.3 Overall Description

This section give an overview of our web application is to create a robust and reliable system that not only monitors air quality but also predicts its future trends.This step represents a critial step forward in environmental management and how we safeguard the air we breathe.

3.3.1 Product Perspective

The product perspective of this Air Quality Detection System involves designing and integrating these components to provide accurate,timely and actionable information to the users abut the air they breathe.

3.3.2 Product Functionality

This system can empower users to make informed decesion about their health and well-being in relation to air pollution .

3.3.3 Users and Characteristics

Each user group may have different needs,preferences and levels of technical expertise,so the Air Quality Detection System should be designed with user-centric features and interfaces to accomodate the diverse characterstics effectively.

3.4 Specific Requirements

3.4.1 Hardware Requirements

- System: IBM-Compatible PC
- Processor: Pentium IV or above
- Speed: Above 1GHz
- RAM capacity: 512 MB
- Hard Dsk drive: 40 GB
- Keyboard: Standard
- Mouse: Standard
- Monitor: SVGA Color

3.4.2 Software Requirements

- Operating System:Windows or ubuntu
- Languages used: Python Django
- Data Collection : CSV
- Technologies used: HTML,CSS,Bootsrap

3.5 Functional Requirements

It contains three main modules.

- 1.Login Form
- 2.Registration form
- 3. Prediction window

login Form

The login page is for the logging in the necessary credentials into the project for accessing into the main index page of the project.

Registration Form

In case if the user doesn't have an account he/she might not be able to log in.Therefore a registration form is created to enter the provided details into the form and after registering the user cn go back to the login page to log in the necessary credentials.

PRODUCT MANAGEMENT SYSTEM

PROJECT REPORT

Submitted By

ABHISHEK R NAIR

Reg. No. CCAVBCA016

for the award of the Degree of

Bachelor of Computer Application (BCA)

in Computer Application
(University of Calicut)

under the guidance of

Ms. Vandana T.V

Assistant Professor

BACHELOR OF COMPUTER APPLICATION
DEPARTMENT OF COMPUTER SCIENCE
CHRIST COLLEGE(Autonomous)
IRINJALAKUDA, KERALA
2021-2024

DEPARTMENT OF COMPUTER SCIENCE

CHRIST COLLEGE (AUTONOMOUS)

IRINJALAKUDA

CERTIFICATE

This is to certify that the project report entitled ***"Product Management System"*** *is a bonafide record of the project work done by* ***Abhishek R Nair*** *in partial fulfillment of the requirement for the sixth semester of* ***Bachelor of Computer Application*** *in Department of Computer Science of* ***CHRIST COLLEGE (AUTONOMOUS) IRINJALAKUDA***

.

Ms. Vandana T.V
Assistant Professor,CS
Internal Guide

Ms. Sini Thomas
Head of the Department
Computer Science

EXTERNAL EXAMINER **INTERNAL EXAMINER**

KALIPARAMBIL STORES

TO WHOMSOEVER IT MAY CONCERN

This is to certify that **Mr.ABHISHEK R NAIR**
(CCAVBCA016)
student of
CHRIST COLLEGE (AUTONOMOUS), IRINJALAKUDA
(Bachelor Of Computer Application)
has successfully completed their project
PRODUCT MANAGEMENT SYSTEM under the guidance of
Ms. VANDANA TV (guide) and implemented in
KALIPARAMBIL STORES on 29-01-2024
during the academic year 2021-2024.

MUJEEB.KB

KALIPARAMBIL STORES
MANAGER

KALIPARAMBIL STORES
Proprietor

Vellikulangara Junction

DECLARATION

We hereby declare that this project work **"PRODUCT MANAGEMENT SYSTEM"** submitted to Christ College (Autonomous) Irinjalakuda, affiliated to Calicut University in partial fulfillment of the requirement for the award of the Bachelor of Computer Applications, is a record of original work done by us, under the guidance of Ms. VANDANA T.V, Department of computer Science.

Place: Irinjalakuda ABHISHEK R NAIR

ACKNOWLEDGEMENT

First and foremost we like to thank Lord almighty for his providence and for being the guiding light throughout the project. We wish to express my sincere gratitude to our beloved Department head for giving me all the facilities for our project. We take this opportunity to express my gratitude to the class teacher Ms. SOWMYA P.S and head of the department Ms. SINI THOMAS who has been supported us throughout the course of this project. We are thankful for her aspiring guidance and valuable advice during the project work. We express my sincere thanks to my project guide Ms. VANDANA T.V for supporting and guiding throughout the project.We would take this opportunity to specially thank all other faculty members for their constant and continuous motivation. Finally we would like to thank my family and friends for giving valuable advice and moral support throughout our project.

ABSTRACT

PRODUCT MANAGEMENT SYSTEM for Retail Stores represents a groundbreaking solution aimed at revolutionizing inventory management and enhancing operational efficiency within the retail sector. This innovative web application offers a comprehensive suite of features, including intuitive inventory management, proactive notifications for expiring products, promotion management capabilities, robust search functionality, seamless integration with point-of-sale systems, and insightful dashboard analytics. Through its user-friendly interface and advanced functionalities, the system empowers retail store owners and employees to optimize inventory processes, drive sales, and make data-driven decisions. With future enhancements focused on supply chain integration, mobile accessibility, and advanced analytics, the proposed system promises to redefine the retail experience and set new standards for success in the industry.

Contents

Chapter 1

1 Introduction

The Product Management System for Retail Stores heralds a new era of efficiency and optimization within the bustling world of retail. This innovative web application emerges as a comprehensive solution tailored to address the complex challenges inherent in managing store inventory and streamlining retail operations. At its core, this project aims to empower retail store owners and employees with a robust platform that seamlessly integrates inventory management, notification systems for expiring products, promotion management, product search functionality, electronic bill generation, and insightful dashboard analytics. By combining cutting-edge technology with intuitive design, the system revolutionizes the way retail businesses approach their day-to-day operations, fostering increased productivity, minimized wastage, and enhanced customer satisfaction. At the forefront of this project lies a user-friendly interface that serves as the gateway to a myriad of powerful features. Through a series of meticulously designed forms and interactive elements, users can effortlessly navigate the system, adding, editing, and removing products while inputting vital details such as product names, expiry dates, quantities, and prices. This meticulous attention to detail ensures that the inventory database remains accurate and up-to-date, providing users with a solid foundation upon which to build their retail endeavors. One of the standout features of the system is its proactive approach to managing expiring products. Through a sophisticated notification system, users receive timely alerts when products are nearing their expiry date, enabling them to take swift action to prevent wastage and capitalize on opportunities to drive sales through targeted promotions.

1.1 Overview

The Product Management System is born from a vision to revolutionize the way retail stores manage their inventory, sales, and operational tasks. With the ever-increasing complexity of product management in modern retail, our team recognized the need for a robust, user-friendly solution that integrates essential features to enhance efficiency and productivity.

Chapter 2

2 System Analysis

2.1 Purpose

This documentation serves as a comprehensive resource for understanding the design, functionality, and implementation of the Product Management System. Whether you are a developer looking to extend the system, a store manager seeking to optimize operations, or an investor evaluating the potential impact of our solution, this document provides the insights and information necessary to engage with our project effectively.

2.1.1 Existing System

The current Product Management System in use at our retail store represents a fundamental tool for managing inventory, sales, and operational tasks. Developed to address the challenges of product management in a retail environment, the system offers a range of essential features aimed at facilitating efficient store operations and enhancing customer satisfaction.

2.1.2 Proposed System

The proposed Product Management System for Retail Stores is a comprehensive solution designed to revolutionize inventory management and streamline retail operations. Key features include an intuitive user interface for efficient inventory management, automated notifications for expiring products, promotion management functionality to drive sales, robust search capabilities for quick access to product information, seamless integration with point-of-sale systems for smooth transactions, and insightful dashboard analytics for data-driven decision-making. Future scope includes integration with supply chain management for automated ordering and replenishment, development of a mobile application version for increased accessibility, and enhancement of analytics capabilities for deeper insights. Overall, the proposed system aims to empower retail store owners and employees with the tools they need to optimize their operations, minimize costs, and enhance customer satisfaction in today's competitive retail environment. Advantages of proposed system :unique page are provided, Automatic expiry notification when product reaches its expiry date and qr code is provided to each product get a overview about the product

2.2 Problem definition

To know what the problem is and what the needs are before developing it. Here we design a web application of retail stores for managing and tracking their product details

2.3 Feasibility study

After the problem is clearly understood and solutions proposed, the next step is to conduct the feasibility study. Feasibility study is defined as evaluation or analysis of the potential impact of the proposed project or program. The objective is to determine whether the proposed system is feasible. There are three aspects of feasibility study to which the proposed system is subjected as discussed below.

2.3.1 Technical Feasibility

Technical feasibility assess whether the current technical resources are sufficient for the new system. We can upgrade the level of technology for supporting our webapplication. We check whether the proposed system can be implemented in the present system without supporting the existing hardware.

2.3.2 Economical Feasibility

Economic feasibility determines whether the time and money are available to develop the system. There is no additional hardware used to develop the site, it is inexpensive to build.

2.3.3 Operational Feasibility

Operational feasibility determines if the human resources are available to operate the system once it has been installed. No extra training is needed to use this system. Anyone who has knowledge in internet in english language can easily use the system. The resources that are required to implement or install are already available with the organization.

Chapter 3

3 Software Requirement Specification

3.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the PRODUCT MANAGEMENT SYSTEM. It illustrate the purpose and complete description for the development of the system. It explain system constraints, interface and interactions with other external applications. This document is primarily intended to be proposed to a store for managing and tracking the product details and help the user to gain profit.

3.2 Scope

Our project has made it easier for store to manage products easily and systematically . We can get all information about the product by having a look at this web application. In the future drone delivery can also be included.

3.3 Overall Description

The system will include features such as user authentication, product management, expiry date notifications, discount management, electronic bill monitoring, and sales analytics, providing stakeholders with real-time insights into product performance and sales trends. By prioritizing non-functional requirements such as performance, security, usability, and reliability, the system will offer a secure, scalable, and user-friendly solution tailored to the needs of retail environments. With a focus on minimizing waste, optimizing inventory turnover, and enhancing customer satisfaction, this project seeks to empower retail businesses to thrive in an increasingly competitive market landscape.

3.3.1 Product Perspective

From a product perspective, the Product Management System for Retail Stores serves as a crucial tool to address the specific needs and challenges faced by retail businesses in managing their inventory, sales, and operational tasks. The system provides a centralized platform for store personnel to efficiently manage product information, monitor stock levels, and make informed decisions regarding pricing, promotions, and inventory optimization. With features such as expiry date notifications and dynamic discount management, the system enables proactive measures to minimize waste and maximize profitability. Additionally, the integration of sales analytics capabilities empowers stakeholders to gain valuable insights into product performance, customer preferences, and market trends, facilitating data-driven decision-making and strategic planning. By offering a comprehensive suite of functionalities and prioritizing user experience, security,

and scalability, the system aims to enhance operational efficiency, drive business growth, and ultimately contribute to the success and sustainability of retail businesses.

3.3.2 Product Functionality

Through this system admin can include various product data.User can getoverall control of the web application

3.3.3 Users and Characteristics

There are two types of users that interact with the site admin and employee. Both of them has the priviledge to login and register into web application.Both ofthe usercan add product.But only admin have the priviledge to access into the database

3.4 Specific Requirements

3.4.1 Hardware Requirements

- System: IBM-Compatible PC
- Processor: Pentium IV or above
- Speed: Above 1GHz
- RAM capacity: 512 MB
- Hard Dsk drive: 40 GB
- Keyboard: Standard
- Mouse: Standard
- Monitor: SVGA Color

3.4.2 Software Requirements

- Operating System:Windows or ubuntu
- Languages used: Python Django
- Database : MySql
- Technologies used: HTML,Javascript,CSS,Bootstrap

3.5 Functional Requirements

It contains three main modules.

- 1.Admin
- 2.User or Employee

Admin

An Admin account is used for editing or managing the web application dynamically by Admin panel. The admin can add new products, and have the access to database.

User

The user or The employee can register ar login the web apllication. Also the user can add product into the database and for the current status of the product. but they did not have the right to access into the database.

3.6 Non Functional Requirements

Non-functional requirements define the overall qualities or attributes of the resulting system. Non-functional system place restrictions on the product being developed, the developed process, and specify external constraints that the product must meet. Examples of non-functional requirements include safety, security, usability, reliability and performance requirements. Project management issues (costs, time and schedule) are often considered as non-functional requirements. The principal non - functional constraints which are relevant to critical systems :

- performance
- security
- safety
- usability

Performance

Performance requirements concern the speed of operation of a system.Type of performance requirements :

- Response requirements (how quickly the system reacts to a user input).
- Throughput requirements (how much the system can accomplish within a specified amount of time).
- Availability requirements (is the system available for service when requested by end users). The speed of operation of this system is adequate for the requirements.

Reliability

- Reliability is the ability of a system to perform its required function under stated conditions for a specified period of time.

- constraints on the runtime behavior of the system. This system is reliable because its functionalities can be done on the required conditions.

Safety

Safety requirements are not required which exclude unsafe situation from the possible solution of the system.

Usability

Usability is the ease with which a user can learn to operate, prepare inputs for, and interpret outputs of system or components. Usability requirements include :

- information error messages.
- well-formed user interfaces.

3.7 Interface Requirements

3.7.1 Hardware interfaces

The system must run over the internet, all the hardware shall require to connect internet will be hardware interface for the system. As for example modem, WAN-LAN.

3.7.2 Software interfaces

Software interface required for the working of the project is the appropriate operating system.

3.7.3 Communication interfaces

The user of the site communicate using the network connectivity and the data set is accessed

3.8 Security Requirements

- User accesses only their account.
- Validation of input is handled.
- This application containing the computer systems is physically secured against arms or surreptitious entry by intruders.
- Users must be authorized carefully to reduce changes of any such user giving access to an intruder in exchange for a bribe or other favour.

3.9 Platform Used

Windows 10 is a major version of the Microsoft Windows operating system that was released on July 29, 2015.It is built on the Windows NT kernel and follows windows 8.Part of the reason Microsoft decided to name the 2015 release "Windows 10"(and skipped "windows 9")is because the operating system is designed to be a new direction for Microsoft. One of the primary aims of windows 10 is to Unify the windows experience across multiple devices, such desktop computers, tablets, and smartphones. As part of this effort, Microsoft developed Windows10 Mobile alongside Windows 10 to replaces Windows Phone - Microsoft's previous mobile OS. Windows 10 also integrates other Microsoft services, such as Xbox Live and the Cortana voice recognition assistant.

3.10 Technologies Used

Python Django

Python Django is a powerful web framework that facilitates rapid development and clean, pragmatic design. It follows the "don't repeat yourself" (DRY) principle, enabling developers to build complex web applications with efficiency and simplicity. Django's batteries-included philosophy means it comes with many built-in features and functionalities, including an ORM (Object-Relational Mapping) for database interactions, a robust authentication system, and a powerful templating engine. Its versatility allows developers to create various types of web applications, from content management systems to e-commerce platforms. Django's scalability and security features make it a popular choice among developers for building secure and scalable web applications. Additionally, Django's active community and extensive documentation provide ample support for developers at all levels, making it an excellent framework for both beginners and experienced developers alike.

MySQL

MySQL, pronounced either "My S-Q-L" or "My Squel, " is an open source relational database management system. It is based on the structure query language (SQL), which is used for adding, removing and modifying information in the database. Standard SQL commands, such as ADD, DROP, INSERT,and UPDATE can be used with MySQL.

What distinguishes Python Django from something like client - side Javascript is that the code is executed on the server, generating HTML which is then sent to the client. The client would receive the results of running that script, but would not know what the underlying code was. You can even configure your web server to process all your HTML files with Python Django, and then there's really no way that users can tell what you have up your sleeve. The best things in using Python Django are that it is extremely simple for a newcomer, but offers many advanced features for a professional programmer.

Chapter 4

4 Design Document

4.1 Purpose

The Product Management System for Retail Stores web application serves as a comprehensive solution designed to optimize inventory management and enhance operational efficiency in retail settings. Its primary objectives include streamlining inventory management tasks by providing a centralized platform for adding, editing, and deleting products, as well as sending timely notifications for expiring products to prevent wastage. Additionally, the application facilitates promotion management, enabling users to offer discounts or special deals on expiring products to drive sales. It enhances the customer experience by offering features such as product search functionality and electric bill generation, ensuring a smooth transaction process. Furthermore, the application provides valuable insights through its dashboard analytics, empowering store owners to make informed decisions and optimize business strategies based on data-driven insights.

4.2 Scope

The scope of the Product Management System for Retail Stores web application encompasses a range of functionalities aimed at optimizing inventory management and enhancing retail operations. It includes features such as efficient inventory management through adding, editing, and deleting products, automated expiry notifications to prevent wastage, promotion management for expiring products to drive sales, product search functionality for quick access to information, electric bill generation for seamless transactions, and dashboard analytics providing valuable insights into store performance. By incorporating these features, the application aims to streamline processes, improve efficiency, and empower store owners with data-driven decision-making capabilities in their retail endeavors.

4.3 Overview

The purpose of this document is to help the reader to visualize the solution to the project presented.This document verifies how the design meet the requirement stimulated in the SRS document through design viewpoints.The design viewpoints will cover all design elements presented before. This document will provide a direct approach to the development of this project hence reducing feature creep and ponitedly determine the quality of the design.

4.4 Data Design

Database are the storehouses of data used in the software systems.The data is stored in tables inside the database.Several tables are created for the manipulation of the data for the system.Two essential settings for a database are

- Primary Key- The field that is unique for all the record occurrence.
- Foreign Key- The field used to set relation between tables.

Normalization

- Normalization is the process of organizing the data in the database.
- Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate the undesirable characteristics like Insertion, Update and Deletion Anomalies.
- Normalization divides the larger table into the smaller table and links them using relationship.
- The normal form is used to reduce redundancy from the database table.

Tables

Products

Name	DataType	Constraints	Description
id	int(11)	Primarykey	ID of product
name	varchar(25)	Notnull	Name of product
category	varchar(100)	Notnull	category of product
quantity	int(11)	Notnull	quantity of product
price	int(20)	Notnull	Prices of product
expiry_date	date	Notnull	expiry date of product
vendor	varchar(250)	Notnull	Name of vendor

Sales

Name	DataType	Constraints	Description
id	int(11)	Primarykey	ID of product
item	varchar(50)	Notnull	Name of item
price_item	int(11)	Notnull	Price of item
total_price	int(11)	Notnull	Total price of item
payment method	varchar(200)	Notnull	Payment method for product
customer name	varchar(20)	Notnull	Name of the customer
date	date	Notnull	Date of product purchase

Bill

Name	DataType	Constraints	Description
item	varchar(25)	Notnull	Item name of the product
quantity	int(9)	Notnull	Quantity of product
price	int(9)	Notnull	Price of the product
payment_method	varchar(25)	Notnull	payment method for the product

Staff

Name	DataType	Constraints	Description
id	int(11)	Primarykey	id of staff
user name	varchar(25)	Notnull	Name of staff
phone number	int(10)	Notnull	Phone number of the staff
status	varchar(30)	Notnull	Status of the staff
role	varchar(10)	Notnull	Role of the staff

Expiry date

Name	DataType	Constraints	Description
id	int(9)	Primarykey	ID of the product
name	varchar(25)	Foreignkey	Name from Product table
category	varchar(30)	Foreignkey	Category from Product table
quantity	int(11)	Foreignkey	Quantity from Product table
price	int(11)	Foreignkey	Price from Product table
expiry date	date	Foriegnkey	Expiry date from Product table
vendor	varchar(10)	Foriegnkey	Vendor from Product table

Chapter 5

5 Development of the System

The development of the Product Management System for Retail Stores web application involves utilizing a combination of front-end technologies like HTML, CSS, and JavaScript alongside a back-end framework such as Django or Flask, supported by a suitable database management system. Following an agile methodology, the development process encompasses stages of requirement gathering, system design, implementation, testing, deployment, and ongoing maintenance. Collaboration and communication among team members, stakeholders, and end-users are prioritized throughout, facilitated by project management tools and regular meetings. Security considerations are paramount, with the implementation of best practices for secure coding, data encryption, and access control to safeguard sensitive information. Ultimately, the development aims to deliver a robust, user-friendly solution that optimizes inventory management, enhances operational efficiency, and ensures data security in retail environments.

Chapter 6

6 System Testing

Testing is the penultimate step of software development.An elaborate testing of the data is prepared and the system is using the test data,while doing testing,errors are noted and correction is made.The users are trained to operate the developed system.Both hardware and software securities are made to run the developed system successfully.System testing is aimed at ensuring the system works accurately before the live operation commences.

6.1 Test Plan

6.1.1 Scope

This test plan will cover the following testing activities as identified in the testing strategy.

- White Box Testing

White box testing is a testing technique, that examines the program structure and derives test data from the program logic/code. In white box testing, the UI is bypassed. Inputs and outputs of ratings are tested directly at the code level and the results are compared against specifications. It Reveals errors in "hidden" code.

- Black Box Testing

Black - box testing is a method of software testing that examines the functionality of an application based on the specifications. Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end - user would.

- Unit Testing

The module includes four main components to undergo unit testing. It checks the syntax error, logic error and validity of the program correctness. This test will be performed by the developers.

- Integration Testing

After all components pass the unit testing, they get to be tested whether they work correctly when they are running concurrently and communicating to each other. Integration tests exercise an entire subsystem and ensure that a set of components play nicely together.

- Internal Data Testing

We will test the validity of the data before it enters the database to avoid any problems that may face us in the database. We will test the encryption of the personal information of all the users along with the user names and passwords to ensure maximum security of the users privacy.

6.1.2 Software risk issues

In this section, the plan is to test the risk involved in critical issues such as: Difficult to run on visual studio due to large loading time. Some inherent software risks such as complexity exist; also these issues need to be identified. Proper network connection Working of Mysql database

6.1.3 Features to be tested

- Test whether correct user name and password allows you to login.
- Test whether invalid user name and password prevents you from login.
- Test whether there is any connection problem in the Server.
- Test whether the student and admin details are entered correctly.
- Test whether invalid data entry allows saving data successfully.
- Test whether all pages are loaded correctly.
- Test whether watermark is embedded and extracted properly.
- Test whether the provided security works correctly.

6.2 Test consolidation

6.2.1 Test item

The items or features to be tested in the test cases are included in the document. Each and every user input is tested. The present condition of the system is tested. It is checked to make sure that environment is ready for the application to work.

6.2.2 Input specifications

These are the inputs required to execute the test case. All required inputs including data elements and values are given below.

Data	Values
Email	A valid email ID
Password	Combination of characters and numbers
Phone number	10 Digit number

Chapter 7

7 System Implementation and Maintenance

7.1 Implementation

System implementation is the conversion of new system into an operating one which involves creating compatible files, training clients and installing hardware. User training is crucial for minimizing resistance to change and giving chance to prove its worth. Training aids user friendly manuals and healthy screens provide the user with a good start. Software maintenance follows conversion to the extent that changes are necessary to maintain satisfactory operations relative to changes in the user's environment. Maintenance often includes minor enhancements or corrections to the problem that surface late in the systems operations. In the implementation phase, the team builds the components either from scratch or by composition. Given the architecture document meant from the design phase and the requirement document from the analysis phase, the team should build exactly what has been requested, though there is still room for innovation and flexibility. For example, a component may be narrowly designed for this particular system, or the component may be made more general to satisfy a reusability

- Careful planning
- Investigation of system and constraints
- Design the methods to achieve changeover.
- Training the staff in the changed phase.
- Evaluation of change over method.
- The method of implementation and time scale to be adopted are found out initially.

7.2 Maintenance

This phase occurs as a result of deploying the whole system at the end users organization. They will perform the beta testing at the end users and inform to the developers about any needed modification to the application. The customer records all the problems that are encountered during the beta testing and reports these to the developer at regular intervals.

7.2.1 Corrective Maintenance

Even with the best quality assurance activities, it is likely that the customer will uncover defects in the software. Corrective maintenance changes the software to correct the defects. Corrective Maintenance activity may consist of repair,

restoration or replacement of equipment. This activity will be result of a regular inspection, which identifies the failure in time for corrective maintenance to be planned and scheduled, then performed during a routine maintenance shutdown.

7.2.2 Adaptive Maintenance

Over time, the original environment(CPU, operating system, business rules, external product characteristics) for which the software was developed is likely to change. Adaptive maintenance results in modification to the software to accommodate changes to its external environment.

7.2.3 Enhanced Maintenance

As software is used, the customer/user will recognize additional functions that will provide the benefit. Perfect maintenance extends the software beyond its original functional requirements.

7.2.4 Preventive Maintenance

Computer software deteriorates due to change, and because of this preventive maintenance often called software re-engineering, must be conducted to enable the software to serve the needs of its end users. Preventive maintenance makes changes to computer programs so that they can be more easily corrected, adapted and enhanced.

Chapter 8

8 Conclusion and Future Scope

8.1 Conclusion

In conclusion, the Product Management System for Retail Stores stands as a transformative solution poised to redefine how retail businesses operate in the modern landscape. By combining intuitive design with powerful features such as inventory management, expiring product notifications, promotion management, and insightful analytics, the system empowers store owners and employees to streamline operations, minimize wastage, and drive profitability. With future enhancements focused on integration with supply chain management, mobile accessibility, and advanced analytics, the system is well-positioned to adapt and evolve alongside the changing needs of the retail industry. Ultimately, the proposed system promises to revolutionize the retail experience, fostering increased efficiency, agility, and customer satisfaction in an ever-evolving marketplace.

8.2 Future Scope

- Drone delivery
- Expansion to Multi-location Support
- Implementation of Customer Relationship Management (CRM) Features
- Integration with IoT Devices
- Integration with Supply Chain Management

Appendix

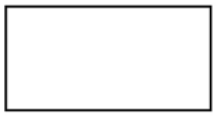
A Data Flow Diagrams

Data flow is the one of the best way of documenting the entire functionality of the system.For the system ,which will have data flows in and have some processing inside and then some data flow out from the system can be documented or represented effectively by means of data flow out from the system can be documented or represented effectively by means of data flow diagrams. The data flow diagram are a diagrammatic representation of the system,which has input,process and output.Once any system is represented using a data flow diagram we can identify the following things easily:

- Various entities interacting with the system are identified
- Flow of data from one entity to another is identified
- The various processes involved in between the interaction of two or more entities in the system are clearly pointed out
- The various data stores which hold the data in between the process,are clearly identified

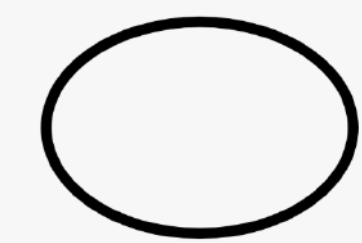
Some Data Flow Diagram charting forms:

A.1 External source or receiver



A source or sink is a person or part of organization ,which enters or receives information from the system,but is considered to be outlining the contest of data flow model.

A.2 Transform process

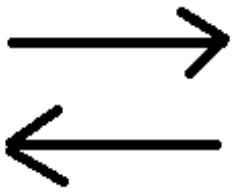


A process represents transformation where incoming data flowsare changed into outgoing data flow

A.3 Data Store

A data store is repository of data that is to be stored for use by one or more process may be as simple as buffer or queue or sophisticated as relational database. They should have clear names.If a process merely uses the contest of store and does not alter it,the arrowhead goes only form the store to the process. If a process alters the details in the store then double-headed arrow is used.

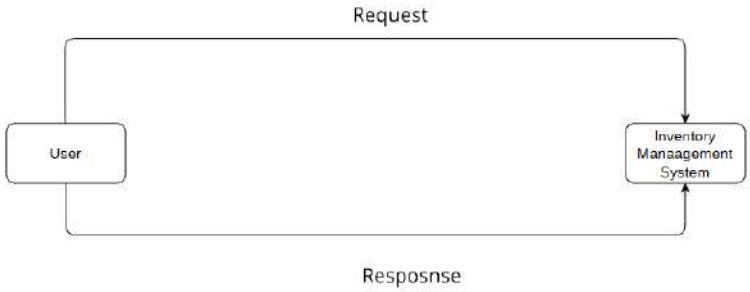
A.4 Data flow



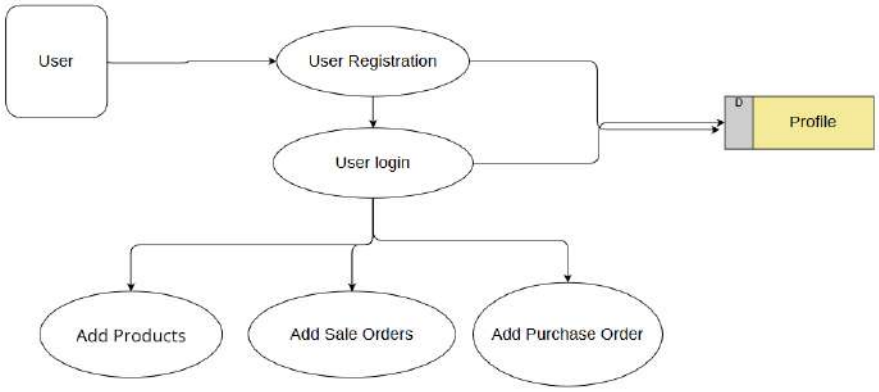
A data flow is a route, which enable packets of data to travel from one point to another.Data may flow,with arrowhead pointing in the direction of the flow

B Data Flow Diagrams

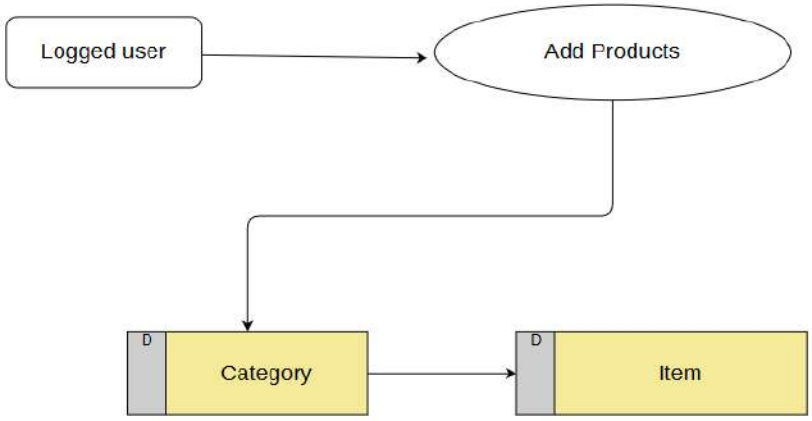
B.1 Level 0



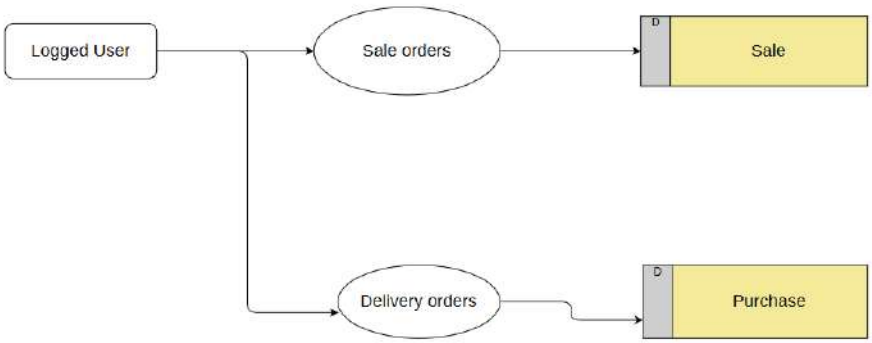
B.2 Level 1


User
User Registration
User login
D
Profile
Add Products
Add Sale Orders
Add Purchase Order

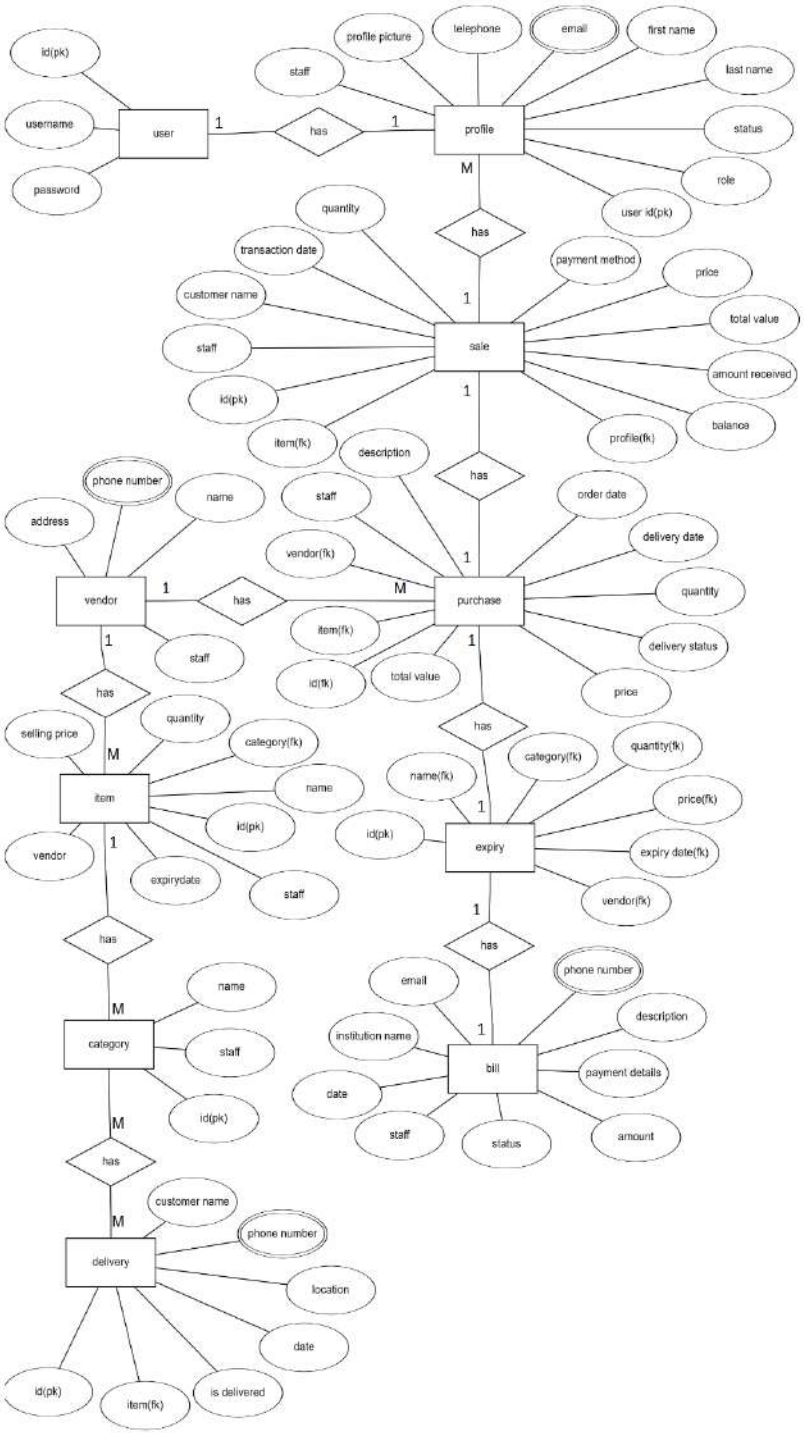
B.3 Level 2



B.4 Level 3

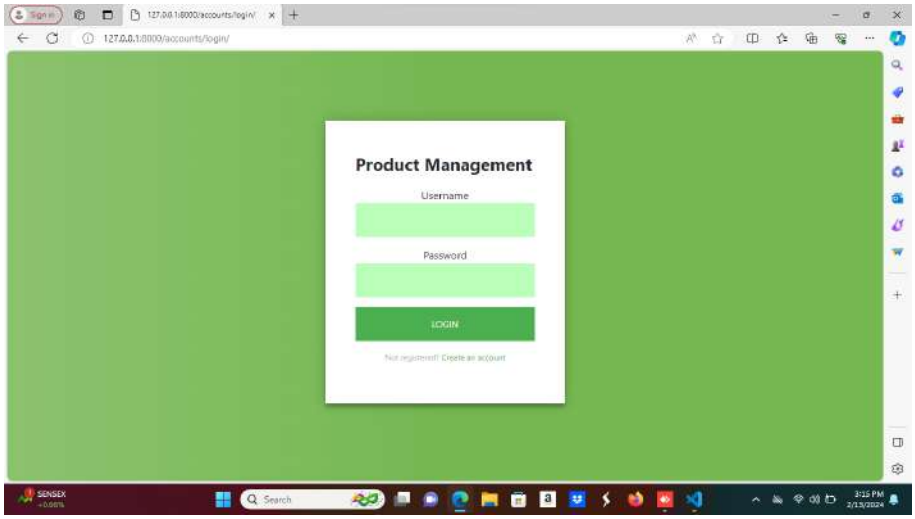


ER Diagram

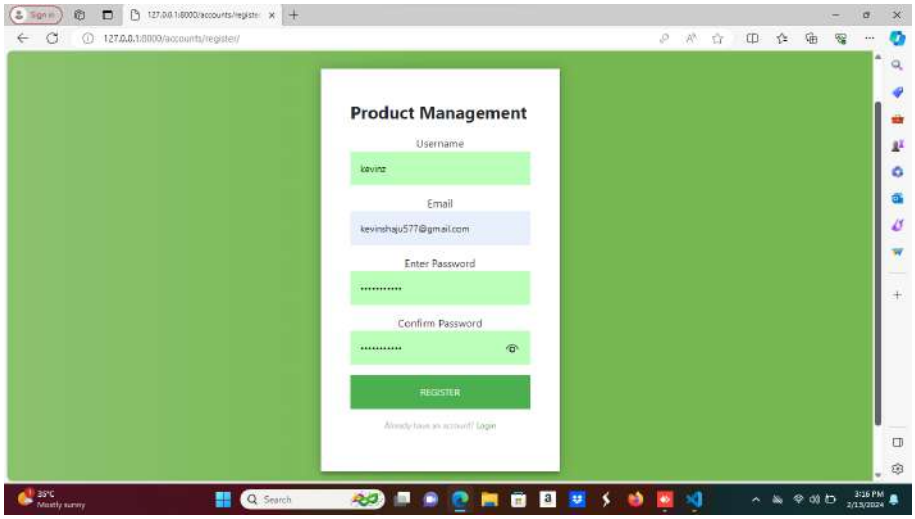

id(pk)
username
password
user
1
has
1
staff
profile picture
telephone
email
first name
last name
profile
status
role
user id(pk)
M
has
quantity
transaction date
customer name
staff
id(pk)
item(fk)
description
sale
payment method
price
total value
amount received
balance
profile(fk)
1
has
phone number
name
address
vendor
1
has
M
staff
vendor(fk)
order date
delivery date
quantity
delivery status
price
purchase
item(fk)
id(fk)
total value
1
has
selling price
quantity
category(fk)
name
id(pk)
item
M
1
vendor
expirydate
staff
has
name(fk)
category(fk)
quantity(fk)
price(fk)
expiry date(fk)
vendor(fk)
id(pk)
expiry
1
has
name
staff
id(pk)
category
M
has
email
phone number
description
institution name
payment details
date
staff
status
amount
bill
customer name
phone number
location
date
delivery
id(pk)
item(fk)
is delivered

C USER INTERFACES

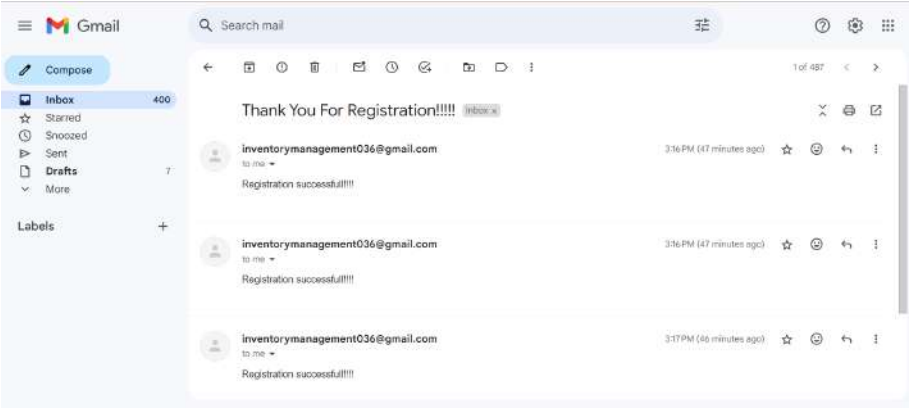
C.1 LOGIN



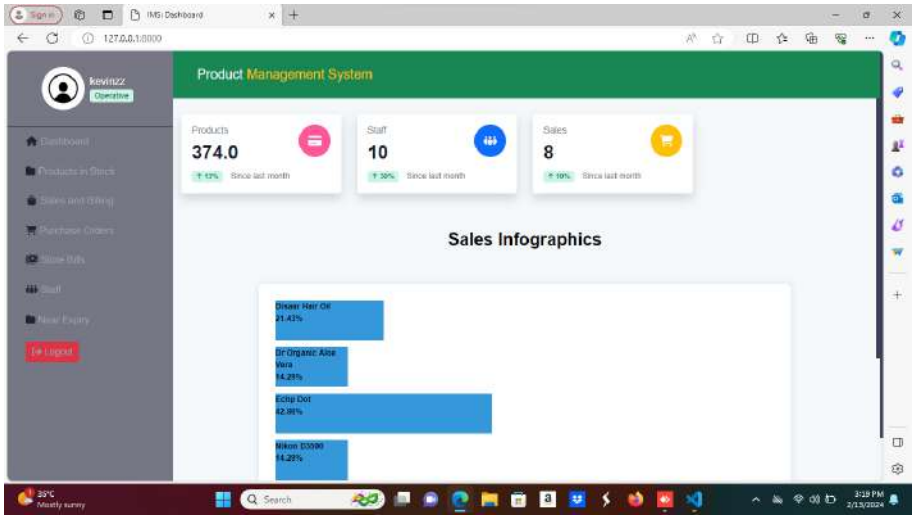
C.2 REGISTER


Product Management
Username
kevinz
Email
kevinshaju577@gmail.com
Enter Password
Confirm Password
REGISTER

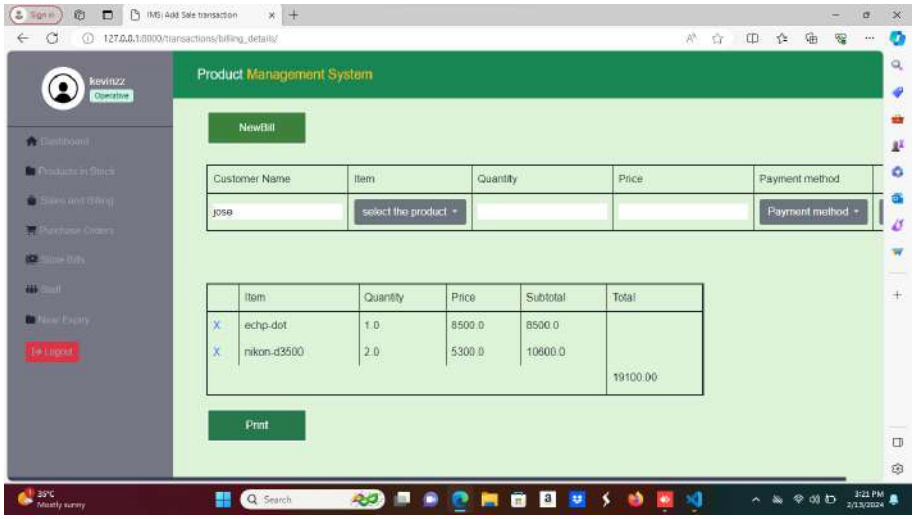
C.3 MAIL


Gmail
Search mail
Compose
Inbox 400
Starred
Snoozed
Sent
Drafts 7
More
Labels
Thank You For Registration!!!!!
inventorymanagement036@gmail.com
Registration successful!!!!
inventorymanagement036@gmail.com
Registration successful!!!!
inventorymanagement036@gmail.com
Registration successful!!!!

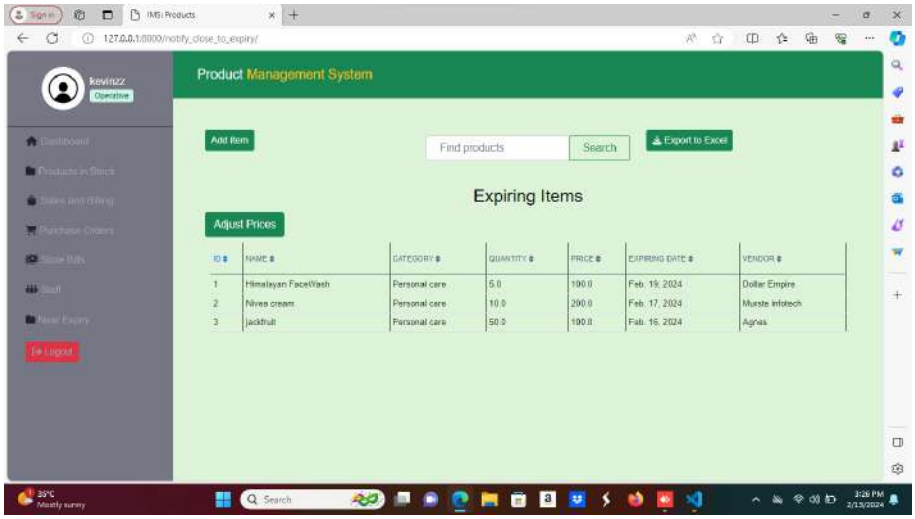
C.4 HOME


Product Management System
Products
374.0
Staff
10
Sales
8
Sales Infographics

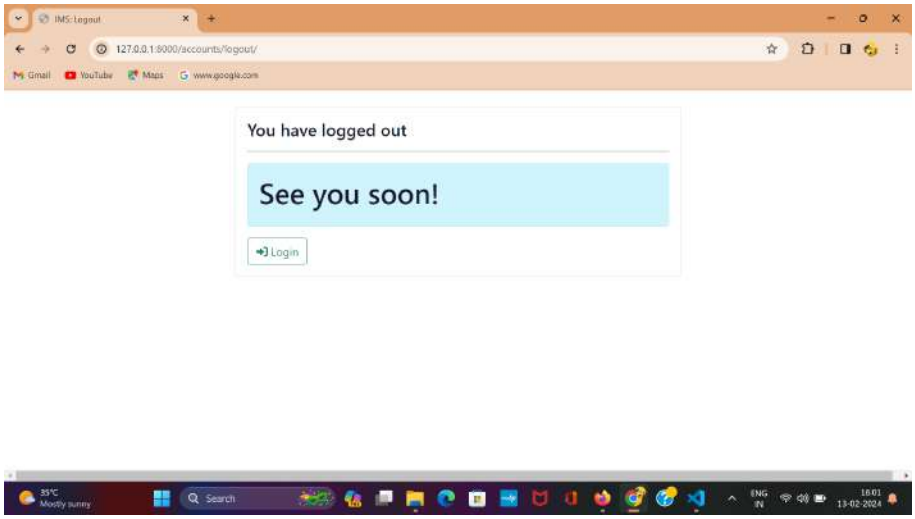
C.5 BILL


Product Management System
NewBill
Customer Name
Item
Quantity
Price
Payment method
jose
select the product
Payment method
Item
Quantity
Price
Subtotal
Total
echp-dot
1.0
8500.0
8500.0
nikon-d3500
2.0
5300.0
10600.0
19100.00
Print

C.6 EXPIRY


Product Management System
Add Item
Find products
Search
Export to Excel
Expiring Items
Adjust Prices
ID
NAME
CATEGORY
QUANTITY
PRICE
EXPIRING DATE
VENDOR
1 Himalayan FaceWash Personal care 5.0 100.0 Feb. 19, 2024 Dollar Empire
2 Nivea cream Personal care 10.0 200.0 Feb. 17, 2024 Murste Infotech
3 jackfruit Personal care 50.0 100.0 Feb. 16, 2024 Agnes

C.7 LOGOUT


IMS: Logout
127.0.0.1:8000/accounts/logout/
Gmail
YouTube
Maps
www.google.com
You have logged out
See you soon!
Login
35°C
Mostly sunny
Search
ENG
IN
16:01
13-02-2024

D CODE

login.html

```
[breaklines=true]
{% load crispy_forms_tags %}{% load static %}
<html>
  <head>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist
/css/bootstrap.min.css"
      rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3Q
DgpJLIm9Nao0Yz1ztcQTwFspd3yD
       65VohhpuuCOmLASjC"
      crossorigin="anonymous" />
  </head>
<body>
  <div class="login-page">
    <style>
    .login-page {
      width: 360px;
      padding: 8% 0 0;
      margin: auto;
    }
    .form {
      position: relative;
      z-index: 1;
      background: #FFFFFF;
      max-width: 360px;
      margin: 0 auto 100px;
      padding: 45px;
      text-align: center;
      box-shadow: 0 0 20px 0 rgba(0, 0, 0, 0.2), 0 5px 5px 0
 rgba(0, 0, 0, 0.24);
    }
    .form input {
      outline: 0;
      background: #bbffbb;
      width: 100%;
      border: 0;
      margin: 0 0 15px;
      padding: 15px;
      box-sizing: border-box;
      font-size: 14px;
    }
    .form button {
      text-transform: uppercase;
```

```
  outline: 0;
  background: #4CAF50;
  width: 100%;
  border: 0;
  padding: 15px;
  color: #FFFFFF;
  font-size: 14px;
  -webkit-transition: all 0.3 ease;
  transition: all 0.3 ease;
  cursor: pointer;
}
.form button:hover,.form button:active,.form button:focus {
  background: #43A047;
}
.form .message {
  margin: 15px 0 0;
  color: #b3b3b3;
  font-size: 12px;
}
.form .message a {
  color: #4CAF50;
  text-decoration: none;
}
.form .register-form {
  display: none;
}
.container {
  position: relative;
  z-index: 1;
  max-width: 300px;
  margin: 0 auto;
}
.container:before, .container:after {
  content: "";
  display: block;
  clear: both;
}
.container .info {
  margin: 50px auto;
  text-align: center;
}
.container .info h1 {
  margin: 0 0 15px;
  padding: 0;
  font-size: 36px;
  font-weight: 300;
```

```
      color: #1a1a1a;
    }
    .container .info span {
      color: #4d4d4d;
      font-size: 12px;
    }
    .container .info span a {
      color: #000000;
      text-decoration: none;
    }
    .container .info span .fa {
      color: #EF3B3A;
    }
    body {
      background: #76b852; /* fallback for old browsers */
      background: rgb(141,194,111);
      background: linear-gradient(90deg, rgba(141,194,111,1) 0%,
 rgba(118,184,82,1) 50%);
      -webkit-font-smoothing: antialiased;
      -moz-osx-font-smoothing: grayscale;
    }
    </style>
    <div class="form">
      <div>
        <strong class="navbar-brand" style="font-size: 26px;">
Product<span class=
             "text-warning">ms</span></strong>
      </div>
      <div class="text-danger">
        {% if messages %} {% for message in messages %}
        <div class="alert alert-{{message.tags}}">
            {{ message }}
        </div>
        {% endfor %} {% endif %}

    {% if form.non_field_errors %}
        <ul>
            {% for error in form.non_field_errors %}
                <li>{{ error }}</li>
            {% endfor %}
        </ul>
    {% endif %}
  </div>
      <form class="login-form" method="POST">
        {% csrf_token %}
        <p>{{ message }}</p>
```

```
        <label>Username</label>
        <span class="text-danger">{{ form.username.errors }}
</span>
        {{form.username}}
        <label>Password</label>
        <span class="text-danger">{{ form.password.errors }
}</span>
        {{form.password}}
        <button  type="submit" name="button">login
</button>
        <p class="message">Not registered? <a href=
            "{% url 'user-register' %}">Create an account
</a></p>
      </form>
    </div>
  </div>
</body>
</html>
```

register.html

```
{% load crispy_forms_tags %}{% load static %}
<html>
  <head>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@
5.0.2/dist/css/bootstrap.min.css"
      rel="stylesheet" integrity="sha384-EVSTQN3/azpr
G1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD
                    65VohhpuuCOmLASjC"
      crossorigin="anonymous" />
  </head>
<body>
  <div class="login-page">
    <style>
    .login-page {
      width: 360px;
      padding: 8% 0 0;
      margin: auto;
    }
    .form {
      position: relative;
      z-index: 1;
      background: #FFFFFF;
      max-width: 360px;
```

```
      margin: 0 auto 100px;
      padding: 45px;
      text-align: center;
      box-shadow: 0 0 20px 0 rgba(0, 0, 0, 0.2), 0
5px 5px 0 rgba(0, 0, 0, 0.24);
    }
    .form input {
      outline: 0;
      background: #bbffbb;
      width: 100%;
      border: 0;
      margin: 0 0 15px;
      padding: 15px;
      box-sizing: border-box;
      font-size: 14px;
    }
    .form button {
      text-transform: uppercase;
      outline: 0;
      background: #4CAF50;
      width: 100%;
      border: 0;
      padding: 15px;
      color: #FFFFFF;
      font-size: 14px;
      -webkit-transition: all 0.3 ease;
      transition: all 0.3 ease;
      cursor: pointer;
    }
    .form button:hover,.form button:active,
.form button:focus {
      background: #43A047;
    }
    .form .message {
      margin: 15px 0 0;
      color: #b3b3b3;
      font-size: 12px;
    }
    .form .message a {
      color: #4CAF50;
      text-decoration: none;
    }
    .form .register-form {
      display: none;
    }
    .container {
```

```
      position: relative;
      z-index: 1;
      max-width: 300px;
      margin: 0 auto;
    }
    .container:before, .container:after {
      content: "";
      display: block;
      clear: both;
    }
    .container .info {
      margin: 50px auto;
      text-align: center;
    }
    .container .info h1 {
      margin: 0 0 15px;
      padding: 0;
      font-size: 36px;
      font-weight: 300;
      color: #1a1a1a;
    }
    .container .info span {
      color: #4d4d4d;
      font-size: 12px;
    }
    .container .info span a {
      color: #000000;
      text-decoration: none;
    }
    .container .info span .fa {
      color: #EF3B3A;
    }
    body {
      background: #76b852; /* fallback for old browsers */
      background: rgb(141,194,111);
      background: linear-gradient(90deg,
rgba(141,194,111,1) 0%, rgba(118,184,82,1) 50%);
      -webkit-font-smoothing: antialiased;
      -moz-osx-font-smoothing: grayscale;
    }
    </style>
    <div class="form">
      <div>
        <strong class="navbar-brand" style=
"font-size: 26px;">Inventory<span class=
       "text-warning">ms</span></strong>
```

```
      </div>
      <div class="text-danger">
        {% if messages %} {% for message in messages %}
        <div class="alert alert-{{message.tags}}">
            {{ message }}
        </div>
        {% endfor %} {% endif %}

    {% if form.non_field_errors %}
        <ul>
            {% for error in form.non_field_errors %}
                <li>{{ error }}</li>
            {% endfor %}
        </ul>
    {% endif %}
  </div>
      <form class="login-form" method="POST">
        {% csrf_token %}
        <p>{{ message }}</p>
        <label>Username</label>
        <span class="text-danger">
{{ form.username.errors }}</span>
        {{form.username}}
        <label>Email</label>
        <span class="text-danger">{{ form.email.errors }}
</span>
        {{form.email}}
        <label>Enter Password</label>
        <span class="text-danger">{{ form.password.errors }}
</span>
        {{form.password1}}
        <label>Confirm Password</label>
        <span class="text-danger">{{ form.password2.errors }}
</span>
        {{form.password2}}
        <button  type="submit" name="button">Register
</button>
     <p class="message">Already have an account? <a href=
                            "{% url 'user-login' %}">Login</a></p>
      </form>
    </div>
  </div>
</body>
</html>
```

style.css

```
form input, select {
 outline: 0;
 background: #bbffbb;
 width: 100%;
 border: 0;
 margin: 0 0 15px;
 padding: 15px;
 box-sizing: border-box;
 font-size: 14px;
  }
#body-row {
 margin-left: 0;
 margin-right: 0;
}

#sidebar-container {
 min-height: 100vh;
 background-color: #4CAF50;
 padding: 0;
 /* flex: unset; */
}

.sidebar-expanded {
 width: 230px;
}

.sidebar-collapsed {
 /*width: 60px;*/
 width: 100px;
}

/* ----------| Menu item*/
#sidebar-container .list-group a {
 height: 50px;
 color: white;
}

/* ----------| Submenu item*/
#sidebar-container .list-group li.list-group-item {
 background-color: #4CAF50;
}

#sidebar-container .list-group .sidebar-submenu a {
 height: 45px;
 padding-left: 30px;
}
```

```
/* ----------| Separators */
.sidebar-separator-title {
 background-color: #4CAF50;
 height: 35px;
}

.sidebar-separator {
 background-color: #4CAF50;
 height: 25px;
}

.logo-separator {
 background-color: #4CAF50;
 height: 60px;
}

a.bg-dark {
 background-color: #4CAF50 !important;
}

@import 'https://fonts.googleapis.com/css2?
family=Inter:wght@300;400;500;600&display=
                swap" rel="stylesheet';

:root {
--dk-gray-100: #F3F4F6;
--dk-gray-200: #E5E7EB;
--dk-gray-300: #D1D5DB;
--dk-gray-400: #9CA3AF;
--dk-gray-500: #6B7280;
--dk-gray-600: #4B5563;
--dk-gray-700: #374151;
--dk-gray-800: #1F2937;
--dk-gray-900: #111827;
--dk-dark-bg: #313348;
--dk-darker-bg: #2a2b3d;
--navbar-bg-color: #6f6486;
--sidebar-bg-color: #252636;
--sidebar-width: 250px;
}

* {
margin: 0;
padding: 0;
box-sizing: border-box;
}
```

```
body {
font-family: 'Inter', sans-serif;
background-color: #def4d9;
}

a {
text-decoration: none;
link-style: none;
}
#wrapper {
margin-left: var(--sidebar-width);
transition: all .3s ease-in-out;
}

#wrapper.fullwidth {
margin-left: 0;
}



/** ---------------------------------
-- Sidebar
--------------------------------- */
.sidebar {
font-size: .925rem;
background-color: var(--sidebar-bg-color);
width: var(--sidebar-width);
transition: all .3s ease-in-out;
transform: translateX(0);
z-index: 9999999
}

.sidebar .close-aside {
position: absolute;
top: 7px;
right: 7px;
cursor: pointer;
color: #EEE;
}

.sidebar .sidebar-header {
border-bottom: 1px solid #2a2b3c
}

.sidebar .sidebar-header h5 a {
```

```
color: var(--dk-gray-300)
}

.sidebar .sidebar-header p {
color: var(--dk-gray-400);
font-size: .825rem;
}

.sidebar .search .form-control ~ i {
color: #2b2f3a;
right: 40px;
top: 22px;
}

.sidebar > ul > li {
padding: .7rem 1.75rem;
}

.sidebar ul > li > a {
color: var(--dk-gray-400);
text-decoration: none;
}

/* Start numbers */
.sidebar ul > li > a > .num {
line-height: 0;
border-radius: 3px;
font-size: 14px;
padding: 0px 5px
}

.sidebar ul > li > i {
font-size: 18px;
margin-right: .7rem;
color: var(--dk-gray-500);
}

.sidebar ul > li.has-dropdown > a:after {
content: '\eb3a';
font-family: unicons-line;
font-size: 1rem;
line-height: 1.8;
float: right;
color: var(--dk-gray-500);
transition: all .3s ease-in-out;
}
```

```
.sidebar ul .opened > a:after {
transform: rotate(-90deg);
}


.show-sidebar {
transform: translateX(-270px);
}

@media (max-width: 767px) {
.sidebar ul > li {
 padding-top: 12px;
 padding-bottom: 12px;
}

.sidebar .search {
 padding: 10px 0 10px 30px
}
}




/** ---------------------------------
-- welcome
--------------------------------- */
.welcome {
color: var(--dk-gray-300);
}

.welcome .content {
background-color: var(--dk-dark-bg);
}

.welcome p {
color: var(--dk-gray-400);
}




/** ---------------------------------
-- Statistics
--------------------------------- */
.statistics {
```

```
color: var(--dk-gray-200);
}

.statistics .box {
background-color: var(--dk-dark-bg);
}

.statistics .box i {
width: 60px;
height: 60px;
line-height: 60px;
}

.statistics .box p {
color: var(--dk-gray-400);
}


/** --------------------------------
-- Please don't do that in real-world projects!
-- overwrite Bootstrap variables instead.
-------------------------------- */

.navbar {
border: none !important;
}
.navbar .dropdown-menu {
right: auto !important;
left: 0 !important;
}
.navbar .navbar-nav>li>a {
color: #EEE !important;
line-height: 55px !important;
padding: 0 10px !important;
}
.navbar .navbar-brand {color:#FFF !important}
.navbar .navbar-nav>li>a:focus,
.navbar .navbar-nav>li>a:hover {color:
 #EEE !important}

.navbar .navbar-nav>.open>a,
.navbar .navbar-nav>.open>a:focus,
.navbar .navbar-nav>.open>a:hover
{background-color: transparent !important;
color: #FFF !important}
```

```
.navbar .navbar-brand {line-height: 55px
 !important; padding: 0 !important}
.navbar .navbar-brand:focus,
.navbar .navbar-brand:hover {color:
#FFF !important}
.navbar>.container .navbar-
brand, .navbar>.container-fluid .navbar-brand
               {margin: 0 !important}
@media (max-width: 767px) {
.navbar>.container-fluid .navbar-brand {
 margin-left: 15px !important;
}
.navbar .navbar-nav>li>a {
 padding-left: 0 !important;
}
.navbar-nav {
 margin: 0 !important;
}
.navbar .navbar-collapse,
.navbar .navbar-form {
 border: none !important;
}
}

.navbar .navbar-nav>li>a {
float: left !important;
}
.navbar .navbar-nav>li>a>span:not(.caret) {
background-color: #e74c3c !important;
border-radius: 50% !important;
height: 25px !important;
width: 25px !important;
padding: 2px !important;
font-size: 11px !important;
position: relative !important;
top: -10px !important;
right: 5px !important
}
.dropdown-menu>li>a {
padding-top: 5px !important;
padding-right: 5px !important;
}
.navbar .navbar-nav>li>a>i {
font-size: 18px !important;
}
```

```
/* Start media query */

@media (max-width: 767px) {
#wrapper {
 margin: 0 !important
}
.statistics .box {
 margin-bottom: 25px !important;
}
.navbar .navbar-nav .open
     .dropdown-menu>li>a {
 color: #CCC !important
}
.navbar .navbar-nav .open .
      dropdown-menu>li>a:hover {
 color: #FFF !important
}
.navbar .navbar-toggle{
 border:none !important;
 color: #EEE !important;
 font-size: 18px !important;
}
.navbar .navbar-toggle:focus, .navbar
 .navbar-toggle:hover
     {background-color: transparent !important}
}


::-webkit-scrollbar {
background: transparent;
width: 5px;
height: 5px;
}

::-webkit-scrollbar-thumb {
background-color: #3c3f58;
}

::-webkit-scrollbar-thumb:hover {
background-color: rgba(0, 0, 0, 0.3);
}
```

```
//sorting
table {
background-color: #000 !important;
}
table.dataTable thead .sorting:after,
table.dataTable thead .sorting:before,
table.dataTable thead .sorting_asc:after,
table.dataTable thead .sorting_asc:before,
table.dataTable thead .sorting_asc_disabled:after,
table.dataTable thead .sorting_asc_disabled:before,
table.dataTable thead .sorting_desc:after,
table.dataTable thead .sorting_desc:before,
table.dataTable thead .sorting_desc_disabled:after,
table.dataTable thead .sorting_desc_disabled:before {
bottom: .5em;
}

//update profile
 /*=========================================================
            Form Section
  =========================================================*/
  form {
    margin: 0 auto;
    width: 50%;
    border-bottom-right-radius: 6px;
    border-bottom-left-radius: 6px;
    padding-bottom: 10px;
  }
  /*changing title style*/
  h1 {
    padding-top: 30px;
    text-align: center;
    color: #FF5722;
    font-size: 26px;
  }

  /*Camera image to upload image*/
  .fa-camera-retro {
      color: #a7a7a7;
      width: 100px;
      height: 100px;
      margin-top: 20px;
      line-height: 100px;
      border: 2px solid #a7a7a7;
      cursor: pointer;
      -webkit-border-radius: 50%;
```

```
     border-radius: 50%;
     box-shadow: 0 15px 20px -10px
rgba(0, 0, 0, 0.3);
 }

 .fa-camera-retro:hover{
   border-color: #6f6f6f;
   box-shadow: 0 3px 5px -5px rgba(0, 0, 0, 0.3);
   transform: translateY(-1px);
   color: #6f6f6f;
 }



 /*=========================================================
           input Section
 ==========================================================*/

 /*Float Label main style*/
 .float-label input, .float-label select {
     -webkit-appearance: none;
     outline: none;
     border: none;
     margin: 0 auto;
     width: 100%;
     display: block;
     -moz-border-radius: 0;
     border-radius: 0;
     font-family: 'Lato', sans-serif;
     font-size: 18px;
     border-bottom: 1px solid rgba(0, 0, 0, 0.7);
     background: transparent;
     color: #757575;
     padding: 5px 15px 7px 10px;
 }

 .title {
   font-family: 'Lato', sans-serif;
   font-size: 18px;
   background: transparent;
   color: #757575;
   font-weight: normal;
   margin-left: 7px;
 }

 .gender, .birthday {
```

```
    text-align: left;
    padding-left: 16px;
    margin-top: -10px;
}

.gender div, .birthday div {
  display: inline-block;
  padding-left: 0;
}

/*add style to radio options*/
.option-input {
  -webkit-appearance: none;
  -moz-appearance: none;
  -ms-appearance: none;
  -o-appearance: none;
  appearance: none;
  position: relative;
  top: 13.33333px;
  right: 0;
  bottom: 0;
  left: 0;
  height: 15px;
  width: 15px;
  transition: all 0.15s ease-out 0s;
  background: #cbd1d8;
  border: none;
  color: #fff;
  cursor: pointer;
  display: inline-block;
  margin-right: 0.5rem;
  outline: none;
  position: relative;
  z-index: 1000;
}
.option-input:hover {
  background: #9faab7;
}
.option-input:checked {
  background: #77cee2;
}
.option-input:checked::before {
  height: 15px;
  width: 15px;
  position: absolute;
  content: '';
```

```
  display: inline-block;
  font-size: 12px;
  text-align: center;
  line-height: 15px;
}
.option-input:checked::after {
  -webkit-animation: click-wave 0.65s;
  -moz-animation: click-wave 0.65s;
  animation: click-wave 0.65s;
  background: #77cee2;
  content: '';
  display: block;
  position: relative;
  z-index: 100;
}
.option-input {
  border-radius: 50%;
}
.option-input::after {
  border-radius: 50%;
}
.radio-inline input[type=radio] {
  position: static;
  margin-left: 0;
  padding-left: 0;
}
.radio-inline {
  vertical-align: baseline;
}
.radio-inline span {
  line-height: 4;
  font-size: 16px;
  padding-left: 4px;
}

 /*Style birthday select */
.float-label select {
  display: inline-block;
  width: 24%;
  margin-left: 20px;
  font-size: 16px;
  color: black;
}
.float-label select:first-child{
  width: 30%;
}
```

```
.birthday .select {
  width: 82%;
}

.float-label > select option:first-child {
  color: #77cee2;
}


#updateProfile {
  padding: 20px 60px;
}

.user-left {
  padding-top: 40px;
}

 /*create a round container to hide overflow image*/
.user-left span {
  display: inline-block;
  width: 250px;
  height: 250px;
  overflow: hidden;
  text-align: center;
  border-radius: 100%;
  /*add shadow and border*/
  border: 2px solid #a9a9a9;
  cursor: pointer;
  -webkit-border-radius: 50%;
  border-radius: 50%;
  box-shadow: 0px 20px 25px -10px rgba(0, 0, 0, 0.3);
}

/*change image effect when hover*/
.user-left span:hover{
  border-color: #b3b3b3;
  box-shadow: 0 3px 5px -5px rgba(0, 0, 0, 0.3);
  transform: translateY(-1px);
}

/*change font margin and padding*/
.user-left h1 {
  padding-top: 0;
}

/*center image and make it round*/
```

```
  .user-image {
    left: 50%;
    margin-left: -100%;
    position: relative;
    width: auto important;
    height: 250px important;
  }

  /*change image opacity*/
  .user-image:hover {
    opacity: 0.5;
    filter: alpha(opacity=50);
  }

  .user-right {
    padding: 20px 20px 40px;
    min-height: 200px;
    /*margin-left: 80px;*/
  }

  /*Change profile title*/
  .user-right h2 {
    text-align: left;
    font-family: 'Roboto', Arial, sans-serif;
    font-weight: 400;
    font-size: 3em;
  }
  .user-right span {
    color: #d2d2d2;
  }

  .user-info {
    padding: 18px;
    width: 100%;
    min-height: 200px;
    border-radius: 3px;
    box-shadow: 0 0 0 0 transparent;
    box-shadow: 0 15px 20px -15px
 rgba(0, 0, 0, 0.3), 0 55px 50px -35px
                          rgba(210, 214, 213, 0.5);

    box-shadow: 0px 2px 2px 2px
rgba(210, 214, 213, 0.4);
  }
```

```
  td p{
    font-size: 20px;
    padding: 10px 15px 10px 10px;
    text-align: left;
  }

  #signOut {
    float: right;
  }
```

logout.html

```
[verbetim]
{% load crispy_forms_tags %}
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta http-equiv="X-UA-Compatible"
content="IE=edge">
        <meta name="viewport"
 content="width=device-width, initial-scale=1.0">
        {% load static %}

        <link rel="stylesheet" href=
     "{% static 'fontawesome/css/all.css' %}"
 type="text/css">
        <link rel="stylesheet" type="text/css" href=
"https://unpkg.com/@webpixels
                                  /css@1.1.5/dist/index.css">
        <link href="https://cdn.jsdelivr.net/npm/
bootstrap@5.0.2/dist/css/
      bootstrap.min.css" rel="stylesheet" integrity=
   "sha384-EVSTQN3/azprG1Anm3QDgp
JLIm9Nao0Yz1ztcQTwFspd
       3yD65VohhpuuCOmLASjC"
 crossorigin="anonymous">
        <title>IMS: Logout</title>
    </head>
    <body>
       <div class="row my-4">
         <div class="col-md-6 offset-md-3">
            <div class="border p-3 bg-white">
             <h4>You have logged out</h4>
```

```
                <hr>
               <div class="alert alert-info">
                  <h1>See you soon!</h1>
                     </div>
          <a class="btn btn-outline-success" href=
"{% url 'user-login' %}"><i class=
"fa-solid fa-right-to-bracket"></i> Login</a>
                 </div>
            </div>
        </div>
    </body>
</html>
```

bill.html

```
{% extends "store/base.html" %}
{% load render_table from django_tables2 %}
{% load querystring from django_tables2 %}
{% load static %}{% block title %}Bills
{%endblock title%}{% block content %}
    <div class="col container p-5">
      <style>
        input[type="text"],select{
            box-sizing: border-box;
            border: 1px solid #ccc;
            border-radius: 4px;
            display: inline-block;
            padding: 6px 20px;
            margin: 8px 0;
        }
        button[type="submit"]{
          padding: 6px 20px;
          margin: 8px 0;
      }
      </style>
      {% comment %} <form method="GET"
class="container">
        {{filter.form}}
        <button type="submit" class="btn btn-sm
btn-success fa fa-search"> Search</button>
      </form> {% endcomment %}
      <div>
     <a class="btn btn-sm btn-success" href=
"{% url 'bill-create' %}">
```

```
                          Add a bill record</a>
     <a class="float-end btn btn-sm btn-success" href=
                            "{% querystring '_export'='xlsx' %}">
          <i class="fa-solid fa-download"></i>
          Export to Excel
        </a>
      </div>
      <div class="m-2">
        <table class= "table table-sm table-responsive">
            <thead>
              <tr class="table-success">
                <th scope="col"><a href=
"{% querystring table.prefixed_order_by_field
=column.order_by_alias.next %}">ID
 <i class="fa-solid fa-sort"></i></a></th>
 <th scope="col">Name <i class="fa-solid fa-sort">
</i></th>
<th scope="col">Description <i class="fa-solid fa-sort">
</i></th>
<th scope="col">Contact number <i class=
"fa-solid fa-sort"></i></th>
 <th scope="col">email<i class="fa-solid fa-sort"></i></th>
 <th scope="col">Payment details<i class="fa-solid fa-sort">
</i></th>
 <th scope="col">Amount <i class="fa-solid fa-sort">

</i></th>
 <th scope="col">Status <i class="fa-solid fa-sort"></i></th>
 <th scope="col"> Action</th>
              </tr>
            </thead>
            <tbody>
            {% for bill in bills %}
              <tr>
                <th scope="row"><a> {{bill.id}}</a></th>
                <td>{{bill.institution_name}}</td>
                <td>{{bill.description}}</td>
                <td>{{bill.phone_number}}</td>
                <td>{{bill.email}}</td>
                <td>{{bill.payment_details}}</td>
                <td>{{bill.amount}}</td>
                <td>{% if bill.status == True%}
                  <span class="badge badge-pill
bg-soft-success text-success me-2">
                    Paid
                  </span>
```

```
                  {% else %}
                  <span class="badge badge-pill bg-soft
-danger text-danger me-2">
                    Pending
                  </span>
                  {% endif %}
                </td>
                <td>
                  <a class="text-info" href=
"{% url 'bill-update' bill.slug %}"><i class=
"fa-solid fa-pen"></i></a>
                  <a class="text-danger float-end" href=
"{% url 'bill-delete' bill.pk %}"><i class=
"fa-solid fa-trash"></i></a>
                </td>
              </tr>
            {% endfor %}
            </tbody>
          </table>
        </div>
          <div class="paginator">
            {% if is_paginated %}
            <ul class="pagination ">
                {% if page_obj.has_previous %}
                <li><a class="btn btn-sm accent-1
 btn-outline-success
 " href="?page={{ page_obj.previous_page_number }}
 ">&laquo;</a></li>
                {% else %}
                <li class="disabled "><span class=
"btn btn-sm btn-outline-success ">
                &laquo;</span></li>
             {% endif %} {% for i in paginator.page_range %}
 {% if page_obj.number == i %}
                <li class="active "><span class=
"btn btn-sm btn-success ">{{ i }}
                    <span class="sr-only ">(current
)</span></span>
                </li>
                {% else %}
      <li><a href="?page={{ i }} " class=
"btn btn-sm btn-outline-success ">{{ i }}</a></li>
                {% endif %} {% endfor %}
 {% if page_obj.has_next %}
                <li><a href="?page={{ page_obj.next_page_number }}
            " class="btn btn-sm btn-outline-success ">&raquo;</a></li>
```

```
                {% else %}
                <li class="disabled ">
<span class="btn btn-sm btn-outline-success ">
                &raquo;</span></li>
                {% endif %}
            </ul>
            {% endif %}
        </div>
    </div>
  {% endblock content %}
```

productlist.html

```
[breaklines=true]
{% extends "store/base.html" %}
{% load render_table from django_tables2 %}
{% load querystring from django_tables2 %}
{% load static %}
{% block title %}Products{%endblock title%}
{% block content %}
    <div class="col container p-5">
      <style>
        input[type="text"],select{
            box-sizing: border-box;
            border: 1px solid #ccc;
            border-radius: 4px;
            display: inline-block;
            padding: 6px 20px;
            margin: 8px 0;
        }
        button[type="submit"]{
          padding: 6px 20px;
          margin: 8px 0;
      }
      </style>
      <div class="row">
        <div class="col-md-4">
          <a class="btn btn-sm btn-success" href=
  "{% url 'product-create' %}">Add Item</a>
        </div>
        <div class="col-md-4">
          <form class="input-group" role=
"search " id="searchform" action=
"{% url 'item_search_list_view' %}
" method="get" accept-charset="utf-8">
            <div class="form-group">
```

```
                <div class="input-group ">
                    <input id="searchbox" name=
"q" type="text" class=
      "form-control " placeholder="Find products">
                    <span class="input-group-btn">
                      <button class=
                "btn btn-outline-success" type=
"submit">Search</i></button>
                    </span>
                </div>
            </div>
        </form>
        </div>
        <div class="col-md-4 float-end">
          <a class="btn btn-sm btn-success" href=
"{% querystring '_export'='xlsx' %}">
            <i class="fa-solid fa-download"></i>
            Export to Excel
          </a>
        </div>
      </div>
      <div class="m-2">
        <table class= "table table-sm table-responsive">
            <thead>
              <tr class="table-success">
                <th scope="col"><a href=
"{% querystring table.prefixed_order_by_field
=column.order_by_alias.next %}">ID
                                <i class="fa-solid fa-sort"></i></a></th>
                <th scope="col">Name <i class=
"fa-solid fa-sort"></i></th>
                <th scope="col">Category <i class=
"fa-solid fa-sort"></i></th>
                <th scope="col">Quantity <i class=
"fa-solid fa-sort"></i></i></th>
                <th scope="col">Price <i class=
"fa-solid fa-sort"></i></th>
                <th scope="col">Expiring date <i class=
"fa-solid fa-sort"></i></th>
                <th scope="col">Vendor <i class=
"fa-solid fa-sort"></i></th>
                <th scope="col"> Action</th>
              </tr>
            </thead>
            <tbody>
            {% for item in items %}
```

```
              <tr>
                <th scope="row"><a> {{item.id}}
</a></th>
                <td>{{item.name}}</td>
                <td>{{item.category}}</td>
                <td>{{item.quantity}}</td>
                <td>{{item.selling_price}}</td>
                <th scope="col">{{item.expiring_date}}
</th>
                <td>{{item.vendor}}</td>
                <td>
                  <a class="text-info" href=
"{% url 'product-update' item.slug %}">
<i class="fa-solid fa-pen"></i></a>
                  <a class="text-danger float-end" href=
"{% url 'product-delete' item.slug %}">
<i class="fa-solid fa-trash"></i></a>
                </td>
              </tr>
            {% endfor %}
            </tbody>
          </table>
        </div>
          <div class="paginator">
            {% if is_paginated %}
            <ul class="pagination ">
                {% if page_obj.has_previous %}
                <li><a class="btn btn-sm accent-1
btn-outline-success
       " href="?page={{ page_obj.previous_page_number }
} ">&laquo;</a></li>
                {% else %}
                <li class="disabled "><span class=
"btn btn-sm btn-outline-success ">
             &laquo;</span></li>
                {% endif %} {% for i in paginator.page_range %}
                        {% if page_obj.number == i %}
                <li class="active "><span class=
"btn btn-sm btn-success ">{{ i }}
                 <span class="sr-only ">(current)
</span></span>
                </li>
                {% else %}
                <li><a href="?page={{ i }} " class=
"btn btn-sm btn-outline-success ">
                      {{ i }}</a></li>
```

```
                {% endif %} {% endfor %} {% if page_obj.has_next %}
                <li><a href="?page={{ page_obj.next_page_number }}
               " class="btn btn-sm btn-outline-success ">&raquo;</a></li>
                {% else %}
                <li class="disabled "><span class=
"btn btn-sm btn-outline-success ">
          &raquo;</span></li>
                {% endif %}
            </ul>
            {% endif %}
        </div>
    </div>
  {% endblock content %}
```

manage.py

```
#!/usr/bin/env python
"""Django's command-line utility for administrative tasks."""
import os
import sys


def main():
    """Run administrative tasks."""
    os.environ.setdefault
('DJANGO_SETTINGS_MODULE', 'InventoryMS.settings')
    try:
        from django.core.management import
 execute_from_command_line
    except ImportError as exc:
        raise ImportError(
            "Couldn't import Django. Are you sure it's installed and "
            "available on your PYTHONPATH environment
 variable? Did you "
            "forget to activate a virtual environment?"
        ) from exc
    execute_from_command_line(sys.argv)


if __name__ == '__main__':
    main()
```

module.py

```
from django.db import models
```

```
from store.models import Item
from accounts.models import Profile, Vendor
from django_extensions.db.fields import AutoSlugField

# Create your models here.

PAYMENT_CHOICES = [
    ('MP', 'MPESA'),
    ('VISA', 'VISA'),
    ('CS', 'CASH'),
    ('VM', 'VOOMA'),
    ('BK', 'BANK')
]

DELIVERY_CHOICES = [
    ('P', 'PENDING'),
    ('S', 'SUCCESSFUL')
]

class Sale(models.Model):
    slug = AutoSlugField(unique=True , populate_from=
                'customer_name')
    item = models.ForeignKey(Item, on_delete=
                    models.CASCADE, blank=True, null=True)
    customer_name =
          models.CharField(max_length=20, null=True, blank=True)
    transaction_date =
       models.DateTimeField(auto_now=True, blank=True, null=True)
    quantity = models.FloatField(default=0.00, blank=False, null=False)
    payment_method = models.CharField(choices=
        PAYMENT_CHOICES, max_length=20, blank='True', null=True)
    price = models.FloatField(default=0.00, blank=False, null=False)
    total_value = models.FloatField(blank=True, null=True)
    amount_received = models.FloatField(default=False,
          blank=False, null=False)
    balance = models.FloatField(default=False, blank=False, null=False)
    profile = models.ForeignKey(Profile, verbose_name=
        ('Served by'), on_delete=models.CASCADE)

    def save(self, *args, **kwargs):
        amt_received = self.amount_received
        price = self.price
        quantity = self.quantity
        self.total_value = price * quantity
        self.balance = amt_received - self.total_value
        super().save(*args, **kwargs)
```

```
    def __str__(self):
        return str(self.item.name)

class Purchase(models.Model):
    slug = AutoSlugField(unique=True , populate_from='vendor')
    item = models.ForeignKey(Item, on_delete=models.CASCADE)
    description = models.TextField(max_length=300
, blank=True, null=True)
    vendor = models.ForeignKey
  (Vendor, related_name='vendor', on_delete=models
.CASCADE, blank=False, null=False)
    order_date = models.DateTimeField(auto_now_add=True)
    delivery_date = models.DateTimeField
    (auto_now=False, auto_now_add=False, blank=True,
null=True, verbose_name=
                                                    ('Delivery Date'))
    quantity = models.FloatField(default=0.00, blank=False, null=False)
    delivery_status = models.CharField(choices=
DELIVERY_CHOICES, max_length=3, default='P', blank=False,
                      null=False, verbose_name=
('Delivery Status'))
    price = models.FloatField(default=0.00, blank=False,
      null=False, verbose_name=
('Price per item(Ksh)'))
    total_value = models.FloatField()

    def save(self, *args, **kwargs):
        quantity = self.quantity
        price = self.price
        self.total_value = price * quantity
        return super().save(*args, **kwargs)

    def __str__(self):
        return self.item.name

    class Meta:
        ordering = ["order_date"]
```

setting.py

```
import os
from pathlib import Path
```

```
# Build paths inside the project like this:
 BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent


# Quick-start development settings -
 unsuitable for production
# See https://docs.djangoproject.com/
en/4.1/howto/deployment/checklist/

# SECURITY WARNING: keep the secret
 key used in production secret!
SECRET_KEY = 'django-insecure
-g_n2+2bznu6e@1wel!i
(&-4tp86_7lop5395ww+i4x%9*7^old'

# SECURITY WARNING:
 don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = []


# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',

    'phonenumber_field',
    'crispy_forms',
    'imagekit',
    'django_extensions',
    'django_filters',
    'django_tables2',

    'store.apps.StoreConfig',
    'accounts.apps.AccountsConfig',
    'transactions.apps.TransactionsConfig',
    'invoice.apps.InvoiceConfig',
    'bills.apps.BillsConfig',
```

```
]

MIDDLEWARE = [
  'django.middleware.security.SecurityMiddleware',
 'django.contrib.sessions.middleware.SessionMiddleware',
 'django.middleware.common.CommonMiddleware',
'django.middleware.csrf.CsrfViewMiddleware',
 'django.contrib.auth.middleware.AuthenticationMiddleware',
'django.contrib.messages.middleware.MessageMiddleware',
'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'InventoryMS.urls'

TEMPLATES = [
    {
 'BACKEND': 'django.template.backends
.django.DjangoTemplates',
        'DIRS': [],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
       'django.template.context_processors.debug',
        'django.template.context_processors.request',
         'django.contrib.auth.context_processors.auth',
          'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]

WSGI_APPLICATION = 'InventoryMS.wsgi.application'


# Database
# https://docs.djangoproject.com/
en/4.1/ref/settings/#databases


DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    }
}
```

```
# Password validation
# https://docs.djangoproject.com/
en/4.1/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.
password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth
.password_validation.MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth
.password_validation.CommonPasswordValidator',
    },
    {
        'NAME': 'django.contrib.auth.
password_validation.NumericPasswordValidator',
    },
]


# Internationalization
# https://docs.djangoproject.com/en/4.1/topics/i18n/

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_TZ = True

LOGIN_URL = 'user-login'
LOGIN_REDIRECT_URL = 'dashboard'
LOGOUT_URL = 'logout'

# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/4.1/howto/static-files/

STATIC_URL = 'static/'
STATICFILES_DIRS = [os.path.join(BASE_DIR,'static')
```

```
]
MEDIA_ROOT = os.path.join(BASE_DIR, 'static/images')
MEDIA_URL = '/images/'

# Default primary key field type
# https://docs.djangoproject.com/
en/4.1/ref/settings/#default-auto-field

DEFAULT_AUTO_FIELD =
 'django.db.models.BigAutoField'

# CRISPY_TEMPLATE_PACK = 'bootstrap4'



EMAIL_BACKEND =
 'django.core.mail.backends.smtp.EmailBackend'
EMAIL_HOST = 'smtp.gmail.com'
EMAIL_USE_TLS = True
EMAIL_PORT = 587
EMAIL_HOST_USER =
 "product management system753@gmail.com"
EMAIL_HOST_PASSWORD = "oyhzlplyolihopys"


# FCM_DJANGO_SETTINGS = {
#     "FCM_SERVER_KEY":
"BC7LHSnv6csnz2VKw5DMASW6n
D2n_FiCsQwjBPUmlOgN1AM
-Qehh2qslANJudWf7R-P1UbL
KwYZJyxu8OiGJwWE",
# }
```

setup.sh

```
[breaklines=true]
#!/bin/bash

echo " Setting up sales-and-inventory-management"

echo " Building Docker image..."
docker build -t sales-and-inventory-management:1.0 .
```

```
if [ $? -ne 0 ]; then
  echo " Error: Docker image build failed!"
  exit 1
fi

echo " Starting Docker containers in detached mode..."
docker run -d -p 8000:8000
 sales-and-inventory-management:1.0

if [ $? -ne 0 ]; then
  echo " Error: Docker container failed to start!"
  exit 1
fi

echo " Setup completed successfully!"
```

signal.py

```
[breaklines=true]
from django.db.models.signals import post_save
from django.dispatch import receiver

from django.contrib.auth.models import User
from .models import Profile


@receiver(post_save, sender=User)
def create_profile(sender, instance,
 created, **kwargs):

    if created:
        Profile.objects.create(user=instance)
        print('profile created!')

@receiver(post_save, sender=User)
def update_profile(sender, instance,
 created, **kwargs):

    if created == False:
        instance.profile.save()
        print('profile updated!')
```

table.py

```
[breaklines=true]
```

```
# tutorial/tables.py
import django_tables2 as tables
from .models import Sale, Purchase
from django.shortcuts import render

class SaleTable(tables.Table):
    class Meta:
        model = Sale
        template_name =
 "django_tables2/semantic.html"
       fields = ('item', 'customer_name',
'transaction_date', 'payment_method', 'quantity',
                    'price', 'total_value', 'amount_received',
 'balance', 'profile')
        order_by_field = 'sort'

class PurchaseTable(tables.Table):
    class Meta:
        model = Purchase
        template_name = "django_tables2/semantic.html"
        fields = ('item', 'vendor', 'order_date',
         'delivery_date', 'quantity',
          'delivery_status', 'price', 'total_value')
        order_by_field = 'sort'
```

url.py

```
[breaklines=true]
from django.urls import path
from . import views
from django.conf.urls.static import static
from django.conf import settings
from .views import (
    PurchaseListView,
    PurchaseDetailView,
    PurchaseCreateView,
    PurchaseUpdateView,
    PurchaseDeleteView,
    SaleListView,
    SaleDetailView,
    SaleCreateView,
    SaleUpdateView,
    SaleDeleteView,
)

urlpatterns = [
```

```
    path('purchases/',
PurchaseListView.as_view(), name="purchaseslist"),
    path('purchase/<slug:slug>/',
PurchaseDetailView.as_view(), name='purchase-detail'),
    path('new-purchase/',
PurchaseCreateView.as_view(), name='purchase-create'),
    path('purchase/<int:pk>/update/',
 PurchaseUpdateView.as_view(), name='purchase-update'),
    path('purchase/<int:pk>/delete/',
 PurchaseDeleteView.as_view(), name='purchase-delete'),
    path('sales/',SaleListView.as_view(), name="saleslist"),
    path('sale/<int:pk>/',
 SaleDetailView.as_view(), name='sale-detail'),
    path('new-sale/', SaleCreateView.as_view(),
 name='sale-create'),
    path('sale/<slug:slug>/update/',
SaleUpdateView.as_view(), name='sale-update'),
    path('sale/<slug:slug>/delete/',
 SaleDeleteView.as_view(), name='sale-delete'),

]
urlpatterns += static(settings.MEDIA_URL,document_roo
t=settings.MEDIA_ROOT)
```

view.py

```
[breaklines=true]
from django.shortcuts import render
from .models import *
from django.contrib.auth.models import User
from .filters import PurchaseFilter, SaleFilter
from django.core.paginator import Paginator, EmptyPage,
 PageNotAnInteger
from django.contrib.auth.mixins import LoginRequiredMixin
, UserPassesTestMixin
from django.views.generic.edit import FormMixin
from django_tables2 import SingleTableView
import django_tables2 as tables
from django.urls import reverse
from django.shortcuts import get_object_or_404
from django_tables2.export.views import ExportMixin
from django_tables2.export.export import TableExport
from .tables import PurchaseTable, SaleTable
from django.core.exceptions import ValidationError
from accounts.models import Profile
from django.views.generic import (
```

```
    ListView,
    DetailView,
    CreateView,
    UpdateView,
    DeleteView
)

class PurchaseListView(ExportMixin, tables.SingleTableView):
    """View to list purchases and export them."""
    model = Purchase
    table_class = SaleTable
    template_name = 'transactions/purchases_list.html'
    context_object_name = 'purchases'
    paginate_by = 10
    SingleTableView.table_pagination = False

class PurchaseDetailView(FormMixin, DetailView):
    """View to display details of a purchase."""
    model = Purchase
    template_name = 'transactions/sale_detail.html'

    def get_success_url(self):
        return reverse('sale-detail', kwargs={'slug':
        self.object.slug})

class PurchaseCreateView(LoginRequiredMixin, CreateView):
    """View to create a new purchase."""
    model = Purchase
    template_name = 'transactions/purchasescreate.html'
    fields = ['item', 'description', 'vendor', 'delivery_date'
, 'quantity', 'delivery_status']

    def form_valid(self, form):
        """Handles the form submission and updates the item's
 quantity."""
        item = form.cleaned_data['item']
        quantity = form.cleaned_data['quantity']

        total_value = item.selling_price * quantity

        form.instance.total_value = total_value
        form.instance.price = item.selling_price

        form.instance.balance = total_value

        form.instance.profile = self.request.user.profile
```

```
        item.quantity += quantity
        item.save()

        return super().form_valid(form)

    def get_success_url(self):
        return reverse('purchaseslist')

    def test_func(self):
        profile_list = Profile.objects.all()
        if self.request.user.profile in profile_list:
            return False
        else:
            return True

class PurchaseUpdateView(LoginRequiredMixin,
UserPassesTestMixin, UpdateView):
    """View to update a purchase."""
    model = Purchase
    template_name = 'transactions/purchaseupdate.html'
    fields = ['item', 'description', 'vendor', 'delivery_date',
 'quantity', 'price', 'delivery_status']

    def test_func(self):
        profiles = Profile.objects.all()
        if self.request.user.profile in profiles:
            return True
        else:
            return False

    def get_success_url(self):
            return reverse('purchaseslist')


class PurchaseDeleteView(LoginRequiredMixin,
 UserPassesTestMixin, DeleteView):
    """View to delete a purchase."""
    model = Purchase
    template_name = 'transactions/purchasedelete.html'

    def test_func(self):
        profiles = Profile.objects.all()
        if self.request.user.profile in profiles:
            return True
        else:
```

```
            return False
    def get_success_url(self):
            return reverse('purchaseslist')

class SaleListView(ExportMixin, tables.SingleTableView):
    """View to list sales and export them."""
    model = Sale
    table_class = SaleTable
    template_name = 'transactions/sales_list.html'
    context_object_name = 'sales'
    paginate_by = 10
    SingleTableView.table_pagination = False

class SaleDetailView(DetailView):
    """View to display details of a sale."""
    model = Sale
    template_name = 'transactions/saledetail.html'


class SaleCreateView(LoginRequiredMixin, CreateView):
    """View to create a new sale."""
    model = Sale
    template_name = 'transactions/salescreate.html'
    fields = ['item', 'customer_name', 'payment_method',
        'quantity', 'amount_received']

    def get_success_url(self):
        return reverse('saleslist')

    def form_valid(self, form):
        """Handles the form submission and
              validates product availability."""
        item = form.cleaned_data['item']
        quantity = form.cleaned_data['quantity']

        if item.quantity < quantity:
 raise ValidationError(f"Only {item.quantity} units of
     '{item.name}' are available.")

        price = item.selling_price

        total_price = price * quantity

        form.instance.price = price
        form.instance.total_value = total_price
```

```
        amount_received = form.cleaned_data['amount_received']
        balance = amount_received - total_price
        form.instance.balance = balance

        form.instance.profile = self.request.user.profile
        return super().form_valid(form)

    def test_func(self):
        profile_list = Profile.objects.all()
        if self.request.user.profile in profile_list:
            return False
        else:
            return True

class SaleUpdateView(LoginRequiredMixin, UserPassesTestMixin,
 UpdateView):
    """View to update a sale."""
    model = Sale
    template_name = 'transactions/sale_update.html'
    fields = ['item', 'customer_name', 'payment_method',
      'quantity', 'price',
                                              'amount_received']

    def test_func(self):
        """Checks if the user has the required permissions to
 access this view."""
        profiles = Profile.objects.all()
        if self.request.user.profile in profiles:
            return True
        else:
            return False

    def get_success_url(self):
        return reverse('saleslist')

    def form_valid(self, form)
 """Handles form submission and sets the profile of the sale."""
        form.instance.profile = self.request.user.profile
        return super().form_valid(form)

class SaleDeleteView(LoginRequiredMixin, UserPassesTestMixin,
 DeleteView):
    """View to delete a sale."""
    model = Sale
    template_name = 'transactions/saledelete.html'
```

```
    def test_func(self):

        """Checks if the user has the required permissions to
     access this view."""
        if self.request.user.is_superuser:
            return True
        else:
            return False
```

LARKSMITE LAB MANAGEMENT

PROJECT REPORT

Submitted By

ADITHYAN K P

Reg. No. CCAVBCA027

For the award of the Degree of

Bachelor of Computer Application (BCA)

(University of Calicut)

under the guidance of

Ms. Soumya P S

Assistant Professor

BACHELOR OF COMPUTER APPLICATION
DEPARTMENT OF COMPUTER SCIENCE
CHRIST COLLEGE
IRINJALAKUDA, KERALA
2021-2024

DEPARTMENT OF COMPUTER SCIENCE

CHRIST COLLEGE (AUTONOMOUS)

IRINJALAKUDA

CERTIFICATE

This is to certify that the project report entitled ***"Larksmite Lab Management"*** *is a bonafide record of the project work done by* ***Adithyan K P*** *in partial fulfillment of the requirement for the sixth semester of* ***Bachelor of Computer Application*** *in Department of Computer Science of* ***CHRIST COLLEGE (AUTONOMOUS) IRINJALAKUDA***

.

Ms. Soumya P S
Assistant Professor
Internal Guide

Ms. Sini Thomas
Head of the Department
Computer Science

EXTERNAL EXAMINER

INTERNAL EXAMINER

DECLARATION

I hereby declare that this project work **"LARKSMITE LAB MANAGEMENT"** submitted to Christ College (Autonomous) Irinjalakuda, affiliated to Calicut University in partial fulfillment of the requirement for the award of the Bachelor of Computer Application, is a record of original work done by me, under the guidance of Ms. SOUMYA P S, Assistant Professor, Department of Computer Science.

Place: Irinjalakuda ADITHYAN K P

ACKNOWLEDGEMENT

First and foremost we wish to thank Lord almighty for his providence and for being the guiding light throughout the project. We take this opportunity to express our gratitude to our beloved class teacher as well as project guide, Ms. SOUMYA P S, who has been hugely supportive throughout the course of this project. We wish to express our sincere gratitude to our head of department, Ms. SINI THOMAS for giving us all the required facilities for our project. We are thankful for their aspiring guidance and valuable advice during the project work. We would like to take this opportunity to specially thank all other faculty members for their constant and continuous motivation. Finally we would like to thank our family and friends for giving valuable advice and moral support throughout our project.

ABSTRACT

Larksmite Lab Management is an innovative software introduced to manage computer labs. The software is enriched with multiple features, including - status monitoring, process monitoring, screen monitoring, power control, message passing and so on. The client side of the software automatically runs in the background in all computers in the lab. All these features make this software powerful and useful in computer labs.

Contents

Chapter 1

1 Introduction

Larksmite is a sophisticated computer lab management software designed to provide administrators with seamless control and oversight of connected devices. Built using Flutter and Dart, the software offers a bird's eye view of the lab's layout, enabling easy device identification. With features such as screen sharing, power controls, and process monitoring, administrators can efficiently manage resources and troubleshoot issues remotely. The project employs RSA public-key cryptography for secure communication and utilizes SQLite for data storage. Larksmite aims to simplify computer lab administration, offering a centralized solution for monitoring, controlling, and maintaining an optimal computing environment.

1.1 Overview

Larksmite is a streamlined computer lab management solution, prioritizing user-friendly interfaces for efficient device identification. With features such as screen sharing, power controls, and detailed reporting, the platform simplifies administration tasks. Utilizing Flutter and Dart, it ensures responsiveness, while employing RSA public-key cryptography and SQLite for secure communication and data storage. Larksmite's core objective is to centralize control and optimize the management of multiple devices within a network.

Chapter 2

2 System Analysis

2.1 Purpose

The purpose of Larksmite is to streamline computer lab management, offering a centralized solution for efficient device control and monitoring. This project aims to simplify administration tasks, enhance accessibility, and optimize the management of multiple devices within a network.

2.1.1 Existing System

Existing computer lab management systems have limitations in scalability, security, and user interface. Most of them also only work on Microsoft Windows, prompting the development of a cross-platform software. This new system aims to enhance performance, address security concerns, and offer a more user-friendly solution for streamlined device control and monitoring.

2.1.2 Proposed System

Larksmite is envisioned as a revolutionary solution for computer lab management, aiming to address the limitations of the existing system. The proposed system prioritizes a centralized and user-friendly approach to streamline device control and monitoring. Larksmite seeks to enhance efficiency, security, and user experience in computer lab administration, providing administrators with an optimized tool for seamless device management

2.2 Problem Definition

The existing computer lab management system faces challenges in scalability, security, and user interface, hindering efficient device control and monitoring. These limitations necessitate the development of Larksmite.

2.3 Feasibility Study

After the problem is clearly understood and solutions proposed, the next step is to conduct the feasibility study. Feasibility study is defined as evaluation or analysis of the potential impact of the proposed project or program. The objective is to determine whether the proposed system is feasible. There are three aspects of feasibility study to which the proposed system is subjected as discussed below.

2.3.1 Technical Feasibility

Technical feasibility assesses whether the current technical resources are sufficient for the new system. The selected combination of Flutter and Dart offers a robust framework for building a responsive and visually appealing user interface, which is cross-platform by default. This ensures a wide array of compatibility.

2.3.2 Economical Feasibility

Economic feasibility determines whether time and money are available to develop the system. There is no additional hardware needed to develop the software, neither does it need any extra hardware to run.

2.3.3 Operational Feasibility

Operational feasibility determines if the human resources are available to operate the system once it has been installed. No extra training is needed to use this system. Anyone who has basic computer knowledge can easily use the system. The resources that are required to implement or install are already available with the organization.

Chapter 3

3 Software Requirement Specification

3.1 Purpose

The purpose of the software specification is to precisely outline the requirements of Larksmite, serving as a blueprint for its development. This document articulates the specific objectives the software aims to achieve, detailing user interactions, system behaviors, and performance expectations. It is primarily intended for the computer lab administrator for managing and understanding the proposed system.

3.2 Scope

Our project scope encompasses the development of Larksmite, a computer lab management software. It includes features such as computer lab layout visualization, real-time screen monitoring, power controls, and detailed reporting to optimize the efficiency of computer lab administration.

3.3 Overall Description

This section gives an overview of our software - Larksmite. The software offers an intuitive interface, allowing administrators a bird's eye view of the lab layout for streamlined device identification. Incorporating features such as real-time screen monitoring, power controls, and detailed reporting, Larksmite ensures efficient and secure administration.

3.3.1 Product Perspective

LarkSmite is mainly used for managing a computer lab in any institutions. It can be used to monitor each computer screens, orchestrate power options and generate reports.

3.3.2 Product Functionality

Tasks such as device identification, real-time screen monitoring, power controls, process monitoring and detailed reporting are various functions of LarkSmite

3.3.3 Users and Characteristics

There are two types of users within the LarkSmite system: administrators and clients. Administrators, tasked with overseeing computer labs, require access to comprehensive management features, while clients, interacting with the client computer system, remain oblivious to the existence of monitoring tool which always runs in the background.

3.4 Specific Requirements

3.4.1 Hardware Requirements

- System: IBM-Compatible PC
- Processor: Pentium IV or above
- Speed: Above 1GHz
- RAM capacity: 512 MB
- Hard Dsk drive: 40 GB
- Keyboard: Standard
- Mouse: Standard
- Monitor: SVGA Color

3.4.2 Software Requirements

- Operating System: Ubuntu or Gnome DE based Linux
- Languages used: Dart
- Database: SQLite
- Technologies used: Flutter, Shelf/Dart

3.5 Functional Requirements

It contains two main modules.

- Teacher
- Student

Teacher

In LarkSmite, teachers have the functional requirement to use the system as administrators to oversee and manage computer labs. This includes tasks such as device identification, real-time screen monitoring, and utilizing power controls, providing them with the necessary tools for effective supervision and control in an educational setting.

Student

Students in LarkSmite have the functional requirements focused on interacting with their specified client computer, performing their assigned lab task.

3.6 Non Functional Requirements

Non-functional requirements define the overall qualities or attributes of the resulting system. Non-functional system place restrictions on the product being developed, the developed process, and specify external constraints that the product must meet. Examples of non-functional requirements include security, usability, reliability and performance requirements. The principal non - functional constraints which are relevant to critical systems:

- Performance
- Security
- Usability

Performance

Performance requirements concern the speed of operation of a system. Types of performance requirements:

- Response requirements (how quickly the system reacts to a user input).
- Throughput requirements (how much the system can accomplish within a specified amount of time).
- Availability requirements (is the system available for service when requested by end users). The speed of operation of this system is adequate for the requirements.

Reliability

- Reliability is the ability of a system to perform its required function under stated conditions for a specified period of time.
- Constraints on the runtime behavior of the system. This system is reliable because its functionalities can be done on the required conditions.

Safety

Safety requirements refer to the specific criteria, conditions, and constraints that must be met to ensure the safe operation of a software system.

Usability

Usability is the ease with which a user can learn to operate, prepare inputs for, and interpret outputs of system or components. Usability requirements include:

- Understandable error messages.
- Well-formed user interfaces.

3.7 Interface Requirements

3.7.1 Hardware Interfaces

The system must be run in a LAN network, all client devices shall be required to connect to the same LAN network, for example, a WIFI access point or an Ethernet LAN network.

3.7.2 Software Interfaces

Software interface required for the working of the project is the appropriate operating system.

3.7.3 Communication Interfaces

The clients communicate using HTTPS and Secure Websocket.

3.8 Security Requirements

- Only authorized clients are allowed to connect.
- All communication between administrators and client devices is encrypted.
- Unauthorized devices cannot act as a server and clients will not connect to imposter servers

3.9 Platform Used

Ubuntu is a Linux distribution based on Debian and is free and open-source, making it a preferred choice for many institutions. It is developed by Canonical. The default distribution uses the Gnome Desktop Environment, with official support for other desktop environments like KDE, XFCE and Mate. The latest stable version is Ubuntu 23.10 (Mantic Minotaur) and the latest LTS (Long-Term Support) version is Ubuntu 22.04 LTS (Jammy Jellyfish). LarkSmite is designed to operate seamlessly within Linux environments, leveraging the robust features and stability offered by this platform.

3.10 Technologies Used

Dart

Dart serves as the cornerstone programming language in LarkSmite, providing the foundation for implementing the software's logic and functionality. Known for its simplicity and efficiency, Dart facilitates the development of a responsive and intuitive system. With features like strong typing and Just-In-Time (JIT) compilation, Dart enhances the codebase, contributing to the overall reliability and performance of LarkSmite's computer lab management solution.

Flutter

Flutter is an open-source UI (User Interface) software development kit (SDK) created by Google. It allows developers to build native applications for Android, Linux, Windows, iOS as well as macOS platforms using a single codebase. Flutter utilizes the Dart programming language. Flutter offers a hot reload feature, allowing developers to see changes in real-time as they modify the code.

SQLite

LarkSmite has opted for SQLite due to its lightweight and embedded nature. The choice of SQLite aligns with the project's goals of simplicity and seamless integration within the context of computer lab management, providing an efficient and self-contained solution for data storage and retrieval.

Chapter 4

4 Design Document

4.1 Purpose

The purpose of LarkSmite is to revolutionize computer lab management by offering administrators a centralized and user-friendly platform. Through intuitive interfaces and features like real-time screen monitoring, power controls, and detailed reporting, LarkSmite aims to streamline tasks for administrators, teachers, and students. Developed with Flutter in Dart language, the system prioritizes responsiveness and scalability, aiming to enhance the efficiency of computer lab administration within educational environments.

Currently built for Linux environment, LarkSmite seeks to optimize the computer lab management experience. With a focus on security through RSA public-key cryptography and SSL, the system aims to provide a robust, secure, and accessible solution for overseeing and controlling multiple devices within educational settings.

4.2 Scope

Our project, LarkSmite, primarily encompasses the development of a computer lab management software tailored for Linux environments. It includes features for teachers, and students, focusing on efficient device control, real-time screen monitoring, and detailed reporting within the educational context

4.3 Overview

The purpose of this document is to provide a comprehensive guide to LarkSmite, a specialized computer lab management software designed for Linux environments. It outlines the features, functionalities, and the technology stack utilized in the development of the system. By detailing the project's scope, objectives, and key components, this document aims to offer a clear understanding of LarkSmite's role in revolutionizing computer lab administration within educational institutions.

4.4 Data Design

Database is the storehouses of data used in the software systems. The data is stored in tables inside the database. Several tables are created for the manipulation of the data for the system. Two essential settings for a database are

- Primary Key- The field that is unique for all the record occurrence.
- Foreign Key- The field used to set relation between tables.

Normalization

- Normalization is the process of organizing the data in the database.
- Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate the undesirable characteristics like Insertion, Update and Deletion Anomalies.
- Normalization divides the larger table into the smaller table and links them using relationship.
- The normal form is used to reduce redundancy from the database table.

Tables

Server Asymmetric Key

Name	DataType	Constraints	Description
key_id	INTEGER(5)	Primary Key	ID of public-private key pair
public_key	BLOB	Not Null	Public key
private_key	BLOB	Not Null	Private key

Certificates

Name	DataType	Constraints	Description
cert_id	INTEGER(5)	Primary Key	ID of SSL certificate
public_cert	BLOB	Not Null	Public key
private_key	BLOB	Not Null	Private key

IP Address

Name	DataType	Constraints	Description
address	TEXT(20)	Not Null	IP Address of server
cert_id	INTEGER(5)	Foreign Key	Corresponding certificate cert_id

Shared User Key

Name	DataType	Constraints	Description
name	VARCHAR(50)	Not Null	Name of client
shared_key	TEXT(6)	Not Null	Authorization key of client

Chapter 5

5 Development of the System

The software is split into two - server side and client side. In the server side, the project is divided based on major features like networking, datastore, user interface. Each feature is further sub-divided into presentation, domain and data layer to avoid unwanted coupling and increase cohesion. This ensures that each submodule is modular, that is, it can be replaced without affecting the other parts.

In the client side, the project is divided into two major parts - platform dependent and platform independent code. The platform independent code implements features which are not dependent on the platform in which it will be run on. The platform dependent code is dependent on the platform, thus it is different on different operating systems and/or versions. This separation is designed using interfaces, which defines common functions which should be implemented by all platform dependent implementors. By employing such a split, we can achieve optimal code sharing and enhance efficiency.

Chapter 6

6 System Testing

Testing is the penultimate step of software development. An elaborate suite of test data is prepared and the system is tested using this test data. Corrections are made using the results of failed tests and any changes are immediately tested against this test suite, to detect regression. System testing is aimed at ensuring the system works accurately before the live operation commences.

6.1 Test Plan

6.1.1 Scope

This test plan will cover the following testing activities as identified in the testing strategy.

White Box Testing: White box testing is a testing technique, that examines the program structure and derives test data from the program logic/code. In white box testing, the UI is bypassed. Testing is done directly at the code level and the results are compared against specifications. It reveals errors in "hidden" code.
Black Box Testing: Black - box testing is a method of software testing that examines the functionality of an application based on the specifications. Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end-user would.
Unit Testing: The module includes four main components to undergo unit testing. It checks the syntax error, logic error and validity of the program correctness. This test will be performed by the developers.
Integration Testing: After all components pass the unit testing, they get to be tested whether they work correctly when they are running concurrently and communicating to each other. Integration tests considers the entire subsystem and ensure that a set of components play nicely together.
Internal Data Testing: We will test the validity of the data before it enters the database to avoid any problems that may be faced in the database. We will test the encryption of the personal information of all the users along with the admin user names and passwords to ensure maximum security of the admin user.

6.1.2 Software Risk Issues

In this section, the plan is to test the risk involved in critical issues such as:

- Difficulty in setting up the dependencies
- Establishing proper network security

- Ensuring data transmission security

6.1.3 Features to be Tested

- Test whether correct admin user name and password allows you to login.
- Test whether invalid admin user name and password prevents you from login.
- Test whether admin can open valid layout files.
- Test whether server software will not crash when opening invalid layout file
- Test whether server software will start service advertising and start server
- Test whether server software is able to generate SSL certificate and RSA key pair
- Test whether server software is able to perform client functions and display results
- Test whether client software continuosly searches for server in network
- Test whether client connects to server with valid client details
- Test whether client automatically connects to server when details are saved
- Test whether client responds to server requests
- Test whether client detects a fake server and doesn't connect
- Test whether client restarts from potential errors and keep running

6.2 Test Consolidation

6.2.1 Test Item

The items or features to be tested in the test cases are included in the document. Each and every input is tested. The present condition of the system is tested. It is checked to make sure that environment is ready for the application to work.

6.2.2 Input Specifications

These are the inputs required to execute the test case. All required inputs including data elements and values are given below.

Data	Values
Username	A valid username
Password	Strong combination of characters and numbers

Chapter 7

7 System Implementation and Maintenance

7.1 Implementation

System implementation is the conversion of a proposed system into an operational one which involves writing code, training clients and installing softwares. User training is crucial for helping them understand the versatily and helpfulness of the new system.

Software maintenance becomes necessary to ensure the system continues to operate satisfactorily in response to changes in the user's environment. Maintenance activities often involve making minor enhancements or corrections to address issues that may arise during system operations.

In the implementation phase, the team builds the components either from scratch or by composition. Given the architecture document from the design phase and the requirement document from the analysis phase, the team should build exactly what has been requested, though there may be still room for innovation and flexibility. For example, a component may be narrowly designed for this particular system, or the component may be made more general to satisfy a reusability

- Careful planning
- Investigation of system and constraints
- Design the methods to achieve changeover.
- Training the staff in the changed phase.
- Evaluation of changeover method.
- The method of implementation and time scale to be adopted are found out initially.

7.2 Maintenance

This phase occurs as a result of deploying the whole system at the end users organization. They will perform the beta testing and inform to the developers about any required modification to the application. The customer records all the problems that are encountered during the beta testing and reports these to the developer at regular intervals.

7.2.1 Corrective Maintenance

Even with the best quality assurance activities, it is likely that the customer will uncover defects in the software. Corrective maintenance changes the software to correct the defects. Corrective Maintenance activity may consist of repair,

restoration or replacement of equipment. This activity will be result of a regular inspection, which identifies the failure in time for corrective maintenance to be planned and scheduled, then performed during a routine maintenance shutdown.

7.2.2 Adaptive Maintenance

Over time, the original environment (CPU, operating system, business rules, external product characteristics) for which the software was developed is likely to change. Adaptive maintenance results in modification to the software to accommodate changes to its external environment.

7.2.3 Enhanced Maintenance

As software is used, the customer/user will recognize additional functions that will provide added benefits. Perfect maintenance extends the software beyond its original functional requirements.

7.2.4 Preventive Maintenance

Computer software deteriorates due to change, and because of this, preventive maintenance often called software re-engineering, must be conducted to enable the software to serve the needs of its end users. Preventive maintenance makes changes to computer programs so that they can be more easily corrected, adapted and enhanced.

Chapter 8

8 Conclusion and Future Scope

8.1 Conclusion

The objective of LarkSmite is to streamline and enhance the efficiency of managing computer labs in educational institutions or any organization that operates computer labs. It enables the admin/teacher to manage all the computers through a single admin computer. The admin is able to monitor the screens of each computer, view running processes, see basic information about the computer, send messages, perform power controls like sleep, restart, shutdown remotely and generate PDF reports with screenshots. The student's computer behaves the same way, with the client software running in the background.

8.2 Future Scope

- Allow sharing and showing teacher's computer screen on all clients
- Allow remote access similar to VNC, RDP and TeamViewer.
- Add file sharing
- Perform seamless upgrade of software

Appendix

A Activity Diagram

Activity Diagrams are used to illustrate the flow of control in a system and refer to the steps involved in the execution of a use case. It is a type of behavioral diagram and we can depict both sequential processing and concurrent processing of activities using an activity diagram ie an activity diagram focuses on the condition of flow and the sequence in which it happens.
Some Activity Diagram charting forms:

A.1 Initial State

The initial state marks the entry point and the state of the system before the application is opened

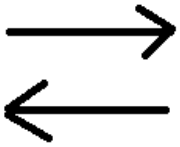
A.2 Final State

The state which the system reaches when a particular process or activity ends is known as a Final State or End State.

A.3 Activity State

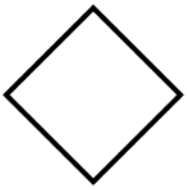
An activity represents execution of an action on objects or by objects. Any action or event that takes place is represented using an activity.

A.4 Control Flow



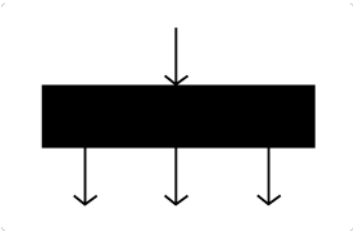
They are used to show the transition from one activity state to another activity state.

A.5 Decision Node



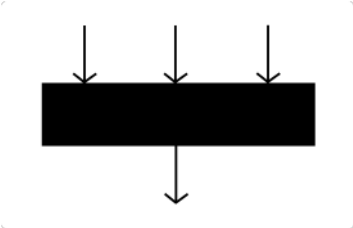
When we need to make a decision before deciding the flow of control, we use the decision node. The outgoing arrows from the decision node can be labelled with conditions.

A.6 Fork



Fork nodes are used to support concurrent activities. We use a fork node when multiple activities get executed concurrently.

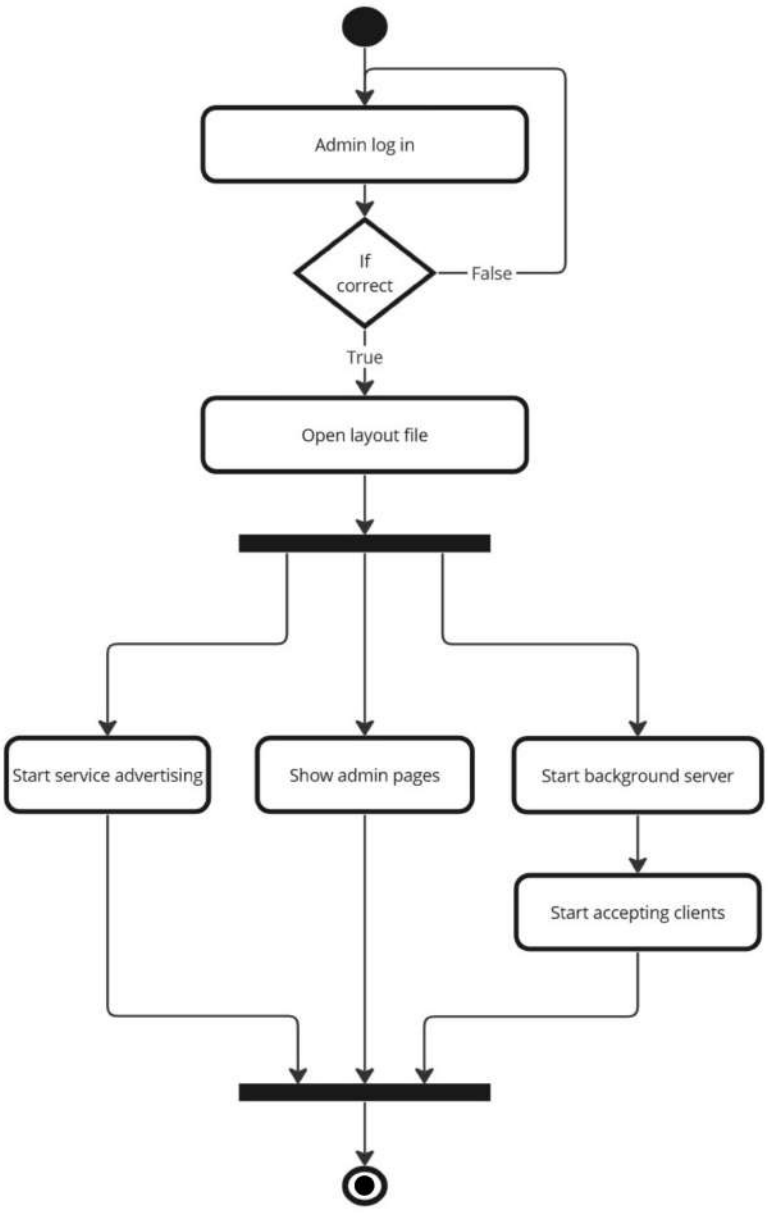
A.7 Join



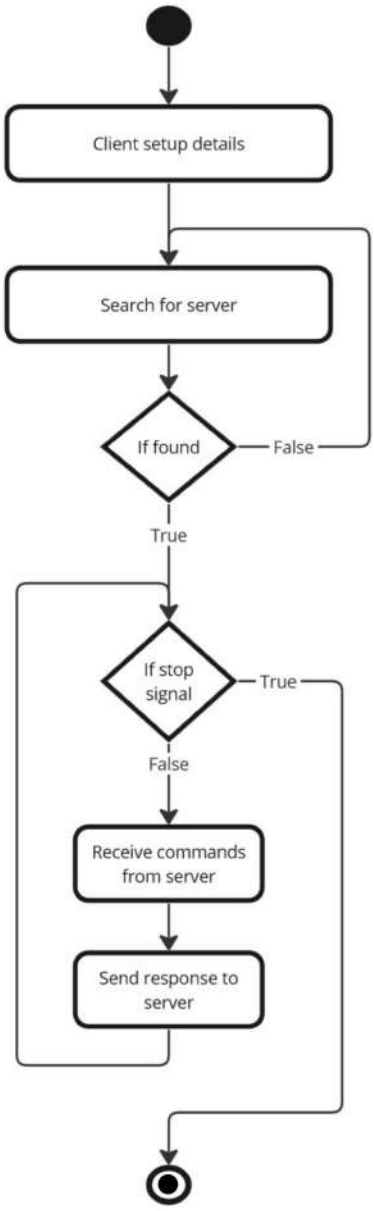
Join nodes are used to support concurrent activities converging into one.

A.8 Activity Diagram

A.8.1 Server Side



A.8.2 Client Side



B Sequence Diagram

Sequence diagrams show elements as they interact over time and they are organized according to object (horizontally) and time (vertically). They capture the interaction between objects in the context of a collaboration. Sequence diagrams are time focused and they show the order of the interaction visually by using the vertical axis of the diagram to represent time.

B.1 Actor

An actor represents a type of role where it interacts with the system and its objects. An actor is always outside the scope of the system. We use actors to depict various roles including human users and other external subjects

B.2 Lifeline

A lifeline is a named element which depicts an individual participant in a sequence diagram. A lifeline always portrays an object internal to the system.

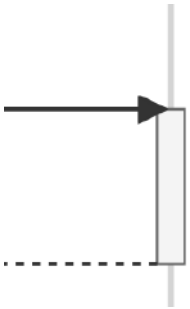
B.3 Messages

Communication between objects is depicted using messages. The messages appear in a sequential order on the lifeline.

B.4 Response Messages

Reply messages are used to show the message being sent from the receiver to the sender.

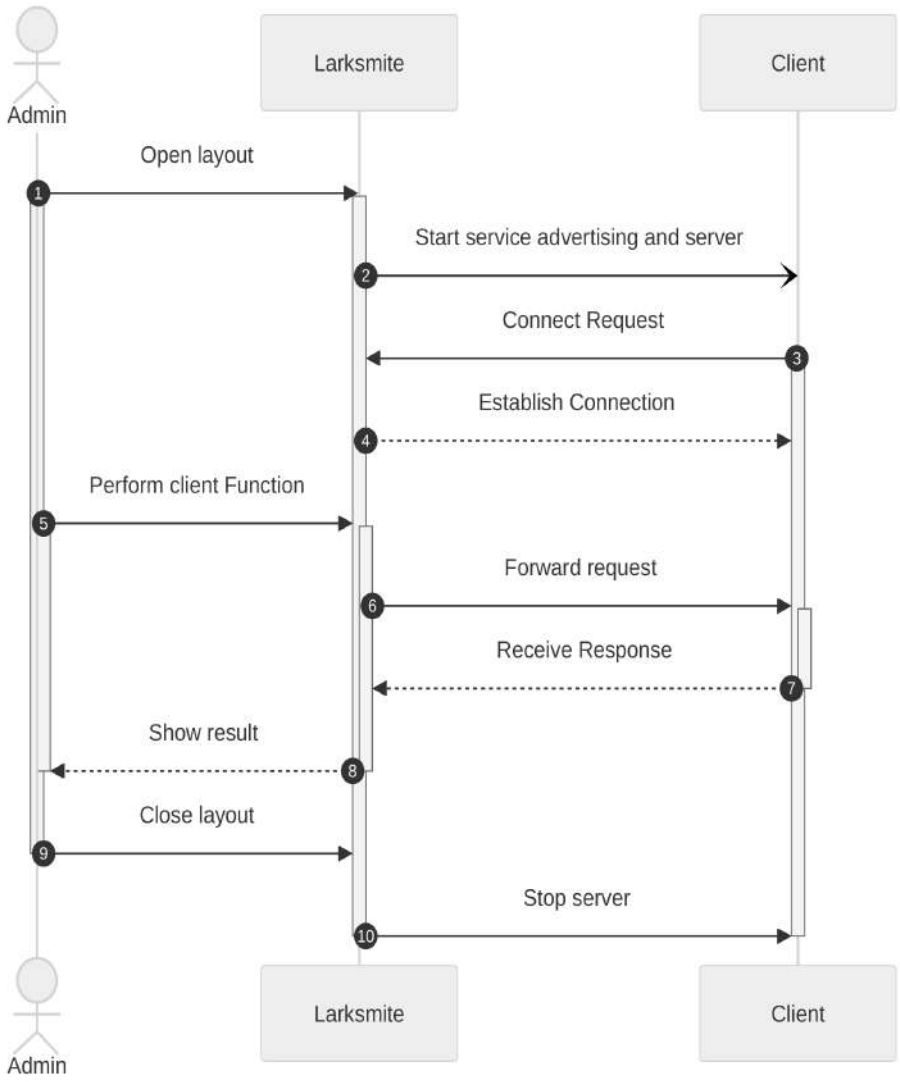
B.5 Activation



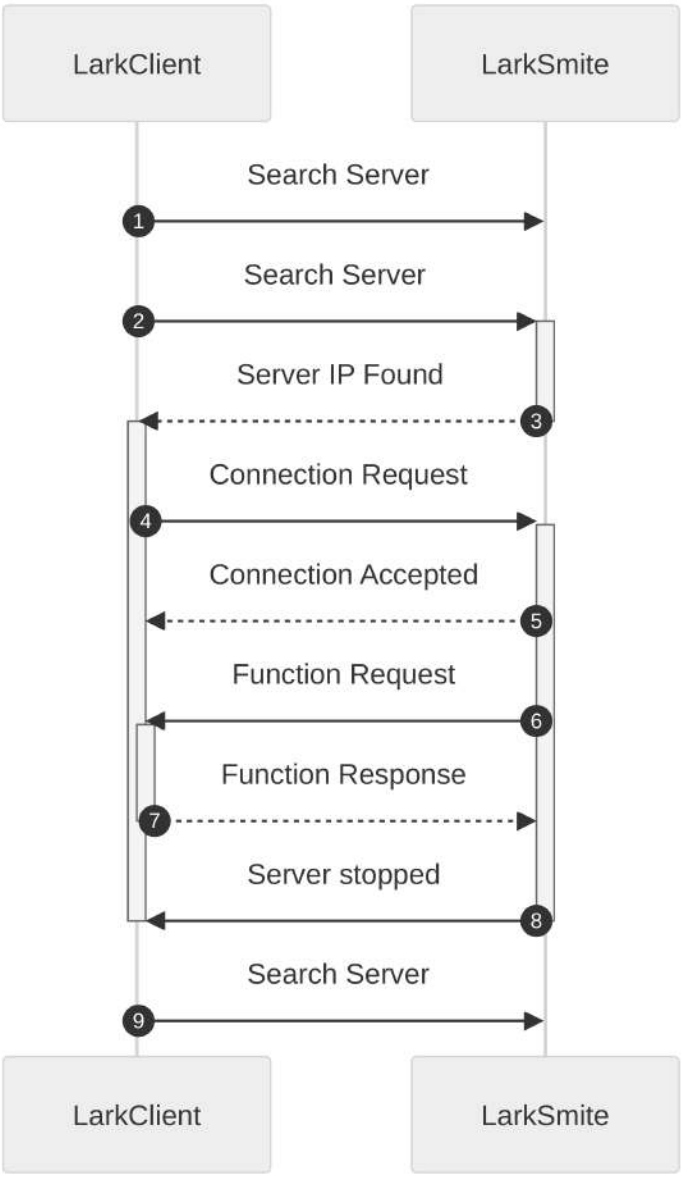
A thin rectangle on a lifeline represents the period during which an element is performing an operation. The top and the bottom of the rectangle is aligned with the initiation and the completion time respectively.

B.6 Sequence Diagram

B.6.1 Server Side

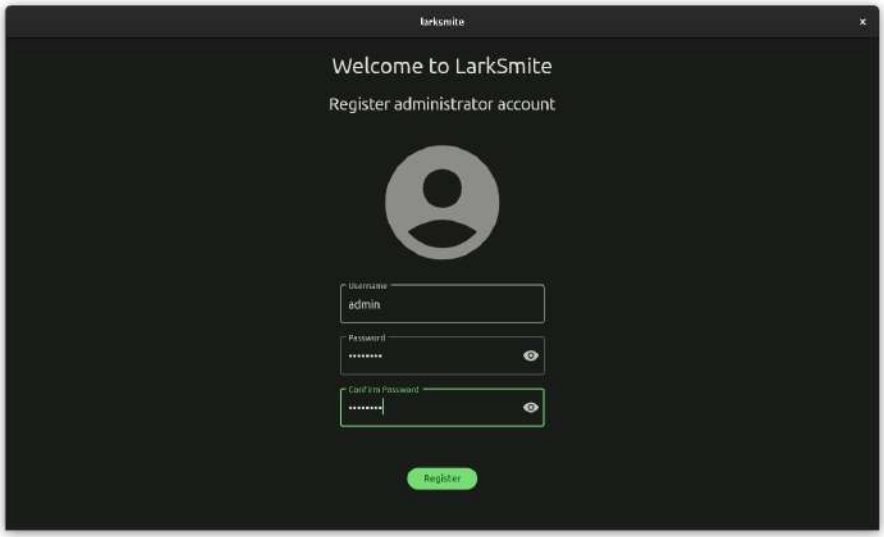


B.6.2 Client Side

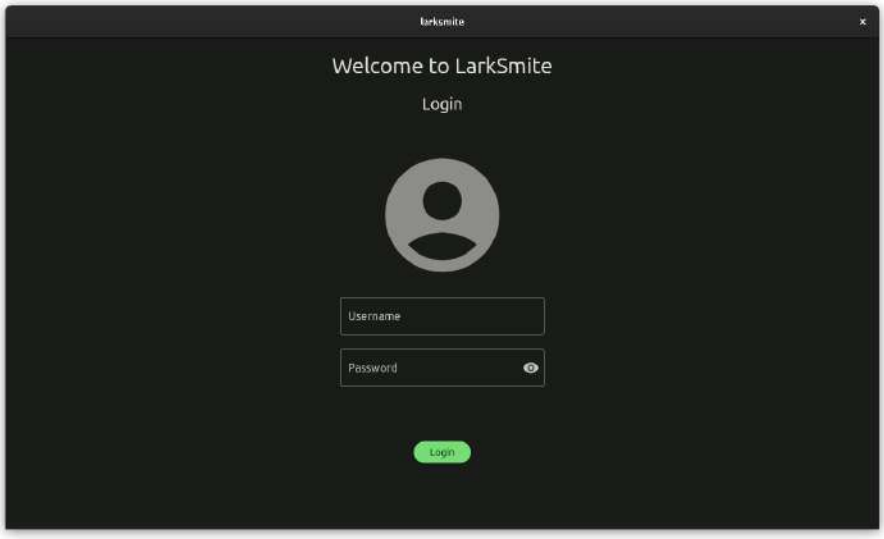


C User Interfaces

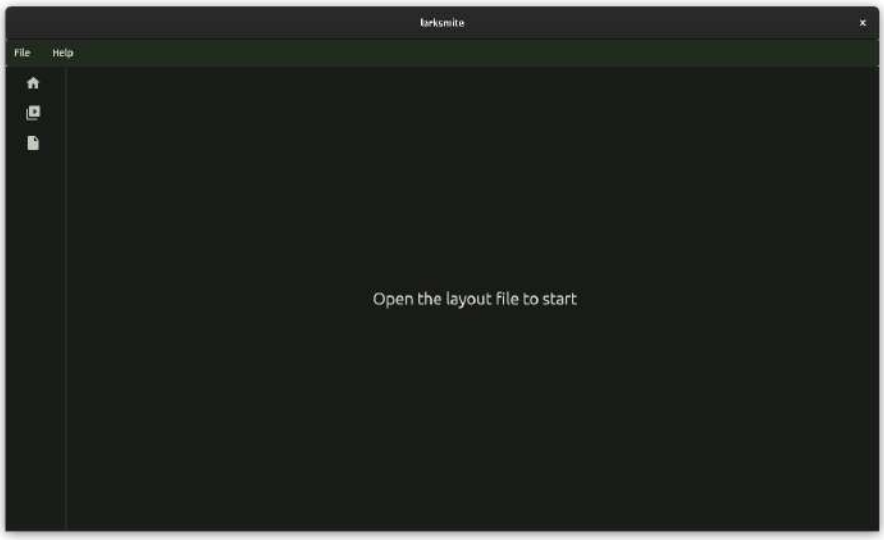
C.1 Admin Registration



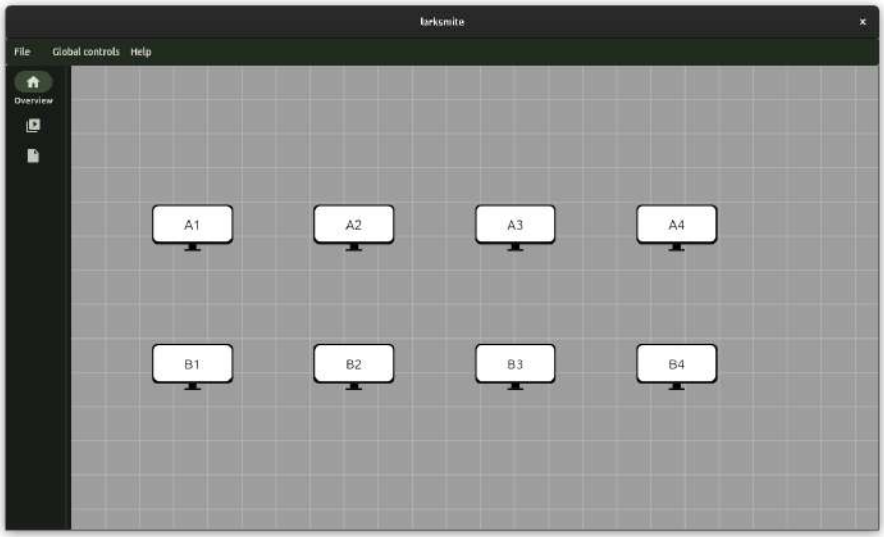
C.2 Admin Login



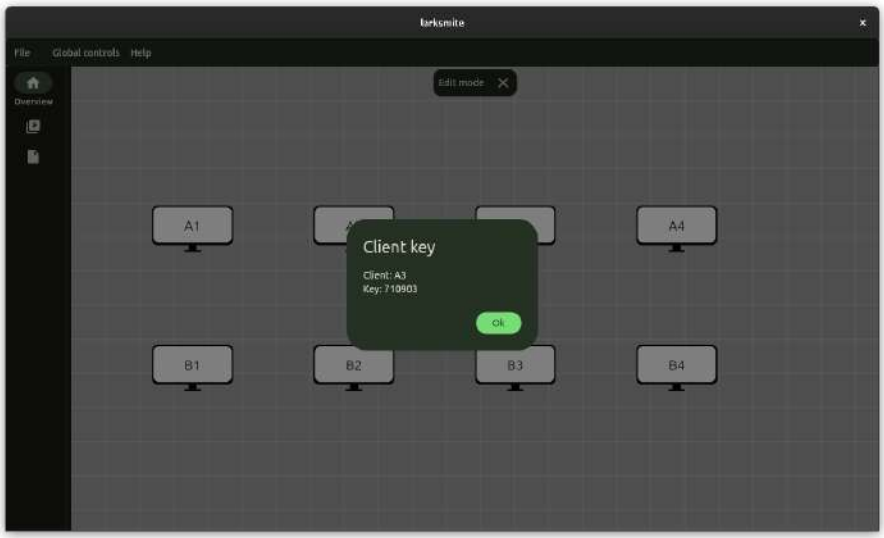
C.3 Home Page



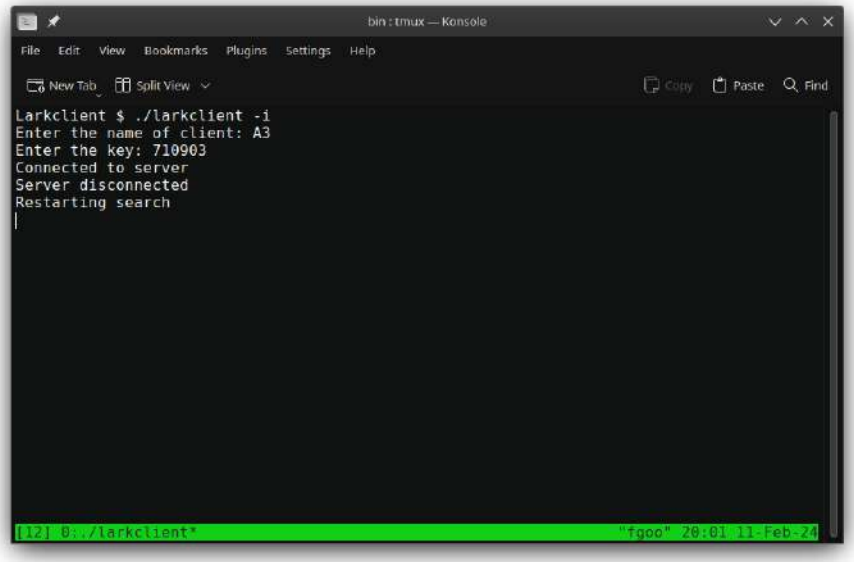
C.4 Layout File Opened



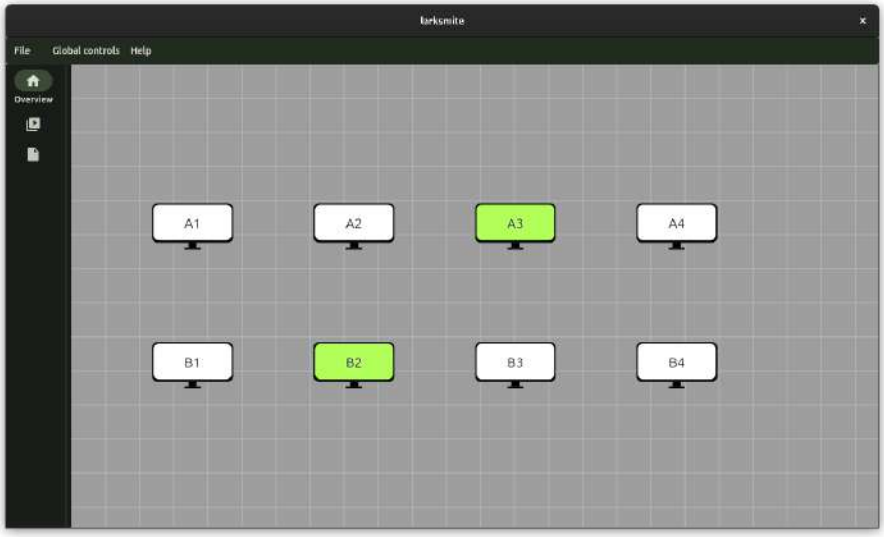
C.5 Adding Clients



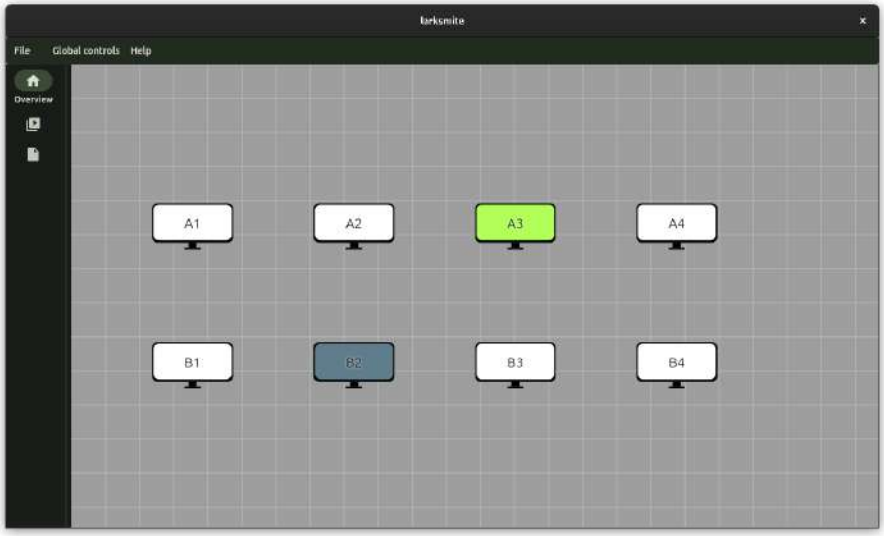
C.6 Settings up Larkclient



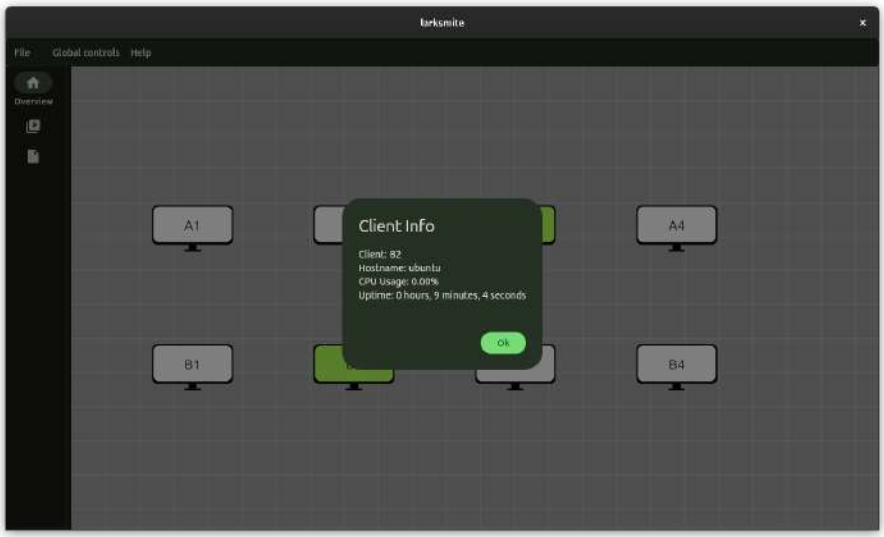
C.7 Client Realtime Status



C.8 Color coded status

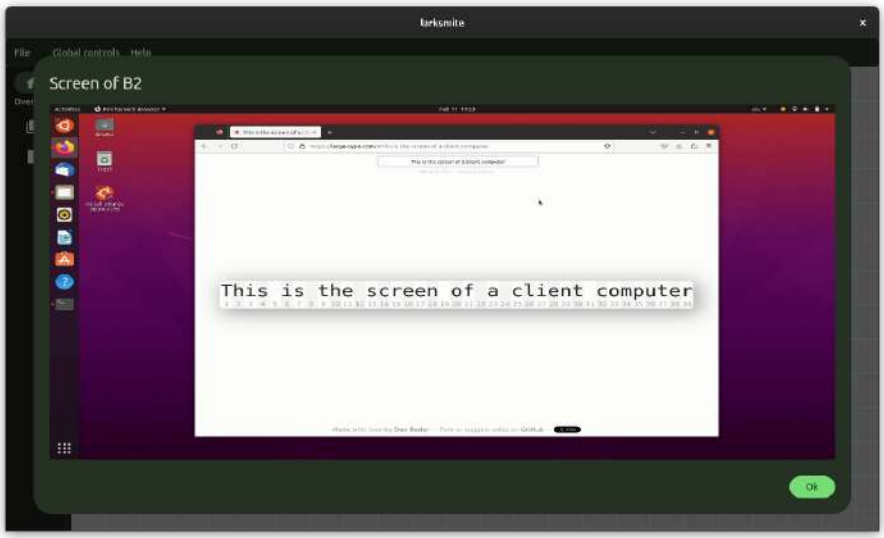


C.9 Client Info

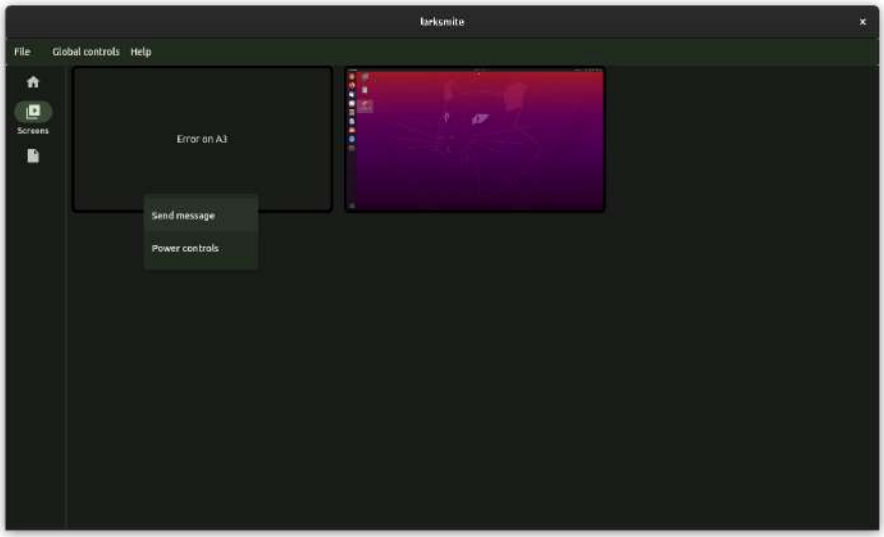


C.10 Client Processes

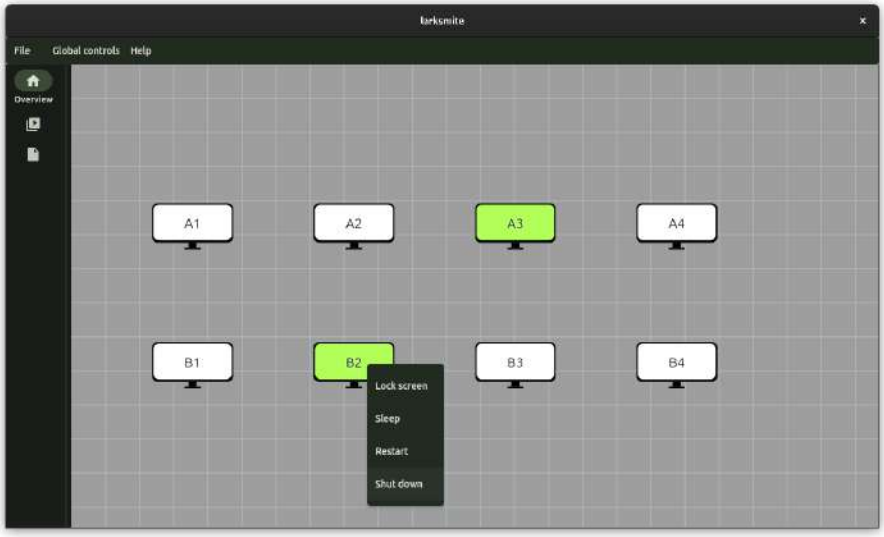
C.11 Single Screen Stream



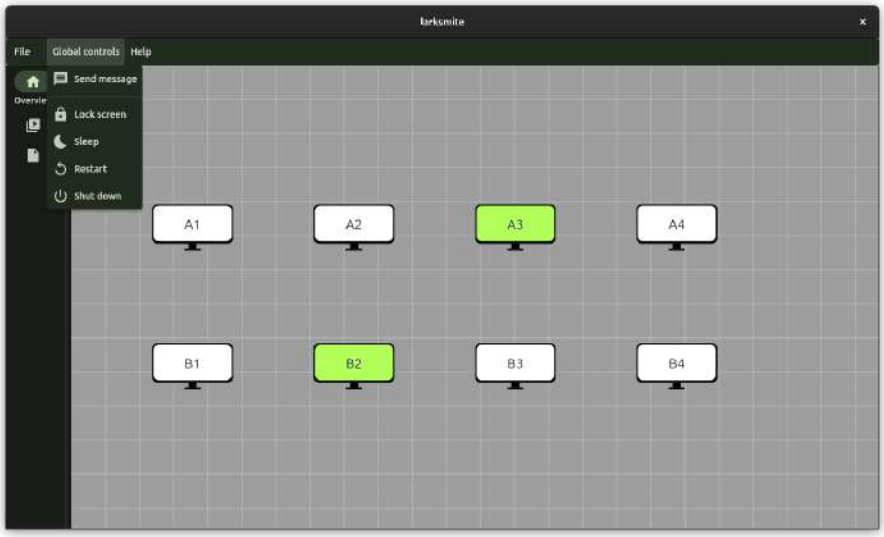
C.12 All Screen Stream



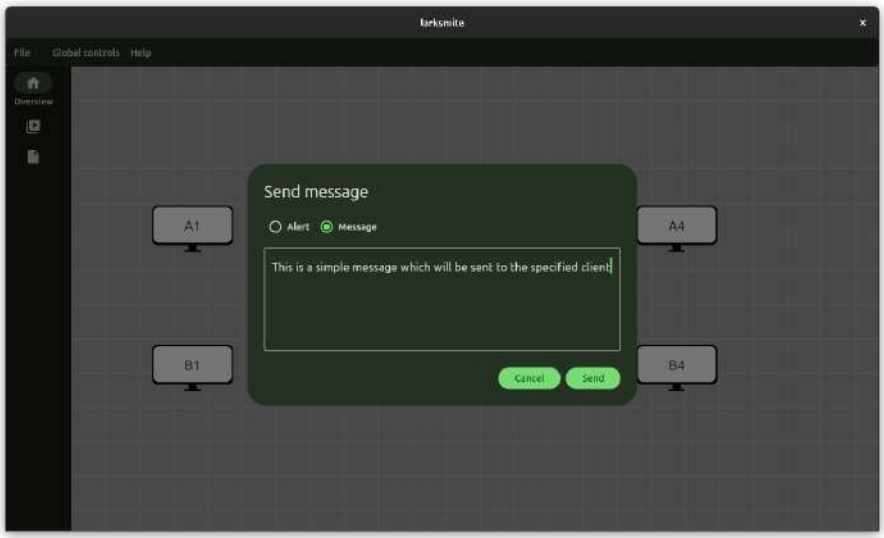
C.13 Power Control



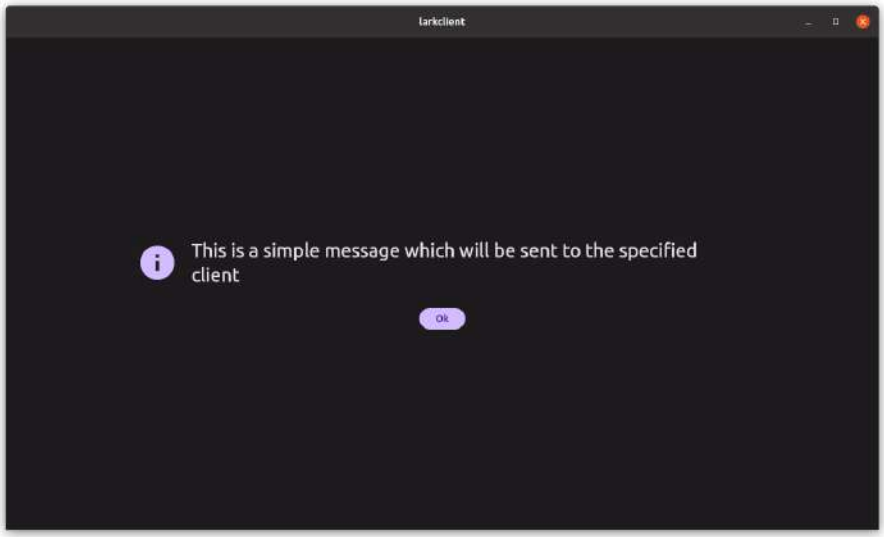
C.14 Global Control



C.15 Sending Message



C.16 Message on Client Side



D Code

main.dart

```
import 'package:flutter_bloc/flutter_bloc.dart';
import 'package:flutter/material.dart';
// import 'package:yaru/yaru.dart';
import 'core/injection_container.dart' as core_di;
import 'features/login/login_injection_container.dart' as login_
    di;
import 'features/dashboard/dashboard_injection_container.dart' as
     dashboard_di;
import 'features/networking/networking_injection_container.dart'
    as networking_di;
import 'features/datastore/datastore_injection_container.dart' as
     datastore_di;
import 'core/cubit/core_data_cubit.dart';
import 'core/router.dart';
import 'core/logging.dart' as logging;

void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await logging.initLogger();
  await core_di.init();
  await datastore_di.init();
  await login_di.init();
  await networking_di.init();
  await dashboard_di.init();
  runApp(const LarkSmiteApp());
}

class LarkSmiteApp extends StatelessWidget {
  const LarkSmiteApp({super.key});
  @override
  Widget build(BuildContext context) {
    return BlocProvider(
      create: (context) => CoreDataCubit(firstStartChecker: core_
          di.sl()),
      child: MaterialApp.router(
        title: 'LarkSmite',
        // theme: yaruLight,
        // darkTheme: yaruDark,
        //
        debugShowCheckedModeBanner: false,
        routerConfig: LarkSmiteRouter.router,
        darkTheme: ThemeData(
```

```
          brightness: Brightness.dark,
          // colorScheme: ColorScheme.fromSeed(seedColor: Colors.
              amber),
          colorSchemeSeed: Colors.green,
          useMaterial3: true,
        ),
      ),
    );
  }
}
```

dashboard.dart

```
import 'package:flutter/material.dart';
import 'package:larksmite/features/dashboard/presentation/cubit/
    app_menu_bar_cubit.dart';
import 'package:larksmite/features/dashboard/presentation/cubit/
    report_view_cubit.dart';
import 'package:larksmite/features/dashboard/presentation/cubit/
    screen_view_cubit.dart';
import 'dashboard_view_frame.dart';
import '../widget/navigation_rail_side.dart';
import '../cubit/birds_eye_view_cubit.dart';
import '../cubit/dashboard_cubit.dart';
import 'package:flutter_bloc/flutter_bloc.dart';
import '../../dashboard_injection_container.dart' as dashboard_di
    ;
import 'package:larksmite/features/networking/networking_
    injection_container.dart'
    as networking_di;
import 'package:larksmite/features/datastore/datastore_injection_
    container.dart'
    as datastore_di;
import 'package:file_picker/file_picker.dart';
import '../widget/app_menu_bar.dart';

class DashboardFrame extends StatefulWidget {
  const DashboardFrame({super.key});

  @override
  State<DashboardFrame> createState() => _DashboardFrameState();
}

class _DashboardFrameState extends State<DashboardFrame> {
  @override
  Widget build(BuildContext context) {
    return MultiBlocProvider(
```

```
providers: [
  BlocProvider(
    create: (context) => DashboardCubit(
      loadLabLayout: dashboard_di.sl(),
      serviceAdvertising: networking_di.sl(),
      manageClientConnections: networking_di.sl(),
    ),
  ),
  BlocProvider(
    create: (context) => BirdsEyeViewCubit(
      manageClientConnections: networking_di.sl(),
      larksmiteDatastore: datastore_di.sl(),
    ),
  ),
  BlocProvider(
    create: (context) => ScreenViewCubit(
      manageClientConnections: networking_di.sl(),
    ),
  ),
  BlocProvider(
    create: (context) => ReportViewCubit(
      manageClientConnections: networking_di.sl(),
    ),
  ),
  BlocProvider(
    create: (context) => AppMenuBarCubit(),
  ),
],
child: BlocListener<DashboardCubit, DashboardState>(
  listener: (context, state) async {
    switch (state) {
      case DashboardShowFilePicker():
        FilePickerResult? filePickerResult =
            await FilePicker.platform.pickFiles(
          initialDirectory: ".",
          dialogTitle: "Open layout file",
          type: FileType.custom,
          allowedExtensions: ["json"],
        );
        if (filePickerResult case FilePickerResult(:final
            files)
            when files.isNotEmpty && files.single.path !=
                null) {
          if (mounted) {
            context.read<DashboardCubit>().loadFromFile(
                  filePath: files.single.path!,
```

```
                );
              }
            } else if (mounted) {
              context.read<DashboardCubit>().layoutNotLoaded();
            }
          case DashboardLayoutLoaded(:final labLayout):
            context.read<AppMenuBarCubit>().layoutLoaded();
            context
                .read<BirdsEyeViewCubit>()
                .gotLabLayout(labLayout: labLayout);
            context
                .read<ScreenViewCubit>()
                .gotLabLayout(labLayout: labLayout);
            context
                .read<ReportViewCubit>()
                .gotLabLayout(labLayout: labLayout);
          default:
            context.read<AppMenuBarCubit>().layoutNotLoaded();
            context.read<BirdsEyeViewCubit>().noLabLayout();
            context.read<ScreenViewCubit>().noLabLayout();
            context.read<ReportViewCubit>().noLabLayout();
            break;
        }
      },
      child: const AppMenuBar(
        body: Scaffold(
          body: Row(
            children: [
              NavigationRailSide(),
              VerticalDivider(),
              DashboardViewFrame()
            ],
          ),
        ),
      ),
    ),
  );
 }
}
```

larkclient_coordinator.dart

```
import 'dart:async';
import 'dart:io';
import 'dart:typed_data';
import 'package:async/async.dart';
import 'package:fpdart/fpdart.dart';
```

```
import 'package:larkclient/core/entity/client_name_and_key.dart';
import 'package:larkclient/core/entity/larksmite_server.dart';
import 'package:larkclient/core/logging.dart';
import 'package:larkclient/core/status_types.dart';
import 'package:larkclient/feature/datastore/domain/usecase/
    server_and_client_details.dart';
import 'package:larkclient/feature/networking/domain/usecase/
    manage_server_conn.dart';
import 'package:larkclient/feature/platform_event_handler/domain/
    repository/platform_event_handler.dart';
import 'package:larksmite_shared/larskmite_shared_entites.dart';

class LarkClientCoordinator {
  final ServerAndClientDetails serverAndClientDetails;
  final ManageServerConnection manageServerConnection;
  HttpClient? _httpClient;
  LarkClientCoordinator({
    required this.serverAndClientDetails,
    required this.manageServerConnection,
    PlatformEventHandler? platformEventHandler,
  }) : _platformEventHandlerSupplied = platformEventHandler;
  final PlatformEventHandler? _platformEventHandlerSupplied;
  PlatformEventHandler get platformEventHandler {
    if (_platformEventHandlerSupplied != null) {
      return _platformEventHandlerSupplied!;
    } else {
      throw UnimplementedError(
        "PlatformEventHandler is not implemented for this platform
            ",
      );
    }
  }

  Stream<Either<Failure, Success>> startEverything({
    ClientNameAndKey? clientNameAndKey,
  }) async* {
    try {
      platformEventHandler;
    } on UnimplementedError catch (error, stacktrace) {
      getLogger().severe(
        "Event handler isn't implemented for this platform",
        error,
        stacktrace,
      );
      yield Either.left(
        NoPlatformEventHandlerAvailable(
```

```
      message: "This platform is not supported",
    ),
  );
  return;
} on Exception catch (error, stacktrace) {
  getLogger().severe(
    "Unexpected error while trying to access
        PlatformEventHandler",
    error,
    stacktrace,
  );
  yield Either.left(
    UnknownUnrecoverableFailure(
      message:
          "Unexpected error while checking if this platform is
              supported",
    ),
  );
  return;
}
final handlerStream = handlePlatformSpecificEvents();

// If the passed clientNameAndKey is null, it (hopefully)
    means its not the first time
// But if it's not null, then it means its the first time
ClientNameAndKey clientDetails;
bool isFirstTime;
if (clientNameAndKey != null) {
  clientDetails = clientNameAndKey;
  isFirstTime = true;
} else {
  final clientNameAndKeyResult =
      await serverAndClientDetails.getSavedClientNameAndKey();
  switch (clientNameAndKeyResult) {
    case Right(:final value):
      clientDetails = value;
      isFirstTime = false;
    case Left(:final value):
      getLogger().severe(
        "Failure while trying to retrieve saved client details
            ",
      );
      yield Either.left(value);
      return;
  }
}
```

```
  final connectionResult = connectToServerSecurely(
    clientNameAndKey: clientDetails,
    isFirstTime: isFirstTime,
  );
  yield* StreamGroup.merge([handlerStream, connectionResult]);
  getLogger().config("All streams finished in startEverything");
  return;
}

/// If [retryTimes] is 0, it will retry until we get successful
     result
/// The result of calling [shouldStopImmediately] with any
    encountered failure
/// is used to determine if we should stop immediately, useful
    in cases where
/// [retryTimes] is inifnite but it should stop when a certain
    condition is met
Future<Either<Failure, T>> _retryTask<T>({
  required Future<Either<Failure, T>> Function() task,
  int retryTimes = 0,
  Duration delayBetweenRetries = const Duration(seconds: 3),
  bool Function(Failure failure)? shouldStopImmediately,
}) async {
  bool retryCondition(int iInside, int retryTimesInside) {
    return (iInside < retryTimesInside || retryTimesInside == 0)
        ;
  }

  Failure? lastFailure;
  shouldStopImmediately ??= (_) => false;
  for (int i = 0; retryCondition(i, retryTimes); i++) {
    final result = await task();
    switch (result) {
      case Right(:final value):
        return Either.right(value);
      case Left(:final value):
        if (shouldStopImmediately(value)) {
          return Either.left(value);
        } else {
          lastFailure = value;
        }
    }
    await Future.delayed(delayBetweenRetries);
  }
  lastFailure ??=
```

```
      NoMoreRetries(message: "No more retries available for task
          ");
  return Either.left(lastFailure);
}

/// This will yield succesful event, if it encounters any
    failure, it will yield
/// that Failure and close immediately, its up to the caller to
     handle it.
/// This version of connection method is more try-and-fail type
     than
/// ask-for-permission type, thus simplifying stuff massively
Stream<Either<Failure, Success>> connectToServerSecurely({
  required ClientNameAndKey clientNameAndKey,
  bool isFirstTime = false,
}) async* {
  getLogger().config(
      "This is${isFirstTime ? '' : ' not'} the first time we're
          starting");
  // Search until we get the server
  LarkSmiteServer searchedServer;

  final searchedServerResult = await _retryTask(
    task: () => manageServerConnection.searchAndGetServer(),
    delayBetweenRetries: const Duration(seconds: 1),
    shouldStopImmediately: (failure) => failure is
        ServerSearchCancelled,
  );
  switch (searchedServerResult) {
    case Right(:final value):
      searchedServer = value;
    case Left(:final value):
      // Since retryTimes is 0 for server searching task, it
          should be able to
      // get the server eventually, but if it's unable to do so,
           it must be
      // the cancelled scenario
      getLogger().severe("Searching server failed", value);
      yield Either.left(value);
      return;
  }
  if (!isFirstTime) {
    getLogger()
        .config("It's not first start, so checking server by
            challenge");
    final isServerTrustedResult = await serverDoChallenge(
```

```
    larkSmiteServer: searchedServer,
  );
  bool isServerTrusted;
  switch (isServerTrustedResult) {
    case Right(:final value):
      isServerTrusted = value;
    case Left(:final value):
      yield Either.left(value);
      return;
  }
  if (!isServerTrusted) {
    yield Either.left(
      ServerChallengeFailed(
        message:
            "${searchedServer.ipAddress} gave unsuccessful
                response for challenge, not trusting it.",
      ),
    );
    return;
  } else {
    getLogger().config("Server completed challenge successfuly
        ");
  }
}

// If we're here, it means the server is trustworthy, or its
    the first time
// which case, we assume its trusted
Uint8List certBinary;

final certBinaryResult = await _retryTask(
  task: () => manageServerConnection.downloadCertificate(
    larkSmiteServer: searchedServer,
  ),
  retryTimes: 5,
);
switch (certBinaryResult) {
  case Right(:final value):
    certBinary = value;
  case Left(:final value):
    getLogger().severe("Unable to download certificate", value
        );
    yield Either.left(value);
    return;
}
final securityContext = SecurityContext();
```

```
securityContext.setTrustedCertificatesBytes(certBinary);
final httpClient = HttpClient(context: securityContext);
_httpClient = httpClient;
final connectionStream = manageServerConnection.
    connectToServer(
  clientNameAndKey: clientNameAndKey,
  httpClient: httpClient,
  ipAddress: searchedServer.ipAddress,
  isFirstStartFlag: isFirstTime,
);
yield* connectionStream.transform(
  StreamTransformer.fromHandlers(
    handleData: (data, sink) async {
      switch (data) {
        case Right(value: final connection):
          if (isFirstTime) {
            getLogger().config(
              "Since it's the first time, "
              "saving public key of server and name + key of
                  client",
            );
            // We don't want any stale data
            await serverAndClientDetails.clearAllData();
            await serverAndClientDetails.saveClientNameAndKey(
              clientNameAndKey: clientNameAndKey,
            );
            await serverAndClientDetails.savePublicKey(
              rsaPublicKey: connection.rsaPublicKeyAbstract,
            );
          }
          sink.add(
            Either.right(
              ConnectServerSuccess(message: "Connected to
                  server"),
            ),
          );
        case Left(:final value):
          sink.add(Either.left(value));
      }
    },
    handleDone: (sink) {
      sink.close();
    },
    handleError: (error, stackTrace, sink) {
      getLogger().severe(
        "Unknown error while transforming in
```

```
            connectToServerSecurely",
          error,
          stackTrace,
        );
        sink.close();
      },
    ),
  );
  return;
}

/// This will handle all events from server. Call this before
    connecting so as to
/// not miss any events
Stream<Either<Failure, Success>> handlePlatformSpecificEvents()
    {
  // Returning as broadcast stream since cancelling this will
      interfere with
  // the cleanup process, but if its a broadcast stream, it will
       continue
  // cleaning up and finish naturally
  return platformEventHandler
      .handleEvents(
        serverCommunicatorDuplex:
            manageServerConnection.serverCommunicatorDuplex,
      )
      .asBroadcastStream();
}

Future<Either<Failure, bool>> serverDoChallenge({
  required LarkSmiteServer larkSmiteServer,
}) async {
  final savedPublicKeyResult =
      await serverAndClientDetails.getPreviouslySavedPublicKey()
          ;
  RSAPublicKeyAbstract rsaPublicKey;
  switch (savedPublicKeyResult) {
    case Right(:final value):
      rsaPublicKey = value;
    case Left(:final value):
      return Either.left(value);
  }
  return manageServerConnection.performChallenge(
    larkSmiteServer: larkSmiteServer,
    rsaPublicKey: rsaPublicKey,
  );
```

```
  }

  Future<void> stopEverything() async {
    await manageServerConnection.stopSearching();
    await manageServerConnection.disconnectFromServer();
    _httpClient?.close(force: true);
    await platformEventHandler.stopHandling();
  }
}
```

manage_server_conn.dart

```
import 'dart:async';
import 'dart:io';
import 'dart:typed_data';
import 'package:fpdart/fpdart.dart';
import 'package:larkclient/core/entity/client_name_and_key.dart';
import 'package:larkclient/core/entity/events_from_server/events_
    from_server.dart';
import 'package:larkclient/core/entity/events_to_server/events_to
    _server.dart';
import 'package:larkclient/core/entity/larksmite_server.dart';
import 'package:larkclient/core/entity/server_communicator_duplex
    .dart';
import 'package:larkclient/core/logging.dart';
import 'package:larkclient/core/status_types.dart';
import 'package:larkclient/feature/networking/domain/repository/
    server_repository.dart';
import 'package:larksmite_shared/larksmite_shared.dart';
import 'package:larksmite_shared/larskmite_shared_entites.dart';
import 'package:uuid/v4.dart';

class ManageServerConnection {
  final ServerRepository serverRepository;
  ServerCommunicatorDuplex get serverCommunicatorDuplex =>
      serverRepository.serverCommunicatorDuplex;
  ManageServerConnection({required this.serverRepository});
  Future<Either<Failure, LarkSmiteServer>> searchAndGetServer()
      async {
    return serverRepository.searchForServer();
  }

  Future<void> stopSearching() async {
    await serverRepository.stopSearchingForServer();
  }

  Future<Either<Failure, RecieveType>> _sendAndWaitForEvent<
```

```
    SendType extends EventsToSendToServer,
    RecieveType extends EventsRecievedFromServer>({
  required SendType sendEvent,
  Duration timeoutDuration = const Duration(seconds: 10),
}) async {
  final result = await sendAndWaitForEvent<SendType, RecieveType
      ,
      ServerErrorResponse, EventsToSendToServer,
          EventsRecievedFromServer>(
    sendEvent: sendEvent,
    sendSink: serverCommunicatorDuplex.eventsToSendToServer,
    recieveStream: serverCommunicatorDuplex.
        eventsRecievedFromServer,
    timeoutDuration: timeoutDuration,
  );
  return result;
}

Future<Either<Failure, ConnectionResponse>> _getConnectResult({
  required ClientNameAndKey clientNameAndKey,
  required bool isFirstStartFlag,
}) async {
  return _sendAndWaitForEvent<ConnectionRequest,
      ConnectionResponse>(
    sendEvent: ConnectionRequest.generate(
      clientNameAndKey: clientNameAndKey,
      isFirstTime: isFirstStartFlag,
    ),
    timeoutDuration: const Duration(seconds: 15),
  );
}

Stream<Either<Failure, ConnectionAccepted>> connectToServer({
  required ClientNameAndKey clientNameAndKey,
  required HttpClient httpClient,
  required InternetAddress ipAddress,
  required bool isFirstStartFlag,
}) {
  final connectToServerStream =
      serverRepository.connectToServerAndListenToEvents(
    httpClient: httpClient,
    ipAddress: ipAddress,
  );
  return connectToServerStream.transform(
    StreamTransformer.fromHandlers(
      handleData: (data, sink) async {
```

```
switch (data) {
  case Right():
    final connectionResult = await _getConnectResult(
      clientNameAndKey: clientNameAndKey,
      isFirstStartFlag: isFirstStartFlag,
    );
    switch (connectionResult) {
      case Right(value: ConnectionAccepted()):
        sink.add(Either.right(
            connectionResult.value as ConnectionAccepted)
                );
      case Right(value: ConnectionDeclined()):
        // Its Right here since, the connection was
            successful and the event was
        // recieved correctly
        sink.add(
          Either.left(
            ClientDetailsAreWrong(
              message:
                  "Server declined connection because
                      client details are wrong",
            ),
          ),
        );
      // Changing the types of [ErrorResponseFailure] and
          [TimeoutFailure]
      // from larksmite_shared to our own failure types
          which extends
      // [RecoverableFailure]
      case Left(
          value: ErrorResponseFailure(
            :final message,
            errorResponse: ServerErrorResponse(),
          )
        ):
        sink.add(Either.left(ServerErrorFailure(message:
            message)));
      case Left(value: TimeoutFailure(:final message)):
        sink.add(Either.left(ServerTimeoutFailure(message
            : message)));
      case Left(:final value):
        sink.add(Either.left(value));
    }
  case Left(:final value):
    sink.add(Either.left(value));
}
```

```
      },
      handleDone: (sink) {
        sink.close();
      },
      handleError: (error, stackTrace, sink) {
        getLogger().severe(
          "Error while transforming events in connectToServer",
          error,
          stackTrace,
        );
        sink.close();
      },
    ),
  );
}

Future<Either<Failure, Uint8List>> downloadCertificate({
  required LarkSmiteServer larkSmiteServer,
}) async {
  return serverRepository.downloadCertificate(
      larkSmiteServer: larkSmiteServer);
}

Future<Either<Failure, bool>> performChallenge({
  required LarkSmiteServer larkSmiteServer,
  required RSAPublicKeyAbstract rsaPublicKey,
}) async {
  final randomString = const Uuid V4().generate();
  final encryptedData = await LarksmiteCryptoRsaPointyCastle().
      encryptString(
    rsaPublicKey: rsaPublicKey as RSAPublicKeyPointyCastle,
    string: randomString,
  );

  final challengeResponse = await serverRepository.
      performChallenge(
    larkSmiteServer: larkSmiteServer,
    serverDoChallengeDecrypt: ServerDoChallengeDecrypt(
      encryptedData: encryptedData,
    ),
  );
  switch (challengeResponse) {
    case Left(value: ServerChallengeFailed()):
      return Either.right(false);
    case Left(:final value):
      return Either.left(value);
```

```
      case Right(:final value):
        // Simple equality check, the real check is if the server
            was able to perform
        // the decryption and send the plain text back
        return Either.right(
          value.decryptedString == randomString,
        );
    }
  }

  Future<Either<Failure, Success>> disconnectFromServer() async {
    return serverRepository.disconnectFromServer();
  }
}
```

linux_event_handler.dart

```
import 'dart:async';
import 'dart:io';
import 'package:fpdart/fpdart.dart';
import 'package:gnome_screenrecord/gnome_screenrecord.dart';
import 'package:larkclient/core/constants.dart';
import 'package:larkclient/core/entity/events_from_server/events_
    from_server.dart';
import 'package:larkclient/core/entity/events_to_server/events_to
    _server.dart';
import 'package:larkclient/core/entity/server_communicator_duplex
    .dart';
import 'package:larkclient/core/logging.dart';
import 'package:larkclient/core/status_types.dart';
import 'package:larkclient/core/useful_helpers.dart';
import 'package:larkclient/feature/platform_event_handler/data/
    datasource/linux_proc_source.dart';
import 'package:larkclient/feature/platform_event_handler/data/
    datasource/linux_run_bash_source.dart';
import 'package:larkclient/feature/platform_event_handler/domain/
    entity/screencast_object.dart';
import 'package:larkclient/feature/platform_event_handler/domain/
    repository/platform_event_handler.dart';
import 'package:larksmite_shared/larskmite_shared_entites.dart';
import 'package:process_monitor/process_monitor.dart';

class LinuxPlatformEventHandler implements PlatformEventHandler {
  final LinuxProcSource linuxProcSource;
  final LinuxRunBashSource linuxRunBashSource;
  ScreencastObj? _screencastObj;
  StreamController<Either<Failure, Success>>? _eventStatusStream;
```

```
LinuxPlatformEventHandler({
  required this.linuxProcSource,
  required this.linuxRunBashSource,
});
Future<ProcessModel> getRootProcess() {
  getLogger().config("Trying to get processes");
  final processMonitor = ProcessMonitor.instance;
  return processMonitor.getProcessTreeRootNode();
}

Future<ClientInfo> getClientInfo() async {
  getLogger().config("Trying to get client info");
  return ClientInfo(
    hostname: Platform.localHostname,
    uptime: await linuxProcSource.getUptime(),
    cpuUsage: await linuxProcSource.getCpuUsage(),
  );
}

Future<void> showMessageGui({
  required String message,
  required MessageType messageType,
}) async {
  getLogger().config("Showing GUI message to user");
  await linuxRunBashSource.showMessageGui(
    message: message,
    messageType: messageType,
  );
}

Future<Uri> getScreencastLink({
  required ScreenCastResolution screenCastResolution,
  required InternetAddress clientIp,
}) async {
  //TODO P3: find a way to make this less kludgy
  getLogger().config("Beginning screencast procedure");
  if (_screencastObj == null) {
    getLogger().config("Getting screen resolution");
    final gnomeRecorder = GnomeScreenRecorder();
    final res = await linuxRunBashSource.getScreenResolution();
    final int width, height;
    width = ((screenCastResolution.resolutionPercentage / 100) *
         res.width)
        .toInt();
    height = ((screenCastResolution.resolutionPercentage / 100)
```

```
    * res.height)
    .toInt();
Stream<DbusRecordStatus> screencastStatusStream;
getLogger().config("About to run gnomeScreenRecord");

screencastStatusStream = gnomeRecorder.startRecording(
  height: height,
  width: width,
  internetAddress: clientIp,
  port: screencastPort,
);

final streamLinkToSend = Completer<Uri>();
final screencastListener = screencastStatusStream.listen(
  // Fortunately we don't need to cancel this stream since
      it finishes when
  // the recording stops, but we should probably find a way
      to make it cleaner
  (event) {
    getLogger().config("Got event $event from gnome screen
        record");
    switch (event) {
      case DbusRecordStarted(
          :final streamLink,
          :final dbusMethodResponse,
        ):
        getLogger().config(
          "Dbus start screen record response: $
              dbusMethodResponse",
        );
        _screencastObj = ScreencastObj(
          streamLinkToSend: streamLink,
          screenRecorder: gnomeRecorder,
        );
        streamLinkToSend.complete(streamLink);
      case DbusRecordWaiting():
        getLogger().config("Screencasting... Waiting for stop
            signal");
      case DbusRecordEnded(:final dbusMethodResponse):
        getLogger().config(
          "Dbus stop screen record response: $
              dbusMethodResponse",
        );
      case DbusError(:final dBusMethodResponseException):
        getLogger().warning(
            "Unable to start recording, response: $
```

```
                dBusMethodResponseException");
            streamLinkToSend.completeError(ScreencastException())
                ;
        }
      },
      cancelOnError: true,
    );
    await streamLinkToSend.future;
    // await screencastListen.cancel();
  } else {
    getLogger()
        .config("Skipping some steps since screencast is already
            running");
  }

  final screencastLink = Uri.parse(
    "tcp://${clientIp.address}:$screencastPort",
  );

  getLogger().config("Sending screen cast link: $screencastLink
      ");
  return screencastLink;
}

Future<void> stopScreencast() async {
  getLogger().config("Stopping screencast");
  if (_screencastObj != null) {
    await _screencastObj!.screenRecorder.stopRecording();
    _screencastObj = null;
  }
  getLogger().config("Stopped screencast");
}

Future<void> handlePowerSignal({
  required PowerSignal powerSignal,
}) async {
  await Future.delayed(powerSignal.performSignalAfter);
  switch (powerSignal.powerSignalType) {
    case PowerSignalType.lockScreen:
      await linuxRunBashSource.lockScreen();
    case PowerSignalType.sleep:
      await linuxRunBashSource.sleepComputer();
    case PowerSignalType.restart:
      await linuxRunBashSource.restartComputer();
    case PowerSignalType.powerOff:
      await linuxRunBashSource.shutdownComputer();
```

```
  }
}

Future<List<int>> getScreenshot() {
  return linuxRunBashSource.getScreenshot();
}

@override
Stream<Either<Failure, Success>> handleEvents({
  required ServerCommunicatorDuplex serverCommunicatorDuplex,
}) async* {
  getLogger().config("Starting the Linux event handler");
  _eventStatusStream = StreamController();

  final serverEventListener =
      serverCommunicatorDuplex.eventsRecievedFromServer.listen(
    (event) async {
      switch (event) {
        case ClientProcessesRequired(:final token):
          try {
            final rootProcess = await getRootProcess();
            serverCommunicatorDuplex.eventsToSendToServer.add(
              ClientProcessesResponse.generateResponseTo(
                responseToToken: token,
                rootProcess: rootProcess,
              ),
            );
            _eventStatusStream?.add(
              Either.right(
                EventHandledSuccessfuly(
                  message: "Handled get process request
                      successfuly",
                  eventToken: token,
                ),
              ),
            );
          } on Exception catch (error, stacktrace) {
            getLogger().warning(
                "Exception while handling ClientProcessRequired",
                error,
                stacktrace);
            serverCommunicatorDuplex.eventsToSendToServer.add(
              ClientErrorResponse.generateResponseTo(
                responseToToken: token,
                message: "Unable to get processes",
                errorString: error.toString(),
```

```
          stacktraceString: stacktrace.toString(),
        ),
      );
      _eventStatusStream?.add(
        Either.left(
          EventHandleFailed(
            message: "Handling of GetProcesses failed",
            eventToken: token,
          ),
        ),
      );
    }
  case ClientInfoRequired(:final token):
    try {
      final clientInfo = await getClientInfo();
      serverCommunicatorDuplex.eventsToSendToServer.add(
        ClientInfoResponse.generateResponseTo(
          responseToToken: token,
          clientInfo: clientInfo,
        ),
      );
      _eventStatusStream?.add(
        Either.right(
          EventHandledSuccessfuly(
            message: "Handled get client info successfuly",
            eventToken: token,
          ),
        ),
      );
    } on Exception catch (error, stacktrace) {
      getLogger().warning("Exception while handling
          ClientInfoRequired",
          error, stacktrace);
      serverCommunicatorDuplex.eventsToSendToServer.add(
        ClientErrorResponse.generateResponseTo(
          responseToToken: token,
          message: "Unable to get info",
          errorString: error.toString(),
          stacktraceString: stacktrace.toString(),
        ),
      );
      _eventStatusStream?.add(
        Either.left(
          EventHandleFailed(
            message: "Handling of GetInfo failed",
            eventToken: token,
```

```
        ),
      ),
    );
  }
case ClientStatusRequired(:final token):
  serverCommunicatorDuplex.eventsToSendToServer.add(
    ClientStatusResponse.generateResponseTo(
      responseToToken: token,
      clientStatus: ClientStatus.ok,
    ),
  );
  _eventStatusStream?.add(
    Either.right(
      EventHandledSuccessfuly(
        message: "Handled get client status successfuly",
        eventToken: token,
      ),
    ),
  );
case PowerSignalRecieved(:final token, :final
    powerSignal):
  try {
    handlePowerSignal(powerSignal: powerSignal);
    serverCommunicatorDuplex.eventsToSendToServer.add(
      ClientAcknowledgeResponse.generateResponseTo(
        responseToToken: token,
      ),
    );
    _eventStatusStream?.add(
      Either.right(
        EventHandledSuccessfuly(
          message: "Handled power signal successfuly",
          eventToken: token,
        ),
      ),
    );
  } on Exception catch (error, stacktrace) {
    getLogger().warning(
        "Exception while handling PowerSignalRecieved",
        error,
        stacktrace);
    serverCommunicatorDuplex.eventsToSendToServer.add(
      ClientErrorResponse.generateResponseTo(
        responseToToken: token,
        message: "Unable to handle power signal",
        errorString: error.toString(),
```

```
          stacktraceString: stacktrace.toString(),
        ),
      );
      _eventStatusStream?.add(
        Either.left(
          EventHandleFailed(
            message: "Handling of power signal failed",
            eventToken: token,
          ),
        ),
      );
    }
  case ClientScreencastRequired(
      :final token,
      :final screenCastResolution,
      :final ipAddressOfClient,
    ):
    try {
      final screencastLink = await getScreencastLink(
        screenCastResolution: screenCastResolution,
        clientIp: ipAddressOfClient,
      );
      serverCommunicatorDuplex.eventsToSendToServer.add(
        ClientScreencastResponse.generateResponseTo(
          responseToToken: token,
          screencastUri: screencastLink,
        ),
      );
      _eventStatusStream?.add(
        Either.right(
          EventHandledSuccessfuly(
            message: "Handled get screencast successfuly",
            eventToken: token,
          ),
        ),
      );
    } on Exception catch (error, stacktrace) {
      getLogger().warning(
          "Exception while handling
              ClientScreencastRequired",
          error,
          stacktrace);
      serverCommunicatorDuplex.eventsToSendToServer.add(
        ClientErrorResponse.generateResponseTo(
          responseToToken: token,
          message: "Unable to get screencast",
```

```
          errorString: error.toString(),
          stacktraceString: stacktrace.toString(),
        ),
      );
      _eventStatusStream?.add(
        Either.left(
          EventHandleFailed(
            message: "Handling of screen cast failed",
            eventToken: token,
          ),
        ),
      );
    }
  case ClientStopScreencast(
      :final token,
    ):
    try {
      await stopScreencast();
      serverCommunicatorDuplex.eventsToSendToServer.add(
        ClientAcknowledgeResponse.generateResponseTo(
          responseToToken: token,
        ),
      );
      _eventStatusStream?.add(
        Either.right(
          EventHandledSuccessfuly(
            message: "Handled stop screencast successfuly",
            eventToken: token,
          ),
        ),
      );
    } on Exception catch (error, stacktrace) {
      getLogger().warning(
          "Exception while handling ClientStopScreencast",
          error,
          stacktrace);
      serverCommunicatorDuplex.eventsToSendToServer.add(
        ClientErrorResponse.generateResponseTo(
          responseToToken: token,
          message: "Unable to stop screencast",
          errorString: error.toString(),
          stacktraceString: stacktrace.toString(),
        ),
      );
      _eventStatusStream?.add(
        Either.left(
```

```
        EventHandleFailed(
          message: "Handling of stop screen cast failed",
          eventToken: token,
        ),
      ),
    );
  }
case ClientScreenshotRequired(
    :final token,
  ):
  try {
    final photoBinary = await getScreenshot();
    serverCommunicatorDuplex.eventsToSendToServer.add(
      ClientScreenshotResponse.generateResponseTo(
        responseToToken: token,
        photoBinary: photoBinary.toList(),
      ),
    );
    _eventStatusStream?.add(
      Either.right(
        EventHandledSuccessfuly(
          message: "Handled get screenshot successfuly",
          eventToken: token,
        ),
      ),
    );
  } on Exception catch (error, stacktrace) {
    getLogger().warning(
        "Exception while handling ClientGetScreenshot",
        error,
        stacktrace);
    serverCommunicatorDuplex.eventsToSendToServer.add(
      ClientErrorResponse.generateResponseTo(
        responseToToken: token,
        message: "Unable to get screenshot",
        errorString: error.toString(),
        stacktraceString: stacktrace.toString(),
      ),
    );
    _eventStatusStream?.add(
      Either.left(
        EventHandleFailed(
          message: "Handling of get screenshot failed",
          eventToken: token,
        ),
      ),
```

```
    );
  }
case ClientShowMessage(
    :final token,
    :final message,
    :final messageType,
  ):
  try {
    await showMessageGui(message: message, messageType:
        messageType);
    serverCommunicatorDuplex.eventsToSendToServer.add(
      ClientAcknowledgeResponse.generateResponseTo(
        responseToToken: token,
      ),
    );
    _eventStatusStream?.add(
      Either.right(
        EventHandledSuccessfuly(
          message: "Handled show message successfuly",
          eventToken: token,
        ),
      ),
    );
  } on Exception catch (error, stacktrace) {
    getLogger().warning("Exception while handling
        ClientShowMessage",
        error, stacktrace);
    serverCommunicatorDuplex.eventsToSendToServer.add(
      ClientErrorResponse.generateResponseTo(
        responseToToken: token,
        message: "Unable to show message",
        errorString: error.toString(),
        stacktraceString: stacktrace.toString(),
      ),
    );
    _eventStatusStream?.add(
      Either.left(
        EventHandleFailed(
          message: "Handling of showing message failed",
          eventToken: token,
        ),
      ),
    );
  }
case ServerAdvertisementRecieved():
case ConnectionAccepted():
```

```
      case ConnectionDeclined():
      case ServerErrorResponse():
        // No need to handle them here, since they're platform
            independent events
        break;
    }
  },
  cancelOnError: true,
);

serverEventListener.onDone(
  () async {
    getLogger().config(
        "Finished listening to events from server in Linux
            event handler");
    _eventStatusStream?.addIfNotClosed(
      Either.left(
        EventsFromServerFinished(
          message: "Finished listening to events in Linux event
              handler",
        ),
      ),
    );
    await _eventStatusStream?.close();
  },
);

serverEventListener.onError(
  (error, stacktrace) async {
    getLogger().severe(
      "Server event listener had an error",
      error,
      stacktrace,
    );
    _eventStatusStream?.addIfNotClosed(
      Either.left(
        EventsFromServerFinished(
          message:
              "Events from server ended with error in Linux
                  event handler",
        ),
      ),
    );
    await _eventStatusStream?.close();
  },
);
```

```
    // Will keep running until it's closed by the top two factors
        or external stop
    getLogger().config("Going to start listening to
        eventStatusStream");
    yield* _eventStatusStream!.stream;
    getLogger().config("Cleaning up Linux event handler");
    await stopScreencast();
    await serverEventListener.cancel();
    _eventStatusStream = null;
    getLogger().config("Finished cleaning up Linux event handler")
        ;
    return;
  }

  @override
  Future<void> stopHandling() async {
    getLogger().config("Stopping linux platform handler");
    await stopScreencast();
    _eventStatusStream?.addIfNotClosed(
      Either.left(
        EventHandlerCancelled(
          message: "Event handler stopped",
        ),
      ),
    );
    getLogger().config("Closing the eventStatus stream");
    await _eventStatusStream?.sink.close();
    await _eventStatusStream?.close();
  }
}
```

DEPARTMENT OF COMPUTER SCIENCE

CHRIST COLLEGE (AUTONOMOUS)

IRINJALAKUDA

CERTIFICATE

This is to certify that the project report entitled ***"BREAST CANCER PREDICTION"*** *is a bonafide record of the project work done by* ***ADITHYAN V S*** *in partial fulfillment of the requirement for the sixth semester of* ***Bachelor of Computer Application*** *in Department of Computer Science of* ***CHRIST COLLEGE (AUTONOMOUS) IRINJALAKUDA***

.

Mr.Joju Sebastian
Assistant Professor,CS
Internal Guide

Ms. Sini Thomas
Head of the Department
Computer Science

EXTERNAL EXAMINER

INTERNAL EXAMINER

BREAST CANCER PREDICTION

PROJECT REPORT

Submitted By

ADITHYAN V S

Reg. No. CCAVBCA018

For the award of the Degree of

Bachelor of Computer Application

(BCA)

(University of Calicut)

under the guidance of

Mr. Joju Sebastian

Assistant Professor

BACHELOR OF COMPUTER APPLICATION
DEPARTMENT OF COMPUTER SCIENCE
CHRIST COLLEGE
IRINJALAKUDA, KERALA

2021-2024

DECLARATION

We here by declare that this project work **"BREAST CANCER PREDICTION"** submitted by Christ College (Autonomous)Irinjalakuda,affliated to Calicut University in partial fullfillment of the requirement for the award of the Bachelor of Computer Application, is a record of original work done by us,under the guidance of Mr.JOJU SEBASTIAN,Department of Computer Science.

Place: Irinjalakuda ADITHYAN V S

ACKNOWLEDGEMENT

First and foremost we like to thank Lord almighty for his providence and for being the guiding light throughout the project. We wish to express my sincere gratitude to our beloved Department head for giving me all the facilities for our project. We take this opportunity to express my gratitude to the class teacher Ms. SOWMYA PS and head of the department Ms. SINI THOMAS who has been supported us throughout the course of this project. We are thankful for her aspiring guidance and valuable advice during the project work. We express my sincere thanks to my project guide Mr.JOJU SEBASTIAN for supporting and guiding throughout the project.We would take this opportunity to specially thank all other faculty members for their constant and continuous motivation. Finally we would like to thank my family and friends for giving valuable advice and moral support throughout our project.

ABSTRACT

BREAST CANCER PREDICTION is a desktop application that uses CNN algorithm to identify cancer and non cancerous cells in medica lorganization. This system automates the process of predicting cancer by inputting histopathological images and analyzing them to identify cancer and non cancer cells.

Contents

Chapter 1

1 Introduction

"Breast Cancer Prediction" introduces a revolutionary solution for predicting breast cancer, replacing traditional methods. This system incorporates deep learning algorithms to enhance the entire prediction process. The advantages of this project includes heightened accuracy, time efficiency, improved security, and reduced costs. As medical organizations embrace this user-friendly and innovative system, they are propelled into a more streamlined and technologically advanced future of breast cancer prediction.

.

1.1 Overview

The objective of the "Breast Cancer Prediction" project is to develop and implement a cutting-edge solution that automates breast cancer prediction using CNN algorithm. The goal is to replace traditional prediction methods with a more accurate, efficient, and secure solution, ultimately improving outcomes for patients and medical professionals.

Chapter 2

2 System Analysis

2.1 Purpose

Breast Cancer Prediction is a desktop applications that uses CNN algorithm to identify cancer and non-cancer cells in medical organizations. The system automates the process of predicting cancer by inputing histopathological images and analysing them to identify cancer and non-cancer cells. The system is typically composed of a software. The histopathological image of individuals browsed from the system are taken and send to the software, upon clicking predict button the image is analysed and compared with the trained dataset. The dataset contains information about each person's multiple histopathological images cells. The weights of the images are recorded and based on the weights of the new images the result is predicted.

2.1.1 Existing System

The existing system of the breast cancer prediction primarily relies on traditional prediction methods, such as using machine learning algorithms which predicts optimally the cancer and non-cancer cells from the trained histopathological dataset. Here, when a new histopathological image is taken the result accuracy is poor.

2.1.2 Proposed System

Breast Cancer Prediction is a desktop application that is in need for breast cancer identification on lab. Here, Project wise there is a GUI created that takes an image as input and classifies that it's a cancerous or non-cancerous. In this system when a new histopathological image is given here the result is predicted optimally and accuracy is good due to the use of CNN algorithm. Medically speaking Biopsy is taken place here .The histopathological images (Histology – Study of tissues , Pathology – Study of diseases) are taken from the monitor using advanced electron microscope by selecting a sample tissue and then processed and cut into thin sections then stained with Hematoxylin – Eosin (H&E).

2.2 Problem definition

The proposed project aims to tackle the limitation and inefficiency of traditional prediction methods by developing "Breast Cancer Prediction". The existing system predicts the cancer using a machine learning algorithm which only predicts the result based on the image given as input is the trained histopathological images which often lead to inaccuracies. To address these issues, this project will leverage deep learning method to provide a more accurate, secure, and

user-friendly approach to breast cancer prediction. By integrating seamlessly with existing systems and adapting to varying environmental conditions, this system promises to revolutionize breast cancer prediction, delivering enhanced accuracy, efficiency, and security for medical organizations.

2.3 FEASIBILITY STUDY

After the problem is clearly understood and solutions proposed, the next step is to conduct the feasibility study. Feasibility study is defined as evaluation or analysis of the potential impact of the proposed project or program. The objective is to determine whether the proposed system is feasible. There are three aspects of feasibility study to which the proposed system is subjected as discussed below.

2.3.1 Technical Feasibility

The proposed breast cancer prediction system using CNN algorithms is technically feasible with all required tools readily available. The system, implemented in Python, leverages CNN for its speed, efficiency, and low resource requirements. However, the accuracy of these algorithms depends on the number of trained images, necessitating thorough testing and optimization. The system requires a well-curated database of histopathological images, with regular updates and management crucial for accurate prediction. Robust measures for data protection and compliance with privacy regulations are essential due to the sensitive nature of the data involved. In conclusion, the technical feasibility of the system is evident, and with proper planning, testing, and attention to privacy and security, it can be a viable solution for various settings, enhancing operational efficiency.

2.3.2 Economical Feasibility

In this proposed system, all the tools used are free and open source. So that development cost is minimum. The hardware is used to run the system. It is built in our laptop. And also hardware requirement is feasible and not much breast cancer prediction system maintenance is required. The development of the system will not need a huge amount of money. It will be economically feasible. And the money spend for the application will be worth. Hence, the proposed system is economically feasible.

2.3.3 Operational Feasibility

The feasibility of a breast cancer prediction system using CNN algorithms is crucial.CNN's ease of implementation reduces development time and costs, and its low resource requirements allow deployment on various hardware configurations. The system requires a well-maintained dataset of pre-registered faces, regular updates, and management to accommodate user base changes. Addressing privacy and security concerns is critical due to the sensitive nature of

biometric data. User acceptance is a significant factor for successful implementation, requiring effective communication of benefits, addressing concerns, and providing adequate training. With proper planning, thorough testing, and compliance with privacy regulations, a breast cancer prediction system using CNN can predict optimal breast cancer results and enhance operational efficiency.

Chapter 3

3 Software Requirement Specification

3.1 Purpose

The breast cancer diagnosis system aims to leverage deep learning algorithms to accurately analyze histopathological images and automatically predict whether cancer is present. Key goals are achieving real-time cancer screening with minimal errors, extracting insightful image features, training robust models CNN, and obtaining analytics to uncover patterns in cancer data - ultimately enhancing efficiency, reliability and research capabilities compared to manual diagnosis processes. The system requires histopathology image datasets to train models to classify images as normal or malignant based on texture, shape and cell characteristics. It will provide a practical tool set to augment and automate parts of the breast cancer screening process.

3.2 Scope

The scope of this project encompasses the development of a software-based solution for automated breast cancer prediction from histopathological images. It leverages deep learning techniques to analyze cell structure and identify malignant indicators.

3.3 Overall Description

The desktop application caters to medical professionals, offering streamlined analysis of digitized histopathology slide images. Convolutional Neural Networks, it assesses cell morphology, texture, shape, and spatial characteristics to classify images as malignant or benign with confidence scores. The software guides users through preprocessing, feature extraction, model execution, and post-processing steps. Results, metrics, and visual heatmaps highlighting malignant regions are easily accessible. The back-end architecture includes a database, modeling module, processing pipelines, integration logic, and access control layers. Through rigorous training and testing with labeled datasets, the system ensures accuracy, aiding medical decision-making and enhancing patient care.

3.3.1 Product Perspective

The breast cancer prediction system aims to integrate smoothly with current medical systems, empowering healthcare professionals with accurate diagnostics. It aligns with organizational goals of enhancing patient care and operational efficiency, offering a user-friendly interface and scalability to adapt to diverse healthcare settings and technological advances.

3.3.2 Product Functionality

The breast cancer prediction system functions by analyzing histopathological images using CNN algorithm to classify cancerous and non-cancerous cells. Users input images into the system, which then processes them, compares them with a trained dataset, and generates predictions with high accuracy. The system offers a user-friendly interface for seamless interaction, contributes to faster and more accurate cancer detection, and integrates with existing medical infrastructure to enhance overall diagnostic capabilities.

3.3.3 Users and Characteristics

The breast cancer prediction system caters primarily to healthcare professionals involved in cancer diagnosis, such as doctors, radiologists, and pathologists. These users typically possess medical expertise and are familiar with diagnostic processes. They require a system that is intuitive, reliable, and capable of providing accurate predictions to support their clinical decision-making. Additionally, administrators responsible for implementing and maintaining the system within healthcare organizations may also interact with it, requiring a broader understanding of its functionality and integration capabilities.

3.4 Specific Requirements

3.4.1 Hardware Requirements

- System: IBM-Compatible PC
- Processor: Intel Core i3 or above
- Speed: Above 1GHz
- RAM capacity: 4 GB
- Hard Dsk drive: 500 GB
- Keyboard: Standard
- Mouse: Standard
- Monitor: SVGA Color

3.4.2 Software Requirements

- Front End:Python
- Back End: Python
- IDE: Anaconda

3.5 Functional Requirements

It contains three main modules.

- 1.Automated Cancer Prediction
- 2.Real-Time Prediction
- 3.User Interface

Automated Cancer Prediction

The system should automate the process of predicting breast cancer using CNN algorithms based on input histopathological images.

Real-Time Prediction

The system should offer real-time breast cancer prediction capabilities, enabling medical professionals to obtain results promptly.

User Interface

The user interface should be user-friendly and intuitive, allowing medical professionals to input histopathological images easily and view predicted results.

3.6 Non Functional Requirements

Non-functional requirements define the overall qualities or attributes of the resulting system. Non-functional system place restrictions on the product being developed, the developed process, and specify external constraints that the product must meet. Examples of non-functional requirements include safety, security, usability, reliability and performance requirements. Project management issues (costs, time and schedule) are often considered as non-functional requirements. The principal non - functional constraints which are relevant to critical systems :

- performance
- security
- safety
- usability

Performance

Performance requirements concern the speed of operation of a system. Types of performance requirements :

- Response requirements (how quickly the system reacts to a user input).

- Throughput requirements (how much the system can accomplish within a specified amount of time).
- Availability requirements (is the system available for service when requested by end users). The speed of operation of this system is adequate for the requirements.

Reliability

- Reliability is the ability of a system to perform its required function under stated conditions for a specified period of time.
- constraints on the runtime behavior of the system. This system is reliable because its functionalities can be done on the required conditions.

Safety

Safety requirements are not required which exclude unsafe situation from the possible solution of the system.

Usability

Usability is the ease with which a user can learn to operate, prepare inputs for, and interpret outputs of system or components. Usability requirements include :

- information error messages.
- well-formed user interfaces.

3.7 Interface Requirements

3.7.1 Hardware interfaces

The system must run over the internet, all the hardware shall require to connect internet will be hardware interface for the system. As for example modem, WAN-LAN.

3.7.2 Software interfaces

Software interface required for the working of the project is the appropriate operating system.

3.7.3 Communication interfaces

The user of the site communicate using the network connectivity and the data set is accessed

3.8 Security Requirements

- User accesses only their account.
- Validation of input is handled.
- This application containing the computer systems is physically secured against arms or surreptitious entry by intruders.
- Users must be authorized carefully to reduce changes of any such user giving access to an intruder in exchange for a bribe or other favour.

3.9 Platform Used

Windows 11 is the operating system chosen for the "Breast Cancer Prediction" project. Windows 10 provides a user-friendly interface and robust support for various hardware components and software applications. It is widely used in both personal and professional settings, making it a suitable choice for medical organizations implementing the breast cancer prediction system. Additionally, Windows 11 offers security features and regular updates to ensure system stability and reliability.

3.10 Technologies Used

PYTHON

Python is a versatile, high-level programming language known for its simplicity and readability. It offers dynamic typing and automatic memory management, making it suitable for various domains such as web development, data analysis, artificial intelligence, and automation. Python's extensive standard library provides pre-built modules and packages for tasks ranging from file manipulation to network programming. Its interpreted nature allows for rapid development and easy debugging. Python supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Its popularity is fueled by a vibrant community, extensive documentation, and numerous third-party libraries. Overall, Python's simplicity, flexibility, and robust ecosystem make it an ideal choice for beginners and seasoned developers alike

CNN

Convolutional Neural Networks (CNNs) are a class of deep neural networks primarily used for image recognition and computer vision tasks. They consist of multiple layers, including convolutional layers, pooling layers, and fully connected layers. CNNs excel at learning hierarchical representations of features from input data. Convolutional layers apply filters to input images, extracting features such as edges and textures. Pooling layers reduce spatial dimensions while preserving important features. Fully connected layers combine extracted features for classification or regression tasks. CNNs leverage parameter sharing and local connectivity to efficiently learn patterns from large datasets. Their success in image recognition tasks has made them a cornerstone in various fields, including autonomous driving, medical imaging, and facial recognition

Chapter 4

4 Design Document

4.1 Purpose

The purpose of the Design Document for the "Breast Cancer Prediction" project is to provide a concise yet comprehensive overview of the system's architecture, components, functionalities, and interactions. It serves as a roadmap for development, ensuring alignment with project goals and requirements. Additionally, the document facilitates communication and collaboration among team members and stakeholders, guiding informed decision-making and ensuring the successful development and implementation of the breast cancer prediction system.

4.2 Scope

The scope of the "Breast Cancer Prediction" project involves developing an automated breast cancer prediction system using deep learning algorithms. This includes designing a user-friendly interface for inputting histopathological images and implementing backend logic for image processing and prediction generation. Additionally, the project encompasses addressing data security and privacy concerns to ensure compliance with medical regulations and standards. Ultimately, the goal is to deliver a reliable and efficient system that aids medical professionals in accurately predicting breast cancer outcomes.

4.3 Overview

The Design Document for the "Breast Cancer Prediction" project provides an overview of the system architecture, functionalities, and key design considerations. It outlines the scope of the project, which involves developing an automated breast cancer prediction system using and deep learning algorithms. The document addresses the purpose of the project, which is to create a reliable and efficient system that aids medical professionals in accurately predicting breast cancer outcomes. Additionally, it highlights the importance of data security and privacy, as well as adherence to ethical and legal standards. Overall, the Design Document serves as a comprehensive blueprint for the development and implementation of the breast cancer prediction system.

4.4 Data Design

Database are the storehouses of data used in the software systems.The data is stored in tables inside the database.Several tables are created for the manipulation of the data for the system.Two essential settings for a database are

- Primary Key- The field that is unique for all the record occurrence.

- Foreign Key- The field used to set relation between tables.

Normalization

- Normalization is the process of organizing the data in the database.
- Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate the undesirable characteristics like Insertion, Update and Deletion Anomalies.
- Normalization divides the larger table into the smaller table and links them using relationship.
- The normal form is used to reduce redundancy from the database table.

Chapter 5

5 Development of the System

The "Breast Cancer prediction" system was developed using an agile approach, incorporating deep learning algorithms to provide an accurate and user-friendly solution for breast cancer prediction in medical organizations, replacing traditional prediction methods.

Chapter 6

6 System Testing

Testing is the penultimate step of software development.An elaborate testing of the data is prepared and the system is using the test data,while doing testing,errors are noted and correction is made.The users are trained to operate the developed system.Both hardware and software securities are made to run the developed system successfully.System testing is aimed at ensuring the system works accurately before the live operation commences.

6.1 Test Plan

6.1.1 Scope

This test plan will cover the following testing activities as identified in the testing strategy.

- White Box Testing

White box testing is a testing technique, that examines the program structure and derives test data from the program logic/code. In white box testing, the UI is bypassed. Inputs and outputs of ratings are tested directly at the code level and the results are compared against specifications. It Reveals errors in "hidden" code.

- Black Box Testing

Black - box testing is a method of software testing that examines the functionality of an application based on the specifications. Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end - user would.

- Unit Testing

The module includes four main components to undergo unit testing. It checks the syntax error, logic error and validity of the program correctness. This test will be performed by the developers.

- Integration Testing

After all components pass the unit testing, they get to be tested whether they work correctly when they are running concurrently and communicating to each other. Integration tests exercise an entire subsystem and ensure that a set of components play nicely together.

- Internal Data Testing

We will test the validity of the data before it enters the database to avoid any problems that may face us in the database. We will test the encryption of the personal information of all the users along with the user names and passwords to ensure maximum security of the users privacy.

6.1.2 Software risk issues

The "Breast Cancer Prediction" system leverages complex deep learning algorithms to predict breast cancer from histopathological images. While this provides enhanced accuracy compared to traditional methods, it also introduces certain software risks that need to be proactively managed. Key risks include model overfitting on limited training data, algorithmic limitations in capturing real-world variations, integration challenges between system components, performance bottlenecks when dealing with large images, lack of explainability of predictions, and potential security vulnerabilities given the sensitive nature of medical data. Careful software design, extensive testing and validation, performance tuning, security hardening, and adoption of explainable AI techniques are crucial to mitigate these risks. Additionally, bias in training data, software complexity, and obscure failures related to machine learning models require rigorous monitoring and maintenance processes. Overall, the advanced techniques used in Breast Cancer Prediction make software risk management an integral part of developing a robust and reliable solution.

6.1.3 Features to be tested

- Image Upload: Test uploading of histopathology images in different formats, sizes, and resolutions. Verify proper validation and preprocessing.
- User Interface: Test all UI flows and elements like buttons, forms, navigation, and data displays. Check for responsiveness across devices.
- Cancer Prediction: Test prediction accuracy with diverse sample images. Check for correct classification of cancer vs non-cancer.
- Model Training: Validate training process completed without errors. Confirm models are saved properly for prediction.
- Performance: Test system response times and resource usage under different normal and peak load conditions.
- Security: Validate only authorized access to prediction results. Check for data encryption and other security measures.
- Error Handling: Verify proper response for invalid inputs or missing images. Check for fail-safe behavior.
- Integration: Test seamless data flow between UI, preprocessing, and prediction components. Confirm no errors during integration.
- Compatibility: Validate working on different OS versions and hardware configurations per compatibility matrix.
- Functional Flows: Test end-to-end flows like user login, image upload, prediction, and result display for expected behavior.

6.2 Test consolidation

6.2.1 Test item

The items or features to be tested in the test cases are included in the document. Each and every user input is tested. The present condition of the system is tested. It is checked to make sure that environment is ready for the application to work.

6.2.2 Input specifications

These are the inputs required to execute the test case. All required inputs including data elements and values are given below.

Data	Values
Image	histopathological images

Chapter 7

7 System Implementation and Maintenance

7.1 Implementation

System implementation is the conversion of new system into an operating one which involves creating compatible files, training clients and installing hardware. User training is crucial for minimizing resistance to change and giving chance to prove its worth. Training aids user friendly manuals and healthy screens provide the user with a good start. Software maintenance follows conversion to the extent that changes are necessary to maintain satisfactory operations relative to changes in the user's environment. Maintenance often includes minor enhancements or corrections to the problem that surface late in the systems operations. In the implementation phase, the team builds the components either from scratch or by composition. Given the architecture document meant from the design phase and the requirement document from the analysis phase, the team should build exactly what has been requested, though there is still room for innovation and flexibility. For example, a component may be narrowly designed for this particular system, or the component may be made more general to satisfy a reusability

- Careful planning
- Investigation of system and constraints
- Design the methods to achieve changeover.
- Training the staff in the changed phase.
- Evaluation of change over method.
- The method of implementation and time scale to be adopted are found out initially.

7.2 Maintenance

This phase occurs as a result of deploying the whole system at the end users organization. They will perform the beta testing at the end users and inform to the developers about any needed modification to the application. The customer records all the problems that are encountered during the beta testing and reports these to the developer at regular intervals.

7.2.1 Corrective Maintenance

Even with the best quality assurance activities, it is likely that the customer will uncover defects in the software. Corrective maintenance changes the software to correct the defects. Corrective Maintenance activity may consist of repair,

restoration or replacement of equipment. This activity will be result of a regular inspection, which identifies the failure in time for corrective maintenance to be planned and scheduled, then performed during a routine maintenance shutdown.

7.2.2 Adaptive Maintenance

Over time, the original environment(CPU, operating system, business rules, external product characteristics) for which the software was developed is likely to change. Adaptive maintenance results in modification to the software to accommodate changes to its external environment.

7.2.3 Enhanced Maintenance

As software is used, the customer/user will recognize additional functions that will provide the benefit. Perfect maintenance extends the software beyond its original functional requirements.

7.2.4 Preventive Maintenance

Computer software deteriorates due to change, and because of this preventive maintenance often called software re-engineering, must be conducted to enable the software to serve the needs of its end users. Preventive maintenance makes changes to computer programs so that they can be more easily corrected, adapted and enhanced.

Chapter 8

8 Conclusion and Future Scope

8.1 Conclusion

Breast Cancer prediction systems have seen remarkable advancements, providing accurate and efficient solutions for cancer prediction. Looking ahead, the scope for further development is vast, with potential enhancements to enhance accuracy, security, and user experience. Strengthening privacy and security measures are essential aspects to consider. Here, any histopathological images can be given as inputs and optimal result is predicted.vThe integration of AI-based adaptive learning can enhance system accuracy, particularly in challenging environments. Breast cancer prediction systems will continue to be a vital and transformative tool for medical organizations and institutions in the future

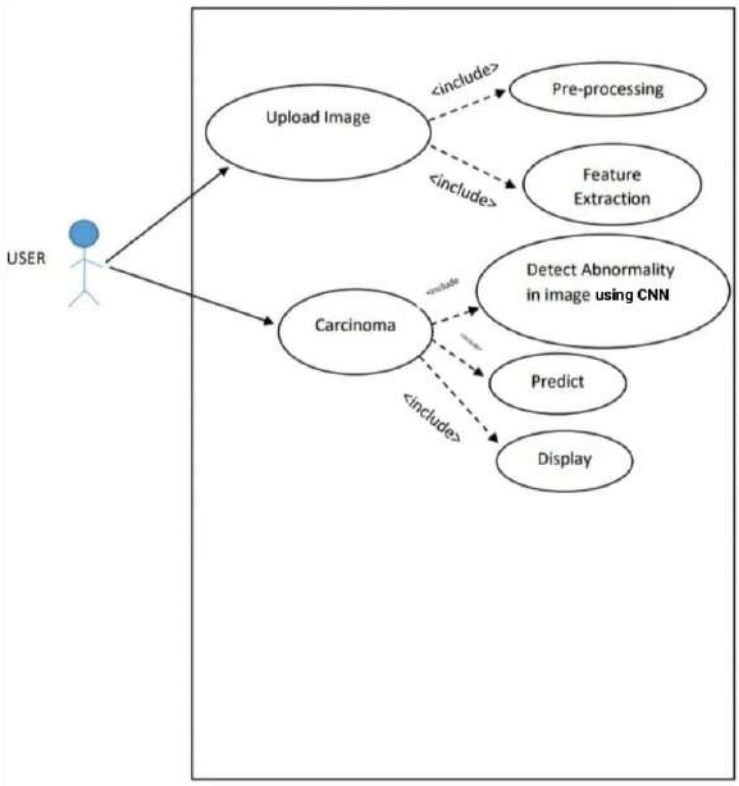
8.2 Future Scope

- Web based platform: Accessible to all the users where they interact with the doctor and results are available online.
- Privacy and Data Protection: Enhancing privacy features and adopting privacy-preserving algorithms will be crucial to safeguarding user data and complying with evolving data protection regulations.
- User Interface and Experience: Investing in user-friendly interfaces and seamless integration with existing systems will encourage greater user acceptance and adoption of the system.
- Real-time Analytics: Developing real-time attendance analytics can provide valuable insights into cancer prediction trends, enabling medical organizations to make data-driven decisions. More image should be trained or the accuracy may vary. Therefore, Several new models should be trained and the more f1 – score giving model should be taken in the future.
- Customization Options: Offering customization options to tailor the system to specific organizational needs and requirements will increase its versatility and adaptability.
- Cross-Platform Compatibility: Ensuring compatibility with various devices and operating systems will make the system accessible and widely applicable in different settings.

Appendix

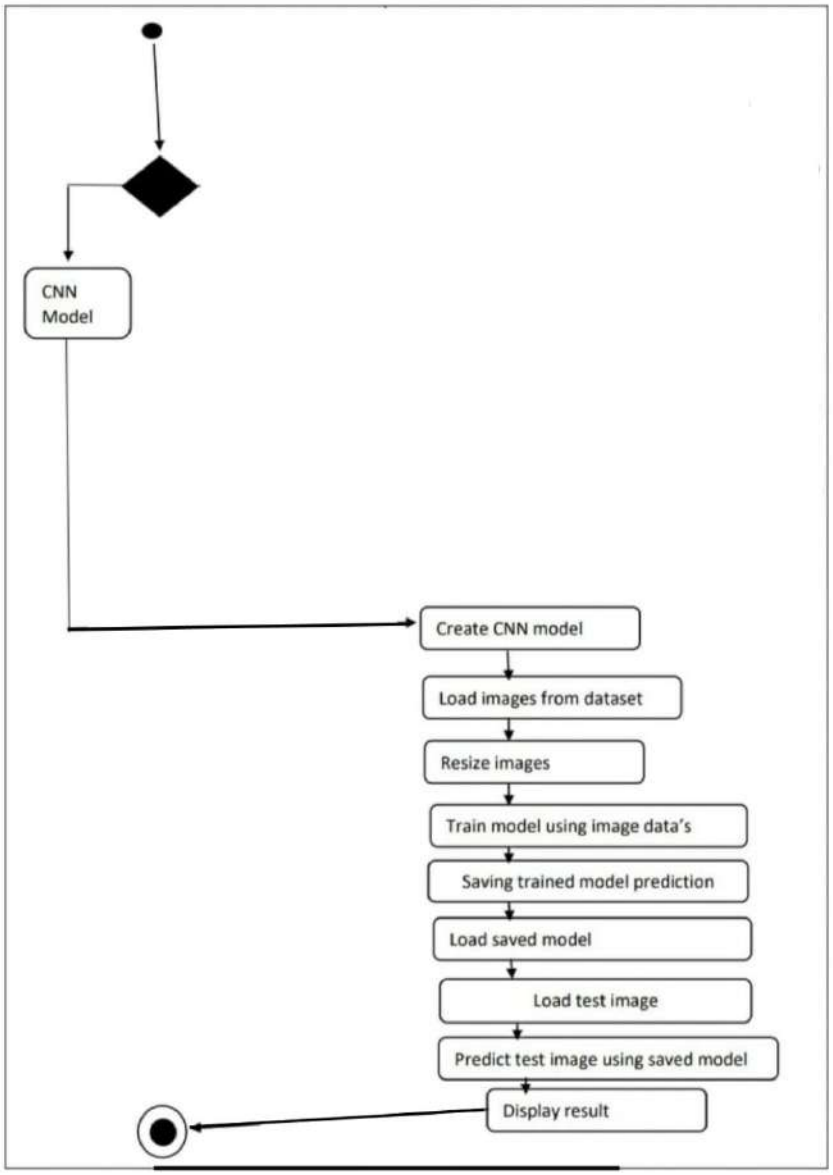
A Usecase Diagram

A.1



B Activity Diagram

B.1



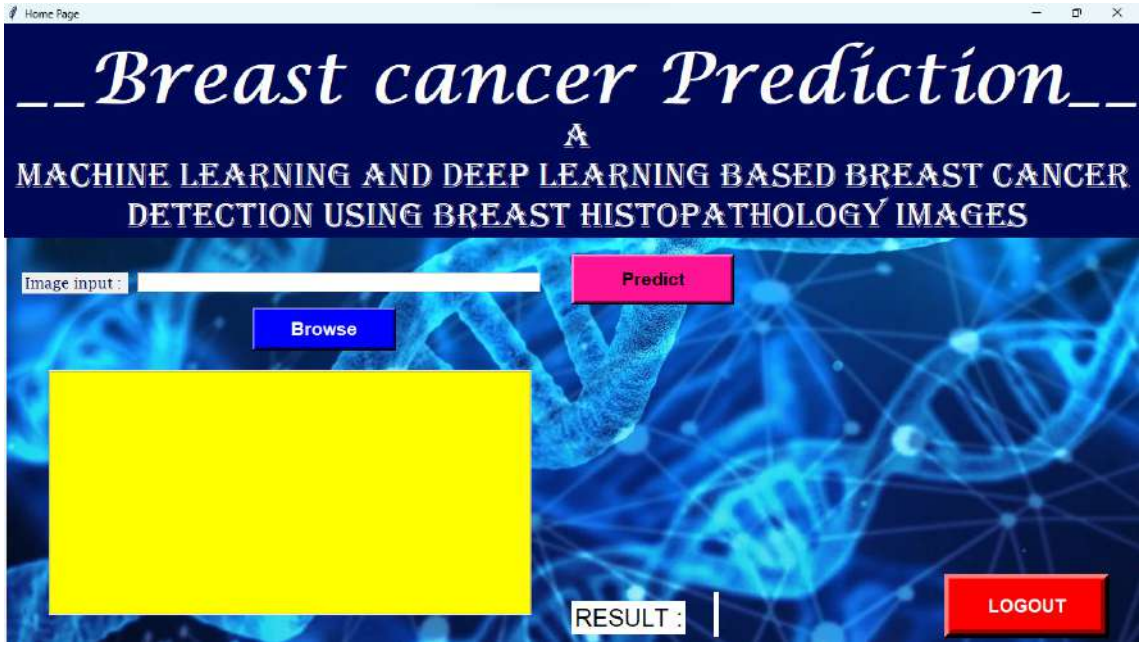
B.2

2. **Breast Cancer**

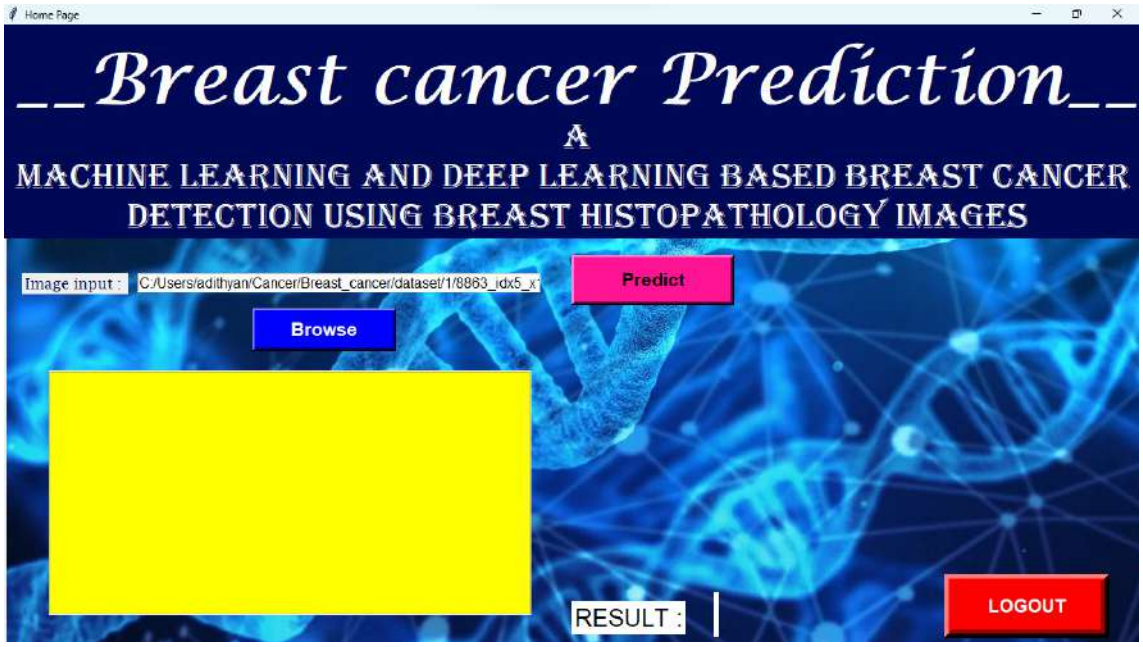
System	**Breast cancer**

C USER INTERFACES

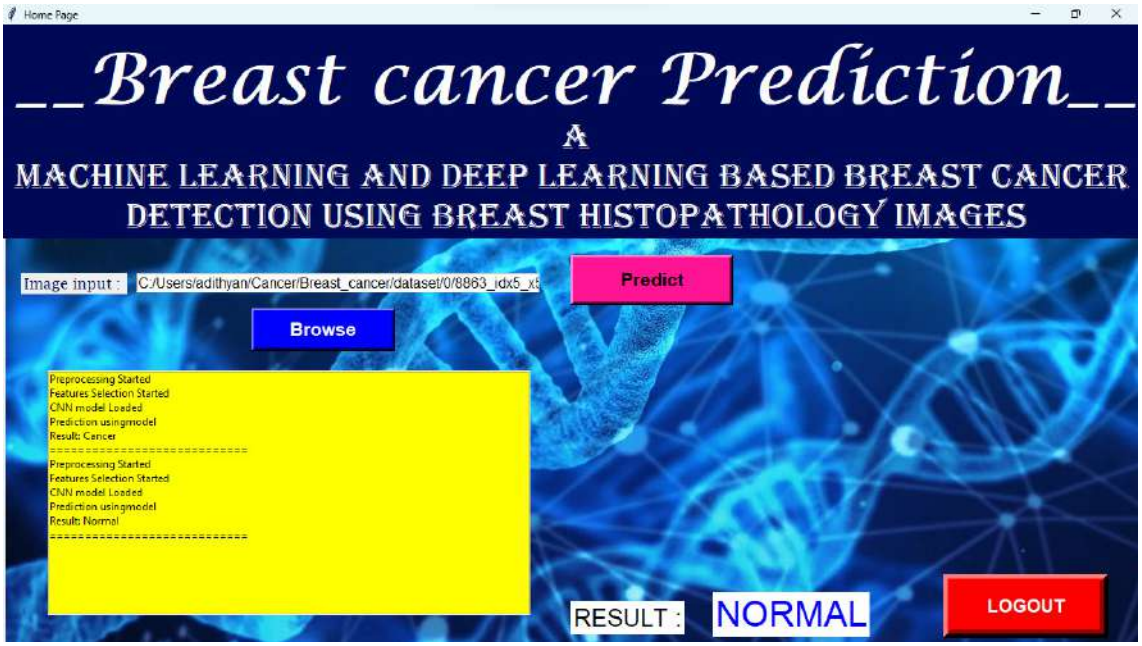
C.1 HOME



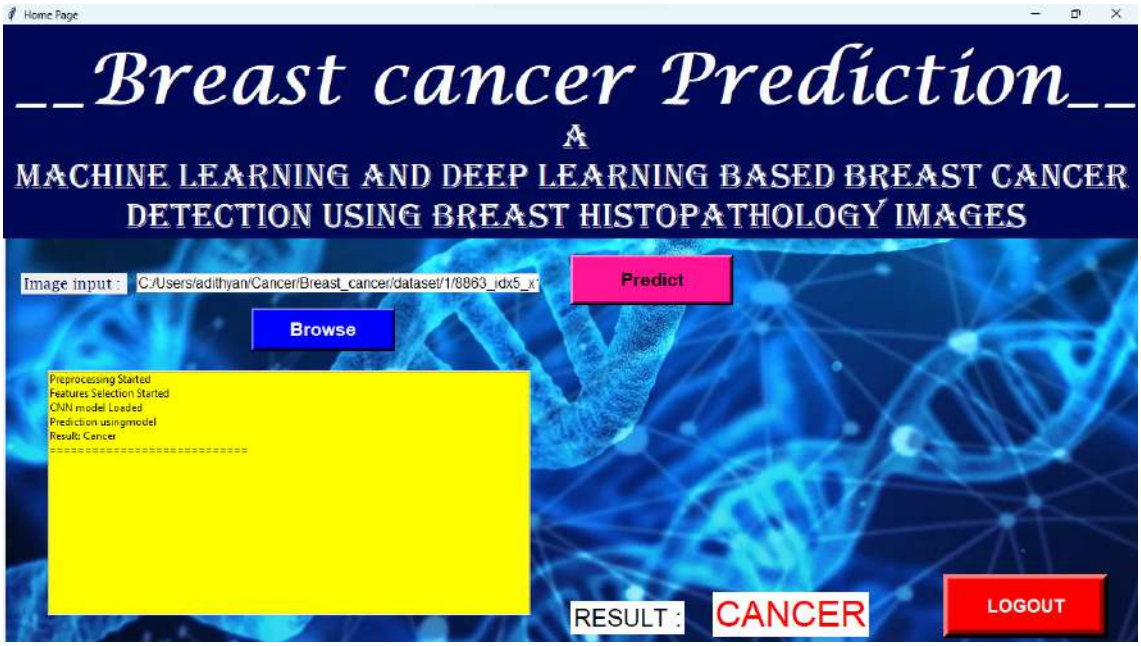
C.2 IMAGE LOCATION



C.3 NORMAL RESULT



C.4 CANCEROUS RESULT



D CODE

preprocess.py

```
import cv2

import numpy as np
import random
from imutils import paths
import os
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
import json
from tensorflow.keras.preprocessing.image import
    img_to_array
from tensorflow.keras.utils import to_categorical
from sklearn import metrics
import pickle

import mahotas as mt
from skimage.filters import gabor

def extract_features(gray):
    textures = mt.features.haralick(gray)
    # cv2.imshow("textures",textures)
    # cv2.waitKey(0)
        # take the mean of it and return it
    ht_mean  = textures.mean(axis=0)
    # print(ht_mean)
    return ht_mean


def build_filters():
        filters = []
        ksize = 31
        for theta in np.arange(0, np.pi, np.pi / 16):
            kern = cv2.getGaborKernel((ksize, ksize),
                4.0, theta, 10.0, 0.5, 0, ktype=cv2.CV_32F
                )
            kern /= 1.5*kern.sum()
            filters.append(kern)
            return filters

def process(img, filters):
        accum = np.zeros_like(img)
```

```
        for kern in filters:
            fimg = cv2.filter2D(img, cv2.CV_8UC3, kern)
            np.maximum(accum, fimg, accum)
            return accum


filters=build_filters()



data = []
label = []
print("[INFO] loading images...")
img_dir=sorted(list(paths.list_images("dataset")))
random.shuffle(img_dir)
print("[INFO]   Preprocessing...")
count=0
for i in img_dir:
    img = cv2.imread(i)
    img=cv2.resize(img, (64,64))
    img1=process(img,filters)
    gray = cv2.cvtColor(img1, cv2.COLOR_BGR2GRAY)
    res2=extract_features(gray)
    # print(res2)
    # print(type(res2))
    # imgdata=img_to_array(res2)
    data.append(res2)
    lb=i.split(os.path.sep)[-2]
    if(lb=="1"):
        label.append(1)
    else:
        label.append(0)
print(len(data))
print(len(label))
```

[breaklines=true]

SvmTrain.py

```
import cv2

import numpy as np
import random
from imutils import paths
import os
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
```

```
import json
from tensorflow.keras.preprocessing.image import
    img_to_array
from tensorflow.keras.utils import to_categorical
from sklearn import metrics
import pickle

import mahotas as mt
from skimage.filters import gabor

def extract_features(gray):
    textures = mt.features.haralick(gray)
    # cv2.imshow("textures",textures)
    # cv2.waitKey(0)
        # take the mean of it and return it
    ht_mean  = textures.mean(axis=0)
    # print(ht_mean)
    return ht_mean


def build_filters():
        filters = []
        ksize = 31
        for theta in np.arange(0, np.pi, np.pi / 16):
            kern = cv2.getGaborKernel((ksize, ksize),
                4.0, theta, 10.0, 0.5, 0, ktype=cv2.CV_32F
                )
            kern /= 1.5*kern.sum()
            filters.append(kern)
            return filters

def process(img, filters):
        accum = np.zeros_like(img)
        for kern in filters:
            fimg = cv2.filter2D(img, cv2.CV_8UC3, kern)
            np.maximum(accum, fimg, accum)
            return accum


filters=build_filters()



data = []
label = []
print("[INFO] loading images...")
```

```
img_dir=sorted(list(paths.list_images("dataset")))
random.shuffle(img_dir)
print("[INFO]   Preprocessing...")
count=0
for i in img_dir:
    img = cv2.imread(i)
    img=cv2.resize(img, (64,64))
    img1=process(img,filters)
    gray = cv2.cvtColor(img1, cv2.COLOR_BGR2GRAY)
    res2=extract_features(gray)
    # print(res2)
    # print(type(res2))
    # imgdata=img_to_array(res2)
    data.append(res2)
    lb=i.split(os.path.sep)[-2]
    if(lb=="1"):
        label.append(1)
    else:
        label.append(0)
print(len(data))
print(len(label))
lenofimage=len(data)
traindata=np.array(data).reshape(lenofimage,-1)
trainlabel=np.array(label)
print(traindata.shape)
print(trainlabel.shape)
X_train, X_test, y_train, y_test = train_test_split(
    traindata, trainlabel, test_size=0.25, random_state
    =42)
print(X_train.shape)
print(y_train.shape)
SVMclassifier = SVC(kernel='linear', random_state=0)
SVMclassifier.fit(X_train, y_train)
# Saving Trained Model
# dbfile=open("SVMmodel","wb")
# pickle.dump(SVMclassifier,dbfile)
# dbfile.close()
y_pred= SVMclassifier.predict(X_test)
print(y_pred)
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))

# 62.38
```

[breaklines=true] **Svmtest.py**

```
import cv2
```

```
import numpy as np
import random
from imutils import paths
import os
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
import json
from tensorflow.keras.preprocessing.image import
    img_to_array
from tensorflow.keras.utils import to_categorical
from sklearn import metrics
import pickle

import mahotas as mt
from skimage.filters import gabor



def extract_features(gray):
    textures = mt.features.haralick(gray)
    # cv2.imshow("textures",textures)
    # cv2.waitKey(0)
        # take the mean of it and return it
    ht_mean  = textures.mean(axis=0)
    # print(ht_mean)
    return ht_mean


def build_filters():
        filters = []
        ksize = 31
        for theta in np.arange(0, np.pi, np.pi / 16):
            kern = cv2.getGaborKernel((ksize, ksize),
                4.0, theta, 10.0, 0.5, 0, ktype=cv2.CV_32F
                )
            kern /= 1.5*kern.sum()
            filters.append(kern)
            return filters

def process(img, filters):
        accum = np.zeros_like(img)
        for kern in filters:
            fimg = cv2.filter2D(img, cv2.CV_8UC3, kern)
            np.maximum(accum, fimg, accum)
```

```
            return accum


# img=cv2.imread("im1.JPG")


filters=build_filters()


img = cv2.imread("31.png")
img=cv2.resize(img, (64,64))
img1=process(img,filters)
gray = cv2.cvtColor(img1, cv2.COLOR_BGR2GRAY)
res2=extract_features(gray)

dbfile=open("SVMmodel","rb")
adamodel=pickle.load(dbfile)
dbfile.close()
# Prediction based on selected model
y_pred = adamodel.predict(res2.reshape(1,-1))
print(y_pred)
```

[breaklines=true] **CNNTrain.py**

```
import cv2

import numpy as np
import random
from imutils import paths
import os
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
import json
from tensorflow.keras.preprocessing.image import
    img_to_array
from tensorflow.keras.utils import to_categorical
from sklearn import metrics
import pickle
import pandas as pd
import numpy as np
import os
from glob import glob
import random
import matplotlib.pylab as plt
from tensorflow.keras.models import model_from_json
```

```
from sklearn.model_selection import train_test_split
import tensorflow as tf
import keras
from keras.utils.np_utils import to_categorical
from keras.models import Sequential
from keras.preprocessing.image import ImageDataGenerator

from tensorflow.keras.optimizers import SGD, RMSprop,
    Adam, Adagrad, Adadelta
from keras.layers import Dense, Dropout, Activation,
    Flatten, BatchNormalization, Conv2D, MaxPool2D,
    MaxPooling2D
from tensorflow.keras.callbacks import EarlyStopping


data = []
label = []
print("[INFO] loading images...")
img_dir=sorted(list(paths.list_images("dataset")))
random.shuffle(img_dir)
print("[INFO]  Preprocessing...")
count=0
for i in img_dir:
    img = cv2.imread(i)
    img=cv2.resize(img, (64,64))
    imgdata=img_to_array(img)
    data.append(imgdata)
    lb=i.split(os.path.sep)[-2]
    if(lb=="1"):
        label.append(1)
    else:
        label.append(0)

traindata=np.array(data)
trainlabel=np.array(label)

X_train, X_test, y_train, y_test = train_test_split(
    traindata, trainlabel, test_size=0.25, random_state
    =42)
print(X_train.shape)
print(y_train.shape)

model = Sequential()

# Convolutional layer and maxpool layer 1
model.add(Conv2D(32,(3,3),activation='relu',input_shape
```

```
    =(64,64,3)))
model.add(MaxPool2D(2,2))

# Convolutional layer and maxpool layer 2
model.add(Conv2D(64,(3,3),activation='relu'))
model.add(MaxPool2D(2,2))

# Convolutional layer and maxpool layer 3
model.add(Conv2D(128,(3,3),activation='relu'))
model.add(MaxPool2D(2,2))

# Convolutional layer and maxpool layer 4
model.add(Conv2D(128,(3,3),activation='relu'))
model.add(MaxPool2D(2,2))

# This layer flattens the resulting image array to 1D
    array
model.add(Flatten())

# Hidden layer with 512 neurons and Rectified Linear Unit
     activation function
model.add(Dense(512,activation='relu'))

# Output layer with single neuron which gives 0 for Cat
    or 1 for Dog
#Here we use sigmoid activation function which makes our
    model output to lie between 0 and 1
model.add(Dense(1,activation='sigmoid'))

# model = Sequential()
# model.add(Conv2D(64, kernel_size=3, activation='relu',
    input_shape=(50,50,3)))
# model.add(Conv2D(32, kernel_size=3, activation='relu'))
# model.add(Flatten())
# model.add(Dense(2, activation="softmax"))
# adam = Adam(learning_rate=0.0001)
model.compile(loss="binary_crossentropy", optimizer="adam
    ", metrics=['accuracy'])

history = model.fit(
    X_train, y_train,
    validation_data=(X_test, y_test),
    epochs= 10,
    batch_size=10
    )
Y_pred = model.predict(X_test)
```

```
print(Y_pred)
#serialize model to JSON
model_json = model.to_json()
with open("CNNmodel.json", "w") as json_file:
        json_file.write(model_json)
# serialize weights to HDF5
model.save_weights("CNNmodelw.h5")
print("Saved model to disk")
#0.9133
```

[breaklines=true]
CNNTest.py

```
from tensorflow.keras.preprocessing.image import
    img_to_array
from tensorflow.keras.models import load_model
import numpy as np
import argparse
import imutils
import cv2
from tensorflow.keras.models import model_from_json


img = cv2.imread("10.png")
img=cv2.resize(img, (64,64))
img = img_to_array(img)
##img = img.reshape(1, 100, 36, 1)

json_file = open('CNNmodel.json', 'r')
loaded_model_json = json_file.read()
json_file.close()
model = model_from_json(loaded_model_json)
# load weights into new model
model.load_weights("CNNmodelw.h5")
print("Loaded model from disk")
img = np.expand_dims(img, axis = 0)
result = model.predict(img)
res1=result[0][0]
if(res1>0.32):
    print("Cancer")
else:
    print("Normal")
print("res=>",result[0][0])
result=np.argmax(result[0][0])
print(result)
```

[breaklines=true] **Resnet_train.py**

```
import numpy as np
import random
import cv2
from glob import glob
import matplotlib.pyplot as plt
from imutils import paths
import os
from PIL import Image
import keras
from tensorflow.keras.preprocessing.image import
    ImageDataGenerator
from tensorflow.keras.applications import ResNet50
from tensorflow.keras.applications.resnet50 import
    preprocess_input
from tensorflow.keras import Model, layers
from tensorflow.keras.models import load_model,
    model_from_json
from tensorflow.keras.preprocessing.image import
    img_to_array
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.models import model_from_json

from sklearn.model_selection import train_test_split



conv_base = ResNet50(
    include_top=False,
    weights='imagenet')

for layer in conv_base.layers:
    layer.trainable = False
x = conv_base.output
x = layers.GlobalAveragePooling2D()(x)
x = layers.Dense(128, activation='relu')(x)
predictions = layers.Dense(2, activation='softmax')(x)
model = Model(conv_base.input, predictions)

model.compile(loss='sparse_categorical_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])


data = []
label = []
```

```
print("[INFO] loading images...")
img_dir=sorted(list(paths.list_images("dataset")))
random.shuffle(img_dir)
print("[INFO]   Preprocessing...")
count=0
for i in img_dir:
    img = cv2.imread(i)
    img=cv2.resize(img,(256,256))
    imgdata=img_to_array(img)
    data.append(imgdata)
    lb=i.split(os.path.sep)[-2]
    if(lb=="0"):
        label.append(0)
    else:
        label.append(1)
print(len(data))
print(len(label))
lenofimage=len(data)
traindata=np.array(data)
trainlabel=np.array(label)
train_labels = to_categorical(trainlabel, num_classes=2)
print(traindata.shape)
print(train_labels.shape)
X_train, X_test, y_train, y_test = train_test_split(
    traindata, trainlabel, test_size=0.25, random_state
    =42)
print(X_train.shape)
print(y_train.shape)
print(X_test.shape)
print(y_test.shape)
```

```
model.fit(X_train, y_train, epochs=3, validation_split
    =0.2, batch_size=5)

model_json = model.to_json()
with open("resnet_model.json", "w") as json_file:
        json_file.write(model_json)
# serialize weights to HDF5
model.save_weights("resnetw_model.h5")
print("[INFO] Saved model to disk")
y=model.predict(X_train)
print(y)
```

[breaklines=true] **Densenettrain.py**

```
import numpy as np
import random
import cv2
from glob import glob
import matplotlib.pyplot as plt
from imutils import paths
import os
from PIL import Image

import keras
from tensorflow.keras.preprocessing.image import
    ImageDataGenerator
from tensorflow.keras.applications.densenet import
    DenseNet121
from keras import Model, layers
from keras.models import load_model, model_from_json
from tensorflow.keras.preprocessing.image import
    img_to_array
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.models import model_from_json
from tensorflow.keras.layers import Dense,
    GlobalAveragePooling2D,Convolution2D,
    BatchNormalization
from tensorflow.keras.layers import Flatten,MaxPooling2D,
    Dropout
from sklearn.model_selection import train_test_split




from tensorflow.keras.layers import Dense, Dropout, Input
    ,Flatten, Conv2D, MaxPool2D, BatchNormalization,
    AveragePooling2D, GlobalAveragePooling2D
from tensorflow.keras.models import Model,Sequential,
    load_model
from tensorflow.keras.optimizers import Adam
def build_densenet():
    densenet = DenseNet121(weights='imagenet',
        include_top=False)

    input = Input(shape=(256, 256, 3))
```

```
    x = Conv2D(3, (3, 3), padding='same')(input)

    x = densenet(x)

    x = GlobalAveragePooling2D()(x)
    x = BatchNormalization()(x)
    x = Dropout(0.5)(x)
    x = Dense(256, activation='relu')(x)
    x = BatchNormalization()(x)
    x = Dropout(0.5)(x)

    # multi output
    output = Dense(15,activation = 'softmax', name='root
        ')(x)


    # model
    model = Model(input,output)

    optimizer = Adam(lr=0.002, beta_1=0.9, beta_2=0.999,
        epsilon=0.1, decay=0.0)
    model.compile(loss='sparse_categorical_crossentropy',
         optimizer=optimizer, metrics=['accuracy'])
    model.summary()

    return model

model = build_densenet()



data = []
label = []
print("[INFO] loading images...")
img_dir=sorted(list(paths.list_images("dataset")))
random.shuffle(img_dir)
print("[INFO]   Preprocessing...")
count=0
for i in img_dir:
    img = cv2.imread(i)
    img=cv2.resize(img,(256,256))
    imgdata=img_to_array(img)
    data.append(imgdata)
    lb=i.split(os.path.sep)[-2]
    if(lb=="0"):
        label.append(0)
```

```
    else:
        label.append(1)
print(len(data))
print(len(label))
lenofimage=len(data)
traindata=np.array(data)
trainlabel=np.array(label)
train_labels = to_categorical(trainlabel, num_classes=2)
print(traindata.shape)
print(train_labels.shape)
X_train, X_test, y_train, y_test = train_test_split(
    traindata, trainlabel, test_size=0.25, random_state
    =42)
print(X_train.shape)
print(y_train.shape)
print(X_test.shape)
print(y_test.shape)


model.fit(X_train, y_train, epochs=10, validation_split
    =0.2, batch_size=5,verbose=1)

# model_json = model.to_json()
# with open("densenet_model.json", "w") as json_file:
#         json_file.write(model_json)
# # serialize weights to HDF5
# model.save_weights("densenetw_model.h5")
# print("[INFO] Saved model to disk")
```

[breaklines=true] **GUI_new.py**

```
from tkinter import *
import time
import re
#Import scikit-learn metrics module for accuracy
    calculation
import pickle
from PIL import Image, ImageTk
import cv2
from tkinter.filedialog import askopenfile
from tensorflow.keras.preprocessing.image import
    img_to_array
from tensorflow.keras.models import load_model
import numpy as np
import argparse
import imutils
```

```
import cv2
from tensorflow.keras.models import model_from_json
from tensorflow.keras.preprocessing.image import
    img_to_array
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.applications.vgg16 import
    preprocess_input
from tensorflow.keras.models import model_from_json

json_file = open('resnet_model.json', 'r')
loaded_model_json = json_file.read()
json_file.close()
modelres = model_from_json(loaded_model_json)
# load weights into new model
modelres.load_weights("resnetw_model.h5")
print("Loaded Resnet model from disk")

json_file = open('densenet_model.json', 'r')
loaded_model_json = json_file.read()
json_file.close()
modeldense = model_from_json(loaded_model_json)
# load weights into new model
modeldense.load_weights("densenet.h5")

print("Loaded Sensenet model from disk")

json_file = open('CNNmodel.json', 'r')
loaded_model_json = json_file.read()
json_file.close()
cnnmodel = model_from_json(loaded_model_json)
# load weights into new model
cnnmodel.load_weights("CNNmodelw.h5")
print("Loaded model from disk")

def pp(a):
    global mylist
    mylist.insert(END, a)



def predict(val):
    print(val)
    img = cv2.imread(val)
    imgr=cv2.resize(img, (64,64))
    imgarr = img_to_array(imgr)
    ##img = img.reshape(1, 100, 36, 1)
```

```
imgarr = np.expand_dims(imgarr, axis = 0)
result = cnnmodel.predict(imgarr)
res1=result[0][0]
print(res1)

if(res1>0.32):
    cnnres=1
else:
    cnnres=0
print("cnnpred==>",cnnres)
imgres=cv2.resize(img,(256,256))
imgdata=img_to_array(imgres)
print(imgdata)
# print(type(imgdata))
# print(imgdata.shape)

train_ds= np.expand_dims(imgdata, axis=0)
# print(train_ds.shape.)

yres=modelres.predict(train_ds)
print(yres)
resres=np.argmax(yres[0])
print("respred==>",resres)

ydense=modeldense.predict(train_ds)
print(ydense)
denseres=np.argmax(ydense[0])
print("densepred==>",denseres)

reslist=[cnnres,resres,denseres]

print("result list==>",reslist)
finalres=max(reslist)
if(finalres==1):
    resc="Cancer"
    print("Cancer")
    root.after(3100, lambda :shrslt.config(text="
        CANCER",fg="red"))

else:
    resc="Normal"
    print("Normal")
    root.after(3100, lambda :shrslt.config(text="
        NORMAL",fg="blue"))
```

```
    root.after(500, lambda : pp("Preprocessing Started ")
        )
    root.after(2000, lambda : pp("Features Selection
        Started "))
    root.after(2200, lambda : pp("CNN model Loaded"))
    root.after(2400, lambda : pp("Prediction usingmodel
        "))
    root.after(2600, lambda : pp("Result: "+resc))
    root.after(2800, lambda : pp
        ("=========================================="))


def browseim():
    global cimg,shrslt,E1
    path = askopenfile()
    n=path.name
    print(path)
    E1.delete(0,"end")
    E1.insert(0, n)

#def upload_file():
 #    global cimg,shrslt,E1
 #    root=Tk()
 #    f_types=[('Png Files','*.png')]
 #    filename=filedialog.askopenfilename(filetypes=
     f_types)
 #   cimg=ImageTk.PhotoImage(file=filename)

def userHome():
    global root, mylist,shrslt,E1
    root = Tk()
    root.geometry("1400x625+0+0")
    root.title("Home Page")

    image = Image.open("dna.png")
    image = image.resize((1500, 725), Image.ANTIALIAS)
    pic = ImageTk.PhotoImage(image)
    lbl_reg=Label(root,image=pic,anchor=CENTER)
    lbl_reg.place(x=0,y=0)

    #----------------------INFO TOP----------------
    lblinfo = Label(root, font=( 'lucida calligraphy'
        ,61, 'bold' ),text="__Breast cancer Prediction__",
        fg="white",bg="#000955",bd=10,anchor='w')
```

```
lblinfo.place(x=0,y=0)

lblinfo2 = Label(root, font=( 'Algerian' ,31 ),text="
    A\n Machine learning and Deep learning based
    Breast cancer \n   detection using Breast
    Histopathology Images   ",fg="white",bg="#000955",
    anchor='w')
lblinfo2.place(x=0,y=100)
lblinfo3 = Label(root, font=( 'Lucida fax' ,0 ),text
    ="Image input : ",fg="#000955",anchor='w')
lblinfo3.place(x=20,y=280)
E1 = Entry(root,width=50,font="Will&Grace")
E1.place(x=150,y=280)
mylist = Listbox(root,width=90, height=17,bg="yellow
    ")

mylist.place( x = 50, y = 390 )
btntrn=Button(root,padx=10,pady=2, bd=4 ,fg="white",
    font=('ariel' ,16,'bold'),width=10, text="Browse",
     bg="blue",command=lambda:browseim())
btntrn.place(x=280, y=320)
# btntrn=Button(root,padx=10,pady=2, bd=4 ,fg="white",
   font=('ariel' ,16,'bold'),width=10, text="image",
   bg="red",command=lambda:upload_file())
#btntrn.place(x=700, y=400)
btnhlp=Button(root,padx=40,pady=6, bd=4 ,fg="black",
    font=('ariel' ,16,'bold'),width=7, text="Predict",
     bg="deep pink",command=lambda:predict(E1.get()))
btnhlp.place(x=640, y=260)



rslt = Label(root, font=( 'aria' ,20, ),text="RESULT
    :",fg="black",bg="white",anchor=W)
rslt.place(x=640,y=650)
shrslt = Label(root, font=( 'aria' ,30, ),text="",fg
    ="blue",bg="white",anchor=W)
shrslt.place(x=800,y=640)

def qexit():
    root.destroy()

btnexit=Button(root,padx=16,pady=8, bd=10 ,fg="white
    ",font=('ariel' ,16,'bold'),width=10, text="LOGOUT
    ", bg="red",command=qexit)
btnexit.place(x=1060, y=620)
```

```
    root.mainloop()

userHome()
```

[breaklines=true]

AIR QUALITY INDEX

PROJECT REPORT

Submitted By

Agnal Biju
Reg. No. CCAVBCA001

For the award of the Degree of

Bachelor of Computer Application

(BCA)

(University of Calicut)

under the guidance of

Ms. Viji Viswanathan

Assistant Professor

BACHELOR OF COMPUTER APPLICATION
DEPARTMENT OF COMPUTER SCIENCE
CHRIST COLLEGE
IRINJALAKUDA, KERALA
2021-2024

DEPARTMENT OF COMPUTER SCIENCE

CHRIST COLLEGE (AUTONOMOUS)

IRINJALAKUDA

CERTIFICATE

This is to certify that the project report entitled ***"Air Quality Index"*** *is a bonfied record of the project work done by* ***Agnal Biju*** *in partial fulfillment of the requirement for the sixth semester of* ***Bachelor of Computer Application*** *in Department of Computer Science of* **CHRIST COLLEGE (AUTONOMOUS) IRINJALAKUDA**

Ms. Viji Viswanathan
Assistant Professor,CS
Internal Guide

Ms. Sini Thomas
Head of the Department
Computer Science

EXTERNAL EXAMINER

INTERNAL EXAMINER

DECLARATION

We hereby declare that this project work **"AIR QUALITY INDEX"** submitted to Christ College (Autonomous) Irinjalakuda, affiliated to Calicut University in partial fulfillment of the requirement for the award of the Bachelor of Computer Application, is a record of original work done by us, under the guidance of Ms. VIJI VISWANATHAN, Department of computer Science.

Place: Irinjalakuda AGNAL BIJU

ACKNOWLEDGEMENT

First and foremost we like to thank Lord almighty for his providence and for being the guiding light throughout the project. We wish to express my sincere gratitude to our beloved Department head for giving me all the facilities for our project. We take this opportunity to express my gratitude to the class teacher Ms. SOUMYA P.S and head of the department Ms. SINI THOMAS who has been supported us throughout the course of this project. We are thankful for her aspiring guidance and valuable advice during the project work. We express my sincere thanks to my project guide Ms. VIJI VISWANATHAN for supporting and guiding throughout the project.We would take this opportunity to specially thank all other faculty members for their constant and continuous motivation. Finally we would like to thank my family and friends for giving valuable advice and moral support throughout our project.

ABSTRACT

Air Quality Index is a innovative web application that takes on paramount importance as it addresses the need for timely and accurate information about air quality. The web application is enriched with a login form ,a registratin form and a prediction window where we enter the prediction attributes.This application gives accurate air qualty and a graphical visualization for the same. All these features make this website more adaptable and user-friendly.

Contents

Chapter 1

1 Introduction

Air quality is a critical component of environmental health management and its significance cannot be overstated.Air pollution has become a most pressing concern and it has affected not only the environment but also the health of billions of people. Air quality indexing takes on importance as it addresses the timely and accurate information about the air quality.The purpose of the system is to create a robust and reliable system that not only monitors air quality but also predicts its future trends.This step represents a critial step forward in environmental management and how we safeguard the air we breathe.

.

1.1 Overview

The core objective of the project is to develop an Air quality indexing and prediction system that harness the power of data and predictive analytics to provide real-time insights to air quality. The development of this system can lead to a healthier and more sustainable future.

Chapter 2

2 System Analysis

2.1 Purpose

The purpose of this project is to develop an Air quality indexing and prediction system that harness the power of data and predictive analytics to provide real-time insights to air quality.

2.1.1 Existing System

Presently all the processes gives very simple outlook. It is provided with login and registeration pages. The web application is developed using HTML,CSS and Bootstrap in the front end. Backend is developed using Python and the framework being django. Limitations of existing system :No Graph provided,No login and registeration pages provided,

2.1.2 Proposed System

As a software developer, we are expected to design and develop any program that works efficiently without any delays.For Convinence we provider a login and registeration pages to make it a web application type project . And with the addition of a Graph to make the project more understandable to the users.

2.2 Problem definition

To know what the problem is and what the needs are before developing it. Here we design a web application for detecting the air quality

2.3 FEASIBILITY STUDY

After the problem is clearly understood and solutions proposed, the next step is to conduct the feasibility study. Feasibility study is defined as evaluation or analysis of the potential impact of the proposed project or program. The objective is to determine whether the proposed system is feasible. There are three aspects of feasibility study to which the proposed system is subjected as discussed below.

2.3.1 Technical Feasibility

Technical feasibility assess whether the current technical resources are sufficient for the new system. We can upgrade the level of technology for supporting our web application. We check whether the proposed system can be implemented in the present system without supporting the existing hardware.

2.3.2 Economical Feasibility

Economic feasibility determines whether the time and money are available to develop the system. There is no additional hardware used to develop the site, it is inexpensive to build.

2.3.3 Operational Feasibility

Operational feasibility determines if the human resources are available to operate the system once it has been installed. No extra training is needed to use this system. Anyone who has knowledge in internet in english language can easily use the system. The resources that are required to implement or install are already available with the organization.

Chapter 3

3 Software Requirement Specification

3.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the Project Air quality detection system. It illustrate the purpose and complete description for the development of the system. It explain system constraints, interface and interactions with other external applications. This document is primarily intended to harness the power of data and predictive analytics to provide real-time insights to air quality

3.2 Scope

Our project has made it easier to detect the air quality in real-time . We can get all information about the quality of air wheather it is good or moderate or poor etc with a suitable graph included .

3.3 Overall Description

This section give an overview of our web application is to create a robust and reliable system that not only monitors air quality but also predicts its future trends.This step represents a critial step forward in environmental management and how we safeguard the air we breathe.

3.3.1 Product Perspective

The product perspective of this Air Quality Detection System involves designing and integrating these components to provide accurate,timely and actionable information to the users abut the air they breathe.

3.3.2 Product Functionality

This system can empower users to make informed decesion about their health and well-being in relation to air pollution .

3.3.3 Users and Characteristics

Each user group may have different needs,preferences and levels of technical expertise,so the Air Quality Detection System should be designed with user-centric features and interfaces to accomodate the diverse characterstics effectively.

3.4 Specific Requirements

3.4.1 Hardware Requirements

- System: IBM-Compatible PC
- Processor: Pentium IV or above
- Speed: Above 1GHz
- RAM capacity: 512 MB
- Hard Dsk drive: 40 GB
- Keyboard: Standard
- Mouse: Standard
- Monitor: SVGA Color

3.4.2 Software Requirements

- Operating System:Windows or ubuntu
- Languages used: Python Django
- Data Collection : CSV
- Technologies used: HTML,CSS,Bootsrap

3.5 Functional Requirements

It contains three main modules.

- 1.Login Form
- 2.Registration form
- 3. Prediction window

login Form

The login page is for the logging in the necessary credentials into the project for accessing into the main index page of the project.

Registration Form

In case if the user doesn't have an account he/she might not be able to log in.Therefore a registration form is created to enter the provided details into the form and after registering the user cn go back to the login page to log in the necessary credentials.

Prediction Window

Once logging in a prediction window is displayed with the necessary pollutant information in the air and by typing can give the accurate air quality information.

3.6 Non Functional Requirements

Non-functional requirements define the overall qualities or attributes of the resulting system. Non-functional system place restrictions on the product being developed, the developed process, and specify external constraints that the product must meet. Examples of non-functional requirements include safety, security, usability, reliability and performance requirements. Project management issues (costs, time and schedule) are often considered as non-functional requirements. The principal non - functional constraints which are relevant to critical systems :

- performance
- security
- safety
- usability

Performance

Performance requirements concern the speed of operation of a system. Types of performance requirements :

- Response requirements (how quickly the system reacts to a user input).
- Throughput requirements (how much the system can accomplish within a specified amount of time).
- Availability requirements (is the system available for service when requested by end users). The speed of operation of this system is adequate for the requirements.

Reliability

- Reliability is the ability of a system to perform its required function under stated conditions for a specified period of time.
- Constraints on the runtime behavior of the system. This system is reliable because its functionalities can be done on the required conditions.

Safety

Safety requirements are not required which exclude unsafe situation from the possible solution of the system.

Usability

Usability is the ease with which a user can learn to operate, prepare inputs for, and interpret outputs of system or components. Usability requirements include :

- information error messages.
- well-formed user interfaces.

3.7 Interface Requirements

3.7.1 Hardware interfaces

The system must run over the internet, all the hardware shall require to connect internet will be hardware interface for the system. As for example modem, WAN-LAN.

3.7.2 Software interfaces

Software interface required for the working of the project is the appropriate operating system.

3.7.3 Communication interfaces

The user of the site communicate using the network connectivity and the data set is accessed

3.8 Security Requirements

- User accesses only their account.
- Validation of input is handled.
- This application containing the computer systems is physically secured against arms or surreptitious entry by intruders.
- Users must be authorized carefully to reduce changes of any such user giving access to an intruder in exchange for a bribe or other favour.

3.9 Platform Used

Windows 10 is a major version of the Microsoft Windows operating system that was released on July 29, 2015.It is built on the Windows NT kernel and follows windows 8.Part of the reason Microsoft decided to name the 2015 release "Windows 10"(and skipped "windows 9")is because the operating system is designed to be a new direction for Microsoft. One of the primary aims of windows 10 is to Unify the windows experience across multiple devices, such desktop computers, tablets, and smartphones. As part of this effort, Microsoft developed Windows10 Mobile alongside Windows 10 to replaces Windows Phone

- Microsoft's previous mobile OS. Windows 10 also integrates other Microsoft services, such as Xbox Live and the Cortana voice recognition assistant.

3.10 Technologies Used

Python Django

Django is a high level Python web framework that enables rapid developement of secure,scalable web applications,emphasizing clean pragmatic design and the principle of "don't repeat yourself"(DRY).It includes built-in features for authentication,database modeling,URL routing,templating and more facilitating efficient development and maintenance of complex web projects.

CSV

Comma-separated values (CSV) is a text file format that uses commas to separate values. A CSV file stores tabular data (numbers and text) in plain text, where each line of the file typically represents one data record. Each record consists of the same number of fields, and these are separated by commas in the CSV file.

HTML

HTML, or Hypertext Markup Language, is a markup language for the web that defines the structure of web pages.It is one of the most basic building blocks of every website, so it's crucial to learn if you want to have a career in web development.This structure alone is not enough to make a web page look good and interactive. So you'll use assisted technologies such as CSS and JavaScript etc to make your HTML beautiful and add interactivity, respectively.

CSS

CSS stands for Cascading Style Sheets.It describes how HTML elements are to be displayed on screen, paper, or in other media. CSS saves a lot of work. It can control the layout of multiple web pages all at once. External stylesheets are stored in CSS files.

Bootstrap

Bootstrap is a free, open source front-end development framework for the creation of websites and web apps. Designed to enable responsive development of mobile-first websites, Bootstrap provides a collection of syntax for template designs.

Chapter 4

4 Design Document

4.1 Purpose

The Air Quality Index is a web application. The main purpose of this web application is to develop an Air quality indexing and prediction system that harness the power of data and predictive analytics to provide real-time insights to air quality.The benefit of this web application is very high because the air pollution has become a most pressing concern and it has affected not only the environment but also the health of billions of people.This web application is easy to use and aesthetic in appearance.The user can register and view the details of the air quality.The web application is easy to handle registration and predicting the accurate air quality. Purpose of this document is to give the detailed description of the architecture and system design for the software.The system design document shows the software will be structured to satisfy the requirement identified in the software requirement specification.It is a translation of requirements into a description of the software structure,software component,interfaces and data necessary for the implementation phase.In a complete software design document,each requirements must be traceable to one or more design entities.

4.2 Scope

The Air Quality Index is a web application which helps in predicting the accurate air quality an .Which also helps in leading to a more healthier and more sustainable future .

4.3 Overview

The purpose of this document is to help the reader to visualize the solution to the project presented.This document verifies how the design meet the requirement stimulated in the SRS document through design viewpoints.The design viewpoints will cover all design elements presented before. This document will provide a direct approach to the development of this project hence reducing feature creep and ponitedly determine the quality of the design.

4.4 Data Design

Database are the storehouses of data used in the software systems.The data is stored in tables inside the database.Several tables are created for the manipulation of the data for the system.Two essential settings for a database are

- Primary Key- The field that is unique for all the record occurrence.
- Foreign Key- The field used to set relation between tables.

Normalization

- Normalization is the process of organizing the data in the database.
- Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate the undesirable characteristics like Insertion, Update and Deletion Anomalies.
- Normalization divides the larger table into the smaller table and links them using relationship.
- The normal form is used to reduce redundancy from the database table.

Tables

Login Form

Name	DataType	Constraints	Description
Username	varchar(45)	Primarykey	to enter name of user
Password	text	Notnull	to set password

Registration Form

Name	DataType	Constraints	Description
FirstName	varchar(50)	Primarykey	To enter first name of the person
LastName	varchar(45)	Notnull	To enter last name of the person
Username	varchar(50)	Notnull	Name of the users
Email	varchar(100)	Notnull	Email of the users
password	text	Notnul	Password of users
Confirm Password	text	Notnull	Confirm the Password of users

Chapter 5

5 Development of the System

This website can be decomposed into a number of submodules. The submodules of web application are Login Form, Registration Form and Prediction Window. Each sub module have specific objectives,to avoid unwanted coupling between modules and to increase cohesion, modules are again decomposed into submodules.

Chapter 6

6 System Testing

Testing is the penultimate step of software development.An elaborate testing of the data is prepared and the system is using the test data,while doing testing,errors are noted and correction is made.The users are trained to operate the developed system.Both hardware and software securities are made to run the developed system successfully.System testing is aimed at ensuring the system works accurately before the live operation commences.

6.1 Test Plan

6.1.1 Scope

This test plan will cover the following testing activities as identified in the testing strategy.

- White Box Testing

White box testing is a testing technique, that examines the program structure and derives test data from the program logic/code. In white box testing, the UI is bypassed. Inputs and outputs of ratings are tested directly at the code level and the results are compared against specifications. It Reveals errors in "hidden" code.

- Black Box Testing

Black - box testing is a method of software testing that examines the functionality of an application based on the specifications. Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end - user would.

- Unit Testing

The module includes four main components to undergo unit testing. It checks the syntax error, logic error and validity of the program correctness. This test will be performed by the developers.

- Integration Testing

After all components pass the unit testing, they get to be tested whether they work correctly when they are running concurrently and communicating to each other. Integration tests exercise an entire subsystem and ensure that a set of components play nicely together.

- Internal Data Testing

We will test the validity of the data before it enters the database to avoid any problems that may face us in the database. We will test the encryption of the personal information of all the users along with the user names and passwords to ensure maximum security of the users privacy.

6.1.2 Software risk issues

In this section, the plan is to test the risk involved in critical issues such as: Difficult to run on visual studio due to large loading time. Some inherent software risks such as complexity exist; also these issues need to be identified. Proper network connection Working of Mysql database

6.1.3 Features to be tested

- Test whether correct user name and password allows you to login.
- Test whether invalid user name and password prevents you from login.
- Test whether there is any connection problem in the Server.
- Test whether the student and admin details are entered correctly.
- Test whether invalid data entry allows saving data successfully.
- Test whether all pages are loaded correctly.
- Test whether watermark is embedded and extracted properly.
- Test whether the provided security works correctly.

6.2 Test consolidation

6.2.1 Test item

The items or features to be tested in the test cases are included in the document. Each and every user input is tested. The present condition of the system is tested. It is checked to make sure that environment is ready for the application to work.

6.2.2 Input specifications

These are the inputs required to execute the test case. All required inputs including data elements and values are given below.

Data	Values
Email	A valid email ID
Password	Combination of characters and numbers
Confirm Password	Combination of characters and numbers

Chapter 7

7 System Implementation and Maintenance

7.1 Implementation

System implementation is the conversion of new system into an operating one which involves creating compatible files, training clients and installing hardware. User training is crucial for minimizing resistance to change and giving chance to prove its worth. Training aids user friendly manuals and healthy screens provide the user with a good start. Software maintenance follows conversion to the extent that changes are necessary to maintain satisfactory operations relative to changes in the user's environment. Maintenance often includes minor enhancements or corrections to the problem that surface late in the systems operations. In the implementation phase, the team builds the components either from scratch or by composition. Given the architecture document meant from the design phase and the requirement document from the analysis phase, the team should build exactly what has been requested, though there is still room for innovation and flexibility. For example, a component may be narrowly designed for this particular system, or the component may be made more general to satisfy a reusability

- Careful planning
- Investigation of system and constraints
- Design the methods to achieve changeover.
- Training the staff in the changed phase.
- Evaluation of change over method.
- The method of implementation and time scale to be adopted are found out initially.

7.2 Maintenance

This phase occurs as a result of deploying the whole system at the end users organization. They will perform the beta testing at the end users and inform to the developers about any needed modification to the application. The customer records all the problems that are encountered during the beta testing and reports these to the developer at regular intervals.

7.2.1 Corrective Maintenance

Even with the best quality assurance activities, it is likely that the customer will uncover defects in the software. Corrective maintenance changes the software to correct the defects. Corrective Maintenance activity may consist of repair,

restoration or replacement of equipment. This activity will be result of a regular inspection, which identifies the failure in time for corrective maintenance to be planned and scheduled, then performed during a routine maintenance shutdown.

7.2.2 Adaptive Maintenance

Over time, the original environment(CPU, operating system, business rules, external product characteristics) for which the software was developed is likely to change. Adaptive maintenance results in modification to the software to accommodate changes to its external environment.

7.2.3 Enhanced Maintenance

As software is used, the customer/user will recognize additional functions that will provide the benefit. Perfect maintenance extends the software beyond its original functional requirements.

7.2.4 Preventive Maintenance

Computer software deteriorates due to change, and because of this preventive maintenance often called software re-engineering, must be conducted to enable the software to serve the needs of its end users. Preventive maintenance makes changes to computer programs so that they can be more easily corrected, adapted and enhanced.

Chapter 8

8 Conclusion and Future Scope

8.1 Conclusion

In conclusion, our "Air Quality Indexing and Prediction" project represents a significant advancement in the field of environmental health and public well-being. This project's primary objective was to develop a comprehensive system that enables accurate air quality prediction, real-time assessment, and informed decision-making.

8.2 Future Scope

- Our "Air Quality Indexing and Prediction" project has laid a strong foundation for future enhancements and expansions. As technology continues to evolve and environmental challenges persist, there are several exciting possibilities for further development and improvement.

Appendix

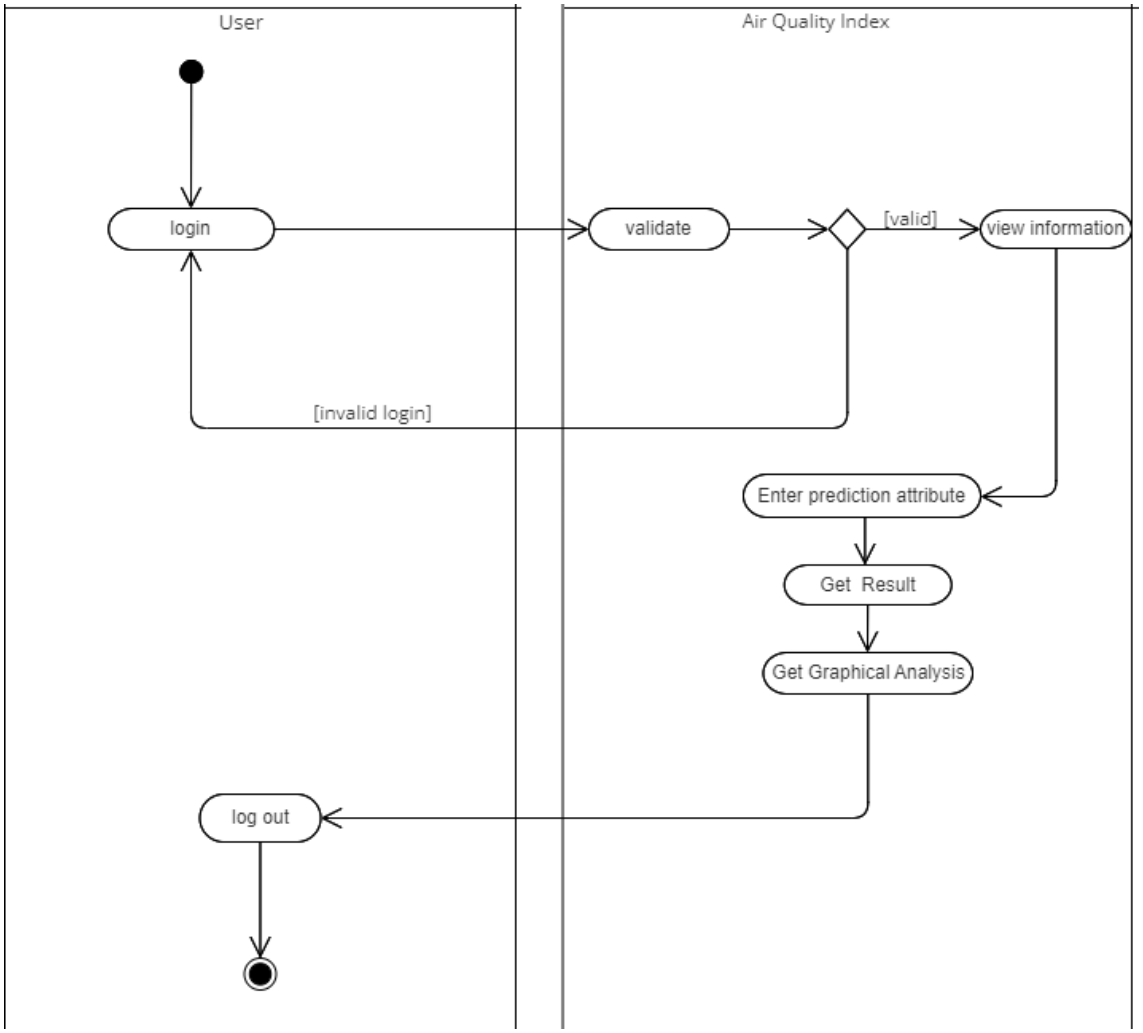
A Activity Diagram

Activity diagrams show the flow of one activity to another within a system or process. Even complex systems can be visualized using activity diagrams. They are used within organizations to model customer journeys, to show the process of receiving an order through shipping to the customer, and to model sales processes.

The Uses of Activity Diagram :

- We describe what causes a particular event using an activity diagram.
- An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed.
- They are used in business and process modeling where their primary use is to depict the dynamic aspects of a system.
- Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques.
- It does not show any message flow from one activity to another. Activity diagram is sometimes considered as the flowchart.

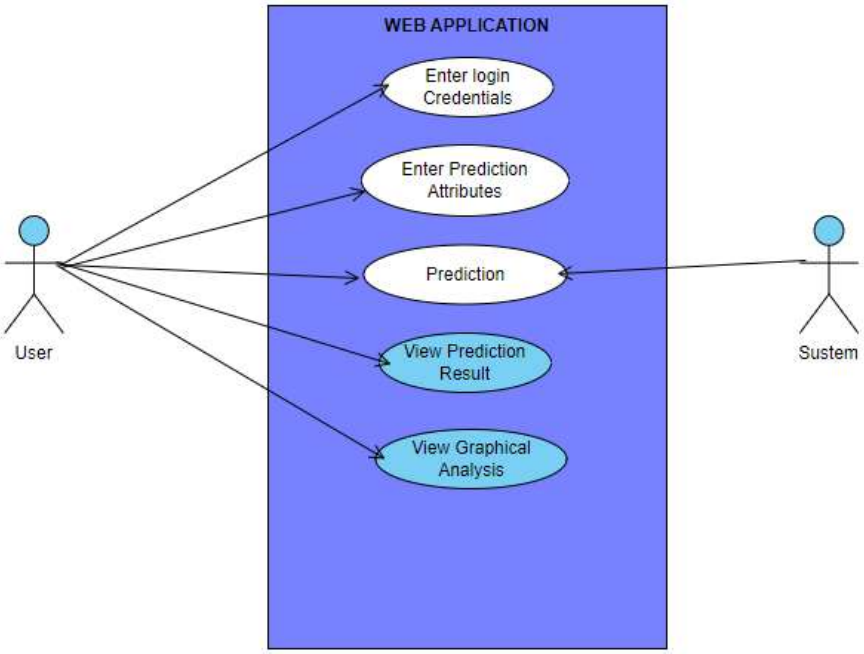
Activity Diagram for Air Quality Index Using Machine Learning



USE CASE DIAGRAM

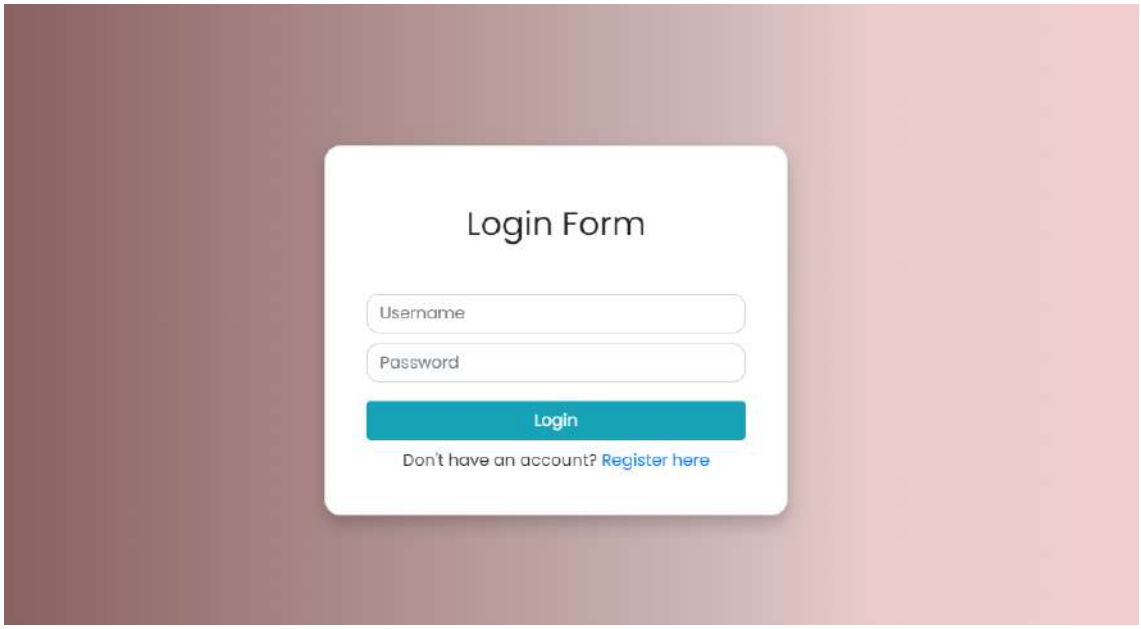
A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.

USE CASE DIAGRAM FOR AIR QUALITY INDEX

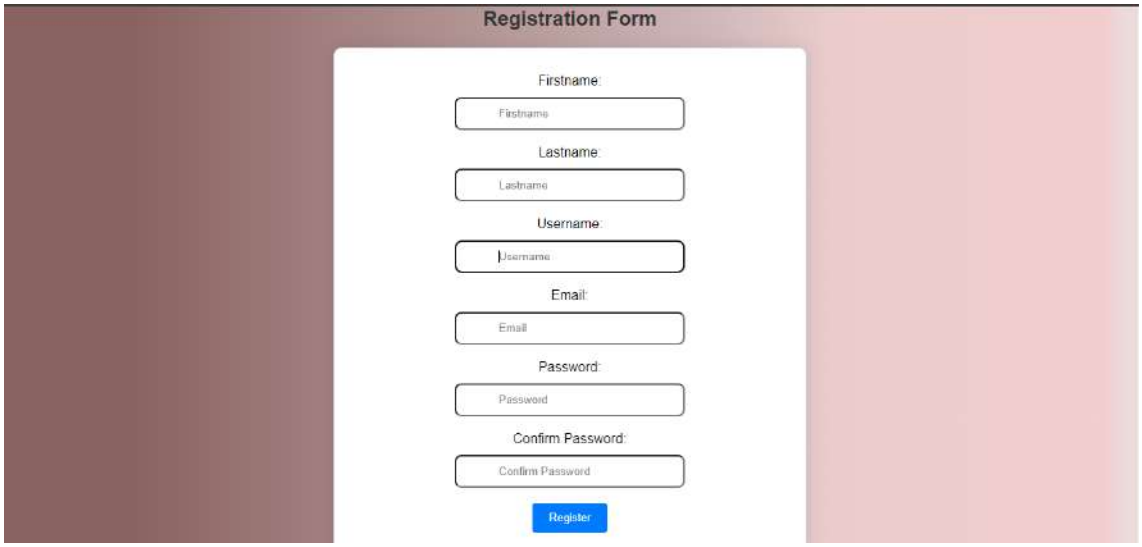


B USER INTERFACES

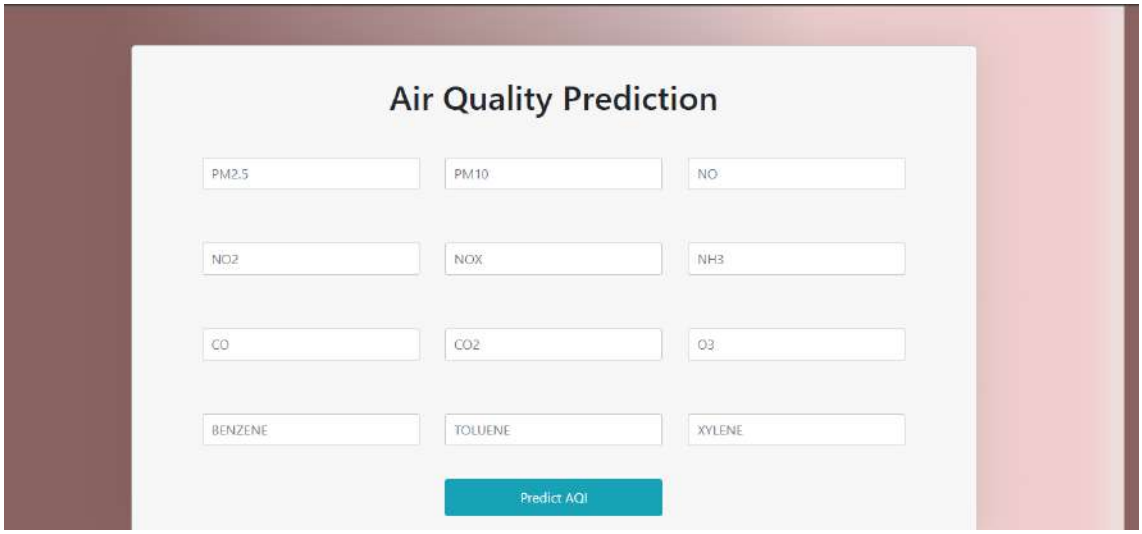
B.1 LOGIN FORM



B.2 REGISTRATON FORM



B.3 PREDICTION WINDOW



C CODE

Login page

```
[breaklines=true]
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Air Quality Detection System - Login</title>
  <link rel="stylesheet"
      href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
  <style>
    @import url
        ('https://fonts.googleapis.com/css2?family=Agdasima&family=Poppins&display=swap');
body {
    font-family: 'Poppins', sans-serif;
    margin: 0;
    padding: 0;
    background: rgb(137,98,97);
    background: linear-gradient(90deg, rgba(137,98,97,1) 11
%, rgba(155,120,119,1) 24%, rgba(169,137,136,1) 34
%, rgba(180,150,150,1) 42%, rgba(208,184,184,1) 62
%, rgb(235, 204, 204) 73%, rgb(242, 206, 206) 97%, rgb(237, 220, 220) 99%);
    display: flex;
    justify-content: center;
    align-items: center;
    min-height: 100vh;
  }

  .container {
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100%;
  }

  .card {
    width: 450px;
    background-color: white;
    border-radius: 15px;
    padding: 40px;
    backdrop-filter: blur(10px);
    box-shadow: 0 8px 16px rgba(0, 0, 0, 0.2);
  }
```

```
h2 {

  text-align: center;
  margin-bottom: 20px;
}

form {
  display: flex;
  flex-direction: column;
}

label {

  margin-bottom: 5px;
}

input {
  padding: 10px;
  margin-bottom: 10px;
  border: none;
  border-radius: 5px;
  background-color: rgba(255, 255, 255, 0.8);
}

button {
  padding: 10px;
  background-color: #fff;
  color: #498ffc;
  border: none;
  border-radius: 5px;
  cursor: pointer;
}

button:hover {
  background-color: #70c1ff;
}

@media (max-width: 580px) {
  .card {
    width: 100%;
    max-width: 450px;
  }
}

  </style>
```

```
</head>
<body>
    <div class="container">
        <div class="card">
          <h2 class="mt-3 mb-5">Login Form</h2>
          <form method="post" action="{% url 'log' %}">
            {% csrf_token %}
            <!-- <label for="username">Username</label> -->
            {{form.username.error}}
                                {{form.username}}
            <!-- <label for="password">Password</label> -->
            {{form.password.error}}
            {{form.password}}

            <button type="submit" class="mt-2 btn btn-info">Login</button>
          </form>
          <div class="register-link text-center mt-2">
            Don't have an account? <a href="{% url 'reg' %}">Register here</a>
        </div>
        </div>
      </div>
    <!-- <div class="container mt-5">

        <div class="row justify-content-center">
          <div class="col-md-6">
              <div class="card mt-5">
                  <div class="card-header">
                      <h2 class="text-center">Login</h2>
                   </div>
                   <div class="card-body">
                      <form method="post" action="{% url 'log' %}">
                          {% csrf_token %}
                         <div class="form-group">
                            <label for="username">Username:</label>
                             {{form.username.error}}
                             {{form.username}}
                          </div>
                           <div class="form-group">
                              <label for="password">Password:</label>
                               {{form.password.error}}
                               {{form.password}}
                          </div>
                         <button
     type="submit" class="btn btn-primary btn-block">Login</button>
                      </form>
                  </div>
```

```
              </div>
           </div>
       </div>
       <div class="register-link text-center">
         Don't have an account? <a href="{% url 'reg' %}">Register here</a>
       </div>
  </div> -->
</body>
</html>
```

Registration Form

```
<!DOCTYPE html>
<html>
<head>
    <title>Registration Form</title>
    <style>
        body {
            font-family: 'Poppins', sans-serif;
            background: rgb(137,98,97);
             background: linear-gradient(90deg, rgba(137,98,97,1) 11
%, rgba(155,120,119,1) 24%, rgba(169,137,136,1) 34
%, rgba(180,150,150,1) 42%, rgba(208,184,184,1) 62
%, rgb(235, 204, 204) 73%, rgb(242, 206, 206) 97%, rgb(237, 220, 220) 99%);

            text-align: center;
        }
        h2 {
            color: #333;
        }
        form {
            width: 500px;
            margin: 0 auto;
            background-color: #fff;
            padding: 30px;
            border: 1px solid #ccc;
            border-radius: 10px;
            box-shadow: 0 0 20px rgba(0, 0, 0, 0.1);
        }
        label {
            display: block;
            margin-bottom: 10px;
        }
```

```
        input[type="submit"] {
            background-color: #007bff;
            color: #fff;
            padding: 10px 20px;
            border: none;
            border-radius: 3px;
            cursor: pointer;
        }
        input[type="submit"]:hover {
            background-color: #0056b3;
        }
    </style>
</head>
<body>
    <h2>Registration Form</h2>
    <form action="" method="post">
        {% csrf_token %}
        <label for="first_name">Firstname:</label>
        {{form.first_name.error}}
        {{form.first_name}}
        <br><br>
        <label for="last_name">Lastname:</label>
        {{form.last_name.error}}
        {{form.last_name}}
        <br><br>
        <label for="username">Username:</label>
        {{form.username.error}}
        {{form.username}}
        <br><br>

        <label for="email">Email:</label>
        {{form.email.error}}
        {{form.email}}
        <br><br>

        <label for="password">Password:</label>
        {{form.password1.error}}
        {{form.password1}}
        <br><br>

        <label for="confirm_password">Confirm Password:</label>
        {{form.password2.error}}
        {{form.password2}}
        <br><br>
```

```
      <input  type="submit" value="Register">
      <div class=
        "text-center" style=
        "font-family: Cambria,Cochin,Georgia,Times,'Times New Roman',serif;padding:10px 0px
          Back to<a href="{% url 'log' %}"> Login</a>
        </div>
    </form>
</body>
</html>
```

lndex.html

```
<!doctype html>
<html lang="en">
  <head>
    <title>AQI</title>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!-- Optional JavaScript -->
    <!-- jQuery first, then Popper.js, then Bootstrap JS -->
    <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
 integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>
    <script src=
"https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"
 integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
 crossorigin="anonymous"></script>
    <script src=
"https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
 integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
 crossorigin="anonymous"></script>
    <!-- Bootstrap CSS -->
    <link rel=
"stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css'
 integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
 crossorigin="anonymous">
    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
    <link rel="stylesheet"
 href=
"https://fonts.googleapis.com/css2
?family=Material+Symbols+Outlined:opsz,wght,FILL,GRAD@40,700,0,200" />
    {% load static %}
    <style>
        form {
```

```
            background-color: #f7f7f7;
            padding: 20px;
            border: 1px solid #ccc;
            border-radius: 5px;
            box-shadow: 0 0 50px rgba(0, 0, 0, 0.1);
        }
        body {
            background: rgb(137,98,97);
background: linear-gradient(90deg, rgba(137,98,97,1) 11
%, rgba(155,120,119,1) 24
%, rgba(169,137,136,1) 34
%, rgba(180,150,150,1) 42%, rgba(208,184,184,1) 62
%, rgb(235, 204, 204) 73%, rgb(242, 206, 206) 97%, rgb(237, 220, 220) 99%);
            background-size: cover;
            background-color: #c6c2c2;
        }
        form {
            margin: 0 auto;
            max-width: 1000px;
            margin-top: 3%;
        }
        .col {
            padding: 10x;
        }
        input[type="text"],
        input[type="number"] {
            width: 100%;
            padding: 10px;
            margin-bottom: 15px;
            border: 1px solid #ccc;
            border-radius: 3px;
        }

        label {
            display: block;
            margin-bottom: 10px;
            font-weight: bold;
            color: #333;
        }
        br {
            margin: 10px 0;
        }
        button.btn {
            color: #fff;
            padding: 10px 20px;
            border: none;
```

```
            border-radius: 3px;
            cursor: pointer;
        }

        button.btn:hover {
            background-color: #0056b3;
        }

        h2 {
            font-size: 1.5rem;
            color: #333;
            text-align: center;
            margin-top: 20px;
        }
        h6 {
            padding: 10px;

            margin-left: 95%;
            border-radius: o.75rem;
            text-decoration: none;
        }
    </style>

    </head>
  <body>
    <h6 class="mt-2"><a href="{% url 'log' %}">
<span class="material-symbols-outlined" style="color: white;">
        exit_to_app
        </span></a></h6>
    <form method="post" class="" autocomplete="on" id="myform">
        {% csrf_token %}
        <h1 class="text-center mt-3 mb-5 ">Air Quality Prediction</h1>

        <div class="row ml-5 mr-5 mt-5" >
            <div class="col" >
                {{form.pm25.error}}
                {{form.pm25}}
            </div>
            <div class="col">
                {{form.pm10.error}}
                {{form.pm10}}
            </div>
            <div class="col">
                {{form.no.error}}
                {{form.no}}
            </div>
```

```
</div><br><br>
<div class="row ml-5 mr-5">
    <div class="col">
        {{form.no2.error}}
         {{form.no2}}
    </div>
    <div class="col">
        {{form.nox.error}}
        {{form.nox}}
    </div>
    <div class="col">
        {{form.nh3.error}}
        {{form.nh3}}
    </div>
</div><br><br>
<div class="row ml-5 mr-5">
    <div class="col">
        {{form.co.error}}
        {{form.co}}
    </div>
    <div class="col">
        {{form.so2.error}}
        {{form.so2}}
    </div>
    <div class="col">
        {{form.o3.error}}
        {{form.o3}}
    </div>
</div><br><br>
<div class="row ml-5 mr-5">
    <div class="col">
        {{form.benzene.error}}
        {{form.benzene}}
    </div>
    <div class="col">
        {{form.toluene.error}}
        {{form.toluene}}
    </div>
    <div class="col">
        {{form.xylene.error}}
        {{form.xylene}}
    </div>
</div><br>
<div class="row ml-5 mr-5">
    <div class="col"></div>
    <div class="col ">
```

```
                <button type="submit" class="btn btn-info btn-block ">Predict AQI</button>
            </div>
            <div class="col"></div>
        </div>

    </form>
    <div class="row" style="width: 100%;">
          <div class="col">
            {% if prediction %}
            <h2 id="output">Predicted AQI: {{ prediction }}</h2>
            <div class="text-center mb-3">
            <img c; src="{{ plot_url }}" alt="Pie Chart" >
        {% endif %}
          </div>
        </div>
    </div>


    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
    <!-- <script>
        document.addEventListener("DOMContentLoaded", function() {
            setTimeout(function() {
                document.getElementById('output').style.display = 'none';
            }, 5000);
        });

        </script> -->

</body>
</html>
```

data.py

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
df=pd.read_csv('D:\Abhay_project\Air Quality\city_day.csv')
df

df.dtypes

df.columns
```

```
df.shape

df.info()

df.nunique()

df.describe()

df.dropna(inplace=True)
df=df.reset_index(drop=True)
df

# Select only numeric columns for correlation calculation
numeric_columns = df.select_dtypes(include=[np.number])

# Calculate the correlation matrix
correlation_matrix = numeric_columns.corr()

# Create a heatmap to visualize the correlations
plt.figure(figsize=(12, 10))
sns.heatmap(correlation_matrix, cmap="coolwarm", annot=True)

df_city_day = df.copy()
df_city_day.columns

pollutants = ['PM2.5', 'PM10', 'NO',
 'NO2', 'NOx', 'NH3','CO', 'SO2', 'O3',
 'Benzene', 'Toluene', 'Xylene']
df_city_day = df_city_day[pollutants]
print('Distribution of different pollutants in last 5 years')
df_city_day.plot(kind='line',figsize=(18,18),cmap='coolwarm',subplots=True,fontsize=9)

x=df.drop(['City','Date','AQI_Bucket'],axis=1).values
y=df['AQI_Bucket'].values
x

y

from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=.3,random_state=42)
x_train

x_test

y_train
```

```
y_test

from sklearn.neighbors import KNeighborsClassifier
model=KNeighborsClassifier()
model.fit(x_train,y_train)
y_pred=model.predict(x_test)
y_pred

df1=pd.DataFrame({'y_test':y_test,'y_pred':y_pred})
df1

from sklearn.metrics import accuracy_score,r2_score,confusion_matrix,classification_report
print('confusion_matrix','\n',confusion_matrix(y_test,y_pred))
print('*'*100)
print('accuracy_score is:',accuracy_score(y_test,y_pred))
print('*'*100)
print('classification report:',classification_report(y_test,y_pred))
```

views.py

```
from django.shortcuts import render,redirect
from .forms import AirQualityCheckForm
import pandas as pd
from .models import *
from .forms import *
from django.views.generic import FormView,CreateView,TemplateView
from django.contrib.auth.models import User
from django.contrib.auth import authenticate,login
from django.urls import reverse_lazy



class LoginView(FormView):
    template_name="login.html"
    form_class=LogForm
    def post(self,request,*args,**kwargs):
        log_form=LogForm(data=request.POST)
        if log_form.is_valid():
            us=log_form.cleaned_data.get('username')
            ps=log_form.cleaned_data.get('password')
            user=authenticate(request,username=us,password=ps)
            if user:
                login(request,user)
```

```
                return redirect('h')
            else:
                return render(request,'login.html',{"form":log_form})
        else:
            return render(request,'login.html',{"form":log_form})


class RegView(CreateView):
     form_class=Reg
     template_name="reg.html"
     model=User
     success_url=reverse_lazy("log")



# Load the CSV data
df = pd.read_csv('C:/Users/User/Downloads/Air Quality (2)/Air Quality/city_day.csv')

# Define the prediction function
def predict_aqi(pm25, pm10, no, no2, nox, nh3, co, so2, o3, benzene, toluene, xylene):
    # Use the input values to filter the DataFrame and calculate AQI
    filtered_data = df[
        (df['PM2.5'] == pm25) &
        (df['PM10'] == pm10) &
        (df['NO'] == no) &
        (df['NO2'] == no2) &
        (df['NOx'] == nox) &
        (df['NH3'] == nh3) &
        (df['CO'] == co) &
        (df['SO2'] == so2) &
        (df['O3'] == o3) &
        (df['Benzene'] == benzene) &
        (df['Toluene'] == toluene) &
        (df['Xylene'] == xylene)
    ]

    if not filtered_data.empty:
        prediction = filtered_data['AQI_Bucket'].values[0]
    else:
        prediction = "No matching data found"

    return prediction

def predict_air_quality(request):
    if request.method == 'POST':
        form = AirQualityCheckForm(request.POST)
```

```
        if form.is_valid():
            user_input = form.cleaned_data
            pm25 = user_input['pm25']
            pm10 = user_input['pm10']
            no = user_input['no']
            no2 = user_input['no2']
            nox = user_input['nox']
            nh3 = user_input['nh3']
            co = user_input['co']
            so2 = user_input['so2']
            o3 = user_input['o3']
            benzene = user_input['benzene']
            toluene = user_input['toluene']
            xylene = user_input['xylene']

            # Use the prediction function to predict AQI
            prediction = predict_aqi(pm25, pm10,
                  no, no2, nox, nh3, co, so2, o3, benzene, toluene, xylene)
            plot_url = create_pie_chart(pm25, pm10,
                 no,no2,nox,nh3,co,so2,o3,benzene,toluene,xylene)
            context = {
                'form': form,
                'prediction': prediction,
                'plot_url':plot_url
            }
            return render(request, 'index.html', context)
    else:
        form = AirQualityCheckForm()

    context = {
        'form': form,
    }
    return render(request, 'index.html', context)



from django.shortcuts import render
import matplotlib.pyplot as plt
from io import BytesIO
import base64



def create_pie_chart(pm25, pm10,no,no2,nox,nh3,co,so2,o3,benzene,toluene,x):
    plt.figure()
    labels = ['PM2.5', 'PM10', 'NO' ,'NO2','NOX','NH3','CO','SO2','O3','B','T','X']
```

```
    data = [pm25, pm10, no,no2,nox,nh3,co,so2,o3,benzene,toluene,x]
    colors = ['red', 'yellow', 'green','blue', 'gray', 'orange','cyan',
      'lightcoral', 'violet','black','greenyellow','brown']
    explode=(0.1,0.1,0.6,0.1,0.6,0.1,0.6,0.1,0.1,0.5,0,0.5)


    plt.pie(data, labels=labels, explode=explode, colors=colors,
        autopct='%1.1f%%', startangle=90)
    plt.legend()
    plt.axis('equal')

    # Save the plot to a BytesIO object
    image_stream = BytesIO()
    plt.savefig(image_stream, format='png')
    image_stream.seek(0)

    # Encode the image to base64 for embedding in HTML
    plot_url = base64.b64encode(image_stream.read()).decode('utf-8')

    return f'{plot_url}'
```

urls.py

```
[breaklines=true]
from django.urls import path
from .views import *

urlpatterns = [
    path('reg/',RegView.as_view(),name='reg'),
    path('predict/',predict_air_quality,name='h'),
]
```

Settings.py

```
"""
Django settings for quality project.

Generated by 'django-admin startproject' using Django 4.2.6.

For more information on this file, see
https://docs.djangoproject.com/en/4.2/topics/settings/

For the full list of settings and their values, see
```

```
https://docs.djangoproject.com/en/4.2/ref/settings/
"""

from pathlib import Path

# Build paths inside the project like this: BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent


# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/4.2/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'django-insecure-@f_-vc23$@iwqs-s%1d84j&^c+v6hl@v5bc$u@rf36l^vd3jn6'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = []


# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'setup'
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'quality.urls'

TEMPLATES = [
```

```
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]

WSGI_APPLICATION = 'quality.wsgi.application'


# Database
# https://docs.djangoproject.com/en/4.2/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
    }
}


# Password validation
# https://docs.djangoproject.com/en/4.2/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]
```

```
# Internationalization
# https://docs.djangoproject.com/en/4.2/topics/i18n/

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_TZ = True


# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/4.2/howto/static-files/

STATIC_URL = 'static/'

# Default primary key field type
# https://docs.djangoproject.com/en/4.2/ref/settings/#default-auto-field

DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
```

Forms.py

```
from django import forms
from django.contrib.auth.forms import UserCreationForm
from django.contrib.auth.models import User

class LogForm(forms.Form):
    username=forms.CharField
      (widget=forms.TextInput
      (attrs={"placeholder":"Username",
       "class":"form-control","style":"border-radius: 0.75rem; "}))
    password=forms.CharField
      (widget=forms.PasswordInput
      (attrs={"placeholder":"Password",
      "class":"form-control","style":"border-radius: 0.75rem; "}))


class Reg(UserCreationForm):
    class Meta:
        model=User
        fields=['first_name','last_name','email','username','password1','password2']
```

```
    first_name=forms.CharField
     (widget=forms.TextInput
    (attrs=
  {"placeholder":"Firstname",
     "class":"form-control","style":"border-radius: 0.5rem;padding:10px 50px;"}))
    last_name=forms.CharField
   (widget=forms.TextInput
    (attrs=
   {"placeholder":"Lastname",
    "class":"form-control","style":"border-radius: 0.5rem;padding:10px 50px; "}))
     username=forms.CharField
   (widget=forms.TextInput
   (attrs=
     {"placeholder":"Username",
     "class":"form-control","style":"border-radius: 0.5rem;padding:10px 50px; "}))
      email=forms.CharField
     (widget=forms.TextInput
      (attrs=
        {"placeholder":"Email",
        "class":"form-control","style":"border-radius: 0.5rem;padding:10px 50px; "}))
    password1=forms.CharField
     (widget=forms.PasswordInput
     (attrs=
      {"placeholder":"Password",
       "class":"form-control","style":"border-radius: 0.5rem;padding:10px 50px; "}))
    password2=forms.CharField
     (widget=forms.PasswordInput
      (attrs=
       {"placeholder":"Confirm Password",
        "class":"form-control","style":"border-radius: 0.5rem;padding:10px 50px; "}))
```

```
class AirQualityCheckForm(forms.Form):
    pm25 = forms.FloatField
                   (label='PM2.5',widget=forms.TextInput
                   (attrs={"placeholder":"PM2.5","class":"form-control","id":"pm25"}))
    pm10 = forms.FloatField
                  (label='PM10',widget=forms.TextInput
                  (attrs={"placeholder":"PM10","class":"form-control","id":"pm10"}))
    no = forms.FloatField
                  (label='NO',widget=forms.TextInput
                  (attrs={"placeholder":"NO","class":"form-control"}))
    no2 = forms.FloatField
                  (label='NO2',widget=forms.TextInput
```

```
                (attrs={"placeholder":"NO2","class":"form-control"}))
    nox = forms.FloatField
                (label='NOx',widget=forms.TextInput
                (attrs={"placeholder":"NOX","class":"form-control"}))
    nh3 = forms.FloatField
                (label='NH3',widget=forms.TextInput
                (attrs={"placeholder":"NH3","class":"form-control"}))
    co = forms.FloatField
                (label='CO',widget=forms.TextInput
                (attrs={"placeholder":"CO","class":"form-control"}))
    so2 = forms.FloatField
                (label='SO2',widget=forms.TextInput
                (attrs={"placeholder":"CO2","class":"form-control"}))
    o3 = forms.FloatField
                (label='O3',widget=forms.TextInput
                (attrs={"placeholder":"O3","class":"form-control"}))
    benzene = forms.FloatField
                (label='Benzene',widget=forms.TextInput
                (attrs={"placeholder":"BENZENE","class":"form-control"}))
    toluene = forms.FloatField
                (label='Toluene',widget=forms.TextInput
                (attrs={"placeholder":"TOLUENE","class":"form-control"}))
    xylene = forms.FloatField
                (label='Xylene',widget=forms.TextInput
                (attrs={"placeholder":"XYLENE","class":"form-control"}))
```

Model.py

```
[breaklines=true]

from django.db import models

class AirQualityData(models.Model):
    city = models.CharField(max_length=100)
    date = models.DateField(auto_now_add=True)
    aqi_bucket = models.CharField(max_length=20)
    pm25 = models.FloatField()
    pm10 = models.FloatField()
    no = models.FloatField()
    no2 = models.FloatField()
    nox = models.FloatField()
    nh3 = models.FloatField()
```

```
    co = models.FloatField()
    so2 = models.FloatField()
    o3 = models.FloatField()
    benzene = models.FloatField()
    toluene = models.FloatField()
    xylene = models.FloatField()
    aqi = models.CharField(max_length=255)

    def __str__(self):
        return f"{self.city} - {self.date}"
```

WEED DETECTION WEBSITE

PROJECT REPORT

Submitted By

AHALYA M

Reg. No. CCAVBCA002

For the award of the Degree of

Bachelor of computer application (BCA)

(University of Calicut)

under the guidance of

Ms. Sini Thomas

Head of the Department

BACHELOR OF COMPUTER APPLICATION
DEPARTMENT OF COMPUTER SCIENCE
CHRIST COLLEGE (AUTONOMOUS)
IRINJALAKUDA, KERALA

2021-2024

DEPARTMENT OF COMPUTER SCIENCE

CHRIST COLLEGE (AUTONOMOUS)

IRINJALAKUDA

CERTIFICATE

This is to certify that the project report entitled ***"Weed Detection Website"*** *is a bonfied record of the project work done by* ***Ahalya M*** *in partial fulfillment of the requirement for the sixth semester of* ***Bachelor of Computer Application*** *in Department of Computer Science of* **CHRIST COLLEGE (AUTONOMOUS) IRINJALAKUDA**.

Ms. Sini Thomas
Head of the Department
Internal Guide

Ms. Sini Thomas
Head of the Department
Computer Science

EXTERNAL EXAMINER

INTERNAL EXAMINER

DECLARATION

I hereby declare that this project work **"WEED DETECTION WEBSITE"** submitted to Christ College (Autonomous) Irinjalakuda, affiliated to Calicut University in partial fulfillment of the requirement for the award of the Bachelor of Computer Application, is a record of original work done by me, under the guidance of Ms. SINI THOMAS, Department of computer Science.

Place: Irinjalakuda AHALYA M

ACKNOWLEDGEMENT

First and foremost I would like to thank Lord almighty for his providence and for being the guiding light throughout the project. We wish to express my sincere gratitude to our beloved Department head for giving me all the facilities for our project. I take this opportunity to express my gratitude to the class teacher Ms. SOWMYA P.S and Head of the Department Ms. SINI THOMAS who has supported me throughout the course of this project. I am thankful for her aspiring guidance and valuable advice during the project work. I express my sincere thanks to my project guide Ms. SINI THOMAS for supporting and guiding throughout the project.I would take this opportunity to specially thank all other faculty members for their constant and continuous motivation. Finally i would like to thank my family and friends for giving valuable advice and moral support throughout my project.

ABSTRACT

WEED DETECTION WEBSITE offers a user-friendly platform for real-time analysis of agricultural field images. Utilizing machine learning algorithms, the site detects weeds and identifies crops, providing immediate feedback on weed presence and crop recognition. With a focus on efficiency and sustainability, the platform aims to enhance agricultural productivity and reduce herbicide usage. Ongoing development focuses on accuracy refinement and functionality expansion to become a key tool in weed management practices.

Contents

Chapter 1

1 Introduction

Welcome to our groundbreaking project at the intersection of agriculture and technology, where we harness the power of machine learning to revolutionize weed control practices. In response to the pressing need for sustainable solutions in modern agriculture, our project aims to develop an intelligent weed detection and management system. Weeds represent a persistent threat to crop yield and quality, requiring effective and eco-friendly control measures. Traditional methods fall short in terms of efficiency, cost-effectiveness, and environmental impact. By embracing machine learning algorithms, we aspire to address these challenges and pave the way for a more resilient and sustainable agricultural future. Our project focuses on leveraging machine learning techniques, such as computer vision and deep learning, to accurately identify and classify weeds in agricultural fields. Integrated with cutting-edge technologies, including drones and autonomous machinery, oursystem will enable real-time weed detection and targeted intervention. Through collaboration with farmers, researchers, and industry partners, we seek to deliver practical solutions that empower agricultural stakeholders to optimize crop production while minimizing environmental harm. Join us on this journey as we harness the power of machine learning to cultivate innovation and sustainability in agriculture.

1.1 Overview

The primary objective of this project is to develop a machine learning-based system for automated weed detection in agricultural fields. By leveraging machine learning algorithms, the system aims to accurately identify and differentiate between crops and weeds in real-time, facilitating timely and targeted weed management interventions.

Chapter 2

2 System Analysis

2.1 Purpose

The primary purpose is to develop a reliable and efficient system for automated weed detection in agricultural fields. By accurately identifying and mapping weeds, farmers can implement targeted weed management strategies, leading to reduced weed competition and improved crop health.

2.1.1 Proposed System

The project seeks to promote sustainable farming practices by reducing the reliance on chemical herbicides. By enabling precision weed control through machine learning, the project contributes to minimizing the environmental impact associated with herbicide use, such as soil contamination and water pollution. It collects high-resolution images of agricultural fields using drones, satellites, or ground-based sensors.

2.2 Problem definition

To know what the problem is and what the needs are before developing it. Here we design a website for weed detection.

2.3 FEASIBILITY STUDY

After the problem is clearly understood and solutions proposed, the next step is to conduct the feasibility study. Feasibility study is defined as evaluation or analysis of the potential impact of the proposed project or program. The objective is to determine whether the proposed system is feasible. There are three aspects of feasibility study to which the proposed system is subjected as discussed below.

2.3.1 Technical Feasibility

The proposed project of weed detection using machine learning demonstrates strong technical feasibility, leveraging advancements in data acquisition, machine learning algorithms, and computational resources. High-resolution imagery of agricultural fields, obtained through drones, satellites, or ground-based sensors, provides ample data for training machine learning models.

2.3.2 Economical Feasibility

The proposed project of weed detection using machine learning exhibits promising economical feasibility, driven by the potential for cost savings and increased

efficiency in agricultural weed management practices. By automating weed detection through machine learning, farmers can reduce labor costs associated with manual weed control and minimize expenses on chemical herbicides.

2.3.3 Operational Feasibility

The operational feasibility of the proposed project on weed detection using machine learning is promising, facilitated by the availability of scalable technology solutions and the adaptability of modern farming practices. With the increasing adoption of precision agriculture techniques, farmers are increasingly open to integrating advanced technologies into their operations.

Chapter 3

3 Software Requirement Specification

3.1 Purpose

The purpose of the Weed Detection Website project is to provide a user-friendly and accessible platform for farmers, agricultural researchers, and land managers to detect and assess the presence of weeds in images of agricultural fields. Through the integration of machine learning algorithms, users can upload images to the website and receive real-time analysis indicating the percentage of weeds present in the image, as well as whether the image contains crops.

3.2 Scope

The scope of the Weed Detection Website project encompasses the development of a user-friendly web platform allowing farmers, agricultural researchers, and land managers to upload images of agricultural fields for real-time analysis. The platform will utilize machine learning algorithms to detect weeds within the uploaded images, providing users with immediate feedback on the percentage of weeds present and the recognition of crops. Key features include image upload functionality, weed detection, crop recognition, real-time analysis, results display, and accessibility across various devices. Exclusions consist of hardware development for image capture and advanced features like predictive modeling.

3.3 Overall Description

The Weed Detection Website project aims to revolutionize weed management in agriculture by providing a user-friendly online platform for real-time analysis of images from agricultural fields. Leveraging machine learning algorithms, the website allows users to upload images and receive instant feedback on the presence of weeds and recognition of crops. By empowering farmers, agricultural researchers, and land managers with accurate and accessible tools for weed detection, this project promises to enhance efficiency, reduce herbicide usage, and promote sustainable agricultural practices.

3.3.1 Product Perspective

The Weed Detection Website provides farmers and researchers with a user-friendly platform to upload images of agricultural fields for instant analysis. Using machine learning algorithms, the site detects weeds and recognizes crops, aiding in timely decision-making and sustainable weed management practices, ultimately leading to improved crop yields and environmental stewardship. Ongoing development aims to enhance accuracy and expand functionality for widespread adoption in agriculture.

3.3.2 Product Functionality

At its core, the platform allows users to effortlessly upload images of agricultural fields for analysis. Leveraging advanced machine learning algorithms, the website accurately detects the presence of weeds and identifies crops within the uploaded images in real-time.

3.3.3 Users and Characteristics

Users of the Weed Detection Website include farmers, agricultural researchers, and land managers. They rely on the platform to monitor and manage weed infestations, optimize crop yields, and promote sustainable land management practices. Key characteristics of users include a strong interest in technology for weed management, varying technical proficiency, and a commitment to enhancing agricultural productivity and environmental stewardship.

3.4 Specific Requirements

3.4.1 Hardware Requirements

- System: IBM-Compatible PC
- Processor: Pentium IV or above
- Speed: Above 1GHz
- RAM capacity: 512 MB
- Hard Dsk drive: 40 GB
- Keyboard: Standard
- Mouse: Standard
- Monitor: SVGA Color

3.4.2 Software Requirements

- Operating System:Windows or ubuntu
- Languages used: PHP
- Database : SQLite
- Technologies used: HTML,Javascript,CSS,Python

3.5 Functional Requirements

It contains one main module.

- User

User

The user or The students can register ar login the website and register for events.Also the user can view the rules and regulation of events provided in the website.And at the final the user can also get the result of each events in their login.

3.6 Non Functional Requirements

Non-functional requirements define the overall qualities or attributes of the resulting system. Non-functional system place restrictions on the product being developed, the developed process, and specify external constraints that the product must meet. Examples of non-functional requirements include safety, security, usability, reliability and performance requirements. Project management issues (costs, time and schedule) are often considered as non-functional requirements. The principal non - functional constraints which are relevant to critical systems :

- performance
- security
- safety
- usability

Performance

Performance requirements concern the speed of operation of a system. Types of performance requirements :

- Response requirements (how quickly the system reacts to a user input).
- Throughput requirements (how much the system can accomplish within a specified amount of time).
- Availability requirements (is the system available for service when requested by end users). The speed of operation of this system is adequate for the requirements.

Reliability

- Reliability is the ability of a system to perform its required function under stated conditions for a specified period of time.
- constraints on the runtime behavior of the system. This system is reliable because its functionalities can be done on the required conditions.

Safety

Safety requirements are not required which exclude unsafe situation from the possible solution of the system.

Usability

Usability is the ease with which a user can learn to operate, prepare inputs for, and interpret outputs of system or components. Usability requirements include :

- information error messages.
- well-formed user interfaces.

3.7 Interface Requirements

3.7.1 Hardware interfaces

The system must run over the internet, all the hardware shall require to connect internet will be hardware interface for the system. As for example modem, WAN-LAN.

3.7.2 Software interfaces

Software interface required for the working of the project is the appropriate operating system.

3.7.3 Communication interfaces

The user of the site communicate using the network connectivity and the data set is accessed

3.8 Security Requirements

- User accesses only their account.
- Validation of input is handled.
- This application containing the computer systems is physically secured against arms or surreptitious entry by intruders.
- Users must be authorized carefully to reduce changes of any such user giving access to an intruder in exchange for a bribe or other favour.

3.9 Platform Used

Windows 10 is a major version of the Microsoft Windows operating system that was released on July 29, 2015.It is built on the Windows NT kernel and follows windows 8.Part of the reason Microsoft decided to name the 2015 release "Windows 10"(and skipped "windows 9")is because the operating system is designed to be a new direction for Microsoft. One of the primary aims of windows 10 is to Unify the windows experience across multiple devices, such desktop computers, tablets, and smartphones. As part of this effort, Microsoft developed Windows10 Mobile alongside Windows 10 to replaces Windows Phone

- Microsoft's previous mobile OS. Windows 10 also integrates other Microsoft services, such as Xbox Live and the Cortana voice recognition assistant.

3.10 Technologies Used

PHP

PHP (recursive acronym for PHP: Hypertext Preprocessor) is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML. What distinguishes PHP from something like client-side Javascript is that the code is executed on the server, generating HTML which is then sent to the client. The client would receive the results of that script, but would not know what the underlying code was. You can even configure your web server to process all your HTML files with PHP, and then there's really no way that users can tell what you have up your sleeve. The best things in using PHP are that it is extremely simple for a newcomer, but offers many advanced features for a professional programmer.

SQLite

SQLite is a lightweight, serverless, self-contained, and open-source relational database management system (RDBMS) widely used in various applications, including mobile apps, desktop software, and web browsers. Unlike traditional client-server databases, SQLite operates without the need for a separate server process, allowing it to be embedded directly into applications and requiring minimal configuration and administration. It stores data in a single file, making it highly portable and easy to deploy. SQLite supports standard SQL syntax and provides features such as ACID transactions, indexes, and triggers, making it suitable for a wide range of use cases. Its small footprint, simplicity, and efficiency make it a popular choice for developers seeking a reliable and scalable database solution for their applications.

What distinguishes PHP from something like client - side Javascript is that the code is executed on the server, generating HTML which is then sent to the client. The client would receive the results of running that script, but would not know what the underlying code was. You can even configure your web server to process all your HTML files with PHP, and then there's really no way that users can tell what you have up your sleeve. The best things in using PHP are that it is extremely simple for a newcomer, but offers many advanced features for a professional programmer.

Chapter 4

4 Design Document

4.1 Purpose

The website will feature an intuitive user interface for uploading images of agricultural fields. Upon upload, images will undergo analysis using machine learning algorithms for weed detection and crop recognition. The system will provide real-time feedback to users, displaying the percentage of weeds present and confirming the presence of crops. The website will be designed to be accessible across various devices and screen sizes. The architecture will include a front-end interface developed using HTML, CSS, and JavaScript, a back-end server implemented using Flask, and a SQLite database to store user and image data. The system will leverage TensorFlow for machine learning tasks and will be deployed using Docker for scalability and ease of deployment.

4.2 Scope

The platform will allow users to upload images and receive instant feedback on the percentage of weeds present and the recognition of crops. Key features include image upload functionality, weed detection, crop recognition, real-time analysis, results display, and accessibility across various devices. Ongoing development will focus on refining accuracy, expanding functionality, and ensuring seamless integration into users' workflows, with the ultimate goal of becoming an indispensable tool for weed management practices.

4.3 Overview

Leveraging machine learning algorithms, the website detects weeds and recognizes crops within the uploaded images, providing users with immediate feedback on the percentage of weeds present and the presence of crops. With an intuitive interface and accessibility across various devices, the platform aims to enhance efficiency, reduce herbicide usage, and promote sustainable agricultural practices. Ongoing development focuses on refining accuracy, expanding functionality, and ensuring seamless integration into users' workflows, with the goal of becoming a valuable tool in weed management practices.

4.4 Data Design

Database are the storehouses of data used in the software systems.The data is stored in tables inside the database.Several tables are created for the manipulation of the data for the system.Two essential settings for a database are

- Primary Key- The field that is unique for all the record occurrence.

- Foreign Key- The field used to set relation between tables.

Normalization

- Normalization is the process of organizing the data in the database.
- Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate the undesirable characteristics like Insertion, Update and Deletion Anomalies.
- Normalization divides the larger table into the smaller table and links them using relationship.
- The normal form is used to reduce redundancy from the database table.

Tables

User Details

Name	DataType	Constraints	Description
id	int(11)	Primarykey	ID
username	varchar(100)	Notnull	Name of user
email	varchar(250)	Notnul	mail of user
password	varchar(250)	Notnul	password of user
confirm password	varchar(250)	Notnul	confirmed password

Chapter 5

5 Development of the System

This website can be decomposed into a number of submodules. The submodules of website of weed detection is student. Each sub module have specific objectives,to avoid unwanted coupling between modules and to increase cohesion, modules are again decomposed into submodules.

Chapter 6

6 System Testing

Testing is the penultimate step of software development.An elaborate testing of the data is prepared and the system is using the test data,while doing testing,errors are noted and correction is made.The users are trained to operate the developed system.Both hardware and software securities are made to run the developed system successfully.System testing is aimed at ensuring the system works accurately before the live operation commences.

6.1 Test Plan

6.1.1 Scope

This test plan will cover the following testing activities as identified in the testing strategy.

- White Box Testing

White box testing is a testing technique, that examines the program structure and derives test data from the program logic/code. In white box testing, the UI is bypassed. Inputs and outputs of ratings are tested directly at the code level and the results are compared against specifications. It Reveals errors in "hidden" code.

- Black Box Testing

Black - box testing is a method of software testing that examines the functionality of an application based on the specifications. Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end - user would.

- Unit Testing

The module includes four main components to undergo unit testing. It checks the syntax error, logic error and validity of the program correctness. This test will be performed by the developers.

- Integration Testing

After all components pass the unit testing, they get to be tested whether they work correctly when they are running concurrently and communicating to each other. Integration tests exercise an entire subsystem and ensure that a set of components play nicely together.

- Internal Data Testing

We will test the validity of the data before it enters the database to avoid any problems that may face us in the database. We will test the encryption of the personal information of all the users along with the user names and passwords to ensure maximum security of the users privacy.

6.1.2 Software risk issues

In this section, the plan is to test the risk involved in critical issues such as: Difficult to run on visual studio due to large loading time. Some inherent software risks such as complexity exist; also these issues need to be identified. Proper network connection Working of Mysql database

6.1.3 Features to be tested

- Test whether correct user name and password allows you to login.
- Test whether invalid user name and password prevents you from login.
- Test whether there is any connection problem in the Server.
- Test whether the user details are entered correctly.
- Test whether invalid data entry allows saving data successfully.
- Test whether all pages are loaded correctly.
- Test whether the provided security works correctly.

6.2 Test consolidation

6.2.1 Test item

The items or features to be tested in the test cases are included in the document. Each and every user input is tested. The present condition of the system is tested. It is checked to make sure that environment is ready for the application to work.

6.2.2 Input specifications

These are the inputs required to execute the test case. All required inputs including data elements and values are given below.

Data	Values
Email	A valid email ID
Password	Combination of characters and numbers
Username	valid username

Chapter 7

7 System Implementation and Maintenance

7.1 Implementation

System implementation is the conversion of new system into an operating one which involves creating compatible files, training clients and installing hardware. User training is crucial for minimizing resistance to change and giving chance to prove its worth. Training aids user friendly manuals and healthy screens provide the user with a good start. Software maintenance follows conversion to the extent that changes are necessary to maintain satisfactory operations relative to changes in the user's environment. Maintenance often includes minor enhancements or corrections to the problem that surface late in the systems operations. In the implementation phase, the team builds the components either from scratch or by composition. Given the architecture document meant from the design phase and the requirement document from the analysis phase, the team should build exactly what has been requested, though there is still room for innovation and flexibility. For example, a component may be narrowly designed for this particular system, or the component may be made more general to satisfy a reusability

- Careful planning
- Investigation of system and constraints
- Design the methods to achieve changeover.
- Training the staff in the changed phase.
- Evaluation of change over method.
- The method of implementation and time scale to be adopted are found out initially.

7.2 Maintenance

This phase occurs as a result of deploying the whole system at the end users organization. They will perform the beta testing at the end users and inform to the developers about any needed modification to the application. The customer records all the problems that are encountered during the beta testing and reports these to the developer at regular intervals.

7.2.1 Corrective Maintenance

Even with the best quality assurance activities, it is likely that the customer will uncover defects in the software. Corrective maintenance changes the software to correct the defects. Corrective Maintenance activity may consist of repair,

restoration or replacement of equipment. This activity will be result of a regular inspection, which identifies the failure in time for corrective maintenance to be planned and scheduled, then performed during a routine maintenance shutdown.

7.2.2 Adaptive Maintenance

Over time, the original environment(CPU, operating system, business rules, external product characteristics) for which the software was developed is likely to change. Adaptive maintenance results in modification to the software to accommodate changes to its external environment.

7.2.3 Enhanced Maintenance

As software is used, the customer/user will recognize additional functions that will provide the benefit. Perfect maintenance extends the software beyond its original functional requirements.

7.2.4 Preventive Maintenance

Computer software deteriorates due to change, and because of this preventive maintenance often called software re-engineering, must be conducted to enable the software to serve the needs of its end users. Preventive maintenance makes changes to computer programs so that they can be more easily corrected, adapted and enhanced.

Chapter 8

8 Conclusion and Future Scope

8.1 Conclusion

The main objective of the project is to develop a system that can capture and scan images of crop fields and identify weeds from crops. Methodology: The project uses image processing and machine learning techniques to detect weeds. The image processing techniques include color segmentation, edge detection, and morphological operations. The machine learning techniques include feature extraction, classification, and localization. Dataset: The project uses a custom dataset of plant images, which contains images of different crop and weed species. The dataset is divided into training, validation, and testing sets.

8.2 Future Scope

- Live detection

Appendix

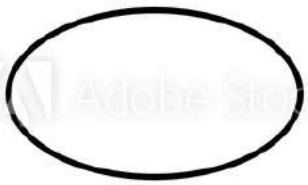
A ACTIVITY DIAGRAM

An activity diagram is a type of UML (Unified Modeling Language) diagram that visually represents the flow of control or the sequence of activities in a system, process, or workflow. It is used to model the behavior of a system and illustrate how different activities or actions relate to each other. Activity diagrams consist of nodes, which represent activities or actions, and edges, which represent the flow of control between activities. Nodes can also have additional properties, such as conditions or constraints, that determine when they are executed. Activity diagrams are particularly useful for visualizing complex processes, identifying bottlenecks, and improving the efficiency of workflows.

A.1 External source or receiver

A source or sink is a person or part of organization ,which enters or receives information from the system,but is considered to be outlining the contest of data flow model.

A.2 Transform process

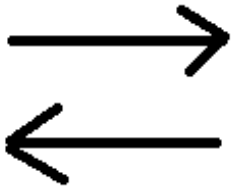


A process represents transformation where incoming data flowsare changed into outgoing data flow

A.3 Input/Output

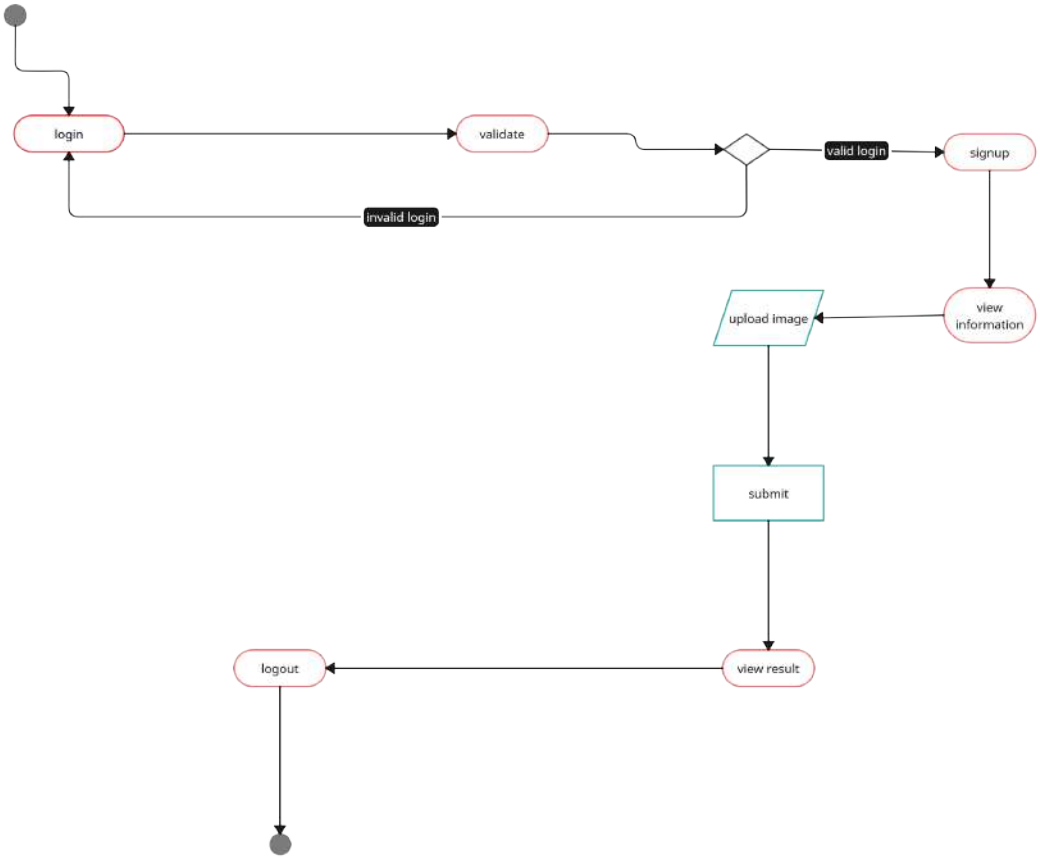
The input/output symbol represents a point in the process where data enters or exits the system. It is used to indicate the input or output of data, information, or materials at a specific point in the workflow. The input/output symbol is typically represented as a rectangle with a thin vertical line on the left side for inputs and a thin vertical line on the right side for outputs.

A.4 Data flow



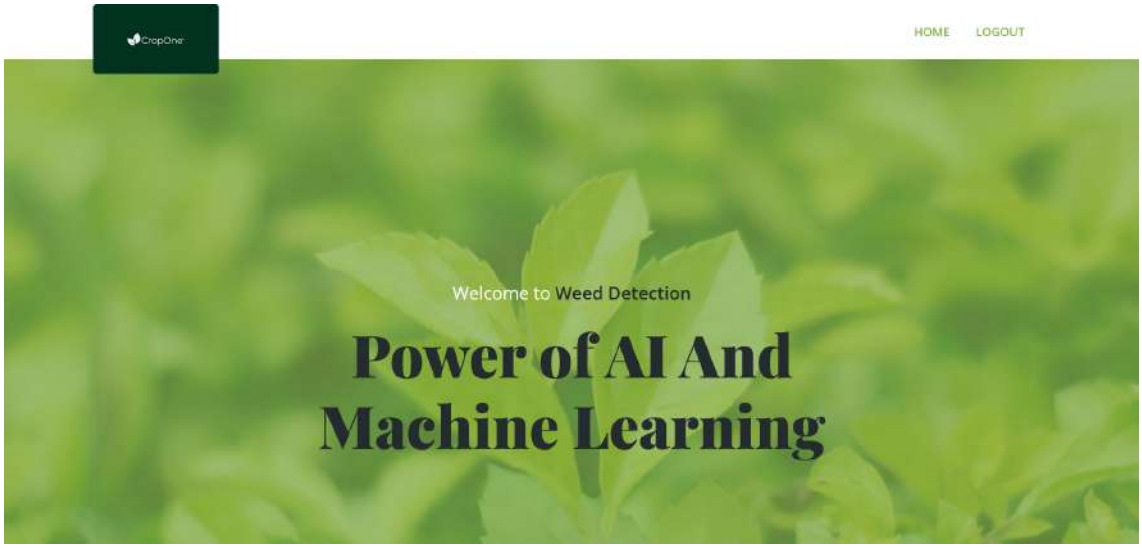
A data flow is a route, which enable packets of data to travel from one point to another.Data may flow,with arrowhead pointing in the direction of the flow

A.5 Diagram

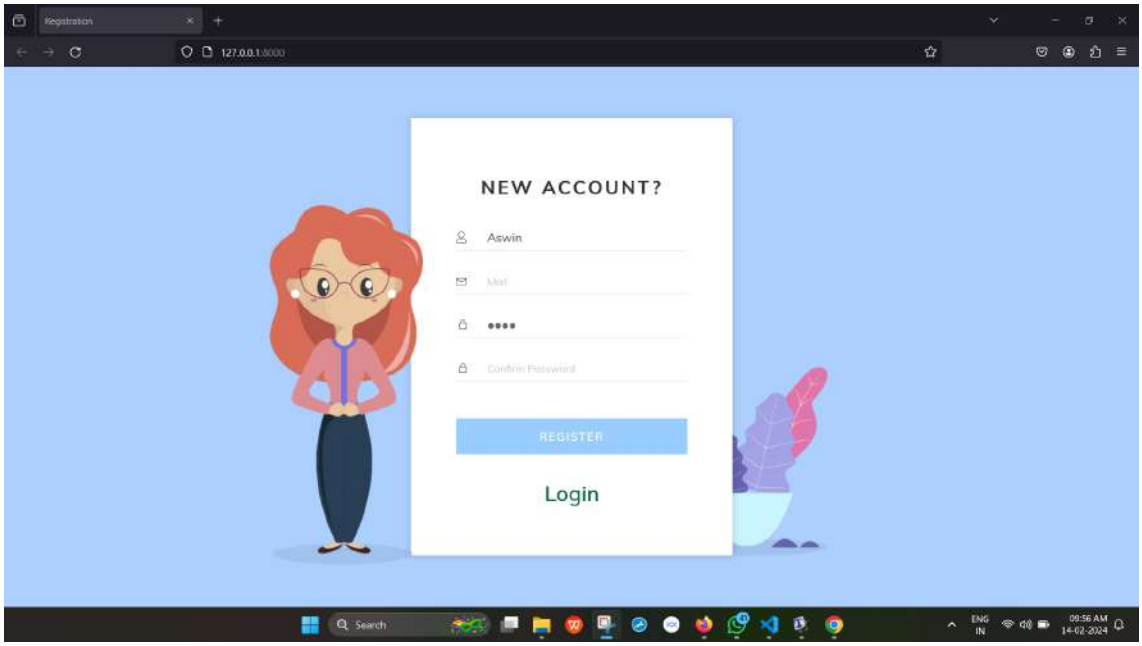

login
validate
valid login
signup
invalid login
view information
upload image
submit
view result
logout

B USER INTERFACES

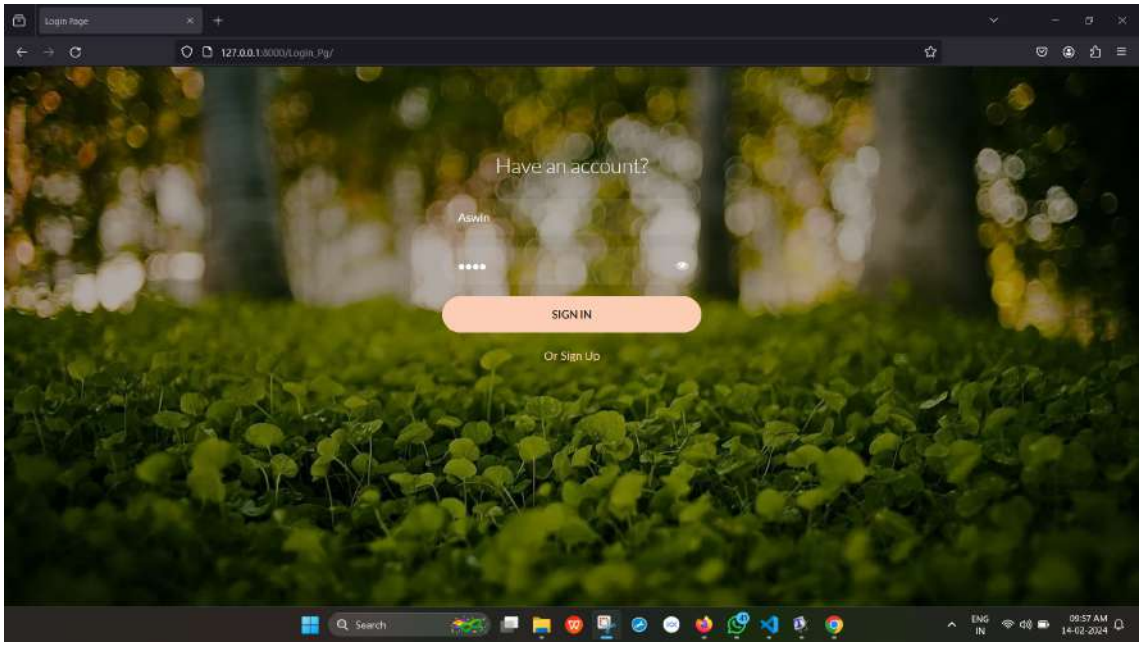
B.1 HOME



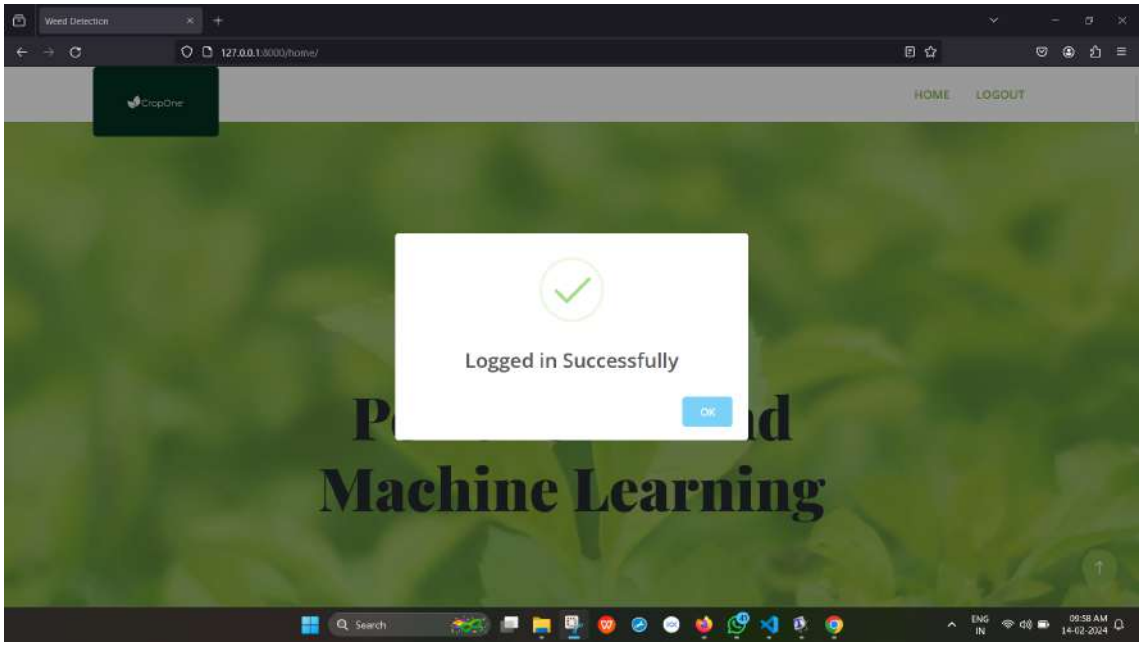
B.2 REGISTER PAGE



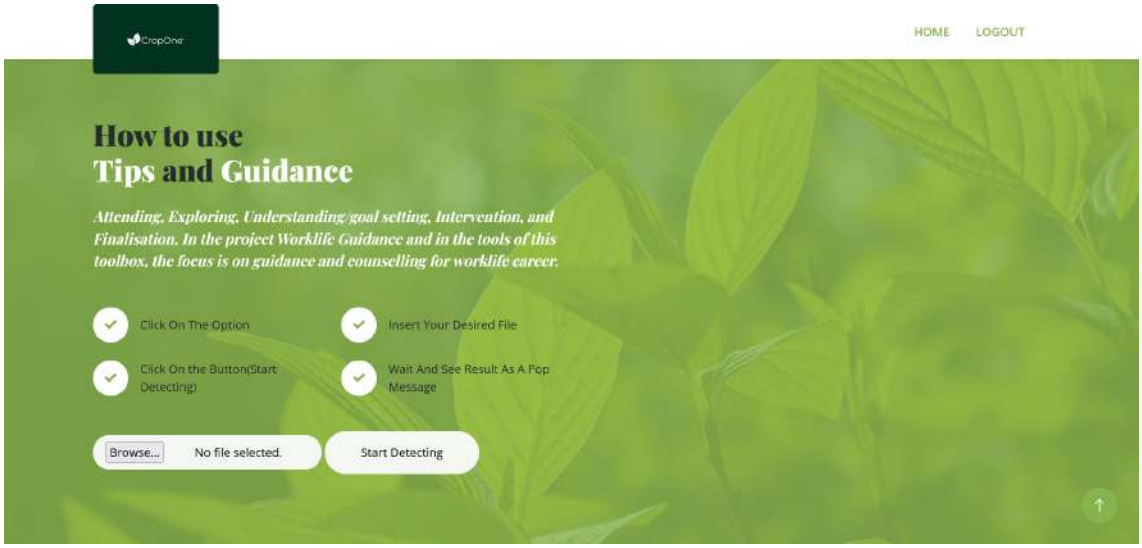
B.3 SIGNUP PAGE



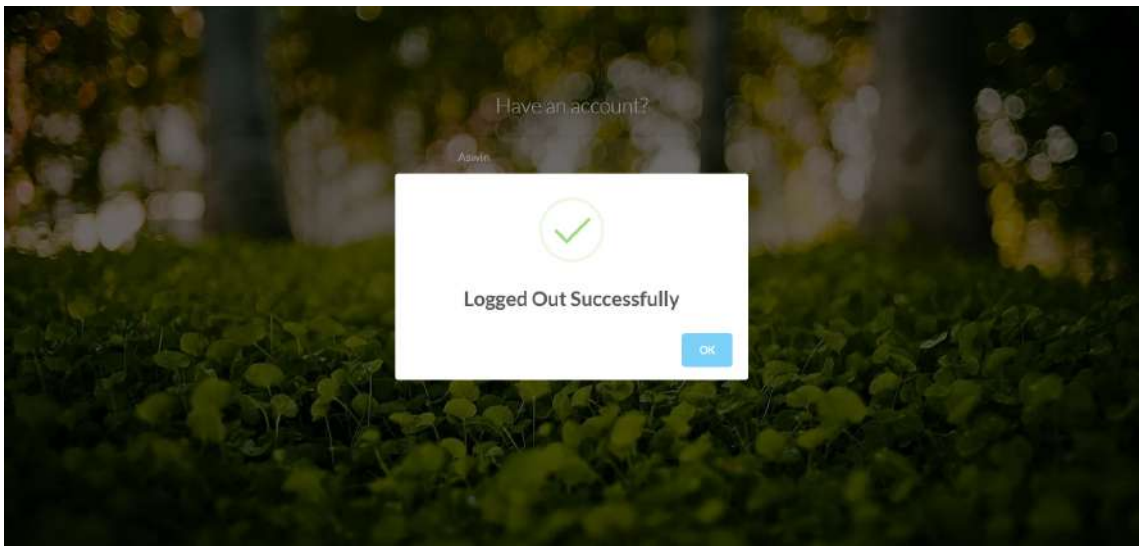
B.4 LOGIN CONFIRMATION



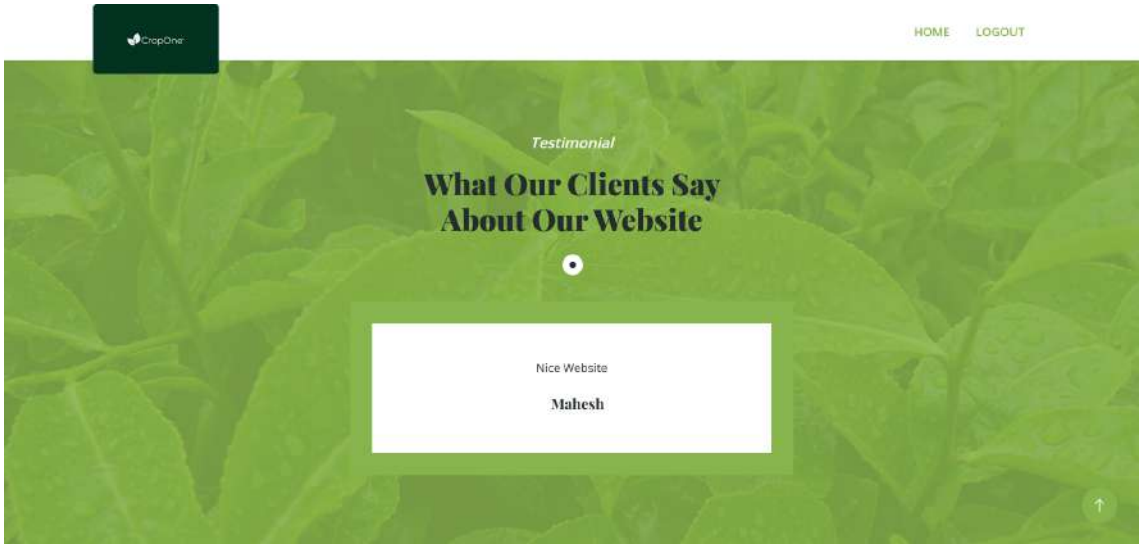
B.5 USER INPUT



B.6 LOGOUT CONFIRMATION



B.7 REVIEW


CropOne
HOME
LOGOUT
Testimonial
What Our Clients Say
About Our Website
Nice Website
Mahesh

C CODE

style.css

```
/********** Template CSS **********/
:root {
    --primary: #88B44E;
    --secondary: #FB9F38;
    --light: #F5F8F2;
    --dark: #252C30;
}

.back-to-top {
    position: fixed;
    display: none;
    right: 30px;
    bottom: 30px;
    z-index: 99;
}

.fw-medium {
    font-weight: 600;
}

.fw-bold {
    font-weight: 700;
}

.fw-black {
    font-weight: 900;
}


/*** Spinner ***/
#spinner {
    opacity: 0;
    visibility: hidden;
    transition: opacity .5s ease-out, visibility 0s linear .5s;
    z-index: 99999;
}

#spinner.show {
    transition: opacity .5s ease-out, visibility 0s linear 0s;
    visibility: visible;
    opacity: 1;
}
```

```
/*** Button ***/
.btn {
    transition: .5s;
    font-weight: 500;
}

.btn-primary,
.btn-outline-primary:hover {
    color: var(--light);
}

.btn-secondary,
.btn-outline-secondary:hover {
    color: var(--dark);
}

.btn-square {
    width: 38px;
    height: 38px;
}

.btn-sm-square {
    width: 32px;
    height: 32px;
}

.btn-lg-square {
    width: 48px;
    height: 48px;
}

.btn-square,
.btn-sm-square,
.btn-lg-square {
    padding: 0;
    display: flex;
    align-items: center;
    justify-content: center;
    font-weight: normal;
}


/*** Navbar ***/
.sticky-top {
```

```
    top: -150px;
    transition: .5s;
}

.navbar .navbar-brand {
    position: absolute;
    padding: 0;
    width: 170px;
    height: 135px;
    top: 0;
    left: 0;
}

.navbar .navbar-nav .nav-link {
    margin-right: 35px;
    padding: 25px 0;
    color: var(--dark);
    font-weight: 600;
    text-transform: uppercase;
    outline: none;
}

.navbar .navbar-nav .nav-link:hover,
.navbar .navbar-nav .nav-link.active {
    color: var(--primary);
}

.navbar .dropdown-toggle::after {
    border: none;
    content: "\f107";
    font-family: "Font Awesome 5 Free";
    font-weight: 900;
    vertical-align: middle;
    margin-left: 8px;
}

@media (max-width: 991.98px) {
    .navbar .navbar-brand {
        width: 126px;
        height: 100px;
    }

    .navbar .navbar-nav .nav-link  {
        margin-right: 0;
        padding: 10px 0;
    }
```

```
    .navbar .navbar-nav {
        margin-top: 75px;
        border-top: 1px solid #EEEEEE;
    }
}

@media (min-width: 992px) {
    .navbar .nav-item .dropdown-menu {
        display: block;
        border: none;
        margin-top: 0;
        top: 150%;
        opacity: 0;
        visibility: hidden;
        transition: .5s;
    }

    .navbar .nav-item:hover .dropdown-menu {
        top: 100%;
        visibility: visible;
        transition: .5s;
        opacity: 1;
    }
}


/*** Header ***/
.carousel-caption {
    top: 0;
    left: 0;
    right: 0;
    bottom: 0;
    display: flex;
    align-items: center;
    background: rgba(136, 180, 78, .7);
    z-index: 1;
}

.carousel-control-prev,
.carousel-control-next {
    width: 15%;
}

.carousel-control-prev-icon,
.carousel-control-next-icon {
```

```
    width: 3.5rem;
    height: 3.5rem;
    border-radius: 3.5rem;
    background-color: var(--dark);
    border: 15px solid var(--dark);
}

@media (max-width: 768px) {
    #header-carousel .carousel-item {
        position: relative;
        min-height: 450px;
    }

    #header-carousel .carousel-item img {
        position: absolute;
        width: 100%;
        height: 100%;
        object-fit: cover;
    }
}

.page-header {
background: linear-gradient(rgba(136, 180, 78, .7), rgba(136,
180, 78, .7)), url(../img/carousel-1.jpg) center center
no-repeat;
    background-size: cover;
}

.page-header .breadcrumb-item+.breadcrumb-item::before {
    color: var(--light);
}

.page-header .breadcrumb-item,
.page-header .breadcrumb-item a {
    font-size: 18px;
    color: var(--light);
}


/*** Section Title ***/
.section-title {
    position: relative;
    margin-bottom: 3rem;
    padding-bottom: 2rem;
}
```

```
.section-title::before {
    position: absolute;
    content: "";
    width: 50%;
    height: 2px;
    bottom: 0;
    left: 0;
    background: var(--primary);
}

.section-title::after {
    position: absolute;
    content: "";
    width: 28px;
    height: 28px;
    bottom: -13px;
    left: calc(25% - 13px);
    background: var(--dark);
    border: 10px solid #FFFFFF;
    border-radius: 28px;
}

.section-title.text-center::before {
    left: 25%;
}

.section-title.text-center::after {
    left: calc(50% - 13px);
}


/*** Products ***/
.product {
background: linear-gradient(rgba(136, 180, 78, .1), rgba(136,
180, 78, .1)),
    url(../img/product-bg.png) left bottom no-repeat;
    background-size: auto;
}

.product-carousel .owl-nav {
    display: flex;
    justify-content: center;
    margin-top: 30px;
}

.product-carousel .owl-nav .owl-prev,
```

```
.product-carousel .owl-nav .owl-next {
    margin: 0 10px;
    width: 55px;
    height: 55px;
    display: flex;
    align-items: center;
    justify-content: center;
    color: #FFFFFF;
    background: var(--primary);
    border-radius: 55px;
    box-shadow: 0 0 45px rgba(0, 0, 0, .15);
    font-size: 25px;
    transition: .5s;
}

.product-carousel .owl-nav .owl-prev:hover,
.product-carousel .owl-nav .owl-next:hover {
    background: #FFFFFF;
    color: var(--primary);
}


/*** About ***/
.video {
background: linear-gradient(rgba(136, 180, 78, .85), rgba(136,
180, 78, .85)),
    url(../img/video-bg.jpg) center center no-repeat;
    background-size: cover;
}

.btn-play {
    position: relative;
    display: block;
    box-sizing: content-box;
    width: 65px;
    height: 75px;
    border-radius: 100%;
    border: none;
    outline: none !important;
    padding: 28px 30px 30px 38px;
    background: #FFFFFF;
}

.btn-play:before {
    content: "";
    position: absolute;
```

```
    z-index: 0;
    left: 50%;
    top: 50%;
    transform: translateX(-50%) translateY(-50%);
    display: block;
    width: 120px;
    height: 120px;
    background: #FFFFFF;
    border-radius: 100%;
    animation: pulse-border 1500ms ease-out infinite;
}

.btn-play:after {
    content: "";
    position: absolute;
    z-index: 1;
    left: 50%;
    top: 50%;
    transform: translateX(-50%) translateY(-50%);
    display: block;
    width: 120px;
    height: 120px;
    background: #FFFFFF;
    border-radius: 100%;
    transition: all 200ms;
}

.btn-play span {
    display: block;
    position: relative;
    z-index: 3;
    width: 0;
    height: 0;
    left: 13px;
    border-left: 40px solid var(--primary);
    border-top: 28px solid transparent;
    border-bottom: 28px solid transparent;
}

@keyframes pulse-border {
    0% {
transform: translateX(-50%) translateY(-50%) translateZ(0)
scale(1);
        opacity: 1;
    }
```

```
    100% {
transform: translateX(-50%) translateY(-50%) translateZ(0)
scale(2);
        opacity: 0;
    }
}

.modal-video .modal-dialog {
    position: relative;
    max-width: 800px;
    margin: 60px auto 0 auto;
}

.modal-video .modal-body {
    position: relative;
    padding: 0px;
}

.modal-video .close {
    position: absolute;
    width: 30px;
    height: 30px;
    right: 0px;
    top: -30px;
    z-index: 999;
    font-size: 30px;
    font-weight: normal;
    color: #FFFFFF;
    background: #000000;
    opacity: 1;
}


/*** Store ***/
.store-item .store-overlay {
    position: absolute;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: center;
    background: rgba(138, 180, 78, .3);
    opacity: 0;
```

```
    transition: .5s;
}

.store-item:hover .store-overlay {
    opacity: 1;
}


/*** Contact ***/
.contact .btn-square {
    width: 100px;
    height: 100px;
    border: 20px solid var(--light);
    background: var(--primary);
    border-radius: 50px;
}


/*** Testimonial ***/
.testimonial {
background: linear-gradient(rgba(136, 180, 78, .85), rgba(136,
180, 78, .85)),
    url(../img/testimonial-bg.jpg) center center no-repeat;
    background-size: cover;
}

.testimonial-item {
    margin: 0 auto;
    max-width: 600px;
    text-align: center;
    background: #FFFFFF;
    border: 30px solid var(--primary);
}

.testimonial-item img {
    width: 60px !important;
    height: 60px !important;
    border-radius: 60px;
}

.testimonial-carousel .owl-dots {
    margin-top: 35px;
    display: flex;
    align-items: flex-end;
    justify-content: center;
}
```

```
.testimonial-carousel .owl-dot {
    position: relative;
    display: inline-block;
    margin: 0 5px;
    width: 15px;
    height: 15px;
    background: var(--primary);
    border-radius: 15px;
    transition: .5s;
}

.testimonial-carousel .owl-dot.active {
    width: 30px;
    background: var(--dark);
}


/*** Footer ***/
.footer {
    color: #B0B9AE;
}

.footer .btn.btn-link {
    display: block;
    margin-bottom: 5px;
    padding: 0;
    text-align: left;
    color: #B0B9AE;
    font-weight: normal;
    text-transform: capitalize;
    transition: .3s;
}

.footer .btn.btn-link::before {
    position: relative;
    content: "\f105";
    font-family: "Font Awesome 5 Free";
    font-weight: 900;
    color: var(--primary);
    margin-right: 10px;
}

.footer .btn.btn-link:hover {
    color: var(--light);
    letter-spacing: 1px;
```

```
    box-shadow: none;
}

.copyright {
    color: #B0B9AE;
}

.copyright {
    background: #252525;
}

.copyright a:hover {
    color: #FFFFFF !important;
}
```

home.html

```
<!DOCTYPE html>
{% load static %}
<html lang="en">

<head>
<meta charset="utf-8">
<title>Weed Detection</title>
<meta content="width=device-width, initial-scale=1.0"
name="viewport">
<meta content="" name="keywords">
<meta content="" name="description">

<!-- Favicon -->
<link href="{% static 'img/favicon.ico' %}" rel="icon">

<!-- Google Web Fonts -->
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com"
crossorigin>
<link href="https://fonts.googleapis.com/css2?
family=Open+Sans:wght@400;600&family=Playfair+Display:wght@700;900&display=swap"
rel="stylesheet">

<!-- Icon Font Stylesheet -->
<link
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.10.0/css/all.min.css"
rel="stylesheet">
<link
```

```
href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.4.1/font/bootstrap-icons.css"
rel="stylesheet">

<!-- Libraries Stylesheet -->
<link href="{% static 'lib/animate/animate.min.css' %}"
rel="stylesheet">
<link href="{% static
'lib/owlcarousel/assets/owl.carousel.min.css' %}"
rel="stylesheet">

<!-- Customized Bootstrap Stylesheet -->
<link href="{% static 'css/bootstrap.min.css' %}"
rel="stylesheet">

<!-- Template Stylesheet -->
<link href="{% static 'css/style.css' %}" rel="stylesheet">
</head>

<body>
<!-- Spinner Start -->
<div id="spinner" class="show bg-white position-fixed
translate-middle w-100 vh-100 top-50
start-50 d-flex align-items-center justify-content-center">
<div class="spinner-border text-primary" role="status"
style="width: 3rem; height: 3rem;"></div>
</div>
<!-- Spinner End -->


<!-- Navbar Start -->
<div class="container-fluid bg-white sticky-top">
<div class="container">
<nav class="navbar navbar-expand-lg bg-white navbar-light py-2
py-lg-0">
<a href="index.html" class="navbar-brand">
<img class="img-fluid" src="{% static 'img/crop1.jpg' %}"
alt="Logo" width="200px" height="60px">
</a>
<button type="button" class="navbar-toggler ms-auto me-0"
data-bs-toggle="collapse" data-bs-target="#navbarCollapse">
<span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="navbarCollapse">
<div class="navbar-nav ms-auto">
<a href="{% url 'home' %}" class="nav-item nav-link
active">Home</a>
```

```
{% if request.session.username %}
<a href="{% url 'Logout_fn' %}" class="nav-item nav-link
active">Logout</a>
{% else %}
<a href="{% url 'RegistrationForm' %}" class="nav-item nav-link
active">Register</a>
{% endif %}

</div>
</div>



</nav>
</div>
</div>
<!-- Navbar End -->


<!-- Carousel Start -->
<div class="container-fluid px-0 mb-5">
<div id="header-carousel" class="carousel slide carousel-fade"
data-bs-ride="carousel">
<div class="carousel-inner">
<div class="carousel-item active">
<img class="w-100" src="{% static 'img/carousel-1.jpg' %}"
alt="Image">
<div class="carousel-caption">
<div class="container">
<div class="row justify-content-center">
<div class="col-lg-7 text-center">
<p class="fs-4 text-white animated zoomIn">Welcome to
<strong class="text-dark">Weed Detection</strong></p>
<h1 class="display-1 text-dark mb-4 animated zoomIn">
Power of AI And Machine Learning</h1>
</div>
</div>
</div>
</div>
</div>

</div>
</div>
</div>
<!-- Carousel End -->
```

```
<!-- About Start -->
<div class="container-xxl py-5">
<div class="container">
<div class="row g-5">
<div class="col-lg-6">
<div class="row g-3">
<div class="col-6 text-end">
<img class="img-fluid bg-white w-200 mb-3 wow fadeIn"
data-wow-delay="0.1s" src="{% static 'img/cropandweed.jpeg' %}"
alt="">
<img class="img-fluid bg-white w-300 wow fadeIn"
data-wow-delay="0.2s" src="{% static 'img/cropandweed2.jpeg' %}"
alt="">
</div>
<div class="col-6">
<img class="img-fluid bg-white w-200 mb-3 wow fadeIn"
data-wow-delay="0.3s" src="{% static 'img/cropandweed3.jpeg' %}"
alt="">
<img class="img-fluid bg-white w-300 wow fadeIn"
data-wow-delay="0.4s" src="{% static 'img/cropandweed4.jpeg' %}"
alt="">
</div>
</div>
</div>
<div class="col-lg-6 wow fadeIn" data-wow-delay="0.5s">
<div class="section-title">
<p class="fs-5 fw-medium fst-italic text-primary">About
Project</p>
<h1 class="display-6">What is weed detection?</h1>
</div>
<div class="row g-3 mb-4">
<div class="col-sm-4">
<img class="img-fluid bg-white w-150" src="{% static
'img/cropandweed5.jpeg' %}" alt="">
</div>
<div class="col-sm-8">
<h5>Weed detection systems are important solutions to one of the
existing
agricultural problems|unmechanized weed control. </h5>
<p class="mb-0"> Weed detection also helps provide a means of
reducing or eliminating herbicide use, mitigating agricultural
environmental
and health impact, and improving sustainability.</p>
</div>
</div>
```

```
<div class="border-top mb-4"></div>
<div class="row g-3">
<div class="col-sm-8">
<h5>What is the objective for weed detection project?</h5>
<p class="mb-0">The main objective has been to obtain a formula
so that a
weed detection system can be developed through binary
classifications.
The initial step of image processing is the detection of green
plants in order to
eliminate all the soil in the image, reducing information that
is not necessary.</p>
</div>
<div class="col-sm-4">
<img class="img-fluid bg-white w-150" src="{% static
'img/cropandweed5.jpeg' %}" alt="">
</div>
</div>
</div>
</div>
</div>
</div>
<!-- About End -->


<!-- Products Start -->
<div class="container-fluid product py-5 my-5">
<div class="container py-5">
<div class="section-title text-center mx-auto wow fadeInUp"
data-wow-delay="0.1s"
 style="max-width: 500px;">
<p class="fs-5 fw-medium fst-italic text-primary">Our Source</p><h1 class="display-6">
What is the role of AI in
agriculture?</h1>
</div>
<div class="owl-carousel product-carousel wow fadeInUp"
data-wow-delay="0.5s">
<a href="" class="d-block product-item rounded">
<img src="img/product-1.jpg" alt="">
<div class="bg-white shadow-sm text-center p-4 position-relative
mt-n5 mx-4">
<h4 class="text-primary">AI Specification</h4>
<span class="text-body">By analyzing data from various sources,
AI can help farmers make data-driven decisions,
optimize resource usage, and reduce environmental impact.AI can
be used to develop
```

```
 image recognition systems that can.
identify weeds with a high degree of accuracy</span>
</div>
</a>
<a href="" class="d-block product-item rounded">
<img src="img/product-2.jpg" alt="">
<div class="bg-white shadow-sm text-center p-4 position-relative
mt-n5 mx-4">
<h4 class="text-primary">General Science</h4>
<span class="text-body">For example, the World Economic Forum
has reported that
 AI integration in agriculture could bring about
a 60% decrease in pesticide usage and a 50% reduction in water
usage.
AI algorithms can analyze the chemical composition of soil
samples to determine which nutrients may be lacking.
AI can also identify or even predict crop diseases. </span>
</div>
</a>
<a href="" class="d-block product-item rounded">
<img src="img/product-3.jpg" alt="">
<div class="bg-white shadow-sm text-center p-4 position-relative
mt-n5 mx-4">
<h4 class="text-primary">Weed</h4>
<span class="text-body">Weeds are unwanted and undesirable
 plants which interfere with the. utilization of land and water
 resources and thus adversely affect human welfare.
They can also be referred as plants out of place. Weeds compete
with the
beneficial and desired vegetation in crop lands, forests,
aquatic systems etc.</span>
</div>
</a>
</div>
</div>
</div>
<!-- Products End -->


<!-- Article Start -->
<div class="container-xxl py-5">
<div class="container">
<div class="row g-5">
<div class="col-lg-5 wow fadeIn" data-wow-delay="0.1s">
<img class="img-fluid" src="{% static 'img/article.jpg' %}"
alt="">
```

```
</div>
<div class="col-lg-6 wow fadeIn" data-wow-delay="0.5s">
<div class="section-title">
<p class="fs-5 fw-medium fst-italic text-primary">Featured
Acticle</p>
<h1 class="display-6">The history of Weed Detection</h1>
</div>
<p class="mb-4">In olden days weed detection was done
by employing some men, especially for weed removal purpose.
They will detect the weeds by checking each and every plant
field
. Then they will pluck them out manually using their
hands or spades. In proposed system the weeds can be detected
by image processing techniques. The main aim of the
project is to identify weed affected area for more seeding.
Since there are different issues with respect to the
developed yields on the field, a standout amongst the majority
is about
weeds which go as a hindrance in the development of the
harvests. Weeds may break down the life and nature of the yields
if
it is not controlled legitimately in time.
This proposed thought centers around to reduce the work and also
the time
needed to identify weeds and expel the same.
During this process they will consider Edge detection and
colour segmentation process.
They will analyze each and every crop with their intensity
colour,
Intensity, edge, Size etc., are been obtained as output.
The colour of the edges and veins of the crop and weed are
white after segmentation process and edge detection,
whereas the rest of the image is totally black in colour.
After the process of Edge detection and Colour
segmentation it has undergone the process of Filtering.
The Filtering can be a type of process in which each and every
crop can be identified and their gain value,
tradeoff's, edge, frequency of crop and weed can be identified
and their threshold values can obtained </p>

</div>
</div>
</div>
</div>
<!-- Article End -->
```

```
<!-- Video Start -->
<div class="container-fluid video my-5">
<div class="container">
<div class="row g-0">
<div class="col-lg-6 py-5 wow fadeIn" data-wow-delay="0.1s">
<div class="py-5">
<h1 class="display-6 mb-4">How to use <br><span
class="text-white">
Tips</span> and <span class="text-white">Guidance</span></h1>
<h5 class="fw-normal lh-base fst-italic text-white
mb-5">Attending, Exploring
, Understanding/goal setting, Intervention, and Finalisation.
In the project Worklife Guidance and in the tools of this
toolbox,
the focus is on guidance and counselling for worklife
career.</h5>
<div class="row g-4 mb-5">
<div class="col-sm-6">
<div class="d-flex align-items-center">
<div class="flex-shrink-0 btn-lg-square bg-white text-primary
rounded-circle me-3">
<i class="fa fa-check"></i>
</div>
<span class="text-dark">Click On The Option</span>
</div>
</div>
<div class="col-sm-6">
<div class="d-flex align-items-center">
<div class="flex-shrink-0 btn-lg-square bg-white text-primary
rounded-circle me-3">
<i class="fa fa-check"></i>
</div>
<span class="text-dark">Insert Your Desired File</span>
</div>
</div>
<div class="col-sm-6">
<div class="d-flex align-items-center">
<div class="flex-shrink-0 btn-lg-square bg-white text-primary
rounded-circle me-3">
<i class="fa fa-check"></i>
</div>
<span class="text-dark">Click On the Button(Start
Detecting)</span>
</div>
```

```
</div>
<div class="col-sm-6">
<div class="d-flex align-items-center">
<div class="flex-shrink-0 btn-lg-square bg-white text-primary
rounded-circle me-3">
<i class="fa fa-check"></i>
</div>
<span class="text-dark">Wait And See Result As A Pop
Message</span>
</div>
</div>
</div>
<form action="{% url 'detect_weed_or_crop' %}" method="post"
enctype="multipart/form-data">
{% csrf_token %}
<input type="file" class="btn btn-light rounded-pill py-2 px-3"
required name="image_file">
{% if request.session.username %}
<button class="btn btn-light rounded-pill py-3 px-5">Start
Detecting</button>
</form>
{% else %}


<a href="{% url 'RegistrationForm' %}" class="btn btn-light
rounded-pill py-3 px-5">Register</a>



{% endif %}


<div class="modal fade" id="exampleModal" tabindex="-1"
aria-labelledby="exampleModalLabel" aria-hidden="true">
<div class="modal-dialog">
<div class="modal-content">
<div class="modal-header">
<h1 class="modal-title fs-5" id="exampleModalLabel">Result</h1>
<button type="button" class="btn-close" data-bs-dismiss="modal"
aria-label="Close"></button>
</div>
<div class="modal-body">
{{result.result}}
</div>
<div class="modal-footer">
<button type="button" class="btn btn-secondary"
```

```
data-bs-dismiss="modal">Close</button>

</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>


<!-- Testimonial Start -->
<div class="container-fluid testimonial py-5 my-5">

<div class="container py-5">
<div class="section-title text-center mx-auto wow fadeInUp"
 data-wow-delay="0.1s" style="max-width: 500px;">
<p class="fs-5 fw-medium fst-italic text-white">Testimonial</p>
<h1 class="display-6">What Our Clients Say About Our
Website</h1>
</div>

<div class="owl-carousel testimonial-carousel wow fadeInUp"
data-wow-delay="0.5s">
{% for i in x %}
<div class="testimonial-item p-4 p-lg-5">
<p class="mb-4">{{i.Description}}</p>
<div class="d-flex align-items-center justify-content-center">
<div class="text-start ms-3">
<h5>{{i.username}}</h5>
</div>
</div>
</div>
{% endfor %}
</div>

</div>

</div>
<!-- Testimonial End -->


<!-- Contact Start -->
<div class="container-xxl contact py-5">
```

```
<div class="container">
<div class="section-title text-center mx-auto wow fadeInUp"
 data-wow-delay="0.1s" style="max-width: 500px;">
<h1 class="display-6">Overview</h1>
</div>
<div class="row justify-content-center wow fadeInUp"
data-wow-delay="0.1s">
<div class="col-lg-8">
<p class="text-center mb-5">Artificial intelligence,
specifically deep learning, is a fast-growing research field
today. One of its
various applications is object recognition,
making use of computer vision. The combination of these
two technologies leads to the purpose of this thesis.
 In this project, a system for the identification of
different crops and weeds has been developed as an
 alternative to the system present on the FarmBot
company's robots. This is done by accessing the
 images through the FarmBot API, using computer
vision for image processing, and artificial intelligence
for the application of transfer learning to a RCNN
that performs the plants identification autonomously.
 The results obtained show that the system
works with an accuracy of 78.10
% for the main crop and 53.12% and 44.76% for the two weeds
considered. Moreover, the coordinates of the weeds
 are also given as results. The performance of the
resulting system is compared both with
 similar projects found during research, and with the current
version of the FarmBot weed detector. </p>
<div class="row g-5">
<div class="col-md-4 text-center wow fadeInUp"
data-wow-delay="0.3s">
<div class="btn-square mx-auto mb-3">
<i class="fa fa-envelope fa-2x text-white"></i>
</div>
<p class="mb-2">name@example.com</p>
<p class="mb-0">name1@example.com</p>
</div>
<div class="col-md-4 text-center wow fadeInUp"
data-wow-delay="0.4s">
<div class="btn-square mx-auto mb-3">
<i class="fa fa-phone fa-2x text-white"></i>
</div>
<p class="mb-2">+012 345 67890</p>
</div>
```

```
<div class="col-md-4 text-center wow fadeInUp"
data-wow-delay="0.5s">
<div class="btn-square mx-auto mb-3">
<i class="fa fa-map-marker-alt fa-2x text-white"></i>
</div>
<p class="mb-2">Irinjalakuda</p>
<p class="mb-0">Thrissur, Kerala</p>
</div>
</div>
</div>
</div>
</div>
</div>
<!-- Contact Start -->

<form action="{% url 'ReviewSave' %}" method="post">
{% csrf_token %}

<div style="margin-left:450px;">
<div class="form-floating">
<input type="hidden" name="uname"
value="{{request.session.username}}">
<input class="form-control" placeholder="Insert Your Review
Here"
id="message" type="text" name="txt" style="height:
120px;width:550px;">
<label for="message">Review</label>
</div>
<br>
<div class="col-12">
{% if request.session.username %}
<button class="btn btn-primary rounded-pill py-3 px-4"
style="margin-left:200px;" type="submit">
Submit Review</button>

{% else %}
<a href="{% url 'RegistrationForm' %}" class="btn btn-primary
rounded-pill py-3 px-4"
style="margin-left:200px;">Register</a>


{% endif %}


</div>
</div>
```

```
</form>


<!-- Footer Start -->
<div class="container-fluid bg-dark footer mt-5 py-5 wow fadeIn"
data-wow-delay="0.1s">
<div class="container py-5">
<div class="row g-5">
<div class="col-lg-3 col-md-6">
<h4 class="text-primary mb-4">Authencity</h4>
<p class="mb-2"><i class="fa fa-map-marker-alt text-primary
me-3">
</i>Fort Street, Ernakulam , Kerala</p>
<p class="mb-2"><i class="fa fa-phone-alt text-primary me-3
"></i>+012 345 67890</p>
<p class="mb-2"><i class="fa fa-envelope text-primary me-3">
</i>name@example.com</p>
<div class="d-flex pt-3">
<a class="btn btn-square btn-primary rounded-circle me-2"
href="#">
<i class="fab fa-twitter"></i></a>
<a class="btn btn-square btn-primary rounded-circle me-2"
href="#">
<i class="fab fa-facebook-f"></i></a>
<a class="btn btn-square btn-primary rounded-circle me-2"
href="#">
<i class="fab fa-youtube"></i></a>
<a class="btn btn-square btn-primary rounded-circle me-2"
href="#">
<i class="fab fa-linkedin-in"></i></a>
</div>
</div>
<div class="col-lg-3 col-md-6">
<h4 class="text-primary mb-4">Quick Links</h4>
<a class="btn btn-link" href="{% url 'home' %}">Home</a>
{% if request.session.username%}
<a class="btn btn-link" href="{% url 'Logout_fn' %}">Logout</a>
{% else %}
<a class="btn btn-link" href="{% url 'RegistrationForm' %}">Sign
Up</a>
{% endif %}
</div>
<div class="col-lg-3 col-md-6">
<h4 class="text-primary mb-4">Business Hours</h4>
<p class="mb-1">Monday - Saturday</p>
<h6 class="text-light">24/5</h6>
```

```
<p class="mb-1">Saturday</p>
<h6 class="text-light">09:00 am - 12:00 pm</h6>
<p class="mb-1">Sunday</p>
<h6 class="text-light">Closed</h6>
</div>
</div>
</div>
</div>
<!-- Footer End -->


<!-- Copyright Start -->
<div class="container-fluid copyright py-4">
<div class="container">
<div class="row">
<div class="col-md-6 text-center text-md-start mb-3 mb-md-0">
&copy; <a class="fw-medium" href="#">Weed Detection</a>, All
Right Reserved.
</div>
<div class="col-md-6 text-center text-md-end">
<!--/*** This template is free as long as you keep the footer
author's credit
link/attribution link/backlink. If you'd like to use the
template without the
footer author's credit link/attribution link/backlink, you can
purchase the
Credit Removal License from
"https://htmlcodex.com/credit-removal".
Thank you for your support. ***/-->
</div>
</div>
</div>
</div>
<!-- Copyright End -->


<!-- Back to Top -->
<a href="#" class="btn btn-lg btn-primary btn-lg-square
rounded-circle back-to-top">
<i class="bi bi-arrow-up"></i></a>


<!-- JavaScript Libraries -->
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.1/jquery.min.js"></script>
<script
```

```
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0/dist/js/bootstrap.bundle.min.js">
</script>
<script src="{% static 'lib/wow/wow.min.js' %}"></script>
<script src="{% static 'lib/easing/easing.min.js' %}"></script>
<script src="{% static 'lib/waypoints/waypoints.min.js'
%}"></script>
<script src="{% static 'lib/owlcarousel/owl.carousel.min.js'
%}"></script>

<!-- Template Javascript -->
<script src="{% static 'js/main.js' %}"></script>
<script
src="https://unpkg.com/sweetalert/dist/sweetalert.min.js"></script>
{% if messages %}
{% for i in messages %}
{% if i.tags == 'warning' %}
<script>
swal('{{i}}','', 'warning');
</script>
{% elif i.tags == 'error' %}
<script>
swal('{{i}}','', 'error');
</script>
{% else %}
<script>
swal('{{i}}','', 'success');
</script>
{% endif %}

{% endfor %}
{% endif %}
</body>


</html>
```

login.html

```
<!DOCTYPE html>
{% load static %}
<html>
<head>
<meta charset="utf-8">
<title>Registration</title>
<meta name="viewport" content="width=device-width,
```

```
initial-scale=1.0">

<!-- LINEARICONS -->
<link rel="stylesheet" href="{% static
'assets/fonts/linearicons/style.css' %}">

<!-- STYLE CSS -->
<link rel="stylesheet" href="{% static 'assets/css/style.css'
%}">
</head>

<body>

<div class="wrapper">
<div class="inner">
<img src="{% static 'assets/images/image-1.png' %}" alt=""
class="image-1">
<form action="{% url 'Registration_save' %}" method="post">
{% csrf_token %}
<h3>New Account?</h3>
<div class="form-holder">
<span class="lnr lnr-user"></span>
<input type="text" class="form-control" placeholder="Username"
name="uname" required>
</div>

<div class="form-holder">
<span class="lnr lnr-envelope"></span>
<input type="email" class="form-control" placeholder="Mail"
name="email" required>
</div>
<div class="form-holder">
<span class="lnr lnr-lock"></span>
<input type="password" class="form-control"
placeholder="Password" name="password" required>
</div>
<div class="form-holder">
<span class="lnr lnr-lock"></span>
<input type="password" class="form-control" placeholder="Confirm
Password" name="cpassword" required>
</div>
<button type="submit">
<span>Register</span>
</button>
<br><br>
```

```
<h1 style="text-align:center;"><a style="color: rgb(23, 107,
79);" href="{% url 'Login_Pg' %}">Login</a></h1>
</form>

<img src="{% static 'assets/images/image-2.png' %}" alt=""
class="image-2">
</div>

</div>

<script src="{% static 'assets/js/jquery-3.3.1.min.js'
%}"></script>
<script src="{% static 'assets/js/main.js' %}"></script>
<script
src="https://unpkg.com/sweetalert/dist/sweetalert.min.js"></script>
{% if messages %}
{% for i in messages %}
{% if i.tags == 'warning' %}
<script>
swal('{{i}}','', 'warning');
</script>
{% elif i.tags == 'error' %}
<script>
swal('{{i}}','', 'error');
</script>
{% else %}
<script>
swal('{{i}}','', 'success');
</script>
{% endif %}

{% endfor %}
{% endif %}
</body><!-- This templates was made by Colorlib
(https://colorlib.com) -->
</html>

\\
```

Registrartion.html

```
<!DOCTYPE html>
{% load static %}
<html>
<head>
<meta charset="utf-8">
```

```
<title>Registration</title>
<meta name="viewport" content="width=device-width,
initial-scale=1.0">

<!-- LINEARICONS -->
<link rel="stylesheet" href="{% static
'assets/fonts/linearicons/style.css' %}">

<!-- STYLE CSS -->
<link rel="stylesheet" href="{% static 'assets/css/style.css'
%}">
</head>

<body>

<div class="wrapper">
<div class="inner">
<img src="{% static 'assets/images/image-1.png' %}" alt=""
class="image-1">
<form action="{% url 'Registration_save' %}" method="post">
{% csrf_token %}
<h3>New Account?</h3>
<div class="form-holder">
<span class="lnr lnr-user"></span>
<input type="text" class="form-control" placeholder="Username"
name="uname" required>
</div>

<div class="form-holder">
<span class="lnr lnr-envelope"></span>
<input type="email" class="form-control" placeholder="Mail"
name="email" required>
</div>
<div class="form-holder">
<span class="lnr lnr-lock"></span>
<input type="password" class="form-control"
placeholder="Password" name="password" required>
</div> <div class="form-holder">
<span class="lnr lnr-lock"></span>
<input type="password" class="form-control" placeholder="Confirm
Password" name="cpassword" required>
</div>
<button type="submit">
<span>Register</span>
</button>
<br><br>
```

```
<h1 style="text-align:center;"><a style="color: rgb(23, 107,
79);" href="{% url 'Login_Pg' %}">Login</a></h1>
</form>

<img src="{% static 'assets/images/image-2.png' %}" alt=""
class="image-2">
</div>

</div>

<script src="{% static 'assets/js/jquery-3.3.1.min.js'
%}"></script>
<script src="{% static 'assets/js/main.js' %}"></script>
<script
src="https://unpkg.com/sweetalert/dist/sweetalert.min.js"></script>
{% if messages %}
{% for i in messages %}
{% if i.tags == 'warning' %}
<script>
swal('{{i}}','', 'warning');
</script>
{% elif i.tags == 'error' %}
<script>
swal('{{i}}','', 'error');
</script>
{% else %}
<script>
swal('{{i}}','', 'success');
</script>
{% endif %}

{% endfor %}
{% endif %}
</body>
</html>
```

apps.py

```
from django.apps import AppConfig


class FrontendConfig(AppConfig):
    default_auto_field = 'django.db.models.BigAutoField'
    name = 'Frontend'
```

models.py

```
from django.db import models

# Create your models here.
class Review(models.Model):
username=models.CharField(max_length=500,null=True,blank=True)
Description=models.CharField(max_length=5000,null=True,blank=True)

class Registration(models.Model):
username=models.CharField(max_length=100,null=True,blank=True
)Email=models.EmailField(max_length=100,null=True,blank=True)
Password=models.CharField(max_length=8,null=True,blank=True)
Confirm_Password=models.CharField(max_length=8,null=True,blank=True)
```

urls.py

```
from django.urls import path
from Frontend import views

urlpatterns=[
    path('home/',views.home,name="home"),
path('detect_weed_or_crop/',views.detect_weed_or_crop,name="detect_weed_or_crop"),

    path('ReviewSave/',views.ReviewSave,name="ReviewSave"),
    path('',views.RegistrationForm,name="RegistrationForm"),

path('Registration_save/',views.Registration_save,name="Registration_save"),
    path('Login_Pg/',views.Login_Pg,name="Login_Pg"),

    path('Login_fun/',views.Login_fun,name="Login_fun"),
    path('Logout_fn/',views.Logout_fn,name="Logout_fn"),
]
```

views.py

```
from django.shortcuts import render,redirect
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import numpy as np
from keras.applications.imagenet_utils import preprocess_input
from django.contrib import messages
from PIL import Image
from Frontend.models import Review,Registration
```

```
from django.http import HttpResponse,JsonResponse
from tempfile import NamedTemporaryFile
import io
from django.shortcuts import redirect,render
from keras.applications.imagenet_utils import preprocess_input
from django.contrib import messages
from Frontend.models import Review,Registration
import tensorflow as tf
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.mobilenet_v2 import
preprocess_input
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.image as mpimg




def home(request):
    x=Review.objects.all()
    return render(request,'Home.html',{'x':x})


def detect_weed_or_crop(request):
    if request.method == "POST":
        try:
            model_path = 'weed_final.h5'
            model = tf.keras.models.load_model(model_path)

            uploaded_file = request.FILES['image_file']
            with NamedTemporaryFile(delete=False) as temp_file:
                temp_file.write(uploaded_file.read())
                temp_file_path = temp_file.name
img = image.load_img(temp_file_path, target_size=(224, 224))
  img_array = image.img_to_array(img)
            img_array = np.expand_dims(img_array, axis=0)
            img_data = preprocess_input(img_array)

            predictions = model.predict(img_data)

            class_labels = ["crop", "weed", "objects"]
            class_probabilities = predictions[0]

class_percentages = [prob * 100 for prob in class_probabilities]
            threshold = 0.5
```

```
detected_classes = [class_labels[i] for i, prob in
enumerate(class_probabilities) if prob > threshold]

            img = mpimg.imread(temp_file_path)
            plt.imshow(img)
            plt.axis('off')

if "crop" in detected_classes and "weed" in detected_classes:
  crop_index = class_labels.index("crop")
                weed_index = class_labels.index("weed")

                crop_percentage = class_percentages[crop_index]
                weed_percentage = class_percentages[weed_index]

                weed_to_crop_ratio = weed_percentage /
       crop_percentage if crop_percentage != 0 else 0

title = f'Weed detected in crop: Crop: {crop_percentage:.2f}
%, Weed: {weed_percentage:.2f}%, Weed to Crop Ratio:
{weed_to_crop_ratio:.2f}'
                title_color = 'purple'
            elif "crop" in detected_classes:
                crop_index = class_labels.index("crop")
                crop_percentage = class_percentages[crop_index]
title = f'Crop detected: {crop_percentage:.2f}
%'              title_color = 'green'
            elif "weed" in detected_classes:
                weed_index = class_labels.index("weed")
                weed_percentage = class_percentages[weed_index]
title = f'Weed detected: {weed_percentage:.2f
}%'                title_color = 'red'
            else:
                title = 'Neither weed or crop detected'
                title_color = 'black'

context = {'title': title, 'title_color': title_color,
'class_percentages': class_percentages,
                        'detected_classes': detected_classes}
            print(title)
            messages.success(request, title)

            return render(request, "Home.html", context)
        except Exception as e:
            messages.error(request, f"Error: {str(e)}")
            return render(request, "Home.html")
```

```
    messages.error(request, "Invalid Request")
    return render(request, "Home.html")

def ReviewSave(req):
    if req.method=="POST":
        nm=req.POST.get('uname')
        des=req.POST.get('txt')
        x=Review(username=nm,Description=des)
        x.save()
        messages.success(req,"Review Submitted Successfully")
        return redirect(home)

def RegistrationForm(req):
    return render(req,"Registration.html")



def Registration_save(request):
    if request.method == "POST":
        nm = request.POST.get('uname')
        em = request.POST.get('email')
        passw = request.POST.get('password')
        con = request.POST.get('cpassword')
        if passw != con:
messages.error(request, "Password and confirm password do not
match.")
            return redirect(RegistrationForm)
registration = Registration(username=nm, Email=em,
Password=passw, Confirm_Password=con)
        registration.save()
        messages.success(request,"Registered Succesfully")
        return redirect(Login_Pg)

def Login_Pg(req):
    return render(req,"Login_Pg.html")

def Login_fun(request):
    if request.method=="POST":
        nm=request.POST.get('uname')
        pwd=request.POST.get('password')
if
Registration.objects.filter(username=nm,Password=pwd).exists():
            request.session['username']=nm
            request.session['Password']=pwd
            messages.success(request, "Logged in Successfully")
            return redirect(home)
```

```
        else:
            messages.warning(request, "Check Your Credentials")
            return redirect(Login_Pg)
    else:
messages.warning(request, "Check Your Credentials Or Sign Up ")
 return redirect(Login_Pg)

def Logout_fn(request):
    del request.session['username']
    del request.session['Password']
    messages.success(request, "Logged Out Successfully")
    return redirect(Login_Pg)
```

manage.py

```
#!/usr/bin/env python
"""Django's command-line utility for administrative tasks."""
import os
import sys


def main():
    """Run administrative tasks."""
os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'Weed.settings')    try:
from django.core.management import execute_from_command_line
 except ImportError as exc:
        raise ImportError(
"Couldn't import Django. Are you sure it's installed and
""available on your PYTHONPATH environment variable? Did you "
"forget to activate a virtual environment?"
        ) from exc
    execute_from_command_line(sys.argv)


if __name__ == '__main__':
    main()
```

urls2.py

```
"""Weed URL Configuration

The ‘urlpatterns‘ list routes URLs to views. For more
information please see:
    https://docs.djangoproject.com/en/3.2/topics/http/urls/
```

```
Examples:
Function views
    1. Add an import:  from my_app import views
2. Add a URL to urlpatterns:  path('', views.home, name='home')Class-based views
    1. Add an import:  from other_app.views import Home
2. Add a URL to urlpatterns: path('', Home.as_view(),
name='home')
Including another URLconf
1. Import the include() function: from django.urls import
include, path
2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
"""
from django.contrib import admin
from django.urls import path,include
from django.contrib.staticfiles.urls import
staticfiles_urlpatterns,static
from . import settings
import Frontend.urls
from Frontend import views
urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include(Frontend.urls)),
    path('/',views.RegistrationForm,name="RegistrationForm")
]

urlpatterns+=staticfiles_urlpatterns()
urlpatterns+=static(settings.MEDIA_URL,document_root=settings.MEDIA_ROOT)
```

FOGGY ROAD ALERT SYSTEM

PROJECT REPORT

Submitted By

ALJO POULOSE

Reg. No. CCAVBCA028

For the award of the Degree of

Bachelor of Computer Application(BCA)

(University of Calicut)

under the guidance of

Ms. Sharon Joy C

Assistant Professor

BACHELOR OF COMPUTER APPLICATION
DEPARTMENT OF COMPUTER SCIENCE
CHRIST COLLEGE
IRINJALAKUDA, KERALA

2021-2024

DEPARTMENT OF COMPUTER SCIENCE

CHRIST COLLEGE (AUTONOMOUS)

IRINJALAKUDA

CERTIFICATE

This is to certify that the project report entitled ***"Foggy Road Alert System"*** *is a bonafide record of the project work done by* ***Aljo Poulose*** *in partial fulfillment of the requirement for the sixth semester of* ***Bachelor of Computer Application*** *in Department of Computer Science of* ***CHRIST COLLEGE (AUTONOMOUS) IRINJALAKUDA***

.

Ms. Sharon Joy C
Assistant Professor,CS
Internal Guide

Ms. Sini Thomas
Head of the Department
Computer Science

EXTERNAL EXAMINER

INTERNAL EXAMINER

DECLARATION

We hereby declare that this project work **"FOGGY ROAD ALERT SYSTEM"** submitted to Christ College (Autonomous) Irinjalakuda, affiliated to Calicut University in partial fulfillment of the requirement for the award of the Bachelor of Computer Application, is a record of original work done by us, under the guidance of Ms. SHARON JOY C , Department of Computer Science.

Place: Irinjalakuda ALJO POULOSE

ACKNOWLEDGEMENT

First and foremost we like to thank Lord almighty for his providence and for being the guiding light throughout the project. We wish to express my sincere gratitude to our beloved Department head for giving me all the facilities for our project. We take this opportunity to express my gratitude to the class teacher Ms. SOUMYA P.S and head of the department Ms. SINI THOMAS who has been supported us throughout the course of this project. We are thankful for her aspiring guidance and valuable advice during the project work. We express my sincere thanks to my project guide Ms. SHARON JOY C for supporting and guiding throughout the project.We would take this opportunity to specially thank all other faculty members for their constant and continuous motivation. Finally we would like to thank my family and friends for giving valuable advice and moral support throughout our project.

ABSTRACT

FOGGY ROAD ALERT SYSTEM is an innovative application of computer vision and deep learning technologies, aims to significantly enhance road safety by providing real-time visibility information during adverse weather conditions. Leveraging Python, OpenCV, TensorFlow, and Keras, the system employs a Convolutional Neural Network (CNN)-based fog detection model trained on a diverse dataset to analyse images captured by strategically placed cameras. Real-time monitoring ensures prompt responses to changing fog conditions, and an alerting mechanism notifies authorities when intensity thresholds are exceeded. The user-friendly interface allows for visualizing real-time fog data and historical trends. The system's successful deployment in fog-prone areas demonstrates its potential to reduce accidents and improve traffic management. Recommendations for future enhancements include exploring advanced sensors and dynamic threshold adjustments, further solidifying its role in proactive road safety measures. Additionally, the system seamlessly integrates with Django, offering features such as real-time fog identification, continuous monitoring, user authentication, adaptability for model retraining, and scalable deployment. The comprehensive documentation ensures ease of use, fostering continuous improvement through advancements in deep learning techniques and user feedback. This integration harmonizes cutting-edge capabilities and an intuitive web application, effectively addressing the challenges posed by foggy environmental conditions.

Contents

Chapter 1

1 Introduction

Fog is a major cause of road accidents worldwide. To improve road safety and traffic management, we propose a foggy road alert system that uses a web application, a fog detection model, and an email notification system. The web application allows users to upload videos of foggy roads and view the results of fog detection and alert generation. The fog detection model uses deep learning to classify each video frame as foggy or non-foggy, and calculate the average fog level. The email notification system sends an alert to the nearest traffic police station if the fog level is high.

1.1 Overview

The FOGGY ROAD ALERT SYSTEM aims to develop a web application that warns authorities about foggy roads using video analysis and email notification. The web application allows users to upload videos of foggy roads and view the fog detection results. The fog detection model uses deep learning to classify each video frame as foggy or non-foggy, and calculate the average fog level. The email notification system sends an alert to the nearest traffic police station if the fog level is high.It is a User friendly application.

Chapter 2

2 System Analysis

2.1 Purpose

The main purpose of this application owned by Krishnapriya K ,Aljo Poulose and Sreejishna K is to warn authorities about foggy roads using video analysis and email notification to prevent accidents and road blocks.

2.1.1 Existing System

The existing systems are using sensors placed on the roads.These sensors detect the fog and generates the warning messages to authorities.Limitationss of this sytems are sensosrs are costly,requires periodic maintenance,less accuracy

2.1.2 Proposed System

The proposed system is a web application which allows users to upload videos of foggy roads and view the fog detection results. The fog detection model uses deep learning to classify each video frame as foggy or non-foggy, and calculate the average fog level. The email notification system sends an alert to the nearest traffic police station's email address if the fog level is high. The proposed system aims to improve road safety and traffic management by providing timely and accurate information about the fog conditions on the roads.

2.2 Problem definition

To understand the problem and the needs before developing it, we are designing a foggy road alert system to detect fog on roads and generate warning messages for the traffic police.

2.3 FEASIBILITY STUDY

After the problem is clearly understood and solutions proposed, the next step is to conduct the feasibility study. Feasibility study is defined as evaluation or analysis of the potential impact of the proposed project or program. The objective is to determine whether the proposed system is feasible. There are three aspects of feasibility study to which the proposed system is subjected as discussed below.

2.3.1 Technical Feasibility

Technical feasibility assess whether the current technical resources are sufficient for the new system. We can upgrade the level of technology for supporting our

website. We check whether the proposed system can be implemented in the present system without supporting the existing hardware.

2.3.2 Economical Feasibility

Economic feasibility determines whether the time and money are available to develop the system. There is no additional hardware used to develop the site, it is inexpensive to build.

2.3.3 Operational Feasibility

Operational feasibility determines if the human resources are available to operate the system once it has been installed. No extra training is needed to use this system. Anyone who has knowledge in internet in English language can easily use the system. The resources that are required to implement or install are already available with the organization.

Chapter 3

3 Software Requirement Specification

3.1 Purpose

The purpose of this document is to outline the software requirements for the Foggy Road Alert System. It provides a comprehensive overview of the functionalities, constraints, and interactions necessary for the development of the system. This document serves as a blueprint for designing and implementing a robust solution aimed at enhancing road safety during foggy conditions. It delineates the system's capabilities to detect fog, generate timely alerts to traffic authorities.This document aims to facilitate the creation of an efficient and effective foggy road alert system that mitigates risks and improves overall road safety.

3.2 Scope

The Foggy Road Alert System aims to detect foggy conditions in real-time and provide timely alerts to drivers and authorities, enhancing road safety during low visibility conditions.

3.3 Overall Description

This section provides an overview of our Foggy Road Alert System. The project entails detecting the fog percentage in uploaded videos, triggering a warning email to the nearest traffic police authorities if the percentage surpasses a predefined threshold. Upon receiving the alert, authorities can take necessary measures to manage traffic on foggy roads, thereby reducing accidents associated with fog in various locations.The main aim of our project is to ensure the safety of people travelling through these foggy roads.

3.3.1 Product Perspective

The Foggy Road Alert System operates within the broader context of road safety and transportation management systems. From a product perspective, it serves as a critical component in enhancing road safety by specifically addressing the challenges posed by foggy conditions.The system is a standalone applocation that integrates CNN-based image processing for fog detection.It interacts with users through a Django based web interface

3.3.2 Product Functionality

The Foggy Road Alert System is equipped with several key functionalities aimed at bolstering road safety during foggy conditions. Leveraging sophisticated algorithms, the system automates the process of fog detection in uploaded videos,

ensuring prompt alerts when hazardous conditions arise. Robust access controls are implemented to guarantee that only authorized users can access the fog detection features, thereby enhancing system security. Powered by Django, the system's web application provides users with an intuitive interface, facilitating seamless interaction and ease of use. Additionally, the system delivers real-time alerts to users and traffic authorities, enabling swift responses to hazardous foggy conditions and ultimately enhancing overall road safety.

3.3.3 Users and Characteristics

The primary users of the system are traffic police authorities responsible for monitoring road conditions and managing traffic flow. They utilize the system to receive timely fog alerts and coordinate responses to ensure road safety.Secondary users include drivers who rely on the system's alerts and recommendations to navigate safely during foggy conditions.The characteristics include technical proficiency,time sensitivty,reliability and accesibilty.Then there are administators who oversee the overall functioning and maintenance of the system.Administrators can monitor and manage the generation and dissemination of fog alerts.

3.4 Specific Requirements

3.4.1 Hardware Requirements

- System: IBM-Compatible PC
- Processor: Intel Core i3
- Speed: Above 1GHz
- RAM capacity: 4 GB
- Hard Dsk drive: 500 GB

3.4.2 Software Requirements

- Front End : HTML,CSS,Bootstrap
- Back End : Python,Django
- IDE : PyCharm Community Edition 2023.1.3

3.5 Functional Requirements

It contains three main modules.

- 1.Fog Detection
- 2.Alert Generation
- 3.User Interface

Fog Detection

This part detects fog in videos or camera feeds.It uses algorithms to analyze images and determine fog density. This module is responsible for the core functionality of detecting foggy conditions.

Alert Generation

This module handles the generation and dissemination of fog alerts to traffic authorities.Develop algorithms for generating fog alerts based on detected fog density and predefined thresholds.

User Interface

This module encompasses the web interface and user interactions with the Foggy Road Alert System.Design and develop user-friendly interfaces for interacting with fog detection features and alerts.

3.6 Non Functional Requirements

Non-functional requirements define the overall qualities or attributes of the resulting system. Non-functional system place restrictions on the product being developed, the developed process, and specify external constraints that the product must meet. Examples of non-functional requirements include safety, security, usability, reliability and performance requirements. Project management issues (costs, time and schedule) are often considered as non-functional requirements. The principal non - functional constraints which are relevant to critical systems :

- performance
- security
- safety
- usability

Performance

Performance requirements concern the speed of operation of a system. Types of performance requirements :

- Response requirements (how quickly the system reacts to a user input).
- Throughput requirements (how much the system can accomplish within a specified amount of time).
- Availability requirements (is the system available for service when requested by end users). The speed of operation of this system is adequate for the requirements.

Reliability

- Reliability is the ability of a system to perform its required function under stated conditions for a specified period of time.
- constraints on the runtime behavior of the system. This system is reliable because its functionalities can be done on the required conditions.

Safety

- Safety requirements are not required which exclude unsafe situation from the possible solution of the system.

Usability

Usability is the ease with which a user can learn to operate, prepare inputs for,and interpret outputs of system or components. Usability requirements include :

- information error messages.
- well-formed user interfaces.

3.7 Interface Requirements

3.7.1 Hardware interfaces

The system must run over the internet, all the hardware shall require to connect internet will be hardware interface for the system. As for example modem, WAN-LAN.

3.7.2 Software interfaces

Software interface required for the working of the project is the appropriate operating system.

3.7.3 Communication interfaces

The user of the site communicate using the network connectivity and the data set is accessed

3.8 Security Requirements

- Validation of input is handled.
- This application containing the computer systems is physically secured against arms or surreptitious entry by intruders.
- Users must be authorized carefully to reduce changes of any such user giving access to an intruder in exchange for a bribe or other favour.

3.9 Platform Used

The platform used for developing and running the project is Windows 11, the latest version of Microsoft's operating system. Windows 11 offers a redesigned and refreshed user interface, new tools and features, and improved security and performance. Windows 11 supports various web development frameworks and languages, such as HTML, CSS, JavaScript, PHP, Python, and more. Windows 11 also enables the use of machine learning models, such as the one used for fog detection, through libraries and tools like TensorFlow, PyTorch, OpenCV, and others. Windows 11 is compatible with most devices that meet the minimum system requirements.

3.10 Technologies Used

HTML

HTML stands for HyperText Markup Language. It is the standard markup language for creating web pages. HTML uses elements, which are defined by tags, to structure and label the content of a web page. HTML elements can have attributes, which provide additional information or functionality. HTML can also include other technologies, such as CSS and JavaScript, to enhance the appearance and behavior of web pages.

CSS

CSS stands for Cascading Style Sheets. It is a language that allows you to style and layout web pages by applying rules to HTML elements. CSS can control the appearance, position, size, color, font, animation, and more of the web content. CSS can also adapt to different devices, screen sizes, and orientations. CSS is one of the core technologies of the web, along with HTML and JavaScript.

PYTHON

Python is a versatile, high-level programming language known for its simplicity and readability. It offers dynamic typing and automatic memory management, making it suitable for various domains such as web development, data analysis, artificial intelligence, and automation. Python's extensive standard library provides pre-built modules and packages for tasks ranging from file manipulation to network programming. Its interpreted nature allows for rapid development and easy debugging. Python supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Its popularity is fueled by a vibrant community, extensive documentation, and numerous third-party libraries. Overall, Python's simplicity, flexibility, and robust ecosystem make it an ideal choice for beginners and seasoned developers alike.

DJANGO

Django is a high-level Python web framework that simplifies web development by emphasizing reusability, rapid development, and the principle of DRY (Don't Repeat Yourself). It follows the Model-View-Controller (MVC) architectural pattern, with models defining data structures, views handling user interface logic, and templates rendering HTML. Django's built-in features include an ORM (Object-Relational Mapping) for interacting with databases, a secure authentication system, URL routing, and a powerful admin interface for managing content. Its scalability and extensibility are further enhanced by a vast ecosystem of third-party packages. Django's emphasis on convention over configuration and its adherence to best practices make it a popular choice for building robust and maintainable web applications.

CNN

Convolutional Neural Networks (CNNs) are a class of deep neural networks primarily used for image recognition and computer vision tasks. They consist of multiple layers, including convolutional layers, pooling layers, and fully connected layers. CNNs excel at learning hierarchical representations of features from input data. Convolutional layers apply filters to input images, extracting features such as edges and textures. Pooling layers reduce spatial dimensions while preserving important features. Fully connected layers combine extracted features for classification or regression tasks. CNNs leverage parameter sharing and local connectivity to efficiently learn patterns from large datasets. Their success in image recognition tasks has made them a cornerstone in various fields, including autonomous driving, medical imaging, and facial recognition.

Chapter 4

4 Design Document

4.1 Purpose

The purpose of the foggy road alert system project is to address the safety concerns posed by reduced visibility due to foggy conditions on roads. By leveraging a combination of web application interface and a trained convolutional neural network (CNN) model, the system aims to detect and quantify the level of fog in video frames. Through real-time analysis, it provides timely alerts to the nearest traffic police stations via email when the fog level exceeds a predefined threshold. This proactive approach intends to enhance road safety by enabling authorities to take precautionary measures promptly, such as issuing advisories, implementing speed restrictions, or deploying resources to manage traffic effectively, thereby reducing the likelihood of accidents and ensuring the well-being of commuters.

4.2 Scope

The project aims to create a web application using a trained CNN model to detect fog levels in uploaded videos, issuing alerts to traffic police stations for hazardous conditions.

4.3 Overview

The purpose of this document is to help the reader to visualize the solution to the project presented.This document verifies how the design meet the requirement stimulated in the SRS document through design viewpoints.The design viewpoints will cover all design elements presented before. This document will provide a direct approach to the development of this project hence reducing feature creep and ponitedly determine the quality of the design.

4.4 Data Design

Database are the storehouses of data used in the software systems.The data is stored in tables inside the database.Several tables are created for the manipulation of the data for the system.Two essential settings for a database are

- Primary Key- The field that is unique for all the record occurrence.
- Foreign Key- The field used to set relation between tables.

Normalization

- Normalization is the process of organizing the data in the database.

- Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate the undesirable characteristics like Insertion, Update and Deletion Anomalies.
- Normalization divides the larger table into the smaller table and links them using relationship.
- The normal form is used to reduce redundancy from the database table.

Chapter 5

5 Development of the System

The development phase of the Foggy Road Alert System using CNN with Django involves the practical implementation of the proposed project. It encompasses several key steps and components that contribute to the seamless integration of Convolutional Neural Networks (CNN) for fire detection with the Django web framework

Chapter 6

6 System Testing

Testing is the penultimate step of software development.An elaborate testing of the data is prepared and the system is using the test data,while doing testing,errors are noted and correction is made.The users are trained to operate the developed system.Both hardware and software securities are made to run the developed system successfully.System testing is aimed at ensuring the system works accurately before the live operation commences.

6.1 Test Plan

6.1.1 Scope

This test plan will cover the following testing activities as identified in the testing strategy.

- White Box Testing

White box testing is a testing technique, that examines the program structure and derives test data from the program logic/code. In white box testing, the UI is bypassed. Inputs and outputs of ratings are tested directly at the code level and the results are compared against specifications. It Reveals errors in "hidden" code.

- Black Box Testing

Black - box testing is a method of software testing that examines the functionality of an application based on the specifications. Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end - user would.

- Unit Testing

The module includes four main components to undergo unit testing. It checks the syntax error, logic error and validity of the program correctness. This test will be performed by the developers.

- Integration Testing

After all components pass the unit testing, they get to be tested whether they work correctly when they are running concurrently and communicating to each other. Integration tests exercise an entire subsystem and ensure that a set of components play nicely together.

- Internal Data Testing

We will test the validity of the data before it enters the database to avoid any problems that may face us in the database. We will test the encryption of the personal information of all the users along with the user names and passwords to ensure maximum security of the users privacy.

6.1.2 Software risk issues

In this section, the plan is to test the risk involved in critical issues such as: Difficult to run on visual studio due to large loading time. Some inherent software risks such as complexity exist; also these issues need to be identified. Proper network connection Working of Mysql database

6.1.3 Features to be tested

- Test whether correct user name and password allows you to login.
- Test whether invalid user name and password prevents you from login.
- Test whether there is any connection problem in the Server.
- Test whether the student and admin details are entered correctly.
- Test whether invalid data entry allows saving data successfully.
- Test whether all pages are loaded correctly.
- Test whether watermark is embedded and extracted properly.
- Test whether the provided security works correctly.

6.2 Test consolidation

6.2.1 Test item

The items or features to be tested in the test cases are included in the document. Each and every user input is tested. The present condition of the system is tested. It is checked to make sure that environment is ready for the application to work.

6.2.2 Input specifications

The input specifications for testing the Foggy Road Alert System invlove varoius type of data to assess system's perfomance comprehensively.The main user input to be tested is uploaded videos.The model trained using CNN also will be tested

Chapter 7

7 System Implementation and Maintenance

7.1 Implementation

System implementation is the conversion of new system into an operating one which involves creating compatible files, training clients and installing hardware. User training is crucial for minimizing resistance to change and giving chance to prove its worth. Training aids user friendly manuals and healthy screens provide the user with a good start. Software maintenance follows conversion to the extent that changes are necessary to maintain satisfactory operations relative to changes in the user's environment. Maintenance often includes minor enhancements or corrections to the problem that surface late in the systems operations. In the implementation phase, the team builds the components either from scratch or by composition. Given the architecture document meant from the design phase and the requirement document from the analysis phase, the team should build exactly what has been requested, though there is still room for innovation and flexibility. For example, a component may be narrowly designed for this particular system, or the component may be made more general to satisfy a reusability

- Careful planning
- Investigation of system and constraints
- Design the methods to achieve changeover.
- Training the staff in the changed phase.
- Evaluation of change over method.
- The method of implementation and time scale to be adopted are found out initially.

7.2 Maintenance

This phase occurs as a result of deploying the whole system at the end users organization. They will perform the beta testing at the end users and inform to the developers about any needed modification to the application. The customer records all the problems that are encountered during the beta testing and reports these to the developer at regular intervals.

7.2.1 Corrective Maintenance

Even with the best quality assurance activities, it is likely that the customer will uncover defects in the software. Corrective maintenance changes the software to correct the defects. Corrective Maintenance activity may consist of repair,

restoration or replacement of equipment. This activity will be result of a regular inspection, which identifies the failure in time for corrective maintenance to be planned and scheduled, then performed during a routine maintenance shutdown.

7.2.2 Adaptive Maintenance

Over time, the original environment(CPU, operating system, business rules, external product characteristics) for which the software was developed is likely to change. Adaptive maintenance results in modification to the software to accommodate changes to its external environment.

7.2.3 Enhanced Maintenance

As software is used, the customer/user will recognize additional functions that will provide the benefit. Perfect maintenance extends the software beyond its original functional requirements.

7.2.4 Preventive Maintenance

Computer software deteriorates due to change, and because of this preventive maintenance often called software re-engineering, must be conducted to enable the software to serve the needs of its end users. Preventive maintenance makes changes to computer programs so that they can be more easily corrected, adapted and enhanced.

Chapter 8

8 Conclusion and Future Scope

8.1 Conclusion

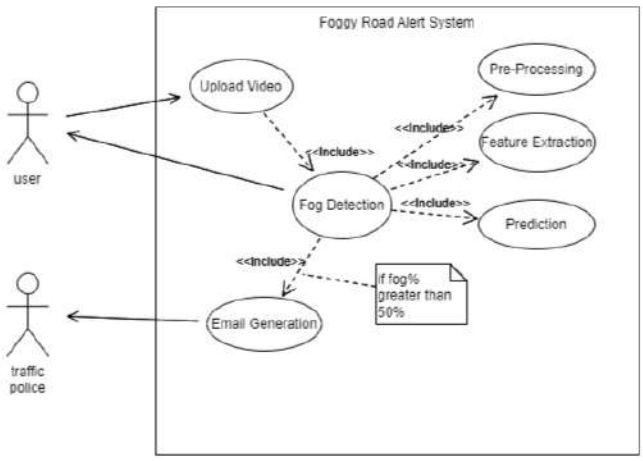
In conclusion, the Foggy Road Alert System serves two primary stakeholders: administrators and traffic authorities. Administrators utilize the dynamic admin panel to manage system features effectively, ensuring smooth operation and optimal performance. Traffic authorities play a critical role in receiving timely alerts generated by the system, enabling them to implement necessary traffic control measures and ensure road safety during foggy conditions. By providing administrators and traffic authorities with essential tools and insights, the Foggy Road Alert System contributes to enhanced road safety and improved traffic management practices.

8.2 Future Scope

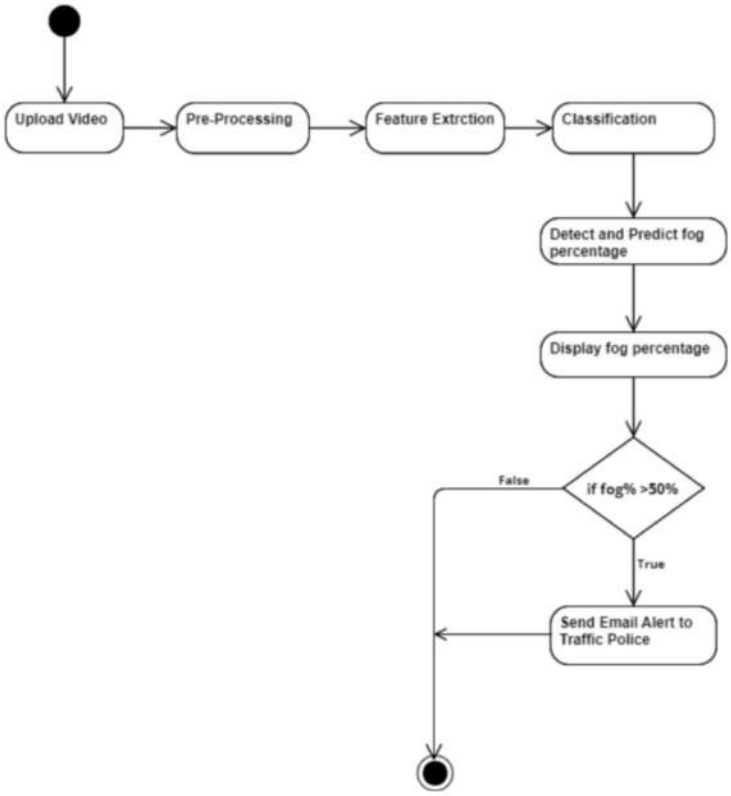
- Implement predictive analytics for proactive measures.
- Collaborate with navigation systems for safer routes.
- Engage users with crowdsourced fog reports.
- Ensure scalable and reliable system architecture.

Appendix

A Usecase Diagram

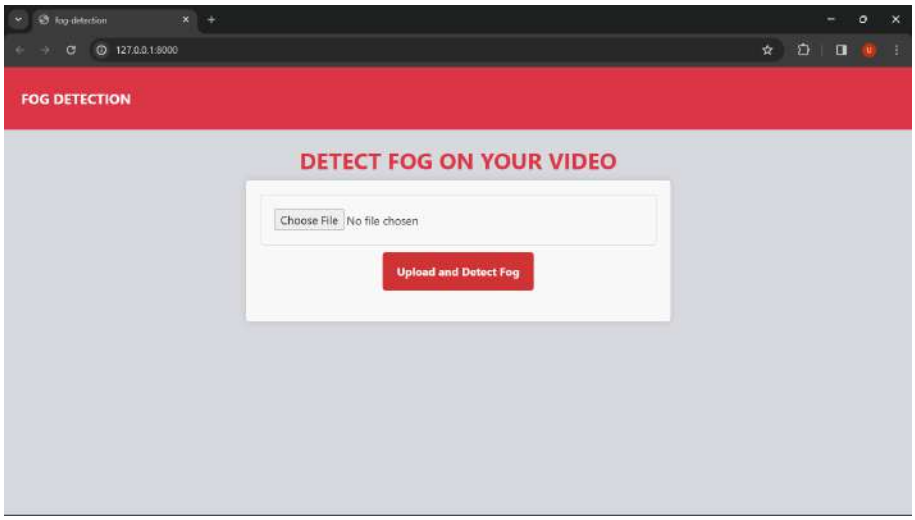


B Activity Diagram

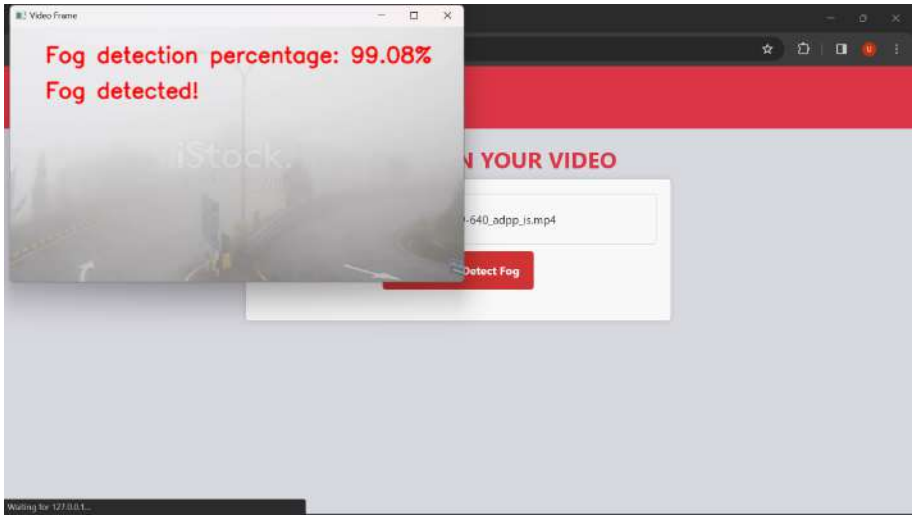


C USER INTERFACES

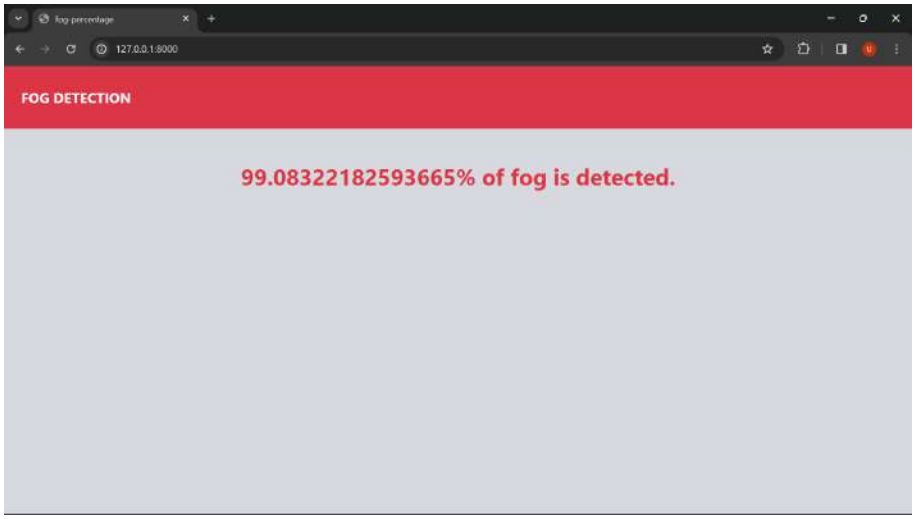
C.1



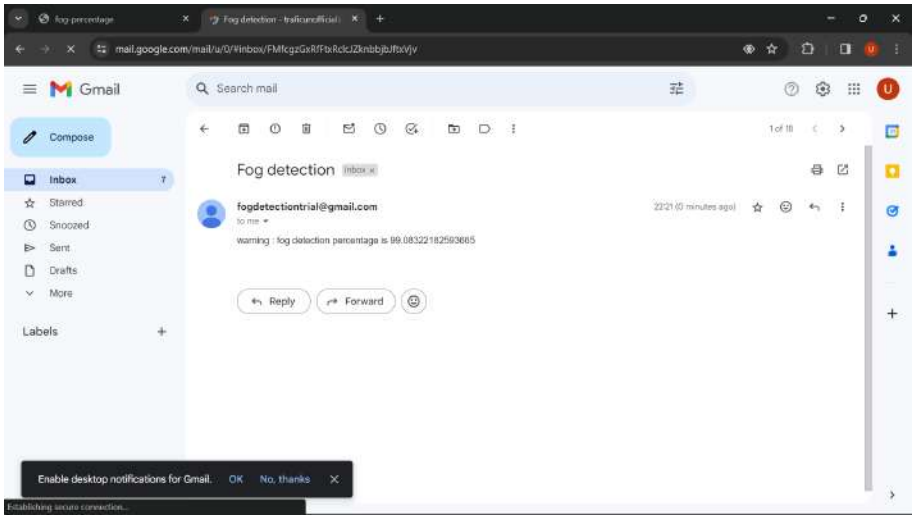
C.2 FOG DETECTION



C.3 FOG PERCENTAGE CALCULATION



C.4 ALERT


Gmail
Search mail
Compose
Inbox
Starred
Snoozed
Sent
Drafts
More
Labels
Fog detection
fogdetectiontrial@gmail.com
warning : fog detection percentage is 99.08322182593665
Reply
Forward
Enable desktop notifications for Gmail.
OK
No, thanks

D CODE

views.py

```
from django.shortcuts import render
import cv2
import numpy as np
from django.core.files.storage import FileSystemStorage
import os
from django.core.mail import send_mail
from keras.models import load_model

def home(request):
    if request.method=="POST" and 'video_file' in request
        .FILES:
        videofile =request.FILES['video_file']
        file_storage=FileSystemStorage()
        filename=file_storage.save(videofile.name,
            videofile)

        video_path =os.path.join(file_storage.location,
            filename)
        if not os.path.exists(video_path):
            error_message = "Video file does not exist."
            return render(request, 'detection_error.html
                ', {'error': error_message})

        cap= cv2.VideoCapture(video_path)
        import tensorflow as tf
        print("TensorFlow version:", tf.__version__)

        model = load_model('C:/Users/USER/OneDrive/
            Desktop/pfog/fog_detection-1/fog_detection/
            fog_detection_project/keras_model.h5')

        total_fog_percetage = []

        if not cap.isOpened():
            error_message="Error opening video file."
            return render(request,'detection_error.html
                ',{'error':error_message})

        while True:

            ret, frame = cap.read()
```

```
        if not ret:
            break

        resized_frame = cv2.resize(frame, (224, 224))

        fog_probability = model.predict(np.
            expand_dims(resized_frame, axis=0))[0, 0]

        threshold = 0.5

        if fog_probability > threshold:

            total_fog_percetage.append(
                fog_probability*100)
            percentage_text = f"Fog detection
                percentage: {fog_probability * 100:.2f
                }%"
            cv2.putText(frame, percentage_text, (50,
                50), cv2.FONT_HERSHEY_SIMPLEX, 1, (0,
                0, 255), 2, cv2.LINE_AA)

            cv2.putText(frame, "Fog detected!", (50,
                100), cv2.FONT_HERSHEY_SIMPLEX, 1, (0,
                 0, 255), 2, cv2.LINE_AA)
        else:

            no_fog_percentage = 0

            cv2.putText(frame, "No fog detected",
                (50, 100), cv2.FONT_HERSHEY_SIMPLEX,
                1, (0, 255, 0), 2, cv2.LINE_AA)

        cv2.imshow('Video Frame', frame)

        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

cap.release()
cv2.destroyAllWindows()

try:
    average_fog_percentage = sum(
        total_fog_percetage) / len(
        total_fog_percetage)
```

```
            if  average_fog_percentage >= 50 :
                subject = 'Fog detection'
                mail_content =f'warning : fog detection
                    percentage is {average_fog_percentage
                    }'
                from_mail = 'fogdetectiontrial@gmail.com'
                recipient_list = ['traficunofficial@gmail
                    .com']
                send_mail(subject, mail_content,
                    from_mail, recipient_list)

                return render(request,'fog_percentage.
                    html',{'f': average_fog_percentage })

        except ZeroDivisionError:
            print("Error: Division by zero. Unable to
                calculate the average.")

            return render(request,'fog_percentage.html
                ',{'f':no_fog_percentage})
    else:
        return render(request,'home.html')
```

fogdetection(2-01-2024).ipynb

```
import os
import cv2
import numpy as np
import tensorflow as tf
from tensorflow.keras.preprocessing import image
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing.image import
    ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D,
     Flatten, Dense

train_dir = "C:/Users/my/OneDrive/Desktop/project/
    fog_detection/fog_dataset"

train_datagen = ImageDataGenerator(rescale=1./255,
    validation_split=0.2)

train_generator = train_datagen.flow_from_directory(
    train_dir,
```

```
    target_size=(150, 150),
    batch_size=32,
    class_mode='binary',
    subset='training'
)

validation_generator = train_datagen.flow_from_directory(
    train_dir,
    target_size=(150, 150),
    batch_size=32,
    class_mode='binary',
    subset='validation'
)

model = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(32, (3, 3), activation='relu',
        input_shape=(150, 150, 3)),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Conv2D(64, (3, 3), activation='relu')
        ,
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Conv2D(128, (3, 3), activation='relu
        '),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(512, activation='relu'),
    tf.keras.layers.Dense(1, activation='sigmoid')
])

model.compile(optimizer='adam', loss='binary_crossentropy
    ', metrics=['accuracy'])
model.fit(train_generator, validation_data=
    validation_generator, batch_size=32, epochs=20,
    validation_steps=len(validation_generator))
model.save("fog_clear_model_2.h5")
model = load_model("keras.model.h5")
```

home.html

```
<!DOCTYPE html>
{% load static %}
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
        initial-scale=1.0">
```

```
    <title>fog-detection</title>
    <link rel="stylesheet" href="https://maxcdn.
        bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min
        .css">
 <link rel="stylesheet" href="{% static 'home.css' %}">
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.
    com/bootstrap/4.5.2/css/bootstrap.min.css">

  </head>
<body style="background-color: rgb(216, 216, 223);">

  <nav class="navbar-nav navbar-expand-lg navbar-light bg
      -danger p-4">
  <a class="navbar-brand" href="{% url 'fog_detection:
      home' %}" style="color: white;font-weight: bold;">
      FOG DETECTION</a>

  </nav>

  <div class="pt-4">
  <h2 style="font-weight:bold ;" class="text-danger">
      DETECT FOG ON YOUR VIDEO </h2>
<div class="form-container">
<form method="POST" enctype="multipart/form-data">
  {% csrf_token %}
  <div class="text-center">
  <input type="file" name="video_file">

  <input  type="submit" value="Upload and Detect Fog">
</div>

</form>
</div>
</div>


</body>
</html>
```

fogpercentage.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
        initial-scale=1.0">
```

```
    <title>fog-percentage</title>
    <link rel="stylesheet" href="https://maxcdn.
        bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min
        .css">

</head>
<body style="background-color: rgb(216, 216, 223);">
    <nav class="navbar-nav navbar-expand-lg navbar-light
        bg-danger p-4">
        <a class="navbar-brand" href="{% url '
            fog_detection:home' %}" style="color: white;
            font-weight: bold;">FOG DETECTION</a>

        </nav>
    <div class="form-container pt-5">

        <h2 class="text-danger font-weight-bold text-
            center">{{f}}% of fog is detected.</h2>

    </div>

</body>
</html>
```

settings.py

```
from pathlib import Path
import os

BASE_DIR = Path(__file__).resolve().parent.parent

SECRET_KEY = 'django-insecure-uqy^qc_gs6#n6a1juaeyc^x2qy!
    l$@18xom0b3*&g$b7%!j@68'

DEBUG = True

ALLOWED_HOSTS = []

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'fog_detection',
]
```

```
MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware
        ',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.
        AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware
        ',
    'django.middleware.clickjacking.
        XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'fog_detection_project.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.
            DjangoTemplates',
        'DIRS': [ BASE_DIR / 'templates'],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug
                    ',
                'django.template.context_processors.
                    request',
                'django.contrib.auth.context_processors.
                    auth',
                'django.contrib.messages.
                    context_processors.messages',
            ],
        },
    },
]

WSGI_APPLICATION = 'fog_detection_project.wsgi.
    application'

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
    }
```

```
}

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.
            UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.
            MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.
            CommonPasswordValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.
            NumericPasswordValidator',
    },
]

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_TZ = True

STATIC_URL = 'static/'
STATICFILES_DIRS = [BASE_DIR / 'static']

DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
MEDIA_URL = '/media/'
MEDIA_ROOT = os.path.join(BASE_DIR , 'media')

EMAIL_BACKEND = 'django.core.mail.backends.smtp.
    EmailBackend'
EMAIL_HOST = 'smtp.gmail.com'
EMAIL_USE_TLS = True
EMAIL_PORT = 587
EMAIL_HOST_USER = 'fogdetectiontrial@gmail.com'
EMAIL_HOST_PASSWORD = 'anmh saxu exmh pfyr'
```

LARKSMITE LAB MANAGEMENT

PROJECT REPORT

Submitted By

AMAN ANIFER

Reg. No. CCAVBCA004

For the award of the Degree of

Bachelor of Computer Application (BCA)

(University of Calicut)

under the guidance of

Ms. Soumya P S

Assistant Professor

BACHELOR OF COMPUTER APPLICATION
DEPARTMENT OF COMPUTER SCIENCE
CHRIST COLLEGE
IRINJALAKUDA, KERALA
2021-2024

DEPARTMENT OF COMPUTER SCIENCE

CHRIST COLLEGE (AUTONOMOUS)

IRINJALAKUDA

CERTIFICATE

This is to certify that the project report entitled ***"Larksmite Lab Management"*** *is a bonafide record of the project work done by* ***Aman Anifer*** *in partial fulfillment of the requirement for the sixth semester of* ***Bachelor of Computer Application*** *in Department of Computer Science of* ***CHRIST COLLEGE (AUTONOMOUS) IRINJALAKUDA***

.

Ms. Soumya P S
Assistant Professor
Internal Guide

Ms. Sini Thomas
Head of the Department
Computer Science

EXTERNAL EXAMINER

INTERNAL EXAMINER

DECLARATION

I hereby declare that this project work **"LARKSMITE LAB MANAGEMENT"** submitted to Christ College (Autonomous) Irinjalakuda, affiliated to Calicut University in partial fulfillment of the requirement for the award of the Bachelor of Computer Application, is a record of original work done by me, under the guidance of Ms. SOUMYA P S, Assistant Professor, Department of Computer Science.

Place: Irinjalakuda AMAN ANIFER

ACKNOWLEDGEMENT

First and foremost we wish to thank Lord almighty for his providence and for being the guiding light throughout the project. We take this opportunity to express our gratitude to our beloved class teacher as well as project guide, Ms. SOUMYA P S, who has been hugely supportive throughout the course of this project. We wish to express our sincere gratitude to our head of department, Ms. SINI THOMAS for giving us all the required facilities for our project. We are thankful for their aspiring guidance and valuable advice during the project work. We would like to take this opportunity to specially thank all other faculty members for their constant and continuous motivation. Finally we would like to thank our family and friends for giving valuable advice and moral support throughout our project.

ABSTRACT

Larksmite Lab Management is an innovative software introduced to manage computer labs. The software is enriched with multiple features, including - status monitoring, process monitoring, screen monitoring, power control, message passing and so on. The client side of the software automatically runs in the background in all computers in the lab. All these features make this software powerful and useful in computer labs.

Contents

Chapter 1

1 Introduction

Larksmite is a sophisticated computer lab management software designed to provide administrators with seamless control and oversight of connected devices. Built using Flutter and Dart, the software offers a bird's eye view of the lab's layout, enabling easy device identification. With features such as screen sharing, power controls, and process monitoring, administrators can efficiently manage resources and troubleshoot issues remotely. The project employs RSA public-key cryptography for secure communication and utilizes SQLite for data storage. Larksmite aims to simplify computer lab administration, offering a centralized solution for monitoring, controlling, and maintaining an optimal computing environment.

1.1 Overview

Larksmite is a streamlined computer lab management solution, prioritizing user-friendly interfaces for efficient device identification. With features such as screen sharing, power controls, and detailed reporting, the platform simplifies administration tasks. Utilizing Flutter and Dart, it ensures responsiveness, while employing RSA public-key cryptography and SQLite for secure communication and data storage. Larksmite's core objective is to centralize control and optimize the management of multiple devices within a network.

Chapter 2

2 System Analysis

2.1 Purpose

The purpose of Larksmite is to streamline computer lab management, offering a centralized solution for efficient device control and monitoring. This project aims to simplify administration tasks, enhance accessibility, and optimize the management of multiple devices within a network.

2.1.1 Existing System

Existing computer lab management systems have limitations in scalability, security, and user interface. Most of them also only work on Microsoft Windows, prompting the development of a cross-platform software. This new system aims to enhance performance, address security concerns, and offer a more user-friendly solution for streamlined device control and monitoring.

2.1.2 Proposed System

Larksmite is envisioned as a revolutionary solution for computer lab management, aiming to address the limitations of the existing system. The proposed system prioritizes a centralized and user-friendly approach to streamline device control and monitoring. Larksmite seeks to enhance efficiency, security, and user experience in computer lab administration, providing administrators with an optimized tool for seamless device management

2.2 Problem Definition

The existing computer lab management system faces challenges in scalability, security, and user interface, hindering efficient device control and monitoring. These limitations necessitate the development of Larksmite.

2.3 Feasibility Study

After the problem is clearly understood and solutions proposed, the next step is to conduct the feasibility study. Feasibility study is defined as evaluation or analysis of the potential impact of the proposed project or program. The objective is to determine whether the proposed system is feasible. There are three aspects of feasibility study to which the proposed system is subjected as discussed below.

2.3.1 Technical Feasibility

Technical feasibility assesses whether the current technical resources are sufficient for the new system. The selected combination of Flutter and Dart offers a robust framework for building a responsive and visually appealing user interface, which is cross-platform by default. This ensures a wide array of compatibility.

2.3.2 Economical Feasibility

Economic feasibility determines whether time and money are available to develop the system. There is no additional hardware needed to develop the software, neither does it need any extra hardware to run.

2.3.3 Operational Feasibility

Operational feasibility determines if the human resources are available to operate the system once it has been installed. No extra training is needed to use this system. Anyone who has basic computer knowledge can easily use the system. The resources that are required to implement or install are already available with the organization.

Chapter 3

3 Software Requirement Specification

3.1 Purpose

The purpose of the software specification is to precisely outline the requirements of Larksmite, serving as a blueprint for its development. This document articulates the specific objectives the software aims to achieve, detailing user interactions, system behaviors, and performance expectations. It is primarily intended for the computer lab administrator for managing and understanding the proposed system.

3.2 Scope

Our project scope encompasses the development of Larksmite, a computer lab management software. It includes features such as computer lab layout visualization, real-time screen monitoring, power controls, and detailed reporting to optimize the efficiency of computer lab administration.

3.3 Overall Description

This section gives an overview of our software - Larksmite. The software offers an intuitive interface, allowing administrators a bird's eye view of the lab layout for streamlined device identification. Incorporating features such as real-time screen monitoring, power controls, and detailed reporting, Larksmite ensures efficient and secure administration.

3.3.1 Product Perspective

LarkSmite is mainly used for managing a computer lab in any institutions. It can be used to monitor each computer screens, orchestrate power options and generate reports.

3.3.2 Product Functionality

Tasks such as device identification, real-time screen monitoring, power controls, process monitoring and detailed reporting are various functions of LarkSmite

3.3.3 Users and Characteristics

There are two types of users within the LarkSmite system: administrators and clients. Administrators, tasked with overseeing computer labs, require access to comprehensive management features, while clients, interacting with the client computer system, remain oblivious to the existence of monitoring tool which always runs in the background.

3.4 Specific Requirements

3.4.1 Hardware Requirements

- System: IBM-Compatible PC
- Processor: Pentium IV or above
- Speed: Above 1GHz
- RAM capacity: 512 MB
- Hard Dsk drive: 40 GB
- Keyboard: Standard
- Mouse: Standard
- Monitor: SVGA Color

3.4.2 Software Requirements

- Operating System: Ubuntu or Gnome DE based Linux
- Languages used: Dart
- Database: SQLite
- Technologies used: Flutter, Shelf/Dart

3.5 Functional Requirements

It contains two main modules.

- Teacher
- Student

Teacher

In LarkSmite, teachers have the functional requirement to use the system as administrators to oversee and manage computer labs. This includes tasks such as device identification, real-time screen monitoring, and utilizing power controls, providing them with the necessary tools for effective supervision and control in an educational setting.

Student

Students in LarkSmite have the functional requirements focused on interacting with their specified client computer, performing their assigned lab task.

3.6 Non Functional Requirements

Non-functional requirements define the overall qualities or attributes of the resulting system. Non-functional system place restrictions on the product being developed, the developed process, and specify external constraints that the product must meet. Examples of non-functional requirements include security, usability, reliability and performance requirements. The principal non - functional constraints which are relevant to critical systems:

- Performance
- Security
- Usability

Performance

Performance requirements concern the speed of operation of a system. Types of performance requirements:

- Response requirements (how quickly the system reacts to a user input).
- Throughput requirements (how much the system can accomplish within a specified amount of time).
- Availability requirements (is the system available for service when requested by end users). The speed of operation of this system is adequate for the requirements.

Reliability

- Reliability is the ability of a system to perform its required function under stated conditions for a specified period of time.
- Constraints on the runtime behavior of the system. This system is reliable because its functionalities can be done on the required conditions.

Safety

Safety requirements refer to the specific criteria, conditions, and constraints that must be met to ensure the safe operation of a software system.

Usability

Usability is the ease with which a user can learn to operate, prepare inputs for, and interpret outputs of system or components. Usability requirements include:

- Understandable error messages.
- Well-formed user interfaces.

3.7 Interface Requirements

3.7.1 Hardware Interfaces

The system must be run in a LAN network, all client devices shall be required to connect to the same LAN network, for example, a WIFI access point or an Ethernet LAN network.

3.7.2 Software Interfaces

Software interface required for the working of the project is the appropriate operating system.

3.7.3 Communication Interfaces

The clients communicate using HTTPS and Secure Websocket.

3.8 Security Requirements

- Only authorized clients are allowed to connect.
- All communication between administrators and client devices is encrypted.
- Unauthorized devices cannot act as a server and clients will not connect to imposter servers

3.9 Platform Used

Ubuntu is a Linux distribution based on Debian and is free and open-source, making it a preferred choice for many institutions. It is developed by Canonical. The default distribution uses the Gnome Desktop Environment, with official support for other desktop environments like KDE, XFCE and Mate. The latest stable version is Ubuntu 23.10 (Mantic Minotaur) and the latest LTS (Long-Term Support) version is Ubuntu 22.04 LTS (Jammy Jellyfish). LarkSmite is designed to operate seamlessly within Linux environments, leveraging the robust features and stability offered by this platform.

3.10 Technologies Used

Dart

Dart serves as the cornerstone programming language in LarkSmite, providing the foundation for implementing the software's logic and functionality. Known for its simplicity and efficiency, Dart facilitates the development of a responsive and intuitive system. With features like strong typing and Just-In-Time (JIT) compilation, Dart enhances the codebase, contributing to the overall reliability and performance of LarkSmite's computer lab management solution.

Flutter

Flutter is an open-source UI (User Interface) software development kit (SDK) created by Google. It allows developers to build native applications for Android, Linux, Windows, iOS as well as macOS platforms using a single codebase. Flutter utilizes the Dart programming language. Flutter offers a hot reload feature, allowing developers to see changes in real-time as they modify the code.

SQLite

LarkSmite has opted for SQLite due to its lightweight and embedded nature. The choice of SQLite aligns with the project's goals of simplicity and seamless integration within the context of computer lab management, providing an efficient and self-contained solution for data storage and retrieval.

Chapter 4

4 Design Document

4.1 Purpose

The purpose of LarkSmite is to revolutionize computer lab management by offering administrators a centralized and user-friendly platform. Through intuitive interfaces and features like real-time screen monitoring, power controls, and detailed reporting, LarkSmite aims to streamline tasks for administrators, teachers, and students. Developed with Flutter in Dart language, the system prioritizes responsiveness and scalability, aiming to enhance the efficiency of computer lab administration within educational environments.

Currently built for Linux environment, LarkSmite seeks to optimize the computer lab management experience. With a focus on security through RSA public-key cryptography and SSL, the system aims to provide a robust, secure, and accessible solution for overseeing and controlling multiple devices within educational settings.

4.2 Scope

Our project, LarkSmite, primarily encompasses the development of a computer lab management software tailored for Linux environments. It includes features for teachers, and students, focusing on efficient device control, real-time screen monitoring, and detailed reporting within the educational context

4.3 Overview

The purpose of this document is to provide a comprehensive guide to LarkSmite, a specialized computer lab management software designed for Linux environments. It outlines the features, functionalities, and the technology stack utilized in the development of the system. By detailing the project's scope, objectives, and key components, this document aims to offer a clear understanding of LarkSmite's role in revolutionizing computer lab administration within educational institutions.

4.4 Data Design

Database is the storehouses of data used in the software systems. The data is stored in tables inside the database. Several tables are created for the manipulation of the data for the system. Two essential settings for a database are

- Primary Key- The field that is unique for all the record occurrence.
- Foreign Key- The field used to set relation between tables.

Normalization

- Normalization is the process of organizing the data in the database.
- Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate the undesirable characteristics like Insertion, Update and Deletion Anomalies.
- Normalization divides the larger table into the smaller table and links them using relationship.
- The normal form is used to reduce redundancy from the database table.

Tables

Server Asymmetric Key

Name	DataType	Constraints	Description
key_id	INTEGER(5)	Primary Key	ID of public-private key pair
public_key	BLOB	Not Null	Public key
private_key	BLOB	Not Null	Private key

Certificates

Name	DataType	Constraints	Description
cert_id	INTEGER(5)	Primary Key	ID of SSL certificate
public_cert	BLOB	Not Null	Public key
private_key	BLOB	Not Null	Private key

IP Address

Name	DataType	Constraints	Description
address	TEXT(20)	Not Null	IP Address of server
cert_id	INTEGER(5)	Foreign Key	Corresponding certificate cert_id

Shared User Key

Name	DataType	Constraints	Description
name	VARCHAR(50)	Not Null	Name of client
shared_key	TEXT(6)	Not Null	Authorization key of client

Chapter 5

5 Development of the System

The software is split into two - server side and client side. In the server side, the project is divided based on major features like networking, datastore, user interface. Each feature is further sub-divided into presentation, domain and data layer to avoid unwanted coupling and increase cohesion. This ensures that each submodule is modular, that is, it can be replaced without affecting the other parts.

In the client side, the project is divided into two major parts - platform dependent and platform independent code. The platform independent code implements features which are not dependent on the platform in which it will be run on. The platform dependent code is dependent on the platform, thus it is different on different operating systems and/or versions. This separation is designed using interfaces, which defines common functions which should be implemented by all platform dependent implementors. By employing such a split, we can achieve optimal code sharing and enhance efficiency.

Chapter 6

6 System Testing

Testing is the penultimate step of software development. An elaborate suite of test data is prepared and the system is tested using this test data. Corrections are made using the results of failed tests and any changes are immediately tested against this test suite, to detect regression. System testing is aimed at ensuring the system works accurately before the live operation commences.

6.1 Test Plan

6.1.1 Scope

This test plan will cover the following testing activities as identified in the testing strategy.

White Box Testing: White box testing is a testing technique, that examines the program structure and derives test data from the program logic/code. In white box testing, the UI is bypassed. Testing is done directly at the code level and the results are compared against specifications. It reveals errors in "hidden" code.
Black Box Testing: Black - box testing is a method of software testing that examines the functionality of an application based on the specifications. Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end-user would.
Unit Testing: The module includes four main components to undergo unit testing. It checks the syntax error, logic error and validity of the program correctness. This test will be performed by the developers.
Integration Testing: After all components pass the unit testing, they get to be tested whether they work correctly when they are running concurrently and communicating to each other. Integration tests considers the entire subsystem and ensure that a set of components play nicely together.
Internal Data Testing: We will test the validity of the data before it enters the database to avoid any problems that may be faced in the database. We will test the encryption of the personal information of all the users along with the admin user names and passwords to ensure maximum security of the admin user.

6.1.2 Software Risk Issues

In this section, the plan is to test the risk involved in critical issues such as:

- Difficulty in setting up the dependencies
- Establishing proper network security

- Ensuring data transmission security

6.1.3 Features to be Tested

- Test whether correct admin user name and password allows you to login.
- Test whether invalid admin user name and password prevents you from login.
- Test whether admin can open valid layout files.
- Test whether server software will not crash when opening invalid layout file
- Test whether server software will start service advertising and start server
- Test whether server software is able to generate SSL certificate and RSA key pair
- Test whether server software is able to perform client functions and display results
- Test whether client software continuosly searches for server in network
- Test whether client connects to server with valid client details
- Test whether client automatically connects to server when details are saved
- Test whether client responds to server requests
- Test whether client detects a fake server and doesn't connect
- Test whether client restarts from potential errors and keep running

6.2 Test Consolidation

6.2.1 Test Item

The items or features to be tested in the test cases are included in the document. Each and every input is tested. The present condition of the system is tested. It is checked to make sure that environment is ready for the application to work.

6.2.2 Input Specifications

These are the inputs required to execute the test case. All required inputs including data elements and values are given below.

Data	Values
Username	A valid username
Password	Strong combination of characters and numbers

Chapter 7

7 System Implementation and Maintenance

7.1 Implementation

System implementation is the conversion of a proposed system into an operational one which involves writing code, training clients and installing softwares. User training is crucial for helping them understand the versatily and helpfulness of the new system.

Software maintenance becomes necessary to ensure the system continues to operate satisfactorily in response to changes in the user's environment. Maintenance activities often involve making minor enhancements or corrections to address issues that may arise during system operations.

In the implementation phase, the team builds the components either from scratch or by composition. Given the architecture document from the design phase and the requirement document from the analysis phase, the team should build exactly what has been requested, though there may be still room for innovation and flexibility. For example, a component may be narrowly designed for this particular system, or the component may be made more general to satisfy a reusability

- Careful planning
- Investigation of system and constraints
- Design the methods to achieve changeover.
- Training the staff in the changed phase.
- Evaluation of changeover method.
- The method of implementation and time scale to be adopted are found out initially.

7.2 Maintenance

This phase occurs as a result of deploying the whole system at the end users organization. They will perform the beta testing and inform to the developers about any required modification to the application. The customer records all the problems that are encountered during the beta testing and reports these to the developer at regular intervals.

7.2.1 Corrective Maintenance

Even with the best quality assurance activities, it is likely that the customer will uncover defects in the software. Corrective maintenance changes the software to correct the defects. Corrective Maintenance activity may consist of repair,

restoration or replacement of equipment. This activity will be result of a regular inspection, which identifies the failure in time for corrective maintenance to be planned and scheduled, then performed during a routine maintenance shutdown.

7.2.2 Adaptive Maintenance

Over time, the original environment (CPU, operating system, business rules, external product characteristics) for which the software was developed is likely to change. Adaptive maintenance results in modification to the software to accommodate changes to its external environment.

7.2.3 Enhanced Maintenance

As software is used, the customer/user will recognize additional functions that will provide added benefits. Perfect maintenance extends the software beyond its original functional requirements.

7.2.4 Preventive Maintenance

Computer software deteriorates due to change, and because of this, preventive maintenance often called software re-engineering, must be conducted to enable the software to serve the needs of its end users. Preventive maintenance makes changes to computer programs so that they can be more easily corrected, adapted and enhanced.

Chapter 8

8 Conclusion and Future Scope

8.1 Conclusion

The objective of LarkSmite is to streamline and enhance the efficiency of managing computer labs in educational institutions or any organization that operates computer labs. It enables the admin/teacher to manage all the computers through a single admin computer. The admin is able to monitor the screens of each computer, view running processes, see basic information about the computer, send messages, perform power controls like sleep, restart, shutdown remotely and generate PDF reports with screenshots. The student's computer behaves the same way, with the client software running in the background.

8.2 Future Scope

- Allow sharing and showing teacher's computer screen on all clients
- Allow remote access similar to VNC, RDP and TeamViewer.
- Add file sharing
- Perform seamless upgrade of software

Appendix

A Activity Diagram

Activity Diagrams are used to illustrate the flow of control in a system and refer to the steps involved in the execution of a use case. It is a type of behavioral diagram and we can depict both sequential processing and concurrent processing of activities using an activity diagram ie an activity diagram focuses on the condition of flow and the sequence in which it happens.
Some Activity Diagram charting forms:

A.1 Initial State

The initial state marks the entry point and the state of the system before the application is opened

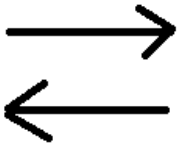
A.2 Final State

The state which the system reaches when a particular process or activity ends is known as a Final State or End State.

A.3 Activity State

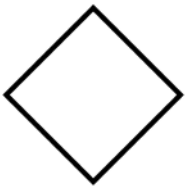
An activity represents execution of an action on objects or by objects. Any action or event that takes place is represented using an activity.

A.4 Control Flow



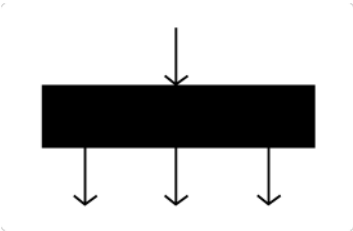
They are used to show the transition from one activity state to another activity state.

A.5 Decision Node



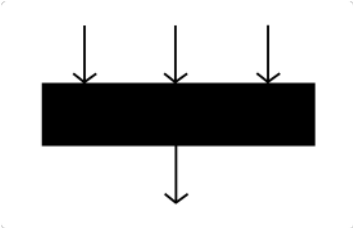
When we need to make a decision before deciding the flow of control, we use the decision node. The outgoing arrows from the decision node can be labelled with conditions.

A.6 Fork



Fork nodes are used to support concurrent activities. We use a fork node when multiple activities get executed concurrently.

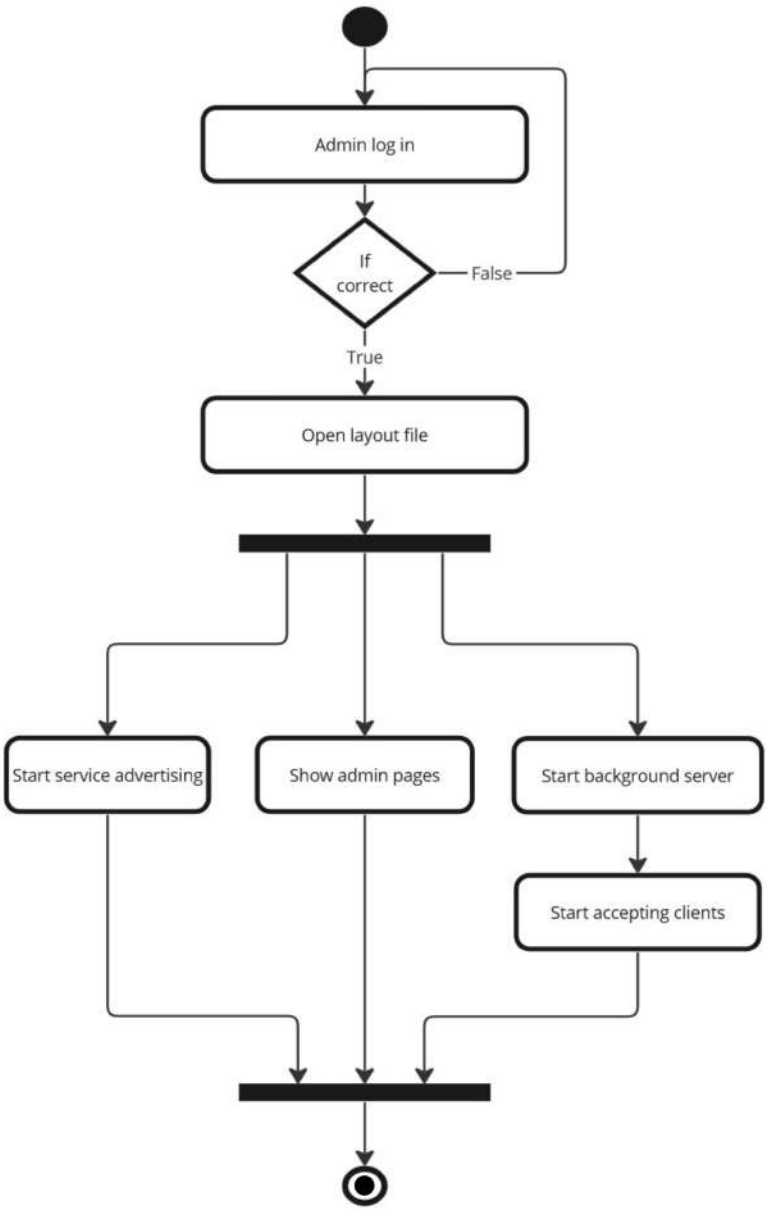
A.7 Join



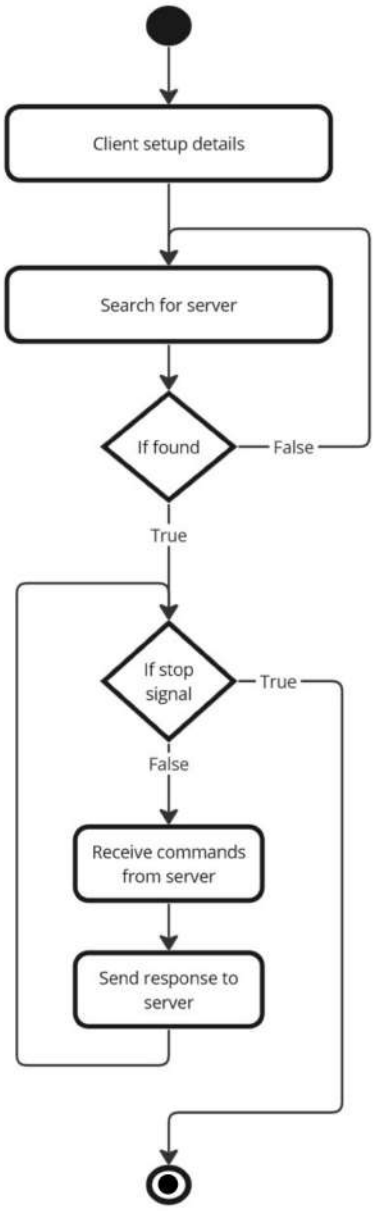
Join nodes are used to support concurrent activities converging into one.

A.8 Activity Diagram

A.8.1 Server Side



A.8.2 Client Side



B Sequence Diagram

Sequence diagrams show elements as they interact over time and they are organized according to object (horizontally) and time (vertically). They capture the interaction between objects in the context of a collaboration. Sequence diagrams are time focused and they show the order of the interaction visually by using the vertical axis of the diagram to represent time.

B.1 Actor

An actor represents a type of role where it interacts with the system and its objects. An actor is always outside the scope of the system. We use actors to depict various roles including human users and other external subjects

B.2 Lifeline

A lifeline is a named element which depicts an individual participant in a sequence diagram. A lifeline always portrays an object internal to the system.

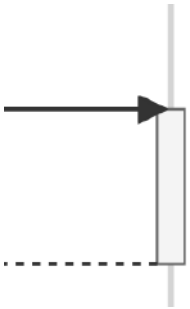
B.3 Messages

Communication between objects is depicted using messages. The messages appear in a sequential order on the lifeline.

B.4 Response Messages

Reply messages are used to show the message being sent from the receiver to the sender.

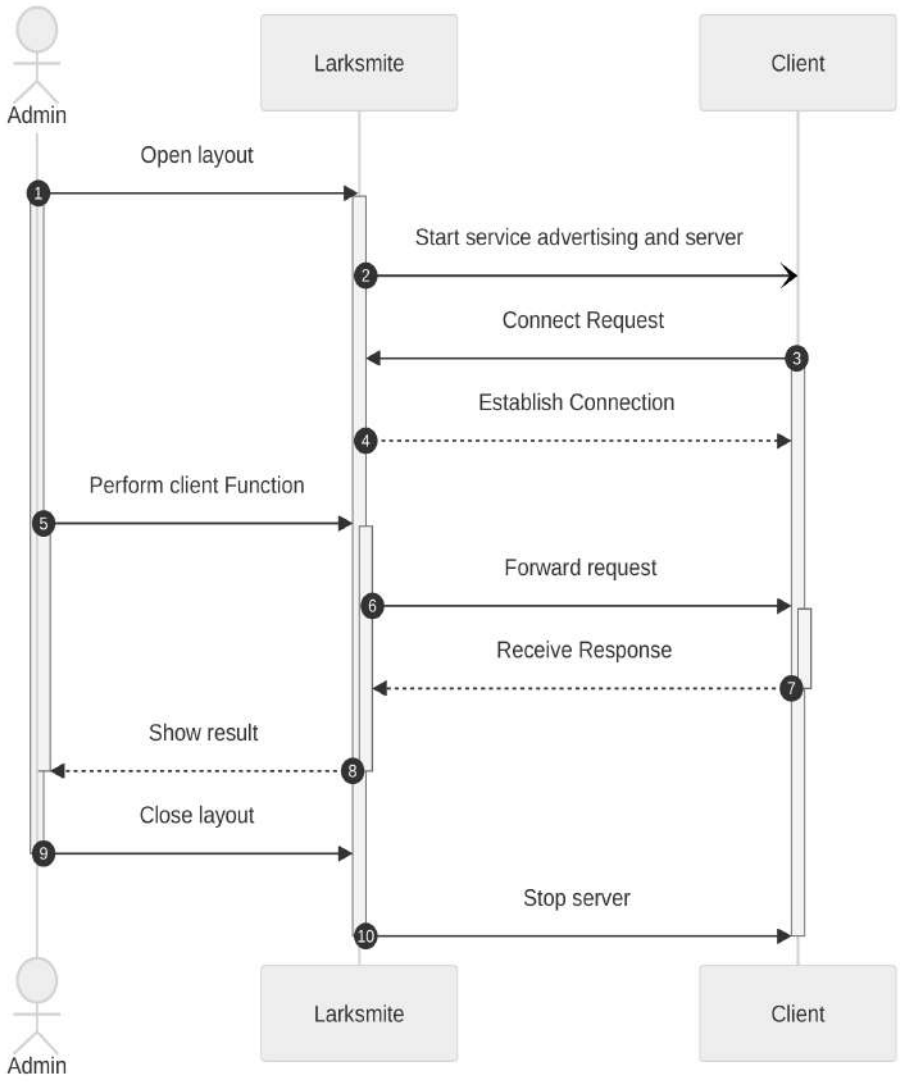
B.5 Activation



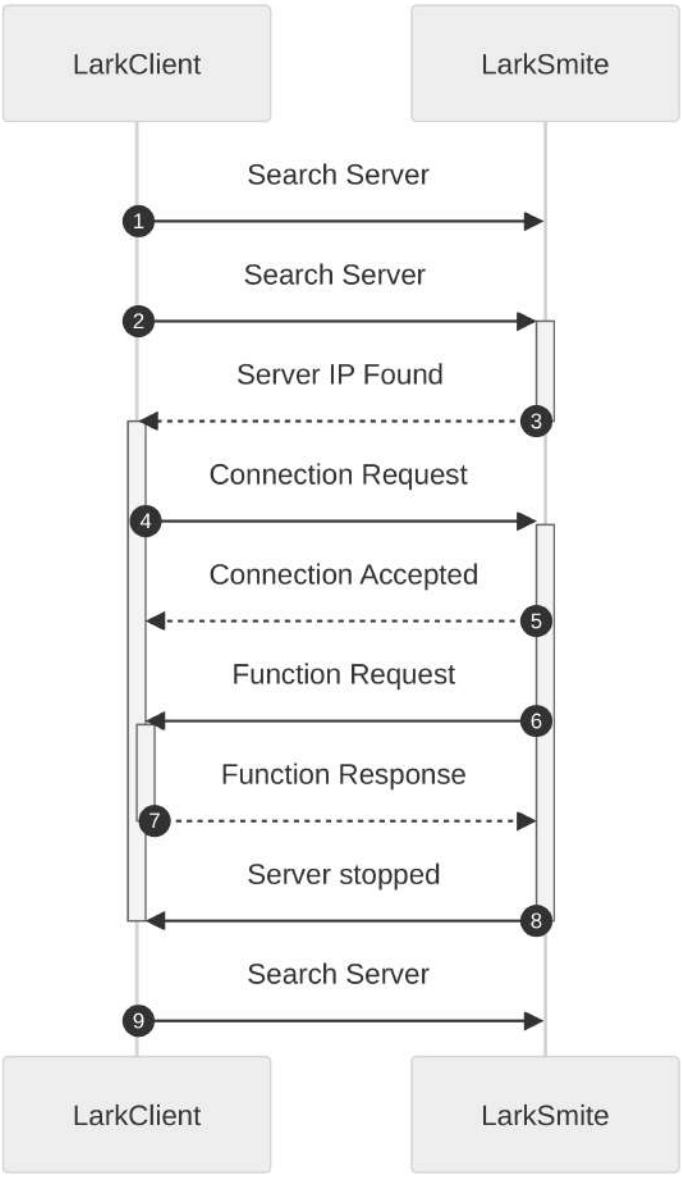
A thin rectangle on a lifeline represents the period during which an element is performing an operation. The top and the bottom of the rectangle is aligned with the initiation and the completion time respectively.

B.6 Sequence Diagram

B.6.1 Server Side

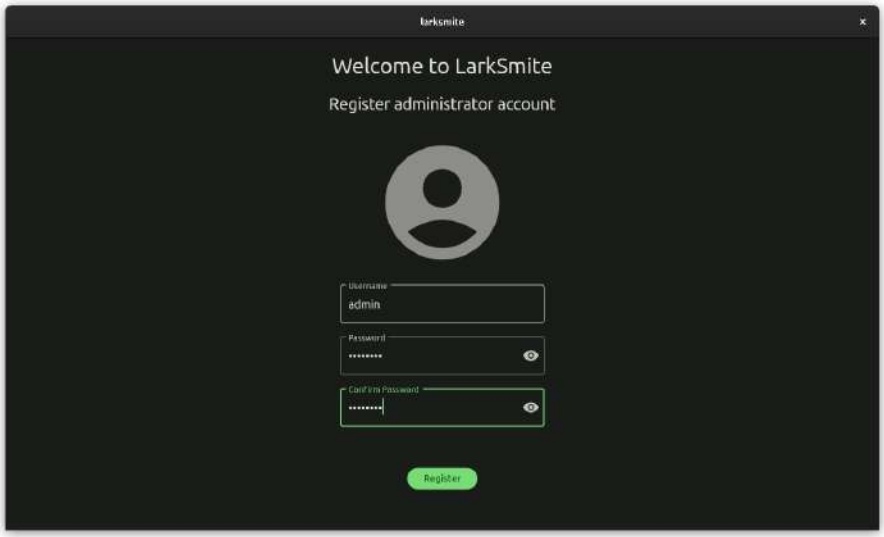


B.6.2 Client Side

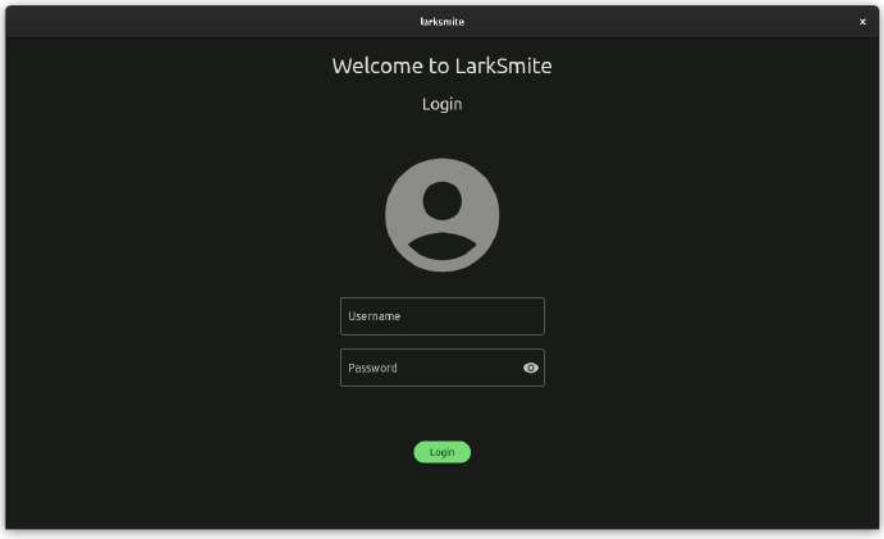


C User Interfaces

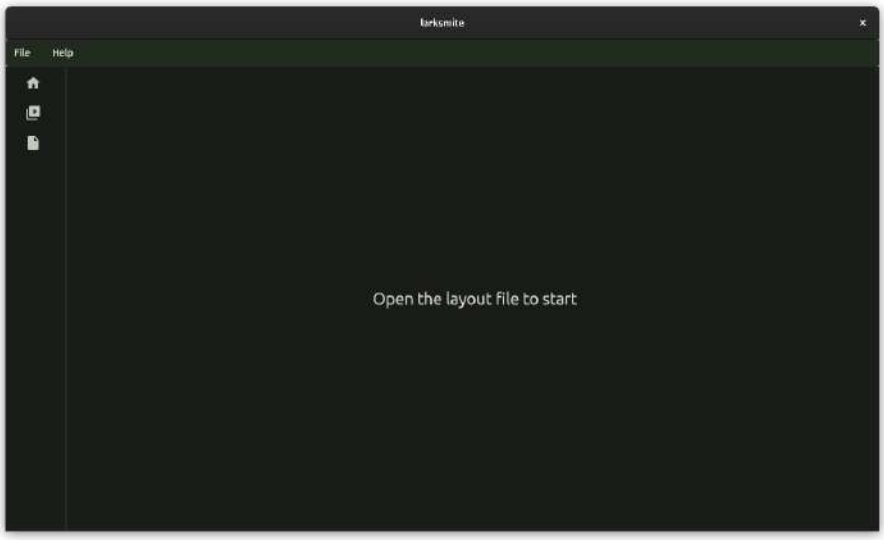
C.1 Admin Registration



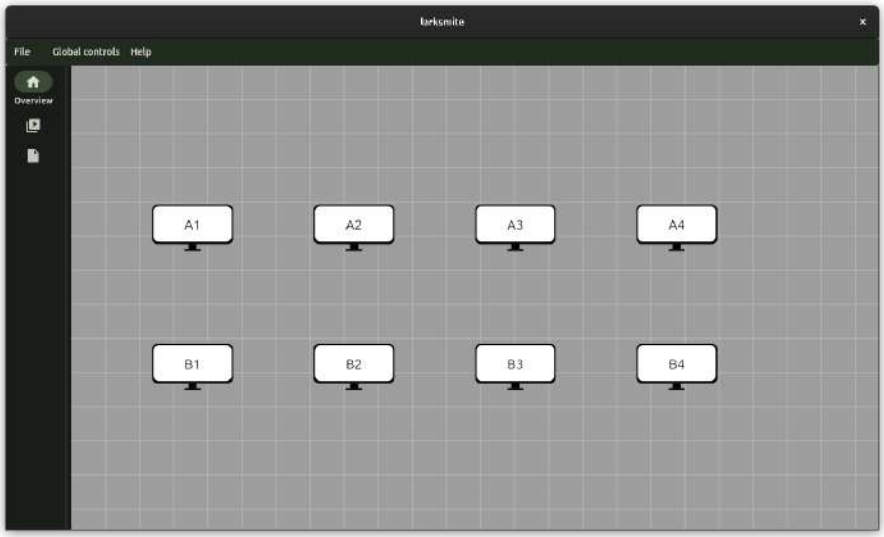
C.2 Admin Login



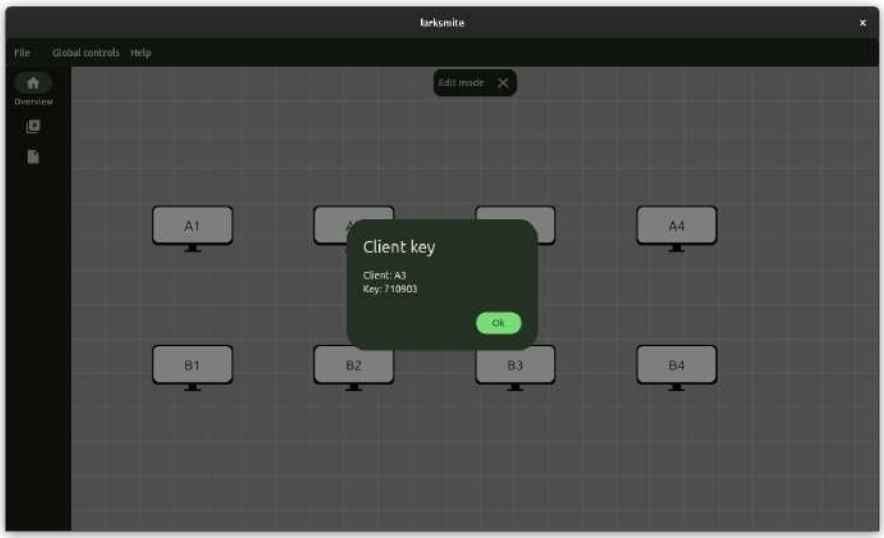
C.3 Home Page



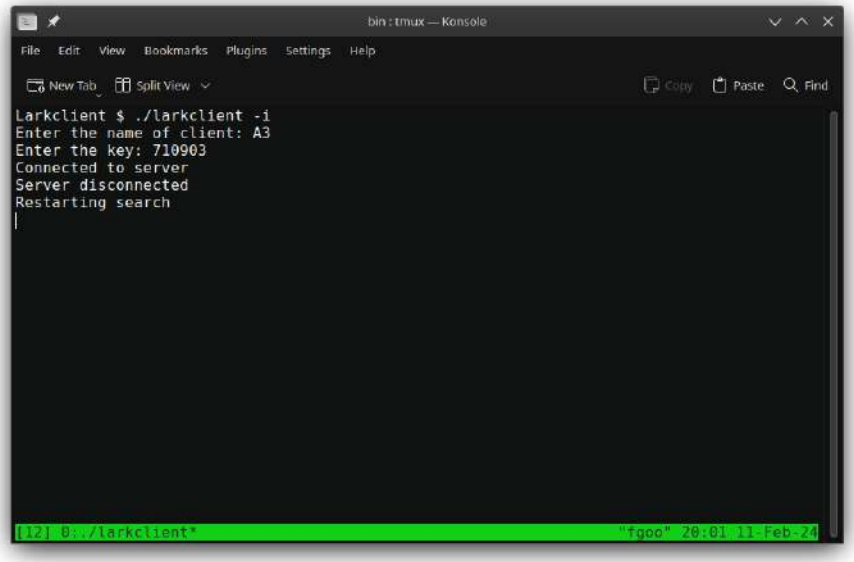
C.4 Layout File Opened



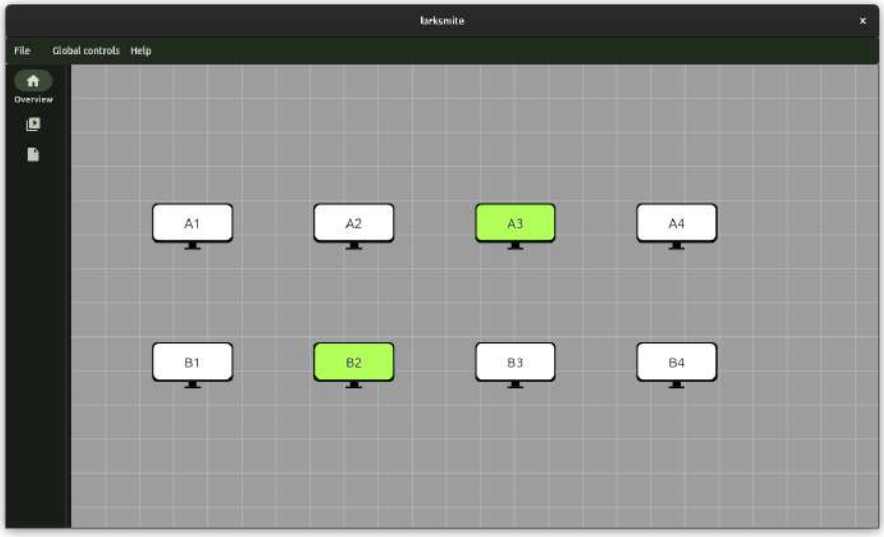
C.5 Adding Clients



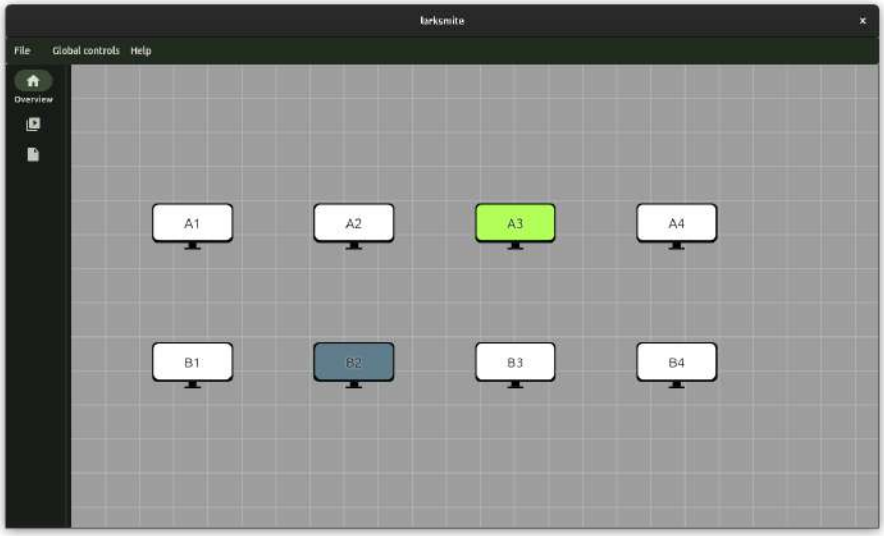
C.6 Settings up Larkclient



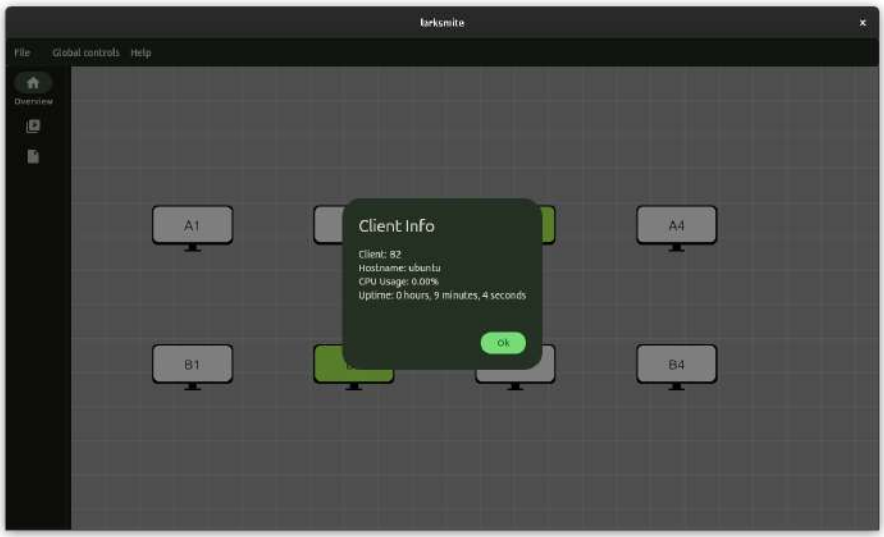
C.7 Client Realtime Status



C.8 Color coded status

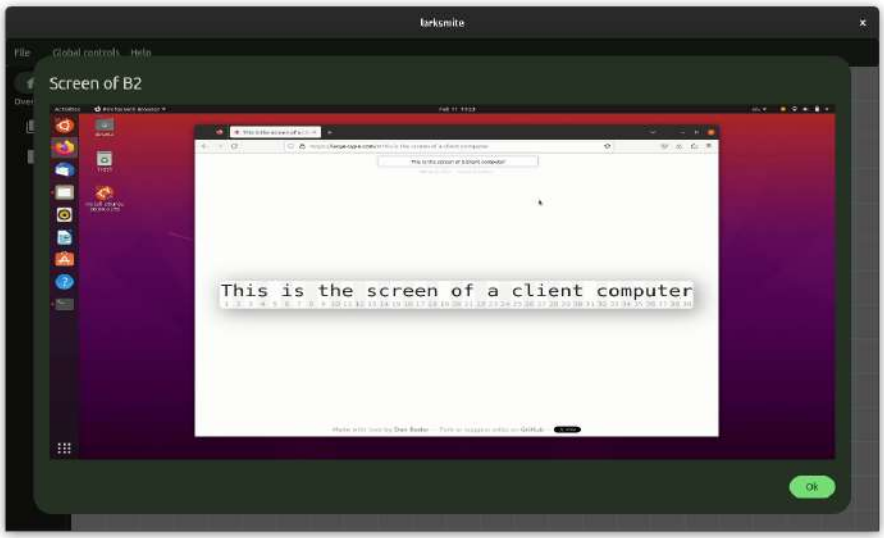


C.9 Client Info

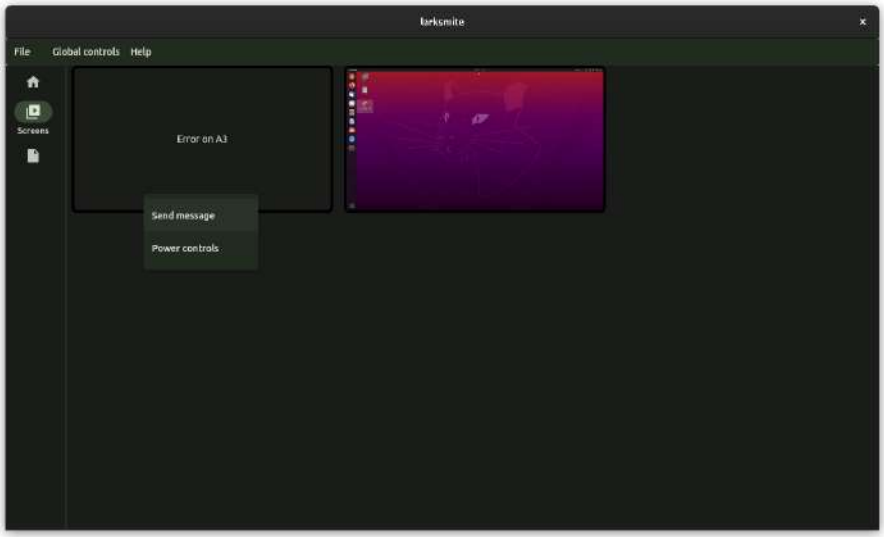


C.10 Client Processes

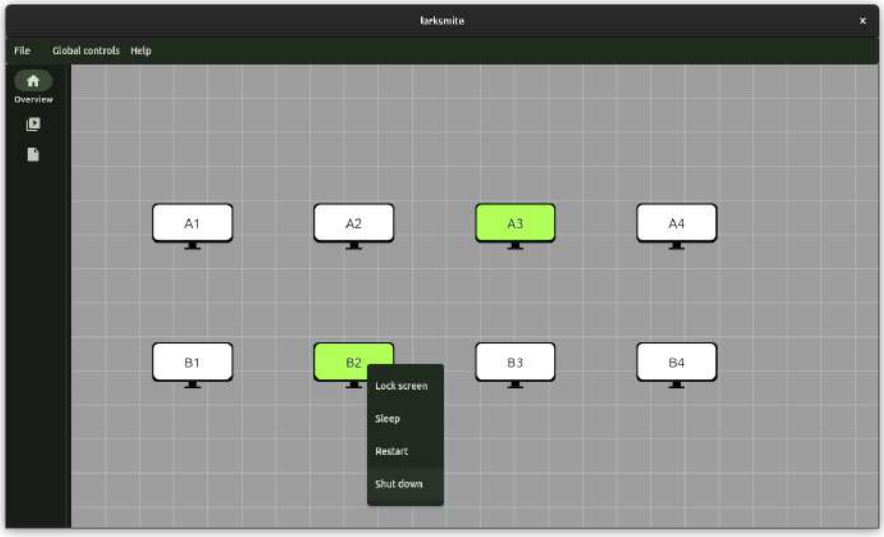
C.11 Single Screen Stream



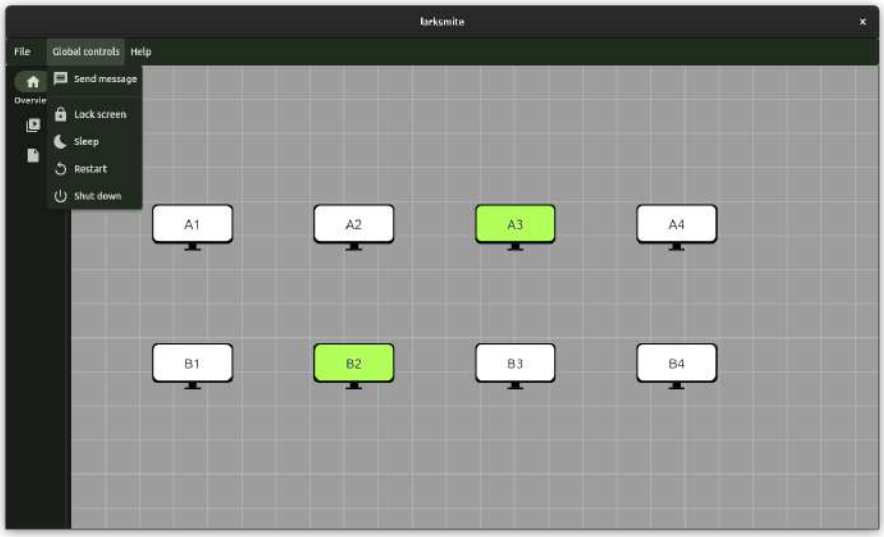
C.12 All Screen Stream



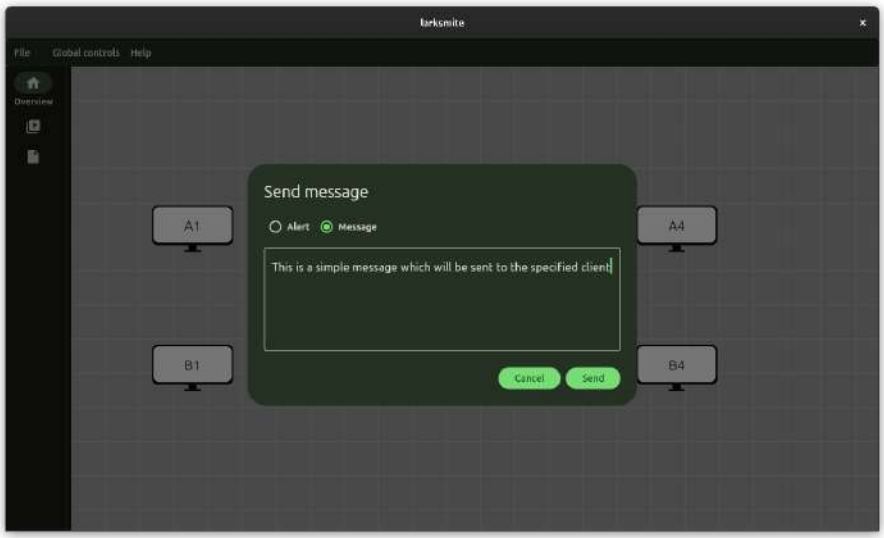
C.13 Power Control



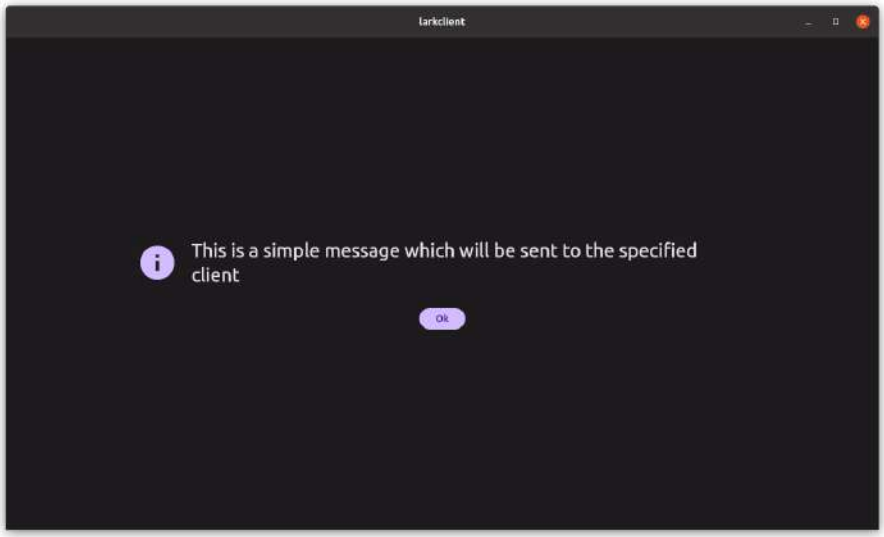
C.14 Global Control



C.15 Sending Message



C.16 Message on Client Side



D Code

main.dart

```
import 'package:flutter_bloc/flutter_bloc.dart';
import 'package:flutter/material.dart';
// import 'package:yaru/yaru.dart';
import 'core/injection_container.dart' as core_di;
import 'features/login/login_injection_container.dart' as login_
    di;
import 'features/dashboard/dashboard_injection_container.dart' as
     dashboard_di;
import 'features/networking/networking_injection_container.dart'
    as networking_di;
import 'features/datastore/datastore_injection_container.dart' as
     datastore_di;
import 'core/cubit/core_data_cubit.dart';
import 'core/router.dart';
import 'core/logging.dart' as logging;

void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await logging.initLogger();
  await core_di.init();
  await datastore_di.init();
  await login_di.init();
  await networking_di.init();
  await dashboard_di.init();
  runApp(const LarkSmiteApp());
}

class LarkSmiteApp extends StatelessWidget {
  const LarkSmiteApp({super.key});
  @override
  Widget build(BuildContext context) {
    return BlocProvider(
      create: (context) => CoreDataCubit(firstStartChecker: core_
          di.sl()),
      child: MaterialApp.router(
        title: 'LarkSmite',
        // theme: yaruLight,
        // darkTheme: yaruDark,
        //
        debugShowCheckedModeBanner: false,
        routerConfig: LarkSmiteRouter.router,
        darkTheme: ThemeData(
```

```
          brightness: Brightness.dark,
          // colorScheme: ColorScheme.fromSeed(seedColor: Colors.
              amber),
          colorSchemeSeed: Colors.green,
          useMaterial3: true,
        ),
      ),
    );
  }
}
```

dashboard.dart

```
import 'package:flutter/material.dart';
import 'package:larksmite/features/dashboard/presentation/cubit/
    app_menu_bar_cubit.dart';
import 'package:larksmite/features/dashboard/presentation/cubit/
    report_view_cubit.dart';
import 'package:larksmite/features/dashboard/presentation/cubit/
    screen_view_cubit.dart';
import 'dashboard_view_frame.dart';
import '../widget/navigation_rail_side.dart';
import '../cubit/birds_eye_view_cubit.dart';
import '../cubit/dashboard_cubit.dart';
import 'package:flutter_bloc/flutter_bloc.dart';
import '../../dashboard_injection_container.dart' as dashboard_di
    ;
import 'package:larksmite/features/networking/networking_
    injection_container.dart'
    as networking_di;
import 'package:larksmite/features/datastore/datastore_injection_
    container.dart'
    as datastore_di;
import 'package:file_picker/file_picker.dart';
import '../widget/app_menu_bar.dart';

class DashboardFrame extends StatefulWidget {
  const DashboardFrame({super.key});

  @override
  State<DashboardFrame> createState() => _DashboardFrameState();
}

class _DashboardFrameState extends State<DashboardFrame> {
  @override
  Widget build(BuildContext context) {
    return MultiBlocProvider(
```

```
providers: [
  BlocProvider(
    create: (context) => DashboardCubit(
      loadLabLayout: dashboard_di.sl(),
      serviceAdvertising: networking_di.sl(),
      manageClientConnections: networking_di.sl(),
    ),
  ),
  BlocProvider(
    create: (context) => BirdsEyeViewCubit(
      manageClientConnections: networking_di.sl(),
      larksmiteDatastore: datastore_di.sl(),
    ),
  ),
  BlocProvider(
    create: (context) => ScreenViewCubit(
      manageClientConnections: networking_di.sl(),
    ),
  ),
  BlocProvider(
    create: (context) => ReportViewCubit(
      manageClientConnections: networking_di.sl(),
    ),
  ),
  BlocProvider(
    create: (context) => AppMenuBarCubit(),
  ),
],
child: BlocListener<DashboardCubit, DashboardState>(
  listener: (context, state) async {
    switch (state) {
      case DashboardShowFilePicker():
        FilePickerResult? filePickerResult =
            await FilePicker.platform.pickFiles(
          initialDirectory: ".",
          dialogTitle: "Open layout file",
          type: FileType.custom,
          allowedExtensions: ["json"],
        );
        if (filePickerResult case FilePickerResult(:final
            files)
            when files.isNotEmpty && files.single.path !=
                null) {
          if (mounted) {
            context.read<DashboardCubit>().loadFromFile(
                  filePath: files.single.path!,
```

```
                );
              }
            } else if (mounted) {
              context.read<DashboardCubit>().layoutNotLoaded();
            }
          case DashboardLayoutLoaded(:final labLayout):
            context.read<AppMenuBarCubit>().layoutLoaded();
            context
                .read<BirdsEyeViewCubit>()
                .gotLabLayout(labLayout: labLayout);
            context
                .read<ScreenViewCubit>()
                .gotLabLayout(labLayout: labLayout);
            context
                .read<ReportViewCubit>()
                .gotLabLayout(labLayout: labLayout);
          default:
            context.read<AppMenuBarCubit>().layoutNotLoaded();
            context.read<BirdsEyeViewCubit>().noLabLayout();
            context.read<ScreenViewCubit>().noLabLayout();
            context.read<ReportViewCubit>().noLabLayout();
            break;
        }
      },
      child: const AppMenuBar(
        body: Scaffold(
          body: Row(
            children: [
              NavigationRailSide(),
              VerticalDivider(),
              DashboardViewFrame()
            ],
          ),
        ),
      ),
    ),
  );
 }
}
```

larkclient_coordinator.dart

```
import 'dart:async';
import 'dart:io';
import 'dart:typed_data';
import 'package:async/async.dart';
import 'package:fpdart/fpdart.dart';
```

```
import 'package:larkclient/core/entity/client_name_and_key.dart';
import 'package:larkclient/core/entity/larksmite_server.dart';
import 'package:larkclient/core/logging.dart';
import 'package:larkclient/core/status_types.dart';
import 'package:larkclient/feature/datastore/domain/usecase/
    server_and_client_details.dart';
import 'package:larkclient/feature/networking/domain/usecase/
    manage_server_conn.dart';
import 'package:larkclient/feature/platform_event_handler/domain/
    repository/platform_event_handler.dart';
import 'package:larksmite_shared/larskmite_shared_entites.dart';

class LarkClientCoordinator {
  final ServerAndClientDetails serverAndClientDetails;
  final ManageServerConnection manageServerConnection;
  HttpClient? _httpClient;
  LarkClientCoordinator({
    required this.serverAndClientDetails,
    required this.manageServerConnection,
    PlatformEventHandler? platformEventHandler,
  }) : _platformEventHandlerSupplied = platformEventHandler;
  final PlatformEventHandler? _platformEventHandlerSupplied;
  PlatformEventHandler get platformEventHandler {
    if (_platformEventHandlerSupplied != null) {
      return _platformEventHandlerSupplied!;
    } else {
      throw UnimplementedError(
        "PlatformEventHandler is not implemented for this platform
            ",
      );
    }
  }

  Stream<Either<Failure, Success>> startEverything({
    ClientNameAndKey? clientNameAndKey,
  }) async* {
    try {
      platformEventHandler;
    } on UnimplementedError catch (error, stacktrace) {
      getLogger().severe(
        "Event handler isn't implemented for this platform",
        error,
        stacktrace,
      );
      yield Either.left(
        NoPlatformEventHandlerAvailable(
```

```
      message: "This platform is not supported",
    ),
  );
  return;
} on Exception catch (error, stacktrace) {
  getLogger().severe(
    "Unexpected error while trying to access
        PlatformEventHandler",
    error,
    stacktrace,
  );
  yield Either.left(
    UnknownUnrecoverableFailure(
      message:
          "Unexpected error while checking if this platform is
              supported",
    ),
  );
  return;
}
final handlerStream = handlePlatformSpecificEvents();

// If the passed clientNameAndKey is null, it (hopefully)
    means its not the first time
// But if it's not null, then it means its the first time
ClientNameAndKey clientDetails;
bool isFirstTime;
if (clientNameAndKey != null) {
  clientDetails = clientNameAndKey;
  isFirstTime = true;
} else {
  final clientNameAndKeyResult =
      await serverAndClientDetails.getSavedClientNameAndKey();
  switch (clientNameAndKeyResult) {
    case Right(:final value):
      clientDetails = value;
      isFirstTime = false;
    case Left(:final value):
      getLogger().severe(
        "Failure while trying to retrieve saved client details
            ",
      );
      yield Either.left(value);
      return;
  }
}
```

```
  final connectionResult = connectToServerSecurely(
    clientNameAndKey: clientDetails,
    isFirstTime: isFirstTime,
  );
  yield* StreamGroup.merge([handlerStream, connectionResult]);
  getLogger().config("All streams finished in startEverything");
  return;
}

/// If [retryTimes] is 0, it will retry until we get successful
     result
/// The result of calling [shouldStopImmediately] with any
    encountered failure
/// is used to determine if we should stop immediately, useful
    in cases where
/// [retryTimes] is inifnite but it should stop when a certain
    condition is met
Future<Either<Failure, T>> _retryTask<T>({
  required Future<Either<Failure, T>> Function() task,
  int retryTimes = 0,
  Duration delayBetweenRetries = const Duration(seconds: 3),
  bool Function(Failure failure)? shouldStopImmediately,
}) async {
  bool retryCondition(int iInside, int retryTimesInside) {
    return (iInside < retryTimesInside || retryTimesInside == 0)
        ;
  }

  Failure? lastFailure;
  shouldStopImmediately ??= (_) => false;
  for (int i = 0; retryCondition(i, retryTimes); i++) {
    final result = await task();
    switch (result) {
      case Right(:final value):
        return Either.right(value);
      case Left(:final value):
        if (shouldStopImmediately(value)) {
          return Either.left(value);
        } else {
          lastFailure = value;
        }
    }
    await Future.delayed(delayBetweenRetries);
  }
  lastFailure ??=
```

```
      NoMoreRetries(message: "No more retries available for task
          ");
  return Either.left(lastFailure);
}

/// This will yield succesful event, if it encounters any
    failure, it will yield
/// that Failure and close immediately, its up to the caller to
     handle it.
/// This version of connection method is more try-and-fail type
     than
/// ask-for-permission type, thus simplifying stuff massively
Stream<Either<Failure, Success>> connectToServerSecurely({
  required ClientNameAndKey clientNameAndKey,
  bool isFirstTime = false,
}) async* {
  getLogger().config(
      "This is${isFirstTime ? '' : ' not'} the first time we're
          starting");
  // Search until we get the server
  LarkSmiteServer searchedServer;

  final searchedServerResult = await _retryTask(
    task: () => manageServerConnection.searchAndGetServer(),
    delayBetweenRetries: const Duration(seconds: 1),
    shouldStopImmediately: (failure) => failure is
        ServerSearchCancelled,
  );
  switch (searchedServerResult) {
    case Right(:final value):
      searchedServer = value;
    case Left(:final value):
      // Since retryTimes is 0 for server searching task, it
          should be able to
      // get the server eventually, but if it's unable to do so,
           it must be
      // the cancelled scenario
      getLogger().severe("Searching server failed", value);
      yield Either.left(value);
      return;
  }
  if (!isFirstTime) {
    getLogger()
        .config("It's not first start, so checking server by
            challenge");
    final isServerTrustedResult = await serverDoChallenge(
```

```
    larkSmiteServer: searchedServer,
  );
  bool isServerTrusted;
  switch (isServerTrustedResult) {
    case Right(:final value):
      isServerTrusted = value;
    case Left(:final value):
      yield Either.left(value);
      return;
  }
  if (!isServerTrusted) {
    yield Either.left(
      ServerChallengeFailed(
        message:
            "${searchedServer.ipAddress} gave unsuccessful
                response for challenge, not trusting it.",
      ),
    );
    return;
  } else {
    getLogger().config("Server completed challenge successfuly
        ");
  }
}

// If we're here, it means the server is trustworthy, or its
    the first time
// which case, we assume its trusted
Uint8List certBinary;

final certBinaryResult = await _retryTask(
  task: () => manageServerConnection.downloadCertificate(
    larkSmiteServer: searchedServer,
  ),
  retryTimes: 5,
);
switch (certBinaryResult) {
  case Right(:final value):
    certBinary = value;
  case Left(:final value):
    getLogger().severe("Unable to download certificate", value
        );
    yield Either.left(value);
    return;
}
final securityContext = SecurityContext();
```

```
securityContext.setTrustedCertificatesBytes(certBinary);
final httpClient = HttpClient(context: securityContext);
_httpClient = httpClient;
final connectionStream = manageServerConnection.
    connectToServer(
  clientNameAndKey: clientNameAndKey,
  httpClient: httpClient,
  ipAddress: searchedServer.ipAddress,
  isFirstStartFlag: isFirstTime,
);
yield* connectionStream.transform(
  StreamTransformer.fromHandlers(
    handleData: (data, sink) async {
      switch (data) {
        case Right(value: final connection):
          if (isFirstTime) {
            getLogger().config(
              "Since it's the first time, "
              "saving public key of server and name + key of
                  client",
            );
            // We don't want any stale data
            await serverAndClientDetails.clearAllData();
            await serverAndClientDetails.saveClientNameAndKey(
              clientNameAndKey: clientNameAndKey,
            );
            await serverAndClientDetails.savePublicKey(
              rsaPublicKey: connection.rsaPublicKeyAbstract,
            );
          }
          sink.add(
            Either.right(
              ConnectServerSuccess(message: "Connected to
                  server"),
            ),
          );
        case Left(:final value):
          sink.add(Either.left(value));
      }
    },
    handleDone: (sink) {
      sink.close();
    },
    handleError: (error, stackTrace, sink) {
      getLogger().severe(
        "Unknown error while transforming in
```

```
            connectToServerSecurely",
          error,
          stackTrace,
        );
        sink.close();
      },
    ),
  );
  return;
}

/// This will handle all events from server. Call this before
    connecting so as to
/// not miss any events
Stream<Either<Failure, Success>> handlePlatformSpecificEvents()
    {
  // Returning as broadcast stream since cancelling this will
      interfere with
  // the cleanup process, but if its a broadcast stream, it will
       continue
  // cleaning up and finish naturally
  return platformEventHandler
      .handleEvents(
        serverCommunicatorDuplex:
            manageServerConnection.serverCommunicatorDuplex,
      )
      .asBroadcastStream();
}

Future<Either<Failure, bool>> serverDoChallenge({
  required LarkSmiteServer larkSmiteServer,
}) async {
  final savedPublicKeyResult =
      await serverAndClientDetails.getPreviouslySavedPublicKey()
          ;
  RSAPublicKeyAbstract rsaPublicKey;
  switch (savedPublicKeyResult) {
    case Right(:final value):
      rsaPublicKey = value;
    case Left(:final value):
      return Either.left(value);
  }
  return manageServerConnection.performChallenge(
    larkSmiteServer: larkSmiteServer,
    rsaPublicKey: rsaPublicKey,
  );
```

```
  }

  Future<void> stopEverything() async {
    await manageServerConnection.stopSearching();
    await manageServerConnection.disconnectFromServer();
    _httpClient?.close(force: true);
    await platformEventHandler.stopHandling();
  }
}
```

manage_server_conn.dart

```
import 'dart:async';
import 'dart:io';
import 'dart:typed_data';
import 'package:fpdart/fpdart.dart';
import 'package:larkclient/core/entity/client_name_and_key.dart';
import 'package:larkclient/core/entity/events_from_server/events_
    from_server.dart';
import 'package:larkclient/core/entity/events_to_server/events_to
    _server.dart';
import 'package:larkclient/core/entity/larksmite_server.dart';
import 'package:larkclient/core/entity/server_communicator_duplex
    .dart';
import 'package:larkclient/core/logging.dart';
import 'package:larkclient/core/status_types.dart';
import 'package:larkclient/feature/networking/domain/repository/
    server_repository.dart';
import 'package:larksmite_shared/larksmite_shared.dart';
import 'package:larksmite_shared/larskmite_shared_entites.dart';
import 'package:uuid/v4.dart';

class ManageServerConnection {
  final ServerRepository serverRepository;
  ServerCommunicatorDuplex get serverCommunicatorDuplex =>
      serverRepository.serverCommunicatorDuplex;
  ManageServerConnection({required this.serverRepository});
  Future<Either<Failure, LarkSmiteServer>> searchAndGetServer()
      async {
    return serverRepository.searchForServer();
  }

  Future<void> stopSearching() async {
    await serverRepository.stopSearchingForServer();
  }

  Future<Either<Failure, RecieveType>> _sendAndWaitForEvent<
```

```
    SendType extends EventsToSendToServer,
    RecieveType extends EventsRecievedFromServer>({
  required SendType sendEvent,
  Duration timeoutDuration = const Duration(seconds: 10),
}) async {
  final result = await sendAndWaitForEvent<SendType, RecieveType
      ,
      ServerErrorResponse, EventsToSendToServer,
          EventsRecievedFromServer>(
    sendEvent: sendEvent,
    sendSink: serverCommunicatorDuplex.eventsToSendToServer,
    recieveStream: serverCommunicatorDuplex.
        eventsRecievedFromServer,
    timeoutDuration: timeoutDuration,
  );
  return result;
}

Future<Either<Failure, ConnectionResponse>> _getConnectResult({
  required ClientNameAndKey clientNameAndKey,
  required bool isFirstStartFlag,
}) async {
  return _sendAndWaitForEvent<ConnectionRequest,
      ConnectionResponse>(
    sendEvent: ConnectionRequest.generate(
      clientNameAndKey: clientNameAndKey,
      isFirstTime: isFirstStartFlag,
    ),
    timeoutDuration: const Duration(seconds: 15),
  );
}

Stream<Either<Failure, ConnectionAccepted>> connectToServer({
  required ClientNameAndKey clientNameAndKey,
  required HttpClient httpClient,
  required InternetAddress ipAddress,
  required bool isFirstStartFlag,
}) {
  final connectToServerStream =
      serverRepository.connectToServerAndListenToEvents(
    httpClient: httpClient,
    ipAddress: ipAddress,
  );
  return connectToServerStream.transform(
    StreamTransformer.fromHandlers(
      handleData: (data, sink) async {
```

```
switch (data) {
  case Right():
    final connectionResult = await _getConnectResult(
      clientNameAndKey: clientNameAndKey,
      isFirstStartFlag: isFirstStartFlag,
    );
    switch (connectionResult) {
      case Right(value: ConnectionAccepted()):
        sink.add(Either.right(
            connectionResult.value as ConnectionAccepted)
                );
      case Right(value: ConnectionDeclined()):
        // Its Right here since, the connection was
            successful and the event was
        // recieved correctly
        sink.add(
          Either.left(
            ClientDetailsAreWrong(
              message:
                  "Server declined connection because
                      client details are wrong",
            ),
          ),
        );
      // Changing the types of [ErrorResponseFailure] and
          [TimeoutFailure]
      // from larksmite_shared to our own failure types
          which extends
      // [RecoverableFailure]
      case Left(
          value: ErrorResponseFailure(
            :final message,
            errorResponse: ServerErrorResponse(),
          )
        ):
        sink.add(Either.left(ServerErrorFailure(message:
            message)));
      case Left(value: TimeoutFailure(:final message)):
        sink.add(Either.left(ServerTimeoutFailure(message
            : message)));
      case Left(:final value):
        sink.add(Either.left(value));
    }
  case Left(:final value):
    sink.add(Either.left(value));
}
```

```
      },
      handleDone: (sink) {
        sink.close();
      },
      handleError: (error, stackTrace, sink) {
        getLogger().severe(
          "Error while transforming events in connectToServer",
          error,
          stackTrace,
        );
        sink.close();
      },
    ),
  );
}

Future<Either<Failure, Uint8List>> downloadCertificate({
  required LarkSmiteServer larkSmiteServer,
}) async {
  return serverRepository.downloadCertificate(
      larkSmiteServer: larkSmiteServer);
}

Future<Either<Failure, bool>> performChallenge({
  required LarkSmiteServer larkSmiteServer,
  required RSAPublicKeyAbstract rsaPublicKey,
}) async {
  final randomString = const Uuid V4().generate();
  final encryptedData = await LarksmiteCryptoRsaPointyCastle().
      encryptString(
    rsaPublicKey: rsaPublicKey as RSAPublicKeyPointyCastle,
    string: randomString,
  );

  final challengeResponse = await serverRepository.
      performChallenge(
    larkSmiteServer: larkSmiteServer,
    serverDoChallengeDecrypt: ServerDoChallengeDecrypt(
      encryptedData: encryptedData,
    ),
  );
  switch (challengeResponse) {
    case Left(value: ServerChallengeFailed()):
      return Either.right(false);
    case Left(:final value):
      return Either.left(value);
```

```
      case Right(:final value):
        // Simple equality check, the real check is if the server
            was able to perform
        // the decryption and send the plain text back
        return Either.right(
          value.decryptedString == randomString,
        );
    }
  }

  Future<Either<Failure, Success>> disconnectFromServer() async {
    return serverRepository.disconnectFromServer();
  }
}
```

linux_event_handler.dart

```
import 'dart:async';
import 'dart:io';
import 'package:fpdart/fpdart.dart';
import 'package:gnome_screenrecord/gnome_screenrecord.dart';
import 'package:larkclient/core/constants.dart';
import 'package:larkclient/core/entity/events_from_server/events_
    from_server.dart';
import 'package:larkclient/core/entity/events_to_server/events_to
    _server.dart';
import 'package:larkclient/core/entity/server_communicator_duplex
    .dart';
import 'package:larkclient/core/logging.dart';
import 'package:larkclient/core/status_types.dart';
import 'package:larkclient/core/useful_helpers.dart';
import 'package:larkclient/feature/platform_event_handler/data/
    datasource/linux_proc_source.dart';
import 'package:larkclient/feature/platform_event_handler/data/
    datasource/linux_run_bash_source.dart';
import 'package:larkclient/feature/platform_event_handler/domain/
    entity/screencast_object.dart';
import 'package:larkclient/feature/platform_event_handler/domain/
    repository/platform_event_handler.dart';
import 'package:larksmite_shared/larskmite_shared_entites.dart';
import 'package:process_monitor/process_monitor.dart';

class LinuxPlatformEventHandler implements PlatformEventHandler {
  final LinuxProcSource linuxProcSource;
  final LinuxRunBashSource linuxRunBashSource;
  ScreencastObj? _screencastObj;
  StreamController<Either<Failure, Success>>? _eventStatusStream;
```

```
LinuxPlatformEventHandler({
  required this.linuxProcSource,
  required this.linuxRunBashSource,
});
Future<ProcessModel> getRootProcess() {
  getLogger().config("Trying to get processes");
  final processMonitor = ProcessMonitor.instance;
  return processMonitor.getProcessTreeRootNode();
}

Future<ClientInfo> getClientInfo() async {
  getLogger().config("Trying to get client info");
  return ClientInfo(
    hostname: Platform.localHostname,
    uptime: await linuxProcSource.getUptime(),
    cpuUsage: await linuxProcSource.getCpuUsage(),
  );
}

Future<void> showMessageGui({
  required String message,
  required MessageType messageType,
}) async {
  getLogger().config("Showing GUI message to user");
  await linuxRunBashSource.showMessageGui(
    message: message,
    messageType: messageType,
  );
}

Future<Uri> getScreencastLink({
  required ScreenCastResolution screenCastResolution,
  required InternetAddress clientIp,
}) async {
  //TODO P3: find a way to make this less kludgy
  getLogger().config("Beginning screencast procedure");
  if (_screencastObj == null) {
    getLogger().config("Getting screen resolution");
    final gnomeRecorder = GnomeScreenRecorder();
    final res = await linuxRunBashSource.getScreenResolution();
    final int width, height;
    width = ((screenCastResolution.resolutionPercentage / 100) *
         res.width)
        .toInt();
    height = ((screenCastResolution.resolutionPercentage / 100)
```

```
    * res.height)
    .toInt();
Stream<DbusRecordStatus> screencastStatusStream;
getLogger().config("About to run gnomeScreenRecord");

screencastStatusStream = gnomeRecorder.startRecording(
  height: height,
  width: width,
  internetAddress: clientIp,
  port: screencastPort,
);

final streamLinkToSend = Completer<Uri>();
final screencastListener = screencastStatusStream.listen(
  // Fortunately we don't need to cancel this stream since
      it finishes when
  // the recording stops, but we should probably find a way
      to make it cleaner
  (event) {
    getLogger().config("Got event $event from gnome screen
        record");
    switch (event) {
      case DbusRecordStarted(
          :final streamLink,
          :final dbusMethodResponse,
        ):
        getLogger().config(
          "Dbus start screen record response: $
              dbusMethodResponse",
        );
        _screencastObj = ScreencastObj(
          streamLinkToSend: streamLink,
          screenRecorder: gnomeRecorder,
        );
        streamLinkToSend.complete(streamLink);
      case DbusRecordWaiting():
        getLogger().config("Screencasting... Waiting for stop
            signal");
      case DbusRecordEnded(:final dbusMethodResponse):
        getLogger().config(
          "Dbus stop screen record response: $
              dbusMethodResponse",
        );
      case DbusError(:final dBusMethodResponseException):
        getLogger().warning(
            "Unable to start recording, response: $
```

```
                dBusMethodResponseException");
            streamLinkToSend.completeError(ScreencastException())
                ;
        }
      },
      cancelOnError: true,
    );
    await streamLinkToSend.future;
    // await screencastListen.cancel();
  } else {
    getLogger()
        .config("Skipping some steps since screencast is already
            running");
  }

  final screencastLink = Uri.parse(
    "tcp://${clientIp.address}:$screencastPort",
  );

  getLogger().config("Sending screen cast link: $screencastLink
      ");
  return screencastLink;
}

Future<void> stopScreencast() async {
  getLogger().config("Stopping screencast");
  if (_screencastObj != null) {
    await _screencastObj!.screenRecorder.stopRecording();
    _screencastObj = null;
  }
  getLogger().config("Stopped screencast");
}

Future<void> handlePowerSignal({
  required PowerSignal powerSignal,
}) async {
  await Future.delayed(powerSignal.performSignalAfter);
  switch (powerSignal.powerSignalType) {
    case PowerSignalType.lockScreen:
      await linuxRunBashSource.lockScreen();
    case PowerSignalType.sleep:
      await linuxRunBashSource.sleepComputer();
    case PowerSignalType.restart:
      await linuxRunBashSource.restartComputer();
    case PowerSignalType.powerOff:
      await linuxRunBashSource.shutdownComputer();
```

```
  }
}

Future<List<int>> getScreenshot() {
  return linuxRunBashSource.getScreenshot();
}

@override
Stream<Either<Failure, Success>> handleEvents({
  required ServerCommunicatorDuplex serverCommunicatorDuplex,
}) async* {
  getLogger().config("Starting the Linux event handler");
  _eventStatusStream = StreamController();

  final serverEventListener =
      serverCommunicatorDuplex.eventsRecievedFromServer.listen(
    (event) async {
      switch (event) {
        case ClientProcessesRequired(:final token):
          try {
            final rootProcess = await getRootProcess();
            serverCommunicatorDuplex.eventsToSendToServer.add(
              ClientProcessesResponse.generateResponseTo(
                responseToToken: token,
                rootProcess: rootProcess,
              ),
            );
            _eventStatusStream?.add(
              Either.right(
                EventHandledSuccessfuly(
                  message: "Handled get process request
                      successfuly",
                  eventToken: token,
                ),
              ),
            );
          } on Exception catch (error, stacktrace) {
            getLogger().warning(
                "Exception while handling ClientProcessRequired",
                error,
                stacktrace);
            serverCommunicatorDuplex.eventsToSendToServer.add(
              ClientErrorResponse.generateResponseTo(
                responseToToken: token,
                message: "Unable to get processes",
                errorString: error.toString(),
```

```
          stacktraceString: stacktrace.toString(),
        ),
      );
      _eventStatusStream?.add(
        Either.left(
          EventHandleFailed(
            message: "Handling of GetProcesses failed",
            eventToken: token,
          ),
        ),
      );
    }
  case ClientInfoRequired(:final token):
    try {
      final clientInfo = await getClientInfo();
      serverCommunicatorDuplex.eventsToSendToServer.add(
        ClientInfoResponse.generateResponseTo(
          responseToToken: token,
          clientInfo: clientInfo,
        ),
      );
      _eventStatusStream?.add(
        Either.right(
          EventHandledSuccessfuly(
            message: "Handled get client info successfuly",
            eventToken: token,
          ),
        ),
      );
    } on Exception catch (error, stacktrace) {
      getLogger().warning("Exception while handling
          ClientInfoRequired",
          error, stacktrace);
      serverCommunicatorDuplex.eventsToSendToServer.add(
        ClientErrorResponse.generateResponseTo(
          responseToToken: token,
          message: "Unable to get info",
          errorString: error.toString(),
          stacktraceString: stacktrace.toString(),
        ),
      );
      _eventStatusStream?.add(
        Either.left(
          EventHandleFailed(
            message: "Handling of GetInfo failed",
            eventToken: token,
```

```
        ),
      ),
    );
  }
case ClientStatusRequired(:final token):
  serverCommunicatorDuplex.eventsToSendToServer.add(
    ClientStatusResponse.generateResponseTo(
      responseToToken: token,
      clientStatus: ClientStatus.ok,
    ),
  );
  _eventStatusStream?.add(
    Either.right(
      EventHandledSuccessfuly(
        message: "Handled get client status successfuly",
        eventToken: token,
      ),
    ),
  );
case PowerSignalRecieved(:final token, :final
    powerSignal):
  try {
    handlePowerSignal(powerSignal: powerSignal);
    serverCommunicatorDuplex.eventsToSendToServer.add(
      ClientAcknowledgeResponse.generateResponseTo(
        responseToToken: token,
      ),
    );
    _eventStatusStream?.add(
      Either.right(
        EventHandledSuccessfuly(
          message: "Handled power signal successfuly",
          eventToken: token,
        ),
      ),
    );
  } on Exception catch (error, stacktrace) {
    getLogger().warning(
        "Exception while handling PowerSignalRecieved",
        error,
        stacktrace);
    serverCommunicatorDuplex.eventsToSendToServer.add(
      ClientErrorResponse.generateResponseTo(
        responseToToken: token,
        message: "Unable to handle power signal",
        errorString: error.toString(),
```

```
          stacktraceString: stacktrace.toString(),
        ),
      );
      _eventStatusStream?.add(
        Either.left(
          EventHandleFailed(
            message: "Handling of power signal failed",
            eventToken: token,
          ),
        ),
      );
    }
  case ClientScreencastRequired(
      :final token,
      :final screenCastResolution,
      :final ipAddressOfClient,
    ):
    try {
      final screencastLink = await getScreencastLink(
        screenCastResolution: screenCastResolution,
        clientIp: ipAddressOfClient,
      );
      serverCommunicatorDuplex.eventsToSendToServer.add(
        ClientScreencastResponse.generateResponseTo(
          responseToToken: token,
          screencastUri: screencastLink,
        ),
      );
      _eventStatusStream?.add(
        Either.right(
          EventHandledSuccessfuly(
            message: "Handled get screencast successfuly",
            eventToken: token,
          ),
        ),
      );
    } on Exception catch (error, stacktrace) {
      getLogger().warning(
          "Exception while handling
              ClientScreencastRequired",
          error,
          stacktrace);
      serverCommunicatorDuplex.eventsToSendToServer.add(
        ClientErrorResponse.generateResponseTo(
          responseToToken: token,
          message: "Unable to get screencast",
```

```
          errorString: error.toString(),
          stacktraceString: stacktrace.toString(),
        ),
      );
      _eventStatusStream?.add(
        Either.left(
          EventHandleFailed(
            message: "Handling of screen cast failed",
            eventToken: token,
          ),
        ),
      );
    }
  case ClientStopScreencast(
      :final token,
    ):
    try {
      await stopScreencast();
      serverCommunicatorDuplex.eventsToSendToServer.add(
        ClientAcknowledgeResponse.generateResponseTo(
          responseToToken: token,
        ),
      );
      _eventStatusStream?.add(
        Either.right(
          EventHandledSuccessfuly(
            message: "Handled stop screencast successfuly",
            eventToken: token,
          ),
        ),
      );
    } on Exception catch (error, stacktrace) {
      getLogger().warning(
          "Exception while handling ClientStopScreencast",
          error,
          stacktrace);
      serverCommunicatorDuplex.eventsToSendToServer.add(
        ClientErrorResponse.generateResponseTo(
          responseToToken: token,
          message: "Unable to stop screencast",
          errorString: error.toString(),
          stacktraceString: stacktrace.toString(),
        ),
      );
      _eventStatusStream?.add(
        Either.left(
```

```
        EventHandleFailed(
          message: "Handling of stop screen cast failed",
          eventToken: token,
        ),
      ),
    );
  }
case ClientScreenshotRequired(
    :final token,
  ):
  try {
    final photoBinary = await getScreenshot();
    serverCommunicatorDuplex.eventsToSendToServer.add(
      ClientScreenshotResponse.generateResponseTo(
        responseToToken: token,
        photoBinary: photoBinary.toList(),
      ),
    );
    _eventStatusStream?.add(
      Either.right(
        EventHandledSuccessfuly(
          message: "Handled get screenshot successfuly",
          eventToken: token,
        ),
      ),
    );
  } on Exception catch (error, stacktrace) {
    getLogger().warning(
        "Exception while handling ClientGetScreenshot",
        error,
        stacktrace);
    serverCommunicatorDuplex.eventsToSendToServer.add(
      ClientErrorResponse.generateResponseTo(
        responseToToken: token,
        message: "Unable to get screenshot",
        errorString: error.toString(),
        stacktraceString: stacktrace.toString(),
      ),
    );
    _eventStatusStream?.add(
      Either.left(
        EventHandleFailed(
          message: "Handling of get screenshot failed",
          eventToken: token,
        ),
      ),
```

```
    );
  }
case ClientShowMessage(
    :final token,
    :final message,
    :final messageType,
  ):
  try {
    await showMessageGui(message: message, messageType:
        messageType);
    serverCommunicatorDuplex.eventsToSendToServer.add(
      ClientAcknowledgeResponse.generateResponseTo(
        responseToToken: token,
      ),
    );
    _eventStatusStream?.add(
      Either.right(
        EventHandledSuccessfuly(
          message: "Handled show message successfuly",
          eventToken: token,
        ),
      ),
    );
  } on Exception catch (error, stacktrace) {
    getLogger().warning("Exception while handling
        ClientShowMessage",
        error, stacktrace);
    serverCommunicatorDuplex.eventsToSendToServer.add(
      ClientErrorResponse.generateResponseTo(
        responseToToken: token,
        message: "Unable to show message",
        errorString: error.toString(),
        stacktraceString: stacktrace.toString(),
      ),
    );
    _eventStatusStream?.add(
      Either.left(
        EventHandleFailed(
          message: "Handling of showing message failed",
          eventToken: token,
        ),
      ),
    );
  }
case ServerAdvertisementRecieved():
case ConnectionAccepted():
```

```
      case ConnectionDeclined():
      case ServerErrorResponse():
        // No need to handle them here, since they're platform
            independent events
        break;
    }
  },
  cancelOnError: true,
);

serverEventListener.onDone(
  () async {
    getLogger().config(
        "Finished listening to events from server in Linux
            event handler");
    _eventStatusStream?.addIfNotClosed(
      Either.left(
        EventsFromServerFinished(
          message: "Finished listening to events in Linux event
              handler",
        ),
      ),
    );
    await _eventStatusStream?.close();
  },
);

serverEventListener.onError(
  (error, stacktrace) async {
    getLogger().severe(
      "Server event listener had an error",
      error,
      stacktrace,
    );
    _eventStatusStream?.addIfNotClosed(
      Either.left(
        EventsFromServerFinished(
          message:
              "Events from server ended with error in Linux
                  event handler",
        ),
      ),
    );
    await _eventStatusStream?.close();
  },
);
```

```
    // Will keep running until it's closed by the top two factors
        or external stop
    getLogger().config("Going to start listening to
        eventStatusStream");
    yield* _eventStatusStream!.stream;
    getLogger().config("Cleaning up Linux event handler");
    await stopScreencast();
    await serverEventListener.cancel();
    _eventStatusStream = null;
    getLogger().config("Finished cleaning up Linux event handler")
        ;
    return;
  }

  @override
  Future<void> stopHandling() async {
    getLogger().config("Stopping linux platform handler");
    await stopScreencast();
    _eventStatusStream?.addIfNotClosed(
      Either.left(
        EventHandlerCancelled(
          message: "Event handler stopped",
        ),
      ),
    );
    getLogger().config("Closing the eventStatus stream");
    await _eventStatusStream?.sink.close();
    await _eventStatusStream?.close();
  }
}
```

CAR TRAFFIC SIGN RECOGNIZER

PROJECT REPORT

Submitted By

AMIN MOHAMMED O.K

Reg. No. CCAVBCA029

for the award of the Degree of

Bachelor (BCA)

of Computer Application
(University of Calicut)

under the guidance of

Ms.Varsha Ganesh

Assistant Professor

BACHELOR OF COMPUTER APPLICATION
DEPARTMENT OF COMPUTER SCIENCE
CHRIST COLLEGE(AUTONOMOUS)
IRINJALAKUDA, KERALA
INDIA

2021-2024

DEPARTMENT OF COMPUTER SCIENCE

CHRIST COLLEGE (AUTONOMOUS)

IRINJALAKUDA

CERTIFICATE

This is to certify that the project report entitled ***"Car Traffic Sign Recognizer"*** *is a bonfied record of the project work done by* ***Amin Mohammed O.K*** *in partial fulfillment of the requirement for the sixth semester of* ***Bachelor of Computer Application*** *in Department of Computer Science of* **CHRIST COLLEGE (AUTONOMOUS) IRINJALAKUDA**.

Ms.Varsha Ganesh
Assistant Professor
Internal Guide

Ms.Sini Thomas
Head of the Department
Computer Science

EXTERNAL EXAMINER

INTERNAL EXAMINER

DECLARATION

I hereby declare that this project work **"CAR TRAFFIC SIGN RECOGNIZER"** submitted to Christ College (Autonomous) Irinjalakuda, affiliated to Calicut University in partial fulfillment of the requirement for the award of the Bachelor of Computer Application, is a record of original work done by me, under the guidance of Ms.VARSHA GANESH, Department of Computer Science.

Place: Irinjalakuda AMIN MOHAMMED O.K

ACKNOWLEDGEMENT

First and foremost, I would like to thank Lord almighty for his providence and for being the guiding light throughout the project. I wish to express my sincere gratitude to our beloved Department for giving us all the facilities for our project. I take this opportunity to express my gratitude to the class teacher Ms.SOUMYA P.S and the Head of the Department Ms.SINI THOMAS who has supported us throughout the course of this project. I am thankful for their aspiring guidance and valuable advices during the project work. I express my sincere thanks to our project guide Ms.VARSHA GANESH for supporting and guiding us throughout the project. I would take this opportunity to specially thank all other faculty members for their constant and continuous motivation. Finally, I would like to thank my family and friends for giving valuable advices and moral support throughout our project.

ABSTRACT

CAR TRAFFIC SIGN RECOGNIZER is a website designed to helps the driver about recognition of road signs to avoid road accidents. It represents an important feature of advanced driver assistance system, contributing to the safety of the drivers, autonomous vehicles as well and to increase driving comfort. The website has three kind of login facilities :- admin login, user login and driver login. All these features make this website more adaptable and user-friendly.

Contents

Chapter 1

1 Introduction

In today's world road conditions drastically improved as compared with past decades. Obviously, vehicle's speed increased. So, on driver's point of view there might be chances of neglecting mandatory road signs while driving. This project explores the system to helps the driver about recognition of road signs to avoid road accidents. An automatic means of detecting and recognizing traffic signs can make a significant contribution to this goal by providing a fast method of detecting, classifying and logging signs. This method helps to develop the inventory accurately and consistently.

The project is mainly focused on automatic recognition of warning signs placed in local roads captured by image clips. The system classifies the traffic signs present in the image into different categories. With this model, we can read and understand traffic signs which are very important task for autonomous vehicles.

1.1 Overview

The objective of the Car Traffic Sign Recognizer website is to design a simple and adaptable website which helps users to increase transportation efficiency, road safety and to reduce the environmental impact with the use of advanced communication technologies. The road signs are placed on either roadside or above the roads. These signs provide mandatory information regarding to guiding, warning and regulating the behaviors to drivers in order to make driving safer and easier. A system is developed to assist the drivers, based on detection and recognition of signs which corrects the most unsafe driving behaviors.

Chapter 2

2 System Analysis

2.1 Purpose

The purpose of the traffic sign recognition project is to develop a sophisticated system that can automatically detect, interpret, and respond to traffic signs in real-time. This project aims to empower vehicles with the ability to perceive and understand the various types of traffic signs encountered on roadways. Ultimately, the goal of this project is to create a robust and reliable solution for enhancing the awareness and decision-making capabilities of drivers and autonomous vehicles alike.

2.1.1 Existing System

Existing systems of traffic sign recognition utilize a combination of cameras, sensors, and sophisticated image processing algorithms to detect and interpret traffic signs in real-time. By capturing images of traffic signs and analyzing them these systems can recognize various types of signs, including speed limits, stop signs, and lane restrictions. Limitations of existing system : The system may still encounter challenges in accurately detecting and interpreting traffic signs, leading to false detections or missed signs, potentially impacting driver safety.

2.1.2 Proposed System

The proposed system of traffic sign recognition aims to address the current limitations by leveraging advancements in artificial intelligence, particularly deep learning algorithms. By training deep neural networks on large datasets of annotated traffic sign images, the system would learn to accurately detect and interpret various types of signs, including speed limits, stop signs, and lane restrictions, across diverse environmental conditions.

2.2 Problem definition

To know what the problem is and what the needs are before developing it. Here we design Car Traffic Sign Recognizer to detect the traffic signs accurately.

2.3 Feasibility Study

After the problem is clearly understood and solutions proposed, the next step is to conduct the feasibility study. Feasibility study is defined as evaluation or analysis of the potential impact of the proposed project or program. The objective is to determine whether the proposed system is feasible. There are three aspects of feasibility study to which the proposed system is subjected as discussed below.

2.3.1 Technical Feasibility

Technical feasibility assess whether the current technical resources are sufficient for the new system. We can upgrade the level of technology for supporting our website. We check whether the proposed system can be implemented in the present system without supporting the existing hardware.

2.3.2 Economical Feasibility

Economic feasibility determines whether the time and money are available to develop the system. There is no additional hardware used to develop the site, it is inexpensive to build.

2.3.3 Operational Feasibility

Operational feasibility determines if the human resources are available to operate the system once it has been installed. No extra training is needed to use this system. Anyone who has knowledge in internet in english language can easily use the system. The resources that are required to implement or install are already available with the organization.

Chapter 3

3 Software Requirement Specification

3.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the CAR TRAFFIC SIGN RECOGNIZER. It illustrate the purpose and complete description for the development of the system. It explain system constraints, interface and interactions with other external applications.

3.2 Scope

The scope of car traffic sign recognizer extends to improving road safety, enhancing driver assistance technologies, optimizing traffic flow, and advancing the capabilities of autonomous vehicles and smart transportation systems.

3.3 Overall Description

This section give an overview of our website, CAR TRAFFIC SIGN RECOGNIZER. This project aims to develop an intelligent system capable of automatically detecting, recognizing, and interpreting traffic signs from images captured by cameras mounted on vehicles. Upon detection, the system may provide visual or auditory alerts to drivers, contributing to improved road safety and compliance with traffic laws. It plays a vital role in enhancing driver assistance systems, navigation, and autonomous vehicle technologies, ultimately leading to safer and more efficient transportation systems.

3.3.1 Product Perspective

The car traffic sign recognition project aims to develop a robust and user-friendly system that enhances driver safety and convenience. The project can develop a comprehensive solution that enhances road safety, improves driver assistance technologies, and advances the capabilities of autonomous vehicles and smart transportation systems.

3.3.2 Product Functionality

The system provides a comprehensive solution that enhances road safety, improves driver assistance technologies, and advances the capabilities of autonomous vehicles and smart transportation systems by detecting the traffic signs and notifying the driver.

3.3.3 Users and Characteristics

There are three types of users that interact with the site - admin, user and driver. Each of these have different tasks which is performed. Users are able to detect the traffic signs, view drivers and book drivers. Drivers can view the bookings and approve it. Admin is able view all the users and drivers and has the authority to delete the users or drivers.

3.4 Specific Requirements

3.4.1 Hardware Requirements

- System: LAPTOP-TE3E25EG
- Processor: i3 8th Gen or above
- Speed: 2.80 GHz
- RAM capacity: 8 GB
- Hard Dsk drive: 128 GB
- Keyboard: Standard
- Mouse: Standard
- Monitor: 18.5” LED Monitor

3.4.2 Software Requirements

- Operating System: Windows
- Languages used: Python, Django
- Database : SQLite3
- Technologies used: HTML,Javascript,CSS

3.5 Functional Requirements

It contains three main modules.

- 1.Admin
- 2.User
- 3.Driver

Admin

An Admin account is used for editing or managing the website dynamically by Admin panel. The admin can view the users and drivers and has the authority to delete any user or driver.

User

The user can detect the traffic signs. There is a feature providing for booking a driver and viewing the bookings.

Driver

The driver can view the bookings and user details. The driver can approve the booking.

3.6 Non Functional Requirements

Non-functional requirements define the overall qualities or attributes of the resulting system. Non-functional system place restrictions on the product being developed, the developed process, and specify external constraints that the product must meet. Examples of non-functional requirements include safety, security, usability, reliability and performance requirements. Project management issues (costs, time and schedule) are often considered as non-functional requirements. The principle non-functional constraints which are relevant to critical systems :

- performance
- security
- safety
- usability

Performance

Performance requirements concern the speed of operation of a system. Types of performance requirements :

- Response requirements (how quickly the system reacts to a user input).
- Throughput requirements (how much the system can accomplish within a specified amount of time).
- Availability requirements (is the system available for service when requested by end users). The speed of operation of this system is adequate for the requirements.

Reliability

- Reliability is the ability of a system to perform its required function under stated conditions for a specified period of time.
- Constraints on runtime behavior of the system. This system is reliable because its functionalities can be done on the required conditions.

Safety

Safety requirements are not required which exclude unsafe situation from the possible solution of the system.

Usability

Usability is the ease with which a user can learn to operate, prepare inputs for, and interpret outputs of system or components.
Usability requirements include :

- information error messages.
- well-formed user interfaces.

3.7 Interface Requirements

3.7.1 Hardware interfaces

The system must run over the internet, all the hardware shall require to connect internet will be hardware interface for the system. As for example modem, WAN-LAN.

3.7.2 Software interfaces

Software interface required for the working of the project is the appropriate operating system.

3.7.3 Communication interfaces

The user of the site communicate using the network connectivity and the data set is accessed

3.8 Security Requirements

- User accesses only their account.
- Validation of input is handled.
- This application containing the computer systems is physically secured against arms or surreptitious entry by intruders.
- Users must be authorized carefully to reduce changes of any such user giving access to an intruder in exchange for a bribe or other favour.

3.9 Platform Used

Windows 11 is the latest iteration of Microsoft's operating system, introducing a refreshed user interface, enhanced productivity features, and improved performance. With a sleek design aesthetic centered around simplicity and productivity, Windows 11 offers users a more intuitive and personalized computing experience.Windows 11 represents Microsoft's vision for the future of computing, blending familiarity with innovation to empower users to do more. Windows 11 brings performance improvements, with optimizations for better energy efficiency and faster wake times. With support for gaming features like DirectStorage and Auto HDR, along with enhanced security measures, Windows 11 aims to cater to a wide range of users, from casual consumers to power users and professionals, ushering in a new era of computing excellence.

3.10 Technologies Used

Python

Python is a high-level programming language renowned for its simplicity, versatility, and readability. With its clean and concise syntax, Python is accessible to beginners while remaining powerful enough for complex applications in various domains, including web development, data analysis, machine learning, artificial intelligence, and automation.Its extensive standard library and robust ecosystem of third-party packages make it a favorite among developers for rapid prototyping, building scalable applications, and solving a wide range of computational problems. Python's interpreted nature and cross-platform compatibility further contribute to its popularity, enabling developers to write code once and run it on multiple platforms without modification. Overall, Python's ease of use, flexibility, and community support make it a go-to choice for developers worldwide.

SQLite3

SQLite3 is a lightweight, self-contained, serverless relational database engine that is widely used for embedded systems, mobile applications, and small-scale database-driven websites. It is part of the SQLite project, which aims to provide a simple, fast, and reliable database solution with minimal setup and administration. SQLite3 is unique in that it operates as a single disk file and requires no configuration or administration, making it extremely easy to deploy and use. Despite its small footprint, SQLite3 supports many

standard SQL features, including transactions, triggers, and views, making it suitable for a wide range of applications. Overall, SQLite3 is an excellent choice for projects requiring a lightweight and efficient database solution without the overhead of a full-fledged database management system.

Chapter 4

4 Design Document

4.1 Purpose

The purpose of a car traffic sign recognizer project is to enhance road safety, improve driver assistance systems, and facilitate the development of autonomous vehicles. By developing an intelligent system capable of automatically detecting, recognizing, and interpreting traffic signs from images captured by onboard cameras, this project aims to provide valuable information to drivers in real-time. The system can alert drivers about speed limits, stop signs, yield signs, and other relevant traffic regulations, helping them make informed decisions and comply with road rules more effectively. Additionally, integrating traffic sign recognition technology into vehicles contributes to the advancement of autonomous driving technologies by enabling vehicles to perceive and understand the surrounding environment, ultimately leading to safer and more efficient transportation systems. Purpose of this document is to give the detailed description of the architecture and system design for the software. The system design document shows the software will be structured to satisfy the requirement identified in the software requirement specification. It is a translation of requirements into a description of the software structure, software component, interfaces and data necessary for the implementation phase. In a complete software design document, each requirements must be traceable to one or more design entities.

4.2 Scope

Car Traffic Sign Recognizer is a website which helps in detecting the traffic signs and alert the driver to ensure safe and comfortable driving.

4.3 Overview

The purpose of this document is to help the reader to visualize the solution to the project presented.This document verifies how the design meet the requirement stimulated in the SRS document through design viewpoints. The design viewpoints will cover all design elements presented before. This document will provide a direct approach to the development

of this project hence reducing feature creep and ponitedly determine the quality of the design.

4.4 Data Design

Database are the storehouses of data used in the software systems.The data is stored in tables inside the database. Several tables are created for the manipulation of the data for the system.Two essential settings for a database are

- Primary Key- The field that is unique for all the record occurrence.
- Foreign Key- The field used to set relation between tables.

Normalization

- Normalization is the process of organizing the data in the database.
- Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate the undesirable characteristics like Insertion, Update and Deletion Anomalies.
- Normalization divides the larger table into the smaller table and links them using relationship.
- The normal form is used to reduce redundancy from the database table.

Tables

textapp_login

Name	DataType	Constraints	Description
id	int	Primarykey	ID of registered people
username	varchar(100)	Notnull	Name of registered people
password	varchar(100)	Notnull	Password of registered people
usertype	varchar(100)	Notnull	Type of registered people
status	varchar(100)	Notnull	Current status of the booking
driverid_id	bigint	Foreignkey	ID of driver

textapp_user

Name	DataType	Constraints	Description
id	int	Primarykey	ID of users
uname	varchar(100)	Notnull	Name of users
uaddress	varchar(100)	Notnul	Address of users
ucontact	varchar(100)	Notnull	Contact number of the users
uemail	varchar(100)	Notnull	Email id of users
upassword	varchar(100)	Notnull	Password of users

textapp_driver

Name	DataType	Constraints	Description
id	int	Primarykey	ID of users
dname	varchar(100)	Notnull	Name of drivers
daddress	varchar(100)	Notnul	Address of drivers
dcontact	varchar(100)	Notnull	Contact number of the drivers
demail	varchar(100)	Notnull	Email id of drivers
dpassword	varchar(100)	Notnull	Password of drivers
status	varchar(100)	Notnull	Current status of the booking

textapp_booking

Name	DataType	Constraints	Description
id	int	Primarykey	ID of users
date	varchar(100)	Notnull	Date of booking
status	varchar(100)	Notnull	Current status of the booking
did_id	bigint	Foreignkey	ID of driver
uid_id	bigint	Foreignkey	ID of user

Chapter 5

5 Development of the System

This website can be decomposed into a number of submodules.The submodules of car traffic sign recognizer are administrator, user and driver. Each submodule have specific objectives, to avoid unwanted coupling between modules and to increase cohesion, modules are again decomposed into submodules.

Chapter 6

6 System Testing

Testing is the penultimate step of software development.An elaborate testing of the data is prepared and the system is using the test data, while doing testing, errors are noted and correction is made. The users are trained to operate the developed system. Both hardware and software securities are made to run the developed system successfully. System testing is aimed at ensuring the system works accurately before the live operation commences.

6.1 Test Plan

6.1.1 Scope

This test plan will cover the following testing activities as identified in the testing strategy.

- White Box Testing

White box testing is a testing technique, that examines the program structure and derives test data from the program logic/code.In white box testing, the UI is bypassed.Inputs and outputs of ratings are tested directly at the code level and the results are compared against specifications.It reveals errors in "hidden" code.

- Black Box Testing

Black-box testing is a method of software testing that examines the functionality of an application based on the specifications.Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end-user would.

- Unit Testing

The module includes four main components to undergo unit testing.It checks the syntax error, logic error and validity of the program correctness.This test will be performed by the developers.

- Integration Testing

After all components pass the unit testing, they get to be tested whether they work correctly when they are running concurrently and communicating to each other.Integration tests exercise an entire subsystem and ensure that a set of components play nicely together.

- Internal Data Testing

We will test the validity of the data before it enters the database to avoid any problems that may face us in the database. We will test the encryption of the personal information of all the users along with the user names and passwords to ensure maximum security of the users privacy.

6.1.2 Software risk issues

In this section, the plan is to test the risk involved in critical issues such as: Difficult to run on visual studio due to large loading time. Some inherent software risks such as complexity exist; also these issues need to be identified. Proper network connection Working of Mysql database

6.1.3 Features to be tested

- Test whether correct user name and password allows you to login.
- Test whether invalid user name and password prevents you from login.
- Test whether there is any connection problem in the Server.
- Test whether the user and admin details are entered correctly.
- Test whether invalid data entry allows saving data successfully.
- Test whether all pages are loaded correctly.
- Test whether watermark is embedded and extracted properly.
- Test whether the provided security works correctly.

6.2 Test consolidation

6.2.1 Test item

The items or features to be tested in the test cases are included in the document. Each and every user input is tested. The present condition of the system is tested. It is checked to make sure that environment is ready for the application to work.

6.2.2 Input specifications

These are the inputs required to execute the test case. All required inputs including data elements and values are given below.

Data	Values
Email	A valid email ID
Password	Combination of characters and numbers
Phone number	10 Digit number

Chapter 7

7 System Implementation and Maintenance

7.1 Implementation

System implementation is the conversion of new system into an operating one which involves creating compatible files, training clients and installing hardware. User training is crucial for minimizing resistance to change and giving chance to prove its worth. Training aids user friendly manuals and healthy screens provide the user with a good start. Software maintenance follows conversion to the extent that changes are necessary to maintain satisfactory operations relative to changes in the user's environment. Maintenance often includes minor enhancements or corrections to the problem that surface late in the systems operations.In the implementation phase, the team builds the components either from scratch or by composition. Given the architecture document meant from the design phase and the requirement document from the analysis phase, the team should build exactly what has been requested, though there is still room for innovation and flexibility. For example, a component may be narrowly designed for this particular system, or the component may be made more general to satisfy a reusability

- Careful planning
- Investigation of system and constraints
- Design the methods to achieve changeover.
- Training the staff in the changed phase.
- Evaluation of change over method.
- The method of implementation and time scale to be adopted are found out initially.

7.2 Maintenance

This phase occurs as a result of deploying the whole system at the end users organization. They will perform the beta testing at the end users and inform to the developers about any needed modification to the application. The customer records all the problems that are encountered during the beta testing and reports these to the developer at regular intervals.

7.2.1 Corrective Maintenance

Even with the best quality assurance activities, it is likely that the customer will uncover defects in the software.Corrective maintenance changes the software to correct the defects. Corrective Maintenance activity may consist of repair, restoration or replacement of equipment. This activity will be result of a regular inspection, which identifies the failure in time for corrective maintenance to be planned and scheduled, then performed during a routine maintenance shutdown.

7.2.2 Adaptive Maintenance

Over time, the original environment(CPU, operating system, business rules, external product characteristics) for which the software was developed is likely to change. Adaptive maintenance results in modification to the software to accommodate changes to its external environment.

7.2.3 Enhanced Maintenance

As software is used, the customer/user will recognize additional functions that will provide the benefit. Perfect maintenance extends the software beyond its original functional requirements.

7.2.4 Preventive Maintenance

Computer software deteriorates due to change, and because of this preventive maintenance often called software re-engineering, must be conducted to enable the software to serve the needs of its end users. Preventive maintenance makes changes to computer programs so that they can be more easily corrected, adapted and enhanced.

Chapter 8

8 Conclusion and Future Scope

8.1 Conclusion

There are three types of users that interact with the system Admin, User and Driver. Each of these three types has different uses of the system so each of them has their own panel. Admin can manage all the features of website dynamically by login on admin panel. The users can detect the traffic signs and can also book a driver and view the bookings. The driver can view the bookings and user details and can approve them.

8.2 Future Scope

- Increased accuracy using better camera device
- Adaptability to dynamic environments

Appendix

A Data Flow Diagram

Data flow is the one of the best way of documenting the entire functionality of the system. For the system ,which will have data flows in and have some processing inside and then some data flow out from the system can be documented or represented effectively by means of data flow out from the system can be documented or represented effectively by means of data flow diagrams. The data flow diagram are a diagrammatic representation of the system,which has input,process and output. Once any system is represented using a data flow diagram we can identify the following things easily:

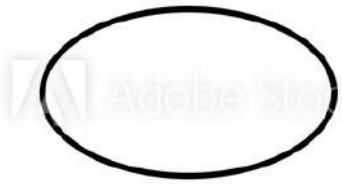
- Various entities interacting with the system are identified
- Flow of data from one entity to another is identified
- The various processes involved in between the interaction of two or more entities in the system are clearly pointed out
- The various data stores which hold the data in between the process,are clearly identified

Some Data Flow Diagram charting forms:

A.1 External source or receiver

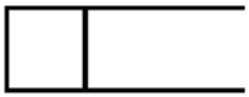
A source or sink is a person or part of organization ,which enters or receives information from the system,but is considered to be outlining the contest of data flow model.

A.2 Transform process



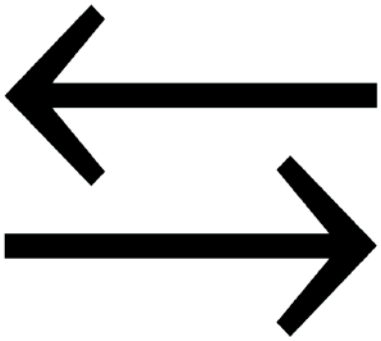
A process represents transformation where incoming data flowsare changed into outgoing data flow

A.3 Data Store



A data store is repository of data that is to be stored for use by one or more process may be as simple as buffer or queue or sophisticated as relational database. They should have clear names.If a process merely uses the contest of store and does not alter it,the arrowhead goes only form the store to the process. If a process alters the details in the store then double-headed arrow is used.

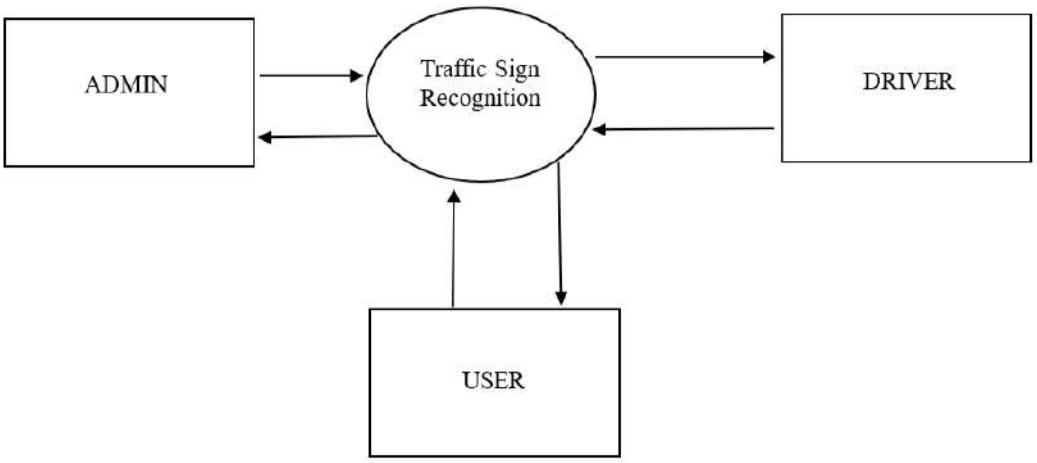
A.4 Data flow



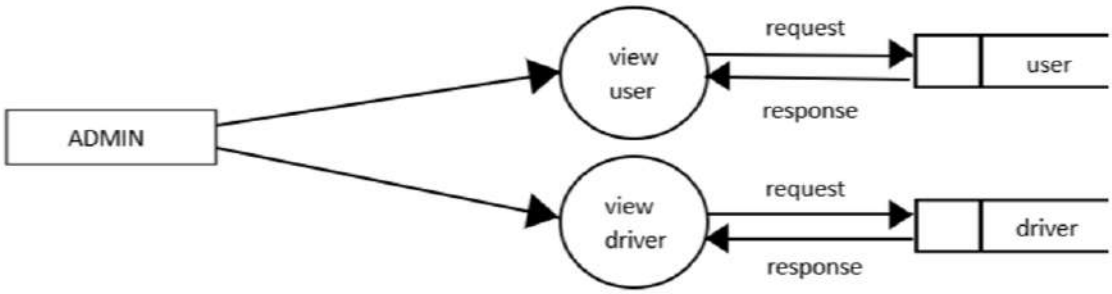
A data flow is a route, which enable packets of data to travel from one point to another.Data may flow,with arrowhead pointing in the direction of the flow

B Data Flow Diagrams

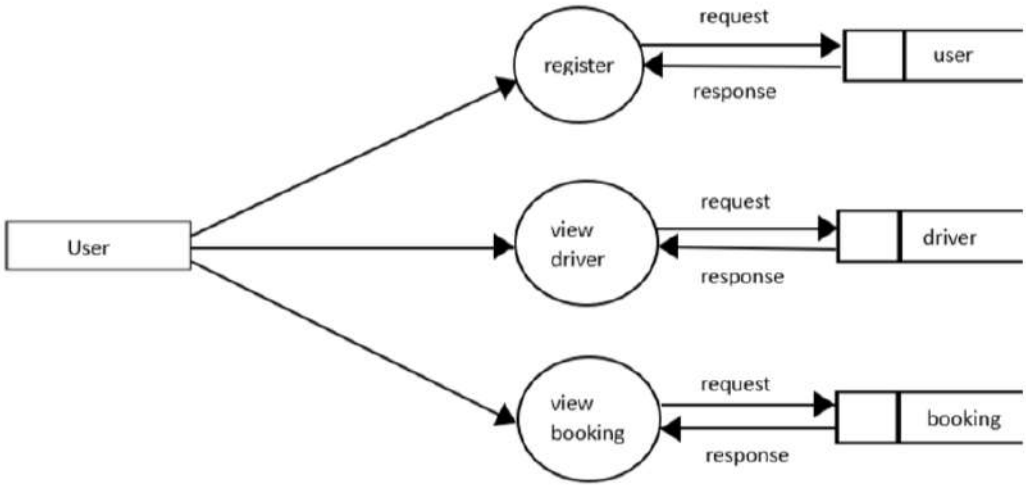
B.1 Level 0



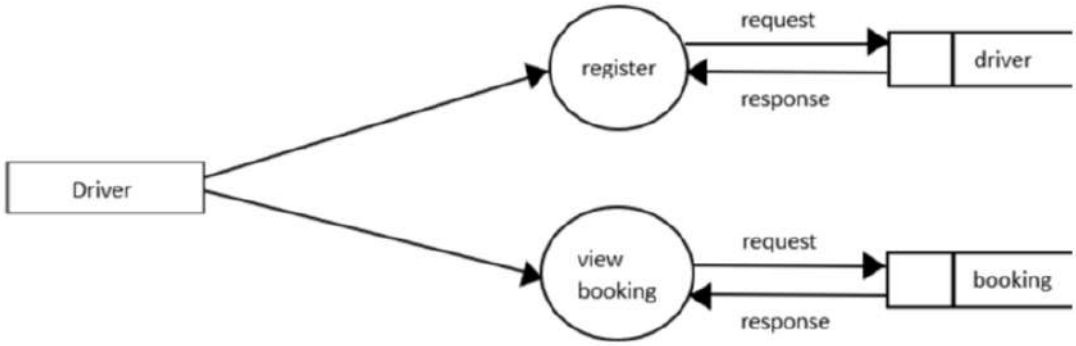
B.2 Level 1.1 - Admin



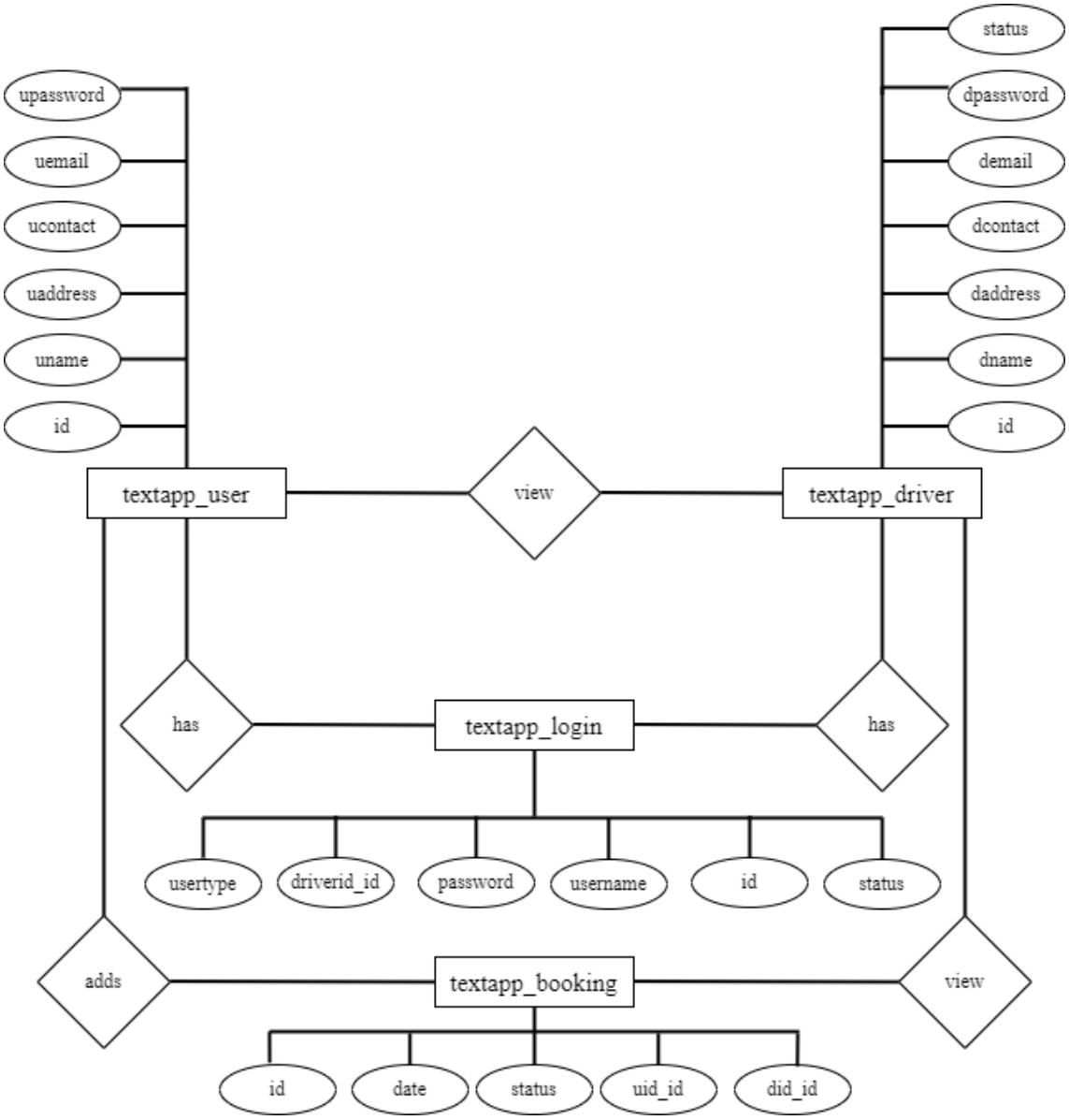
B.3 Level 1.2 - User



B.4 Level 1.3 - Driver

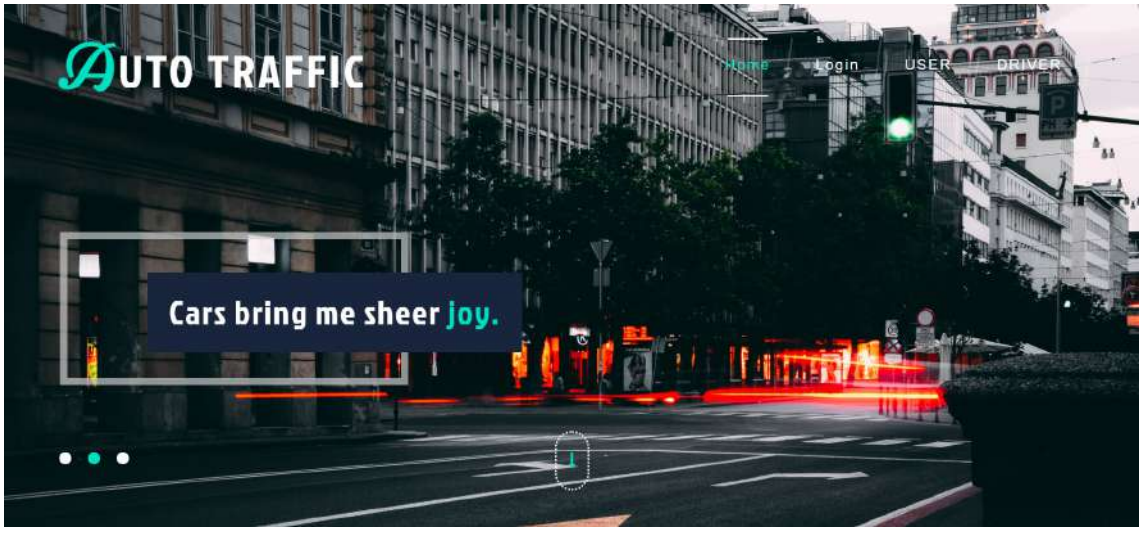


B.5 ER Diagram

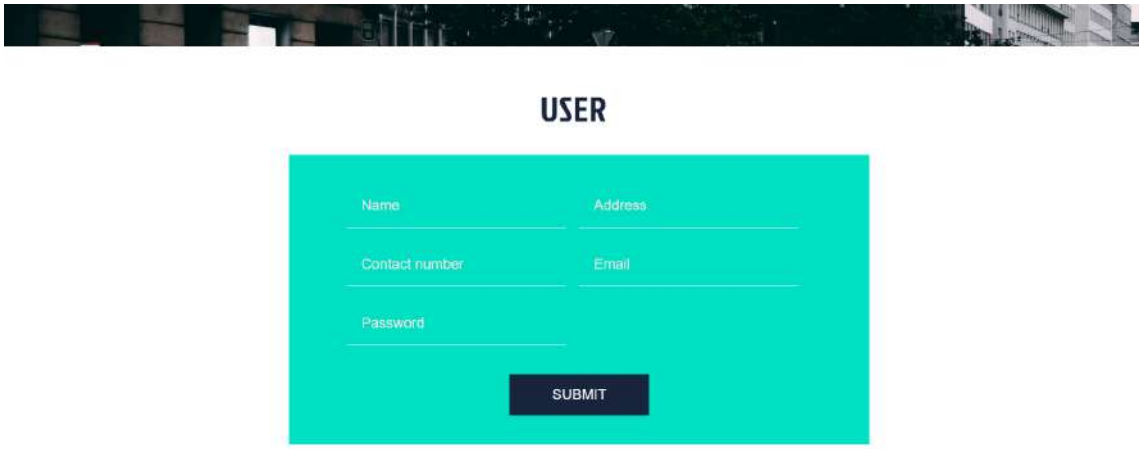


C USER INTERFACES

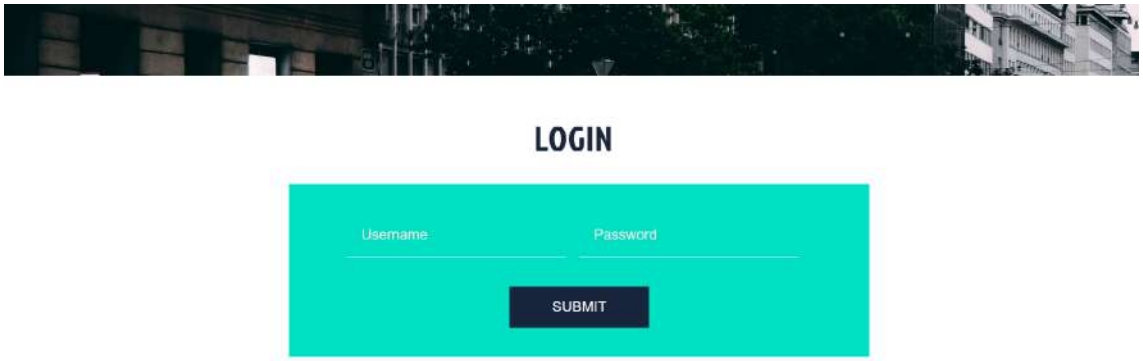
C.1 HOME



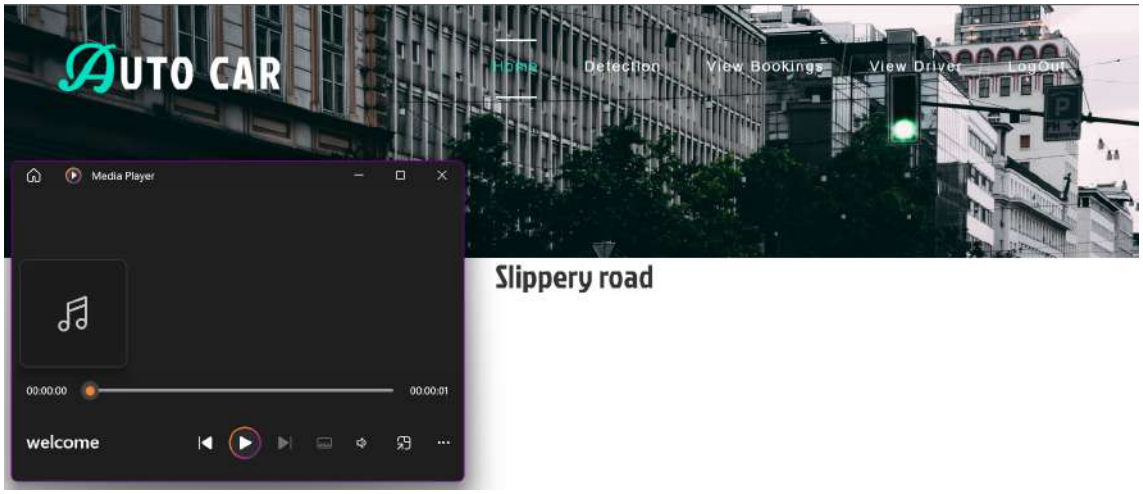
C.2 REGISTRATION



C.3 LOGIN



C.4 DETECTION



C.5 VIEW DRIVERS

NAME	ADDRESS	CONTACT	EMAIL	ACTION
Williams	Kodungalloor	8988713401	williams@gmail.com	Book Now
Simson	Thrissur	9099170787	simson@gmail.com	Book Now
Myna	Chalakkudi	8799136323	myna@gmail.com	Book Now

C.6 VIEW BOOKING

NAME	ADDRESS	CONTACT	EMAIL	STATUS	ACTION
Richard	Chalakkudi	8966139256	richard321@gmail.com		Approve
John	Aluva	8590939355	john@gmail.com		Approve
Philip	Kodungalloor	9890583346	philip@gmail.com		Approve

D Code

urls.py

```
from django.contrib import admin
from django.urls import path
from textapp import views
urlpatterns = [
    path('admin/', admin.site.urls),

    path('index/', views.index),
    path('adminhome/', views.adminhome),
    path('driverhome/', views.driverhome),
    path('userhome/', views.userhome),
    path('', views.index),
    path('login/', views.logins),
    path('userreg/', views.userreg),
    path('driverreg/', views.driverreg),
    path('udp/', views.udp),
    path('adminviewdriver/', views.adminviewdriver),
    path('adminviewuser/', views.adminviewuser),
    path('userviewdriver/', views.userviewdriver),
    path('viewbooking/', views.userviewbooking),
    path('userviewbooking/', views.userviewbooking),
    path('driverviewbooking/', views.driverviewbooking),
    path('ddetection/', views.ddetection),
    path('driverdetection/', views.driverdetection),
    path('imagebyenter/', views.imagebyenter),
    path('prediction/', views.predict),
    path('deletedriver/', views.deletedriver),
    path('deleteuser/', views.deleteuser),
    \# path('udp/', views.udp),

]
```

model.py

```
import os
from tensorflow.keras.preprocessing import image
import matplotlib.pyplot as plt
```

```
import numpy as np
import tensorflow.python.keras.utils as generic\_utils
import random,shutil
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dropout,Conv2D,Flatten,Dense,
MaxPooling2D, BatchNormalization
from tensorflow.keras.models import load\_model


def generator(dir, gen=image.ImageDataGenerator(rescale=1./255), shuffle=True,batch
\_size=1,target\_size=(24,24),class\_mode='categorical' ):

return gen.flow\_from\_directory(dir,batch\_size=batch\_size,shuffle=shuffle,
color\_mode='grayscale',class\_mode=class\_mode,target\_size=target\_size)

BS= 32
TS=(24,24)
train\_batch= generator('data/train',shuffle=True, batch\_size=BS,target\_size=TS)
valid\_batch= generator('data/valid',shuffle=True, batch\_size=BS,target\_size=TS)
SPE= len(train\_batch.classes)//BS
VS = len(valid\_batch.classes)//BS
print(SPE,VS)


\# img,labels= next(train\_batch)
\# print(img.shape)

model = Sequential([
    Conv2D(32, kernel\_size=(3, 3), activation='relu', input\_shape=(24,24,1)),
    MaxPooling2D(pool\_size=(1,1)),
    Conv2D(32,(3,3),activation='relu'),
    MaxPooling2D(pool\_size=(1,1)),
\#32 convolution filters used each of size 3x3
\#again
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D(pool\_size=(1,1)),

\#64 convolution filters used each of size 3x3
\#choose the best features via pooling
```

```
\#randomly turn neurons on and off to improve convergence
    Dropout(0.25),
\#flatten since too many dimensions, we only want a classification output
    Flatten(),
\#fully connected to get all relevant data
    Dense(128, activation='relu'),
\#one more dropout for convergence' sake :)
    Dropout(0.5),
\#output a softmax to squash the matrix into output probabilities
    Dense(2, activation='softmax')
])

model.compile(optimizer='adam',loss='categorical\_crossentropy',metrics=['accuracy'])

model.fit\_generator(train\_batch, validation\_data=valid\_batch,epochs=15,steps\_per

model.save('models/cnnCat2.h5', overwrite=True)
```

models.py

```
[verbetim]

from django.db import models




class Driver(models.Model):
    dname=models.CharField(max\_length=100,blank=True)
    daddress=models.CharField(max\_length=100,blank=True)
    dcontact=models.CharField(max\_length=100,blank=True)
    demail=models.CharField(max\_length=100,blank=True)
    dpassword=models.CharField(max\_length=100,blank=True)
    status=models.CharField(max\_length=100,blank=True)


class Login(models.Model):
    username=models.CharField(max\_length=100,blank=True)
    password=models.CharField(max\_length=100,blank=True)
```

```
    usertype=models.CharField(max\_length=100,blank=True)
    status=models.CharField(max\_length=100,blank=True)
    driverid=models.ForeignKey(Driver,null=True,on\_delete=models.CASCADE)



class User(models.Model):
    uname=models.CharField(max\_length=100,blank=True)
    uaddress=models.CharField(max\_length=100,blank=True)
    ucontact=models.CharField(max\_length=100,blank=True)
    uemail=models.CharField(max\_length=100,blank=True)
    upassword=models.CharField(max\_length=100,blank=True)


class Booking(models.Model):
    uid=models.ForeignKey(User,null=True,on\_delete=models.CASCADE)
    did=models.ForeignKey(Driver,null=True,on\_delete=models.CASCADE)
    date=models.CharField(max\_length=100,blank=True)
    status=models.CharField(max\_length=100,blank=True)
```

views.py

```
\# import email
from http.client import HTTPResponse
from django.shortcuts import render,HttpResponse,HttpResponseRedirect

from .models import *
from django.contrib import messages
from django.db.models import Max, Min,Count,Sum,Avg

import numpy as np
from PIL import Image
import cv2
import tensorflow as tf
import os
from flask\_cors import CORS, cross\_origin
from utils.utils import decodeImage
from predict import traffic
```

```
from django.db.models import Q
import cv2
import tensorflow as tf

def udp(request):
    import os
    from gtts import gTTS
    \# email="admin@gmail.com"
    \# password="admin"

    \# s=Login.objects.create(username=email,password=password,usertype='admin',statu
    \# s.save()
    \# messages.info(request,"already added")
    model\_path = "Traffic.h5"
    loaded\_model = tf.keras.models.load\_model(model\_path)
    videoCaptureObject = cv2.VideoCapture(0)
    print("**************************************************************")
    result = True
    while(result):
        ret,frame = videoCaptureObject.read()
        cv2.imwrite("NewPicture.jpg",frame)
        result = False
    videoCaptureObject.release()
    cv2.destroyAllWindows()
    imagename = "NewPicture.jpg"
    image = cv2.imread(imagename)
    print(image)
    print("**************************************************************")

    image\_fromarray = Image.fromarray(image, 'RGB')
    resize\_image = image\_fromarray.resize((30, 30))
    print("**************************************************************")

    expand\_input = np.expand\_dims(resize\_image,axis=0)
    input\_data = np.array(expand\_input)
    input\_data = input\_data/255
    pred = loaded\_model.predict(input\_data)
    result = pred.argmax()
    print("**************************************************************")
```

```
if result == 0:
    prediction = 'Speed limit (20km/h)'
    print("image",prediction)
elif result == 1:
    prediction = 'Speed limit (30km/h)'
    print("image",prediction)
elif result == 2:
    prediction = 'Speed limit (50km/h)'
    print("image",prediction)
elif result == 3:
    prediction = 'Speed limit (60km/h)'
    print("image",prediction)
elif result == 4:
    prediction = 'Speed limit (70km/h)'
    print("image",prediction)
elif result == 5:
    prediction = 'Speed limit (80km/h)'
    print("image",prediction)
elif result == 6:
    prediction = 'End of speed limit (80km/h)'
    print("image",prediction)
elif result == 7:
    prediction = 'Speed limit (100km/h)'
    print("image",prediction)
elif result == 8:
    prediction = 'Speed limit (120km/h)'
    print("image",prediction)
elif result == 9:
    prediction = 'No passing'
    print("image",prediction)
elif result == 10:
    prediction = 'No passing veh over 3.5 tons'
    print("image",prediction)
elif result == 11:
    prediction = 'Right-of-way at intersection'
    print("image",prediction)
elif result == 12:
    prediction = 'Priority road'
    print("image",prediction)
```

```
elif result == 13:
    prediction = 'Yield'
    print("image",prediction)
elif result == 14:
    prediction = 'Stop'
    print("image",prediction)
elif result == 15:
    prediction = 'No vehicles'
    print("image",prediction)
elif result == 16:
    prediction = 'Veh > 3.5 tons prohibited'
    print("image",prediction)
elif result == 17:
    prediction = 'No entry'
    print("image",prediction)
elif result == 18:
    prediction = 'General caution'
    print("image",prediction)
elif result == 19:
    prediction = 'Dangerous curve left'
    print("image",prediction)
elif result == 20:
    prediction = 'Dangerous curve right'
    print("image",prediction)
elif result == 21:
    prediction = 'Double curve'
    print("image",prediction)
elif result == 22:
    prediction = 'Bumpy road'
    print("image",prediction)
elif result == 23:
    prediction = 'Slippery road'
    print("image",prediction)
elif result == 24:
    prediction = 'Road narrows on the right'
    print("image",prediction)
elif result == 25:
    prediction = 'Road work'
    print("image",prediction)
elif result == 26:
```

```
        prediction = 'Traffic signals'
        print("image",prediction)
    elif result == 27:
        prediction = 'Pedestrians'
        print("image",prediction)
    elif result == 28:
        prediction = 'Children crossing'
        print("image",prediction)
    elif result == 29:
        prediction = 'Bicycles crossing'
        print("image",prediction)
    elif result == 30:
        prediction = 'Beware of ice/snow'
        print("image",prediction)
    elif result == 31:
        prediction = 'Wild animals crossing'
        print("image",prediction)
    elif result == 32:
        prediction = 'End speed + passing limits'
        print("image",prediction)
    elif result == 33:
        prediction = 'Turn right ahead'
        print("image",prediction)
    elif result == 34:
        prediction = 'Turn left ahead'
        print("image",prediction)
    elif result == 35:
        prediction = 'Ahead only'
        print("image",prediction)
    elif result == 36:
        prediction = 'Go straight or right'
        print("image",prediction)
    elif result == 37:
        prediction = 'Go straight or left'
        print("image",prediction)
    elif result == 38:
        prediction = 'Keep right'
        print("image",prediction)
    elif result == 39:
        prediction = 'Keep left'
```

```
        print("image",prediction)
    elif result == 40:
        prediction = 'Roundabout mandatory'
        print("image",prediction)
    elif result == 41:
        prediction = 'End of no passing'
        print("image",prediction)
    elif result == 42:
        prediction = 'End no passing veh > 3.5 tons'
        print("image",prediction)
    language = 'en'
    myobj = gTTS(text=prediction, lang=language, slow=False)
    print("**************************************************")
    print(prediction)
    myobj.save("welcome.mp3")
    os.system("welcome.mp3")
    return render(request,"user/prediction.html",{"prediction":prediction})



    return HttpResponseRedirect("/prediction")


from django.core.files.storage import FileSystemStorage

def imagebyenter(request):
    import os
    from gtts import gTTS
    if request.POST:
        image=request.FILES['image']
        saved\_filename = f"./uploads/{image.name}"
        with open(saved\_filename, "wb") as opened:
            for chunk in image.chunks():
                    opened.write(chunk)

        fs=FileSystemStorage()
        filename=fs.save(image.name,image)
        fileurl=fs.url(filename)

        model\_path = "Traffic.h5"
```

```
loaded\_model = tf.keras.models.load\_model(model\_path)

imagename = image
print(saved\_filename)
image = cv2.imread(saved\_filename)

image\_fromarray = Image.fromarray(image, 'RGB')
resize\_image = image\_fromarray.resize((30, 30))
expand\_input = np.expand\_dims(resize\_image,axis=0)
input\_data = np.array(expand\_input)
input\_data = input\_data/255
pred = loaded\_model.predict(input\_data)
result = pred.argmax()


if result == 0:
    prediction = 'Speed limit (20km/h)'
    print("image",prediction)
elif result == 1:
    prediction = 'Speed limit (30km/h)'
    print("image",prediction)
elif result == 2:
    prediction = 'Speed limit (50km/h)'
    print("image",prediction)
elif result == 3:
    prediction = 'Speed limit (60km/h)'
    print("image",prediction)
elif result == 4:
    prediction = 'Speed limit (70km/h)'
    print("image",prediction)
elif result == 5:
    prediction = 'Speed limit (80km/h)'
    print("image",prediction)
elif result == 6:
    prediction = 'End of speed limit (80km/h)'
    print("image",prediction)
elif result == 7:
    prediction = 'Speed limit (100km/h)'
    print("image",prediction)
elif result == 8:
```

```
        prediction = 'Speed limit (120km/h)'
        print("image",prediction)
    elif result == 9:
        prediction = 'No passing'
        print("image",prediction)
    elif result == 10:
        prediction = 'No passing veh over 3.5 tons'
        print("image",prediction)
    elif result == 11:
        prediction = 'Right-of-way at intersection'
        print("image",prediction)
    elif result == 12:
        prediction = 'Priority road'
        print("image",prediction)
    elif result == 13:
        prediction = 'Yield'
        print("image",prediction)
    elif result == 14:
        prediction = 'Stop'
        print("image",prediction)
    elif result == 15:
        prediction = 'No vehicles'
        print("image",prediction)
    elif result == 16:
        prediction = 'Veh > 3.5 tons prohibited'
        print("image",prediction)
    elif result == 17:
        prediction = 'No entry'
        print("image",prediction)
    elif result == 18:
        prediction = 'General caution'
        print("image",prediction)
    elif result == 19:
        prediction = 'Dangerous curve left'
        print("image",prediction)
    elif result == 20:
        prediction = 'Dangerous curve right'
        print("image",prediction)
    elif result == 21:
        prediction = 'Double curve'
```

```
        print("image",prediction)
    elif result == 22:
        prediction = 'Bumpy road'
        print("image",prediction)
    elif result == 23:
        prediction = 'Slippery road'
        print("image",prediction)
    elif result == 24:
        prediction = 'Road narrows on the right'
        print("image",prediction)
    elif result == 25:
        prediction = 'Road work'
        print("image",prediction)
    elif result == 26:
        prediction = 'Traffic signals'
        print("image",prediction)
    elif result == 27:
        prediction = 'Pedestrians'
        print("image",prediction)
    elif result == 28:
        prediction = 'Children crossing'
        print("image",prediction)
    elif result == 29:
        prediction = 'Bicycles crossing'
        print("image",prediction)
    elif result == 30:
        prediction = 'Beware of ice/snow'
        print("image",prediction)
    elif result == 31:
        prediction = 'Wild animals crossing'
        print("image",prediction)
    elif result == 32:
        prediction = 'End speed + passing limits'
        print("image",prediction)
    elif result == 33:
        prediction = 'Turn right ahead'
        print("image",prediction)
    elif result == 34:
        prediction = 'Turn left ahead'
        print("image",prediction)
```

```
        elif result == 35:
            prediction = 'Ahead only'
            print("image",prediction)
        elif result == 36:
            prediction = 'Go straight or right'
            print("image",prediction)
        elif result == 37:
            prediction = 'Go straight or left'
            print("image",prediction)
        elif result == 38:
            prediction = 'Keep right'
            print("image",prediction)
        elif result == 39:
            prediction = 'Keep left'
            print("image",prediction)
        elif result == 40:
            prediction = 'Roundabout mandatory'
            print("image",prediction)
        elif result == 41:
            prediction = 'End of no passing'
            print("image",prediction)
        elif result == 42:
            prediction = 'End no passing veh > 3.5 tons'
            print("image",prediction)
        language = 'en'
        myobj = gTTS(text=prediction, lang=language, slow=False)
        myobj.save("welcome.mp3")
        os.system("welcome.mp3")
        return render(request,"user/prediction.html",{"prediction":prediction})


    return render(request,"user/enterimage.html")

def predict(request):

    return render(request,"user/prediction.html")
```

```
def index(request):
    return render(request,"common/index.html")

def adminhome(request):
    return render(request,"admin/index.html")

def userhome(request):
    return render(request,"user/index.html")

def driverhome(request):
    return render(request,"driver/index.html")

def driverreg(request):
    if request.POST:
        name=request.POST.get("name")
        address=request.POST.get("address")
        contact=request.POST.get("contact")
        email=request.POST.get("email")
        password=request.POST.get("password")

        s=Driver.objects.create(dname=name,daddress=address,dcontact=contact,demail=e
        s.save()
        did=Driver.objects.aggregate(Max('id'))
        print(did)
        did=did['id\_\_max']
        did=Driver.objects.get(id=did)
        print(did)

    \# print(mark)
        s=Login.objects.create(username=email,password=password,usertype='driver',
        status='requested',driverid=did)
        s.save()
        messages.info(request,"already added")
    return render(request,"common/driverreg.html")



def userreg(request):
```

```
    if request.POST:
        name=request.POST.get("name")
        address=request.POST.get("address")
        contact=request.POST.get("contact")
        email=request.POST.get("email")
        password=request.POST.get("password")

        s=User.objects.create(uname=name,uaddress=address,ucontact=contact,
            uemail=email,upassword=password)
        s.save()
        s=Login.objects.create(username=email,password=password,
             usertype='user',status='requested')
        s.save()
        messages.info(request,"already added")
    return render(request,"common/userreg.html")


def logins(request):
    msg=""
    if request.POST:
        name=request.POST.get("username")
        password=request.POST.get("password")
        print(name)
        print(password)

        s=Login.objects.filter(Q(username=name),Q(password=password))

        try:
            if s[0].usertype=='admin':
                msg="Success"
                return HttpResponseRedirect("/adminhome")
            elif s[0].usertype=='user':
                s=User.objects.get(uemail=name)
                id=s.id
                request.session['uid']=id
                msg="Success"
                return HttpResponseRedirect("/userhome")
            elif s[0].usertype=='driver':
                s=Driver.objects.get(demail=name)
                id=s.id
```

```
                request.session['did']=id
                msg="Success"
                return HttpResponseRedirect("/driverhome")

        except:
            msg="invalid Username Or password"


    \# return HttpResponseRedirect('/login')

    return render(request,"common/login.html",{'msg':msg})




def adminviewdriver(request):
    s=Driver.objects.filter()
    print(s)
    if request.GET:
        id=request.GET.get("id")
        s=Driver.objects.get(id=id)
        username=s.demail
        s=Login.objects.filter(username=username).update(status='approved')
        s=Driver.objects.filter(id=id).update(status='approved')
        return HttpResponseRedirect("/adminviewdriver")
    return render(request,"admin/viewdriver.html",{"data":s})

def deletedriver(request):
    id=request.GET.get('id')
    Driver.objects.filter(id=id).delete()
    return HttpResponseRedirect("/adminviewdriver")


def adminviewuser(request):
    s=User.objects.all()
    print(s)
    return render(request,"admin/viewuser.html",{"data":s})

def deleteuser(request):
    id=request.GET.get('id')
```

```
    User.objects.filter(id=id).delete()
    return HttpResponseRedirect("/adminviewuser")


def userviewdriver(request):
    s=Driver.objects.all()
    uid=request.session['uid']
    if request.GET:
        id=request.GET.get("id")
        driverdetails=Driver.objects.get(id=id)
        userdetails=User.objects.get(id=uid)
        import datetime
        date=datetime.datetime.now()
        s=Booking.objects.create(uid=userdetails,did=driverdetails,date=date,status='
        return HttpResponseRedirect("/userviewdriver")
    return render(request,"user/viewdriver.html",{"data":s})




def userviewbooking(request):
    uid=request.session['uid']
    select=Booking.objects.filter(uid=uid)
    return render(request,"user/viewbooking.html",{"data":select})




def driverviewbooking(request):
    uid=request.session['did']
    print(uid)
    select=Booking.objects.filter(did=uid)
    print(select)
    abcd='approved'
    if request.GET:
        id=request.GET.get("id")
        select=Booking.objects.filter(id=id).update(status='approved')
        return HttpResponseRedirect("/driverhome")

    return render(request,"driver/viewbooking.html",{"data":select,"abcd":abcd})
```

```
def ddetection(request):
    from textapp import cd
    return HttpResponseRedirect("/userhome")



def driverdetection(request):
    from textapp import cd
    return HttpResponseRedirect("/driverhome")
```

predict.py

```
import numpy as np
from PIL import Image
import cv2
import tensorflow as tf

class traffic:
    def \_\_init\_\_(self,filename):
        self.filename =filename


    def trafficsign(self):

        model\_path = "Traffic.h5"
        loaded\_model = tf.keras.models.load\_model(model\_path)

        imagename = self.filename
        image = cv2.imread(imagename)

        image\_fromarray = Image.fromarray(image, 'RGB')
        resize\_image = image\_fromarray.resize((30, 30))
        expand\_input = np.expand\_dims(resize\_image,axis=0)
        input\_data = np.array(expand\_input)
        input\_data = input\_data/255
```

```
pred = loaded\_model.predict(input\_data)
result = pred.argmax()

if result == 0:
    prediction = 'Speed limit (20km/h)'
    return [{"image": prediction}]
elif result == 1:
    prediction = 'Speed limit (30km/h)'
    return [{"image": prediction}]
elif result == 2:
    prediction = 'Speed limit (50km/h)'
    return [{"image": prediction}]
elif result == 3:
    prediction = 'Speed limit (60km/h)'
    return [{"image": prediction}]
elif result == 4:
    prediction = 'Speed limit (70km/h)'
    return [{"image": prediction}]
elif result == 5:
    prediction = 'Speed limit (80km/h)'
    return [{"image": prediction}]
elif result == 6:
    prediction = 'End of speed limit (80km/h)'
    return [{"image": prediction}]
elif result == 7:
    prediction = 'Speed limit (1000km/h)'
    return [{"image": prediction}]
elif result == 8:
    prediction = 'Speed limit (120km/h)'
    return [{"image": prediction}]
elif result == 9:
    prediction = 'No passing'
    return [{"image": prediction}]
elif result == 10:
    prediction = 'No passing veh over 3.5 tons'
    return [{"image": prediction}]
elif result == 11:
    prediction = 'Right-of-way at intersection'
    return [{"image": prediction}]
elif result == 12:
```

```
        prediction = 'Priority road'
        return [{"image": prediction}]
    elif result == 13:
        prediction = 'Yield'
        return [{"image": prediction}]
    elif result == 14:
        prediction = 'Stop'
        return [{"image": prediction}]
    elif result == 15:
        prediction = 'No vehicles'
        return [{"image": prediction}]
    elif result == 16:
        prediction = 'Veh > 3.5 tons prohibited'
        return [{"image": prediction}]
    elif result == 17:
        prediction = 'No entry'
        return [{"image": prediction}]
    elif result == 18:
        prediction = 'General caution'
        return [{"image": prediction}]
    elif result == 19:
        prediction = 'Dangerous curve left'
        return [{"image": prediction}]
    elif result == 20:
        prediction = 'Dangerous curve right'
        return [{"image": prediction}]
    elif result == 21:
        prediction = 'Double curve'
        return [{"image": prediction}]
    elif result == 22:
        prediction = 'Bumpy road'
        return [{"image": prediction}]
    elif result == 23:
        prediction = 'Slippery road'
        return [{"image": prediction}]
    elif result == 24:
        prediction = 'Road narrows on the right'
        return [{"image": prediction}]
    elif result == 25:
        prediction = 'Road work'
```

```
        return [{"image": prediction}]
    elif result == 26:
        prediction = 'Traffic signals'
        return [{"image": prediction}]
    elif result == 27:
        prediction = 'Pedestrians'
        return [{"image": prediction}]
    elif result == 28:
        prediction = 'Children crossing'
        return [{"image": prediction}]
    elif result == 29:
        prediction = 'Bicycles crossing'
        return [{"image": prediction}]
    elif result == 30:
        prediction = 'Beware of ice/snow'
        return [{"image": prediction}]
    elif result == 31:
        prediction = 'Wild animals crossing'
        return [{"image": prediction}]
    elif result == 32:
        prediction = 'End speed + passing limits'
        return [{"image": prediction}]
    elif result == 33:
        prediction = 'Turn right ahead'
        return [{"image": prediction}]
    elif result == 34:
        prediction = 'Turn left ahead'
        return [{"image": prediction}]
    elif result == 35:
        prediction = 'Ahead only'
        return [{"image": prediction}]
    elif result == 36:
        prediction = 'Go straight or right'
        return [{"image": prediction}]
    elif result == 37:
        prediction = 'Go straight or left'
        return [{"image": prediction}]
    elif result == 38:
        prediction = 'Keep right'
        return [{"image": prediction}]
```

```
elif result == 39:
    prediction = 'Keep left'
    return [{"image": prediction}]
elif result == 40:
    prediction = 'Roundabout mandatory'
    return [{"image": prediction}]
elif result == 41:
    prediction = 'End of no passing'
    return [{"image": prediction}]
elif result == 42:
    prediction = 'End no passing veh > 3.5 tons'
    return [{"image": prediction}]
else:
    return [{"ERROR": "Please select another image. !!!"}]
```

UNLOCKING TRACEABILITY AND TRANSPARENCY IN SUPPLY CHAINS WITH BLOCKCHAIN TECHNOLOGY

PROJECT REPORT

Submitted By

DEVANA PRAMOD

Reg. No. CCAVBCA030

For the award of the Degree of

Bachelor (BCA)

of Computer Application

(University of Calicut)

under the guidance of

Ms.Priyanga K.K

Assistant Professor

BCA (Bachelor of Computer Application)
DEPARTMENT OF COMPUTER SCIENCE
CHRIST COLLEGE (AUTONOMOUS)
IRINJALAKUDA, KERALA

2021-2024

DEPARTMENT OF COMPUTER SCIENCE

CHRIST COLLEGE (AUTONOMOUS)

IRINJALAKUDA

CERTIFICATE

This is to certify that the project report entitled ***Unlocking Traceability and Transparency in Supply Chains with Blockchain Technology*** *is a bonfied record of the project work done by* ***Devana Pramod*** *in partial fulfillment of the requirement for the sixth semester of* ***Bachelor of Computer Application*** *in Department of Computer Science of* **CHRIST COLLEGE (AUTONOMOUS) IRINJALAKUDA**.

Ms.Priyanga K.K
Assistant Professor,CS
Internal Guide

Ms.Sini Thomas
Head of the Department
Computer Science

EXTERNAL EXAMINER

INTERNAL EXAMINER

DECLARATION

I hereby declare that this project work **"Unlocking Traceability and Transparency in Supply Chains with Blockchain Technology"** submitted to Christ College (Autonomous) Irinjalakuda, affliated to Calicut University in partial fullfillment of the requirement for the award of the Bachelor of Computer Application, is a record of original work done by me, under the guidance of Ms.PRIYANGA K.K, Assistant Professor, Department of Computer Science.

Place: Irinjalakuda DEVANA PRAMOD

ACKNOWLEDGEMENT

First and foremost, I would like to thank God almighty for his guidance and support throughout the project.I wish to express my sincere gratitude to my beloved Department for providing me all the facilities for my project. I take this opportunity to express my gratitude to the class teacher Ms.SOUMYA P.S and HOD(Head Of Department) Ms.SINI THOMAS who supported me throughout the course of this project.I am thankful for their aspiring guidance and valuable advice during the project work.I express my sincere thanks to my project guide Ms.PRIYANGA K.K for supporting and guiding me throughout the project.I would also like to take this opportunity, to specially thank, all other faculty members for their constant and continuous motivation.Finally, I would like to thank my family and friends for giving valuable advices and moral support throughout my project.

ABSTRACT

Unlocking Traceability and Transparency in Supply Chains with Blockchain Technology is a topic that investigates and proposes a blockchain based traceability framework. Traceability is very crucial for multi-tier production where customers demand transparency and quality assurance. This study proposes a blockchain-based traceability framework to address issues like information asymmetry and low visibility.It suggests using smart contracts and transaction validation rules to ensure secure information sharing.The system aims to build trust among supplychain partners by using a distributed ledger to store and authenticate transactions, offering flexibility and transparency.

Contents

Chapter 1

1 Introduction

Supply chains are intricate systems critical to global operations, but they often lack transparency and traceability, leading to issues like disruptions, counterfeits, and ethical concerns. Blockchain technology, known for its decentralized and immutable record-keeping, offers a solution. This project explores blockchain's potential in enhancing supply chain traceability and transparency across industries like manufacturing, healthcare, food, and luxury goods. It analyses real-world case studies, discusses benefits and limitations, and aims to inform businesses, policymakers, and researchers about the transformative impact of blockchain on supply chains, emphasizing its role in building trust and improving operations.By facilitating transparent and secure traceability throughout the supply chain, blockchain not only enables better information sharing but also fosters trust among supply chain partners spread across the globe.
Additionally, existing methodologies for blockchain-based supply chain traceability and transparency primarily focus on leveraging blockchain's fundamental principles to create a transparent and immutable record of supply chain activities. These methodologies involve the deployment of blockchain networks to record every transaction and product movement, ensuring data transparency. To tackle data privacy issues, existing approaches recommend employing encryption, off-chain storage, and data access controls to protect sensitive information. Smart contracts are commonly utilized to automate and enforce supply chain agreements, reducing manual intervention and enhancing transparency. Regulatory compliance is acknowledged as an essential aspect, with a focus on adapting to existing legal frameworks.
This system suggests using smart contracts and transaction validation rules to ensure secure information sharing. It also aims to build trust among supply chain partners by using a distributed ledger to store and authenticate transactions, offering flexibility and transparency. Traceability and Transparency are the key factors of Blockchain technology.

1.1 Overview

This project explores how blockchain technology can improve supply chain traceability and transparency across various industries. By analyzing real

world case studies, it highlights blockchain's potential to foster trust and enhance operations. The Textile and Clothing sector, facing challenges like opaque supply chains, can benefit from blockchain's ability to ensure transparent and secure traceability, addressing issues such as product recalls and labor concerns. Blockchain technology offers several features that make it particularly suitable for transactions:

- Decentralization: Blockchain operates on a decentralized network of computers (nodes) that collectively validate and record transactions. This removes the need for a central authority, reducing the risk of manipulation or control by a single entity.
- Immutability: Once a transaction is recorded on the blockchain, it becomes extremely difficult to alter or delete. Each block contains a cryptographic hash of the previous block, creating a chain of blocks that are interlinked and resistant to tampering.
- Transparency: All transactions on a blockchain are transparent and can be viewed by anyone with access to the network. This transparency helps to foster trust among participants and can reduce the risk of fraud.
- Security: Blockchain uses cryptographic techniques to secure transactions and ensure the integrity of the network. Transactions are verified by multiple nodes before being added to the blockchain, making it extremely difficult for malicious actors to manipulate the system.
- Smart Contracts: Smart contracts are self-executing contracts with the terms of the agreement directly written into code.They automatically execute and enforce the terms of the agreement when predefined conditions are met, eliminating the need for intermediaries and reducing transaction costs.

Chapter 2

2 System Analysis

2.1 Purpose

The purpose is to explore the potential of blockchain technology in enhancing supply chain traceability and transparency across various industries. It aims to inform stakeholders about the transformative impact of blockchain on supply chains, emphasizing its role in building trust and improving operations.
Supply chains are complex networks crucial for global operations, but they often lack transparency and traceability, leading to issues like disruptions and counterfeits, and ethical concerns. Blockchain technology, with its decentralized and immutable record-keeping, presents a solution. It also smart contracts and cryptographic techniques which enhances security.

2.1.1 Existing System

Blockchain's core features, including cryptographic equations and smart contracts, ensure data integrity and automate agreements within supply chains. Consensus mechanisms like Proof of Work (PoW) and immutable ledgers provide tamper resistance, addressing concerns over data manipulation. Existing methodologies focus on leveraging these principles to create transparent records of supply chain activities using blockchain networks. They tackle challenges such as scalability, interoperability, and data privacy through various techniques like sharding, encryption, and standardized formats. Smart contracts streamline processes and enhance transparency, while regulatory compliance remains a key consideration.
Overall, these methodologies aim to integrate blockchain effectively, considering scalability, interoperability, privacy and regulatory aspects to establish robust and transparent supply chains.

2.1.2 Proposed System

The proposed methodology builds upon blockchain's core features to address challenges in traditional supply chains. It emphasizes the creation of decentralized and immutable ledgers to record every transaction and product movement, ensuring transparency and data integrity. To enhance

scalability, techniques like sharding and sidechains are proposed, along with cross-chain interoperability standards. Privacy concerns are addressed through technologies like zero-knowledge proofs and secure multi-party computation. Smart contracts automate and enforce agreements, reducing manual intervention and disputes. Regulatory compliance is prioritized through collaboration with regulators and pilot projects to refine solutions within regulatory boundaries. We use SQLite3 as database.

Chapter 3

3 Feasibility Study

After the problem is clearly understood and solutions proposed, the next step is to conduct the feasibility study. Feasibility study is defined as evaluation or analysis of the potential impact of the proposed project or program. The objective is to determine whether the proposed system is feasible. There are three aspects of feasibility study to which the proposed system is subjected as discussed below.

3.1 Technical Feasibility

Technical feasibility evaluates whether the existing technical infrastructure is capable of supporting the implementation of blockchain solutions. It involves analyzing whether upgrades or modifications to current systems are necessary to accommodate blockchain integration. Additionally, it examines whether the proposed blockchain system can seamlessly integrate with the existing infrastructure without requiring significant hardware or software changes.

3.2 Economical Feasibility

Economic feasibility focuses on determining whether the required resources, including time and financial investments, are available to develop and deploy the blockchain-based supply chain solution. It entails assessing the cost effectiveness of implementing blockchain technology compared to traditional supply chain management approaches. Since blockchain implementation typically does not require additional hardware investments and offers cost saving benefits in the long run, it is deemed economically feasible.

3.3 Operational Feasibility

Operational feasibility evaluates whether the organization possesses the necessary human resources and capabilities to operate and maintain the blockchain based supply chain solution effectively. It assesses the training requirements for personnel and determines whether the system can be easily adopted and used by stakeholders. Given that the blockchain solutions aim to simplify processes and enhance transparency, minimal additional training is typically required for users familiar with

internet usage and the English language. Moreover, the organization already possesses the necessary resources for implementation, further enhancing operational feasibility.

Chapter 4

4 Software Requirement Specification

4.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the 'Traceability and Transparency in Supply Chains with Blockchain Technology'. It illustrate the purpose and complete description for the development of the system.It explains system constraints, interface and interactions with other external applications. This document is primarily intended to be proposed to make supply chains more transparent and traceable.

4.2 Scope

Our project has made it easier to increase the traceability and transparency of supply chains using blockchain technology. It has helped to increase the trust among consumers towards supply chains.

4.3 Overall Description

Our project emphasizes the importance of blockchain in addressing challenges like scalability, interoperability, data privacy, and regulatory compliance in supply chain management. It also highlights real - world implementations and the transformative impact of blockchain on improving operational efficiency, trust among stakeholders, and compliance with regulatory frameworks. It suggests future research directions, including integrating blockchain with emerging technologies like AI, IoT, and big data analytics to further enhance supply chain management.

4.3.1 Product Perspective

Our project is mainly used to build the trust among different stakeholders and suppliers so we can make the supply chain market boost its way through the economy.

4.3.2 Product Functionality

Through this system we can trace the payment details in our website. It will help in making our system more transparent.

4.3.3 Users and Characteristics

There are two users - admin and user. The admin can regulate everything in the system and they can add new new products. The users or the customers can purchase anything from the website and can make the payments through the wallet installed in it.

4.4 Specific Requirements

4.4.1 Hardware Requirements

- System: LAPTOP-OE7M0PNN
- Processor: 12th Gen Intel(R) Core(TM) i5-1235U
- Speed: Above 1GHz
- RAM capacity: 8 GB
- Hardisk drive: 212 GB

4.4.2 Software Requirements

- Operating System:Windows
- Frontend: React js
- Backend: Python
- Database : SQLite3
- Technologies used: HTML,Javascript

4.5 Functional Requirements

It contains four main modules.

- 1.Home
- 2.API
- 3.User
- 4.Product

Home

Home module provides top-level menus where visitors can go deeper into various areas of the site. It contains cart, products,etc. Home module contains the admin who manages the system.

API

The Module API provides functions that return information about the current operating environment (module, version, and instance). The Modules API also has functions that retrieve the address of a module, a version, or an instance. It also sends API calls to the frontend and backend.

User

The user module allows users to register, log in, and log out. Users benefit from being able to sign on because this associates content they create with their account and allows various permissions to be set for their roles. User can also interact with the system, select product, make payment etc.

Product

The Products module is a catalogue of the products and their descriptions. The product module contains all the products, details about the products,updation etc.

4.6 Non Functional Requirements

Non-functional requirements define the overall qualities or attributes of the resulting system. Non-functional system place restrictions on the product being developed, the developed process, and specify external constraints that the product must meet. Examples of non-functional requirements include safety, security, usability, reliability and performance requirements. Project management issues (costs, time and schedule) are often considered as non-functional requirements. The principal non - functional constraints which are relevant to critical systems :

- performance
- security
- safety

- usability

Performance

Performance requirements concern the speed of operation of a system. Types of performance requirements :

- Response requirements (how quickly the system reacts to a user input).
- Throughput requirements (how much the system can accomplish within a specified amount of time).
- Availability requirements (is the system available for service when requested by end users). The speed of operation of this system is adequate for the requirements.

Reliability

- Reliability is the ability of a system to perform its required function under stated conditions for a specified period of time.
- constraints on the runtime behavior of the system. This system is reliable because its functionalities can be done on the required conditions.

Safety

Safety requirements are not required which exclude unsafe situation from the possible solution of the system.

Usability

Usability is the ease with which a user can learn to operate, prepare inputs for, and interpret outputs of system or components. Usability requirements include :

- information error messages.
- well-formed user interfaces.

4.7 Interface Requirements

4.7.1 Hardware interfaces

The system must run over the internet, all the hardware shall require to connect internet will be hardware interface for the system. As for example modem, WAN-LAN.

4.7.2 Software interfaces

Software interface required for the working of the project is the appropriate operating system.

4.7.3 Communication interfaces

The user of the site communicate using the network connectivity and the data set is accessed.

4.8 Security Requirements

- User accesses only their account.
- Validation of input is handled.
- This application containing the computer systems is physically secured against arms or surreptitious entry by intruders.
- Users must be authorized carefully to reduce changes of any such user giving access to an intruder in exchange for a bribe or other favour.

4.9 Platform Used

Windows 11 is the latest version of Microsoft's operating system, succeeding Windows 10. It brings a redesigned user interface with a centered Start menu, improved window management features like Snap layouts and virtual desktops, enhanced gaming capabilities with DirectStorage and Auto HDR, and better integration with Microsoft Teams. It also introduces support for Android apps through the Microsoft Store and emphasizes security with features like Secure Boot, TPM 2.0, and enhanced protection against ransomware attacks. Overall, Windows 11 aims to provide a more modern, streamlined, and secure computing experience for users.

4.10 Technologies Used

Python

Python is an interpreted, object-oriented, and high-level programming language with dynamic semantics. Its built-in data structures, combined with dynamic typing and dynamic binding, make it attractive for Rapid Application Development. Additionally, Python serves as a scripting or glue language to connect existing components together. Python is a powerful, readable, and versatile language used in various domains, including machine learning, web development, and data analysis. Its ease of use and extensive standard library contribute to its popularity and widespread adoption.

JavaScript

JavaScript is a versatile programming language commonly used for web development. React, a popular JavaScript library, is used for building user interfaces, particularly single-page applications. React allows developers to create reusable UI components and efficiently manage the state of an application. It uses a virtual DOM for optimal performance, updating only the necessary parts of the UI when changes occur. React is often combined with other libraries and frameworks, such as Redux for state management and React Router for navigation, to create complex web applications.

HTML

HTML (Hypertext Markup Language) is the standard markup language for creating web pages and web applications. It provides a structure for content on the web by using a system of tags and attributes to define elements such as headings, paragraphs, links, images, and forms. HTML documents are interpreted by web browsers to render the visual presentation of web pages. HTML is an essential building block of the web and is often combined with CSS (Cascading Style Sheets) for styling and JavaScript for interactivity to create dynamic and visually appealing websites.

SQLite3

SQLite 3 is a lightweight, embedded relational database management system (RDBMS) that is widely used in various applications due to its simplicity, efficiency, and portability. Developed as a self-contained, serverless, and zero-configuration database engine, SQLite 3 is designed to be seamlessly integrated into applications, making it a popular choice for embedded

systems, mobile devices, and desktop software. Its compact nature doesn't compromise on capabilities, as it supports standard SQL features and provides reliable ACID transactions. Additionally, SQLite 3's open-source nature and cross-platform compatibility contribute to its widespread adoption across a diverse range of software development projects. Whether powering mobile apps, web browsers, or desktop applications, SQLite 3's ease of use and robust performance make it a versatile and dependable database solution.

Chapter 5

5 Design Document

5.1 Purpose

For our system, we created a demo website for implementing the blockchain technology in transactions. We came up with a system where we implement blockchain technology in supply chains to make it more traceable and transparent. We came up with a system and implemented it in a demo website in the payment section where blockchain is implemented in transactions, which helps in making the transactions more transparent and trustworthy.
Purpose of this document is to give the detailed description of the architecture and system design for the software.The system design document shows the software will be structured to satisfy the requirement identified in the software requirement specification.It is a translation of requirements into a description of the software structure,software component, interfaces and data necessary for the implementation phase.In a complete software design document,each requirements must be traceable to one or more design entities.

5.2 Scope

The demo website implemented in our system can be adopted by many retail companies and it will help boost trust in consumers and stakeholders.

5.3 Overview

The purpose of this document is to help the reader to visualize the solution to the project presented. This document verifies how the design meet the requirement stimulated in the SRS document through design viewpoints. The design viewpoints will cover all design elements presented before.
This document will provide a direct approach to the development of this project hence reducing feature creep and ponitedly determine the quality of the design.

5.4 Data Design

Database are the storehouses of data used in the software systems.The data is stored in tables inside the database.Several tables are created for the manipulation of the data for the system.Two essential settings for a database are:

- Primary Key- The field that is unique for all the record occurrence.
- Foreign Key- The field used to set relation between tables.

Normalization

- Normalization is the process of organizing the data in the database.
- Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate the undesirable characteristics like Insertion, Update and Deletion Anomalies.
- Normalization divides the larger table into the smaller table and links them using relationship.
- The normal form is used to reduce redundancy from the database table.

Tables

User

Name	DataType	Constraints	Description
id	integer	Primarykey	ID of user
email	string	unique	Email of the users
username	string	unique	Username of user
firstName	string	Notnull	Firstname of user
lastName	string	Notnull	Lastname of user
dateJoined	date	Notnull	Date of join
lastLogin	date	Notnull	Date of last login
password	string	Notnull	Pasword of user
Mobile No.	integer	Notnull	phone number of user
companyName	string	Notnull	Name of the company

Products

Name	DataType	Constraints	Description
id	string	Primarykey	ID of product(auto generated)
name	string(60)	Notnull	Name of product
price	float	Notnull	Price of product
description	string(200)	null	Description of product
image	string(255)	null	Image of product
Warranty_period	integer	Default[0]	period of warranty
created_at	date	Notnull	Date of manufacture
modified_at	date	Notnull	Date of modification
user_id	primary	Foriegn	id from users table

Cart

Name	DataType	Constraints	Description
id	string	Primarykey	cart id (auto generated)
product_id	string	Foreignkey	id from products table
quantity	integer	Default	No. of items in cart
date_added	date	Notnull	date the item was added
user_id	string	Foriegn key	id from user table

NFT details

Name	DataType	Constraints	Description
id	string	Primarykey	auto generated
product_id	string	foriegn key	id from products table
token_url	string(200)	Default[' ']	url of the tokens
expiry_date	date	Default[' ']	expiry date of token
user_id	string	Foriegn key	id from user table
token_id	integer	Unique, Default	id of token
acc_address	string(300)	Default[' ']	Address of the account
redeem	boolean	Default[' TRUE ']	only returns true

Order items

Name	DataType	Constraints	Description
id	string	Primarykey	Auto generated
product_id	string	Foreignkey	id from product table
quantity	integer	Notnull	No. of items ordered
order_date	date	Notnull	date the item was ordered
user_id	string	Foreign key	auto generated

Payment details

Name	DataType	Constraints	Description
id	string	Primarykey	Auto generated
orders_id	string	Foreignkey	id from order items
total_amount	float	Notnull	total amount to be paid
status	boolean	Default[' TRUE ']	status of payment
payment_date	date	Notnull	Date of payment

Order details

Name	DataType	Constraints	Description
id	string	Primarykey	Auto generated
payment_id	string	Foreignkey	id from payment details

Shipping Address

Name	DataType	Constraints	Description
id	string	Primarykey	auto generated
user_id	string	Foriegn key	id from user table
address	string	Notnull	address to be shipped to
city	string	Notnull	city to be shipped to
state	string	Notnull	state to be shipped to
postal_code	integer	Notnull	postal code to be shipped to
date_added	date	Notnull	date to be shipped

Track Repairs

Name	DataType	Constraints	Description
id	string	Primarykey	auto generated
orders_id	string	Foreign key	id from order items
description	string	null	Description of repairs
percent_work_done	float	Notnull	Percentage of repair done
Details_id	string	Foriegn key	Details about payment and order

Chapter 6

6 Development of the System

This system can be decomposed into many number of submodules. The sub modules of our system 'Unlocking Traceability and Transparency in Supply Chains with Blockchain Technology' are home, API, product and user. Each sub module have specific objectives,to avoid unwanted coupling between modules and to increase cohesion, modules are again decomposed into submodules. We have also implemented a demo website to use this system.

Chapter 7

7 System Testing

Testing is the penultimate step of software development.An elaborate testing of the data is prepared and the system is using the test data,while doing testing, errors are noted and correction is made.The users are trained to operate the developed system.Both hardware and software securities are made to run the developed system successfully.System testing is aimed at ensuring the system works accurately before the live operation commences.

7.1 Test Plan

7.1.1 Scope

This test plan will cover the following testing activities as identified in the testing strategy.

- White Box Testing

White box testing is a testing technique, that examines the program structure and derives test data from the program logic/code. In white box testing, the UI is bypassed. Inputs and outputs of ratings are tested directly at the code level and the results are compared against specifications. It reveals errors in "hidden" code.

- Black Box Testing

Black box testing is a method of software testing that examines the functionality of an application based on the specifications. Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end - user would.

- Unit Testing

The module includes four main components to undergo unit testing. It checks the syntax error, logic error and validity of the program correctness. This test will be performed by the developers.

- Integration Testing

After all components pass the unit testing, they get to be tested whether they work correctly when they are running concurrently and communicating to each other. Integration tests exercise an entire subsystem and ensure that a set of components play nicely together.

- Internal Data Testing

We will test the validity of the data before it enters the database to avoid any problems that may face us in the database. We will test the encryption of the personal information of all the users along with the user names and passwords to ensure maximum security of the users privacy.

7.1.2 Software risk issues

In this section, the plan is to test the risk involved in critical issues such as:

- Difficult to run on visual studio due to large loading time
- Some inherent software risks such as complexity exist; also these issues need to be identified.
- Proper network connection
- Working of SQLite3 database

7.1.3 Features to be tested

- Test whether correct user name and password allows you to login.
- Test whether invalid user name and password prevents you from login.
- Test whether there is any connection problem in the Server.
- Test whether invalid data entry allows saving data successfully.
- Test whether products are added to cart successfully.
- Test whether all the data entered are valid.
- Test whether all pages are loaded correctly.
- Test whether the provided security works correctly.

7.2 Test consolidation

7.2.1 Test item

The items or features to be tested in the test cases are included in the document. Each and every user input is tested. The present condition of the system is tested. It is checked to make sure that environment is ready for the application to work.

7.2.2 Input specifications

These are the inputs required to execute the test case. All required inputs including data elements and values are given below.

Data	Values
username	A valid username
Password	Combination of characters and numbers

Chapter 8

8 System Implementation and Maintenance

8.1 Implementation

System implementation is the conversion of new system into an operating one which involves creating compatible files, training clients and installing hardware. User training is crucial for minimizing resistance to change and giving chance to prove its worth. Training aids user friendly manuals and healthy screens provide the user with a good start. Software maintenance follows conversion to the extent that changes are necessary to maintain satisfactory operations relative to changes in the user's environment. Maintenance often includes minor enhancements or corrections to the problem that surface late in the systems operations.
In the implementation phase, the team builds the components either from scratch or by composition. Given the architecture document meant from the design phase and the requirement document from the analysis phase, the team should build exactly what has been requested, though there is still room for innovation and flexibility. For example, a component may be narrowly designed for this particular system, or the component may be made more general to satisfy a reusability

- Careful planning
- Investigation of system and constraints
- Design the methods to achieve changeover.
- Training the staff in the changed phase.
- Evaluation of change over method.
- The method of implementation and time scale to be adopted are found out initially.

8.2 Maintenance

This phase occurs as a result of deploying the whole system at the end users organization. They will perform the beta testing at the end users and inform to the developers about any needed modification to the application. The customer records all the problems that are encountered during the beta testing and reports these to the developer at regular intervals.

8.2.1 Corrective Maintenance

Even with the best quality assurance activities, it is likely that the customer will uncover defects in the software. Corrective maintenance changes the software to correct the defects. Corrective Maintenance activity may consist of repair, restoration or replacement of equipment. This activity will be result of a regular inspection, which identifies the failure in time for corrective maintenance to be planned and scheduled, then performed during a routine maintenance shutdown.

8.2.2 Adaptive Maintenance

Over time, the original environment(CPU, operating system, business rules, external product characteristics) for which the software was developed is likely to change. Adaptive maintenance results in modification to the software to accommodate changes to its external environment.

8.2.3 Enhanced Maintenance

As software is used, the customer/user will recognize additional functions that will provide the benefit. Perfect maintenance extends the software beyond its original functional requirements.

8.2.4 Preventive Maintenance

Computer software deteriorates due to change, and because of this preventive maintenance often called software re-engineering, must be conducted to enable the software to serve the needs of its end users. Preventive maintenance makes changes to computer programs so that they can be more easily corrected, adapted and enhanced.

Chapter 9

9 Conclusion and Future Scope

9.1 Conclusion

This system explores blockchain's impact on enhancing supply chain traceability and transparency, highlighting its transformative potential through real-world case studies. It acknowledges challenges like scalability, data privacy, and regulatory compliance. Ongoing research focuses on scalability solutions, interoperability standards, and privacy-enhancing technologies. Collaboration among academia, industry, and regulators is crucial. Blockchain's integration with AI/ML holds promise for optimizing supply chain processes. Additionally, advancements in cross-chain interoperability, energy-efficient consensus mechanisms, and integration with emerging technologies like IoT, big data analytics, and edge computing are key for improving supply chain management.

9.2 Future Scope

- Integration of DL with Blockchain
- Emerging Technology Integration

Chapter 10

10 References

1. Chen, X., Wang, L., & Wang, L. (2019). Blockchain-based Supply Chain Traceability: A Systematic Review. IEEE Transactions on Industrial Informatics, 15(8), 4608-4619.

2. Durst, S., Klußmann, S., & Arndt, H. (2021). Supply Chain Management 4.0: A Comprehensive Survey and Blockchain-Based Case Study. Sustainability, 13(6), 3210.

3. Idowu, A. (2020). Enhancing Food Traceability in the Supply Chain with Blockchain: A Case Study of Walmart. International Journal of Information Management, 55, 102177.

4. Kim, H. M., Laskowski, M., & Eksioglu, S. (2021). Enhancing Supply Chain Transparency with Blockchain Technology: A Case Study of Maersk's TradeLens Platform. International Journal of Logistics Management, 32(2), 274-291.

5. Marcella, R., Zimbra, D., & Micale, R. (2020). Blockchain for Ethical Sourcing in the Diamond Industry: A Case Study of De Beers' Tracr. Resources Policy, 68, 101814.

6. Mettler, M. (2016). Blockchain Technology in Healthcare: The Revolution Starts Here. IEEE Pulse, 8(4), 35-38.

7. Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. Retrieved from https://bitcoin.org/bitcoin.pdf

8. Kim, Y., & Laskowski, M. (2018). "Toward an ontology-driven block chain design for supply chain provenance." IEEE Transactions on Engineering Management, 65(4), 545-558.

9. Garg, S., & Baweja, K. (2019). "Blockchain-based supply chain management for sustainable business practices." Production Planning & Control, 30(11-12), 964-979.

10. Qu, W., Wu, B., & Wang, H. (2020). "A blockchain-based quality traceability system for agri-food supply chain." Journal of Food Engineering, 289, 110187.

11. Yao, L., Liu, Y., & Fan, S. (2019). "A blockchain-based supply chain quality management framework: A case study of an online retailer." International Journal of Production Research, 57(7), 2045-2063.

12. Tapscott, D., & Tapscott, A. (2016). "Blockchain revolution: how the technology behind bitcoin is changing money, business, and the world." Penguin.

13. Kshetri, N. (2018). "Will blockchain emerge as a tool to break the poverty chain in the Global South?" Third World Quarterly, 39(8), 1439-1458.

14. Al-Mansoori, A., & Al-Fuqaha, A. (2019). "Blockchain technology for social impact: Opportunities and challenges ahead." IEEE Transactions on Technology and Society, 1(1), 42-47.

15. Li, X., & Liang, G. (2017). "Blockchain-based system for secure data storage with private keyword search." IEEE Transactions on Services Computing, 11(5), 817-831.

16. Mengelkamp, E., Notheisen, B., & Weinhardt, C. (2018). "A blockchain based smart grid: towards sustainable local energy markets." Computer Science-Research and Development, 33(1-2), 207-214.

17. Li, W., & Mannan, M. (2017). "Securing internet of things with lightweight blockchain." IEEE Cloud Computing, 4(4), 32-39.

18. Christidis, K., & Devetsikiotis, M. (2016). "Blockchains and smart contracts for the internet of things." IEEE Access, 4, 2292-2303.

19. Noor, A. K., Salim, S. S., & Shuja, J. (2020). "An IoT and blockchain based food traceability system: A case study of Saudi Arabian dates." IEEE Access, 8, 190207-190218.

20. Zheng, Z., Xie, S., Dai, H. N., Chen, X., & Wang, H. (2018). "Blockchain challenges and opportunities: A survey." International Journal of Web and Grid Services, 14(4), 352-375.

21. Park, S., Lee, K., Kim, S., & Kim, H. (2019). "Blockchain-based supply chain traceability system for food safety." Electronics, 8(6), 668.

22. Tse, D. (2021). "Blockchain-Based Supply Chain Management: A Case Study in the Automotive Industry." Journal of Open Innovation: Technology, Market, and Complexity, 7(1), 38.

23. Zhang, C., Xu, H., & Zhao, M. (2018). "A new era of mass data and blockchain Internet of Things." IEEE Transactions on Industrial Informatics, 15(3), 1430-1438.

24. Jiao, S., Zhang, J., & Jiang, J. (2019). "A blockchain-based supply chain quality management framework: A case study of online retailer." International Journal of Production Research, 57(7), 2045-2063.

25. Korpela, K., Hallikas, J., & Dahlberg, T. (2017). "Digital supply chain transformation toward blockchain integration." Production Planning & Control, 28(11-12), 929-944.

Appendix

A Data Flow Diagram

Data flow is the one of the best way of documenting the entire functionality of the system.For the system, which will have data flows in and have some processing inside and then some data flow out from the system can be documented or represented effectively by means of data flow out from the system can be documented or represented effectively by means of data flow diagrams. The data flow diagram are a diagrammatic representation of the system,which has input,process and output.Once any system is represented using a data flow diagram we can identify the following things easily:

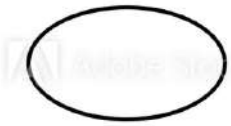
- Various entities interacting with the system are identified
- Flow of data from one entity to another is identified
- The various processes involved in between the interaction of two or more entities in the system are clearly pointed out
- The various data stores which hold the data in between the process,are clearly identified

Some Data Flow Diagram charting forms are given below:

A.1 External source or receiver

A source or sink is a person or part of organization ,which enters or receives information from the system,but is considered to be outlining the contest of data flow model.

A.2 Transform process

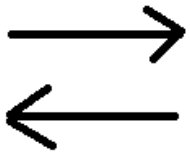


A process represents transformation where incoming data flows are changed into outgoing data flow.

A.3 Data Store

A data store is repository of data that is to be stored for use by one or more process may be as simple as buffer or queue or sophisticated as relational database. They should have clear names.If a process merely uses the contest of store and does not alter it,the arrowhead goes only form the store to the process. If a process alters the details in the store then double headed arrow is used.

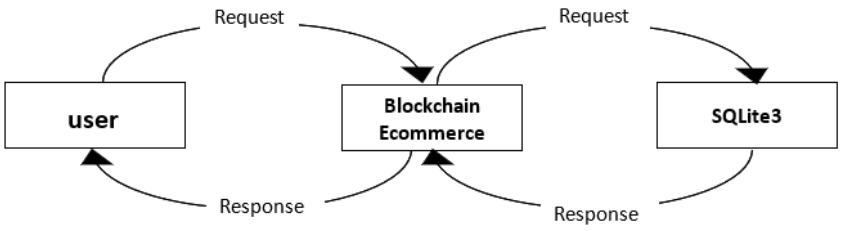
A.4 Data flow



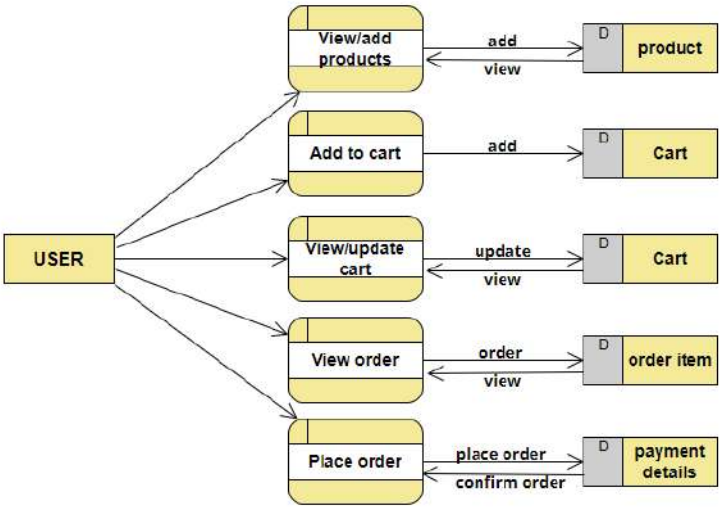
A data flow is a route, which enable packets of data to travel from one point to another.Data may flow,with arrowhead pointing in the direction of the flow.

A.5 Diagrams

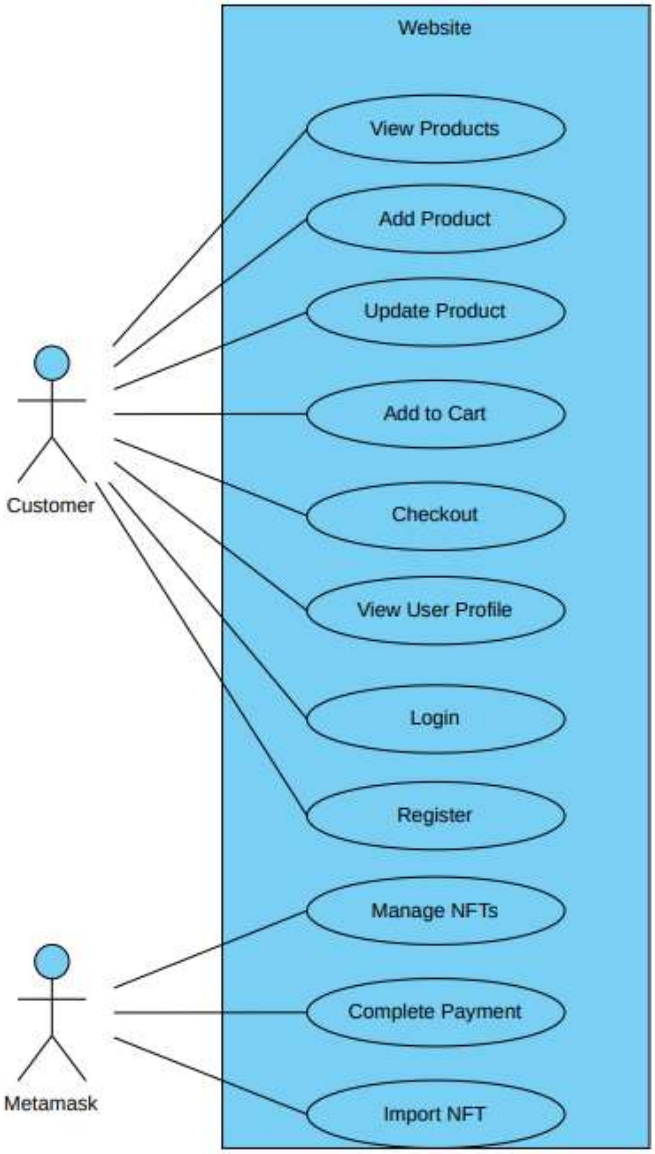
A.5.1 Level 0



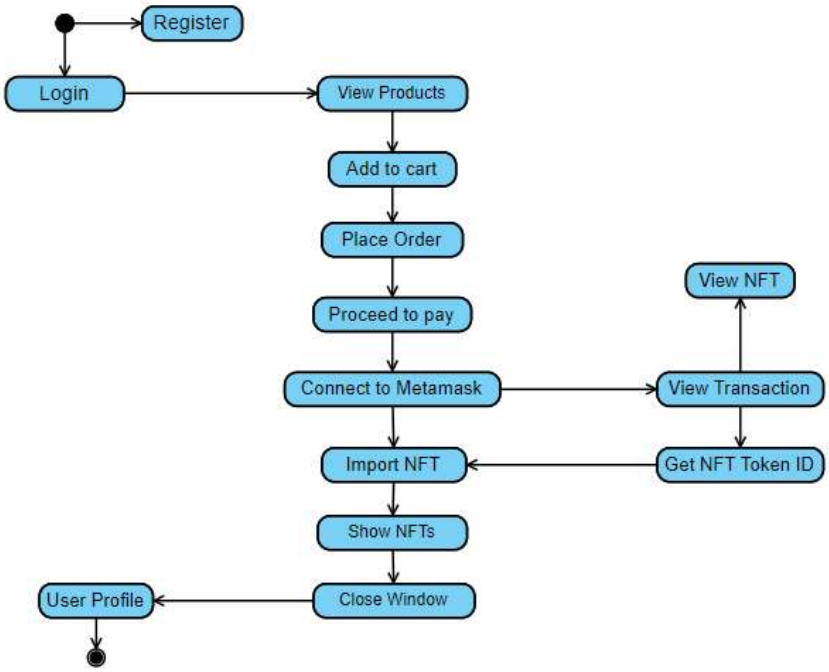
A.5.2 Level 1


USER
View/add products
Add to cart
View/update cart
View order
Place order
add
view
add
update
view
order
view
place order
confirm order
D
product
Cart
Cart
order item
payment details

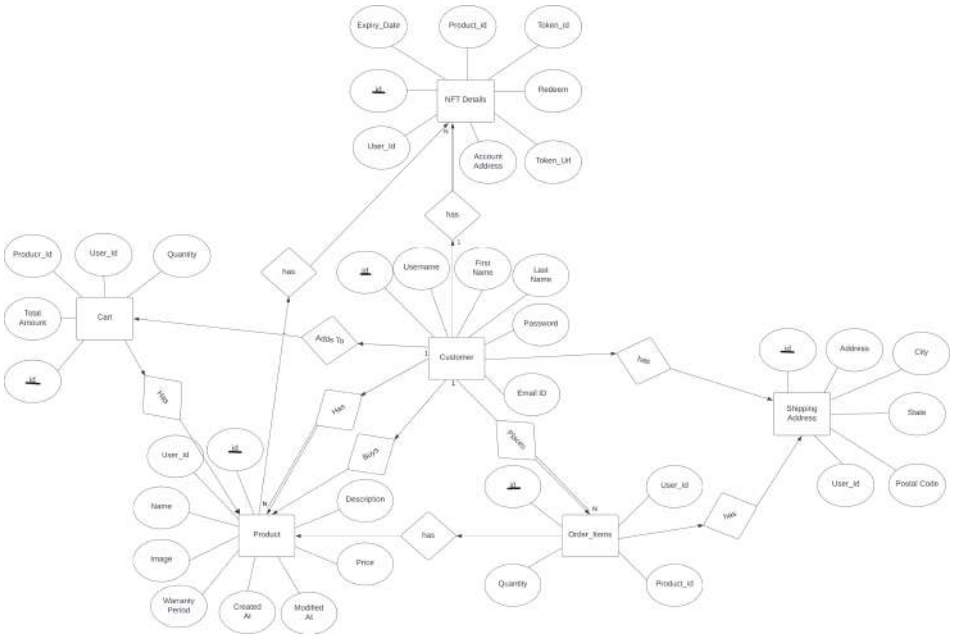
B Use Case Diagram



C Activity Diagram



D ER Diagram


Expiry_Date
Product_id
Token_id
NFT Details
Redeem
User_Id
Account Address
Token_Uri
has
Producr_id
User_Id
Quantity
Total Amount
Cart
Adds To
Customer
Username
First Name
Last Name
Password
Email ID
has
Address
City
State
Shipping Address
User_Id
Postal Code
Has
Has
Buys
Places
User_Id
Name
Description
Image
Price
Warranty Period
Created At
Modified At
Product
has
Order_Items
User_Id
Product_id
Quantity
has

E Front Page Of The IEEE Paper Published

2023 3rd International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON)
Karnataka, India. Dec 29-31, 2023

Unlocking Traceability and Transparency in Retail Supply Chains with Blockchain Technology

Devana Pramod
Department of Computer Science,
Christ college (Autonomous),
Irinjalakuda, Kerala-680125

Hana Nasreen
Department of Computer Science,
Christ college (Autonomous),
Irinjalakuda, Kerala-680125

Liya Johnson
Department of Computer Science,
Christ college (Autonomous),
Irinjalakuda, Kerala-680125

Megna Biju
Department of Computer Science,
Christ college (Autonomous),
Irinjalakuda, Kerala-680125

Priyanga K.K
Department of Computer Science,
Christ college (Autonomous),
Irinjalakuda, Kerala-680125
priyasumesh111@gmail.com

***Abstract*: Traceability has emerged as a prime requirement for a multi-tier and multi-site production. It enables visibility and caters to the consumer requirements of transparency and quality assurance. Textile and clothing industry is one such example that requires traceability implementation to address prevailing problems of information asymmetry and low visibility. Customers find it difficult to access product data that can facilitate ethical buying practices or assure product authenticity. Besides, it is challenging for stakeholders to share crucial information in an insecure environment with risk of data manipulations and fear of losing information advantage. In this context, this study investigates and proposes a blockchain-based traceability framework for traceability in multi-tier textile and clothing supply chain. It conceptualizes the interaction of supply chain partners, and related network architecture at the organizational level and smart contract and transaction validation rules at the operational level. To illustrate the application of the proposed framework, the study presents an example of organic cotton supply chain using blockchain with customized smart contract and transaction rules. It finally demonstrates the applicability of the developed blockchain by testing it under two parameters. The proposed system can build a technology-based trust among the supply chain partners, where the distributed ledger can be used to store and authenticate supply chain transactions. Further, the blockchain-based traceability system would provide a unique opportunity, flexibility, and authority to all partners to trace-back their supply network and create transparent and sustainable supply chain.**

Keywords: Blockchain, Traceability, Manufacturing, Textile and Clothing, Information sharing, Supply chain

I. Introduction

Supply chains are intricate systems critical to global operations, but they often lack transparency and traceability, leading to issues like disruptions, counterfeits, and ethical concerns. Blockchain technology, known for its decentralized and immutable record-keeping, offers a solution. This paper explores blockchain's potential in enhancing supply chain traceability and transparency across industries like manufacturing, healthcare, food, and luxury goods. [1] It analyses real-world case studies, discusses benefits and limitations, and aims to inform businesses, policymakers, and researchers about the transformative impact of blockchain on supply chains, emphasizing its role in building trust and improving operations.

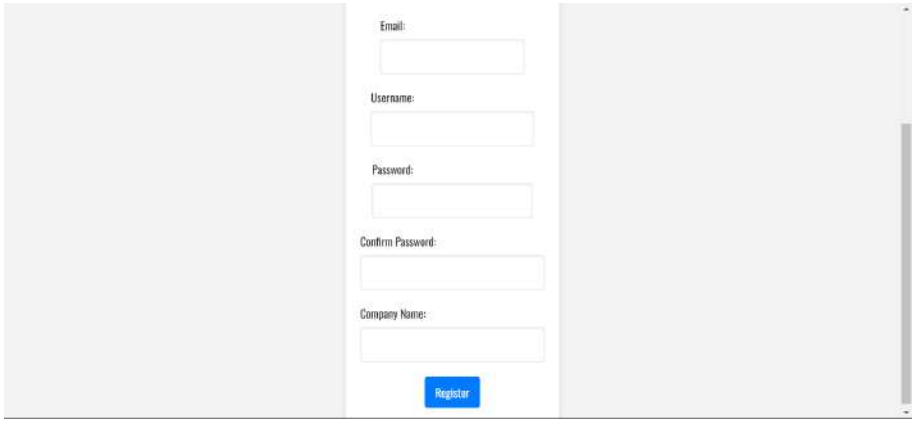
Industries worldwide are grappling with mounting pressure to transition towards sustainable production practices. The Textile and Clothing (T&C) sector, in particular, faces a unique set of challenges stemming from volatile consumer demands, intense competition, and the complexities of opaque supply chains. These issues have given rise to problems like product recalls and concerns over labour practices. [2] Enter blockchain technology, offering a robust solution. By facilitating transparent and secure traceability throughout the supply chain, blockchain not only enables better information sharing but also fosters trust among supply chain partners spread across the globe. What's noteworthy is that this blockchain-based approach is not limited to the T&C industry alone; with some adjustments, it can be applied to various other industries, making it a versatile tool for promoting sustainability and transparency across different sectors.

Additionally, existing methodologies for blockchain-based supply chain traceability and transparency primarily focus on leveraging blockchain's fundamental principles to create a transparent and immutable record of supply chain activities. These methodologies involve the deployment of blockchain networks to record every transaction and product movement, ensuring data transparency. [3] However, challenges related to scalability are often addressed through various consensus mechanisms and data partitioning techniques. While interoperability remains a concern, some methodologies suggest the use of standardized data formats and interoperability frameworks to facilitate communication between disparate blockchain networks. To tackle data privacy issues, existing approaches recommend employing encryption, off-chain storage, and data access controls to protect sensitive information. Smart contracts are commonly utilized to automate and enforce supply chain agreements, reducing manual intervention and enhancing transparency. Regulatory compliance is acknowledged as an essential aspect, with a focus on adapting to existing legal frameworks. [3] Some methodologies also propose conducting pilot projects to navigate regulatory challenges and refine blockchain-based solutions within regulatory boundaries. Overall, existing methodologies aim to create robust and transparent supply chains by integrating blockchain technology while considering scalability, interoperability, privacy, and regulatory aspects.

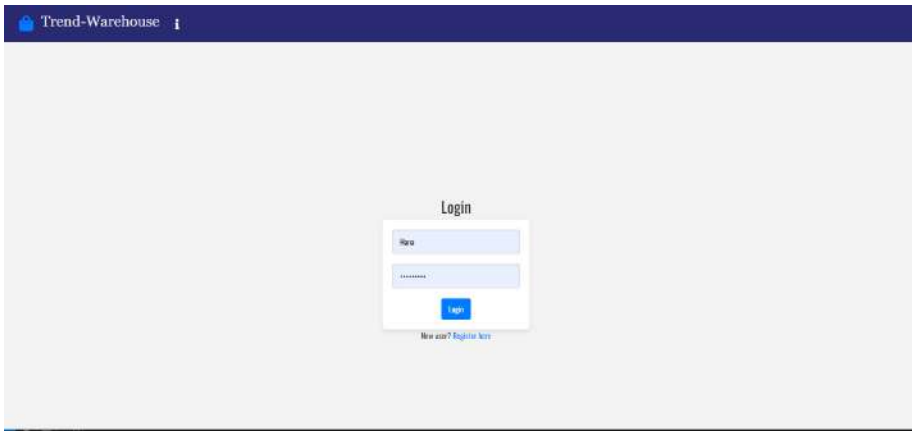
979-8-3503-1912-5/23/$31.00 ©2023 IEEE

F USER INTERFACES

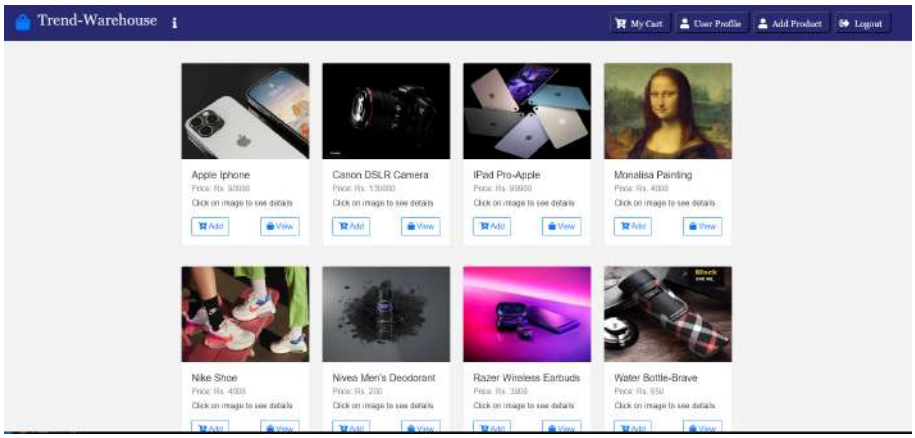
F.1 REGISTRATION



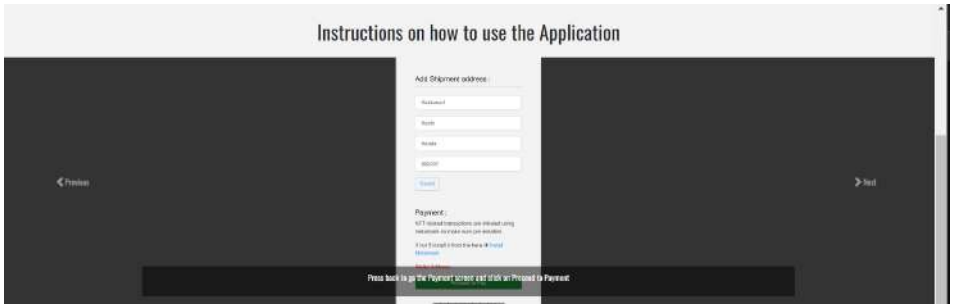
F.2 LOGIN



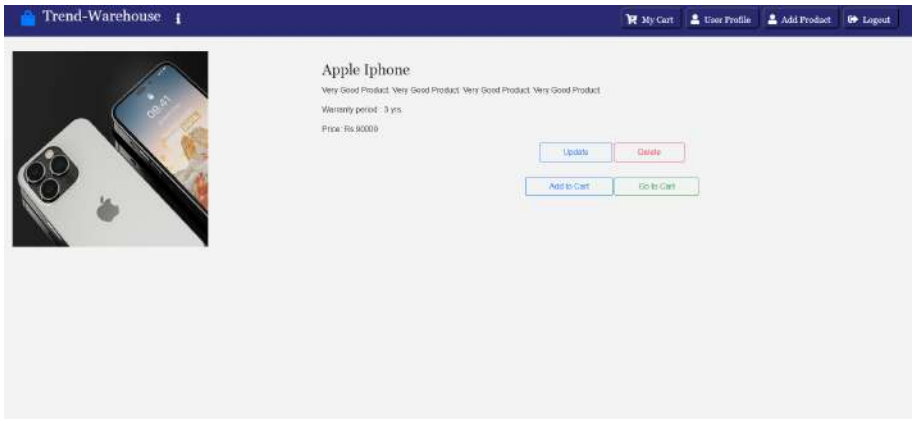
F.3 HOME


Trend-Warehouse
My Cart
User Profile
Add Product
Logout
Apple Iphone
Canon DSLR Camera
IPad Pro-Apple
Monalisa Painting
Nike Shoe
Nivea Men's Deodorant
Razer Wireless Earbuds
Water Bottle-Brave

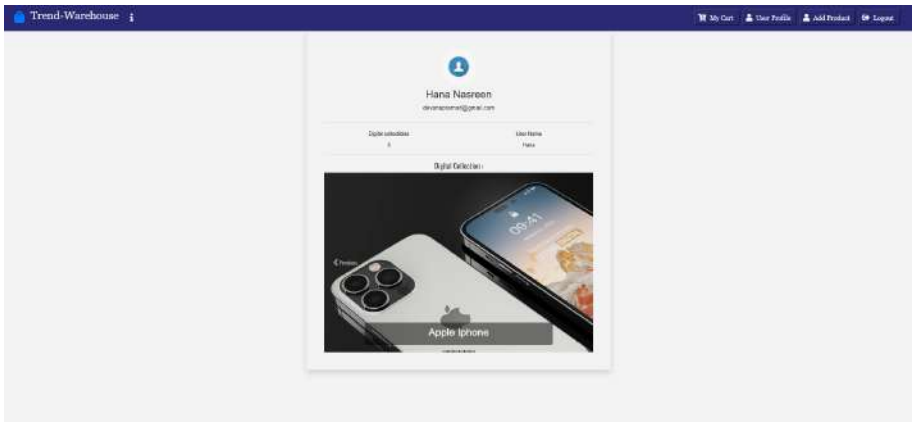
F.4 INSTRUCTIONS



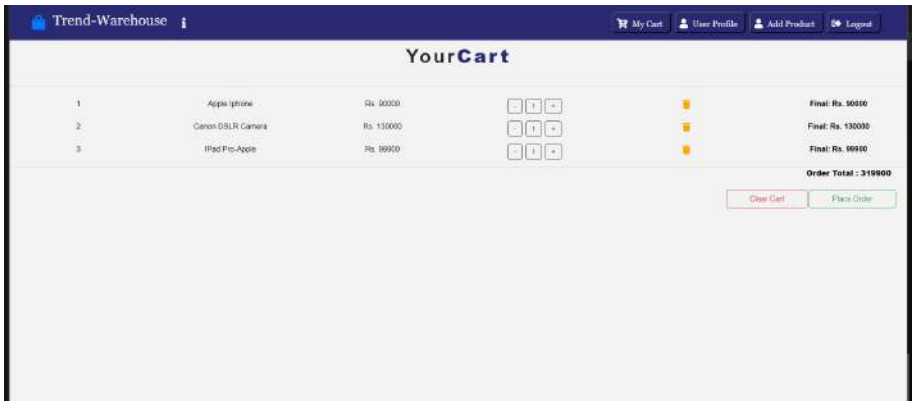
F.5 PRODUCT DESCRIPTION



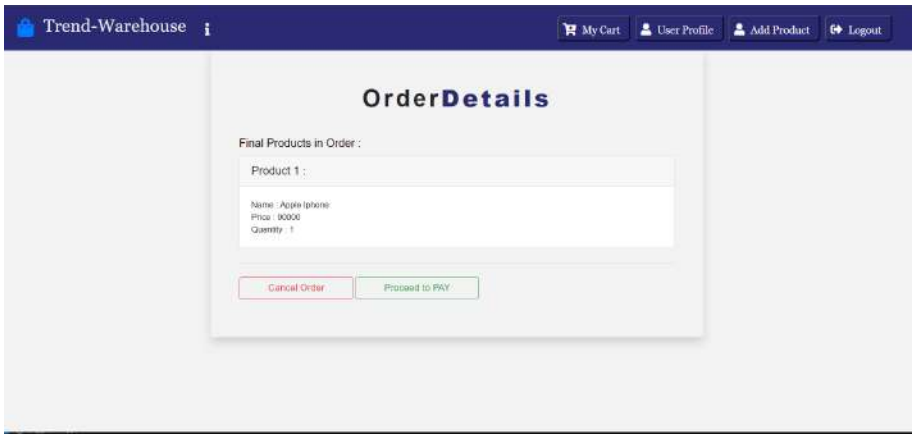
F.6 USER PROFILE



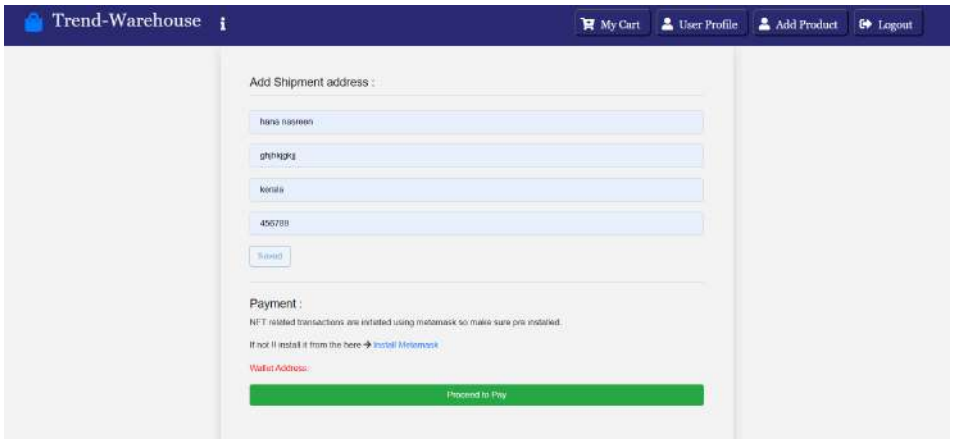
F.7 CART DETAILS

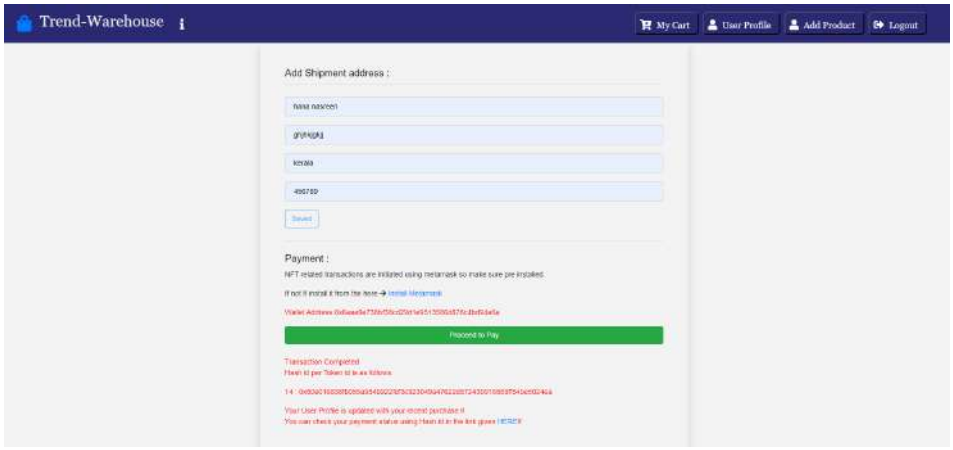


F.8 ORDER DETAILS

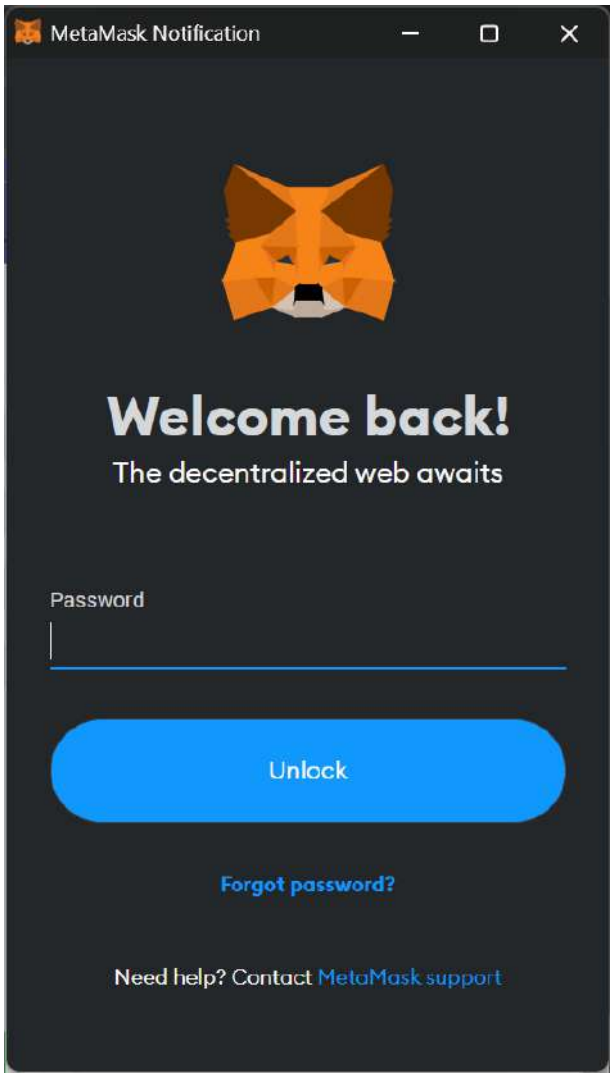


F.9 PAYMENT

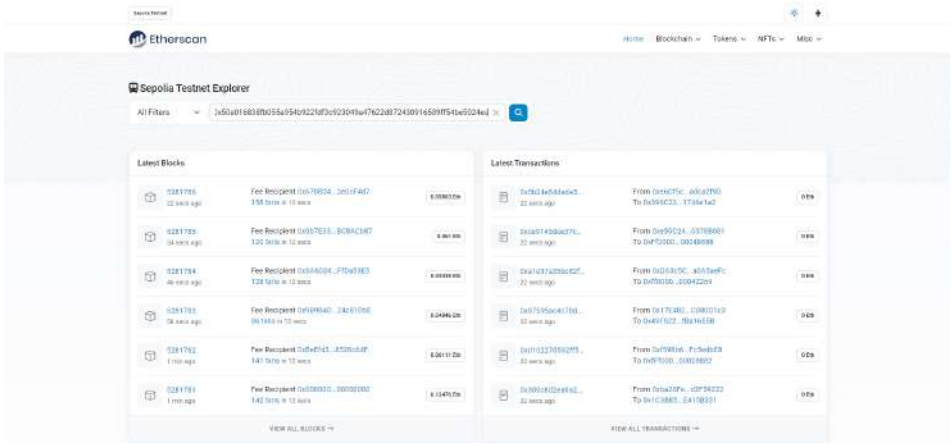

Trend-Warehouse
My Cart
User Profile
Add Product
Logout
Add Shipment address :
hana nasreen
ghhkgkg
kerala
456789
Saved
Payment :
NFT related transactions are initated using metamask so make sure pre installed.
If not ll install it from the here → Install Metamask
Wallet Address:
Proceed to Pay

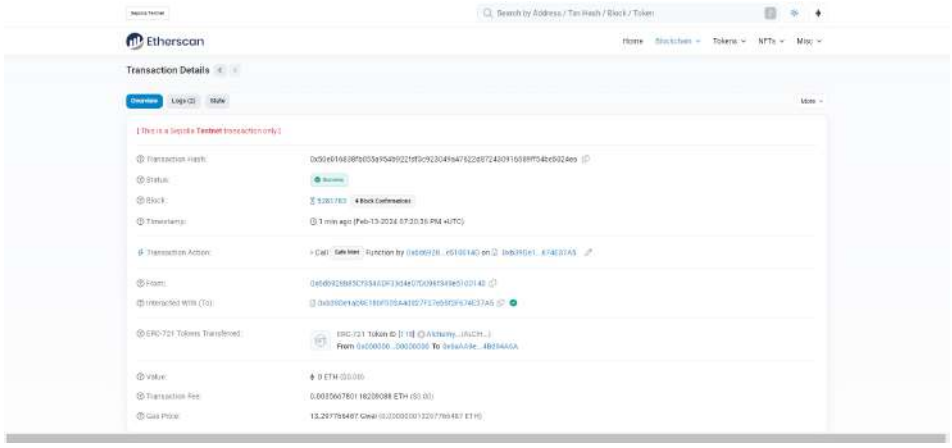

Trend-Warehouse
My Cart
User Profile
Add Product
Logout
Add Shipment address :
hana nasreen
ghhkgkg
kerala
456789
Saved
Payment :
Proceed to Pay

F.10 METAMASK

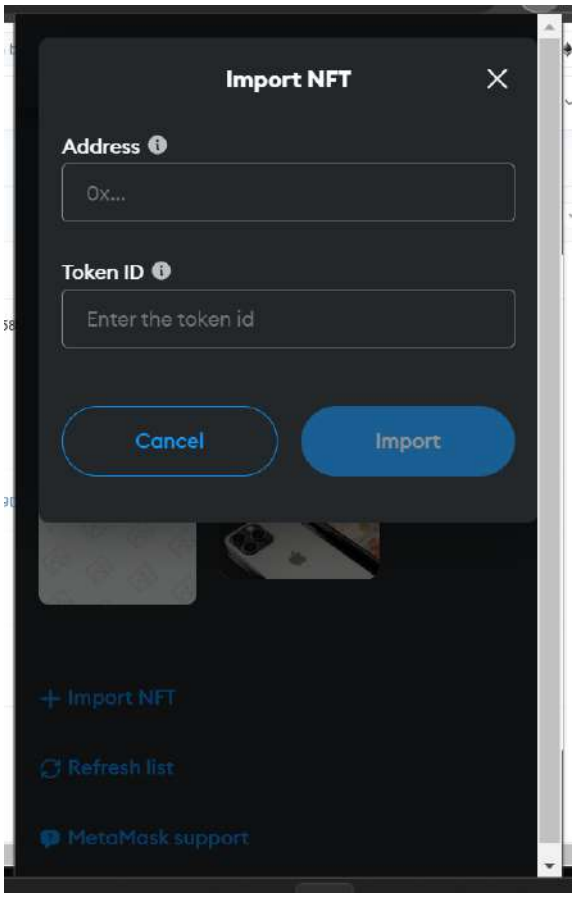


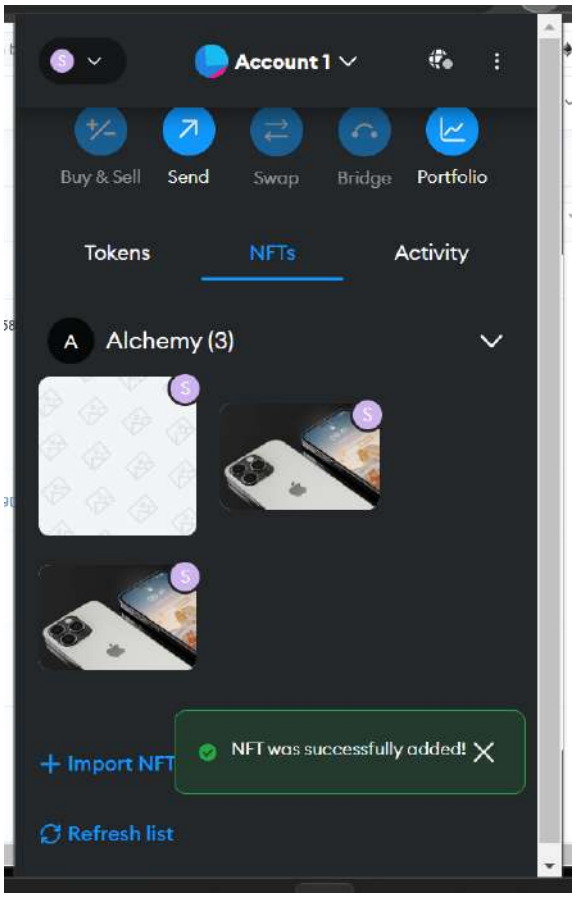
F.11 WALLET


Etherscan
Sepolia Testnet Explorer
Latest Blocks
Latest Transactions


Etherscan
Transaction Details

F.12 NFT DETAILS





G CODE

models.py

```
from django.db import models
from django.contrib.auth.hashers import make_password, check_password
from home.models import Account


class User(models.Model):
    User_type = (
        ('R', 'Retailer'),
        ('C', 'Customer'),
    )
    email = models.EmailField(max_length=60, unique=True)
    username = models.CharField(max_length=30, unique=True)
    first_name = models.CharField(max_length=30, null=True)
    last_name = models.CharField(max_length=30, null=True)
    date_joined = models.DateTimeField(auto_now_add=True)
    last_login = models.DateTimeField(auto_now=True)
    password = models.CharField(max_length=30)
    mobile_no = models.IntegerField(blank=True, null=True)
    company_name = models.CharField(max_length=50, null=True)

    def __str__(self):
        return self.username

    def set_password(self, raw_password):
        self.password = raw_password

    def check_password(self, raw_password):
        return raw_password==self.password


def get_product_image_filepath(self, filename):
    return 'product_images/' + str(self.pk) + '/product_image.png'


class Product(models.Model):
```

```
    name = models.CharField(max_length=60)
    price = models.FloatField()
    description = models.CharField(max_length=200, null=True)
    image = models.ImageField(max_length=255, upload_to=
                                        get_product_image_filepath,
         null=True, blank=True)
    warranty_period = models.IntegerField(default=0)
    created_at = models.DateTimeField(auto_now_add=True)
    modified_at = models.DateTimeField(auto_now=True)
    user_id = models.ForeignKey(User, on_delete=models.CASCADE, default=3)

    def get_product_image_filename(self):
        substring = f'product_images/{self.pk}/'
        if substring in self.image:
            result = self.image[str(self.image).index(substring):]
        else:
            result = None  # or some other default value or action

        return result

    def __str__(self):
        return self.name


class Order_Items(models.Model):
    product_id = models.ForeignKey(Product, on_delete=models.CASCADE)
    user_id = models.ForeignKey(User, on_delete=models.CASCADE)
    quantity = models.IntegerField()
    order_date = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return str(self.user_id)


class Payment_Details(models.Model):
    order_id = models.ForeignKey(Order_Items, on_delete=models.CASCADE)
    total_amount = models.FloatField()
    status = models.BooleanField(default=True)
    payment_date = models.DateTimeField(auto_now_add=True)
```

```
    def __str__(self):
        return str(self.status)


class Order_Details(models.Model):
    payment_id = models.ForeignKey(Payment_Details, on_delete=
        models.CASCADE)

    def __str__(self):
        return str(self.payment_id)
```

views.py

```
from django.shortcuts import render
from rest_framework import status, generics
from rest_framework.permissions import AllowAny
from .models import User
from .serializers import *
```

urls.py

```
from django.urls import path,include
from . import views
from rest_framework import routers

urlpatterns = [
    path('product/',views.product_list),
    path('product/<int:id>/add/',views.AddProduct),
    path('product/<int:id>/',views.product_detail),
    path('product/<int:rid>/<int:pid>/update/',views.update_product_detail),
    path('product/<int:id>/delete/',views.delete_product),
    path('cart/<int:id>/',views.cart_list),
    path('cart/<int:uid>/<int:pid>/delete/',views.remove_item),
    path('cart/<int:uid>/<int:pid>/add/',views.AddItem),
    path('cart/<int:id>/clear/',views.clear_cart),
    path('cart/<int:uid>/<int:pid>/',views.update_quant_and_total),
    path('order/<int:id>/',views.order_items),
```

```
    path('order/<int:id>/add/',views.AddOrder),
    path('order/<int:id>/ship/',views.ship_address),
    path('order/<int:id>/detail/',views.order_details),
    path('cart/<int:uid>/<int:pid>/dec/',views.dec_quant_and_total),
    path('cart/<int:id>/total/',views.total_cart_price),
    path('order/<int:id>/cancel/',views.cancel_order),
    path('token/<int:uid>/<str:id>/',views.pinata_file_upload),
    path('user/<int:id>/',views.user_detail),
    path('login/', views.login_view),
    path('register/', views.RegisterView.as_view()),
    path('nft/<int:id>/',views.valid_nft),
    path('nft/<int:tid>/<str:uname>/<str:id>/update/',views.update_nft_detail),
    path('warranty/<int:tid>/',views.update_warranty),
    path('redeem/<int:tid>/',views.update_redeem),
]
```

views.py

```
from django.contrib.auth import authenticate, login
from django.http import JsonResponse
from django.views.decorators.csrf import csrf_exempt
from django.views.decorators.http import require_http_methods
from user.serializers import UserSerializer, ProductSerializer,
 Order_ItemsSerializer, Payment_DetailsSerializer, \
    Order_DetailsSerializer, RegisterSerializer
from user.models import User, Product, Order_Items,
 Payment_Details, Order_Details
from product.models import Cart, Shipping_Address, Track_Repairs, NFT_Details
from product.serializer import CartSerializer, Shipping_AddressSerializer,
 Track_RepairsSerializer, \
    NFT_DetailsSerializer
from rest_framework.decorators import api_view
from rest_framework.views import APIView
from rest_framework.response import Response
from rest_framework import status
from django.db.models import F
import random, requests, json
from web3 import Web3, HTTPProvider
from web3.gas_strategies.rpc import rpc_gas_price_strategy
```

```
from datetime import datetime, timedelta, timezone
from web3.middleware import geth_poa_middleware
from dateutil import relativedelta
from django.core.mail import send_mail
from django.conf import settings
import re

@csrf_exempt
@require_http_methods(["POST"])
def login_view(request):
    data = json.loads(request.body)
    username = data.get('username')
    password = data.get('password')
    user = authenticate(request, username=username, password=password)
    if user is not None:
        login(request, user)
        return JsonResponse({"message": "Login successful!", "user_id": user.id},
 status=200)
    else:
        return JsonResponse({"message": "Invalid username or password."},
   status=400)


class RegisterView(APIView):
    def post(self, request, format='json'):
        serializer = RegisterSerializer(data=request.data)
        if serializer.is_valid():
            user = serializer.save()
            if user:
           return Response(serializer.data, status=status.HTTP_201_CREATED)
       return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)


@api_view(['GET'])
def user_detail(request, id):
    try:
        user = User.objects.get(pk=id)
    except User.DoesNotExist:
        return Response(status=status.HTTP_404_NOT_FOUND)
    serializer = UserSerializer(user)
```

```
    return Response(serializer.data)


@api_view(['GET'])
def product_list(request):
    product = Product.objects.all().order_by('name')
    serializer = ProductSerializer(product, many=True)
    return JsonResponse(serializer.data, safe=False)


@api_view(['POST'])
def AddProduct(request, id):
    data = request.data.copy()
    data['user_id'] = id
    serializer = ProductSerializer(data=data)
    if serializer.is_valid():
        serializer.save()
      return Response(serializer.data, status=status.HTTP_201_CREATED)
    else:
      return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)


@api_view(['GET'])
def product_detail(request, id):
    try:
        product = Product.objects.get(pk=id)
    except Product.DoesNotExist:
        return Response(status=status.HTTP_404_NOT_FOUND)

    serializer = ProductSerializer(product)
    return Response(serializer.data)


@api_view(['PUT'])
def update_product_detail(request, rid, pid):
    try:
        product = Product.objects.get(pk=pid)
    except Product.DoesNotExist:
        return Response(status=status.HTTP_404_NOT_FOUND)
    data = request.data.copy()
```

```
    data['user_id'] = rid
    serializer = ProductSerializer(product, data=data)
    if serializer.is_valid():
        serializer.save()
        return Response(serializer.data)
    return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)


@api_view(['DELETE'])
def delete_product(request, id):
    try:
        product = Product.objects.get(pk=id)
    except Product.DoesNotExist:
        return Response(status=status.HTTP_404_NOT_FOUND)

    product.delete()
    return Response(status=status.HTTP_204_NO_CONTENT)


@api_view(['GET'])
def cart_list(request, id):
    try:
        cart = Cart.objects.filter(user_id=id)
    except Cart.DoesNotExist:
        return Response(status=status.HTTP_404_NOT_FOUND)
    serializer = CartSerializer(cart, many=True)
    ls = []
    total = 0
    for i in serializer.data:
        dict1 = {}
        product = Product.objects.get(id=i['product_id'])
        dict1['id'] = i['id']
        dict1['date_added'] = i['date_added']
        dict1['product_id'] = i['product_id']
        dict1['user_id'] = i['user_id']
        dict1['name'] = product.name
        dict1['price'] = product.price
        dict1['quantity'] = i['quantity']
        dict1['total_amount'] = i['total_amount']
        dict1['inCart'] = True
```

```
        ls.append(dict1)
    return Response(ls)


@api_view(['GET'])
def total_cart_price(request, id):
    try:
        cart = Cart.objects.filter(user_id=id)
    except Cart.DoesNotExist:
        return Response(status=status.HTTP_404_NOT_FOUND)
    serializer = CartSerializer(cart, many=True)
    ls = []
    total = 0
    for i in serializer.data:
        dict1 = {}
        product = Product.objects.get(id=i['product_id'])
        dict1['id'] = i['id']
        dict1['date_added'] = i['date_added']
        dict1['product_id'] = i['product_id']
        dict1['user_id'] = i['user_id']
        dict1['name'] = product.name
        dict1['price'] = product.price
        dict1['quantity'] = i['quantity']
        dict1['total_amount'] = i['total_amount']
        total += i['total_amount']
        ls.append(dict1)
    final = []
    dict2 = {'total': total}
    final.append(dict2)
    return Response(final)


@api_view(['DELETE'])
def remove_item(request, uid, pid):
    try:
        cart = Cart.objects.filter(user_id=uid, product_id=pid)
    except Cart.DoesNotExist:
        return Response(status=status.HTTP_404_NOT_FOUND)
    cart.delete()
    return Response(status=status.HTTP_204_NO_CONTENT)
```

```
@api_view(['POST'])
def AddItem(request,uid,pid):
    request.data['user_id']=uid
    request.data['product_id']=pid
    request.data['quantity'] = 1
    product = Product.objects.get(pk=pid)
    serializer1 = ProductSerializer(product)
    request.data['total_amount']=serializer1.data['price']
    serializer=CartSerializer(data=request.data)
    if serializer.is_valid():
        serializer.save()
        return Response(serializer.data,status=status.HTTP_201_CREATED)


@api_view(['DELETE'])
def clear_cart(request, id):
    try:
        cart = Cart.objects.filter(user_id=id)
    except Cart.DoesNotExist:
        return Response(status=status.HTTP_404_NOT_FOUND)
    cart.delete()
    return Response(status=status.HTTP_204_NO_CONTENT)


@api_view(['PUT'])
def update_quant_and_total(request, uid, pid):
    try:
        cart = Cart.objects.filter(user_id=uid)
    except Cart.DoesNotExist:
        return Response(status=status.HTTP_404_NOT_FOUND)

    serializer = CartSerializer(cart, many=True)
    updated_cart_data = []

    for cart_item in serializer.data:
        if cart_item['product_id'] == pid:
            cart_instance = Cart.objects.get(id=cart_item['id'])
            product = Product.objects.get(id=cart_item['product_id'])
```

```
            new_quantity = cart_item['quantity'] + 1
            new_total_amount = cart_item['total_amount'] + product.price

            # Prevent quantity from going below zero
            if new_quantity >= 0:
                cart_instance.quantity = new_quantity
                cart_instance.total_amount = new_total_amount
                updated_cart_data.append(cart_instance)

    if updated_cart_data:
        request.data['user_id'] = uid
        request.data['product_id'] = pid
        request.data['quantity'] = max(0, updated_cart_data[0].quantity)
 # Ensure quantity is not negative
        request.data['total_amount'] = updated_cart_data[0].total_amount

        serializer = CartSerializer(updated_cart_data[0], data=request.data)

    if serializer.is_valid():
        serializer.save()
        return Response(serializer.data)
    return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)


@api_view(['PUT'])
def dec_quant_and_total(request, uid, pid):
    try:
        cart = Cart.objects.filter(user_id=uid)
    except Cart.DoesNotExist:
        return Response(status=status.HTTP_404_NOT_FOUND)
    serializer = CartSerializer(cart, many=True)
    ls = []
    for i in serializer.data:
        if i['product_id'] == pid:
            if i['quantity'] > 1:
                dict1 = {}
                cart1 = Cart.objects.get(id=i['id'])
                product = Product.objects.get(id=i['product_id'])
                dict1['quantity'] = i['quantity'] - 1
```

```
                dict1['total_amount'] = i['total_amount'] - product.price
                ls.append(dict1)
    request.data['user_id'] = uid
    request.data['product_id'] = pid
    request.data['quantity'] = ls[0]['quantity']
    request.data['total_amount'] = ls[0]['total_amount']
    serializer = CartSerializer(cart1, data=request.data)
    if serializer.is_valid():
        serializer.save()
        return Response(serializer.data)
    return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)



@api_view(['GET'])
def order_items(request, id):
    order = Order_Items.objects.filter(user_id=id)
    serializer = Order_ItemsSerializer(order, many=True)
    ls = []
    for i in serializer.data:
        dict1 = {}
        product = Product.objects.get(id=i['product_id'])
        dict1['id'] = i['id']
        dict1['user_id'] = i['user_id']
        dict1['order_date'] = i['order_date']
        dict1['name'] = product.name
        dict1['price'] = product.price * i['quantity']
        dict1['quantity'] = i['quantity']
        ls.append(dict1)
    return Response(ls)


@api_view(['POST'])
def AddOrder(request, id):
    try:
        cart = Cart.objects.filter(user_id=id)
        order = Order_Items.objects.filter(user_id=id)
    except Cart.DoesNotExist or Order_Items.DoesNotExist:
        return Response(status=status.HTTP_404_NOT_FOUND)
    order.delete()
```

```
    serializer = CartSerializer(cart, many=True)
    ls = []
    for i in serializer.data:
        dict1 = {}
        dict1['product_id'] = i['product_id']
        dict1['user_id'] = i['user_id']
        dict1['quantity'] = i['quantity']
        ls.append(dict1)
    for i in range(len(ls)):
        request.data['user_id'] = ls[i]['user_id']
        request.data['product_id'] = ls[i]['product_id']
        request.data['quantity'] = ls[i]['quantity']
        serializer = Order_ItemsSerializer(data=request.data)
        if serializer.is_valid():
            serializer.save()
    return Response(ls)


@api_view(['DELETE'])
def cancel_order(request, id):
    try:
        order = Order_Items.objects.filter(user_id=id)
    except Order_Items.DoesNotExist:
        return Response(status=status.HTTP_404_NOT_FOUND)
    order.delete()
    return Response(status=status.HTTP_204_NO_CONTENT)


@api_view(['POST'])
def ship_address(request, id):
    try:
        ship = Shipping_Address.objects.filter(user_id=id)
        order = Order_Items.objects.filter(user_id=id)
        ship.delete()
    except Order_Items.DoesNotExist:
        return Response(status=status.HTTP_404_NOT_FOUND)
    request.data['user_id'] = id
    serializer = Shipping_AddressSerializer(data=request.data)
    if serializer.is_valid():
        serializer.save()
```

```
        return Response(serializer.data)
    return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)


@api_view(['POST'])
def order_details(request, id):
    request.data['payment_id'] = id
    serializer = Order_DetailsSerializer(data=request.data)
    if serializer.is_valid():
        serializer.save()
        return Response(serializer.data, status=status.HTTP_201_CREATED)


@api_view(['POST'])
def pinata_file_upload(request, uid, id):
    oorder = Order_Items.objects.filter(user_id=uid)
    oserializer = Order_ItemsSerializer(oorder, many=True)
    ls = []
    for i in oserializer.data:
        dicto = {}
        product = Product.objects.get(id=i['product_id'])
        dicto['id'] = i['id']
        dicto['product_id'] = i['product_id']
        dicto['user_id'] = i['user_id']
        dicto['order_date'] = i['order_date']
        dicto['name'] = product.name
        dicto['price'] = product.price * i['quantity']
        dicto['quantity'] = i['quantity']
        ls.append(dicto)
    final = {}
    try1 = []
    for i in ls:
        print('outer')
        finallist = []
        try2 = {}
        for j in range(i['quantity']):
            print('inner')
            product = Product.objects.filter(pk=i['product_id'])
            serializer = ProductSerializer(product, many=True)
            img_name = serializer.data[0]['image'].split('/')[-1]
```

```
            file_path = "D:/blockchain 2nd part/Backend/blockchain" + \
                        serializer.data[0]['image']
            url = "https://api.pinata.cloud/pinning/pinFileToIPFS"
            payload = {}
            files = [
                ('file', (img_name, open(file_path, 'rb'), 'image/png'))
            ]
            headers = {
             'Authorization': 'Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c
            2VySW5mb3JtYXRpb24iOnsiaWQiOiJkYTk2OTc4ZS02NWJiLTQwOGEtYTAzMS0yZGFh
            N2ViY2IyMDMiLCJlbWFpbCI6ImpvZWxzYW1qb2huc29uQHlhaG9vLmNvbSIsImVtYWls
            X3ZlcmlmaWVkIjp0cnVlLCJwaW5fcG9saWN5Ijp7InJlZ2lvbnMiOlt7ImlkIjoiRlJB
            MSIsImRlc2lyZWRSZXBsaWNhdGlvbkNvdW50IjoxfSx7ImlkIjoiTllDMSIsImRlc2l
            yZWRSZXBsaWNhdGlvbkNvdW50IjoxfV0sInZlcnNpb24iOjF9LCJtZmFfZW5hYmxlZ
            CI6ZmFsc2UsInN0YXR1cyI6IkFDVElWRSJ9LCJhdXRoZW50aWNhdGlvblR5cGU
            iOiJzY29wZWRLZXkiLCJzY29wZWRLZXlLZXkiOiI3MWEzNjJkYjMxMGY4YzdlNTB
            kMyIsInNjb3BlZEtleVNlY3JldCI6IjM2MjllZGJhMzc2Y2E0MDkwZWRiMzg4NjE2Z
            TA2MTFhODJjYzRjYjIxNTU2ZjIxMWVkZWQzZjA2ZGNmYWJjNzUiLCJpYXQiOjE
          3MDc4NDE1Njh9.Xhc2XxMBq5dIKWW-593JBH571oTMVxgtx4Op5Tt6KWE'
            }

            response = requests.request("POST", url, headers=headers,
data=payload, files=files)
            hashval = response.text.split('"')[3]
            url = "https://api.pinata.cloud/pinning/pinJSONToIPFS"
            img_url = "https://amethyst-realistic-camel-859.mypinata.cloud
/ipfs/" + hashval + "?pinataGatewayToken=MgL8ssK-
torTK_vfvilq04TZ6BpK1cdpQFCc6fRLD-K8dQ4g8f2SplukNJ-sGXrJ"
            Product_name = serializer.data[0]['name']
            time = serializer.data[0]['warranty_period'] * 31556952
            x = datetime.now() + timedelta(seconds=time)
            payload = json.dumps({
                "description": "Token URL Generation",
                "external_url": img_url,
                "image": img_url,
                "name": Product_name + "-Warranty NFT",
                "attributes": [
                    {
                        "trait_type": "Expiry",
                        "value": str(x)
```

```
                    }
                ]
            })
            headers = {
              'Authorization': 'Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1
          c2VySW5mb3JtYXRpb24iOnsiaWQiOiJkYTk2OTc4ZS02NWJiLTQwOGEtYTAzMS0yZGFh
          2ViY2IyMDMiLCJlbWFpbCI6ImpvZWxzYW1qb2huc29uQHlhaG9vLmNvbSIsImVtY
           WlsX3ZlcmlmaWVkIjp0cnVlLCJwaW5fcG9saWN5Ijp7InJlZ2lvbnMiOlt7ImlkI
           joiRlJBMSIsImRlc2lyZWRSZXBsaWNhdGlvbkNvdW50IjoxfSx7ImlkIjoiTllDM
          SIsImRlc2lyZWRSZXBsaWNhdGlvbkNvdW50IjoxfV0sInZlcnNpb24iOjF9LCJtZmFfZW5
          hYmxlZCI6ZmFsc2UsInN0YXR1cyI6IkFDVElWRSJ9LCJhdXRoZW50aWNhdGlvblR5cG
          UiOiJzY29wZWRLZXkiLCJzY29wZWRLZXlLZXkiOiI3MWEzNjJkYjMxMGY4YzdlNTBkMy
          IsInNjb3BlZEtleVNlY3JldCI6IjM2MjllZGJhMzc2Y2E0MDkwZWRiMzg4NjE2ZTA2M
          TFhODJjYzRjYjIxNTU2ZjIxMWVkZWQzZjA2ZGNmYWJjNzUiLCJpYXQiOjE3MDc4
            NDE1Njh9.Xhc2XxMBq5dIKWW-593JBH571oTMVxgtx40p5Tt6KWE',
                'Content-Type': 'application/json'
            }

            response = requests.request("POST", url, headers=headers,
        data=payload)
            hashval2 = response.text.split('"')[3]
            json_url = "https://amethyst-realistic-camel-859.mypinata.cloud
/ipfs/" + hashval2 + "?pinataGatewayToken=MgL8ssK-torTK_vfvilq04
TZ6BpK1cdpQFCc6fRLD-K8dQ4g8f2SplukNJ-sGXrJ"

            w3 = Web3(HTTPProvider('https://eth-sepolia.g.alchemy.com/v2/
GtRs0mmRk7tpkqbRpthhc5HGQAi8wDGC'))
            w3.middleware_onion.inject(geth_poa_middleware, layer=0)
            abi = '''[
    {
        "inputs": [],
        "stateMutability": "nonpayable",
        "type": "constructor"
    },
    {
        "anonymous": false,
        "inputs": [
            {
                "indexed": true,
                "internalType": "address",
```

```
                "name": "owner",
                "type": "address"
            },
            {
                "indexed": true,
                "internalType": "address",
                "name": "approved",
                "type": "address"
            },
            {
                "indexed": true,
                "internalType": "uint256",
                "name": "tokenId",
                "type": "uint256"
            }
        ],
        "name": "Approval",
        "type": "event"
    },
    {
        "anonymous": false,
        "inputs": [
            {
                "indexed": true,
                "internalType": "address",
                "name": "owner",
                "type": "address"
            },
            {
                "indexed": true,
                "internalType": "address",
                "name": "operator",
                "type": "address"
            },
            {
                "indexed": false,
                "internalType": "bool",
                "name": "approved",
                "type": "bool"
            }
```

```
        ],
        "name": "ApprovalForAll",
        "type": "event"
    },
    {
        "anonymous": false,
        "inputs": [
            {
                "indexed": true,
                "internalType": "address",
                "name": "previousOwner",
                "type": "address"
            },
            {
                "indexed": true,
                "internalType": "address",
                "name": "newOwner",
                "type": "address"
            }
        ],
        "name": "OwnershipTransferred",
        "type": "event"
    },
    {
        "anonymous": false,
        "inputs": [
            {
                "indexed": true,
                "internalType": "address",
                "name": "from",
                "type": "address"
            },
            {
                "indexed": true,
                "internalType": "address",
                "name": "to",
                "type": "address"
            },
            {
                "indexed": true,
```

```
                "internalType": "uint256",
                "name": "tokenId",
                "type": "uint256"
            }
        ],
        "name": "Transfer",
        "type": "event"
    },
    {
        "inputs": [
            {
                "internalType": "uint256",
                "name": "tid",
                "type": "uint256"
            }
        ],
        "name": "applyExpiryDiscount",
        "outputs": [],
        "stateMutability": "nonpayable",
        "type": "function"
    },
    {
        "inputs": [
            {
                "internalType": "address",
                "name": "to",
                "type": "address"
            },
            {
                "internalType": "uint256",
                "name": "tokenId",
                "type": "uint256"
            }
        ],
        "name": "approve",
        "outputs": [],
        "stateMutability": "nonpayable",
        "type": "function"
    },
    {
```

```
        "inputs": [
            {
                "internalType": "address",
                "name": "owner",
                "type": "address"
            }
        ],
        "name": "balanceOf",
        "outputs": [
            {
                "internalType": "uint256",
                "name": "",
                "type": "uint256"
            }
        ],
        "stateMutability": "view",
        "type": "function"
    },
    {
        "inputs": [
            {
                "internalType": "uint256",
                "name": "tokenId",
                "type": "uint256"
            }
        ],
        "name": "getApproved",
        "outputs": [
            {
                "internalType": "address",
                "name": "",
                "type": "address"
            }
        ],
        "stateMutability": "view",
        "type": "function"
    },
    {
        "inputs": [
            {
```

```
                "internalType": "uint256",
                "name": "startwarr",
                "type": "uint256"
            },
            {
                "internalType": "uint256",
                "name": "warrPeriod",
                "type": "uint256"
            }
        ],
        "name": "getTime",
        "outputs": [
            {
                "internalType": "uint256",
                "name": "",
                "type": "uint256"
            }
        ],
        "stateMutability": "view",
        "type": "function"
    },
    {
        "inputs": [
            {
                "internalType": "address",
                "name": "owner",
                "type": "address"
            },
            {
                "internalType": "address",
                "name": "operator",
                "type": "address"
            }
        ],
        "name": "isApprovedForAll",
        "outputs": [
            {
                "internalType": "bool",
                "name": "",
                "type": "bool"
```

```
            }
        ],
        "stateMutability": "view",
        "type": "function"
    },
    {
        "inputs": [],
        "name": "name",
        "outputs": [
            {
                "internalType": "string",
                "name": "",
                "type": "string"
            }
        ],
        "stateMutability": "view",
        "type": "function"
    },
    {
        "inputs": [],
        "name": "owner",
        "outputs": [
            {
                "internalType": "address",
                "name": "",
                "type": "address"
            }
        ],
        "stateMutability": "view",
        "type": "function"
    },
    {
        "inputs": [
            {
                "internalType": "uint256",
                "name": "tokenId",
                "type": "uint256"
            }
        ],
        "name": "ownerOf",
```

```
        "outputs": [
            {
                "internalType": "address",
                "name": "",
                "type": "address"
            }
        ],
        "stateMutability": "view",
        "type": "function"
    },
    {
        "inputs": [],
        "name": "renounceOwnership",
        "outputs": [],
        "stateMutability": "nonpayable",
        "type": "function"
    },
    {
        "inputs": [
            {
                "internalType": "uint256",
                "name": "tid",
                "type": "uint256"
            }
        ],
        "name": "replacement",
        "outputs": [
            {
                "internalType": "uint256",
                "name": "",
                "type": "uint256"
            }
        ],
        "stateMutability": "nonpayable",
        "type": "function"
    },
    {
        "inputs": [
            {
                "internalType": "address",
```

```
                "name": "to",
                "type": "address"
            },
            {
                "internalType": "uint256",
                "name": "tokenId",
                "type": "uint256"
            },
            {
                "internalType": "uint256",
                "name": "warrantyDuration",
                "type": "uint256"
            },
            {
                "internalType": "string",
                "name": "uri",
                "type": "string"
            }
        ],
        "name": "safeMint",
        "outputs": [],
        "stateMutability": "nonpayable",
        "type": "function"
    },
    {
        "inputs": [
            {
                "internalType": "address",
                "name": "from",
                "type": "address"
            },
            {
                "internalType": "address",
                "name": "to",
                "type": "address"
            },
            {
                "internalType": "uint256",
                "name": "tokenId",
                "type": "uint256"
```

```
            }
        ],
        "name": "safeTransferFrom",
        "outputs": [],
        "stateMutability": "nonpayable",
        "type": "function"
    },
    {
        "inputs": [
            {
                "internalType": "address",
                "name": "from",
                "type": "address"
            },
            {
                "internalType": "address",
                "name": "to",
                "type": "address"
            },
            {
                "internalType": "uint256",
                "name": "tokenId",
                "type": "uint256"
            },
            {
                "internalType": "bytes",
                "name": "data",
                "type": "bytes"
            }
        ],
        "name": "safeTransferFrom",
        "outputs": [],
        "stateMutability": "nonpayable",
        "type": "function"
    },
    {
        "inputs": [
            {
                "internalType": "address",
                "name": "operator",
```

```
                "type": "address"
            },
            {
                "internalType": "bool",
                "name": "approved",
                "type": "bool"
            }
        ],
        "name": "setApprovalForAll",
        "outputs": [],
        "stateMutability": "nonpayable",
        "type": "function"
    },
    {
        "inputs": [
            {
                "internalType": "bytes4",
                "name": "interfaceId",
                "type": "bytes4"
            }
        ],
        "name": "supportsInterface",
        "outputs": [
            {
                "internalType": "bool",
                "name": "",
                "type": "bool"
            }
        ],
        "stateMutability": "view",
        "type": "function"
    },
    {
        "inputs": [],
        "name": "symbol",
        "outputs": [
            {
                "internalType": "string",
                "name": "",
                "type": "string"
```

```
            }
        ],
        "stateMutability": "view",
        "type": "function"
    },
    {
        "inputs": [
            {
                "internalType": "uint256",
                "name": "tokenId",
                "type": "uint256"
            }
        ],
        "name": "tokenURI",
        "outputs": [
            {
                "internalType": "string",
                "name": "",
                "type": "string"
            }
        ],
        "stateMutability": "view",
        "type": "function"
    },
    {
        "inputs": [
            {
                "internalType": "address",
                "name": "from",
                "type": "address"
            },
            {
                "internalType": "address",
                "name": "to",
                "type": "address"
            },
            {
                "internalType": "uint256",
                "name": "tid",
                "type": "uint256"
```

```
            }
        ],
        "name": "transfer",
        "outputs": [],
        "stateMutability": "nonpayable",
        "type": "function"
    },
    {
        "inputs": [
            {
                "internalType": "address",
                "name": "from",
                "type": "address"
            },
            {
                "internalType": "address",
                "name": "to",
                "type": "address"
            },
            {
                "internalType": "uint256",
                "name": "tokenId",
                "type": "uint256"
            }
        ],
        "name": "transferFrom",
        "outputs": [],
        "stateMutability": "nonpayable",
        "type": "function"
    },
    {
        "inputs": [
            {
                "internalType": "address",
                "name": "newOwner",
                "type": "address"
            }
        ],
        "name": "transferOwnership",
        "outputs": [],
```

```
        "stateMutability": "nonpayable",
        "type": "function"
    },
    {
        "inputs": [
            {
                "internalType": "uint256",
                "name": "tid",
                "type": "uint256"
            }
        ],
        "name": "warrantyProvider",
        "outputs": [
            {
                "internalType": "uint256",
                "name": "",
                "type": "uint256"
            }
        ],
        "stateMutability": "view",
        "type": "function"
    }
]'''

            greeter = w3.eth.contract(
                address= "0xb39De1ab9E18bF003A4d927F27eb5f2F674E37A5",
                abi=abi)

            account_from = {
                "private_key": "0x7f66d0e36f46ce1a5d788077a5edd19dd11
  dbbc1a96b8ac3d6a2863579c94662",
                "address": '0x6d6928b85Cf354ADF33d4e07D096f349e5100140',
            }
            address_to = id
            # address_to = request.POST.get("to_add")
            print(address_to)
            token_id = random.randint(100000000000000, 999999999999999)

            reset_tx = greeter.functions.safeMint(Web3.toChecksumAddress
              (address_to),token_id, time,
```

```
                                        json_url).buildTransaction(
            {
                'from': account_from['address'],
                'nonce': w3.eth.get_transaction_count
 (account_from['address']),
            }
        )
        tx_create = w3.eth.account.sign_transaction(reset_tx, account_from
['private_key'])
        tx_hash = w3.eth.send_raw_transaction(tx_create.rawTransaction)
        tx_receipt = w3.eth.wait_for_transaction_receipt(tx_hash)

        request.data['product_id'] = i['product_id']
        request.data['token_url'] = json_url
        request.data['expiry_date'] = x
        request.data['token_id'] = token_id
        request.data['user_id'] = uid
        request.data['acc_address'] = address_to
        serializer = NFT_DetailsSerializer(data=request.data)
        if serializer.is_valid():
            serializer.save()
            try2[str(token_id)] = str(tx_receipt.transactionHash.hex())
            print(tx_receipt.transactionHash.hex())
            finallist.append(str(tx_receipt.transactionHash.hex()))
        try2[str(i['product_id'])] = finallist
    try1.append(try2)
    final[str(i['product_id'])] = finallist
subject = 'Warranty Receipt'
message = ""
for ele in try1:
    message += json.dumps(ele)
    print(message)
    message += "\n"
try:
    user = User.objects.get(pk=uid)
except User.DoesNotExist:
    return Response(status=status.HTTP_404_NOT_FOUND)
serializer = UserSerializer(user)
email_from = settings.EMAIL_HOST_USER
recipient_list = [serializer.data['email'], ]
```

```
    message = re.sub(r"[\([{})\]]", "", message)
    message = '\n'.join(message.split(','))
    msg2 = '''Your transaction is successful.
Below are the respective token ids and hash values of the purchased products.
You can cross verify from polygon test website.
Here is the link of the test website https://sepolia.etherscan.io'''
    final_msg = msg2 + '\n\n' + message + '\n' + 'Regards' + '\n' + 'Alchemy'
    send_mail(subject, final_msg, email_from, recipient_list)
    return Response(try1)


@api_view(['GET'])
def valid_nft(request, id):
    try:
        nft = NFT_Details.objects.filter(user_id=id)
    except NFT_Details.DoesNotExist:
        return Response(status=status.HTTP_404_NOT_FOUND)
    serializer = NFT_DetailsSerializer(nft, many=True)
    ls = []
    for i in serializer.data:
        d = datetime.fromisoformat(i['expiry_date'][:-1]).
     astimezone(timezone.utc)
        exp_date = d.strftime('%Y-%m-%d %H:%M:%S.%f')
        exp_date = datetime.strptime(exp_date, '%Y-%m-%d %H:%M:%S.%f')
        d1 = datetime.strptime(str(datetime.now().date()), "%Y-%m-%d")
        d2 = datetime.strptime(str(exp_date.date()), "%Y-%m-%d")
        z = relativedelta.relativedelta(d2, d1)
        dict1 = {}
        product = Product.objects.get(id=i['product_id'])
        serializer1 = ProductSerializer(product)
        dict1['id'] = i['id']
        dict1['token_url'] = i['token_url']
        dict1['expiry_date'] = exp_date.date()
        dict1['token_id'] = i['token_id']
        dict1['user_id'] = i['user_id']
        dict1['product_id'] = i['product_id']
        dict1['name'] = serializer1.data['name']
        dict1['image'] = serializer1.data['image']
        dict1['acc_address'] = i['acc_address']
        dict1['diff'] = str(z.years) + 'Years,' + str(z.months) + 'months,'
```

```
    + str(z.days) + 'days'
        dict1['redeem'] = i['redeem']
        ls.append(dict1)
    return Response(ls)


@api_view(['PUT'])
def update_nft_detail(request, tid, uname, id):
    try:
        nft = NFT_Details.objects.get(token_id=tid)
    except NFT_Details.DoesNotExist:
        return Response(status=status.HTTP_404_NOT_FOUND)
    user = User.objects.get(username=uname)
    request.data['user_id'] = user.id
    request.data['acc_address'] = id
    serializer = NFT_DetailsSerializer(nft, data=request.data)
    if serializer.is_valid():
        serializer.save()
        return Response(serializer.data)
    return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)


@api_view(['PUT'])
def update_warranty(request, tid):
    try:
        nft = NFT_Details.objects.filter(token_id=tid)
    except NFT_Details.DoesNotExist:
        return Response(status=status.HTTP_404_NOT_FOUND)
    add = tid % 10
    serializer1 = NFT_DetailsSerializer(nft, many=True)
    ans = serializer1.data[0]['expiry_date']
    d = datetime.fromisoformat(ans[:-1]).astimezone(timezone.utc)
    ans1 = d.strftime('%Y-%m-%d %H:%M:%S.%f')
    new_date = datetime.strptime(ans1, '%Y-%m-%d %H:%M:%S.%f')
    final = new_date + relativedelta.relativedelta(months=add)
    nft1 = NFT_Details.objects.get(id=serializer1.data[0]['id'])
    request.data['expiry_date'] = final
    serializer = NFT_DetailsSerializer(nft1, data=request.data)
    if serializer.is_valid():
        serializer.save()
```

```
        return Response(serializer.data)
    return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)


@api_view(['PUT'])
def update_redeem(request, tid):
    try:
        nft = NFT_Details.objects.get(token_id=tid)
    except NFT_Details.DoesNotExist:
        return Response(status=status.HTTP_404_NOT_FOUND)
    request.data['redeem'] = False
    serializer = NFT_DetailsSerializer(nft, data=request.data)
    if serializer.is_valid():
        serializer.save()
        return Response(serializer.data)
    return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)
```

models.py

```
from django.db import models
from user.models import User, Product, Order_Items, Payment_Details,
          Order_Details


class Cart(models.Model):
    product_id = models.ForeignKey(Product, on_delete=models.CASCADE)
    user_id = models.ForeignKey(User, on_delete=models.CASCADE)
    quantity = models.IntegerField(default=0)
    total_amount = models.FloatField(default=0)
    date_added = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return str(self.user_id)


class Shipping_Address(models.Model):
    user_id = models.ForeignKey(User, on_delete=models.CASCADE, default=0)
    address = models.CharField(max_length=200)
    city = models.CharField(max_length=200)
```

```
    state = models.CharField(max_length=200)
    postal_code = models.IntegerField()
    date_added = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return str(self.user_id)


class Track_Repairs(models.Model):
    order_id = models.ForeignKey(Order_Items, on_delete=models.CASCADE)
    description = models.CharField(max_length=200, null=True)
    details_id = models.ForeignKey(Order_Details, on_delete=models.CASCADE)
    percent_work_done = models.FloatField()

    def __str__(self):
        return self.order_id


class NFT_Details(models.Model):
    product_id = models.ForeignKey(Product, on_delete=models.CASCADE, default=0)
    user_id = models.ForeignKey(User, on_delete=models.CASCADE, default=0)
    token_url = models.CharField(max_length=200, default='')
    expiry_date = models.DateTimeField(default='')
    token_id = models.IntegerField(unique=True, default=0)
    acc_address = models.CharField(max_length=300, default='')
    redeem = models.BooleanField(default=True)

    def _str_(self):
        return str(self.token_id)
```

serializer.py

```
from django.db.models import fields
from rest_framework import serializers
from .models import Cart,Shipping_Address,Track_Repairs,NFT_Details
from user.serializers import ProductSerializer


class CartSerializer(serializers.ModelSerializer):
```

```
    class Meta:
        model = Cart
        fields = '__all__'


class Shipping_AddressSerializer(serializers.ModelSerializer):
    class Meta:
        model = Shipping_Address
        fields = '__all__'


class Track_RepairsSerializer(serializers.ModelSerializer):
    class Meta:
        model = Track_Repairs
        fields = '__all__'


class NFT_DetailsSerializer(serializers.ModelSerializer):
    class Meta:
        model = NFT_Details
        fields = '__all__'
```

Smart Contract.sol

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.20;

import "@openzeppelin/contracts@5.0.0/token/ERC721/ERC721.sol";
import "@openzeppelin/contracts@5.0.0/token/ERC721/extensions/ERC721
  URIStorage.sol";
import "@openzeppelin/contracts@5.0.0/access/Ownable.sol";

contract Alchemy is ERC721, ERC721URIStorage, Ownable {
    uint256 private _nextTokenId;

    constructor(address initialOwner)
        ERC721("Alchemy", "ALCH")
        Ownable(initialOwner)
    {}
```

```
    function safeMint(address to, string memory uri) public onlyOwner {
        uint256 tokenId = _nextTokenId++;
        _safeMint(to, tokenId);
        _setTokenURI(tokenId, uri);
    }

    // The following functions are overrides required by Solidity.

    function tokenURI(uint256 tokenId)
        public
        view
        override(ERC721, ERC721URIStorage)
        returns (string memory)
    {
        return super.tokenURI(tokenId);
    }

    function supportsInterface(bytes4 interfaceId)
        public
        view
        override(ERC721, ERC721URIStorage)
        returns (bool)
    {
        return super.supportsInterface(interfaceId);
    }
}
```

PaymentScreen.js

```
import axios from 'axios';
import React, { useState } from 'react';
import { useNavigate } from 'react-router-dom';
import Divider from "@material-ui/core/Divider";
import "bootstrap/dist/css/bootstrap.min.css";
import { useWeb3React } from '@web3-react/core'

const PaymentScreen = () => {
  const navigate = useNavigate();
```

```
const [address, setAddress] = useState("")
const [state, setState] = useState("")
const [city, setCity] = useState("")
const [postalCode, setPostalCode] = useState("")
const [isSave, setSave] = useState(false)
const [acc, setacc] = useState("")
const [response, setResponse] = useState([])
const [trans, settrans] = useState(false);

const { library } = useWeb3React();
const userId = localStorage.getItem('user_id');

async function connect() {
  if (window.ethereum) {
    console.log('detected');

    try {
      await window.ethereum.enable()

    } catch (ex) {
      console.log('Error connecting...');
    }

  } else {
    alert('Meta Mask not detected');
  }
  const accounts = await window.ethereum.request
 ({ method: 'eth_requestAccounts' });
  setacc(accounts[0]);
  console.log(accounts[0], "abc");
   await transaction(accounts[0]);

}
async function transaction(acc) {
  console.log(acc)
  try {
    const response = await axios.post(`http://127.0.0.1:8000/api/token/
  ${userId}/${acc}/`);
    settrans(true);
```

```
      console.log(response.data);
      setResponse(response.data);
    } catch (error) {
      console.error('Error performing transaction:', error);
      // Handle error as needed
    }
  }

  async function connect() {
    if (window.ethereum) {
      console.log('MetaMask detected');

      try {
        await window.ethereum.enable();
        const accounts = await window.ethereum.request
    ({ method: 'eth_requestAccounts' });
        setacc(accounts[0]);
        console.log(accounts[0]);
        await transaction(accounts[0]);
// Call transaction function after enabling and getting accounts
      } catch (error) {
        console.error('Error connecting or getting accounts:', error);
        // Handle error as needed
      }

    } else {
      alert('MetaMask not detected');
    }
  }



  const addShippingAddress = async () => {
    let formField = new FormData()
    formField.append('address', address)
    formField.append('city', city)
    formField.append('state', state)
    formField.append('postalCode', postalCode)
    console.log(formField.getAll('city'));
    await axios({
```

```
    method: 'post',
    url: 'http://127.0.0.1:8000/api/order/1/ship/',
    data: {
      "address": address,
      "city": city,
      "state": state,
      "postal_code": postalCode
    }
  }).then(response => {
    setSave(true);
    console.log(response.data);

  })
}

return (

  <div className="container">
    <div className="w-75 mx-auto shadow p-5">

      <h5 style={{ color: 'black', fontFamily: "arial", fontSize: 19 }}>
 Add Shipment address :</h5>
      <Divider /><br></br>
      <div className="form-group">
        <input style={{ fontFamily: "arial", fontSize: 14 }}
          type="text"
          className="form-control form-control-lg"
          placeholder="Enter shipping Address :"
          name="address"
          value={address}
          onChange={(e) => setAddress(e.target.value)}
        />
      </div>
      <div className="form-group">
        <input style={{ fontFamily: "arial", fontSize: 14 }}
          type="text"
          className="form-control form-control-lg"
          placeholder="Enter City :"
          name="city"
          value={city}
```

```
            onChange={(e) => setCity(e.target.value)}
          />
        </div>
        <div className="form-group">
          <input style={{ fontFamily: "arial", fontSize: 14 }}
            type="text"
            className="form-control form-control-lg"
            placeholder="Enter State :"
            name="state"
            value={state}
            onChange={(e) => setState(e.target.value)}
          />
        </div>

        <div className="form-group">
          <input style={{ fontFamily: "arial", fontSize: 14 }}
            type="text"
            className="form-control form-control-lg"
            placeholder="Enter Postal Code :"
            name="postalCode"
            value={postalCode}
            onChange={(e) => setPostalCode(e.target.value)}
          />
        </div>
        <button style={{ fontFamily: "arial", fontSize: 14 }}
      variant="outlined"
          className="btn btn-outline-primary mr-2"
disabled={isSave ? true : false} onClick={() => addShippingAddress()}>
          {isSave ? (
            <p className="text-capitalize mb-0" disabled>
              {""}
              Saved
            </p>
          ) : (
            <p className="text-capitalize mb-0" >save</p>
          )}</button><br></br><br></br>
        <Divider /><br></br>
        <h5 style={{ color: 'black', fontFamily: "arial", fontSize: 19 }}>
Payment :</h5>
        <p style={{ fontFamily: "arial", fontSize: 14 }}>
```

```
NFT related transactions are initiated using metamask so make
 sure pre installed.</p>
       <p style={{ fontFamily: "arial", fontSize: 14 }}>
If not !! install it from the here <i className=
   "fas fa-arrow-right"></i> {' '}
         <a href="https://chrome.google.com/webstore/detail/metamask/
nkbihfbeogaeaoehlefnkodbefgpgknn?hl=en">Install Metamask</a>
       </p>

       <p style={{ color: "red", fontFamily: "arial", fontSize: 14 }}>
  Wallet Address:{acc}</p>

       <button style={{ fontFamily: "arial", fontSize: 14 }}
  className="btn btn-success btn-block"
         onClick={() => connect()}>Proceed to Pay</button><br></br>
       <div >
         {trans ? (
           <p style={{ color: 'red', fontFamily: "arial", fontSize: 14 }}>
             {""}
        Transaction Completed <br></br>Hash id per Token id is as follows :
           </p>
         ) : (
           <p></p>
         )}
       </div>
       <p>{response.map((i) =>
       (<div style={{ color: 'red', fontFamily: "arial", fontSize: 14 }}>{
         `${Object.keys(i)[0]}`} : {Object.values(i)[0]
         }</div>))}</p>
       <div >
         {trans ? (
           <p style={{ color: 'red', fontFamily: "arial", fontSize: 14 }}>
             {""}
      Your User Profile is updated with your recent purchase !! <br></br>
      You can check your payment status using Hash id in the link given
      <a href="https://sepolia.etherscan.io/"> HERE</a>!!
           </p>
         ) : (
           <p></p>
         )}
```

```
        </div>

      </div>
    </div>

  );
};

export default PaymentScreen;
```

NFTDetails.js

```
import axios from 'axios';
import React, {useState, useEffect} from 'react';
import { useNavigate, Link, useLocation } from 'react-router-dom';
import Web3 from 'web3';
import contractABI from "../Contract.json";
import Modal from "react-bootstrap/Modal";
import Button from 'react-bootstrap/Button';

const NFTDetails = () => {
  const location= useLocation();
  const [Products, setProducts] = useState([]);
  const [isRepair, setRepair] = useState(false);
  const [isCheck, setCheck] = useState(false);
  const [username, setUsername] = useState(null)
  const [acc, setAcc] = useState(null)
  const [response,setResponse] = useState([])
  const navigate = useNavigate();
  const [trans,settrans]=useState(false);
  const [pro,setPro]=useState(true);
  const { token_id: token_id, product_id:product_id,
    acc_address:acc_address, expiry_date:expiry_date,diff:diff} =
       location.state;
  const date1=new Date(expiry_date);
  const date2=new Date();
  const [isShow, invokeModal] = React.useState(false)

  const initModal = () => {
     return invokeModal(!isShow)
```

```
  }


  const getSingleProducts = async () => {
    const  { data } = await axios.get(‘http://127.0.0.1:8000/api/
   product/${product_id}/‘)
    console.log(data);
    setProducts(data);
  }

 async function transferHistory(){
  if (window.ethereum) {
    window.web3 = new Web3(window.ethereum);
    const web3 = new Web3(window.ethereum);
    const block = await web3.eth.getBlockNumber();
    const contractAddress = "0xb39De1ab9E18bF003A4d927F27eb5f2F674E37A5"
    const contractInstance = new web3.eth.Contract(contractABI,
       contractAddress);
    contractInstance.getPastEvents(’Transfer’, {
                  // filter:
  {from:’0xbf6f03450452271073877Bb4A36A5c4ED6244957’ },
                  fromBlock: block-999,
                  toBlock: ’latest’
    }, (error, events) => {
        if (!error){
            settrans(true)
            console.log(events)
            setResponse(events)
        }
    })
  } else {
    console.error("Web3 provider not available");
}


  }


  async function repair(){
    if(window.ethereum) {
```

```
    console.log('detected');

    try {
        await window.ethereum.enable()

    } catch (ex) {
      console.log('Error connecting...');
    }

  } else {
    alert('Meta Mask not detected');
  }
  const accounts = await window.ethereum.request
 ({ method: 'eth_requestAccounts' });

  console.log(accounts[0]);
  if(accounts[0]==acc_address && date1>date2){
    setRepair(true);
    console.log("repair initiated");
  }else{
    setCheck(true);
  }


}
async function transfer(username,acc){
  window.web3 = new Web3(window.ethereum);
  const web3 = new Web3(window.ethereum);
  console.log(username,acc);

  const contractAddress = "0x74EE8f104FCABE391C8d9d07A14b7bc1A650bf54"
  const contractInstance = new web3.eth.Contract
  (contractABI,contractAddress);
  const transaction = {
        from: acc_address,
        to: "0x74EE8f104FCABE391C8d9d07A14b7bc1A650bf54",
 //contractAddress of the concerned token (same in data below)
        data: contractInstance.methods.transfer(
          acc_address,
          acc,
```

```
            token_id
          ).encodeABI()
          //value given by user should be multiplied by 1000
        };
        await window.web3.eth
          .sendTransaction(transaction)
          .on("confirmation", function (confirmationNumber, result) {
            if (result && confirmationNumber === 1) {
              const transactionHash = result.transactionHash;
              console.log("transaction" , transactionHash);

            }

        });
        console.log("abc");
       await updateTransfer(username,acc);

}
async function redeem(){
  window.web3 = new Web3(window.ethereum);
  const web3 = new Web3(window.ethereum);
  console.log(username,acc);

  const contractAddress = "0x74EE8f104FCABE391C8d9d07A14b7bc1A650bf54"
  const contractInstance = new web3.eth.Contract
 (contractABI,contractAddress);
  const transaction = {
        from: acc_address,
        to: "0x74EE8f104FCABE391C8d9d07A14b7bc1A650bf54",
 //contractAddress of the concerned token (same in data below)
        data: contractInstance.methods.applyExpiryDiscount(
          token_id
        ).encodeABI()
        //value given by user should be multiplied by 1000
      };
      await window.web3.eth
        .sendTransaction(transaction)
        .on("confirmation", function (confirmationNumber, result) {
          if (result && confirmationNumber === 1) {
            const transactionHash = result.transactionHash;
```

```
            console.log("transaction" , transactionHash);

          }

       });
       console.log("abc");
      await updateWarranty();
      await updateRedeem();

}

const updateWarranty=async ()=>{
  await axios({
    method: 'PUT',
    url: `http://127.0.0.1:8000/api/warranty/${token_id}/`,

}).then(response => {
    alert('Surprise !! Your Warranty duration is extended !!')
    console.log(response.data);
    navigate("/UserProfile");
})

}
const updateRedeem=async ()=>{
  await axios({
    method: 'PUT',
    url: `http://127.0.0.1:8000/api/redeem/${token_id}/`,

}).then(response => {

    console.log(response.data);
    navigate("/UserProfile");
})

}

const updateTransfer = async (username,acc) => {
  console.log(username,acc,token_id)

  await axios({
```

```
      method: 'PUT',
      url: `http://127.0.0.1:8000/api/nft/${token_id}/
    ${username}/${acc}/update/`,

  }).then(response => {
      alert('Transfer Completed!!')
      console.log(response.data);
      navigate("/UserProfile");
  })

 }


  useEffect(() => {
    getSingleProducts();
  },[product_id])

    return (
      <div class="e-card e-card-horizontal" style={{marginLeft:30,
   marginTop:30,marginBottom:30,marginRight:30}}>
      <div class="row no-gutters">
          <div class="col-md-4">
          <img src={'http://127.0.0.1:8000' + Products.image}
     height="400" width="400"/>
          </div>
          <div class="col-md-8">
          <div class="card-body">
          <h5 style={{fontFamily: "Georgia, serif",fontSize:30 }}
  class="card-title">{Products.name}</h5>
          <p style={{fontFamily: "arial"}}>User account address :
        {acc_address}</p>
            <p style={{fontFamily: "arial"}}>Token Id: {token_id}</p>
           <p style={{fontFamily: "arial"}}>Expiry Date: {expiry_date}</p>
           <p style={{fontFamily: "arial"}}>Warranty time remaining :
          {diff}</p>
           <p style={{fontFamily: "arial"}}>{Products.description}</p>
           <p style={{fontFamily: "arial"}}>Price: Rs.{Products.price}</p>
          <p class="card-text" style={{fontFamily: "arial"}}>
  <small class="text-muted">
In case of any defect within the warranty period apply for
```

```
          repair/replacement!</small></p>

          <button style={{fontFamily: "arial",fontSize:14}}
    variant="outlined"
                  className="btn btn-outline-primary mr-2" disabled =
 {isRepair?true:false} onClick={()=>repair()}>
                  {isRepair?(
                     <p className="text-capitalize mb-0" disabled>
                       {""}
                       Repair initiated
                     </p>
                     ):(
                         <p className="text-capitalize mb-0" >
   Need Repair?</p>
                     )}</button>{' '}

           <button style={{fontFamily: "arial",fontSize:14}}
    variant="outlined"
          className="btn btn-outline-primary mr-2"onClick={initModal}>
 Transfer</button>{' '}
                   <Modal show={isShow}>
                     <Modal.Header closeButton onClick={initModal}>
                     <Modal.Title style={{fontFamily:"arial"}} >
Add Receiver's Details</Modal.Title>
                     </Modal.Header>
                     <Modal.Body>

                     <div>
                     <label style={{fontFamily:"arial",fontSize:14}}>
Add Username:</label><br></br>
                          <div className="form-group">
                            <input style={{fontFamily: "arial",fontSize:14}}
                             type="text"
                             className="form-control form-control-lg"
                             placeholder="Enter Reciever's username"
                             name="username"
                             value={username}
                             onChange={(e) => setUsername(e.target.value)}

                             />
```

```
                            </div>
 <label style={{fontFamily:"arial",fontSize:14}}>
Add Receiver's Metamask Account address:</label><br></br>
                            <div className="form-group">
                              <input style={{fontFamily: "arial",fontSize:14}}
                               type="text"
                               className="form-control form-control-lg"
                               placeholder="Enter metamask account address"
                               name="acc"
                               value={acc}
                               onChange={(e) => setAcc(e.target.value)}

                              />
                            </div>
                        </div>

                        </Modal.Body>
                        <Modal.Footer>
                        <Button style={{fontFamily:"arial"}}
     variant="outlined"
classNameName="btn btn-outline-danger mb-3 px-5" onClick={initModal}>
                            Close
                        </Button>

                        <Button style={{fontFamily:"arial"}}
    variant="outlined"
 classNameName="btn btn-outline-success mb-3 px-5"
 onClick={() => transfer(username,acc)}>
                            Transfer
                        </Button>

                        </Modal.Footer>
                    </Modal>
                     <button style={{fontFamily: "arial",fontSize:14}}
   variant="outlined"
                     classNameName="btn btn-outline-primary mr-2"
                    onClick={() => transferHistory()}>
See transfer History</button>{' '}
                    <button style={{fontFamily: "arial",fontSize:14}}
  variant="outlined"  disabled = {redeem? false:true}
```

```
                    className="btn btn-outline-success mr-2"
onClick={() => redeem()}>
                  {redeem?(
                    <p className="text-capitalize mb-0">
                      Redeem Gift</p>
                    ):(
                        <p className="text-capitalize mb-0" disabled>
Already Redeemed</p>
                    )}</button><br></br>

                 <p> {isCheck?(
                    <p style={{fontFamily:"arial",color:"red"}}
className="text-capitalize mb-0" disabled>
                      {""}<br></br>
       Make sure you are connected with same metamask account.
<br></br>if connected! then your warranty period is over!!!
  Repair cant be initiated.
                    </p>
                    ):(
                        <p className="text-capitalize mb-0" ></p>
                    )}</p>
                <div >
                {trans?(
         <p style={{ color: 'red',fontFamily:"arial",fontSize:14}}>
                      {""}
                      Transfer history of this product is as follows:
                    </p>
                    ):(
                        <p></p>
                    )}
                </div>
                <div>
                  {response.map((res, index) => {
                    if (res.returnValues[0]!=
                    "0x0000000000000000000000000000000000000000"
                    && res.returnValues[1]!=
                    "0x0000000000000000000000000000000000000000"){
         return <div style={{ color: 'Brown',fontFamily:"arial"
            ,fontSize:12}}>
                      <p>{index}</p>
```

```
                          <p>from :{res.returnValues[0]}</p>
                          <p>to :{res.returnValues[1]}</p>
                          <p>Token ID :{res.returnValues[2]}</p>
                        </div>;

                        }

                      })}
                    </div>

           </div>
          </div>
          </div>
        </div>

    );
};

export default NFTDetails;
```

WEBSITE OF ABILITY CONNECT

PROJECT REPORT

Submitted By

DON DAVIS

Reg. No. CCAVBCA031

For the award of the Degree of

Bachelor of COMPUTER APPLICATION(BCA)

in Computer Science

(University of Calicut)

under the guidance of

Ms. Rasmi P M

Assistant Professor

BACHELOR OF COMPUTER APPLICATION
DEPARTMENT OF COMPUTER SCIENCE
CHRIST COLLEGE
IRINJALAKUDA, KERALA

2021-2024

DEPARTMENT OF COMPUTER SCIENCE

CHRIST COLLEGE (AUTONOMOUS)

IRINJALAKUDA

CERTIFICATE

This is to certify that the project report entitled ***"Website of Ability Connect"*** *is a bonfied record of the project work done by* ***Don Davis*** *in partial fulfillment of the requirement for the sixth semester of* ***Bachelor of Computer Science*** *in Department of Computer Science of* **CHRIST COLLEGE (AUTONOMOUS) IRINJALAKUDA**

Ms. Rasmi P M
Assistant Professor,CS
Internal Guide

Ms. Sini Thomas
Head of the Department
Computer Science

EXTERNAL EXAMINER

INTERNAL EXAMINER

DECLARATION

We here by declare that this project work **"WEBSITE OF ABILITY CONNECT"** submitted by Christ College (Autonomous)Irinjalakuda,affliated to Calicut University in partial fullfillment of the requirement for the award of the Bachelor of Computer Application, is a record of original work done by us,under the guidance of Ms.RASMI P M,Department of Computer Science.

Place: Irinjalakuda DON DAVIS

ACKNOWLEDGEMENT

First and foremost we like to thank Lord almighty for his providence and for being the guiding light throughout the project. We wish to express my sincere gratitude to our beloved Department head for giving me all the facilities for our project. We take this opportunity to express my gratitude to the class teacher Ms.SOWMYA P.S and head of the department Ms.SINI THOMAS who has been supported us throughout the course of this project. We are thankful for her aspiring guidance and valuable advice during the project work. We express my sincere thanks to my project guide Ms.RASMI P M for supporting and guiding throughout the project.We would take this opportunity to specially thank all other faculty members for their constant and continuous motivation. Finally we would like to thank my family and friends for giving valuable advice and moral support throughout our project.

ABSTRACT

WEBSITE OF ABILITY CONNECT is a innovative website introduced as a learning platform for differently abled students in assistance of Computer Science Department of Christ College(Autonomous) Irinjalakuda. The website is enriched with two kind of login facilities - student login, teachers login. It is a learning platform and the main features are- student registration, learning and exam dashboards and so on. All these features make this website more adaptable and user-friendly.

Contents

Chapter 1

1 Introduction

Education is the cornerstone of empowerment and growth, yet education systems often fail to accommodate the diverse needs of students with various disabilities .Our website endeavors to help the learning process by crafting inclusive assessments specifically tailored to each student's abilities, while also introducing a spectrum of learning methods—audio, text, video, and interactive activities that can help foster a learning environment.

.

1.1 Overview

The objective of the Ability Connect Website is to design an simple and adaptable website that helps the users in their learning process by tailoring different modes through which they can access education . Different modes of examinations, in-website voice navigation and other features helps website to be user friendly and create a learning environment for differently abled students .

Chapter 2

2 System Analysis

2.1 Purpose

The main purpose of the website owned by Don Davis, Adhithyan T.J, Leyon T.John ,Mithra prothasis is to make a user friendly website as a inclusive learning platform for diffrently abled students.

2.1.1 Existing System

In the existing system, there may be limited opportunities for disabled children to learn or develop their skills independently, especially at a young age. While schools and special education programs may offer support from teachers and other aides, there may be gaps in providing opportunities for self-directed learning and skill development outside of structured classroom settings. Additionally, parents of disabled children may face challenges in providing supplemental learning experiences or resources at home due to lack of time, knowledge, or access to suitable materials. Limitations of existing system : Self-directed learning and skill development outside of structured classroom settings and difficulties in accesing diffrent modes of learning according to disabilities.

2.1.2 Proposed System

The materials will be designed to be accessible to children with various disabilities. Users can specify any disabilities or special needs they have to ensure that the platform can customize them.The website will incorporate learning technologies to adjust the difficulty level and pacing of activities based on the student's performance and progress. This ensures that each student is appropriately challenged and supported throughout their learning journey. Proposed system aims to create an inclusive and empowering learning environment for disabled children. The website will provide feedback to students on their performance and progress, including scores, achievements, and areas for improvement. Parents or teachers may also have access to progress reports to monitor the student's development and provide additional support as needed. Other Features : A message system that sends notification to user mail about the new contents or exams assigned.

2.2 Problem definition

To know what the problem is and what the needs are before developing it. Here we design a website as a learning platform for differently abled students.

2.3 Feasibility Study

After the problem is clearly understood and solutions proposed, the next step is to conduct the feasibility study. Feasibility study is defined as evaluation or analysis of the potential impact of the proposed project or program. The objective is to determine whether the proposed system is feasible. There are three aspects of feasibility study to which the proposed system is subjected as discussed below.

2.3.1 Technical Feasibility

Technical feasibility assess whether the current technical resources are sufficient for the new system. We can upgrade the level of technology for supporting our website. We check whether the proposed system can be implemented in the present system without supporting the existing hardware.

2.3.2 Economical Feasibility

Economic feasibility determines whether the time and money are available to develop the system. There is no additional hardware used to develop the site, it is inexpensive to build.

2.3.3 Operational Feasibility

Operational feasibility determines if the human resources are available to operate the system once it has been installed. No extra training is needed to use this system. Anyone who has knowledge in internet in english language can easily use the system. The resources that are required to implement or install are already available with the organization.

Chapter 3

3 Software Requirement Specification

3.1 Purpose

The purpose of creating an educational website for differently-abled students is multifaceted and revolves around addressing the unique needs and challenges that these students may face. The educational website for differently-abled students is to create an inclusive, accessible, and supportive online learning environment that empowers individuals with diverse abilities to thrive academically and personally. The project aims to bridge gaps, break down barriers, and contribute to a more inclusive educational landscape.

3.2 Scope

The scope of the "Educational Website for Differently-Abled Students" project is a comprehensive initiative aimed at developing an inclusive and accessible online learning platform. The project will focus on catering to the unique needs of differently-abled students, educators, and administrators, providing a supportive and engaging environment for learning. It also aims to create a holistic and inclusive educational platform that not only meets the immediate needs of its users but also lays the groundwork for ongoing growth and enhancement.

3.3 Overall Description

The project for developing an "Educational Website for Differently-Abled Students" is a visionary initiative with the primary goal of creating an inclusive and accessible online learning platform. This comprehensive website is designed to cater to the diverse educational needs of students with varying abilities, ensuring they have equal access to quality learning resources and a supportive community. At the heart of the project is the creation of an adaptive learning environment. The website will feature a dynamic system that tailors educational content to the individual needs, preferences, and learning styles of differently-abled students.

3.3.1 Product Perspective

The "Educational Website for Differently Abled Students" involves considering how the website fits into the broader context of educational technology and the needs of its users.

3.3.2 Product Functionality

Through this website teachers can upload study materials and quizzes as a part of exam and teacher can see the result of each student. The students get notified

if new material or new test been assigned. And the main highlight is that this website can be voice controlled specially for blind students.

3.3.3 Users and Characteristics

There are two types of users admin or teacher and student. In admin page the teacher can upload new study materials and tests, teacher can also see the growth of each student. And teachers can give suggestions for each type of student. In the student page they can see all the study materials and tests. There will be an entertainment section where there will be small stories, contents and games.

3.4 Specific Requirements

3.4.1 Hardware Requirements

- System: IBM-Compatible PC
- Processor: Intel Core i3 Or Above
- Speed: Above 1GHz
- RAM capacity: 4 GB Or Above
- Hard Disk Drive: 256 GB Or Above

3.4.2 Software Requirements

- Front End : HTML, CSS, Javascript
- Back End : Python
- Database : Sqlite3
- IDE : Visual Studio Code

3.5 Functional Requirements

It contains two main modules.

- 1.Teacher/Admin
- 2.Student

Admin

An admin account is used for editing or managing the website dynamically by admin panel. The admin can add new study materials, new tests and they can also review student performance and assign necessary suggestions according to their performance. The admin will be acting as teacher role.

Student

The user or the student can login the website, and can access the study materials assignd by the admin or teacher. The will be tests, suggestions, and small entertainment section for leisure time.

3.6 Non Functional Requirements

Non-functional requirements define the overall qualities or attributes of the resulting system. Non-functional system place restrictions on the product being developed, the developed process, and specify external constraints that the product must meet. Examples of non-functional requirements include safety, security, usability, reliability and performance requirements. Project management issues (costs, time and schedule) are often considered as non-functional requirements. The principal non - functional constraints which are relevant to critical systems :

- performance
- security
- safety
- usability

Performance

Performance requirements concern the speed of operation of a system. Types of performance requirements :

- Response requirements (how quickly the system reacts to a user input).
- Throughput requirements (how much the system can accomplish within a specified amount of time).
- Availability requirements (is the system available for service when requested by end users). The speed of operation of this system is adequate for the requirements.

Reliability

- Reliability is the ability of a system to perform its required function under stated conditions for a specified period of time.
- constraints on the runtime behavior of the system. This system is reliable because its functionalities can be done on the required conditions.

Safety

Safety requirements are not required which exclude unsafe situation from the possible solution of the system.

Usability

Usability is the ease with which a user can learn to operate, prepare inputs for, and interpret outputs of system or components. Usability requirements include :

- information error messages.
- well-formed user interfaces.

3.7 Interface Requirements

3.7.1 Hardware interfaces

The system must run over the internet, all the hardware shall require to connect internet will be hardware interface for the system. As for example modem, WAN-LAN.

3.7.2 Software interfaces

Software interface required for the working of the project is the appropriate operating system.

3.7.3 Communication interfaces

The user of the site communicate using the network connectivity and the data set is accessed

3.8 Security Requirements

- User accesses only their account.
- Validation of input is handled.
- This application containing the computer systems is physically secured against arms or surreptitious entry by intruders.
- Users must be authorized carefully to reduce changes of any such user giving access to an intruder in exchange for a bribe or other favour.

3.9 Platform Used

Windows 10 is a major version of the Microsoft Windows operating system that was released on July 29, 2015.It is built on the Windows NT kernel and follows windows 8.Part of the reason Microsoft decided to name the 2015 release "Windows 10"(and skipped "windows 9")is because the operating system is designed to be a new direction for Microsoft. One of the primary aims of windows 10 is to Unify the windows experience across multiple devices, such desktop computers, tablets, and smartphones. As part of this effort, Microsoft developed Windows10 Mobile alongside Windows 10 to replaces Windows Phone

- Microsoft's previous mobile OS. Windows 10 also integrates other Microsoft services, such as Xbox Live and the Cortana voice recognition assistant.

3.10 Technologies Used

JavaScript

JavaScript is a programming language that enables interactivity and dynamic behavior on web pages. It can be used to manipulate the HTML and CSS of a webpage, handle user interactions, fetch data from servers, and much more.HTML and CSS can be embedded with JavaScript to create a working efficient website. HTML is the standard markup language for creating web pages. It provides the structure of a webpage by using elements or tags to define different parts of the content. CSS is used to style the HTML elements and define their appearance on the webpage. It allows you to control the layout, colors, fonts, and other visual aspects of your website.

Sqlite3

SQLite is a lightweight, serverless, self-contained, and embedded SQL database engine. It's widely used in various applications due to its simplicity, efficiency, and ease of integration. In the context of a website, SQLite can be utilized to store and manage data on the server-side, providing persistence for web applications. The Sqlite libraries provide APIs for performing CRUD (Create, Read, Update, Delete) operations, executing SQL queries, and managing database connections. By integrating SQLite into your website, you can efficiently store and manage data, enabling features like user authentication, data persistence, and dynamic content generation.

Chapter 4

4 Design Document

4.1 Purpose

The purpose of the website is to provide accessible and inclusive educational resources for disabled students with visual and auditory impairments. The website aims to support these students in their studies by offering:

- Accessible Study Materials: The website provides notes and educational materials in various accessible formats, such as voice notes for blind students and video/photo classes for students with hearing impairments.
- Interactive Learning Tools: Students can engage with interactive learning tools tailored to their specific needs, including audio descriptions for visual content and subtitles or sign language interpretation for video content.
- Assessments and Tests: The website offers assessments and tests designed to accommodate the needs of disabled students, ensuring fair evaluation of their understanding and progress.
- Teacher Support and Suggestions: Teachers can provide personalized support and suggestions to disabled students through the platform, offering guidance and assistance to help them succeed academically.
- Voice Command Input: For blind students, the website incorporates voice command input functionality, allowing them to navigate the platform and interact with content using voice commands, enhancing accessibility and usability.

4.2 Scope

The website aims to provide accessible study materials, including voice notes for blind students and video/photo classes for those with hearing impairments. It facilitates interactive learning through tests, offers teacher suggestions, and incorporates voice command input for blind users, fostering an inclusive educational environment for disabled students.

4.3 Overview

The purpose of this document is to help the reader to visualize the solution to the project presented.This document verifies how the design meet the requirement stimulated in the SRS document through design viewpoints.The design viewpoints will cover all design elements presented before. This document will provide a direct approach to the development of this project hence reducing feature creep and ponitedly determine the quality of the design.

4.4 Data Design

Database are the storehouses of data used in the software systems.The data is stored in tables inside the database.Several tables are created for the manipulation of the data for the system.Two essential settings for a database are

- Primary Key- The field that is unique for all the record occurrence.
- Foreign Key- The field used to set relation between tables.

Normalization

- Normalization is the process of organizing the data in the database.
- Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate the undesirable characteristics like Insertion, Update and Deletion Anomalies.
- Normalization divides the larger table into the smaller table and links them using relationship.
- The normal form is used to reduce redundancy from the database table.

Tables

Login

Name	DataType	Constraints	Description
username	varchar(100)	Notnull	Username of student
password	varchar(100)	Notnull	Password of student

Student

Name	DataType	Constraints	Description
studentid	Charfield(50)	Primarykey	ID of users
firstname	Charfield(100)	Notnull	First name of users
lastname	Charfield(100)	Notnul	Last name of users
email	Emailfield)	Notnull	Email of the users
gender	Charfield(100)	Notnull	Gender of the user
age	PositiveIntegerfield	Notnull	Age of user
disability	Charfield(100)	Notnull	Disability of user
access technology	Charfield(200)	Notnul	Users access technology

Question Table

Name	DataType	Constraints	Description
type	Charfield	Not Null	Question Type
text_id	Charfield	Foreignkey	Text Question
image	Filefield	Foreignkey	Image Question
audio	Filefield	Foreignkey	Audio Question

Scoremodel Table

Name	DataType	Constraints	Description
student	Charfield	Foreignkey	Student Name
score	Integerfield	Foreignkey	Exam Score
category	Charfield	Foreignkey	Category
suggestion	Textfield	Not Null	Suggestions

Suggestions

Name	DataType	Constraints	Description
suggestion	Textfield	Notnull	Suggestions
category	Charfield	Foreignkey	Category
video	Filefield	Foreignkey	Suggested Video
audio	Filefield)	Foreignkey	Suggested audio

Chapter 5

5 Development of the System

The website will feature a user-friendly interface with options for visually impaired students to access voice notes and hearing-impaired students to engage in video and photo classes. It will include an integrated testing platform, teacher feedback system, and voice command functionality for blind students, aiming to enhance accessibility and support disabled students in their studies.

Chapter 6

6 System Testing

Testing is the penultimate step of software development.An elaborate testing of the data is prepared and the system is using the test data,while doing testing,errors are noted and correction is made.The users are trained to operate the developed system.Both hardware and software securities are made to run the developed system successfully.System testing is aimed at ensuring the system works accurately before the live operation commences.

6.1 Test Plan

6.1.1 Scope

This test plan will cover the following testing activities as identified in the testing strategy.

- White Box Testing

White box testing is a testing technique, that examines the program structure and derives test data from the program logic/code. In white box testing, the UI is bypassed. Inputs and outputs of ratings are tested directly at the code level and the results are compared against specifications. It Reveals errors in "hidden" code.

- Black Box Testing

Black - box testing is a method of software testing that examines the functionality of an application based on the specifications. Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end - user would.

- Unit Testing

The module includes four main components to undergo unit testing. It checks the syntax error, logic error and validity of the program correctness. This test will be performed by the developers.

- Integration Testing

After all components pass the unit testing, they get to be tested whether they work correctly when they are running concurrently and communicating to each other. Integration tests exercise an entire subsystem and ensure that a set of components play nicely together.

- Internal Data Testing

We will test the validity of the data before it enters the database to avoid any problems that may face us in the database. We will test the encryption of the personal information of all the users along with the user names and passwords to ensure maximum security of the users privacy.

6.1.2 Software risk issues

In this section, the plan is to test the risk involved in critical issues such as: Difficult to run on visual studio due to large loading time. Some inherent software risks such as complexity exist; also these issues need to be identified. Proper network connection Working of Mysql database

6.1.3 Features to be tested

- Test whether correct user name and password allows you to login.
- Test whether invalid user name and password prevents you from login.
- Test whether there is any connection problem in the Server.
- Test whether the student and admin details are entered correctly.
- Test whether invalid data entry allows saving data successfully.
- Test whether all pages are loaded correctly.
- Test whether watermark is embedded and extracted properly.
- Test whether the provided security works correctly.

6.2 Test consolidation

6.2.1 Test item

The items or features to be tested in the test cases are included in the document. Each and every user input is tested. The present condition of the system is tested. It is checked to make sure that environment is ready for the application to work.

6.2.2 Input specifications

These are the inputs required to execute the test case. All required inputs including data elements and values are given below.

Data	Values
Email	A valid email ID
Password	Combination of characters and numbers
Phone number	10 Digit number

Chapter 7

7 System Implementation and Maintenance

7.1 Implementation

System implementation is the conversion of new system into an operating one which involves creating compatible files, training clients and installing hardware. User training is crucial for minimizing resistance to change and giving chance to prove its worth. Training aids user friendly manuals and healthy screens provide the user with a good start. Software maintenance follows conversion to the extent that changes are necessary to maintain satisfactory operations relative to changes in the user's environment. Maintenance often includes minor enhancements or corrections to the problem that surface late in the systems operations. In the implementation phase, the team builds the components either from scratch or by composition. Given the architecture document meant from the design phase and the requirement document from the analysis phase, the team should build exactly what has been requested, though there is still room for innovation and flexibility. For example, a component may be narrowly designed for this particular system, or the component may be made more general to satisfy a reusability

- Careful planning
- Investigation of system and constraints
- Design the methods to achieve changeover.
- Training the staff in the changed phase.
- Evaluation of change over method.
- The method of implementation and time scale to be adopted are found out initially.

7.2 Maintenance

This phase occurs as a result of deploying the whole system at the end users organization. They will perform the beta testing at the end users and inform to the developers about any needed modification to the application. The customer records all the problems that are encountered during the beta testing and reports these to the developer at regular intervals.

7.2.1 Corrective Maintenance

Even with the best quality assurance activities, it is likely that the customer will uncover defects in the software. Corrective maintenance changes the software to correct the defects. Corrective Maintenance activity may consist of repair,

restoration or replacement of equipment. This activity will be result of a regular inspection, which identifies the failure in time for corrective maintenance to be planned and scheduled, then performed during a routine maintenance shutdown.

7.2.2 Adaptive Maintenance

Over time, the original environment(CPU, operating system, business rules, external product characteristics) for which the software was developed is likely to change. Adaptive maintenance results in modification to the software to accommodate changes to its external environment.

7.2.3 Enhanced Maintenance

As software is used, the customer/user will recognize additional functions that will provide the benefit. Perfect maintenance extends the software beyond its original functional requirements.

7.2.4 Preventive Maintenance

Computer software deteriorates due to change, and because of this preventive maintenance often called software re-engineering, must be conducted to enable the software to serve the needs of its end users. Preventive maintenance makes changes to computer programs so that they can be more easily corrected, adapted and enhanced.

Chapter 8

8 Conclusion and Future Scope

8.1 Conclusion

There are three types of users that interact with the system Admin,Teacher and Student.Each of these three types has different uses of the system so each of them has their own panel.Admin can manage all the features of website dynamically by login on admin panel.Teacher can view the details about their event assigned by the admin and publish the result of the event.And the student can register for events and view the results of the events.

8.2 Future Scope

- Advanced adaptive learning technologies can be implemented.
- Advanced AI Chatbots

Appendix

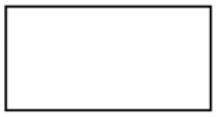
A Data Flow Diagram

Data flow is the one of the best way of documenting the entire functionality of the system.For the system ,which will have data flows in and have some processing inside and then some data flow out from the system can be documented or represented effectively by means of data flow out from the system can be documented or represented effectively by means of data flow diagrams. The data flow diagram are a diagrammatic representation of the system,which has input,process and output.Once any system is represented using a data flow diagram we can identify the following things easily:

- Various entities interacting with the system are identified
- Flow of data from one entity to another is identified
- The various processes involved in between the interaction of two or more entities in the system are clearly pointed out
- The various data stores which hold the data in between the process,are clearly identified

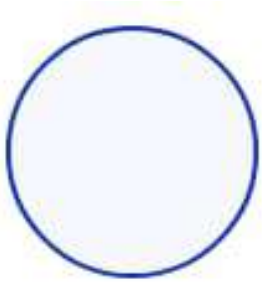
Some Data Flow Diagram charting forms:

A.1 External source or receiver



A source or sink is a person or part of organization ,which enters or receives information from the system,but is considered to be outlining the contest of data flow model.

A.2 Transform process

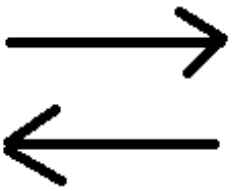


A process represents transformation where incoming data flowsare changed into outgoing data flow

A.3 Data Store

A data store is repository of data that is to be stored for use by one or more process may be as simple as buffer or queue or sophisticated as relational database. They should have clear names.If a process merely uses the contest of store and does not alter it,the arrowhead goes only form the store to the process. If a process alters the details in the store then double-headed arrow is used.

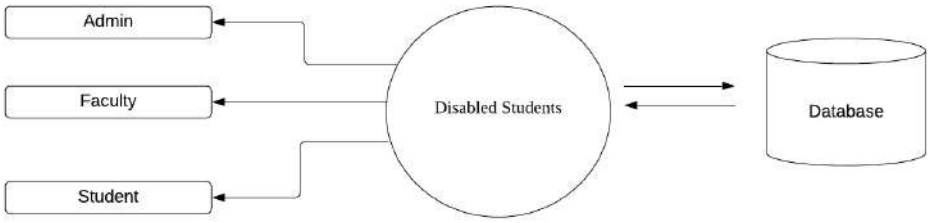
A.4 Data flow



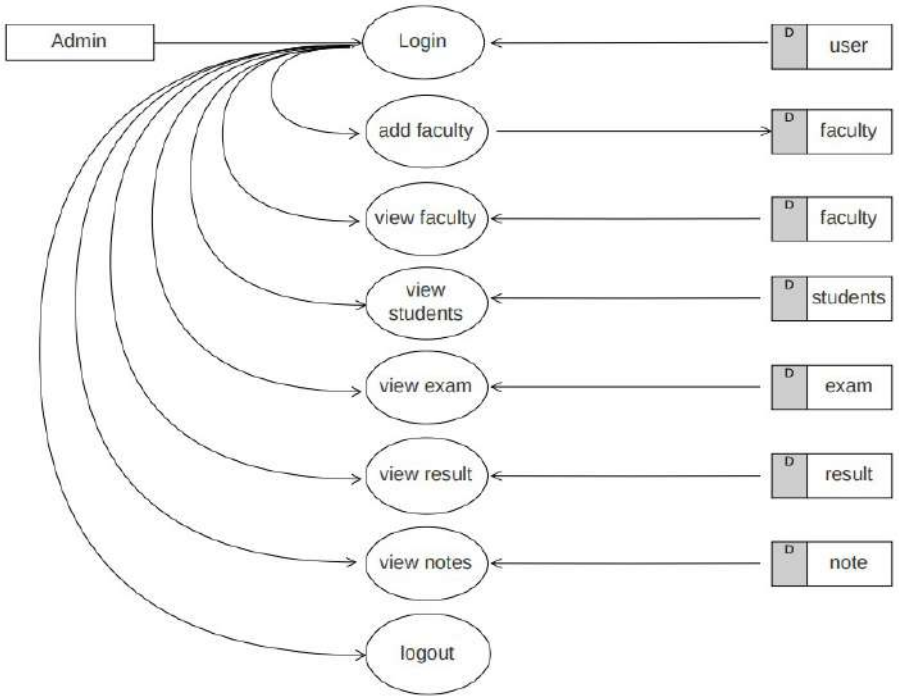
A data flow is a route, which enable packets of data to travel from one point to another.Data may flow,with arrowhead pointing in the direction of the flow

B Data Flow Diagrams

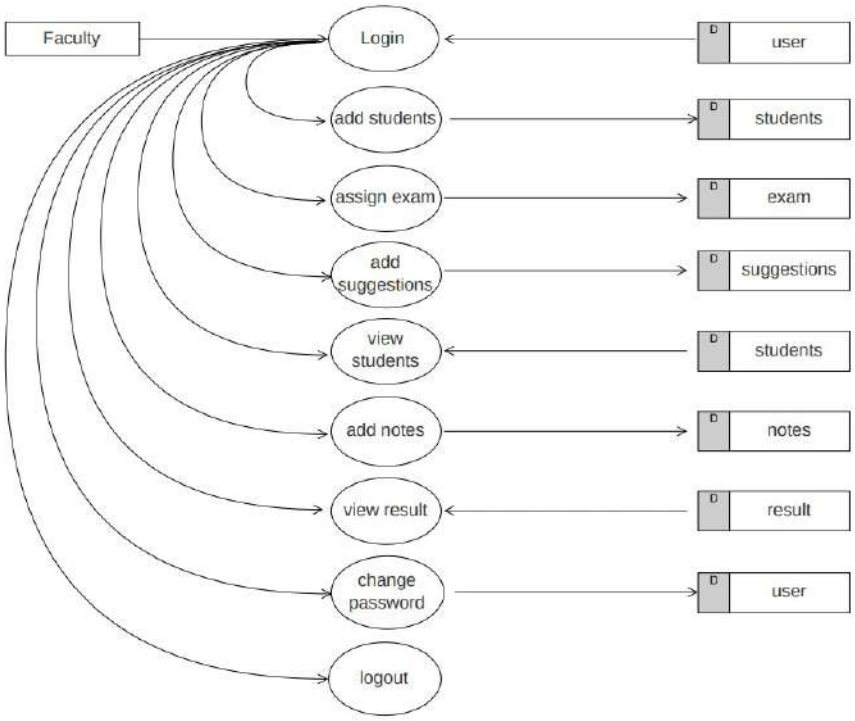
B.1 Level 0



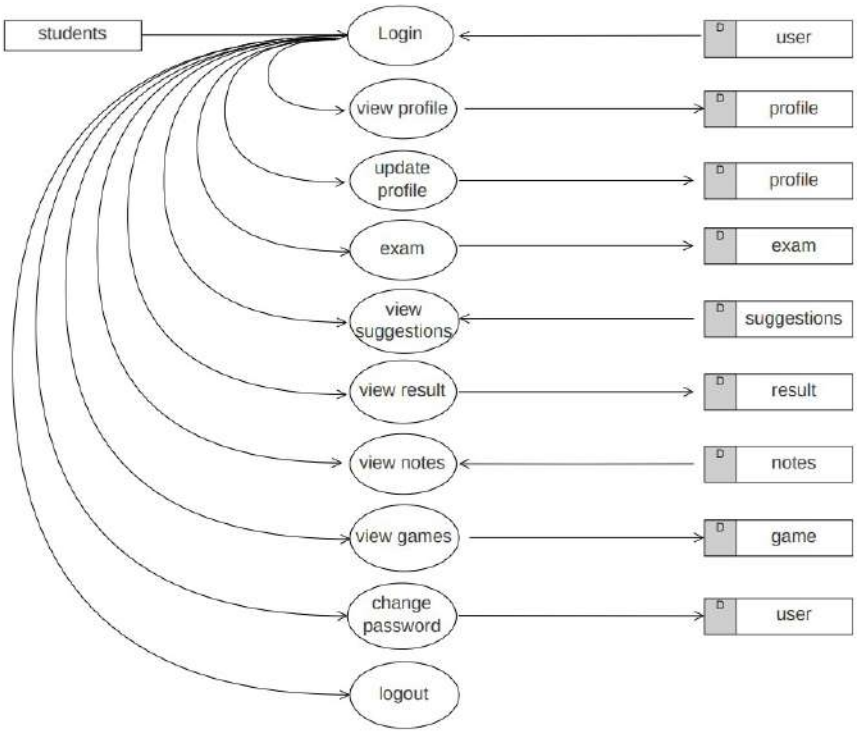
B.2 Level 1.1 - Admin



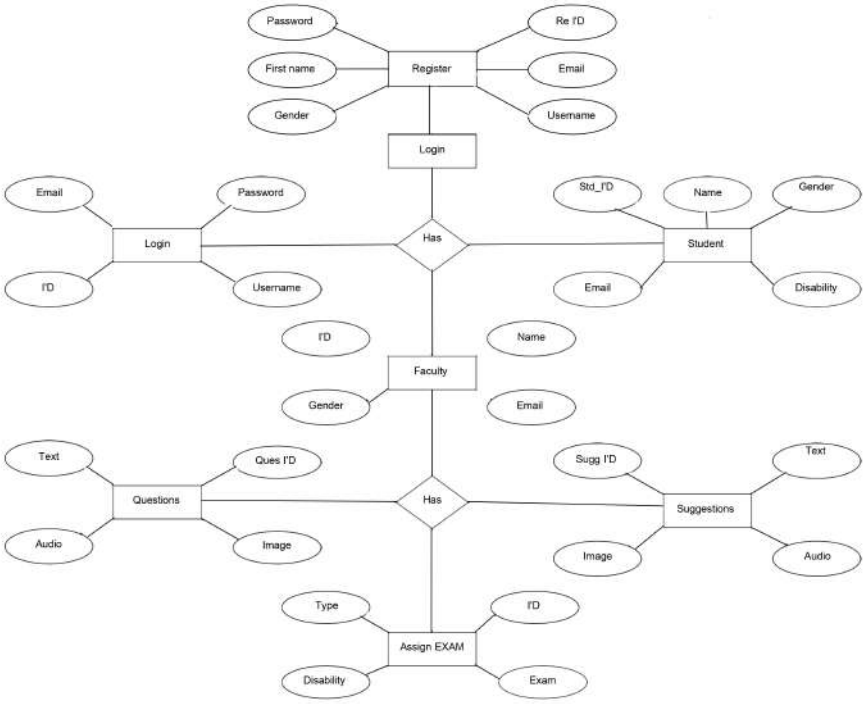
B.3 Level 1.2 - Faculty



B.4 Level 1.3 - Students

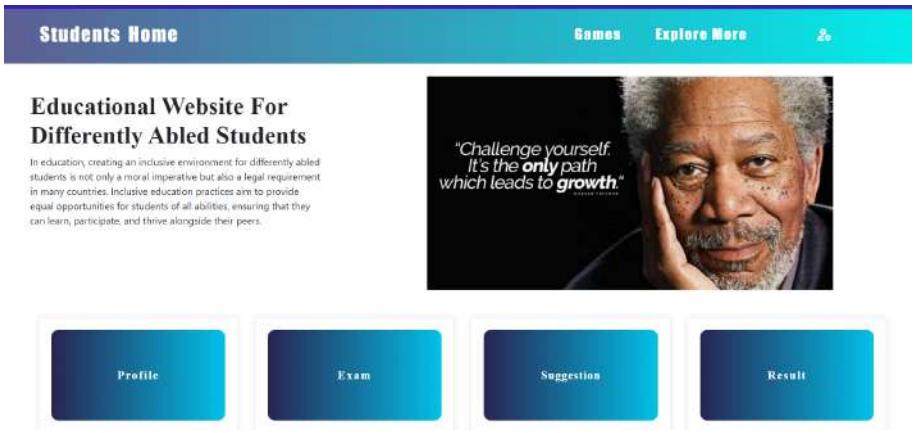


B.5 ER DIAGRAM

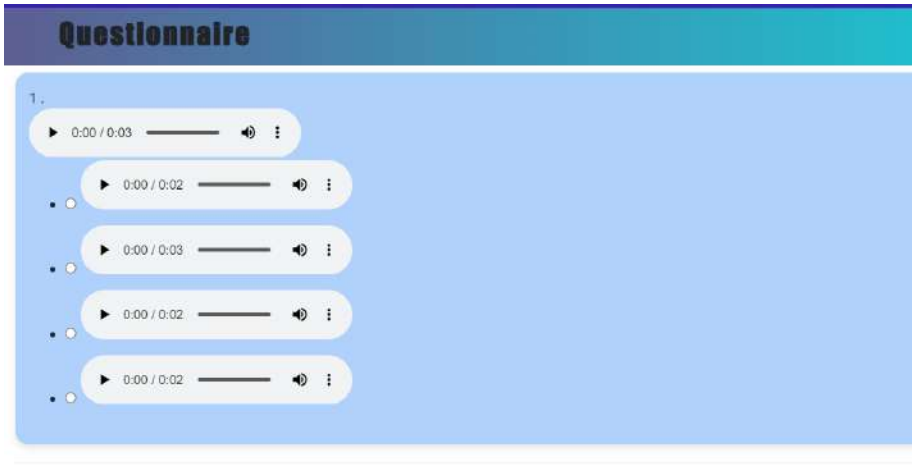


C USER INTERFACES

C.1 HOME



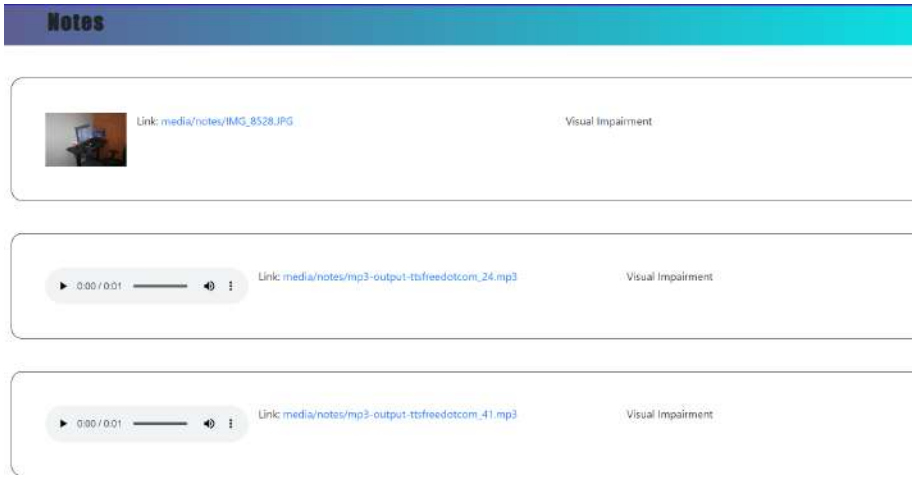
C.2 EXAM



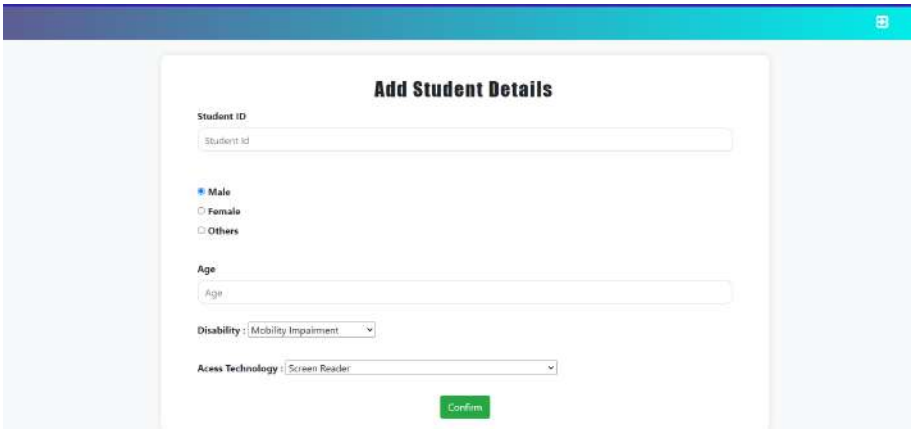
C.3 RESULT


Result
TestScore
1
Category
Very Poor
Result is here Check it....
No.
Question
Result
1
0:00 / 0:03
False
2
0:00 / 0:03
False
3
0:00 / 0:02
False
4
0:00 / 0:06
True

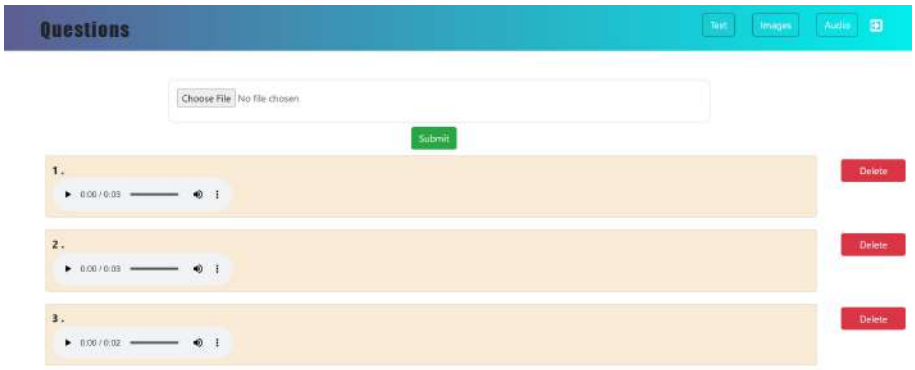
C.4 NOTES


Notes
Link: media/notes/IMG_8528.JPG
Visual Impairment
0:00 / 0:01
Link: media/notes/mp3-output-ttsfreedotcom_24.mp3
Visual Impairment
0:00 / 0:01
Link: media/notes/mp3-output-ttsfreedotcom_41.mp3
Visual Impairment

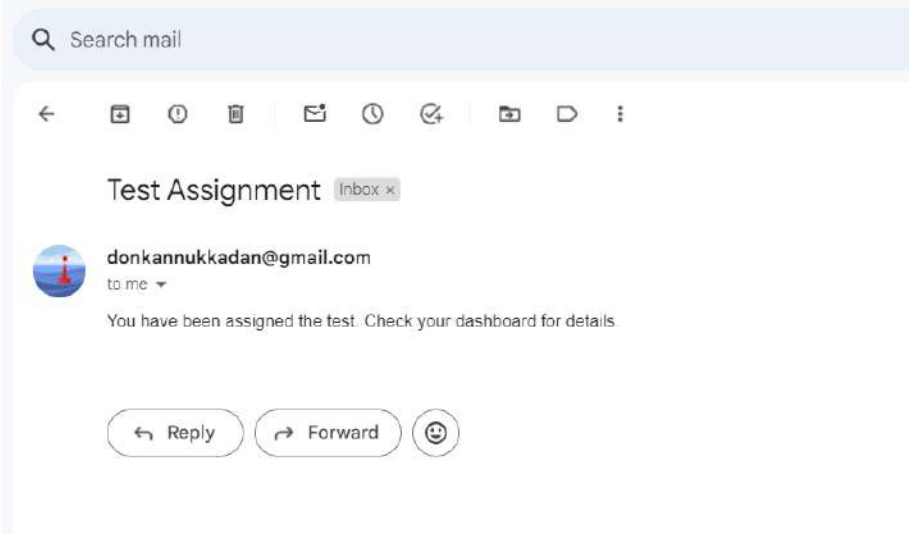
C.5 REGISTRATION



C.6 QUESTIONS WINDOW



C.7 NOTIFICATION



D CODE

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width,
      initial-scale=1.0">
  <!-- <link rel="stylesheet" href="styles.css"> -->
  <script src="toggle-sideNav.js" defer></script>
  <script src="featured-games.js" defer></script>

  <title>Game for mentally challenged people</title>
  <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,
        initial-scale=1, shrink-to-fit=no">
    <!-- Optional JavaScript -->
    <!-- jQuery first, then Popper.js, then Bootstrap JS
        -->
    <script src="https://code.jquery.com/jquery-3.3.1.
        slim.min.js" integrity="sha384-q8i/X+965
        DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8
        abtTE1Pi6jizo" crossorigin="anonymous"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/
        popper.js/1.14.7/umd/popper.min.js" integrity="
        sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9W
      O1clHTMGa3JDZwrnQq4sF86dIHNDz0W1" crossorigin="
          anonymous"></script>
    <script src="https://stackpath.bootstrapcdn.com/
        bootstrap/4.3.1/js/bootstrap.min.js" integrity="
        sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/
        nJGzIxFDsf4x0xIM+B07jRM" crossorigin="anonymous
        "></script>
    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://stackpath.
        bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min
        .css" integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH
        /1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
        crossorigin="anonymous">
    <link rel="stylesheet" href="https://fonts.googleapis
        .com/css2?family=Material+Symbols+Sharp:opsz,wght,
        FILL,GRAD@48,700,0,200" />
```

```
    <link rel="stylesheet" href="https://fonts.googleapis
        .com/css2?family=Material+Symbols+Outlined:opsz,
        wght,FILL,GRAD@24,700,0,200" />
   <link rel="stylesheet" href="https://fonts.googleapis.
       com/css2?family=Material+Symbols+Rounded:opsz,wght,
       FILL,GRAD@40,600,0,200" />
  </head>
<style>
  #play-now{
    text-decoration: none;
  }
</style>
<body>
{% load static %}
<div style="margin-left:95%;text-decoration: none;padding
    : 4px;border-radius: 0.5rem;" ><a href="{% url 'sh'
    %}" class="text-black ml-3" style="color: black;"><
    span class="material-symbols-outlined">
   exit_to_app
  </span></a></div>
    <section class="hero-wrapper">
        <div class="container">
          <p class="greeting " style="color: green;">
            Brain Games For Disabled Students!!!
          </p>
        </div>
      </section>
    <section id="play-now" class="program-wrapper">
      <div class="program-container container">
        <h3 class="title">
            Featured Games
        </h3>
        <div class="program-content">
          <div class="row">
            <div class="col">
                  <a href="{% url 'animal' %}" ><div
                      class="program-detail">
                    <img src="{% static 'images/first.png
                        ' %}" alt="reloading" width="350px
                        " height="450px">
                    <h5 class="mt-1">who am i ?</h5>
                  <p>A great game for the brain! It
                      improves visual scanning, planning,
                      and spatial memory!! </p>
                </div></a>
            </div>
```

```
            <div class="col">
              <a href="{% url 'math' %}"><div class="
                  program-detail">
                <img src="{% static 'images/third.jpg'
                    %}" alt="icon of a speaker" width="550
                    px" height="450px">
                <h5 class="mt-1">Fun with numbers</h5>
              <p>Helps to quickly solve the elementary
                  math problems!! </p>
            </div></a>
            </div>
        </div>
        <div class="row mt-5">
         <div class="col">
           <a href="{% url 'memory' %}"><div class="
               program-detail">
           <img src="{% static 'images/forth.jpg' %}"
              alt="icon of a speaker" width="500px">
           <h5>Behind the scenes</h5>
         <p>Helps to improve Visual Scanning and memory
             while remembering the cards!! </p>
         </div></a>
         </div>
        </div>
         </div>
    </section>
  <script>
   function playWelcomeMessage() {
         const welcomeMessage = new
            SpeechSynthesisUtterance('welcome games');
         window.speechSynthesis.speak(welcomeMessage);
         }
   playWelcomeMessage();
  var recognition = new webkitSpeechRecognition();
  recognition.continuous = true;
  recognition.lang = 'en-US';
  recognition.onresult = function(event) {
    var result = event.results[event.results.length -
        1][0].transcript.toLowerCase();
    console.log("result",result);
    var currentQuestionIndex=0;
     if (result.includes('back to home')) {
        // Redirect to the home page
        window.location.href = '{% url "sh" %}';
    }
};
```

```
 recognition.onerror = function(event) {
      console.error('Speech recognition error:', event.
          error);
  };
  recognition.onend = function() {
      // Restart recognition when it ends
      recognition.start();
  };
  // Start speech recognition
  recognition.start();
  </script>
</body>
</html>
```

views.py

```
from typing import Any
from django.forms.models import BaseModelForm
from django.http import HttpResponse
from django.shortcuts import render,redirect,
    HttpResponseRedirect
from .models import *
from .models import Question
from .forms import *
from django.views.generic import FormView,CreateView,
    UpdateView,TemplateView,View
from django.contrib.auth import authenticate,login,logout
from django.urls import reverse_lazy
from django.contrib.auth.hashers import make_password
from django.http import FileResponse
from django.shortcuts import get_object_or_404
from .models import Suggestion
from student_app.forms import ChangePasswordForm
from django.forms import formset_factory
# Create your views here.

import pandas as pd
import random
from django.db.models.signals import post_save
from django.dispatch import receiver
from django.contrib.auth import get_user_model
from django.http import JsonResponse

class MainHome(TemplateView):
    template_name="mainhome.html"
```

```
class ClearDataView(View):
    def post(self, request):
        StudentAnswer.objects.all().delete()
        StudentAnswerImage.objects.all().delete()
        StudentAnswerAudio.objects.all().delete()
        ScoreModel.objects.all().delete()
        return JsonResponse({'message': 'Data cleared
            successfully'})
@receiver(post_save, sender=Student)
def create_user_from_student(sender, instance, created,
    **kwargs):
    if created:
        # User = get_user_model()
        CustUser= get_user_model()
        student_id_id=instance.id
        username = instance.std_id
        password = 'admin@123'  # Set your desired
            default password here
        # User.objects.create_user(username=username,
            password=password)
        CustUser.objects.create_user(username=username,
            password=password,student_id_id=student_id_id)
        # ScoreModel.objects.create(student_id=
            student_id_id)
class LoginView(FormView):
    template_name="login.html"
    form_class=LogForm
    def post(self,request,*args,**kwargs):
        log_form=LogForm(data=request.POST)
        if log_form.is_valid():
            us=log_form.cleaned_data.get('username')
            ps=log_form.cleaned_data.get('password')
            user=authenticate(request,username=us,
                password=ps)
            if user:
                login(request,user)
                if request.user.is_superuser == 1:
                    return redirect('h')
                else:
                    return redirect('sh')
            else:
                return render(request,'login.html',{"form
                    ":log_form})
        else:
            return render(request,'login.html',{"form":
                log_form})
```

```
class AddStudent(CreateView):
    template_name='addstudent.html'
    model=Student
    form_class=StudentForm
    success_url=reverse_lazy('stu')
class QuestView(TemplateView):
    template_name='test.html'
    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)
        id=kwargs.get('pk')
        context['stu']=Student.objects.get(id=id)
        context['ques']=Question.objects.all()
        return context
# class QuestViewAll(TemplateView):
#     template_name='Assignexam.html'
#     def get_context_data(self, **kwargs):
#         context = super().get_context_data(**kwargs)
#         context['stu']=Student.objects.all()
#         context['ques'] = Question.objects.all()
#         return context

# class QuestViewImageAll(TemplateView):
#     template_name='testallimage.html'
#     def get_context_data(self, **kwargs):
#         context = super().get_context_data(**kwargs)
#         context['stu']=Student.objects.all()
#         context['ques'] = QuestionImages.objects.all()
#         return context

# class QuestViewAudioAll(TemplateView):
#     template_name='testallaudio.html'
#     def get_context_data(self, **kwargs):
#         context = super().get_context_data(**kwargs)
#         context['stu']=Student.objects.all()
#         context['ques'] = QuestionAudio.objects.all()
#         return context

def Test(request,**kwargs):
    if request.method == 'POST':
        id=kwargs.get('pk')
        stu=Student.objects.get(id=id)
        que=Question.objects.all()
        assignment = StudentAnswer(student=stu, question=
            que)
        assignment.save()
        return redirect('det')
```

```
from django.core.mail import send_mail

def AssignView(request,**kwargs):
    if request.method == 'POST':
        # Get all available tests
        tests = Question.objects.all()

        # Get all students
        students = Student.objects.filter(disability__in
            =["Mobility Impairment", "Learning Disability
            ", "Autism Spectrum Disorder", "Speech
            Impairment", "Intellectual Disability"])
        students_with_email_sent = set()
        # Assign all tests to all students
        for test in tests:
            for student in students:

            assignment, created = StudentAnswer.objects.
                get_or_create(student=student, question=
                test)
             if created:
            assignment.save()
            subject = 'Test Assignment'
            message = f'You have been assigned the test.
                Check your dashboard for details.'
            from_email = 'testhelloability@gmail.com'
            to_email = [student.email]
            if student in students_with_email_sent:
                      continue
             send_mail(subject, message, from_email,
                 to_email, fail_silently=False)
             students_with_email_sent.add(student)
        return redirect('testall')

    # Retrieve all available tests
    tests = Question.objects.all()

    return render(request, 'Assignexam.html', {'tests':
        tests})
def AssignImageView(request,**kwargs):
    if request.method == 'POST':
        # Get all available tests

        tests = QuestionImages.objects.all()
```

```
        # Get all students
        students = Student.objects.filter(disability="
            Hearing Impairment")
        students_with_email_sent = set()
        # Assign all tests to all students
        for test in tests:
            for student in students:

                assignment, created = StudentAnswerImage.
                    objects.get_or_create(student=student,
                     question=test)
                if created:
                 assignment.save()
                 subject = 'Test Assignment'
                 message = f'You have been assigned the
                     test. Check your dashboard for
                     details.'
                 from_email = 'testhelloability@gmail.com
                     '
                 to_email = [student.email]
                 if student in students_with_email_sent:
                  continue
                 send_mail(subject, message, from_email,
                     to_email, fail_silently=False)
                 students_with_email_sent.add(student)

        return redirect('testallvisual')

    # Retrieve all available tests
    tests = QuestionImages.objects.all()

    return render(request, 'testallimage.html', {'tests':
        tests})
def AssignAudioView(request,**kwargs):
    if request.method == 'POST':
        # Get all available tests
        tests = QuestionAudio.objects.all()

        # Get all students
        students = Student.objects.filter(disability="
            Visual Impairment")
        students_with_email_sent = set()

        # Assign all tests to all students
        for test in tests:
            for student in students:
```

```
                    assignment, created = StudentAnswerAudio.
                        objects.get_or_create(student=student,
                         question=test)
                    if created:
                     assignment.save()
                     subject = 'Test Assignment'
                     message = 'You have been assigned the
                         test. Check your dashboard for
                         details.'
                     from_email = 'donkannukkadan@gmail.com'
                     to_email = [student.email]
                     if student in students_with_email_sent:
                       continue
                     # Send email
                      send_mail(subject, message, from_email,
                           to_email, fail_silently=False)

                     # Add the student to the set of students
                          with sent emails
                      students_with_email_sent.add(student)

            return redirect('testallhear')

        # Retrieve all available tests
        tests = QuestionAudio.objects.all()

        return render(request, 'testallaudio.html', {'tests':
            tests})

from random import sample

class Quesadd(CreateView):
    template_name="quesadd.html"
    model=Question
    form_class=QuesForm
    success_url=reverse_lazy('qans')
    def get_context_data(self, **kwargs) :
        context = super().get_context_data(**kwargs)
        context['tests'] = Question.objects.all()
        return context

class Quesimage(CreateView):
    template_name="quesimage.html"
    model=QuestionImages
    form_class=QuesFormImage
    success_url=reverse_lazy('qansimg')
```

```
    def get_context_data(self, **kwargs) :
        context = super().get_context_data(**kwargs)
        context['tests'] = QuestionImages.objects.all()
        return context

class Quesaudio(CreateView):
    template_name="quesaudio.html"
    model=QuestionAudio
    form_class=QuesFormAudio
    success_url=reverse_lazy('qansaudio')
    def get_context_data(self, **kwargs) :
        context = super().get_context_data(**kwargs)
        context['tests'] = QuestionAudio.objects.all()
        return context

class QuesUpdate(UpdateView):
    template_name="questionupdate.html"
    model=Question
    form_class=QuesForm
    success_url=reverse_lazy('qdel')
    def get_context_data(self, **kwargs) :
        context = super().get_context_data(**kwargs)
        context['tests']=Question.objects.all()
        return context

class QuesAnsUpdView(CreateView):
    template_name="quesansupdate.html"
    model=Answer
    form_class=QuesAnsForm
    success_url=reverse_lazy('qdel')
    def form_valid(self, form):
        # Save the main answer
        main_answer = form.save(commit=False)
        main_answer.save()

        # Get additional options from the form data
        option_texts = [self.request.POST.get(f'option_{i
            }') for i in range(1, 4)]

        # Create and save additional options
        for option_text in option_texts:
            if option_text:
                answer_option = Answer(question=form.
                    cleaned_data['question'], text=
                    option_text)
                answer_option.save()
```

```
        return super().form_valid(form)

class QuesAnsImageView(CreateView):
    template_name = "quesansimage.html"
    model = AnswerImages
    form_class = QuesAnsFormImg
    success_url = reverse_lazy('qimg')

    def form_valid(self, form):
        # Save the main answer
        main_answer = form.save(commit=False)
        main_answer.save()

        # Handle file uploads for options
        for i in range(1, 4):
            option_file = self.request.FILES.get(f'
                option_{i}')
            if option_file:
                AnswerImages.objects.create(
                    question=main_answer.question,
                    fileans=option_file
                )
        return super().form_valid(form)

class QuesAnsAudioView(CreateView):
    template_name="quesansaudio.html"
    model=AnswerAudio
    form_class=QuesAnsFormAudio
    success_url=reverse_lazy('qaudio')
    def form_valid(self, form):
        # Save the main answer
        main_answer = form.save(commit=False)
        main_answer.save()

        # Handle file uploads for options
        for i in range(1, 4):
            option_file = self.request.FILES.get(f'
                option_{i}')
            if option_file:
                AnswerAudio.objects.create(
                    question=main_answer.question,
                    fileans=option_file
                )

        return super().form_valid(form)
```

```
class DeleteView(View):
    def get(self,req,*args,**kwargs):
        id=kwargs.get('pk')
        dl=Question.objects.get(id=id)
        dl.delete()
        return redirect('qdel')

class DeleteImgView(View):
    def get(self,req,*args,**kwargs):
        id=kwargs.get('pk')
        dl=QuestionImages.objects.get(id=id)
        dl.delete()
        return redirect('qimg')

class DeleteAudioView(View):
    def get(self,req,*args,**kwargs):
        id=kwargs.get('pk')
        dl=QuestionAudio.objects.get(id=id)
        dl.delete()
        return redirect('qaudio')

# class Quesdel(TemplateView):
#     template_name="quesdel.html"
#     def get_context_data(self, **kwargs) :
#         context = super().get_context_data(**kwargs)
#         context['tests']=Question.objects.all()
#         return context

class SugView(TemplateView):
    template_name="suggestions.html"
    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)
        context['data']=Suggestion.objects.all().order_by
            ('cat')
        return context

class SuggTextView(CreateView):
    template_name="suggtext.html"
    model=Suggestion
    form_class=SugForm
    success_url=reverse_lazy('st')

class SuggVideoView(CreateView):
    template_name="suggvideo.html"
    model=Suggestion
```

```
    form_class=SugVideoForm
    success_url=reverse_lazy('sv')

class SuggAudioView(CreateView):
    template_name="suggaudio.html"
    model=Suggestion
    form_class=SugAudioForm
    success_url=reverse_lazy('sadd')

def view_video(request, video_id):
    video = get_object_or_404(Suggestion, pk=video_id)
    video_path = video.video.path
    response = FileResponse(open(video_path, 'rb')) #
        Adjust content_type as needed
    return response

def view_videoo(request, videoo_id):
    video = get_object_or_404(ScoreModel, pk=videoo_id)
    video_path = video.video.path
    response = FileResponse(open(video_path, 'rb')) #
        Adjust content_type as needed
    return response

def play_audio(request, audio_id):
    audio_recording = get_object_or_404(Suggestion, pk=
        audio_id)
    audio_file = audio_recording.audio
    response = FileResponse(open(audio_file.path, 'rb'))
    return response

def play_audioo(request, audio_id):
    audio_recording = get_object_or_404(ScoreModel, pk=
        audio_id)
    audio_file = audio_recording.audio
    response = FileResponse(open(audio_file.path, 'rb'))
    return response

class DeleteViewSug(View):
    def get(self,req,*args,**kwargs):
        id=kwargs.get('pk')
        dl=Suggestion.objects.get(id=id)
        dl.delete()
        return redirect('sadd')

class ChangePasswordViewHome(FormView):
    template_name="changepshome.html"
```

```
    form_class=ChangePasswordForm
    def post(self,request,*args,**kwargs):
        form_data=ChangePasswordForm(data=request.POST)
        if form_data.is_valid():
            current=form_data.cleaned_data.get("
                current_password")
            new=form_data.cleaned_data.get("new_password
                ")
            confirm=form_data.cleaned_data.get("
                confirm_password")
            user=authenticate(request,username=request.
                user.username,password=current)
            if user:
                if new==confirm:
                    user.set_password(new)
                    user.save()
                    logout(request)
                    return redirect("log")
                else:
                    return redirect("cp")
            else:
                return redirect("cp")
        else:
            return render(request,"changepassword.html
                ",{"form":form_data})

class SuggestionUpdateView(UpdateView):
    template_name='suggestionupdate.html'
    model=Suggestion
    form_class=SuggestionForm
    success_url=reverse_lazy('sadd')
```

PRODUCT MANAGEMENT SYSTEM

PROJECT REPORT

Submitted By

ELVIS M.E

Reg. No. CCAVBCA032

for the award of the Degree of

Bachelor of Computer Application (BCA)

in Computer Application
(University of Calicut)

under the guidance of

Ms. Vandana T.V

Assistant Professor

BACHELOR OF COMPUTER APPLICATION
DEPARTMENT OF COMPUTER SCIENCE
CHRIST COLLEGE(Autonomous)
IRINJALAKUDA, KERALA
2021-2024

DEPARTMENT OF COMPUTER SCIENCE

CHRIST COLLEGE (AUTONOMOUS)

IRINJALAKUDA

CERTIFICATE

This is to certify that the project report entitled ***"Product Management System"*** *is a bonafide record of the project work done by* ***Elvis M.E*** *in partial fulfillment of the requirement for the sixth semester of* ***Bachelor of Computer Application*** *in Department of Computer Science of* ***CHRIST COLLEGE (AUTONOMOUS) IRINJALAKUDA***

.

Ms. Vandana T.V
Assistant Professor,CS
Internal Guide

Ms. Sini Thomas
Head of the Department
Computer Science

EXTERNAL EXAMINER

INTERNAL EXAMINER

KALIPARAMBIL STORES

TO WHOMSOEVER IT MAY CONCERN

This is to certify that **Mr.ELVIS M E**
(CCAVBCA032)
student of
CHRIST COLLEGE (AUTONOMOUS), IRINJALAKUDA
(Bachelor Of Computer Application)
has successfully completed their project
PRODUCT MANAGEMENT SYSTEM under the guidance of
Ms. VANDANA TV (guide) and implemented in
KALIPARAMBIL STORES on 29-01-2024
during the academic year 2021-2024.

MUJEEB.KB

KALIPARAMBIL STORES
MANAGER

KALIPARAMBIL STORES

Proprietor

Vellikulangara Junction

DECLARATION

We hereby declare that this project work **"PRODUCT MANAGEMENT SYSTEM"** submitted to Christ College (Autonomous) Irinjalakuda, affiliated to Calicut University in partial fulfillment of the requirement for the award of the Bachelor of Computer Applications, is a record of original work done by us, under the guidance of Ms. VANDANA T.V, Department of computer Science.

Place: Irinjalakuda ELVIS M.E

ACKNOWLEDGEMENT

First and foremost we like to thank Lord almighty for his providence and for being the guiding light throughout the project. We wish to express my sincere gratitude to our beloved Department head for giving me all the facilities for our project. We take this opportunity to express my gratitude to the class teacher Ms. SOWMYA P.S and head of the department Ms. SINI THOMAS who has been supported us throughout the course of this project. We are thankful for her aspiring guidance and valuable advice during the project work. We express my sincere thanks to my project guide Ms. VANDANA T.V for supporting and guiding throughout the project.We would take this opportunity to specially thank all other faculty members for their constant and continuous motivation. Finally we would like to thank my family and friends for giving valuable advice and moral support throughout our project.

ABSTRACT

PRODUCT MANAGEMENT SYSTEM for Retail Stores represents a groundbreaking solution aimed at revolutionizing inventory management and enhancing operational efficiency within the retail sector. This innovative web application offers a comprehensive suite of features, including intuitive inventory management, proactive notifications for expiring products, promotion management capabilities, robust search functionality, seamless integration with point-of-sale systems, and insightful dashboard analytics. Through its user-friendly interface and advanced functionalities, the system empowers retail store owners and employees to optimize inventory processes, drive sales, and make data-driven decisions. With future enhancements focused on supply chain integration, mobile accessibility, and advanced analytics, the proposed system promises to redefine the retail experience and set new standards for success in the industry.

Contents

Chapter 1

1 Introduction

The Product Management System for Retail Stores heralds a new era of efficiency and optimization within the bustling world of retail. This innovative web application emerges as a comprehensive solution tailored to address the complex challenges inherent in managing store inventory and streamlining retail operations. At its core, this project aims to empower retail store owners and employees with a robust platform that seamlessly integrates inventory management, notification systems for expiring products, promotion management, product search functionality, electronic bill generation, and insightful dashboard analytics. By combining cutting-edge technology with intuitive design, the system revolutionizes the way retail businesses approach their day-to-day operations, fostering increased productivity, minimized wastage, and enhanced customer satisfaction. At the forefront of this project lies a user-friendly interface that serves as the gateway to a myriad of powerful features. Through a series of meticulously designed forms and interactive elements, users can effortlessly navigate the system, adding, editing, and removing products while inputting vital details such as product names, expiry dates, quantities, and prices. This meticulous attention to detail ensures that the inventory database remains accurate and up-to-date, providing users with a solid foundation upon which to build their retail endeavors. One of the standout features of the system is its proactive approach to managing expiring products. Through a sophisticated notification system, users receive timely alerts when products are nearing their expiry date, enabling them to take swift action to prevent wastage and capitalize on opportunities to drive sales through targeted promotions.

1.1 Overview

The Product Management System is born from a vision to revolutionize the way retail stores manage their inventory, sales, and operational tasks. With the ever-increasing complexity of product management in modern retail, our team recognized the need for a robust, user-friendly solution that integrates essential features to enhance efficiency and productivity.

Chapter 2

2 System Analysis

2.1 Purpose

This documentation serves as a comprehensive resource for understanding the design, functionality, and implementation of the Product Management System. Whether you are a developer looking to extend the system, a store manager seeking to optimize operations, or an investor evaluating the potential impact of our solution, this document provides the insights and information necessary to engage with our project effectively.

2.1.1 Existing System

The current Product Management System in use at our retail store represents a fundamental tool for managing inventory, sales, and operational tasks. Developed to address the challenges of product management in a retail environment, the system offers a range of essential features aimed at facilitating efficient store operations and enhancing customer satisfaction.

2.1.2 Proposed System

The proposed Product Management System for Retail Stores is a comprehensive solution designed to revolutionize inventory management and streamline retail operations. Key features include an intuitive user interface for efficient inventory management, automated notifications for expiring products, promotion management functionality to drive sales, robust search capabilities for quick access to product information, seamless integration with point-of-sale systems for smooth transactions, and insightful dashboard analytics for data-driven decision-making. Future scope includes integration with supply chain management for automated ordering and replenishment, development of a mobile application version for increased accessibility, and enhancement of analytics capabilities for deeper insights. Overall, the proposed system aims to empower retail store owners and employees with the tools they need to optimize their operations, minimize costs, and enhance customer satisfaction in today's competitive retail environment. Advantages of proposed system :unique page are provided, Automatic expiry notification when product reaches its expiry date and qr code is provided to each product get a overview about the product

2.2 Problem definition

To know what the problem is and what the needs are before developing it. Here we design a web application of retail stores for managing and tracking their product details

2.3 Feasibility study

After the problem is clearly understood and solutions proposed, the next step is to conduct the feasibility study. Feasibility study is defined as evaluation or analysis of the potential impact of the proposed project or program. The objective is to determine whether the proposed system is feasible. There are three aspects of feasibility study to which the proposed system is subjected as discussed below.

2.3.1 Technical Feasibility

Technical feasibility assess whether the current technical resources are sufficient for the new system. We can upgrade the level of technology for supporting our webapplication. We check whether the proposed system can be implemented in the present system without supporting the existing hardware.

2.3.2 Economical Feasibility

Economic feasibility determines whether the time and money are available to develop the system. There is no additional hardware used to develop the site, it is inexpensive to build.

2.3.3 Operational Feasibility

Operational feasibility determines if the human resources are available to operate the system once it has been installed. No extra training is needed to use this system. Anyone who has knowledge in internet in english language can easily use the system. The resources that are required to implement or install are already available with the organization.

Chapter 3

3 Software Requirement Specification

3.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the PRODUCT MANAGEMENT SYSTEM. It illustrate the purpose and complete description for the development of the system. It explain system constraints, interface and interactions with other external applications. This document is primarily intended to be proposed to a store for managing and tracking the product details and help the user to gain profit.

3.2 Scope

Our project has made it easier for store to manage products easily and systematically . We can get all information about the product by having a look at this web application. In the future drone delivery can also be included.

3.3 Overall Description

The system will include features such as user authentication, product management, expiry date notifications, discount management, electronic bill monitoring, and sales analytics, providing stakeholders with real-time insights into product performance and sales trends. By prioritizing non-functional requirements such as performance, security, usability, and reliability, the system will offer a secure, scalable, and user-friendly solution tailored to the needs of retail environments. With a focus on minimizing waste, optimizing inventory turnover, and enhancing customer satisfaction, this project seeks to empower retail businesses to thrive in an increasingly competitive market landscape.

3.3.1 Product Perspective

From a product perspective, the Product Management System for Retail Stores serves as a crucial tool to address the specific needs and challenges faced by retail businesses in managing their inventory, sales, and operational tasks. The system provides a centralized platform for store personnel to efficiently manage product information, monitor stock levels, and make informed decisions regarding pricing, promotions, and inventory optimization. With features such as expiry date notifications and dynamic discount management, the system enables proactive measures to minimize waste and maximize profitability. Additionally, the integration of sales analytics capabilities empowers stakeholders to gain valuable insights into product performance, customer preferences, and market trends, facilitating data-driven decision-making and strategic planning. By offering a comprehensive suite of functionalities and prioritizing user experience, security,

and scalability, the system aims to enhance operational efficiency, drive business growth, and ultimately contribute to the success and sustainability of retail businesses.

3.3.2 Product Functionality

Through this system admin can include various product data.User can getoverall control of the web application

3.3.3 Users and Characteristics

There are two types of users that interact with the site admin and employee. Both of them has the priviledge to login and register into web application.Both ofthe usercan add product.But only admin have the priviledge to access into the database

3.4 Specific Requirements

3.4.1 Hardware Requirements

- System: IBM-Compatible PC
- Processor: Pentium IV or above
- Speed: Above 1GHz
- RAM capacity: 512 MB
- Hard Dsk drive: 40 GB
- Keyboard: Standard
- Mouse: Standard
- Monitor: SVGA Color

3.4.2 Software Requirements

- Operating System:Windows or ubuntu
- Languages used: Python Django
- Database : MySql
- Technologies used: HTML,Javascript,CSS,Bootstrap

3.5 Functional Requirements

It contains three main modules.

- 1.Admin
- 2.User or Employee

Admin

An Admin account is used for editing or managing the web application dynamically by Admin panel. The admin can add new products, and have the access to database.

User

The user or The employee can register ar login the web apllication. Also the user can add product into the database and for the current status of the product. but they did not have the right to access into the database.

3.6 Non Functional Requirements

Non-functional requirements define the overall qualities or attributes of the resulting system. Non-functional system place restrictions on the product being developed, the developed process, and specify external constraints that the product must meet. Examples of non-functional requirements include safety, security, usability, reliability and performance requirements. Project management issues (costs, time and schedule) are often considered as non-functional requirements. The principal non - functional constraints which are relevant to critical systems :

- performance
- security
- safety
- usability

Performance

Performance requirements concern the speed of operation of a system.Type of performance requirements :

- Response requirements (how quickly the system reacts to a user input).
- Throughput requirements (how much the system can accomplish within a specified amount of time).
- Availability requirements (is the system available for service when requested by end users). The speed of operation of this system is adequate for the requirements.

Reliability

- Reliability is the ability of a system to perform its required function under stated conditions for a specified period of time.

- constraints on the runtime behavior of the system. This system is reliable because its functionalities can be done on the required conditions.

Safety

Safety requirements are not required which exclude unsafe situation from the possible solution of the system.

Usability

Usability is the ease with which a user can learn to operate, prepare inputs for, and interpret outputs of system or components. Usability requirements include :

- information error messages.
- well-formed user interfaces.

3.7 Interface Requirements

3.7.1 Hardware interfaces

The system must run over the internet, all the hardware shall require to connect internet will be hardware interface for the system. As for example modem, WAN-LAN.

3.7.2 Software interfaces

Software interface required for the working of the project is the appropriate operating system.

3.7.3 Communication interfaces

The user of the site communicate using the network connectivity and the data set is accessed

3.8 Security Requirements

- User accesses only their account.
- Validation of input is handled.
- This application containing the computer systems is physically secured against arms or surreptitious entry by intruders.
- Users must be authorized carefully to reduce changes of any such user giving access to an intruder in exchange for a bribe or other favour.

3.9 Platform Used

Windows 10 is a major version of the Microsoft Windows operating system that was released on July 29, 2015.It is built on the Windows NT kernel and follows windows 8.Part of the reason Microsoft decided to name the 2015 release "Windows 10"(and skipped "windows 9")is because the operating system is designed to be a new direction for Microsoft. One of the primary aims of windows 10 is to Unify the windows experience across multiple devices, such desktop computers, tablets, and smartphones. As part of this effort, Microsoft developed Windows10 Mobile alongside Windows 10 to replaces Windows Phone - Microsoft's previous mobile OS. Windows 10 also integrates other Microsoft services, such as Xbox Live and the Cortana voice recognition assistant.

3.10 Technologies Used

Python Django

Python Django is a powerful web framework that facilitates rapid development and clean, pragmatic design. It follows the "don't repeat yourself" (DRY) principle, enabling developers to build complex web applications with efficiency and simplicity. Django's batteries-included philosophy means it comes with many built-in features and functionalities, including an ORM (Object-Relational Mapping) for database interactions, a robust authentication system, and a powerful templating engine. Its versatility allows developers to create various types of web applications, from content management systems to e-commerce platforms. Django's scalability and security features make it a popular choice among developers for building secure and scalable web applications. Additionally, Django's active community and extensive documentation provide ample support for developers at all levels, making it an excellent framework for both beginners and experienced developers alike.

MySQL

MySQL, pronounced either "My S-Q-L" or "My Squel, " is an open source relational database management system. It is based on the structure query language (SQL), which is used for adding, removing and modifying information in the database. Standard SQL commands, such as ADD, DROP, INSERT,and UPDATE can be used with MySQL.

What distinguishes Python Django from something like client - side Javascript is that the code is executed on the server, generating HTML which is then sent to the client. The client would receive the results of running that script, but would not know what the underlying code was. You can even configure your web server to process all your HTML files with Python Django, and then there's really no way that users can tell what you have up your sleeve. The best things in using Python Django are that it is extremely simple for a newcomer, but offers many advanced features for a professional programmer.

Chapter 4

4 Design Document

4.1 Purpose

The Product Management System for Retail Stores web application serves as a comprehensive solution designed to optimize inventory management and enhance operational efficiency in retail settings. Its primary objectives include streamlining inventory management tasks by providing a centralized platform for adding, editing, and deleting products, as well as sending timely notifications for expiring products to prevent wastage. Additionally, the application facilitates promotion management, enabling users to offer discounts or special deals on expiring products to drive sales. It enhances the customer experience by offering features such as product search functionality and electric bill generation, ensuring a smooth transaction process. Furthermore, the application provides valuable insights through its dashboard analytics, empowering store owners to make informed decisions and optimize business strategies based on data-driven insights.

4.2 Scope

The scope of the Product Management System for Retail Stores web application encompasses a range of functionalities aimed at optimizing inventory management and enhancing retail operations. It includes features such as efficient inventory management through adding, editing, and deleting products, automated expiry notifications to prevent wastage, promotion management for expiring products to drive sales, product search functionality for quick access to information, electric bill generation for seamless transactions, and dashboard analytics providing valuable insights into store performance. By incorporating these features, the application aims to streamline processes, improve efficiency, and empower store owners with data-driven decision-making capabilities in their retail endeavors.

4.3 Overview

The purpose of this document is to help the reader to visualize the solution to the project presented.This document verifies how the design meet the requirement stimulated in the SRS document through design viewpoints.The design viewpoints will cover all design elements presented before. This document will provide a direct approach to the development of this project hence reducing feature creep and ponitedly determine the quality of the design.

4.4 Data Design

Database are the storehouses of data used in the software systems.The data is stored in tables inside the database.Several tables are created for the manipulation of the data for the system.Two essential settings for a database are

- Primary Key- The field that is unique for all the record occurrence.
- Foreign Key- The field used to set relation between tables.

Normalization

- Normalization is the process of organizing the data in the database.
- Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate the undesirable characteristics like Insertion, Update and Deletion Anomalies.
- Normalization divides the larger table into the smaller table and links them using relationship.
- The normal form is used to reduce redundancy from the database table.

Tables

Products

Name	DataType	Constraints	Description
id	int(11)	Primarykey	ID of product
name	varchar(25)	Notnull	Name of product
category	varchar(100)	Notnull	category of product
quantity	int(11)	Notnull	quantity of product
price	int(20)	Notnull	Prices of product
expiry_date	date	Notnull	expiry date of product
vendor	varchar(250)	Notnull	Name of vendor

Sales

Name	DataType	Constraints	Description
id	int(11)	Primarykey	ID of product
item	varchar(50)	Notnull	Name of item
price_item	int(11)	Notnull	Price of item
total_price	int(11)	Notnull	Total price of item
payment method	varchar(200)	Notnull	Payment method for product
customer name	varchar(20)	Notnull	Name of the customer
date	date	Notnull	Date of product purchase

Bill

Name	DataType	Constraints	Description
item	varchar(25)	Notnull	Item name of the product
quantity	int(9)	Notnull	Quantity of product
price	int(9)	Notnull	Price of the product
payment_method	varchar(25)	Notnull	payment method for the product

Staff

Name	DataType	Constraints	Description
id	int(11)	Primarykey	id of staff
user name	varchar(25)	Notnull	Name of staff
phone number	int(10)	Notnull	Phone number of the staff
status	varchar(30)	Notnull	Status of the staff
role	varchar(10)	Notnull	Role of the staff

Expiry date

Name	DataType	Constraints	Description
id	int(9)	Primarykey	ID of the product
name	varchar(25)	Foreignkey	Name from Product table
category	varchar(30)	Foreignkey	Category from Product table
quantity	int(11)	Foreignkey	Quantity from Product table
price	int(11)	Foreignkey	Price from Product table
expiry date	date	Foriegnkey	Expiry date from Product table
vendor	varchar(10)	Foriegnkey	Vendor from Product table

Chapter 5

5 Development of the System

The development of the Product Management System for Retail Stores web application involves utilizing a combination of front-end technologies like HTML, CSS, and JavaScript alongside a back-end framework such as Django or Flask, supported by a suitable database management system. Following an agile methodology, the development process encompasses stages of requirement gathering, system design, implementation, testing, deployment, and ongoing maintenance. Collaboration and communication among team members, stakeholders, and end-users are prioritized throughout, facilitated by project management tools and regular meetings. Security considerations are paramount, with the implementation of best practices for secure coding, data encryption, and access control to safeguard sensitive information. Ultimately, the development aims to deliver a robust, user-friendly solution that optimizes inventory management, enhances operational efficiency, and ensures data security in retail environments.

Chapter 6

6 System Testing

Testing is the penultimate step of software development.An elaborate testing of the data is prepared and the system is using the test data,while doing testing,errors are noted and correction is made.The users are trained to operate the developed system.Both hardware and software securities are made to run the developed system successfully.System testing is aimed at ensuring the system works accurately before the live operation commences.

6.1 Test Plan

6.1.1 Scope

This test plan will cover the following testing activities as identified in the testing strategy.

- White Box Testing

White box testing is a testing technique, that examines the program structure and derives test data from the program logic/code. In white box testing, the UI is bypassed. Inputs and outputs of ratings are tested directly at the code level and the results are compared against specifications. It Reveals errors in "hidden" code.

- Black Box Testing

Black - box testing is a method of software testing that examines the functionality of an application based on the specifications. Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end - user would.

- Unit Testing

The module includes four main components to undergo unit testing. It checks the syntax error, logic error and validity of the program correctness. This test will be performed by the developers.

- Integration Testing

After all components pass the unit testing, they get to be tested whether they work correctly when they are running concurrently and communicating to each other. Integration tests exercise an entire subsystem and ensure that a set of components play nicely together.

- Internal Data Testing

We will test the validity of the data before it enters the database to avoid any problems that may face us in the database. We will test the encryption of the personal information of all the users along with the user names and passwords to ensure maximum security of the users privacy.

6.1.2 Software risk issues

In this section, the plan is to test the risk involved in critical issues such as: Difficult to run on visual studio due to large loading time. Some inherent software risks such as complexity exist; also these issues need to be identified. Proper network connection Working of Mysql database

6.1.3 Features to be tested

- Test whether correct user name and password allows you to login.
- Test whether invalid user name and password prevents you from login.
- Test whether there is any connection problem in the Server.
- Test whether the student and admin details are entered correctly.
- Test whether invalid data entry allows saving data successfully.
- Test whether all pages are loaded correctly.
- Test whether watermark is embedded and extracted properly.
- Test whether the provided security works correctly.

6.2 Test consolidation

6.2.1 Test item

The items or features to be tested in the test cases are included in the document. Each and every user input is tested. The present condition of the system is tested. It is checked to make sure that environment is ready for the application to work.

6.2.2 Input specifications

These are the inputs required to execute the test case. All required inputs including data elements and values are given below.

Data	Values
Email	A valid email ID
Password	Combination of characters and numbers
Phone number	10 Digit number

Chapter 7

7 System Implementation and Maintenance

7.1 Implementation

System implementation is the conversion of new system into an operating one which involves creating compatible files, training clients and installing hardware. User training is crucial for minimizing resistance to change and giving chance to prove its worth. Training aids user friendly manuals and healthy screens provide the user with a good start. Software maintenance follows conversion to the extent that changes are necessary to maintain satisfactory operations relative to changes in the user's environment. Maintenance often includes minor enhancements or corrections to the problem that surface late in the systems operations. In the implementation phase, the team builds the components either from scratch or by composition. Given the architecture document meant from the design phase and the requirement document from the analysis phase, the team should build exactly what has been requested, though there is still room for innovation and flexibility. For example, a component may be narrowly designed for this particular system, or the component may be made more general to satisfy a reusability

- Careful planning
- Investigation of system and constraints
- Design the methods to achieve changeover.
- Training the staff in the changed phase.
- Evaluation of change over method.
- The method of implementation and time scale to be adopted are found out initially.

7.2 Maintenance

This phase occurs as a result of deploying the whole system at the end users organization. They will perform the beta testing at the end users and inform to the developers about any needed modification to the application. The customer records all the problems that are encountered during the beta testing and reports these to the developer at regular intervals.

7.2.1 Corrective Maintenance

Even with the best quality assurance activities, it is likely that the customer will uncover defects in the software. Corrective maintenance changes the software to correct the defects. Corrective Maintenance activity may consist of repair,

restoration or replacement of equipment. This activity will be result of a regular inspection, which identifies the failure in time for corrective maintenance to be planned and scheduled, then performed during a routine maintenance shutdown.

7.2.2 Adaptive Maintenance

Over time, the original environment(CPU, operating system, business rules, external product characteristics) for which the software was developed is likely to change. Adaptive maintenance results in modification to the software to accommodate changes to its external environment.

7.2.3 Enhanced Maintenance

As software is used, the customer/user will recognize additional functions that will provide the benefit. Perfect maintenance extends the software beyond its original functional requirements.

7.2.4 Preventive Maintenance

Computer software deteriorates due to change, and because of this preventive maintenance often called software re-engineering, must be conducted to enable the software to serve the needs of its end users. Preventive maintenance makes changes to computer programs so that they can be more easily corrected, adapted and enhanced.

Chapter 8

8 Conclusion and Future Scope

8.1 Conclusion

In conclusion, the Product Management System for Retail Stores stands as a transformative solution poised to redefine how retail businesses operate in the modern landscape. By combining intuitive design with powerful features such as inventory management, expiring product notifications, promotion management, and insightful analytics, the system empowers store owners and employees to streamline operations, minimize wastage, and drive profitability. With future enhancements focused on integration with supply chain management, mobile accessibility, and advanced analytics, the system is well-positioned to adapt and evolve alongside the changing needs of the retail industry. Ultimately, the proposed system promises to revolutionize the retail experience, fostering increased efficiency, agility, and customer satisfaction in an ever-evolving marketplace.

8.2 Future Scope

- Drone delivery
- Expansion to Multi-location Support
- Implementation of Customer Relationship Management (CRM) Features
- Integration with IoT Devices
- Integration with Supply Chain Management

Appendix

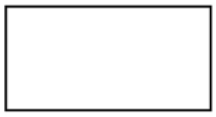
A Data Flow Diagrams

Data flow is the one of the best way of documenting the entire functionality of the system.For the system ,which will have data flows in and have some processing inside and then some data flow out from the system can be documented or represented effectively by means of data flow out from the system can be documented or represented effectively by means of data flow diagrams. The data flow diagram are a diagrammatic representation of the system,which has input,process and output.Once any system is represented using a data flow diagram we can identify the following things easily:

- Various entities interacting with the system are identified
- Flow of data from one entity to another is identified
- The various processes involved in between the interaction of two or more entities in the system are clearly pointed out
- The various data stores which hold the data in between the process,are clearly identified

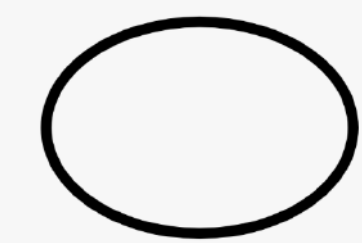
Some Data Flow Diagram charting forms:

A.1 External source or receiver



A source or sink is a person or part of organization ,which enters or receives information from the system,but is considered to be outlining the contest of data flow model.

A.2 Transform process

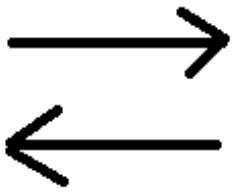


A process represents transformation where incoming data flowsare changed into outgoing data flow

A.3 Data Store

A data store is repository of data that is to be stored for use by one or more process may be as simple as buffer or queue or sophisticated as relational database. They should have clear names.If a process merely uses the contest of store and does not alter it,the arrowhead goes only form the store to the process. If a process alters the details in the store then double-headed arrow is used.

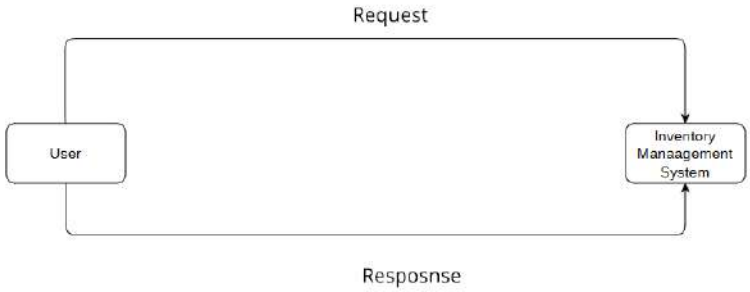
A.4 Data flow



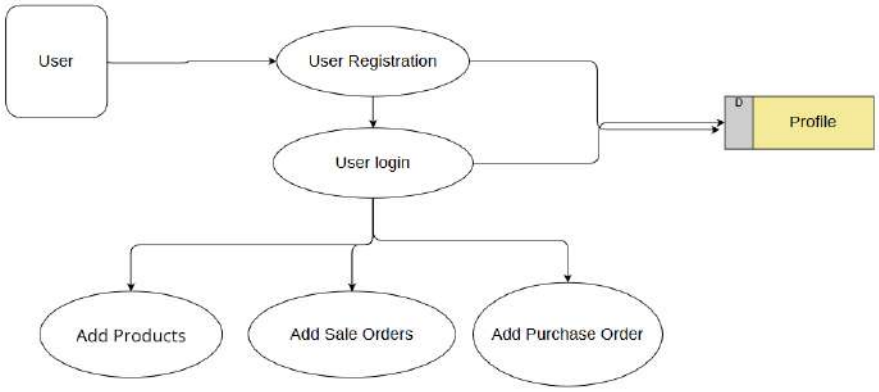
A data flow is a route, which enable packets of data to travel from one point to another.Data may flow,with arrowhead pointing in the direction of the flow

B Data Flow Diagrams

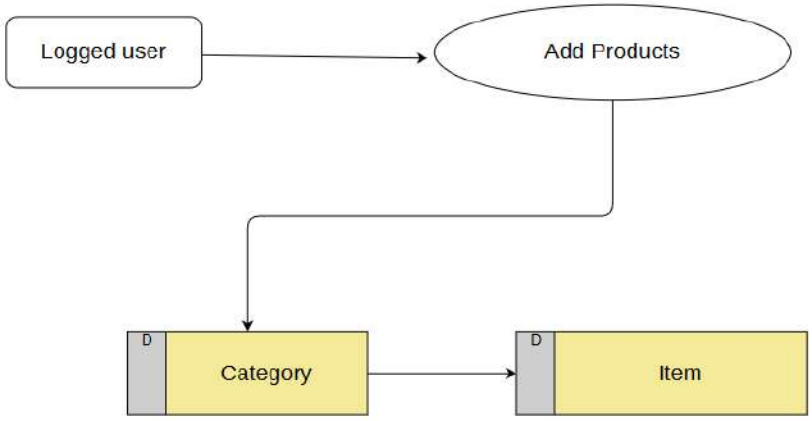
B.1 Level 0



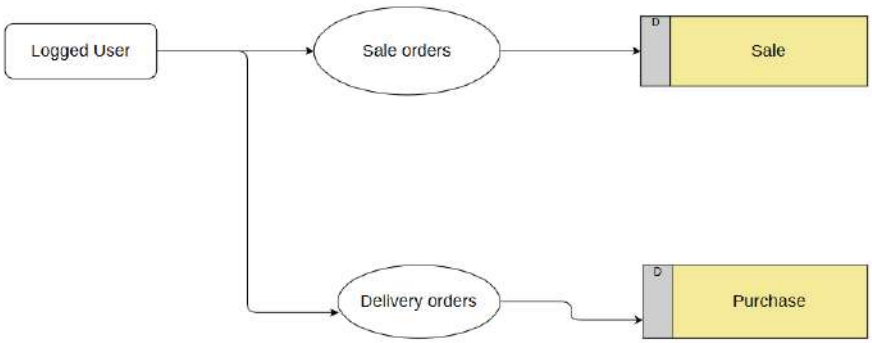
B.2 Level 1


User
User Registration
User login
D
Profile
Add Products
Add Sale Orders
Add Purchase Order

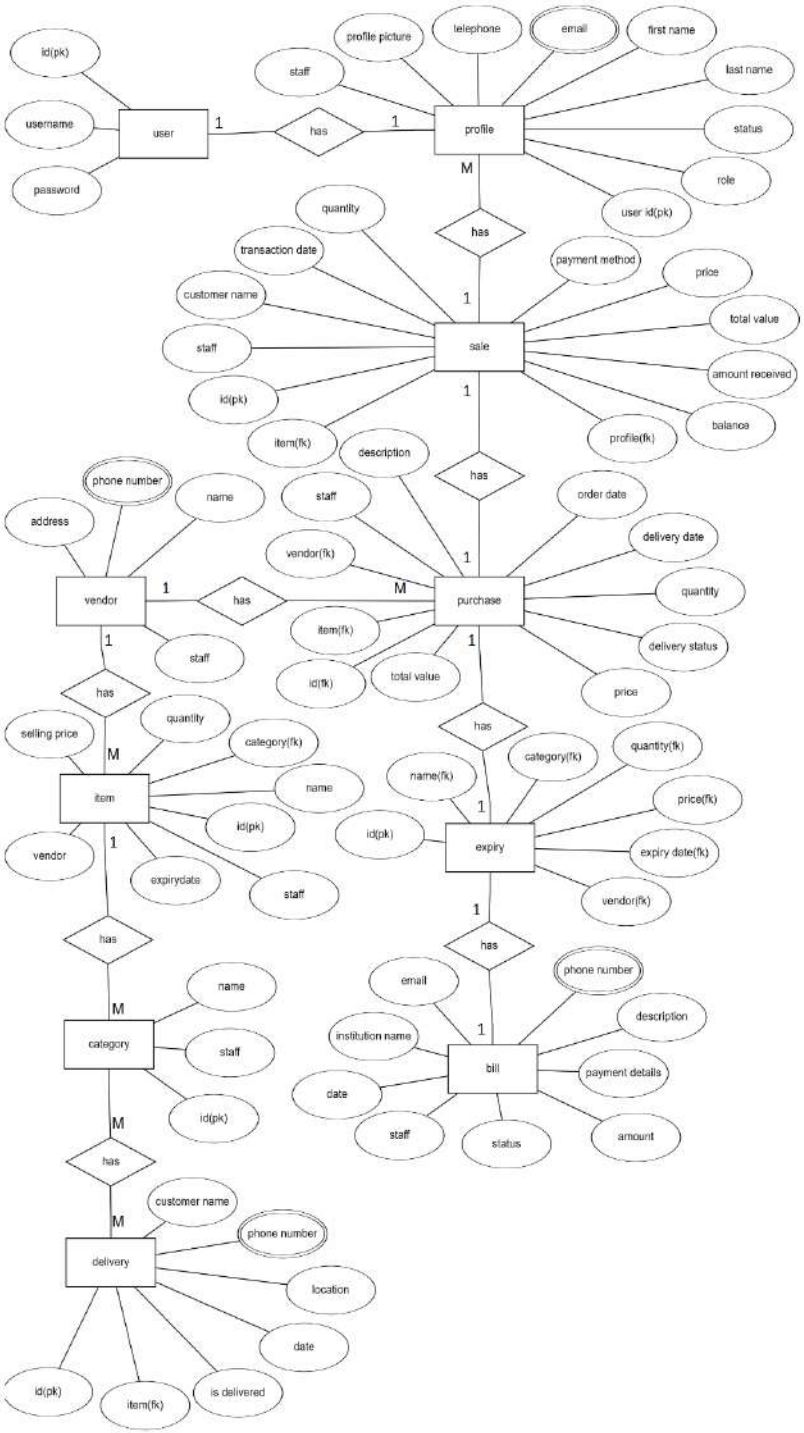
B.3 Level 2



B.4 Level 3

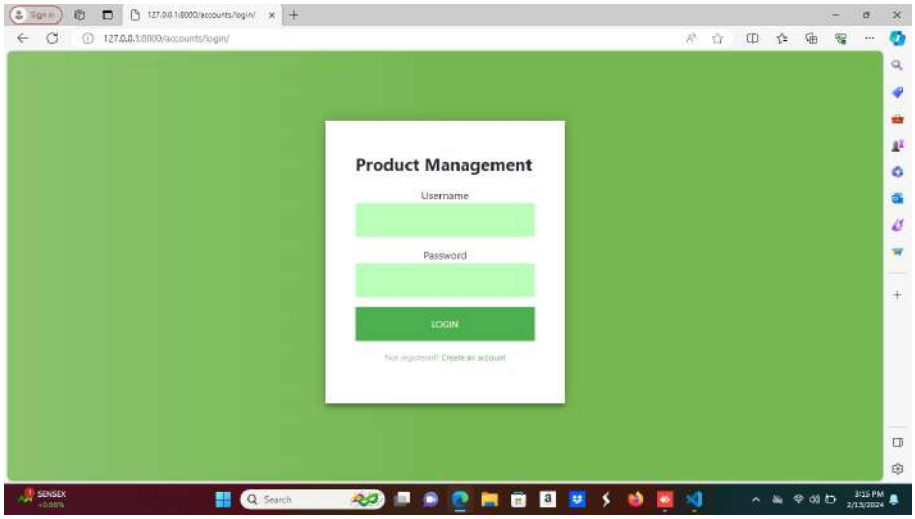


ER Diagram

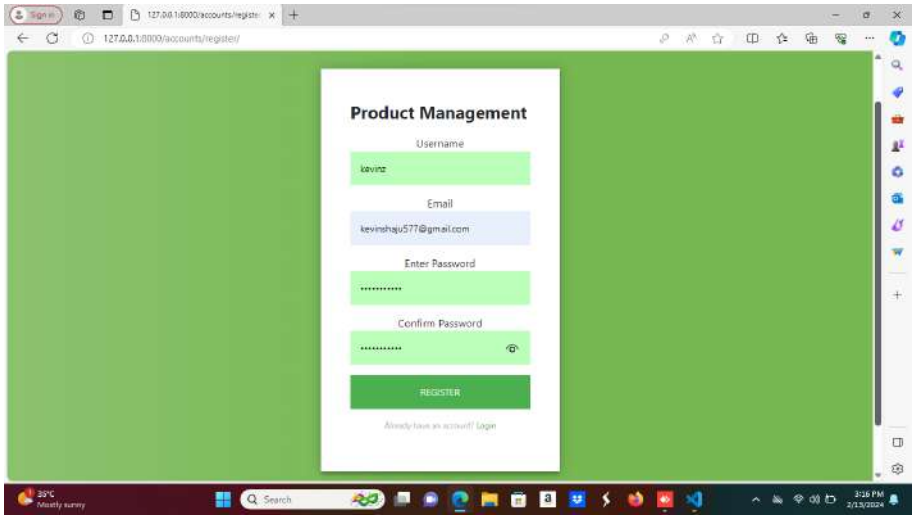

id(pk)
username
password
user
1
has
1
staff
profile picture
telephone
email
first name
last name
profile
status
role
user id(pk)
M
has
quantity
transaction date
customer name
staff
id(pk)
item(fk)
description
sale
payment method
price
total value
amount received
balance
profile(fk)
1
has
phone number
name
address
vendor
1
has
M
staff
vendor(fk)
order date
delivery date
quantity
delivery status
price
purchase
item(fk)
id(fk)
total value
1
has
selling price
quantity
category(fk)
name
id(pk)
item
M
1
vendor
expirydate
staff
has
name(fk)
category(fk)
quantity(fk)
price(fk)
expiry date(fk)
vendor(fk)
id(pk)
expiry
1
has
name
staff
id(pk)
category
M
has
email
phone number
description
institution name
payment details
date
staff
status
amount
bill
customer name
phone number
location
date
delivery
id(pk)
item(fk)
is delivered

C USER INTERFACES

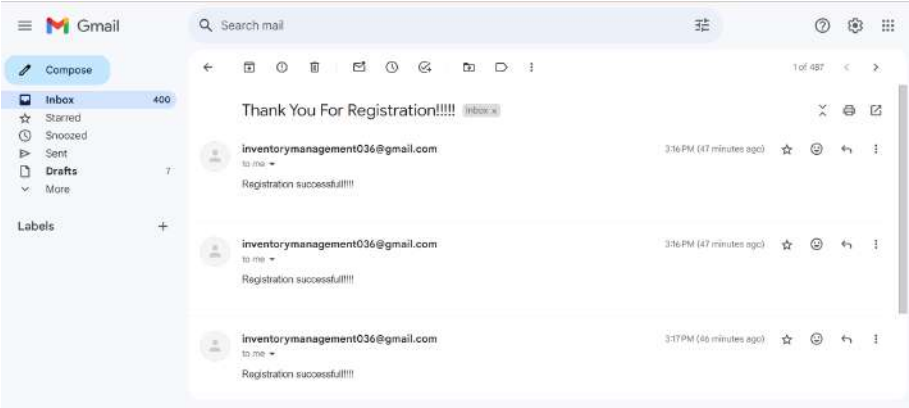
C.1 LOGIN



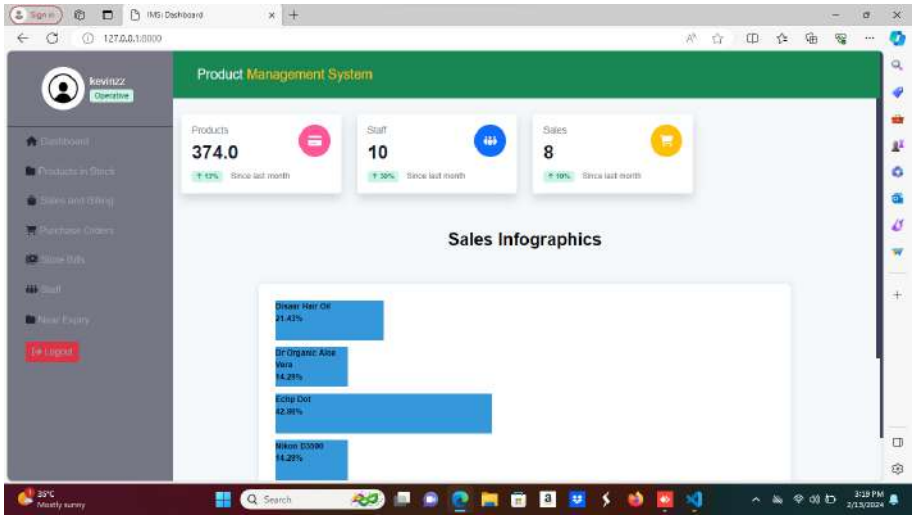
C.2 REGISTER


Product Management
Username
kevinz
Email
kevinshaju577@gmail.com
Enter Password
Confirm Password
REGISTER

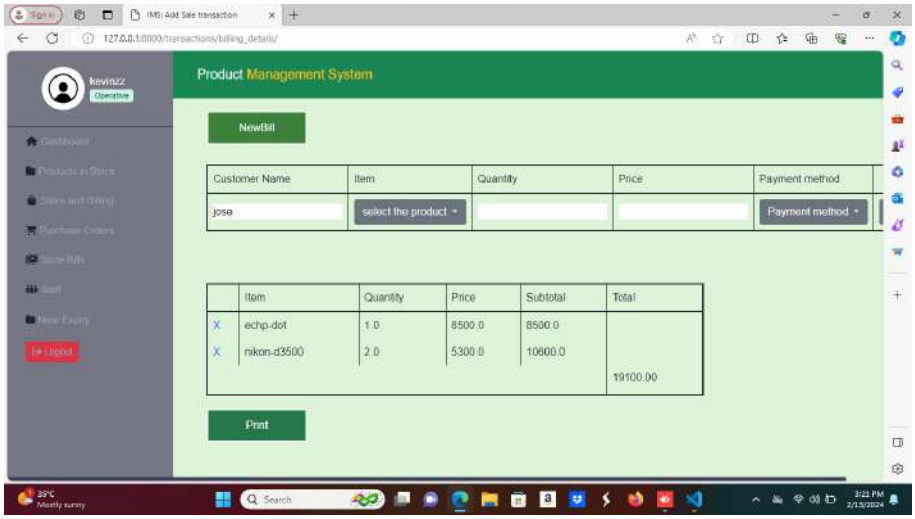
C.3 MAIL


Gmail
Search mail
Compose
Inbox 400
Starred
Snoozed
Sent
Drafts 7
More
Labels
Thank You For Registration!!!!!
inventorymanagement036@gmail.com
Registration successful!!!!
inventorymanagement036@gmail.com
Registration successful!!!!
inventorymanagement036@gmail.com
Registration successful!!!!

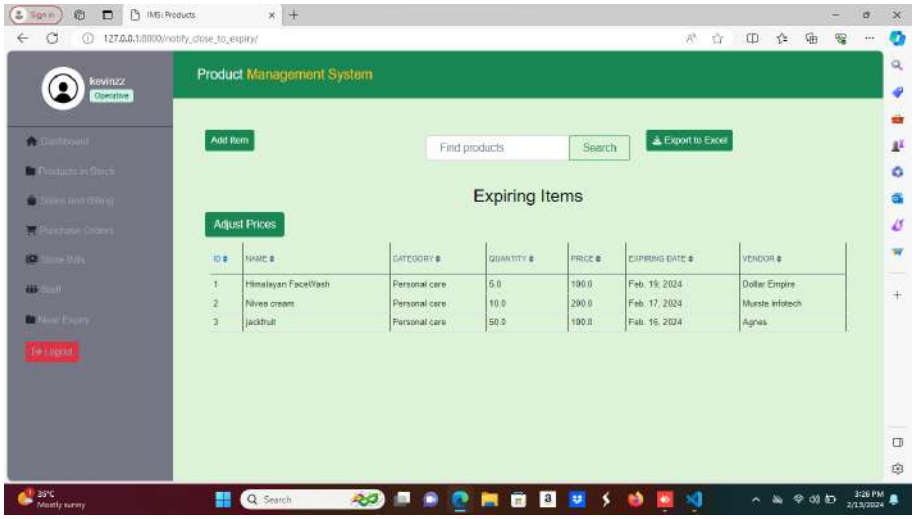
C.4 HOME


Product Management System
Products
374.0
Staff
10
Sales
8
Sales Infographics

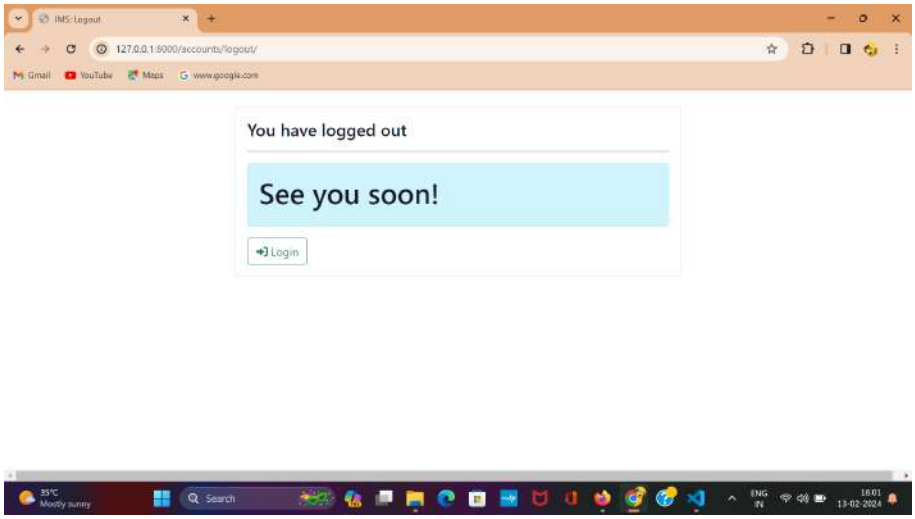
C.5 BILL


Product Management System
NewBill
Customer Name
Item
Quantity
Price
Payment method
jose
select the product
Payment method
Item
Quantity
Price
Subtotal
Total
echp-dot
1.0
8500.0
8500.0
nikon-d3500
2.0
5300.0
10600.0
19100.00
Print

C.6 EXPIRY


Product Management System
Add Item
Find products
Search
Export to Excel
Expiring Items
Adjust Prices
ID
NAME
CATEGORY
QUANTITY
PRICE
EXPIRING DATE
VENDOR
1 Himalayan FaceWash Personal care 5.0 100.0 Feb. 19, 2024 Dollar Empire
2 Nivea cream Personal care 10.0 200.0 Feb. 17, 2024 Murste Infotech
3 jackfruit Personal care 50.0 100.0 Feb. 16, 2024 Agnes

C.7 LOGOUT


IMS: Logout
127.0.0.1:8000/accounts/logout/
Gmail
YouTube
Maps
www.google.com
You have logged out
See you soon!
Login
35°C
Mostly sunny
Search
ENG
IN
16:01
13-02-2024

D CODE

login.html

```
[breaklines=true]
{% load crispy_forms_tags %}{% load static %}
<html>
  <head>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist
/css/bootstrap.min.css"
      rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3Q
DgpJLIm9Nao0Yz1ztcQTwFspd3yD
       65VohhpuuCOmLASjC"
      crossorigin="anonymous" />
  </head>
<body>
  <div class="login-page">
    <style>
    .login-page {
      width: 360px;
      padding: 8% 0 0;
      margin: auto;
    }
    .form {
      position: relative;
      z-index: 1;
      background: #FFFFFF;
      max-width: 360px;
      margin: 0 auto 100px;
      padding: 45px;
      text-align: center;
      box-shadow: 0 0 20px 0 rgba(0, 0, 0, 0.2), 0 5px 5px 0
 rgba(0, 0, 0, 0.24);
    }
    .form input {
      outline: 0;
      background: #bbffbb;
      width: 100%;
      border: 0;
      margin: 0 0 15px;
      padding: 15px;
      box-sizing: border-box;
      font-size: 14px;
    }
    .form button {
      text-transform: uppercase;
```

```
  outline: 0;
  background: #4CAF50;
  width: 100%;
  border: 0;
  padding: 15px;
  color: #FFFFFF;
  font-size: 14px;
  -webkit-transition: all 0.3 ease;
  transition: all 0.3 ease;
  cursor: pointer;
}
.form button:hover,.form button:active,.form button:focus {
  background: #43A047;
}
.form .message {
  margin: 15px 0 0;
  color: #b3b3b3;
  font-size: 12px;
}
.form .message a {
  color: #4CAF50;
  text-decoration: none;
}
.form .register-form {
  display: none;
}
.container {
  position: relative;
  z-index: 1;
  max-width: 300px;
  margin: 0 auto;
}
.container:before, .container:after {
  content: "";
  display: block;
  clear: both;
}
.container .info {
  margin: 50px auto;
  text-align: center;
}
.container .info h1 {
  margin: 0 0 15px;
  padding: 0;
  font-size: 36px;
  font-weight: 300;
```

```
      color: #1a1a1a;
    }
    .container .info span {
      color: #4d4d4d;
      font-size: 12px;
    }
    .container .info span a {
      color: #000000;
      text-decoration: none;
    }
    .container .info span .fa {
      color: #EF3B3A;
    }
    body {
      background: #76b852; /* fallback for old browsers */
      background: rgb(141,194,111);
      background: linear-gradient(90deg, rgba(141,194,111,1) 0%,
 rgba(118,184,82,1) 50%);
      -webkit-font-smoothing: antialiased;
      -moz-osx-font-smoothing: grayscale;
    }
    </style>
    <div class="form">
      <div>
        <strong class="navbar-brand" style="font-size: 26px;">
Product<span class=
             "text-warning">ms</span></strong>
      </div>
      <div class="text-danger">
        {% if messages %} {% for message in messages %}
        <div class="alert alert-{{message.tags}}">
            {{ message }}
        </div>
        {% endfor %} {% endif %}

    {% if form.non_field_errors %}
        <ul>
            {% for error in form.non_field_errors %}
                <li>{{ error }}</li>
            {% endfor %}
        </ul>
    {% endif %}
  </div>
      <form class="login-form" method="POST">
        {% csrf_token %}
        <p>{{ message }}</p>
```

```
        <label>Username</label>
        <span class="text-danger">{{ form.username.errors }}
</span>
        {{form.username}}
        <label>Password</label>
        <span class="text-danger">{{ form.password.errors }
}</span>
        {{form.password}}
        <button  type="submit" name="button">login
</button>
        <p class="message">Not registered? <a href=
            "{% url 'user-register' %}">Create an account
</a></p>
      </form>
    </div>
  </div>
</body>
</html>
```

register.html

```
{% load crispy_forms_tags %}{% load static %}
<html>
  <head>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@
5.0.2/dist/css/bootstrap.min.css"
      rel="stylesheet" integrity="sha384-EVSTQN3/azpr
G1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD
                    65VohhpuuCOmLASjC"
      crossorigin="anonymous" />
  </head>
<body>
  <div class="login-page">
    <style>
    .login-page {
      width: 360px;
      padding: 8% 0 0;
      margin: auto;
    }
    .form {
      position: relative;
      z-index: 1;
      background: #FFFFFF;
      max-width: 360px;
```

```
      margin: 0 auto 100px;
      padding: 45px;
      text-align: center;
      box-shadow: 0 0 20px 0 rgba(0, 0, 0, 0.2), 0
5px 5px 0 rgba(0, 0, 0, 0.24);
    }
    .form input {
      outline: 0;
      background: #bbffbb;
      width: 100%;
      border: 0;
      margin: 0 0 15px;
      padding: 15px;
      box-sizing: border-box;
      font-size: 14px;
    }
    .form button {
      text-transform: uppercase;
      outline: 0;
      background: #4CAF50;
      width: 100%;
      border: 0;
      padding: 15px;
      color: #FFFFFF;
      font-size: 14px;
      -webkit-transition: all 0.3 ease;
      transition: all 0.3 ease;
      cursor: pointer;
    }
    .form button:hover,.form button:active,
.form button:focus {
      background: #43A047;
    }
    .form .message {
      margin: 15px 0 0;
      color: #b3b3b3;
      font-size: 12px;
    }
    .form .message a {
      color: #4CAF50;
      text-decoration: none;
    }
    .form .register-form {
      display: none;
    }
    .container {
```

```
      position: relative;
      z-index: 1;
      max-width: 300px;
      margin: 0 auto;
    }
    .container:before, .container:after {
      content: "";
      display: block;
      clear: both;
    }
    .container .info {
      margin: 50px auto;
      text-align: center;
    }
    .container .info h1 {
      margin: 0 0 15px;
      padding: 0;
      font-size: 36px;
      font-weight: 300;
      color: #1a1a1a;
    }
    .container .info span {
      color: #4d4d4d;
      font-size: 12px;
    }
    .container .info span a {
      color: #000000;
      text-decoration: none;
    }
    .container .info span .fa {
      color: #EF3B3A;
    }
    body {
      background: #76b852; /* fallback for old browsers */
      background: rgb(141,194,111);
      background: linear-gradient(90deg,
rgba(141,194,111,1) 0%, rgba(118,184,82,1) 50%);
      -webkit-font-smoothing: antialiased;
      -moz-osx-font-smoothing: grayscale;
    }
    </style>
    <div class="form">
      <div>
        <strong class="navbar-brand" style=
"font-size: 26px;">Inventory<span class=
       "text-warning">ms</span></strong>
```

```
      </div>
      <div class="text-danger">
        {% if messages %} {% for message in messages %}
        <div class="alert alert-{{message.tags}}">
            {{ message }}
        </div>
        {% endfor %} {% endif %}

    {% if form.non_field_errors %}
        <ul>
            {% for error in form.non_field_errors %}
                <li>{{ error }}</li>
            {% endfor %}
        </ul>
    {% endif %}
  </div>
      <form class="login-form" method="POST">
        {% csrf_token %}
        <p>{{ message }}</p>
        <label>Username</label>
        <span class="text-danger">
{{ form.username.errors }}</span>
        {{form.username}}
        <label>Email</label>
        <span class="text-danger">{{ form.email.errors }}
</span>
        {{form.email}}
        <label>Enter Password</label>
        <span class="text-danger">{{ form.password.errors }}
</span>
        {{form.password1}}
        <label>Confirm Password</label>
        <span class="text-danger">{{ form.password2.errors }}
</span>
        {{form.password2}}
        <button  type="submit" name="button">Register
</button>
     <p class="message">Already have an account? <a href=
                            "{% url 'user-login' %}">Login</a></p>
      </form>
    </div>
  </div>
</body>
</html>
```

style.css

```
form input, select {
 outline: 0;
 background: #bbffbb;
 width: 100%;
 border: 0;
 margin: 0 0 15px;
 padding: 15px;
 box-sizing: border-box;
 font-size: 14px;
  }
#body-row {
 margin-left: 0;
 margin-right: 0;
}

#sidebar-container {
 min-height: 100vh;
 background-color: #4CAF50;
 padding: 0;
 /* flex: unset; */
}

.sidebar-expanded {
 width: 230px;
}

.sidebar-collapsed {
 /*width: 60px;*/
 width: 100px;
}

/* ----------| Menu item*/
#sidebar-container .list-group a {
 height: 50px;
 color: white;
}

/* ----------| Submenu item*/
#sidebar-container .list-group li.list-group-item {
 background-color: #4CAF50;
}

#sidebar-container .list-group .sidebar-submenu a {
 height: 45px;
 padding-left: 30px;
}
```

```
/* ----------| Separators */
.sidebar-separator-title {
 background-color: #4CAF50;
 height: 35px;
}

.sidebar-separator {
 background-color: #4CAF50;
 height: 25px;
}

.logo-separator {
 background-color: #4CAF50;
 height: 60px;
}

a.bg-dark {
 background-color: #4CAF50 !important;
}

@import 'https://fonts.googleapis.com/css2?
family=Inter:wght@300;400;500;600&display=
                swap" rel="stylesheet';

:root {
--dk-gray-100: #F3F4F6;
--dk-gray-200: #E5E7EB;
--dk-gray-300: #D1D5DB;
--dk-gray-400: #9CA3AF;
--dk-gray-500: #6B7280;
--dk-gray-600: #4B5563;
--dk-gray-700: #374151;
--dk-gray-800: #1F2937;
--dk-gray-900: #111827;
--dk-dark-bg: #313348;
--dk-darker-bg: #2a2b3d;
--navbar-bg-color: #6f6486;
--sidebar-bg-color: #252636;
--sidebar-width: 250px;
}

* {
margin: 0;
padding: 0;
box-sizing: border-box;
}
```

```
body {
font-family: 'Inter', sans-serif;
background-color: #def4d9;
}

a {
text-decoration: none;
link-style: none;
}
#wrapper {
margin-left: var(--sidebar-width);
transition: all .3s ease-in-out;
}

#wrapper.fullwidth {
margin-left: 0;
}



/** ---------------------------------
-- Sidebar
--------------------------------- */
.sidebar {
font-size: .925rem;
background-color: var(--sidebar-bg-color);
width: var(--sidebar-width);
transition: all .3s ease-in-out;
transform: translateX(0);
z-index: 9999999
}

.sidebar .close-aside {
position: absolute;
top: 7px;
right: 7px;
cursor: pointer;
color: #EEE;
}

.sidebar .sidebar-header {
border-bottom: 1px solid #2a2b3c
}

.sidebar .sidebar-header h5 a {
```

```
color: var(--dk-gray-300)
}

.sidebar .sidebar-header p {
color: var(--dk-gray-400);
font-size: .825rem;
}

.sidebar .search .form-control ~ i {
color: #2b2f3a;
right: 40px;
top: 22px;
}

.sidebar > ul > li {
padding: .7rem 1.75rem;
}

.sidebar ul > li > a {
color: var(--dk-gray-400);
text-decoration: none;
}

/* Start numbers */
.sidebar ul > li > a > .num {
line-height: 0;
border-radius: 3px;
font-size: 14px;
padding: 0px 5px
}

.sidebar ul > li > i {
font-size: 18px;
margin-right: .7rem;
color: var(--dk-gray-500);
}

.sidebar ul > li.has-dropdown > a:after {
content: '\eb3a';
font-family: unicons-line;
font-size: 1rem;
line-height: 1.8;
float: right;
color: var(--dk-gray-500);
transition: all .3s ease-in-out;
}
```

```
.sidebar ul .opened > a:after {
transform: rotate(-90deg);
}


.show-sidebar {
transform: translateX(-270px);
}

@media (max-width: 767px) {
.sidebar ul > li {
 padding-top: 12px;
 padding-bottom: 12px;
}

.sidebar .search {
 padding: 10px 0 10px 30px
}
}




/** ---------------------------------
-- welcome
--------------------------------- */
.welcome {
color: var(--dk-gray-300);
}

.welcome .content {
background-color: var(--dk-dark-bg);
}

.welcome p {
color: var(--dk-gray-400);
}




/** ---------------------------------
-- Statistics
--------------------------------- */
.statistics {
```

```
color: var(--dk-gray-200);
}

.statistics .box {
background-color: var(--dk-dark-bg);
}

.statistics .box i {
width: 60px;
height: 60px;
line-height: 60px;
}

.statistics .box p {
color: var(--dk-gray-400);
}


/** --------------------------------
-- Please don't do that in real-world projects!
-- overwrite Bootstrap variables instead.
-------------------------------- */

.navbar {
border: none !important;
}
.navbar .dropdown-menu {
right: auto !important;
left: 0 !important;
}
.navbar .navbar-nav>li>a {
color: #EEE !important;
line-height: 55px !important;
padding: 0 10px !important;
}
.navbar .navbar-brand {color:#FFF !important}
.navbar .navbar-nav>li>a:focus,
.navbar .navbar-nav>li>a:hover {color:
 #EEE !important}

.navbar .navbar-nav>.open>a,
.navbar .navbar-nav>.open>a:focus,
.navbar .navbar-nav>.open>a:hover
{background-color: transparent !important;
color: #FFF !important}
```

```
.navbar .navbar-brand {line-height: 55px
 !important; padding: 0 !important}
.navbar .navbar-brand:focus,
.navbar .navbar-brand:hover {color:
#FFF !important}
.navbar>.container .navbar-
brand, .navbar>.container-fluid .navbar-brand
               {margin: 0 !important}
@media (max-width: 767px) {
.navbar>.container-fluid .navbar-brand {
 margin-left: 15px !important;
}
.navbar .navbar-nav>li>a {
 padding-left: 0 !important;
}
.navbar-nav {
 margin: 0 !important;
}
.navbar .navbar-collapse,
.navbar .navbar-form {
 border: none !important;
}
}

.navbar .navbar-nav>li>a {
float: left !important;
}
.navbar .navbar-nav>li>a>span:not(.caret) {
background-color: #e74c3c !important;
border-radius: 50% !important;
height: 25px !important;
width: 25px !important;
padding: 2px !important;
font-size: 11px !important;
position: relative !important;
top: -10px !important;
right: 5px !important
}
.dropdown-menu>li>a {
padding-top: 5px !important;
padding-right: 5px !important;
}
.navbar .navbar-nav>li>a>i {
font-size: 18px !important;
}
```

```
/* Start media query */

@media (max-width: 767px) {
#wrapper {
 margin: 0 !important
}
.statistics .box {
 margin-bottom: 25px !important;
}
.navbar .navbar-nav .open
     .dropdown-menu>li>a {
 color: #CCC !important
}
.navbar .navbar-nav .open .
      dropdown-menu>li>a:hover {
 color: #FFF !important
}
.navbar .navbar-toggle{
 border:none !important;
 color: #EEE !important;
 font-size: 18px !important;
}
.navbar .navbar-toggle:focus, .navbar
 .navbar-toggle:hover
     {background-color: transparent !important}
}


::-webkit-scrollbar {
background: transparent;
width: 5px;
height: 5px;
}

::-webkit-scrollbar-thumb {
background-color: #3c3f58;
}

::-webkit-scrollbar-thumb:hover {
background-color: rgba(0, 0, 0, 0.3);
}
```

```
//sorting
table {
background-color: #000 !important;
}
table.dataTable thead .sorting:after,
table.dataTable thead .sorting:before,
table.dataTable thead .sorting_asc:after,
table.dataTable thead .sorting_asc:before,
table.dataTable thead .sorting_asc_disabled:after,
table.dataTable thead .sorting_asc_disabled:before,
table.dataTable thead .sorting_desc:after,
table.dataTable thead .sorting_desc:before,
table.dataTable thead .sorting_desc_disabled:after,
table.dataTable thead .sorting_desc_disabled:before {
bottom: .5em;
}

//update profile
 /*=========================================================
            Form Section
  =========================================================*/
  form {
    margin: 0 auto;
    width: 50%;
    border-bottom-right-radius: 6px;
    border-bottom-left-radius: 6px;
    padding-bottom: 10px;
  }
  /*changing title style*/
  h1 {
    padding-top: 30px;
    text-align: center;
    color: #FF5722;
    font-size: 26px;
  }

  /*Camera image to upload image*/
  .fa-camera-retro {
      color: #a7a7a7;
      width: 100px;
      height: 100px;
      margin-top: 20px;
      line-height: 100px;
      border: 2px solid #a7a7a7;
      cursor: pointer;
      -webkit-border-radius: 50%;
```

```
     border-radius: 50%;
     box-shadow: 0 15px 20px -10px
rgba(0, 0, 0, 0.3);
 }

 .fa-camera-retro:hover{
   border-color: #6f6f6f;
   box-shadow: 0 3px 5px -5px rgba(0, 0, 0, 0.3);
   transform: translateY(-1px);
   color: #6f6f6f;
 }



 /*=========================================================
           input Section
 ==========================================================*/

 /*Float Label main style*/
 .float-label input, .float-label select {
     -webkit-appearance: none;
     outline: none;
     border: none;
     margin: 0 auto;
     width: 100%;
     display: block;
     -moz-border-radius: 0;
     border-radius: 0;
     font-family: 'Lato', sans-serif;
     font-size: 18px;
     border-bottom: 1px solid rgba(0, 0, 0, 0.7);
     background: transparent;
     color: #757575;
     padding: 5px 15px 7px 10px;
 }

 .title {
   font-family: 'Lato', sans-serif;
   font-size: 18px;
   background: transparent;
   color: #757575;
   font-weight: normal;
   margin-left: 7px;
 }

 .gender, .birthday {
```

```
    text-align: left;
    padding-left: 16px;
    margin-top: -10px;
}

.gender div, .birthday div {
  display: inline-block;
  padding-left: 0;
}

/*add style to radio options*/
.option-input {
  -webkit-appearance: none;
  -moz-appearance: none;
  -ms-appearance: none;
  -o-appearance: none;
  appearance: none;
  position: relative;
  top: 13.33333px;
  right: 0;
  bottom: 0;
  left: 0;
  height: 15px;
  width: 15px;
  transition: all 0.15s ease-out 0s;
  background: #cbd1d8;
  border: none;
  color: #fff;
  cursor: pointer;
  display: inline-block;
  margin-right: 0.5rem;
  outline: none;
  position: relative;
  z-index: 1000;
}
.option-input:hover {
  background: #9faab7;
}
.option-input:checked {
  background: #77cee2;
}
.option-input:checked::before {
  height: 15px;
  width: 15px;
  position: absolute;
  content: '';
```

```
  display: inline-block;
  font-size: 12px;
  text-align: center;
  line-height: 15px;
}
.option-input:checked::after {
  -webkit-animation: click-wave 0.65s;
  -moz-animation: click-wave 0.65s;
  animation: click-wave 0.65s;
  background: #77cee2;
  content: '';
  display: block;
  position: relative;
  z-index: 100;
}
.option-input {
  border-radius: 50%;
}
.option-input::after {
  border-radius: 50%;
}
.radio-inline input[type=radio] {
  position: static;
  margin-left: 0;
  padding-left: 0;
}
.radio-inline {
  vertical-align: baseline;
}
.radio-inline span {
  line-height: 4;
  font-size: 16px;
  padding-left: 4px;
}

 /*Style birthday select */
.float-label select {
  display: inline-block;
  width: 24%;
  margin-left: 20px;
  font-size: 16px;
  color: black;
}
.float-label select:first-child{
  width: 30%;
}
```

```
.birthday .select {
  width: 82%;
}

.float-label > select option:first-child {
  color: #77cee2;
}


#updateProfile {
  padding: 20px 60px;
}

.user-left {
  padding-top: 40px;
}

 /*create a round container to hide overflow image*/
.user-left span {
  display: inline-block;
  width: 250px;
  height: 250px;
  overflow: hidden;
  text-align: center;
  border-radius: 100%;
  /*add shadow and border*/
  border: 2px solid #a9a9a9;
  cursor: pointer;
  -webkit-border-radius: 50%;
  border-radius: 50%;
  box-shadow: 0px 20px 25px -10px rgba(0, 0, 0, 0.3);
}

/*change image effect when hover*/
.user-left span:hover{
  border-color: #b3b3b3;
  box-shadow: 0 3px 5px -5px rgba(0, 0, 0, 0.3);
  transform: translateY(-1px);
}

/*change font margin and padding*/
.user-left h1 {
  padding-top: 0;
}

/*center image and make it round*/
```

```
  .user-image {
    left: 50%;
    margin-left: -100%;
    position: relative;
    width: auto important;
    height: 250px important;
  }

  /*change image opacity*/
  .user-image:hover {
    opacity: 0.5;
    filter: alpha(opacity=50);
  }

  .user-right {
    padding: 20px 20px 40px;
    min-height: 200px;
    /*margin-left: 80px;*/
  }

  /*Change profile title*/
  .user-right h2 {
    text-align: left;
    font-family: 'Roboto', Arial, sans-serif;
    font-weight: 400;
    font-size: 3em;
  }
  .user-right span {
    color: #d2d2d2;
  }

  .user-info {
    padding: 18px;
    width: 100%;
    min-height: 200px;
    border-radius: 3px;
    box-shadow: 0 0 0 0 transparent;
    box-shadow: 0 15px 20px -15px
 rgba(0, 0, 0, 0.3), 0 55px 50px -35px
                          rgba(210, 214, 213, 0.5);

    box-shadow: 0px 2px 2px 2px
rgba(210, 214, 213, 0.4);
  }
```

```
  td p{
    font-size: 20px;
    padding: 10px 15px 10px 10px;
    text-align: left;
  }

  #signOut {
    float: right;
  }
```

logout.html

```
[verbetim]
{% load crispy_forms_tags %}
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta http-equiv="X-UA-Compatible"
content="IE=edge">
        <meta name="viewport"
 content="width=device-width, initial-scale=1.0">
        {% load static %}

        <link rel="stylesheet" href=
     "{% static 'fontawesome/css/all.css' %}"
 type="text/css">
        <link rel="stylesheet" type="text/css" href=
"https://unpkg.com/@webpixels
                                  /css@1.1.5/dist/index.css">
        <link href="https://cdn.jsdelivr.net/npm/
bootstrap@5.0.2/dist/css/
      bootstrap.min.css" rel="stylesheet" integrity=
   "sha384-EVSTQN3/azprG1Anm3QDgp
JLIm9Nao0Yz1ztcQTwFspd
       3yD65VohhpuuCOmLASjC"
 crossorigin="anonymous">
        <title>IMS: Logout</title>
    </head>
    <body>
       <div class="row my-4">
         <div class="col-md-6 offset-md-3">
            <div class="border p-3 bg-white">
             <h4>You have logged out</h4>
```

```
                <hr>
               <div class="alert alert-info">
                  <h1>See you soon!</h1>
                     </div>
          <a class="btn btn-outline-success" href=
"{% url 'user-login' %}"><i class=
"fa-solid fa-right-to-bracket"></i> Login</a>
                 </div>
            </div>
        </div>
    </body>
</html>
```

bill.html

```
{% extends "store/base.html" %}
{% load render_table from django_tables2 %}
{% load querystring from django_tables2 %}
{% load static %}{% block title %}Bills
{%endblock title%}{% block content %}
    <div class="col container p-5">
      <style>
        input[type="text"],select{
            box-sizing: border-box;
            border: 1px solid #ccc;
            border-radius: 4px;
            display: inline-block;
            padding: 6px 20px;
            margin: 8px 0;
        }
        button[type="submit"]{
          padding: 6px 20px;
          margin: 8px 0;
      }
      </style>
      {% comment %} <form method="GET"
class="container">
        {{filter.form}}
        <button type="submit" class="btn btn-sm
btn-success fa fa-search"> Search</button>
      </form> {% endcomment %}
      <div>
     <a class="btn btn-sm btn-success" href=
"{% url 'bill-create' %}">
```

```
                          Add a bill record</a>
     <a class="float-end btn btn-sm btn-success" href=
                            "{% querystring '_export'='xlsx' %}">
          <i class="fa-solid fa-download"></i>
          Export to Excel
        </a>
      </div>
      <div class="m-2">
        <table class= "table table-sm table-responsive">
            <thead>
              <tr class="table-success">
                <th scope="col"><a href=
"{% querystring table.prefixed_order_by_field
=column.order_by_alias.next %}">ID
 <i class="fa-solid fa-sort"></i></a></th>
 <th scope="col">Name <i class="fa-solid fa-sort">
</i></th>
<th scope="col">Description <i class="fa-solid fa-sort">
</i></th>
<th scope="col">Contact number <i class=
"fa-solid fa-sort"></i></th>
 <th scope="col">email<i class="fa-solid fa-sort"></i></th>
 <th scope="col">Payment details<i class="fa-solid fa-sort">
</i></th>
 <th scope="col">Amount <i class="fa-solid fa-sort">

</i></th>
 <th scope="col">Status <i class="fa-solid fa-sort"></i></th>
 <th scope="col"> Action</th>
              </tr>
            </thead>
            <tbody>
            {% for bill in bills %}
              <tr>
                <th scope="row"><a> {{bill.id}}</a></th>
                <td>{{bill.institution_name}}</td>
                <td>{{bill.description}}</td>
                <td>{{bill.phone_number}}</td>
                <td>{{bill.email}}</td>
                <td>{{bill.payment_details}}</td>
                <td>{{bill.amount}}</td>
                <td>{% if bill.status == True%}
                  <span class="badge badge-pill
bg-soft-success text-success me-2">
                    Paid
                  </span>
```

```
                  {% else %}
                  <span class="badge badge-pill bg-soft
-danger text-danger me-2">
                    Pending
                  </span>
                  {% endif %}
                </td>
                <td>
                  <a class="text-info" href=
"{% url 'bill-update' bill.slug %}"><i class=
"fa-solid fa-pen"></i></a>
                  <a class="text-danger float-end" href=
"{% url 'bill-delete' bill.pk %}"><i class=
"fa-solid fa-trash"></i></a>
                </td>
              </tr>
            {% endfor %}
            </tbody>
          </table>
        </div>
          <div class="paginator">
            {% if is_paginated %}
            <ul class="pagination ">
                {% if page_obj.has_previous %}
                <li><a class="btn btn-sm accent-1
 btn-outline-success
 " href="?page={{ page_obj.previous_page_number }}
 ">&laquo;</a></li>
                {% else %}
                <li class="disabled "><span class=
"btn btn-sm btn-outline-success ">
                &laquo;</span></li>
             {% endif %} {% for i in paginator.page_range %}
 {% if page_obj.number == i %}
                <li class="active "><span class=
"btn btn-sm btn-success ">{{ i }}
                    <span class="sr-only ">(current
)</span></span>
                </li>
                {% else %}
      <li><a href="?page={{ i }} " class=
"btn btn-sm btn-outline-success ">{{ i }}</a></li>
                {% endif %} {% endfor %}
 {% if page_obj.has_next %}
                <li><a href="?page={{ page_obj.next_page_number }}
            " class="btn btn-sm btn-outline-success ">&raquo;</a></li>
```

```
                {% else %}
                <li class="disabled ">
<span class="btn btn-sm btn-outline-success ">
                &raquo;</span></li>
                {% endif %}
            </ul>
            {% endif %}
        </div>
    </div>
  {% endblock content %}
```

productlist.html

```
[breaklines=true]
{% extends "store/base.html" %}
{% load render_table from django_tables2 %}
{% load querystring from django_tables2 %}
{% load static %}
{% block title %}Products{%endblock title%}
{% block content %}
    <div class="col container p-5">
      <style>
        input[type="text"],select{
            box-sizing: border-box;
            border: 1px solid #ccc;
            border-radius: 4px;
            display: inline-block;
            padding: 6px 20px;
            margin: 8px 0;
        }
        button[type="submit"]{
          padding: 6px 20px;
          margin: 8px 0;
      }
      </style>
      <div class="row">
        <div class="col-md-4">
          <a class="btn btn-sm btn-success" href=
  "{% url 'product-create' %}">Add Item</a>
        </div>
        <div class="col-md-4">
          <form class="input-group" role=
"search " id="searchform" action=
"{% url 'item_search_list_view' %}
" method="get" accept-charset="utf-8">
            <div class="form-group">
```

```
                <div class="input-group ">
                    <input id="searchbox" name=
"q" type="text" class=
      "form-control " placeholder="Find products">
                    <span class="input-group-btn">
                      <button class=
                "btn btn-outline-success" type=
"submit">Search</i></button>
                    </span>
                </div>
            </div>
        </form>
        </div>
        <div class="col-md-4 float-end">
          <a class="btn btn-sm btn-success" href=
"{% querystring '_export'='xlsx' %}">
            <i class="fa-solid fa-download"></i>
            Export to Excel
          </a>
        </div>
      </div>
      <div class="m-2">
        <table class= "table table-sm table-responsive">
            <thead>
              <tr class="table-success">
                <th scope="col"><a href=
"{% querystring table.prefixed_order_by_field
=column.order_by_alias.next %}">ID
                                <i class="fa-solid fa-sort"></i></a></th>
                <th scope="col">Name <i class=
"fa-solid fa-sort"></i></th>
                <th scope="col">Category <i class=
"fa-solid fa-sort"></i></th>
                <th scope="col">Quantity <i class=
"fa-solid fa-sort"></i></i></th>
                <th scope="col">Price <i class=
"fa-solid fa-sort"></i></th>
                <th scope="col">Expiring date <i class=
"fa-solid fa-sort"></i></th>
                <th scope="col">Vendor <i class=
"fa-solid fa-sort"></i></th>
                <th scope="col"> Action</th>
              </tr>
            </thead>
            <tbody>
            {% for item in items %}
```

```
              <tr>
                <th scope="row"><a> {{item.id}}
</a></th>
                <td>{{item.name}}</td>
                <td>{{item.category}}</td>
                <td>{{item.quantity}}</td>
                <td>{{item.selling_price}}</td>
                <th scope="col">{{item.expiring_date}}
</th>
                <td>{{item.vendor}}</td>
                <td>
                  <a class="text-info" href=
"{% url 'product-update' item.slug %}">
<i class="fa-solid fa-pen"></i></a>
                  <a class="text-danger float-end" href=
"{% url 'product-delete' item.slug %}">
<i class="fa-solid fa-trash"></i></a>
                </td>
              </tr>
            {% endfor %}
            </tbody>
          </table>
        </div>
          <div class="paginator">
            {% if is_paginated %}
            <ul class="pagination ">
                {% if page_obj.has_previous %}
                <li><a class="btn btn-sm accent-1
btn-outline-success
       " href="?page={{ page_obj.previous_page_number }
} ">&laquo;</a></li>
                {% else %}
                <li class="disabled "><span class=
"btn btn-sm btn-outline-success ">
             &laquo;</span></li>
                {% endif %} {% for i in paginator.page_range %}
                        {% if page_obj.number == i %}
                <li class="active "><span class=
"btn btn-sm btn-success ">{{ i }}
                 <span class="sr-only ">(current)
</span></span>
                </li>
                {% else %}
                <li><a href="?page={{ i }} " class=
"btn btn-sm btn-outline-success ">
                      {{ i }}</a></li>
```

```
                {% endif %} {% endfor %} {% if page_obj.has_next %}
                <li><a href="?page={{ page_obj.next_page_number }}
               " class="btn btn-sm btn-outline-success ">&raquo;</a></li>
                {% else %}
                <li class="disabled "><span class=
"btn btn-sm btn-outline-success ">
          &raquo;</span></li>
                {% endif %}
            </ul>
            {% endif %}
        </div>
    </div>
  {% endblock content %}
```

manage.py

```
#!/usr/bin/env python
"""Django's command-line utility for administrative tasks."""
import os
import sys


def main():
    """Run administrative tasks."""
    os.environ.setdefault
('DJANGO_SETTINGS_MODULE', 'InventoryMS.settings')
    try:
        from django.core.management import
 execute_from_command_line
    except ImportError as exc:
        raise ImportError(
            "Couldn't import Django. Are you sure it's installed and "
            "available on your PYTHONPATH environment
 variable? Did you "
            "forget to activate a virtual environment?"
        ) from exc
    execute_from_command_line(sys.argv)


if __name__ == '__main__':
    main()
```

module.py

```
from django.db import models
```

```
from store.models import Item
from accounts.models import Profile, Vendor
from django_extensions.db.fields import AutoSlugField

# Create your models here.

PAYMENT_CHOICES = [
    ('MP', 'MPESA'),
    ('VISA', 'VISA'),
    ('CS', 'CASH'),
    ('VM', 'VOOMA'),
    ('BK', 'BANK')
]

DELIVERY_CHOICES = [
    ('P', 'PENDING'),
    ('S', 'SUCCESSFUL')
]

class Sale(models.Model):
    slug = AutoSlugField(unique=True , populate_from=
                'customer_name')
    item = models.ForeignKey(Item, on_delete=
                    models.CASCADE, blank=True, null=True)
    customer_name =
          models.CharField(max_length=20, null=True, blank=True)
    transaction_date =
       models.DateTimeField(auto_now=True, blank=True, null=True)
    quantity = models.FloatField(default=0.00, blank=False, null=False)
    payment_method = models.CharField(choices=
        PAYMENT_CHOICES, max_length=20, blank='True', null=True)
    price = models.FloatField(default=0.00, blank=False, null=False)
    total_value = models.FloatField(blank=True, null=True)
    amount_received = models.FloatField(default=False,
          blank=False, null=False)
    balance = models.FloatField(default=False, blank=False, null=False)
    profile = models.ForeignKey(Profile, verbose_name=
        ('Served by'), on_delete=models.CASCADE)

    def save(self, *args, **kwargs):
        amt_received = self.amount_received
        price = self.price
        quantity = self.quantity
        self.total_value = price * quantity
        self.balance = amt_received - self.total_value
        super().save(*args, **kwargs)
```

```
    def __str__(self):
        return str(self.item.name)

class Purchase(models.Model):
    slug = AutoSlugField(unique=True , populate_from='vendor')
    item = models.ForeignKey(Item, on_delete=models.CASCADE)
    description = models.TextField(max_length=300
, blank=True, null=True)
    vendor = models.ForeignKey
  (Vendor, related_name='vendor', on_delete=models
.CASCADE, blank=False, null=False)
    order_date = models.DateTimeField(auto_now_add=True)
    delivery_date = models.DateTimeField
    (auto_now=False, auto_now_add=False, blank=True,
null=True, verbose_name=
                                                    ('Delivery Date'))
    quantity = models.FloatField(default=0.00, blank=False, null=False)
    delivery_status = models.CharField(choices=
DELIVERY_CHOICES, max_length=3, default='P', blank=False,
                      null=False, verbose_name=
('Delivery Status'))
    price = models.FloatField(default=0.00, blank=False,
      null=False, verbose_name=
('Price per item(Ksh)'))
    total_value = models.FloatField()

    def save(self, *args, **kwargs):
        quantity = self.quantity
        price = self.price
        self.total_value = price * quantity
        return super().save(*args, **kwargs)

    def __str__(self):
        return self.item.name

    class Meta:
        ordering = ["order_date"]
```

setting.py

```
import os
from pathlib import Path
```

```
# Build paths inside the project like this:
 BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent


# Quick-start development settings -
 unsuitable for production
# See https://docs.djangoproject.com/
en/4.1/howto/deployment/checklist/

# SECURITY WARNING: keep the secret
 key used in production secret!
SECRET_KEY = 'django-insecure
-g_n2+2bznu6e@1wel!i
(&-4tp86_7lop5395ww+i4x%9*7^old'

# SECURITY WARNING:
 don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = []


# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',

    'phonenumber_field',
    'crispy_forms',
    'imagekit',
    'django_extensions',
    'django_filters',
    'django_tables2',

    'store.apps.StoreConfig',
    'accounts.apps.AccountsConfig',
    'transactions.apps.TransactionsConfig',
    'invoice.apps.InvoiceConfig',
    'bills.apps.BillsConfig',
```

```
]

MIDDLEWARE = [
  'django.middleware.security.SecurityMiddleware',
 'django.contrib.sessions.middleware.SessionMiddleware',
 'django.middleware.common.CommonMiddleware',
'django.middleware.csrf.CsrfViewMiddleware',
 'django.contrib.auth.middleware.AuthenticationMiddleware',
'django.contrib.messages.middleware.MessageMiddleware',
'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'InventoryMS.urls'

TEMPLATES = [
    {
 'BACKEND': 'django.template.backends
.django.DjangoTemplates',
        'DIRS': [],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
       'django.template.context_processors.debug',
        'django.template.context_processors.request',
         'django.contrib.auth.context_processors.auth',
          'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]

WSGI_APPLICATION = 'InventoryMS.wsgi.application'


# Database
# https://docs.djangoproject.com/
en/4.1/ref/settings/#databases


DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    }
}
```

```
# Password validation
# https://docs.djangoproject.com/
en/4.1/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.
password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth
.password_validation.MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth
.password_validation.CommonPasswordValidator',
    },
    {
        'NAME': 'django.contrib.auth.
password_validation.NumericPasswordValidator',
    },
]


# Internationalization
# https://docs.djangoproject.com/en/4.1/topics/i18n/

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_TZ = True

LOGIN_URL = 'user-login'
LOGIN_REDIRECT_URL = 'dashboard'
LOGOUT_URL = 'logout'

# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/4.1/howto/static-files/

STATIC_URL = 'static/'
STATICFILES_DIRS = [os.path.join(BASE_DIR,'static')
```

```
]
MEDIA_ROOT = os.path.join(BASE_DIR, 'static/images')
MEDIA_URL = '/images/'

# Default primary key field type
# https://docs.djangoproject.com/
en/4.1/ref/settings/#default-auto-field

DEFAULT_AUTO_FIELD =
 'django.db.models.BigAutoField'

# CRISPY_TEMPLATE_PACK = 'bootstrap4'



EMAIL_BACKEND =
 'django.core.mail.backends.smtp.EmailBackend'
EMAIL_HOST = 'smtp.gmail.com'
EMAIL_USE_TLS = True
EMAIL_PORT = 587
EMAIL_HOST_USER =
 "product management system753@gmail.com"
EMAIL_HOST_PASSWORD = "oyhzlplyolihopys"


# FCM_DJANGO_SETTINGS = {
#     "FCM_SERVER_KEY":
"BC7LHSnv6csnz2VKw5DMASW6n
D2n_FiCsQwjBPUmlOgN1AM
-Qehh2qslANJudWf7R-P1UbL
KwYZJyxu8OiGJwWE",
# }
```

setup.sh

```
[breaklines=true]
#!/bin/bash

echo " Setting up sales-and-inventory-management"

echo " Building Docker image..."
docker build -t sales-and-inventory-management:1.0 .
```

```
if [ $? -ne 0 ]; then
  echo " Error: Docker image build failed!"
  exit 1
fi

echo " Starting Docker containers in detached mode..."
docker run -d -p 8000:8000
 sales-and-inventory-management:1.0

if [ $? -ne 0 ]; then
  echo " Error: Docker container failed to start!"
  exit 1
fi

echo " Setup completed successfully!"
```

signal.py

```
[breaklines=true]
from django.db.models.signals import post_save
from django.dispatch import receiver

from django.contrib.auth.models import User
from .models import Profile


@receiver(post_save, sender=User)
def create_profile(sender, instance,
 created, **kwargs):

    if created:
        Profile.objects.create(user=instance)
        print('profile created!')

@receiver(post_save, sender=User)
def update_profile(sender, instance,
 created, **kwargs):

    if created == False:
        instance.profile.save()
        print('profile updated!')
```

table.py

```
[breaklines=true]
```

```
# tutorial/tables.py
import django_tables2 as tables
from .models import Sale, Purchase
from django.shortcuts import render

class SaleTable(tables.Table):
    class Meta:
        model = Sale
        template_name =
 "django_tables2/semantic.html"
       fields = ('item', 'customer_name',
'transaction_date', 'payment_method', 'quantity',
                    'price', 'total_value', 'amount_received',
 'balance', 'profile')
        order_by_field = 'sort'

class PurchaseTable(tables.Table):
    class Meta:
        model = Purchase
        template_name = "django_tables2/semantic.html"
        fields = ('item', 'vendor', 'order_date',
         'delivery_date', 'quantity',
          'delivery_status', 'price', 'total_value')
        order_by_field = 'sort'
```

url.py

```
[breaklines=true]
from django.urls import path
from . import views
from django.conf.urls.static import static
from django.conf import settings
from .views import (
    PurchaseListView,
    PurchaseDetailView,
    PurchaseCreateView,
    PurchaseUpdateView,
    PurchaseDeleteView,
    SaleListView,
    SaleDetailView,
    SaleCreateView,
    SaleUpdateView,
    SaleDeleteView,
)

urlpatterns = [
```

```
    path('purchases/',
PurchaseListView.as_view(), name="purchaseslist"),
    path('purchase/<slug:slug>/',
PurchaseDetailView.as_view(), name='purchase-detail'),
    path('new-purchase/',
PurchaseCreateView.as_view(), name='purchase-create'),
    path('purchase/<int:pk>/update/',
 PurchaseUpdateView.as_view(), name='purchase-update'),
    path('purchase/<int:pk>/delete/',
 PurchaseDeleteView.as_view(), name='purchase-delete'),
    path('sales/',SaleListView.as_view(), name="saleslist"),
    path('sale/<int:pk>/',
 SaleDetailView.as_view(), name='sale-detail'),
    path('new-sale/', SaleCreateView.as_view(),
 name='sale-create'),
    path('sale/<slug:slug>/update/',
SaleUpdateView.as_view(), name='sale-update'),
    path('sale/<slug:slug>/delete/',
 SaleDeleteView.as_view(), name='sale-delete'),

]
urlpatterns += static(settings.MEDIA_URL,document_roo
t=settings.MEDIA_ROOT)
```

view.py

```
[breaklines=true]
from django.shortcuts import render
from .models import *
from django.contrib.auth.models import User
from .filters import PurchaseFilter, SaleFilter
from django.core.paginator import Paginator, EmptyPage,
 PageNotAnInteger
from django.contrib.auth.mixins import LoginRequiredMixin
, UserPassesTestMixin
from django.views.generic.edit import FormMixin
from django_tables2 import SingleTableView
import django_tables2 as tables
from django.urls import reverse
from django.shortcuts import get_object_or_404
from django_tables2.export.views import ExportMixin
from django_tables2.export.export import TableExport
from .tables import PurchaseTable, SaleTable
from django.core.exceptions import ValidationError
from accounts.models import Profile
from django.views.generic import (
```

```
    ListView,
    DetailView,
    CreateView,
    UpdateView,
    DeleteView
)

class PurchaseListView(ExportMixin, tables.SingleTableView):
    """View to list purchases and export them."""
    model = Purchase
    table_class = SaleTable
    template_name = 'transactions/purchases_list.html'
    context_object_name = 'purchases'
    paginate_by = 10
    SingleTableView.table_pagination = False

class PurchaseDetailView(FormMixin, DetailView):
    """View to display details of a purchase."""
    model = Purchase
    template_name = 'transactions/sale_detail.html'

    def get_success_url(self):
        return reverse('sale-detail', kwargs={'slug':
        self.object.slug})

class PurchaseCreateView(LoginRequiredMixin, CreateView):
    """View to create a new purchase."""
    model = Purchase
    template_name = 'transactions/purchasescreate.html'
    fields = ['item', 'description', 'vendor', 'delivery_date'
, 'quantity', 'delivery_status']

    def form_valid(self, form):
        """Handles the form submission and updates the item's
 quantity."""
        item = form.cleaned_data['item']
        quantity = form.cleaned_data['quantity']

        total_value = item.selling_price * quantity

        form.instance.total_value = total_value
        form.instance.price = item.selling_price

        form.instance.balance = total_value

        form.instance.profile = self.request.user.profile
```

```
        item.quantity += quantity
        item.save()

        return super().form_valid(form)

    def get_success_url(self):
        return reverse('purchaseslist')

    def test_func(self):
        profile_list = Profile.objects.all()
        if self.request.user.profile in profile_list:
            return False
        else:
            return True

class PurchaseUpdateView(LoginRequiredMixin,
UserPassesTestMixin, UpdateView):
    """View to update a purchase."""
    model = Purchase
    template_name = 'transactions/purchaseupdate.html'
    fields = ['item', 'description', 'vendor', 'delivery_date',
 'quantity', 'price', 'delivery_status']

    def test_func(self):
        profiles = Profile.objects.all()
        if self.request.user.profile in profiles:
            return True
        else:
            return False

    def get_success_url(self):
            return reverse('purchaseslist')


class PurchaseDeleteView(LoginRequiredMixin,
 UserPassesTestMixin, DeleteView):
    """View to delete a purchase."""
    model = Purchase
    template_name = 'transactions/purchasedelete.html'

    def test_func(self):
        profiles = Profile.objects.all()
        if self.request.user.profile in profiles:
            return True
        else:
```

```
            return False
    def get_success_url(self):
            return reverse('purchaseslist')

class SaleListView(ExportMixin, tables.SingleTableView):
    """View to list sales and export them."""
    model = Sale
    table_class = SaleTable
    template_name = 'transactions/sales_list.html'
    context_object_name = 'sales'
    paginate_by = 10
    SingleTableView.table_pagination = False

class SaleDetailView(DetailView):
    """View to display details of a sale."""
    model = Sale
    template_name = 'transactions/saledetail.html'


class SaleCreateView(LoginRequiredMixin, CreateView):
    """View to create a new sale."""
    model = Sale
    template_name = 'transactions/salescreate.html'
    fields = ['item', 'customer_name', 'payment_method',
        'quantity', 'amount_received']

    def get_success_url(self):
        return reverse('saleslist')

    def form_valid(self, form):
        """Handles the form submission and
              validates product availability."""
        item = form.cleaned_data['item']
        quantity = form.cleaned_data['quantity']

        if item.quantity < quantity:
 raise ValidationError(f"Only {item.quantity} units of
     '{item.name}' are available.")

        price = item.selling_price

        total_price = price * quantity

        form.instance.price = price
        form.instance.total_value = total_price
```

```
        amount_received = form.cleaned_data['amount_received']
        balance = amount_received - total_price
        form.instance.balance = balance

        form.instance.profile = self.request.user.profile
        return super().form_valid(form)

    def test_func(self):
        profile_list = Profile.objects.all()
        if self.request.user.profile in profile_list:
            return False
        else:
            return True

class SaleUpdateView(LoginRequiredMixin, UserPassesTestMixin,
 UpdateView):
    """View to update a sale."""
    model = Sale
    template_name = 'transactions/sale_update.html'
    fields = ['item', 'customer_name', 'payment_method',
      'quantity', 'price',
                                              'amount_received']

    def test_func(self):
        """Checks if the user has the required permissions to
 access this view."""
        profiles = Profile.objects.all()
        if self.request.user.profile in profiles:
            return True
        else:
            return False

    def get_success_url(self):
        return reverse('saleslist')

    def form_valid(self, form)
 """Handles form submission and sets the profile of the sale."""
        form.instance.profile = self.request.user.profile
        return super().form_valid(form)

class SaleDeleteView(LoginRequiredMixin, UserPassesTestMixin,
 DeleteView):
    """View to delete a sale."""
    model = Sale
    template_name = 'transactions/saledelete.html'
```

```
    def test_func(self):

        """Checks if the user has the required permissions to
     access this view."""
        if self.request.user.is_superuser:
            return True
        else:
            return False
```

WEED DETECTION WEBSITE

PROJECT REPORT

Submitted By

FEBIN VINCENT

Reg. No. CCAVBCA002

For the award of the Degree of

Bachelor of computer application (BCA)

(University of Calicut)

under the guidance of

Ms. Sini Thomas

Head of the Department

BACHELOR OF COMPUTER APPLICATION
DEPARTMENT OF COMPUTER SCIENCE
CHRIST COLLEGE (AUTONOMOUS)
IRINJALAKUDA, KERALA

2021-2024

DEPARTMENT OF COMPUTER SCIENCE

CHRIST COLLEGE (AUTONOMOUS)

IRINJALAKUDA

CERTIFICATE

This is to certify that the project report entitled ***"Weed Detection Website"*** *is a bonfied record of the project work done by* ***Febin Vincent*** *in par- tial fulfillment of the requirement for the sixth semester of* ***Bachelor of Computer Application*** *in Department of Computer Science of* **CHRIST COLLEGE (AUTONOMOUS) IRINJALAKUDA**.

Ms. Sini Thomas
Head of the Department
Internal Guide

Ms. Sini Thomas
Head of the Department
Computer Science

EXTERNAL EXAMINER

INTERNAL EXAMINER

DECLARATION

I hereby declare that this project work **"WEED DETECTION WEB-SITE"** submitted to Christ College (Autonomous) Irinjalakuda, affiliated to Calicut University in partial fulfillment of the requirement for the award of the Bachelor of Computer Application, is a record of original work done by me, under the guidance of Ms. SINI THOMAS, Department of computer Science.

Place: Irinjalakuda FEBIN VINCENT

ACKNOWLEDGEMENT

First and foremost I would like to thank Lord almighty for his providence and for being the guiding light throughout the project. We wish to express my sincere gratitude to our beloved Department head for giving me all the facilities for our project. I take this opportunity to express my gratitude to the class teacher Ms. SOWMYA P.S and Head of the Department Ms. SINI THOMAS who has supported me throughout the course of this project. I am thankful for her aspiring guidance and valuable advice during the project work. I express my sincere thanks to my project guide Ms. SINI THOMAS for supporting and guiding throughout the project.I would take this opportunity to specially thank all other faculty members for their constant and continuous motivation. Finally i would like to thank my family and friends for giving valuable advice and moral support throughout my project.

ABSTRACT

WEED DETECTION WEBSITE offers a user-friendly platform for real-time analysis of agricultural field images. Utilizing machine learning algorithms, the site detects weeds and identifies crops, providing immediate feedback on weed presence and crop recognition. With a focus on efficiency and sustainability, the platform aims to enhance agricultural productivity and reduce herbicide usage. Ongoing development focuses on accuracy refinement and functionality expansion to become a key tool in weed management practices.

Contents

Chapter 1

1 Introduction

Welcome to our groundbreaking project at the intersection of agriculture and technology, where we harness the power of machine learning to revolutionize weed control practices. In response to the pressing need for sustainable solutions in modern agriculture, our project aims to develop an intelligent weed detection and management system. Weeds represent a persistent threat to crop yield and quality, requiring effective and eco-friendly control measures. Traditional methods fall short in terms of efficiency, cost-effectiveness, and environmental impact. By embracing machine learning algorithms, we aspire to address these challenges and pave the way for a more resilient and sustainable agricultural future. Our project focuses on leveraging machine learning techniques, such as computer vision and deep learning, to accurately identify and classify weeds in agricultural fields. Integrated with cutting-edge technologies, including drones and autonomous machinery, oursystem will enable real-time weed detection and targeted intervention. Through collaboration with farmers, researchers, and industry partners, we seek to deliver practical solutions that empower agricultural stakeholders to optimize crop production while minimizing environmental harm. Join us on this journey as we harness the power of machine learning to cultivate innovation and sustainability in agriculture.

1.1 Overview

The primary objective of this project is to develop a machine learning-based system for automated weed detection in agricultural fields. By leveraging machine learning algorithms, the system aims to accurately identify and differentiate between crops and weeds in real-time, facilitating timely and targeted weed management interventions.

Chapter 2

2 System Analysis

2.1 Purpose

The primary purpose is to develop a reliable and efficient system for automated weed detection in agricultural fields. By accurately identifying and mapping weeds, farmers can implement targeted weed management strategies, leading to reduced weed competition and improved crop health.

2.1.1 Proposed System

The project seeks to promote sustainable farming practices by reducing the reliance on chemical herbicides. By enabling precision weed control through machine learning, the project contributes to minimizing the environmental impact associated with herbicide use, such as soil contamination and water pollution. It collects high-resolution images of agricultural fields using drones, satellites, or ground-based sensors.

2.2 Problem definition

To know what the problem is and what the needs are before developing it. Here we design a website for weed detection.

2.3 FEASIBILITY STUDY

After the problem is clearly understood and solutions proposed, the next step is to conduct the feasibility study. Feasibility study is defined as evaluation or analysis of the potential impact of the proposed project or program. The objective is to determine whether the proposed system is feasible. There are three aspects of feasibility study to which the proposed system is subjected as discussed below.

2.3.1 Technical Feasibility

The proposed project of weed detection using machine learning demonstrates strong technical feasibility, leveraging advancements in data acquisition, machine learning algorithms, and computational resources. High-resolution imagery of agricultural fields, obtained through drones, satellites, or ground-based sensors, provides ample data for training machine learning models.

2.3.2 Economical Feasibility

The proposed project of weed detection using machine learning exhibits promising economical feasibility, driven by the potential for cost savings and increased

efficiency in agricultural weed management practices. By automating weed detection through machine learning, farmers can reduce labor costs associated with manual weed control and minimize expenses on chemical herbicides.

2.3.3 Operational Feasibility

The operational feasibility of the proposed project on weed detection using machine learning is promising, facilitated by the availability of scalable technology solutions and the adaptability of modern farming practices. With the increasing adoption of precision agriculture techniques, farmers are increasingly open to integrating advanced technologies into their operations.

Chapter 3

3 Software Requirement Specification

3.1 Purpose

The purpose of the Weed Detection Website project is to provide a user-friendly and accessible platform for farmers, agricultural researchers, and land managers to detect and assess the presence of weeds in images of agricultural fields. Through the integration of machine learning algorithms, users can upload images to the website and receive real-time analysis indicating the percentage of weeds present in the image, as well as whether the image contains crops.

3.2 Scope

The scope of the Weed Detection Website project encompasses the development of a user-friendly web platform allowing farmers, agricultural researchers, and land managers to upload images of agricultural fields for real-time analysis. The platform will utilize machine learning algorithms to detect weeds within the uploaded images, providing users with immediate feedback on the percentage of weeds present and the recognition of crops. Key features include image upload functionality, weed detection, crop recognition, real-time analysis, results display, and accessibility across various devices. Exclusions consist of hardware development for image capture and advanced features like predictive modeling.

3.3 Overall Description

The Weed Detection Website project aims to revolutionize weed management in agriculture by providing a user-friendly online platform for real-time analysis of images from agricultural fields. Leveraging machine learning algorithms, the website allows users to upload images and receive instant feedback on the presence of weeds and recognition of crops. By empowering farmers, agricultural researchers, and land managers with accurate and accessible tools for weed detection, this project promises to enhance efficiency, reduce herbicide usage, and promote sustainable agricultural practices.

3.3.1 Product Perspective

The Weed Detection Website provides farmers and researchers with a user-friendly platform to upload images of agricultural fields for instant analysis. Using machine learning algorithms, the site detects weeds and recognizes crops, aiding in timely decision-making and sustainable weed management practices, ultimately leading to improved crop yields and environmental stewardship. Ongoing development aims to enhance accuracy and expand functionality for widespread adoption in agriculture.

3.3.2 Product Functionality

At its core, the platform allows users to effortlessly upload images of agricultural fields for analysis. Leveraging advanced machine learning algorithms, the website accurately detects the presence of weeds and identifies crops within the uploaded images in real-time.

3.3.3 Users and Characteristics

Users of the Weed Detection Website include farmers, agricultural researchers, and land managers. They rely on the platform to monitor and manage weed infestations, optimize crop yields, and promote sustainable land management practices. Key characteristics of users include a strong interest in technology for weed management, varying technical proficiency, and a commitment to enhancing agricultural productivity and environmental stewardship.

3.4 Specific Requirements

3.4.1 Hardware Requirements

System: IBM-Compatible PC

Processor: Pentium IV or above

Speed: Above 1GHz

RAM capacity: 512 MB

Hard Dsk drive: 40 GB

Keyboard: Standard

Mouse: Standard

Monitor: SVGA Color

3.4.2 Software Requirements

Operating System:Windows or ubuntu

Languages used: PHP

Database : SQLite

Technologies used: HTML,Javascript,CSS,Python

3.5 Functional Requirements

It contains one main module.

User

User

The user or The students can register ar login the website and register for events.Also the user can view the rules and regulation of events provided in the website.And at the final the user can also get the result of each events in their login.

3.6 Non Functional Requirements

Non-functional requirements define the overall qualities or attributes of the resulting system. Non-functional system place restrictions on the product being developed, the developed process, and specify external constraints that the product must meet. Examples of non-functional requirements include safety, security, usability, reliability and performance requirements. Project management issues (costs, time and schedule) are often considered as non-functional requirements. The principal non - functional constraints which are relevant to critical systems :

performance

security

safety

usability

Performance

Performance requirements concern the speed of operation of a system. Types of performance requirements :

Response requirements (how quickly the system reacts to a user input).

Throughput requirements (how much the system can accomplish within a specified amount of time).

Availability requirements (is the system available for service when requested by end users). The speed of operation of this system is adequate for the requirements.

Reliability

Reliability is the ability of a system to perform its required function under stated conditions for a specified period of time.

constraints on the runtime behavior of the system. This system is reliable because its functionalities can be done on the required conditions.

Safety

Safety requirements are not required which exclude unsafe situation from the possible solution of the system.

Usability

Usability is the ease with which a user can learn to operate, prepare inputs for, and interpret outputs of system or components. Usability requirements include :

information error messages.

well-formed user interfaces.

3.7 Interface Requirements

3.7.1 Hardware interfaces

The system must run over the internet, all the hardware shall require to connect internet will be hardware interface for the system. As for example modem, WAN-LAN.

3.7.2 Software interfaces

Software interface required for the working of the project is the appropriate operating system.

3.7.3 Communication interfaces

The user of the site communicate using the network connectivity and the data set is accessed

3.8 Security Requirements

User accesses only their account.

Validation of input is handled.

This application containing the computer systems is physically secured against arms or surreptitious entry by intruders.

Users must be authorized carefully to reduce changes of any such user giving access to an intruder in exchange for a bribe or other favour.

3.9 Platform Used

Windows 10 is a major version of the Microsoft Windows operating system that was released on July 29, 2015.It is built on the Windows NT kernel and follows windows 8.Part of the reason Microsoft decided to name the 2015 release ”Windows 10”(and skipped ”windows 9”)is because the operating system is designed to be a new direction for Microsoft. One of the primary aims of windows 10 is to Unify the windows experience across multiple devices, such desktop computers, tablets, and smartphones. As part of this effort, Microsoft developed Windows10 Mobile alongside Windows 10 to replaces Windows Phone

- Microsoft's previous mobile OS. Windows 10 also integrates other Microsoft services, such as Xbox Live and the Cortana voice recognition assistant.

3.10 Technologies Used

PHP

PHP (recursive acronym for PHP: Hypertext Preprocessor) is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML. What distinguishes PHP from something like client-side Javascript is that the code is executed on the server, generating HTML which is then sent to the client. The client would receive the results of that script, but would not know what the underlying code was. You can even configure your web server to process all your HTML files with PHP, and then there's really no way that users can tell what you have up your sleeve. The best things in using PHP are that it is extremely simple for a newcomer, but offers many advanced features for a professional programmer.

SQLite

SQLite is a lightweight, serverless, self-contained, and open-source relational database management system (RDBMS) widely used in various applications, including mobile apps, desktop software, and web browsers. Unlike traditional client-server databases, SQLite operates without the need for a separate server process, allowing it to be embedded directly into applications and requiring minimal configuration and administration. It stores data in a single file, making it highly portable and easy to deploy. SQLite supports standard SQL syntax and provides features such as ACID transactions, indexes, and triggers, making it suitable for a wide range of use cases. Its small footprint, simplicity, and efficiency make it a popular choice for developers seeking a reliable and scalable database solution for their applications.

What distinguishes PHP from something like client - side Javascript is that the code is executed on the server, generating HTML which is then sent to the client. The client would receive the results of running that script, but would not know what the underlying code was. You can even configure your web server to process all your HTML files with PHP, and then there's really no way that users can tell what you have up your sleeve. The best things in using PHP are that it is extremely simple for a newcomer, but offers many advanced features for a professional programmer.

Chapter 4

4 Design Document

4.1 Purpose

The website will feature an intuitive user interface for uploading images of agricultural fields. Upon upload, images will undergo analysis using machine learning algorithms for weed detection and crop recognition. The system will provide real-time feedback to users, displaying the percentage of weeds present and confirming the presence of crops. The website will be designed to be accessible across various devices and screen sizes. The architecture will include a front-end interface developed using HTML, CSS, and JavaScript, a back-end server implemented using Flask, and a SQLite database to store user and image data. The system will leverage TensorFlow for machine learning tasks and will be deployed using Docker for scalability and ease of deployment.

4.2 Scope

The platform will allow users to upload images and receive instant feedback on the percentage of weeds present and the recognition of crops. Key features include image upload functionality, weed detection, crop recognition, real-time analysis, results display, and accessibility across various devices. Ongoing development will focus on refining accuracy, expanding functionality, and ensuring seamless integration into users' workflows, with the ultimate goal of becoming an indispensable tool for weed management practices.

4.3 Overview

Leveraging machine learning algorithms, the website detects weeds and recognizes crops within the uploaded images, providing users with immediate feedback on the percentage of weeds present and the presence of crops. With an intuitive interface and accessibility across various devices, the platform aims to enhance efficiency, reduce herbicide usage, and promote sustainable agricultural practices. Ongoing development focuses on refining accuracy, expanding functionality, and ensuring seamless integration into users' workflows, with the goal of becoming a valuable tool in weed management practices.

4.4 Data Design

Database are the storehouses of data used in the software systems.The data is stored in tables inside the database.Several tables are created for the manipulation of the data for the system.Two essential settings for a database are

Primary Key- The field that is unique for all the record occurrence.

Foreign Key- The field used to set relation between tables.

Normalization

Normalization is the process of organizing the data in the database.

Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate the undesirable characteristics like Insertion, Update and Deletion Anomalies.

Normalization divides the larger table into the smaller table and links them using relationship.

The normal form is used to reduce redundancy from the database table.

Tables

User Details

Name	DataType	Constraints	Description
id	int(11)	Primarykey	ID
username	varchar(100)	Notnull	Name of user
email	varchar(250)	Notnul	mail of user
password	varchar(250)	Notnul	password of user
confirm password	varchar(250)	Notnul	confirmed password

Chapter 5

5 Development of the System

This website can be decomposed into a number of submodules. The submodules of website of weed detection is student. Each sub module have specific objectives,to avoid unwanted coupling between modules and to increase cohesion, modules are again decomposed into submodules.

Chapter 6

6 System Testing

Testing is the penultimate step of software development.An elaborate testing of the data is prepared and the system is using the test data,while doing testing,errors are noted and correction is made.The users are trained to operate the developed system.Both hardware and software securities are made to run the developed system successfully.System testing is aimed at ensuring the system works accurately before the live operation commences.

6.1 Test Plan

6.1.1 Scope

This test plan will cover the following testing activities as identified in the testing strategy.

White Box Testing

White box testing is a testing technique, that examines the program structure and derives test data from the program logic/code. In white box testing, the UI is bypassed. Inputs and outputs of ratings are tested directly at the code level and the results are compared against specifications. It Reveals errors in ”hidden” code.

Black Box Testing

Black - box testing is a method of software testing that examines the functionality of an application based on the specifications. Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end - user would.

Unit Testing

The module includes four main components to undergo unit testing. It checks the syntax error, logic error and validity of the program correctness. This test will be performed by the developers.

Integration Testing

After all components pass the unit testing, they get to be tested whether they work correctly when they are running concurrently and communicating to each other. Integration tests exercise an entire subsystem and ensure that a set of components play nicely together.

Internal Data Testing

We will test the validity of the data before it enters the database to avoid any problems that may face us in the database. We will test the encryption of the personal information of all the users along with the user names and passwords to ensure maximum security of the users privacy.

6.1.2 Software risk issues

In this section, the plan is to test the risk involved in critical issues such as: Difficult to run on visual studio due to large loading time. Some inherent software risks such as complexity exist; also these issues need to be identified. Proper network connection Working of Mysql database

6.1.3 Features to be tested

Test whether correct user name and password allows you to login.

Test whether invalid user name and password prevents you from login.

Test whether there is any connection problem in the Server.

Test whether the user details are entered correctly.

Test whether invalid data entry allows saving data successfully.

Test whether all pages are loaded correctly.

Test whether the provided security works correctly.

6.2 Test consolidation

6.2.1 Test item

The items or features to be tested in the test cases are included in the document. Each and every user input is tested. The present condition of the system is tested. It is checked to make sure that environment is ready for the application to work.

6.2.2 Input specifications

These are the inputs required to execute the test case. All required inputs including data elements and values are given below.

Data	Values
Email	A valid email ID
Password	Combination of characters and numbers
Username	valid username

Chapter 7

7 System Implementation and Maintenance

7.1 Implementation

System implementation is the conversion of new system into an operating one which involves creating compatible files, training clients and installing hardware. User training is crucial for minimizing resistance to change and giving chance to prove its worth. Training aids user friendly manuals and healthy screens provide the user with a good start. Software maintenance follows conversion to the extent that changes are necessary to maintain satisfactory operations relative to changes in the user's environment. Maintenance often includes minor enhancements or corrections to the problem that surface late in the systems operations. In the implementation phase, the team builds the components either from scratch or by composition. Given the architecture document meant from the design phase and the requirement document from the analysis phase, the team should build exactly what has been requested, though there is still room for innovation and flexibility. For example, a component may be narrowly designed for this particular system, or the component may be made more general to satisfy a reusability

Careful planning

Investigation of system and constraints

Design the methods to achieve changeover.

Training the staff in the changed phase.

Evaluation of change over method.

The method of implementation and time scale to be adopted are found out initially.

7.2 Maintenance

This phase occurs as a result of deploying the whole system at the end users organization. They will perform the beta testing at the end users and inform to the developers about any needed modification to the application. The customer records all the problems that are encountered during the beta testing and reports these to the developer at regular intervals.

7.2.1 Corrective Maintenance

Even with the best quality assurance activities, it is likely that the customer will uncover defects in the software. Corrective maintenance changes the software to correct the defects. Corrective Maintenance activity may consist of repair,

restoration or replacement of equipment. This activity will be result of a regular inspection, which identifies the failure in time for corrective maintenance to be planned and scheduled, then performed during a routine maintenance shutdown.

7.2.2 Adaptive Maintenance

Over time, the original environment(CPU, operating system, business rules, external product characteristics) for which the software was developed is likely to change. Adaptive maintenance results in modification to the software to accommodate changes to its external environment.

7.2.3 Enhanced Maintenance

As software is used, the customer/user will recognize additional functions that will provide the benefit. Perfect maintenance extends the software beyond its original functional requirements.

7.2.4 Preventive Maintenance

Computer software deteriorates due to change, and because of this preventive maintenance often called software re-engineering, must be conducted to enable the software to serve the needs of its end users. Preventive maintenance makes changes to computer programs so that they can be more easily corrected, adapted and enhanced.

Chapter 8

8 Conclusion and Future Scope

8.1 Conclusion

The main objective of the project is to develop a system that can capture and scan images of crop fields and identify weeds from crops. Methodology: The project uses image processing and machine learning techniques to detect weeds. The image processing techniques include color segmentation, edge detection, and morphological operations. The machine learning techniques include feature extraction, classification, and localization. Dataset: The project uses a custom dataset of plant images, which contains images of different crop and weed species. The dataset is divided into training, validation, and testing sets.

8.2 Future Scope

Live detection

Appendix

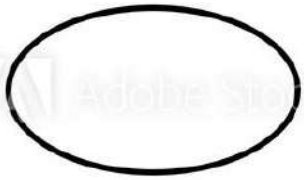
A ACTIVITY DIAGRAM

An activity diagram is a type of UML (Unified Modeling Language) diagram that visually represents the flow of control or the sequence of activities in a system, process, or workflow. It is used to model the behavior of a system and illustrate how different activities or actions relate to each other. Activity diagrams consist of nodes, which represent activities or actions, and edges, which represent the flow of control between activities. Nodes can also have additional properties, such as conditions or constraints, that determine when they are executed. Activity diagrams are particularly useful for visualizing complex processes, identifying bottlenecks, and improving the efficiency of workflows.

A.1 External source or receiver

A source or sink is a person or part of organization ,which enters or receives information from the system,but is considered to be outlining the contest of data flow model.

A.2 Transform process

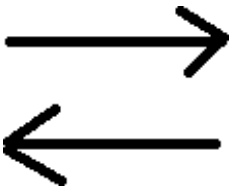


A process represents transformation where incoming data flowsare changed into outgoing data flow

A.3 Input/Output

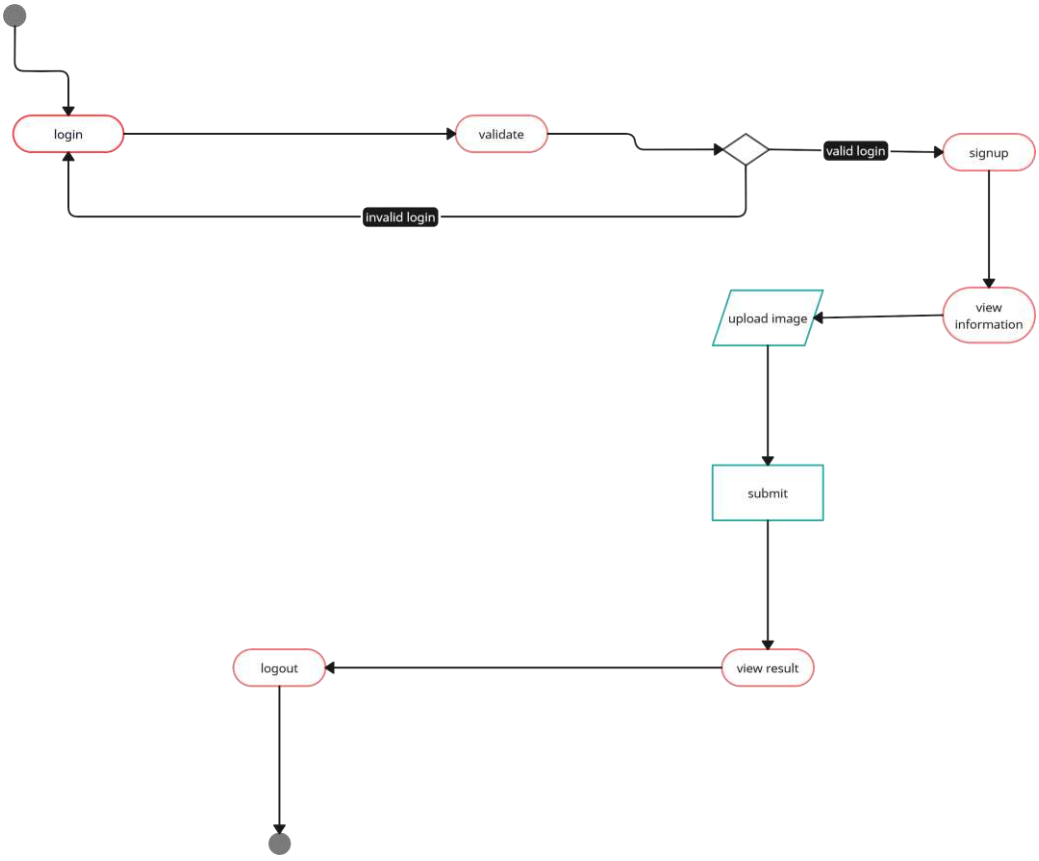
The input/output symbol represents a point in the process where data enters or exits the system. It is used to indicate the input or output of data, information, or materials at a specific point in the workflow. The input/output symbol is typically represented as a rectangle with a thin vertical line on the left side for inputs and a thin vertical line on the right side for outputs.

A.4 Data flow



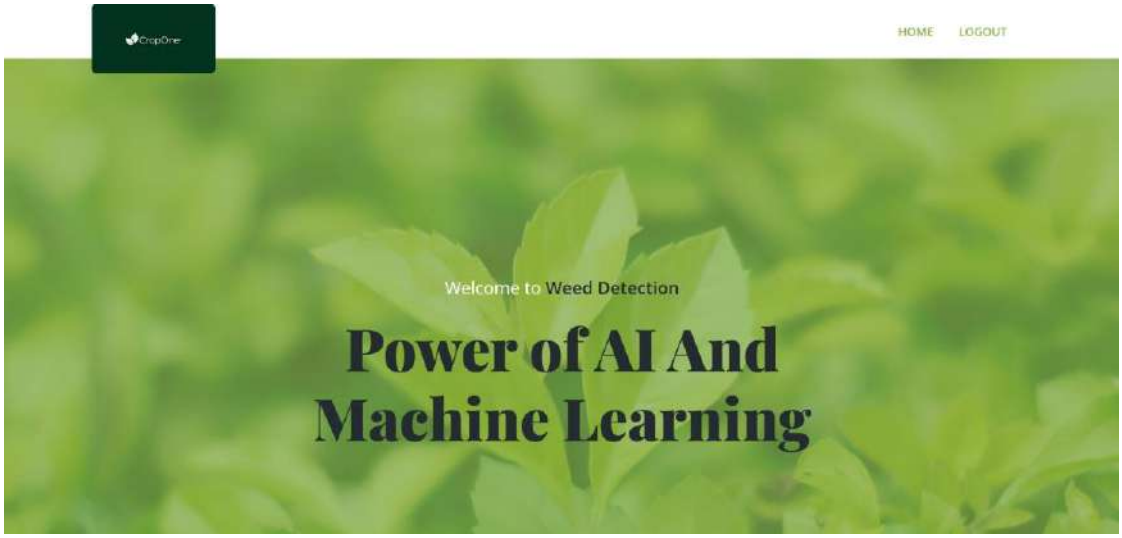
A data flow is a route, which enable packets of data to travel from one point to another.Data may flow,with arrowhead pointing in the direction of the flow

A.5 Diagram

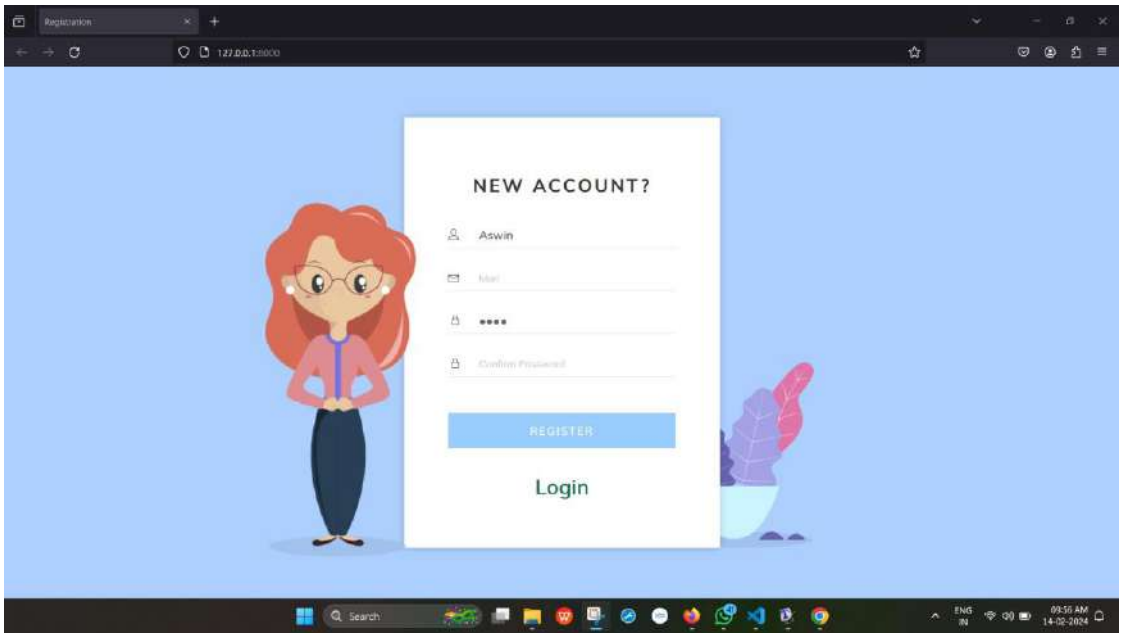

login
validate
valid login
signup
invalid login
view information
upload image
submit
view result
logout

B USER INTERFACES

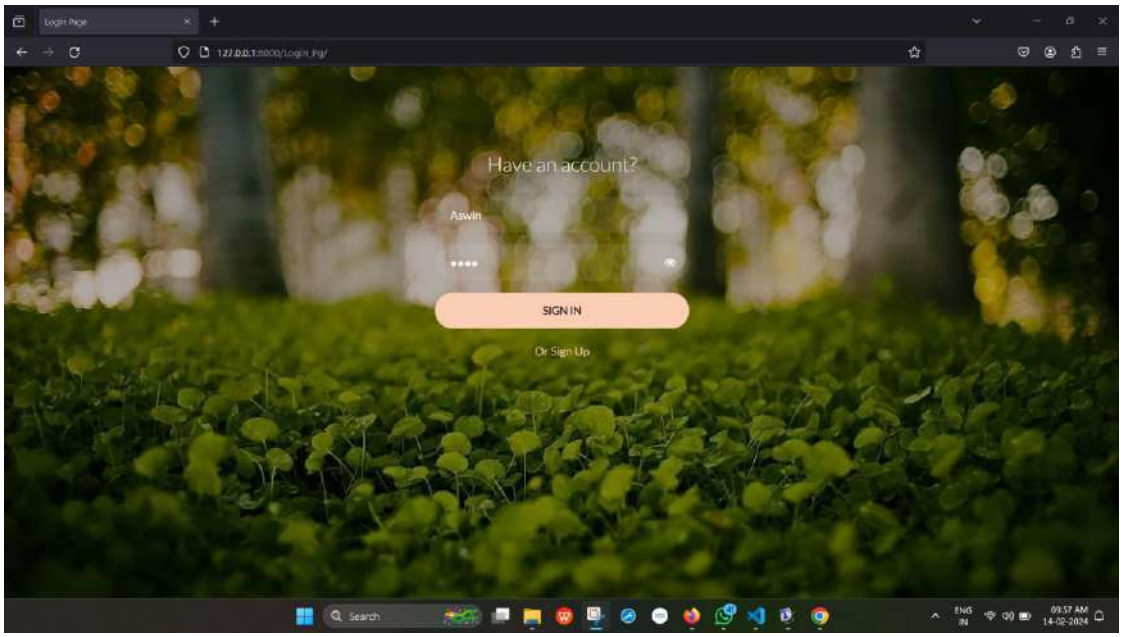
B.1 HOME



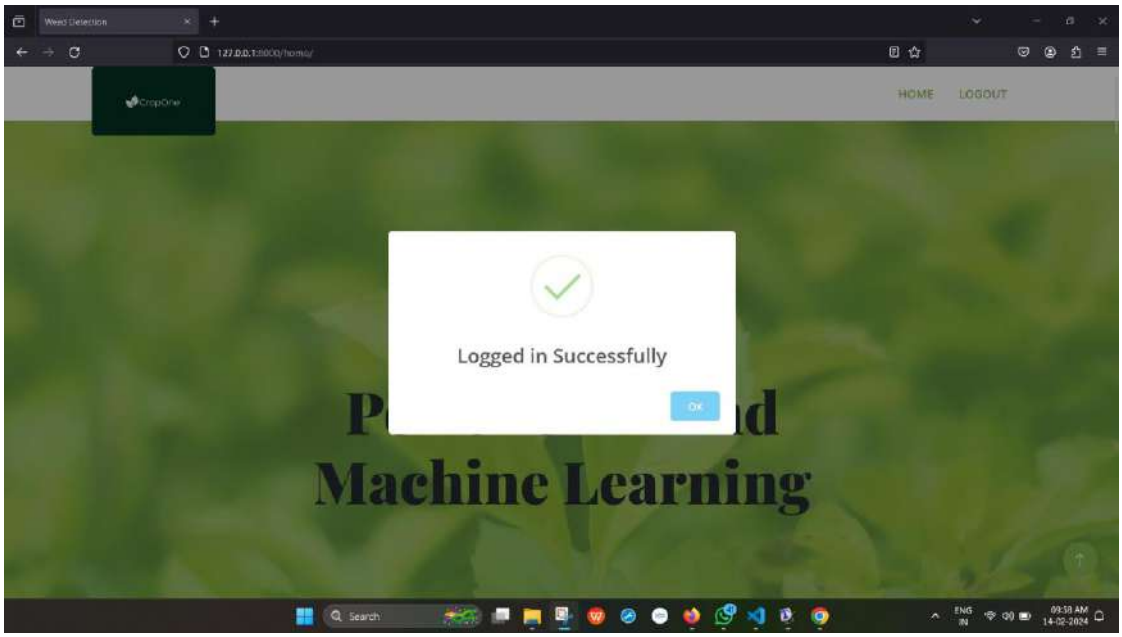
B.2 REGISTER PAGE



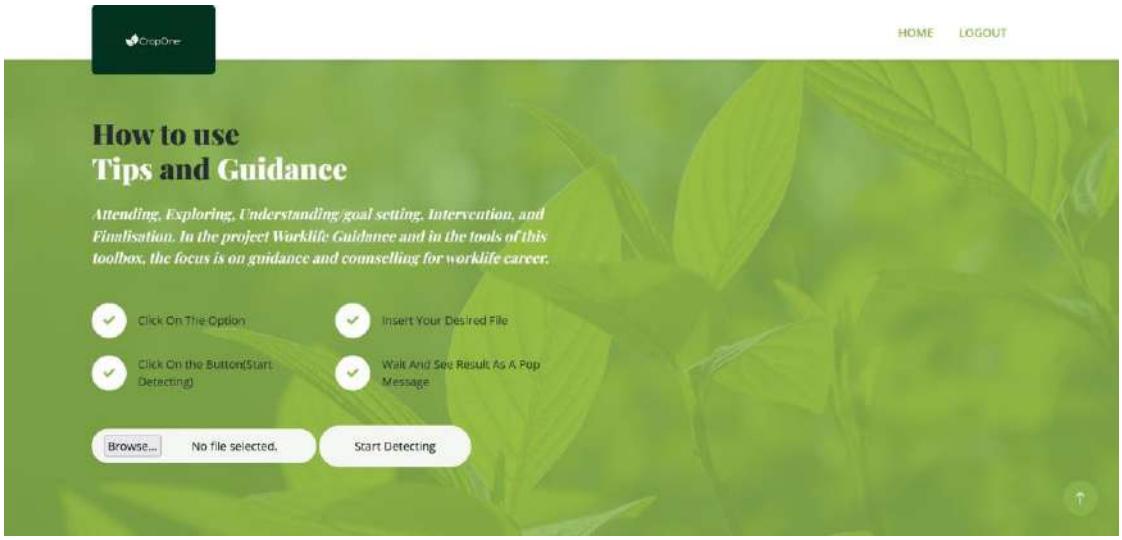
B.3 SIGNUP PAGE



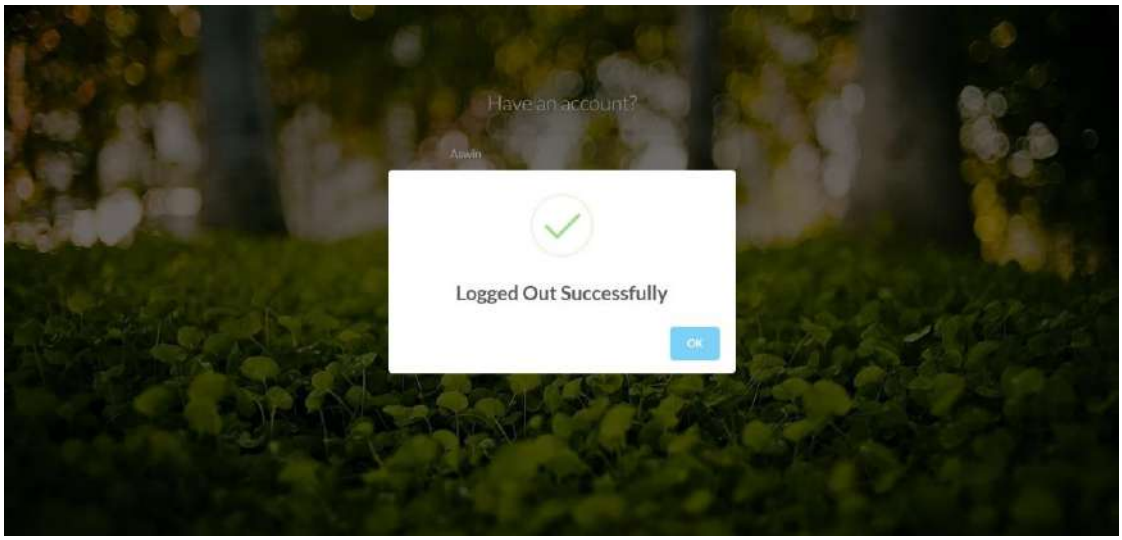
B.4 LOGIN CONFIRMATION



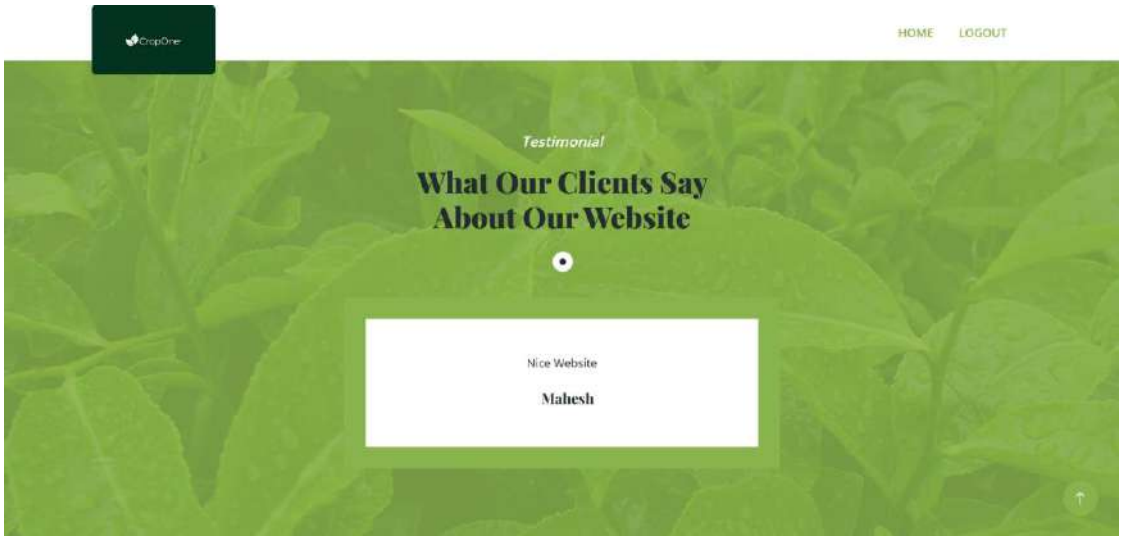
B.5 USER INPUT



B.6 LOGOUT CONFIRMATION



B.7 REVIEW


CropOne
HOME
LOGOUT
Testimonial
What Our Clients Say
About Our Website
Nice Website
Mahesh

C CODE

style.css

```
/********** Template CSS **********/
:root {
    --primary: #88B44E;
    --secondary: #FB9F38;
    --light: #F5F8F2;
    --dark: #252C30;
}

.back-to-top {
    position: fixed;
    display: none;
    right: 30px;
    bottom: 30px;
    z-index: 99;
}

.fw-medium {
    font-weight: 600;
}

.fw-bold {
    font-weight: 700;
}

.fw-black {
    font-weight: 900;
}


/*** Spinner ***/
#spinner {
    opacity: 0;
    visibility: hidden;
    transition: opacity .5s ease-out, visibility 0s linear .5s;
    z-index: 99999;
}

#spinner.show {
    transition: opacity .5s ease-out, visibility 0s linear 0s;
    visibility: visible;
    opacity: 1;
}
```

```
/*** Button ***/
.btn {
    transition: .5s;
    font-weight: 500;
}

.btn-primary,
.btn-outline-primary:hover {
    color: var(--light);
}

.btn-secondary,
.btn-outline-secondary:hover {
    color: var(--dark);
}

.btn-square {
    width: 38px;
    height: 38px;
}

.btn-sm-square {
    width: 32px;
    height: 32px;
}

.btn-lg-square {
    width: 48px;
    height: 48px;
}

.btn-square,
.btn-sm-square,
.btn-lg-square {
    padding: 0;
    display: flex;
    align-items: center;
    justify-content: center;
    font-weight: normal;
}


/*** Navbar ***/
.sticky-top {
```

```
    top: -150px;
    transition: .5s;
}

.navbar .navbar-brand {
    position: absolute;
    padding: 0;
    width: 170px;
    height: 135px;
    top: 0;
    left: 0;
}

.navbar .navbar-nav .nav-link {
    margin-right: 35px;
    padding: 25px 0;
    color: var(--dark);
    font-weight: 600;
    text-transform: uppercase;
    outline: none;
}

.navbar .navbar-nav .nav-link:hover,
.navbar .navbar-nav .nav-link.active {
    color: var(--primary);
}

.navbar .dropdown-toggle::after {
    border: none;
    content: "\f107";
    font-family: "Font Awesome 5 Free";
    font-weight: 900;
    vertical-align: middle;
    margin-left: 8px;
}

@media (max-width: 991.98px) {
    .navbar .navbar-brand {
        width: 126px;
        height: 100px;
    }

    .navbar .navbar-nav .nav-link  {
        margin-right: 0;
        padding: 10px 0;
    }
```

```
    .navbar .navbar-nav {
        margin-top: 75px;
        border-top: 1px solid #EEEEEE;
    }
}

@media (min-width: 992px) {
    .navbar .nav-item .dropdown-menu {
        display: block;
        border: none;
        margin-top: 0;
        top: 150%;
        opacity: 0;
        visibility: hidden;
        transition: .5s;
    }

    .navbar .nav-item:hover .dropdown-menu {
        top: 100%;
        visibility: visible;
        transition: .5s;
        opacity: 1;
    }
}


/*** Header ***/
.carousel-caption {
    top: 0;
    left: 0;
    right: 0;
    bottom: 0;
    display: flex;
    align-items: center;
    background: rgba(136, 180, 78, .7);
    z-index: 1;
}

.carousel-control-prev,
.carousel-control-next {
    width: 15%;
}

.carousel-control-prev-icon,
.carousel-control-next-icon {
```

```
    width: 3.5rem;
    height: 3.5rem;
    border-radius: 3.5rem;
    background-color: var(--dark);
    border: 15px solid var(--dark);
}

@media (max-width: 768px) {
    #header-carousel .carousel-item {
        position: relative;
        min-height: 450px;
    }

    #header-carousel .carousel-item img {
        position: absolute;
        width: 100%;
        height: 100%;
        object-fit: cover;
    }
}

.page-header {
background: linear-gradient(rgba(136, 180, 78, .7), rgba(136,
180, 78, .7)), url(../img/carousel-1.jpg) center center
no-repeat;
    background-size: cover;
}

.page-header .breadcrumb-item+.breadcrumb-item::before {
    color: var(--light);
}

.page-header .breadcrumb-item,
.page-header .breadcrumb-item a {
    font-size: 18px;
    color: var(--light);
}


/*** Section Title ***/
.section-title    {
    position: relative;
    margin-bottom: 3rem;
    padding-bottom: 2rem;
}
```

```
.section-title::before {
    position: absolute;
    content: "";
    width: 50%;
    height: 2px;
    bottom: 0;
    left: 0;
    background: var(--primary);
}

.section-title::after  {
    position: absolute;
    content: "";
    width: 28px;
    height: 28px;
    bottom: -13px;
    left: calc(25% - 13px);
    background: var(--dark);
    border: 10px solid #FFFFFF;
    border-radius: 28px;
}

.section-title.text-center::before {
    left: 25%;
}

.section-title.text-center::after {
    left: calc(50% - 13px);
}


/*** Products ***/
.product {
background: linear-gradient(rgba(136, 180, 78, .1), rgba(136,
180, 78, .1)),
    url(../img/product-bg.png) left bottom no-repeat;
    background-size: auto;
}

.product-carousel .owl-nav {
    display: flex;
    justify-content: center;
    margin-top: 30px;
}

.product-carousel .owl-nav .owl-prev,
```

```
.product-carousel .owl-nav .owl-next {
    margin: 0 10px;
    width: 55px;
    height: 55px;
    display: flex;
    align-items: center;
    justify-content: center;
    color: #FFFFFF;
    background: var(--primary);
    border-radius: 55px;
    box-shadow: 0 0 45px rgba(0, 0, 0, .15);
    font-size: 25px;
    transition: .5s;
}

.product-carousel .owl-nav .owl-prev:hover,
.product-carousel .owl-nav .owl-next:hover {
    background: #FFFFFF;
    color: var(--primary);
}


/*** About ***/
.video {
background: linear-gradient(rgba(136, 180, 78, .85), rgba(136,
180, 78, .85)),
    url(../img/video-bg.jpg) center center no-repeat;
    background-size: cover;
}

.btn-play {
    position: relative;
    display: block;
    box-sizing: content-box;
    width: 65px;
    height: 75px;
    border-radius: 100%;
    border: none;
    outline: none !important;
    padding: 28px 30px 30px 38px;
    background: #FFFFFF;
}

.btn-play:before {
    content: "";
    position: absolute;
```

```
    z-index: 0;
    left: 50%;
    top: 50%;
    transform: translateX(-50%) translateY(-50%);
    display: block;
    width: 120px;
    height: 120px;
    background: #FFFFFF;
    border-radius: 100%;
    animation: pulse-border 1500ms ease-out infinite;
}

.btn-play:after {
    content: "";
    position: absolute;
    z-index: 1;
    left: 50%;
    top: 50%;
    transform: translateX(-50%) translateY(-50%);
    display: block;
    width: 120px;
    height: 120px;
    background: #FFFFFF;
    border-radius: 100%;
    transition: all 200ms;
}

.btn-play span {
    display: block;
    position: relative;
    z-index: 3;
    width: 0;
    height: 0;
    left: 13px;
    border-left: 40px solid var(--primary);
    border-top: 28px solid transparent;
    border-bottom: 28px solid transparent;
}

@keyframes pulse-border {
    0% {
transform: translateX(-50%) translateY(-50%) translateZ(0)
scale(1);
        opacity: 1;
    }
```

```
    100% {
transform: translateX(-50%) translateY(-50%) translateZ(0)
scale(2);
        opacity: 0;
    }
}

.modal-video .modal-dialog {
    position: relative;
    max-width: 800px;
    margin: 60px auto 0 auto;
}

.modal-video .modal-body {
    position: relative;
    padding: 0px;
}

.modal-video .close {
    position: absolute;
    width: 30px;
    height: 30px;
    right: 0px;
    top: -30px;
    z-index: 999;
    font-size: 30px;
    font-weight: normal;
    color: #FFFFFF;
    background: #000000;
    opacity: 1;
}


/*** Store ***/
.store-item .store-overlay {
    position: absolute;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: center;
    background: rgba(138, 180, 78, .3);
    opacity: 0;
```

```
    transition: .5s;
}

.store-item:hover .store-overlay {
    opacity: 1;
}


/*** Contact ***/
.contact .btn-square {
    width: 100px;
    height: 100px;
    border: 20px solid var(--light);
    background: var(--primary);
    border-radius: 50px;
}


/*** Testimonial ***/
.testimonial {
background: linear-gradient(rgba(136, 180, 78, .85), rgba(136,
180, 78, .85)),
    url(../img/testimonial-bg.jpg) center center no-repeat;
    background-size: cover;
}

.testimonial-item {
    margin: 0 auto;
    max-width: 600px;
    text-align: center;
    background: #FFFFFF;
    border: 30px solid var(--primary);
}

.testimonial-item img {
    width: 60px !important;
    height: 60px !important;
    border-radius: 60px;
}

.testimonial-carousel .owl-dots {
    margin-top: 35px;
    display: flex;
    align-items: flex-end;
    justify-content: center;
}
```

```
.testimonial-carousel .owl-dot {
    position: relative;
    display: inline-block;
    margin: 0 5px;
    width: 15px;
    height: 15px;
    background: var(--primary);
    border-radius: 15px;
    transition: .5s;
}

.testimonial-carousel .owl-dot.active {
    width: 30px;
    background: var(--dark);
}


/*** Footer ***/
.footer {
    color: #B0B9AE;
}

.footer .btn.btn-link {
    display: block;
    margin-bottom: 5px;
    padding: 0;
    text-align: left;
    color: #B0B9AE;
    font-weight: normal;
    text-transform: capitalize;
    transition: .3s;
}

.footer .btn.btn-link::before {
    position: relative;
    content: "\f105";
    font-family: "Font Awesome 5 Free";
    font-weight: 900;
    color: var(--primary);
    margin-right: 10px;
}

.footer .btn.btn-link:hover {
    color: var(--light);
    letter-spacing: 1px;
```

```
    box-shadow: none;
}

.copyright {
    color: #B0B9AE;
}

.copyright {
    background: #252525;
}

.copyright a:hover {
    color: #FFFFFF !important;
}
```

home.html

```
<!DOCTYPE html>
{% load static %}
<html lang="en">

<head>
<meta charset="utf-8">
<title>Weed Detection</title>
<meta content="width=device-width, initial-scale=1.0"
name="viewport">
<meta content="" name="keywords">
<meta content="" name="description">

<!-- Favicon -->
<link href="{% static 'img/favicon.ico' %}" rel="icon">

<!-- Google Web Fonts -->
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com"
crossorigin>
<link href="https://fonts.googleapis.com/css2?
family=Open+Sans:wght@400;600&family=Playfair+Display:wght@700;900&display=swap"
rel="stylesheet">

<!-- Icon Font Stylesheet -->
<link
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.10.0/css/all.min.css"
rel="stylesheet">
<link
```

```
href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.4.1/font/bootstrap-icons.css"
rel="stylesheet">

<!-- Libraries Stylesheet -->
<link href="{% static 'lib/animate/animate.min.css' %}"
rel="stylesheet">
<link href="{% static
'lib/owlcarousel/assets/owl.carousel.min.css' %}"
rel="stylesheet">

<!-- Customized Bootstrap Stylesheet -->
<link href="{% static 'css/bootstrap.min.css' %}"
rel="stylesheet">

<!-- Template Stylesheet -->
<link href="{% static 'css/style.css' %}" rel="stylesheet">
</head>

<body>
<!-- Spinner Start -->
<div id="spinner" class="show bg-white position-fixed
translate-middle w-100 vh-100 top-50
start-50 d-flex align-items-center justify-content-center">
<div class="spinner-border text-primary" role="status"
style="width: 3rem; height: 3rem;"></div>
</div>
<!-- Spinner End -->


<!-- Navbar Start -->
<div class="container-fluid bg-white sticky-top">
<div class="container">
<nav class="navbar navbar-expand-lg bg-white navbar-light py-2
py-lg-0">
<a href="index.html" class="navbar-brand">
<img class="img-fluid" src="{% static 'img/crop1.jpg' %}"
alt="Logo" width="200px" height="60px">
</a>
<button type="button" class="navbar-toggler ms-auto me-0"
data-bs-toggle="collapse" data-bs-target="#navbarCollapse">
<span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="navbarCollapse">
<div class="navbar-nav ms-auto">
<a href="{% url 'home' %}" class="nav-item nav-link
active">Home</a>
```

```
{% if request.session.username %}
<a href="{% url 'Logout_fn' %}" class="nav-item nav-link
active">Logout</a>
{% else %}
<a href="{% url 'RegistrationForm' %}" class="nav-item nav-link
active">Register</a>
{% endif %}

</div>
</div>



</nav>
</div>
</div>
<!-- Navbar End -->


<!-- Carousel Start -->
<div class="container-fluid px-0 mb-5">
<div id="header-carousel" class="carousel slide carousel-fade"
data-bs-ride="carousel">
<div class="carousel-inner">
<div class="carousel-item active">
<img class="w-100" src="{% static 'img/carousel-1.jpg' %}"
alt="Image">
<div class="carousel-caption">
<div class="container">
<div class="row justify-content-center">
<div class="col-lg-7 text-center">
<p class="fs-4 text-white animated zoomIn">Welcome to
<strong class="text-dark">Weed Detection</strong></p>
<h1 class="display-1 text-dark mb-4 animated zoomIn">
Power of AI And Machine Learning</h1>
</div>
</div>
</div>
</div>
</div>

</div>
</div>
</div>
<!-- Carousel End -->
```

```
<!-- About Start -->
<div class="container-xxl py-5">
<div class="container">
<div class="row g-5">
<div class="col-lg-6">
<div class="row g-3">
<div class="col-6 text-end">
<img class="img-fluid bg-white w-200 mb-3 wow fadeIn"
data-wow-delay="0.1s" src="{% static 'img/cropandweed.jpeg' %}"
alt="">
<img class="img-fluid bg-white w-300 wow fadeIn"
data-wow-delay="0.2s" src="{% static 'img/cropandweed2.jpeg' %}"
alt="">
</div>
<div class="col-6">
<img class="img-fluid bg-white w-200 mb-3 wow fadeIn"
data-wow-delay="0.3s" src="{% static 'img/cropandweed3.jpeg' %}"
alt="">
<img class="img-fluid bg-white w-300 wow fadeIn"
data-wow-delay="0.4s" src="{% static 'img/cropandweed4.jpeg' %}"
alt="">
</div>
</div>
</div>
<div class="col-lg-6 wow fadeIn" data-wow-delay="0.5s">
<div class="section-title">
<p class="fs-5 fw-medium fst-italic text-primary">About
Project</p>
<h1 class="display-6">What is weed detection?</h1>
</div>
<div class="row g-3 mb-4">
<div class="col-sm-4">
<img class="img-fluid bg-white w-150" src="{% static
'img/cropandweed5.jpeg' %}" alt="">
</div>
<div class="col-sm-8">
<h5>Weed detection systems are important solutions to one of the
existing
agricultural problems|unmechanized weed control. </h5>
<p class="mb-0"> Weed detection also helps provide a means of
reducing or eliminating herbicide use, mitigating agricultural
environmental
and health impact, and improving sustainability.</p>
</div>
</div>
```

```
<div class="border-top mb-4"></div>
<div class="row g-3">
<div class="col-sm-8">
<h5>What is the objective for weed detection project?</h5>
<p class="mb-0">The main objective has been to obtain a formula
so that a
weed detection system can be developed through binary
classifications.
The initial step of image processing is the detection of green
plants in order to
eliminate all the soil in the image, reducing information that
is not necessary.</p>
</div>
<div class="col-sm-4">
<img class="img-fluid bg-white w-150" src="{% static
'img/cropandweed5.jpeg' %}" alt="">
</div>
</div>
</div>
</div>
</div>
</div>
<!-- About End -->


<!-- Products Start -->
<div class="container-fluid product py-5 my-5">
<div class="container py-5">
<div class="section-title text-center mx-auto wow fadeInUp"
data-wow-delay="0.1s"
 style="max-width: 500px;">
<p class="fs-5 fw-medium fst-italic text-primary">Our Source</p><h1 class="display-6">
What is the role of AI in
agriculture?</h1>
</div>
<div class="owl-carousel product-carousel wow fadeInUp"
data-wow-delay="0.5s">
<a href="" class="d-block product-item rounded">
<img src="img/product-1.jpg" alt="">
<div class="bg-white shadow-sm text-center p-4 position-relative
mt-n5 mx-4">
<h4 class="text-primary">AI Specification</h4>
<span class="text-body">By analyzing data from various sources,
AI can help farmers make data-driven decisions,
optimize resource usage, and reduce environmental impact.AI can
be used to develop
```

```
 image recognition systems that can.
identify weeds with a high degree of accuracy</span>
</div>
</a>
<a href="" class="d-block product-item rounded">
<img src="img/product-2.jpg" alt="">
<div class="bg-white shadow-sm text-center p-4 position-relative
mt-n5 mx-4">
<h4 class="text-primary">General Science</h4>
<span class="text-body">For example, the World Economic Forum
has reported that
 AI integration in agriculture could bring about
a 60% decrease in pesticide usage and a 50% reduction in water
usage.
AI algorithms can analyze the chemical composition of soil
samples to determine which nutrients may be lacking.
AI can also identify or even predict crop diseases. </span>
</div>
</a>
<a href="" class="d-block product-item rounded">
<img src="img/product-3.jpg" alt="">
<div class="bg-white shadow-sm text-center p-4 position-relative
mt-n5 mx-4">
<h4 class="text-primary">Weed</h4>
<span class="text-body">Weeds are unwanted and undesirable
 plants which interfere with the. utilization of land and water
 resources and thus adversely affect human welfare.
They can also be referred as plants out of place. Weeds compete
with the
beneficial and desired vegetation in crop lands, forests,
aquatic systems etc.</span>
</div>
</a>
</div>
</div>
</div>
<!-- Products End -->


<!-- Article Start -->
<div class="container-xxl py-5">
<div class="container">
<div class="row g-5">
<div class="col-lg-5 wow fadeIn" data-wow-delay="0.1s">
<img class="img-fluid" src="{% static 'img/article.jpg' %}"
alt="">
```

```
</div>
<div class="col-lg-6 wow fadeIn" data-wow-delay="0.5s">
<div class="section-title">
<p class="fs-5 fw-medium fst-italic text-primary">Featured
Acticle</p>
<h1 class="display-6">The history of Weed Detection</h1>
</div>
<p class="mb-4">In olden days weed detection was done
by employing some men, especially for weed removal purpose.
They will detect the weeds by checking each and every plant
field
. Then they will pluck them out manually using their
hands or spades. In proposed system the weeds can be detected
by image processing techniques. The main aim of the
project is to identify weed affected area for more seeding.
Since there are different issues with respect to the
developed yields on the field, a standout amongst the majority
is about
weeds which go as a hindrance in the development of the
harvests. Weeds may break down the life and nature of the yields
if
it is not controlled legitimately in time.
This proposed thought centers around to reduce the work and also
the time
needed to identify weeds and expel the same.
During this process they will consider Edge detection and
colour segmentation process.
They will analyze each and every crop with their intensity
colour,
Intensity, edge, Size etc., are been obtained as output.
The colour of the edges and veins of the crop and weed are
white after segmentation process and edge detection,
whereas the rest of the image is totally black in colour.
After the process of Edge detection and Colour
segmentation it has undergone the process of Filtering.
The Filtering can be a type of process in which each and every
crop can be identified and their gain value,
tradeoff's, edge, frequency of crop and weed can be identified
and their threshold values can obtained </p>

</div>
</div>
</div>
</div>
<!-- Article End -->
```

```
<!-- Video Start -->
<div class="container-fluid video my-5">
<div class="container">
<div class="row g-0">
<div class="col-lg-6 py-5 wow fadeIn" data-wow-delay="0.1s">
<div class="py-5">
<h1 class="display-6 mb-4">How to use <br><span
class="text-white">
Tips</span> and <span class="text-white">Guidance</span></h1>
<h5 class="fw-normal lh-base fst-italic text-white
mb-5">Attending, Exploring
, Understanding/goal setting, Intervention, and Finalisation.
In the project Worklife Guidance and in the tools of this
toolbox,
the focus is on guidance and counselling for worklife
career.</h5>
<div class="row g-4 mb-5">
<div class="col-sm-6">
<div class="d-flex align-items-center">
<div class="flex-shrink-0 btn-lg-square bg-white text-primary
rounded-circle me-3">
<i class="fa fa-check"></i>
</div>
<span class="text-dark">Click On The Option</span>
</div>
</div>
<div class="col-sm-6">
<div class="d-flex align-items-center">
<div class="flex-shrink-0 btn-lg-square bg-white text-primary
rounded-circle me-3">
<i class="fa fa-check"></i>
</div>
<span class="text-dark">Insert Your Desired File</span>
</div>
</div>
<div class="col-sm-6">
<div class="d-flex align-items-center">
<div class="flex-shrink-0 btn-lg-square bg-white text-primary
rounded-circle me-3">
<i class="fa fa-check"></i>
</div>
<span class="text-dark">Click On the Button(Start
Detecting)</span>
</div>
```

```
</div>
<div class="col-sm-6">
<div class="d-flex align-items-center">
<div class="flex-shrink-0 btn-lg-square bg-white text-primary
rounded-circle me-3">
<i class="fa fa-check"></i>
</div>
<span class="text-dark">Wait And See Result As A Pop
Message</span>
</div>
</div>
</div>
<form action="{% url 'detect_weed_or_crop' %}" method="post"
enctype="multipart/form-data">
{% csrf_token %}
<input type="file" class="btn btn-light rounded-pill py-2 px-3"
required name="image_file">
{% if request.session.username %}
<button class="btn btn-light rounded-pill py-3 px-5">Start
Detecting</button>
</form>
{% else %}


<a href="{% url 'RegistrationForm' %}" class="btn btn-light
rounded-pill py-3 px-5">Register</a>



{% endif %}


<div class="modal fade" id="exampleModal" tabindex="-1"
aria-labelledby="exampleModalLabel" aria-hidden="true">
<div class="modal-dialog">
<div class="modal-content">
<div class="modal-header">
<h1 class="modal-title fs-5" id="exampleModalLabel">Result</h1>
<button type="button" class="btn-close" data-bs-dismiss="modal"
aria-label="Close"></button>
</div>
<div class="modal-body">
{{result.result}}
</div>
<div class="modal-footer">
<button type="button" class="btn btn-secondary"
```

```
data-bs-dismiss="modal">Close</button>

</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>


<!-- Testimonial Start -->
<div class="container-fluid testimonial py-5 my-5">

<div class="container py-5">
<div class="section-title text-center mx-auto wow fadeInUp"
 data-wow-delay="0.1s" style="max-width: 500px;">
<p class="fs-5 fw-medium fst-italic text-white">Testimonial</p>
<h1 class="display-6">What Our Clients Say About Our
Website</h1>
</div>

<div class="owl-carousel testimonial-carousel wow fadeInUp"
data-wow-delay="0.5s">
{% for i in x %}
<div class="testimonial-item p-4 p-lg-5">
<p class="mb-4">{{i.Description}}</p>
<div class="d-flex align-items-center justify-content-center">
<div class="text-start ms-3">
<h5>{{i.username}}</h5>
</div>
</div>
</div>
{% endfor %}
</div>

</div>

</div>
<!-- Testimonial End -->


<!-- Contact Start -->
<div class="container-xxl contact py-5">
```

```
<div class="container">
<div class="section-title text-center mx-auto wow fadeInUp"
 data-wow-delay="0.1s" style="max-width: 500px;">
<h1 class="display-6">Overview</h1>
</div>
<div class="row justify-content-center wow fadeInUp"
data-wow-delay="0.1s">
<div class="col-lg-8">
<p class="text-center mb-5">Artificial intelligence,
specifically deep learning, is a fast-growing research field
today. One of its
various applications is object recognition,
making use of computer vision. The combination of these
two technologies leads to the purpose of this thesis.
 In this project, a system for the identification of
different crops and weeds has been developed as an
alternative to the system present on the FarmBot
company's robots. This is done by accessing the
images through the FarmBot API, using computer
vision for image processing, and artificial intelligence
for the application of transfer learning to a RCNN
that performs the plants identification autonomously.
 The results obtained show that the system
works with an accuracy of 78.10
% for the main crop and 53.12% and 44.76% for the two weeds
considered. Moreover, the coordinates of the weeds
 are also given as results. The performance of the
resulting system is compared both with
 similar projects found during research, and with the current
version of the FarmBot weed detector. </p>
<div class="row g-5">
<div class="col-md-4 text-center wow fadeInUp"
data-wow-delay="0.3s">
<div class="btn-square mx-auto mb-3">
<i class="fa fa-envelope fa-2x text-white"></i>
</div>
<p class="mb-2">name@example.com</p>
<p class="mb-0">name1@example.com</p>
</div>
<div class="col-md-4 text-center wow fadeInUp"
data-wow-delay="0.4s">
<div class="btn-square mx-auto mb-3">
<i class="fa fa-phone fa-2x text-white"></i>
</div>
<p class="mb-2">+012 345 67890</p>
</div>
```

```
<div class="col-md-4 text-center wow fadeInUp"
data-wow-delay="0.5s">
<div class="btn-square mx-auto mb-3">
<i class="fa fa-map-marker-alt fa-2x text-white"></i>
</div>
<p class="mb-2">Irinjalakuda</p>
<p class="mb-0">Thrissur, Kerala</p>
</div>
</div>
</div>
</div>
</div>
</div>
<!-- Contact Start -->

<form action="{% url 'ReviewSave' %}" method="post">
{% csrf_token %}

<div style="margin-left:450px;">
<div class="form-floating">
<input type="hidden" name="uname"
value="{{request.session.username}}">
<input class="form-control" placeholder="Insert Your Review
Here"
id="message" type="text" name="txt" style="height:
120px;width:550px;">
<label for="message">Review</label>
</div>
<br>
<div class="col-12">
{% if request.session.username %}
<button class="btn btn-primary rounded-pill py-3 px-4"
style="margin-left:200px;" type="submit">
Submit Review</button>

{% else %}
<a href="{% url 'RegistrationForm' %}" class="btn btn-primary
rounded-pill py-3 px-4"
style="margin-left:200px;">Register</a>


{% endif %}


</div>
</div>
```

```
</form>


<!-- Footer Start -->
<div class="container-fluid bg-dark footer mt-5 py-5 wow fadeIn"
data-wow-delay="0.1s">
<div class="container py-5">
<div class="row g-5">
<div class="col-lg-3 col-md-6">
<h4 class="text-primary mb-4">Authencity</h4>
<p class="mb-2"><i class="fa fa-map-marker-alt text-primary
me-3">
</i>Fort Street, Ernakulam , Kerala</p>
<p class="mb-2"><i class="fa fa-phone-alt text-primary me-3
"></i>+012 345 67890</p>
<p class="mb-2"><i class="fa fa-envelope text-primary me-3">
</i>name@example.com</p>
<div class="d-flex pt-3">
<a class="btn btn-square btn-primary rounded-circle me-2"
href="#">
<i class="fab fa-twitter"></i></a>
<a class="btn btn-square btn-primary rounded-circle me-2"
href="#">
<i class="fab fa-facebook-f"></i></a>
<a class="btn btn-square btn-primary rounded-circle me-2"
href="#">
<i class="fab fa-youtube"></i></a>
<a class="btn btn-square btn-primary rounded-circle me-2"
href="#">
<i class="fab fa-linkedin-in"></i></a>
</div>
</div>
<div class="col-lg-3 col-md-6">
<h4 class="text-primary mb-4">Quick Links</h4>
<a class="btn btn-link" href="{% url 'home' %}">Home</a>
{% if request.session.username%}
<a class="btn btn-link" href="{% url 'Logout_fn' %}">Logout</a>
{% else %}
<a class="btn btn-link" href="{% url 'RegistrationForm' %}">Sign
Up</a>
{% endif %}
</div>
<div class="col-lg-3 col-md-6">
<h4 class="text-primary mb-4">Business Hours</h4>
<p class="mb-1">Monday - Saturday</p>
<h6 class="text-light">24/5</h6>
```

```
<p  class="mb-1">Saturday</p>
<h6 class="text-light">09:00 am - 12:00 pm</h6>
<p class="mb-1">Sunday</p>
<h6   class="text-light">Closed</h6>
</div>
</div>
</div>
</div>
<!-- Footer End -->


<!-- Copyright Start -->
<div class="container-fluid copyright py-4">
<div class="container">
<div class="row">
<div class="col-md-6 text-center text-md-start mb-3 mb-md-0">
&copy; <a class="fw-medium" href="#">Weed Detection</a>, All
Right Reserved.
</div>
<div class="col-md-6 text-center text-md-end">
<!--/*** This template is free as long as you keep the footer
author's credit
link/attribution link/backlink. If you'd like to use the
template without the
footer author's credit link/attribution link/backlink, you can
purchase the
Credit Removal License from
"https://htmlcodex.com/credit-removal".
Thank you for your support. ***/-->
</div>
</div>
</div>
</div>
<!-- Copyright End -->


<!-- Back to Top -->
<a href="#" class="btn btn-lg btn-primary btn-lg-square
rounded-circle back-to-top">
<i class="bi bi-arrow-up"></i></a>


<!-- JavaScript Libraries -->
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.1/jquery.min.js"></script>
<script
```

```
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0/dist/js/bootstrap.bundle.min.js">
</script>
<script src="{% static 'lib/wow/wow.min.js' %}"></script>
<script src="{% static 'lib/easing/easing.min.js' %}"></script>
<script src="{% static 'lib/waypoints/waypoints.min.js'
%}"></script>
<script src="{% static 'lib/owlcarousel/owl.carousel.min.js'
%}"></script>

<!-- Template Javascript -->
<script src="{% static 'js/main.js' %}"></script>
<script
src="https://unpkg.com/sweetalert/dist/sweetalert.min.js"></script>
{% if messages %}
{% for i in messages %}
{% if i.tags == 'warning' %}
<script>
swal('{{i}}','', 'warning');
</script>
{% elif i.tags == 'error' %}
<script>
swal('{{i}}','', 'error');
</script>
{% else %}
<script>
swal('{{i}}','', 'success');
</script>
{% endif %}

{% endfor %}
{% endif %}
</body>


</html>
```

login.html

```
<!DOCTYPE html>
{% load static %}
<html>
<head>
<meta charset="utf-8">
<title>Registration</title>
<meta name="viewport" content="width=device-width,
```

```
initial-scale=1.0">

<!-- LINEARICONS -->
<link rel="stylesheet" href="{% static
'assets/fonts/linearicons/style.css' %}">

<!-- STYLE CSS -->
<link rel="stylesheet" href="{% static 'assets/css/style.css'
%}">
</head>

<body>

<div class="wrapper">
<div class="inner">
<img src="{% static 'assets/images/image-1.png' %}" alt=""
class="image-1">
<form action="{% url 'Registration_save' %}" method="post">
{% csrf_token %}
<h3>New Account?</h3>
<div class="form-holder">
<span class="lnr lnr-user"></span>
<input type="text" class="form-control" placeholder="Username"
name="uname" required>
</div>

<div class="form-holder">
<span class="lnr lnr-envelope"></span>
<input type="email" class="form-control" placeholder="Mail"
name="email" required>
</div>
<div class="form-holder">
<span class="lnr lnr-lock"></span>
<input type="password" class="form-control"
placeholder="Password" name="password" required>
</div>
<div class="form-holder">
<span class="lnr lnr-lock"></span>
<input type="password" class="form-control" placeholder="Confirm
Password" name="cpassword" required>
</div>
<button type="submit">
<span>Register</span>
</button>
<br><br>
```

```
<h1 style="text-align:center;"><a style="color: rgb(23, 107,
79);" href="{% url 'Login_Pg' %}">Login</a></h1>
</form>

<img src="{% static 'assets/images/image-2.png' %}" alt=""
class="image-2">
</div>

</div>

<script src="{% static 'assets/js/jquery-3.3.1.min.js'
%}"></script>
<script src="{% static 'assets/js/main.js' %}"></script>
<script
src="https://unpkg.com/sweetalert/dist/sweetalert.min.js"></script>
{% if messages %}
{% for i in messages %}
{% if i.tags == 'warning' %}
<script>
swal('{{i}}','', 'warning');
</script>
{% elif i.tags == 'error' %}
<script>
swal('{{i}}','', 'error');
</script>
{% else %}
<script>
swal('{{i}}','', 'success');
</script>
{% endif %}

{% endfor %}
{% endif %}
</body><!-- This templates was made by Colorlib
(https://colorlib.com) -->
</html>
```

\\

Registrartion.html

```
<!DOCTYPE html>
{% load static %}
<html>
<head>
<meta charset="utf-8">
```

```
<title>Registration</title>
<meta name="viewport" content="width=device-width,
initial-scale=1.0">

<!-- LINEARICONS -->
<link rel="stylesheet" href="{% static
'assets/fonts/linearicons/style.css' %}">

<!-- STYLE CSS -->
<link rel="stylesheet" href="{% static 'assets/css/style.css'
%}">
</head>

<body>

<div class="wrapper">
<div class="inner">
<img src="{% static 'assets/images/image-1.png' %}" alt=""
class="image-1">
<form action="{% url 'Registration_save' %}" method="post">
{% csrf_token %}
<h3>New Account?</h3>
<div class="form-holder">
<span class="lnr lnr-user"></span>
<input type="text" class="form-control" placeholder="Username"
name="uname" required>
</div>

<div class="form-holder">
<span class="lnr lnr-envelope"></span>
<input type="email" class="form-control" placeholder="Mail"
name="email" required>
</div>
<div class="form-holder">
<span class="lnr lnr-lock"></span>
<input type="password" class="form-control"
placeholder="Password" name="password" required>
</div> <div class="form-holder">
<span class="lnr lnr-lock"></span>
<input type="password" class="form-control" placeholder="Confirm
Password" name="cpassword" required>
</div>
<button type="submit">
<span>Register</span>
</button>
<br><br>
```

```
<h1 style="text-align:center;"><a style="color: rgb(23, 107,
79);" href="{% url 'Login_Pg' %}">Login</a></h1>
</form>

<img src="{% static 'assets/images/image-2.png' %}" alt=""
class="image-2">
</div>

</div>

<script src="{% static 'assets/js/jquery-3.3.1.min.js'
%}"></script>
<script src="{% static 'assets/js/main.js' %}"></script>
<script
src="https://unpkg.com/sweetalert/dist/sweetalert.min.js"></script>
{% if messages %}
{% for i in messages %}
{% if i.tags == 'warning' %}
<script>
swal('{{i}}','', 'warning');
</script>
{% elif i.tags == 'error' %}
<script>
swal('{{i}}','', 'error');
</script>
{% else %}
<script>
swal('{{i}}','', 'success');
</script>
{% endif %}

{% endfor %}
{% endif %}
</body>
</html>
```

apps.py

```
from django.apps import AppConfig


class FrontendConfig(AppConfig):
    default_auto_field = 'django.db.models.BigAutoField'
    name = 'Frontend'
```

models.py

```
from django.db import models

# Create your models here.
class Review(models.Model):
username=models.CharField(max_length=500,null=True,blank=True)
Description=models.CharField(max_length=5000,null=True,blank=True)

class Registration(models.Model):
username=models.CharField(max_length=100,null=True,blank=True
)Email=models.EmailField(max_length=100,null=True,blank=True)
Password=models.CharField(max_length=8,null=True,blank=True)
Confirm_Password=models.CharField(max_length=8,null=True,blank=True)
```

urls.py

```
from django.urls import path
from Frontend import views

urlpatterns=[
    path('home/',views.home,name="home"),
path('detect_weed_or_crop/',views.detect_weed_or_crop,name="detect_weed_or_crop"),

    path('ReviewSave/',views.ReviewSave,name="ReviewSave"),
    path('',views.RegistrationForm,name="RegistrationForm"),

path('Registration_save/',views.Registration_save,name="Registration_save"),
    path('Login_Pg/',views.Login_Pg,name="Login_Pg"),

    path('Login_fun/',views.Login_fun,name="Login_fun"),
    path('Logout_fn/',views.Logout_fn,name="Logout_fn"),
]
```

views.py

```
from django.shortcuts import render,redirect
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import numpy as np
from keras.applications.imagenet_utils import preprocess_input
from django.contrib import messages
from PIL import Image
from Frontend.models import Review,Registration
```

```
from django.http import HttpResponse,JsonResponse
from tempfile import NamedTemporaryFile
import io
from django.shortcuts import redirect,render
from keras.applications.imagenet_utils import preprocess_input
from django.contrib import messages
from Frontend.models import Review,Registration
import tensorflow as tf
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.mobilenet_v2 import
preprocess_input
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.image as mpimg



def home(request):
    x=Review.objects.all()
    return render(request,'Home.html',{'x':x})


def detect_weed_or_crop(request):
    if request.method == "POST":
        try:
            model_path = 'weed_final.h5'
            model = tf.keras.models.load_model(model_path)

            uploaded_file = request.FILES['image_file']
            with NamedTemporaryFile(delete=False) as temp_file:
                temp_file.write(uploaded_file.read())
                temp_file_path = temp_file.name
img = image.load_img(temp_file_path, target_size=(224, 224))
  img_array = image.img_to_array(img)
            img_array = np.expand_dims(img_array, axis=0)
            img_data = preprocess_input(img_array)

            predictions = model.predict(img_data)

            class_labels = ["crop", "weed", "objects"]
            class_probabilities = predictions[0]

class_percentages = [prob * 100 for prob in class_probabilities]
            threshold = 0.5
```

```
detected_classes = [class_labels[i] for i, prob in
enumerate(class_probabilities) if prob > threshold]

            img = mpimg.imread(temp_file_path)
            plt.imshow(img)
            plt.axis('off')

if "crop" in detected_classes and "weed" in detected_classes:
  crop_index = class_labels.index("crop")
                weed_index = class_labels.index("weed")

                crop_percentage = class_percentages[crop_index]
                weed_percentage = class_percentages[weed_index]

                weed_to_crop_ratio = weed_percentage /
        crop_percentage if crop_percentage != 0 else 0

title = f'Weed detected in crop: Crop: {crop_percentage:.2f}
%, Weed: {weed_percentage:.2f}%, Weed to Crop Ratio:
{weed_to_crop_ratio:.2f}'
                title_color = 'purple'
            elif "crop" in detected_classes:
                crop_index = class_labels.index("crop")
                crop_percentage = class_percentages[crop_index]
title = f'Crop detected: {crop_percentage:.2f}
%'              title_color = 'green'
            elif "weed" in detected_classes:
                weed_index = class_labels.index("weed")
                weed_percentage = class_percentages[weed_index]
title = f'Weed detected: {weed_percentage:.2f
}%'                 title_color = 'red'
            else:
                title = 'Neither weed or crop detected'
                title_color = 'black'

context = {'title': title, 'title_color': title_color,
'class_percentages': class_percentages,
                        'detected_classes': detected_classes}
            print(title)
            messages.success(request, title)

            return render(request, "Home.html", context)
        except Exception as e:
            messages.error(request, f"Error: {str(e)}")
            return render(request, "Home.html")
```

```
    messages.error(request, "Invalid Request")
    return render(request, "Home.html")

def ReviewSave(req):
    if req.method=="POST":
        nm=req.POST.get('uname')
        des=req.POST.get('txt')
        x=Review(username=nm,Description=des)
        x.save()
        messages.success(req,"Review Submitted Successfully")
        return redirect(home)

def RegistrationForm(req):
    return render(req,"Registration.html")



def Registration_save(request):
    if request.method == "POST":
        nm = request.POST.get('uname')
        em = request.POST.get('email')
        passw = request.POST.get('password')
        con = request.POST.get('cpassword')
        if passw != con:
messages.error(request, "Password and confirm password do not
match.")
            return redirect(RegistrationForm)
registration = Registration(username=nm, Email=em,
Password=passw, Confirm_Password=con)
        registration.save()
        messages.success(request,"Registered Succesfully")
        return redirect(Login_Pg)

def Login_Pg(req):
    return render(req,"Login_Pg.html")

def Login_fun(request):
    if request.method=="POST":
        nm=request.POST.get('uname')
        pwd=request.POST.get('password')
if
Registration.objects.filter(username=nm,Password=pwd).exists():
            request.session['username']=nm
            request.session['Password']=pwd
            messages.success(request, "Logged in Successfully")
            return redirect(home)
```

```
        else:
            messages.warning(request, "Check Your Credentials")
            return redirect(Login_Pg)
    else:
messages.warning(request, "Check Your Credentials Or Sign Up ")
 return redirect(Login_Pg)

def Logout_fn(request):
    del request.session['username']
    del request.session['Password']
    messages.success(request, "Logged Out Successfully")
    return redirect(Login_Pg)
```

manage.py

```
#!/usr/bin/env python
"""Django's command-line utility for administrative tasks."""
import os
import sys


def main():
    """Run administrative tasks."""
os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'Weed.settings')      try:
from django.core.management import execute_from_command_line
 except ImportError as exc:
        raise ImportError(
"Couldn't import Django. Are you sure it's installed and
""available on your PYTHONPATH environment variable? Did you "
"forget to activate a virtual environment?"
        ) from exc
    execute_from_command_line(sys.argv)


if __name__=='_main_':
    main()
```

urls2.py

```
"""Weed URL Configuration

The 'urlpatterns' list routes URLs to views. For more
information please see:
    https://docs.djangoproject.com/en/3.2/topics/http/urls/
```

```
Examples:
Function views
    1. Add an import: from my_app import views
2. Add a URL to urlpatterns: path('', views.home, name='home')Class-based views
    1. Add an import: from other_app.views import Home
2. Add a URL to urlpatterns: path('', Home.as_view(),
name='home')
Including another URLconf
1. Import the include() function: from django.urls import
include, path
2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
"""
from django.contrib import admin
from django.urls import path,include
from django.contrib.staticfiles.urls import
staticfiles_urlpatterns,static
from . import settings
import Frontend.urls
from Frontend import views
urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include(Frontend.urls)),
    path('/',views.RegistrationForm,name="RegistrationForm")
]

urlpatterns+=staticfiles_urlpatterns()
urlpatterns+=static(settings.MEDIA_URL,document_root=settings.MEDIA_ROOT)
```