FPAY AS YOU GO

CSE3001 - SOFTWARE ENGINEERING
J-COMPONENT PROJECT REPORT
Winter 2022-2023

Submitted by

Sanjana Rajeshwar *(20BCE0782)*
Shreya Thaplyal *(20BCE0814)*
Arnav Srivastava *(20BCE0831)*
Danish Ansari (20BCE0932)

Under the Guidance of
Dr. Sahaaya Arul Mary S. A.
(Professor)

<<Computer Science and Engineering >>

Vellore-632014, Tamil Nadu, India

School of Computer Science and Engineering

TABLE OF CONTENTS

CHAPTER – 1: INTRODUCTION

Shuttle services are used by VITians for traversing between academic buildings and hostel buildings. Students often face difficulties in using UPI or cash transactions to make payment for shuttles and often get off without paying when they are in a hurry. Students often get off the bus without paying when they are in a hurry to reach their classes since it is a hustle to make payment using cash for an amount as small as 15 rupees or wait for the mobile data to work to make use of UPI.

There is no way for the shuttle drivers to keep a track of whether the student had paid before leaving. Drivers also struggle to keep an account of all the students that took ride in the shuttle manually. This job can be very hectic and has huge margins for errors.

To prevent such loss for the shuttle services and ease the payment process for passengers, we propose a simple and efficient facial recognition-based payment system which will capture the pictures of boarders as they enter the shuttle and reflect the payment due on their VTOP login. We will be integrating the facial recognition feature with VTOP payments page. It makes the commute convenient for the shuttle boarders, as they don't have to wait in queues to make the payment.

Facial recognition technology can also be used for bus rides tracking. It works by installing cameras on buses that capture an image of each passenger's face as they board the bus. The technology then compares the faces to a database of known faces and records who is present on the bus.

Using facial recognition for bus rides tracking can have several benefits. It can reduce the risk of fraud or errors, as it is more difficult to fake or duplicate a face than a paper ticket or ID card. It can also help to streamline the boarding process, as passengers do not need to present a ticket or ID. However, as with any use of facial recognition technology, there are also concerns around privacy and potential biases in the algorithms used. Some people may be uncomfortable with having their face scanned and stored in a database, and there is a risk that the technology could be used for surveillance purposes.

Therefore, if facial recognition technology is used for bus rides tracking, it is important to ensure that appropriate safeguards are in place to protect privacy and prevent misuse. This may include obtaining consent from passengers, limiting the use of the technology to specific purposes, and ensuring that the algorithms used are fair and unbiased. It is also important to consider the potential impact on vulnerable or marginalized communities, as they may be more likely to face discrimination or harassment if the technology is not used fairly.

Literature Survey:

[1] High Performance Facial Expression Recognition System Using Facial Region Segmentation, Fusion of HOG & LBP Features and Multiclass SVM

The High-Performance Facial Expression Recognition System is a computer vision system that uses facial region segmentation, feature extraction, feature fusion, and classification to recognize facial expressions in real-time. It employs HOG and LBP feature extraction methods and a multiclass SVM classifier to achieve high accuracy in recognizing facial expressions. This system has potential applications in various fields, including human-computer interaction, emotion recognition, and security.

Techniques Discussed:

The High-Performance Facial Expression Recognition System employs several techniques to recognize facial expressions in real-time. These include:

- **Facial region segmentation:** This technique involves detecting and segmenting the face region from the rest of the image using a Viola-Jones object detection algorithm and the Dlib library for facial landmarks detection.
- **Feature extraction:** The system uses two different feature extraction methods, Histogram of Oriented Gradients (HOG) and Local Binary Patterns (LBP), to extract spatial and texture information of the facial region.
- **Feature fusion:** The HOG and LBP features are fused together to create a more robust feature vector that captures both the spatial and texture information of the facial region.
- **Classification:** The system employs a multiclass Support Vector Machine (SVM) classifier to classify the facial expression based on the fused feature vector.

These techniques have been shown to achieve high accuracy in recognizing facial expressions, even in challenging lighting conditions and with partial occlusion of the face.

Advantages:

The High-Performance Facial Expression Recognition System has several advantages, including:

- **Real-time performance:** The system can recognize facial expressions in real-time, making it suitable for applications that require quick and efficient processing.
- **High accuracy:** The system achieves high accuracy in recognizing facial expressions, even in challenging lighting conditions and with partial occlusion of the face.
- **Robust feature extraction:** The system employs both HOG and LBP feature extraction methods to capture both spatial and texture information of the facial region, which improves the robustness of the feature vector.
- **Multiclass classification:** The system uses a multiclass SVM classifier to classify the facial expression, which allows it to recognize multiple facial expressions at the same time.
- **Potential for various applications:** The system has potential applications in various fields, such as human-computer interaction, emotion recognition, and security.

Overall, the High-Performance Facial Expression Recognition System is a reliable and efficient system for recognizing facial expressions, which has the potential to enhance the performance of various applications that require this capability.

Disadvantages:

The High-Performance Facial Expression Recognition System has some potential disadvantages, including:

- **Privacy concerns:** The use of facial recognition technology can raise privacy concerns, as the system captures and stores images of people's faces, which could potentially be misused or accessed without consent.

- **Dependence on facial landmarks:** The system relies on accurate detection of facial landmarks by the Dlib library, which may not always be reliable in certain conditions, such as low-quality images or images with partial occlusion of the face.
- **Sensitivity to image quality:** The accuracy of the system may be affected by image quality, such as low resolution or low lighting conditions, which could lead to misclassification of facial expressions.
- **Potential for biases:** The accuracy of the system may be affected by biases in the training data used to develop the system, which could lead to inaccurate classification of facial expressions for certain groups of people.
- **Cost and implementation:** The system requires specialized hardware and software, which could make it expensive to implement in certain applications.

Overall, these potential disadvantages must be considered when implementing facial expression recognition systems, and appropriate measures must be taken to address them

[2] An Efficient Convolutional Neural Network Approach for Facial Recognition

An Efficient Convolutional Neural Network Approach for Facial Recognition is a computer vision system that uses deep learning techniques to recognize faces. It employs a convolutional neural network (CNN) that consists of several layers of convolutional and pooling operations, followed by fully connected layers that perform the classification.

The system is trained on a large dataset of labeled faces using the backpropagation algorithm, and it achieves high accuracy in recognizing faces, even in challenging lighting conditions and with partial occlusion of the face.

The system's advantages include real-time performance, high accuracy, and robustness to various lighting and occlusion conditions. However, it may also have potential disadvantages such as high computational requirements and sensitivity to variations in facial features due to pose and expression changes.

Overall, the Efficient Convolutional Neural Network Approach for Facial Recognition is a powerful and promising system for facial recognition, with potential applications in various fields, such as security, surveillance, and biometric authentication.

Techniques Discussed:

The Efficient Convolutional Neural Network Approach for Facial Recognition employs several techniques to recognize faces, including:

- **Convolutional neural networks:** The system uses a deep learning architecture that consists of several layers of convolutional and pooling operations, followed by fully connected layers that perform the classification.
- **Data augmentation:** The system uses data augmentation techniques, such as flipping and rotating images, to increase the size of the training dataset and improve the system's ability to recognize faces with different poses and expressions.
- **Transfer learning:** The system utilizes a pre-trained model as the starting point for training, which allows it to leverage the knowledge gained from training on large datasets such as ImageNet to improve the accuracy of facial recognition.
- **Loss functions:** The system employs loss functions such as softmax cross-entropy and L2 regularization to minimize the difference between predicted and actual face labels and prevent overfitting.

These techniques allow the system to learn features that are robust to variations in facial pose, expression, and lighting conditions, which enables it to achieve high accuracy in facial recognition tasks.

Advantages:

The Efficient Convolutional Neural Network Approach for Facial Recognition has several advantages, including:

- **High accuracy:** The system achieves high accuracy in recognizing faces, even in challenging conditions, such as partial occlusion and variations in pose and expression.
- **Robust feature extraction:** The system uses deep learning techniques to learn features that are robust to variations in facial pose, expression, and lighting conditions, which improves the accuracy of facial recognition.
- **Real-time performance:** The system is optimized for real-time performance, making it suitable for applications that require quick and efficient processing, such as surveillance and security systems.
- **Scalability:** The system is scalable, and its performance can be improved by adding more layers to the network or by training on larger datasets.
- **Transfer learning**: The system uses transfer learning, which allows it to leverage pre-trained models and knowledge gained from large datasets to improve the accuracy of facial recognition.

Overall, the Efficient Convolutional Neural Network Approach for Facial Recognition is a reliable and efficient system for recognizing faces, which has the potential to enhance the performance of various applications that require this capability.

Disadvantages:

The Efficient Convolutional Neural Network Approach for Facial Recognition may have some potential disadvantages, including:

- **High computational requirements**: The system requires high computational resources, such as GPUs, to train and run the convolutional neural network, which can be expensive.
- **Dependence on the quality of training data**: The accuracy of the system may be affected by the quality of the training data, such as the number and diversity of faces in the dataset, which may limit the generalizability of the system.
- **Sensitivity to lighting conditions**: The system may be sensitive to lighting conditions, and its accuracy may decrease in low-light conditions or when the face is backlit.
- **Limited ability to recognize disguised faces**: The system may have limited ability to recognize disguised faces, such as those wearing masks or glasses, which can reduce its effectiveness in certain applications.
- Potential for bias: The accuracy of the system may be affected by biases in the training data or the algorithm itself, which can lead to inaccurate recognition of certain groups of people.

Overall, these potential disadvantages must be considered when implementing the Efficient Convolutional Neural Network Approach for Facial Recognition, and appropriate measures must be taken to address them.

[3] Facial Recognition using the OpenCV Libraries of Python for the Pictures of Human Faces Wearing Face Masks during the COVID-19 Pandemic

Facial Recognition using the OpenCV Libraries of Python for the Pictures of Human Faces Wearing Face Masks during the COVID-19 Pandemic is a computer vision system that utilizes OpenCV libraries to recognize faces in images that are wearing face masks.

The system employs several techniques, including face detection using Haar cascades, feature extraction using Local Binary Patterns (LBP), and classification using Support Vector Machines (SVM).

The system is trained on a dataset of labeled face images with and without masks, and it achieves high accuracy in recognizing faces wearing masks, even with variations in the color and texture of the masks.

The system's advantages include real-time performance, high accuracy, and low computational requirements. However, it may also have potential disadvantages such as reduced accuracy with partially occluded faces, and reduced effectiveness with face masks that are very different from those in the training dataset.

Overall, the Facial Recognition using the OpenCV Libraries of Python for the Pictures of Human Faces Wearing Face Masks during the COVID-19 Pandemic is a promising system for recognizing faces wearing masks, which has the potential to enhance the performance of various applications that require this capability during the pandemic.

Techniques Discussed:

The Facial Recognition using the OpenCV Libraries of Python for the Pictures of Human Faces Wearing Face Masks during the COVID-19 Pandemic system employs several techniques to recognize faces wearing masks, including:

- **Haar cascades for face detection**: The system uses Haar cascades to detect faces in images. Haar cascades are a popular technique for object detection that use a series of classifiers to detect specific features in an image.
- **Local Binary Patterns (LBP) for feature extraction**: The system uses LBP to extract features from the face regions of the detected images. LBP is a simple yet effective method for texture analysis that encodes the local structure of an image.
- **Support Vector Machines (SVM) for classification**: The system uses SVMs to classify the extracted features as either a masked or unmasked face. SVMs are a widely used machine learning algorithm that can learn to classify data into different classes based on a set of training examples.
- **Data augmentation**: The system uses data augmentation techniques such as flipping and rotating images to increase the size of the training dataset and improve the system's ability to recognize faces with different poses and expressions.

These techniques allow the system to learn features that are robust to variations in facial pose, expression, and mask texture, which enables it to achieve high accuracy in recognizing faces wearing masks.

Advantages:

The Facial Recognition using the OpenCV Libraries of Python for the Pictures of Human Faces Wearing Face Masks during the COVID-19 Pandemic system has several advantages, including:

- **Real-time performance:** The system is capable of recognizing faces wearing masks in real-time, making it suitable for various applications that require fast and accurate face recognition.
- **High accuracy:** The system achieves high accuracy in recognizing faces wearing masks, even with variations in the color and texture of the masks.
- **Low computational requirements**: The system can run on low-end hardware such as Raspberry Pi, making it suitable for resource-constrained environments.
- **Flexibility**: The system can be easily adapted to different face recognition scenarios by changing the training dataset and tuning the system's parameters.
- **Data augmentation**: The system employs data augmentation techniques that increase the size and diversity of the training dataset, which improves its ability to recognize faces with different poses, expressions, and lighting conditions.

Overall, the Facial Recognition using the OpenCV Libraries of Python for the Pictures of Human Faces Wearing Face Masks during the COVID-19 Pandemic system is a promising approach for recognizing faces wearing masks, with several advantages that make it suitable for a wide range of applications.

Disadvantages:

The Facial Recognition using the OpenCV Libraries of Python for the Pictures of Human Faces Wearing Face Masks during the COVID-19 Pandemic system may also have some potential disadvantages, including:

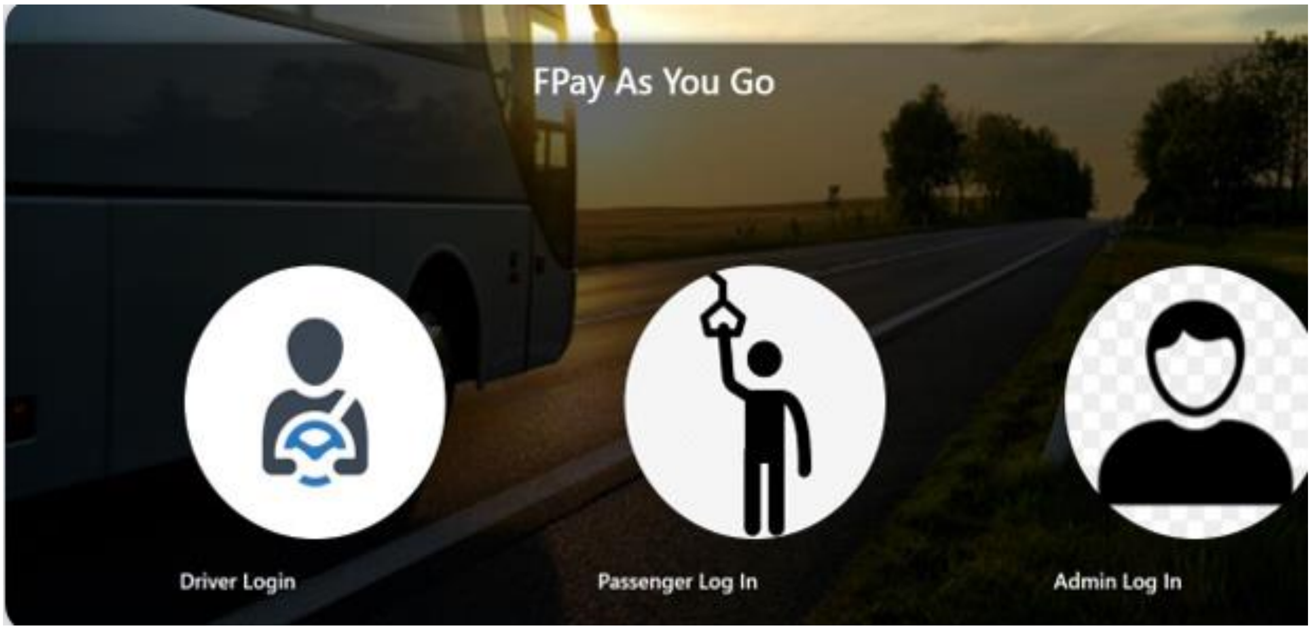
- **Reduced accuracy with partially occluded faces:** The system may have reduced accuracy in recognizing faces that are partially occluded by the mask or other objects.
- **Limited effectiveness with very different masks:** The system may have limited effectiveness in recognizing faces wearing masks that are very different from those in the training dataset.
- **Vulnerability to adversarial attacks**: The system may be vulnerable to adversarial attacks where small modifications to the input image can cause misclassification.
- **Privacy concerns:** The use of facial recognition technology raises privacy concerns, as it can be used to track and monitor individuals without their consent.
- **Bias:** The system may exhibit bias against certain groups of people, depending on the composition of the training dataset and the choice of parameters.

It is important to consider these potential disadvantages and address them appropriately when deploying facial recognition systems in real-world applications.

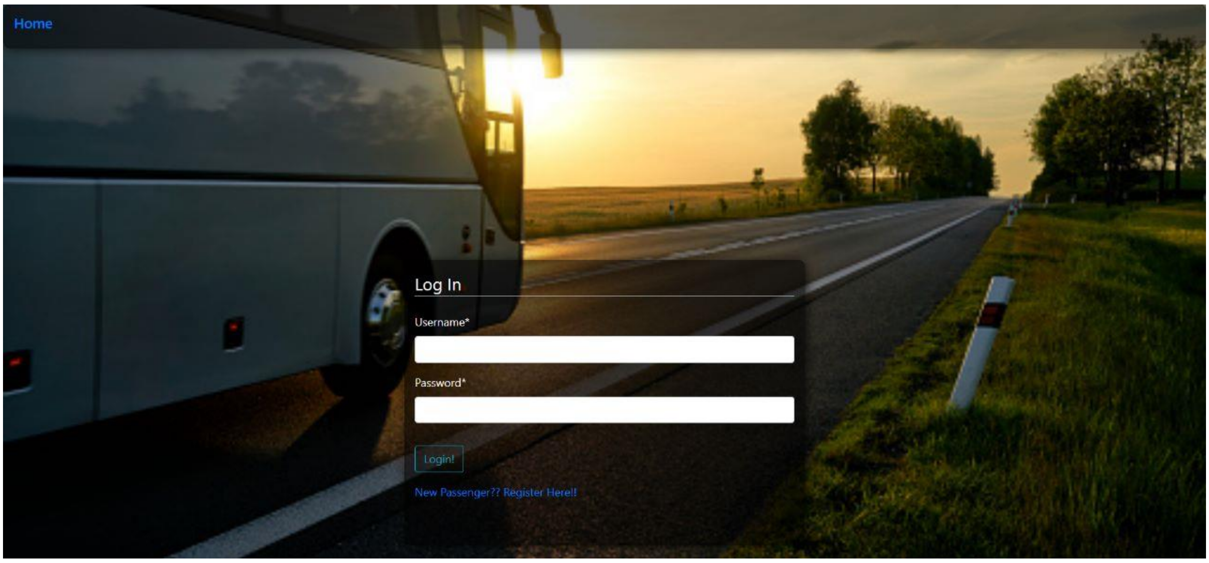
Chapter 2: Login/Registration Module

Process:

- → User Login process is decomposed into Register and Sign-In processes.
- → Driver and Student input valid credentials, the login process matches those credentials against the one stored in Verified User datastore. If it matches, then the user is granted access
- → 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.
- → Our project has separate logins for 3 different kinds of users based on their roles:
  - o Driver
  - o Passenger
  - o Admin



- → The shuttle Driver clicks on Driver's Login and enter his account's username and password
- → The credentials are validated through the database and the Driver is logged into his account after which he can start the system
- → If the credentials entered by the driver are incorrect an error message "Invalid Credentials" is displayed, and the driver is asked to fill the credentials again.
- → Passengers/students can access their accounts by clicking on Passenger's login and entering their authentic credentials.
- → Admins can also login by entering their credentials and then view records on their account


Home

Log In

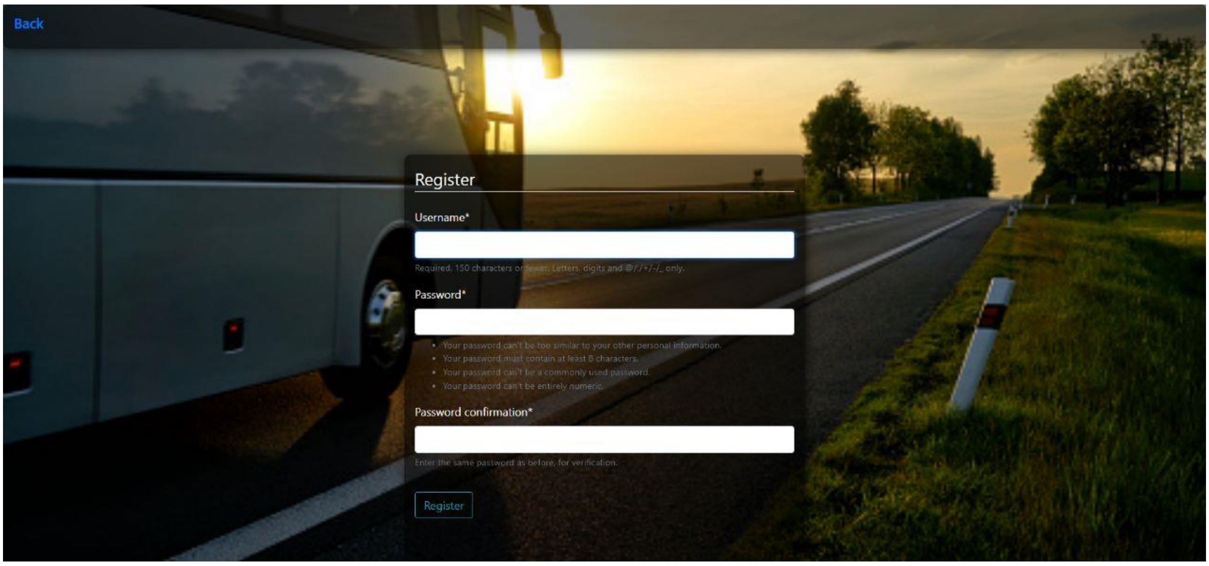
Username*

Password*

Login!

New Passenger?? Register Here!!

→ If it's a new passenger, he/she can register by entering Username and password. Type the password again for confirmation. Password should follow the given criteria for it to be accepted. After registration the passengers have to use the "upload photos" feature on their first login.


Back

Register

Username*

Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

Password*

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation*

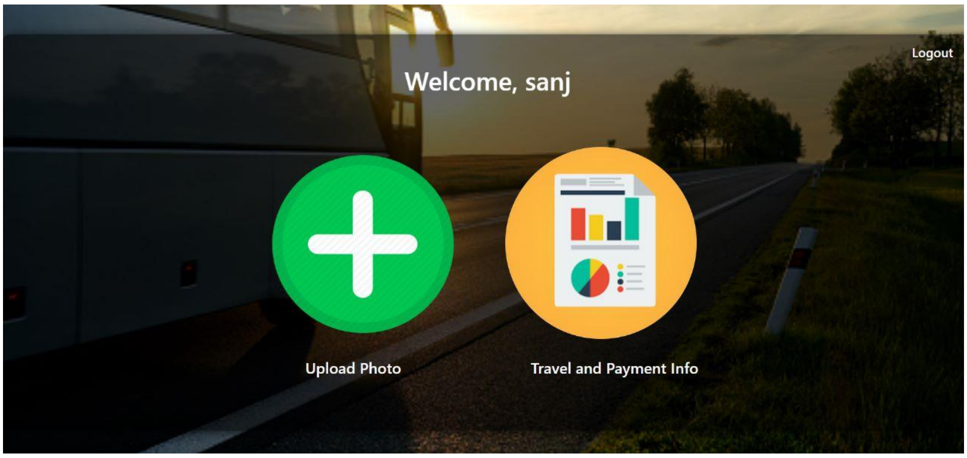
Enter the same password as before, for verification.

Register

Chapter 3: Facial Recognition System

Process:

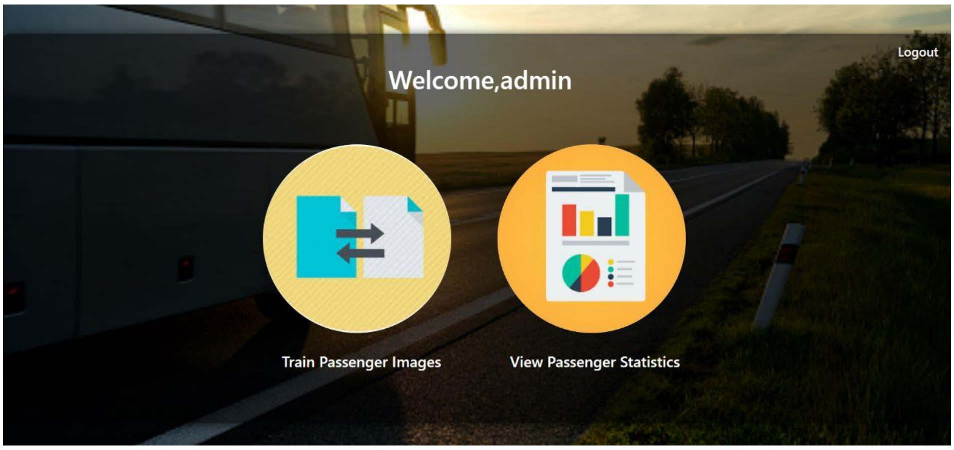
- Passengers will have to use the "Upload Photos" feature on their first login. The system webcam will open and take photos of the passenger.
- Admin will have to use the "Train Images" feature to train the model using the images of the passenger.
- The model used is SVM + HOG (Support Vector Machine + Histogram Oriented Gradients.
- The driver will login to start the facial recognition during shuttle on-boarding.
- The passengers will be recognized, and the ride fare will be reflected in their logins.



WHY SVM + HOG OVER CNN?

We have chosen SVM+HOG as our method of facial recognition due to the following reasons:

- Images of Passengers are trained faster with SVM+HOG method.
- In our application, passengers must be recognized quickly during the shuttle onboard so the computation time should be less. The computation time of SVM+HOG is lesser.
- SVM+HOG can be more robust to variations in lighting and pose than CNN, as the HOG features are based on local gradients and are therefore less sensitive to global changes in illumination or viewpoint.



ALGORITHM/ STEPS FOR IMAGE PROCESSING AND SVM + HOG

1. Initialize the video stream (for Upload Image)
2. Capture images from the video stream.
3. Load HOG face detector and the shape predictor for alignment.
4. Perform Image Processing using OpenCV and face detection using HOG
   a. Re-size the images.
   b. Convert the images to grayscale (for better model efficiency)
   c. Find the coordinates of the image and return a frame around the detected face.
   d. Extract the HOG (Histogram of Oriented Gradients) features from the images.
   e. The HOG features are calculated by dividing each image into small cells and calculating the gradient directions and magnitudes of the pixels within each cell.
5. Store the images along with the passenger identity.
6. When admin clicks on train images, SVM (Support Vector Machine) classifier is trained on the HOG features extracted from the images. The SVM is a supervised learning classifier that learns to separate the positive and negative examples based on the HOG features. The radial basis function kernel (RBF kernel) SVM are used for training and tracking.
7. When the driver starts facial recognition, steps 1, 2, 3 and 4 are repeated.
8. Passengers are identified using the trained SVM classifier and HOG features extracted from the uploaded images.

Chapter 4: Payment Gateway

Why RazorPay?

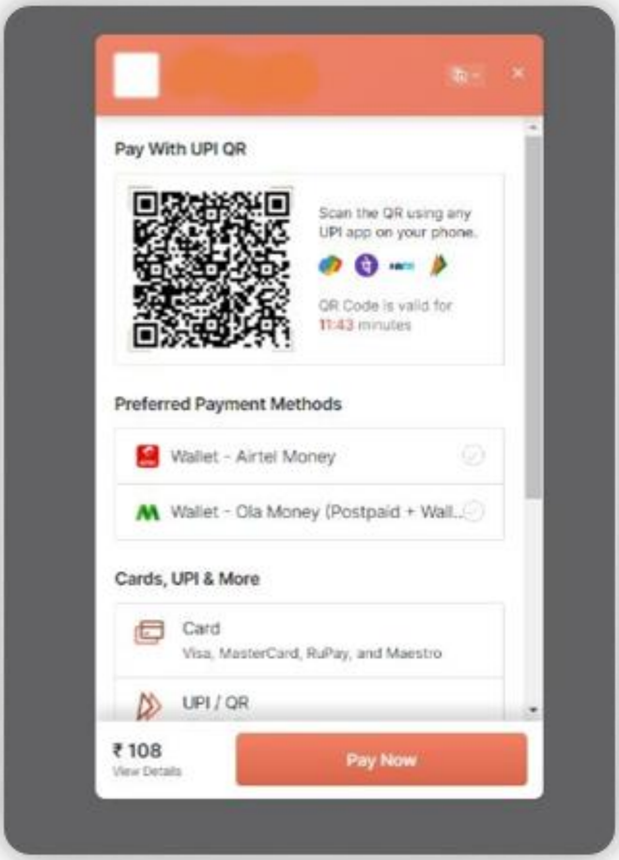
Razorpay is one of the most popular payment gateway solutions in India, providing several advantages over other payment gateways. Here are some of the advantages of Razorpay:

- **Easy Integration:** Razorpay provides a simple and easy-to-use API for integrating its payment gateway into your website or mobile app. This makes it easier for developers to integrate payment functionality into their applications quickly.
- **Multiple Payment Options:** Razorpay supports multiple payment options, including credit cards, debit cards, net banking, UPI, and wallets. This enables your customers to make payments using their preferred payment method, increasing the chances of successful transactions.
- **Faster Settlement:** Razorpay offers faster settlement of funds to your bank account compared to other payment gateways. This means that you can receive your funds quickly, helping you manage your cash flow more efficiently.
- **Competitive Pricing:** Razorpay offers competitive pricing compared to other payment gateways, with no setup or maintenance fees. You only pay a transaction fee for successful transactions.
- **Excellent Support:** Razorpay provides excellent customer support through phone, email, and chat. Their support team is available 24/7 to help you resolve any issues you may encounter while using the payment gateway.
- **Advanced Security**: Razorpay uses the latest security standards to ensure the safety of your transactions and customer data. They are PCI-DSS compliant, which means that they meet the highest security standards in the payment industry.

Overall, Razorpay offers several advantages over other payment gateways, making it a popular choice for businesses in India.

Process:

- The passenger can make use of this integrated gateway to pay their dues.
- In our project, we have integrated the RazorPay payment gateway.
- The passenger is firstly supposed to enter their name, this is done so that the payment is recorded under the passengers' full name, instead of the username (if they have used a different/short form for their username).
- Then they would be directed to the payment interface.
- There they can select among multiple modes of payments like UPI, Credit/debit cards, wallets, etc.



Scan the QR using any UPI app on your phone.

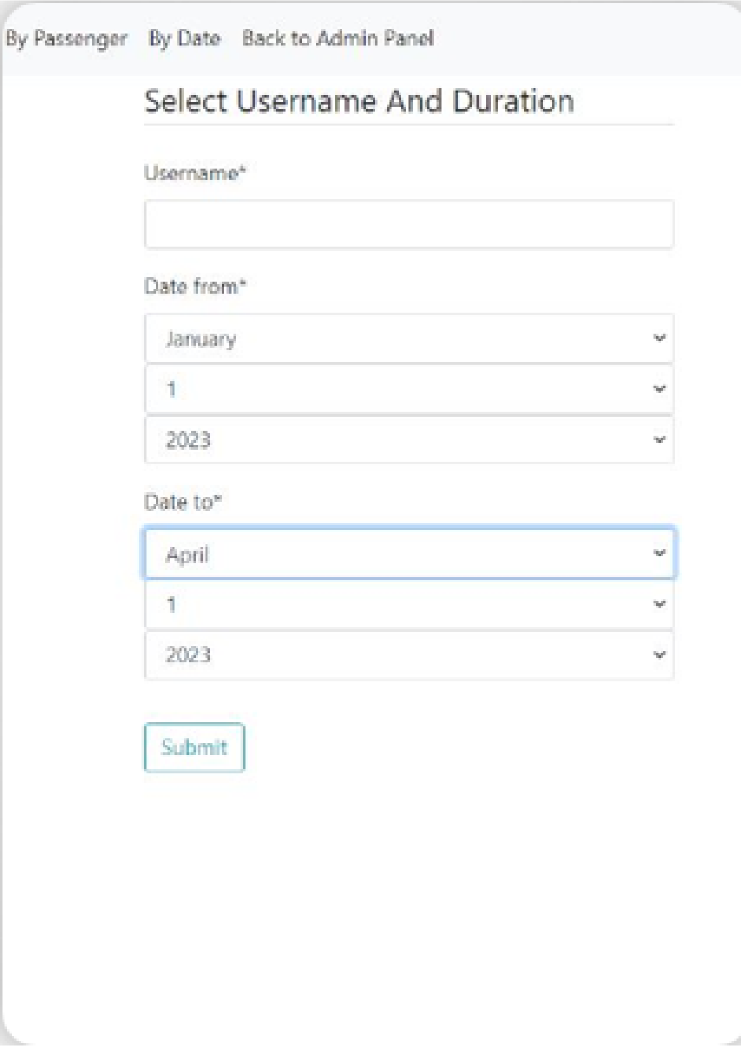
QR Code is valid for 11:43 minutes

Preferred Payment Methods
Wallet - Airtel Money
Wallet - Ola Money (Postpaid + Wall...
Cards, UPI & More
Card
Visa, MasterCard, RuPay, and Maestro
UPI / QR
₹ 108
View Details
Pay Now

- The amount to be paid will be pre-loaded from the database, as per their travel history and corresponding payment status.
- The required details of the payment recipient are pre-fed.
- After selecting the mode of payment (and providing required details if needed) the payment will be processed
- If the payment is successful, the user will be redirected to 'success' page, else will be notified of failure.

Chapter 5: Payment/Travel Records

- → Authenticate
  - ○ Authenticate the admin user by verifying their credentials.
- → Retrieve
  - ○ Retrieve the list of payment records from the database.
- → Display
  - ○ Display the passenger trip records in a table format, showing relevant information such as boarding date, boarding time, amount due, payment date, passenger name, and other trip details.
- → Provide
  - ○ Provide filters or sorting options to allow the admin user to narrow down the payment records based on specific criteria, such as boarding date range, passenger, or trip details.
- → Allow
  - ○ Allow the admin user to search for a specific payment record by entering relevant keywords or information, such as payment ID, passenger name, or trip details.



Select Username And Duration

Username*

Date from*

January

1

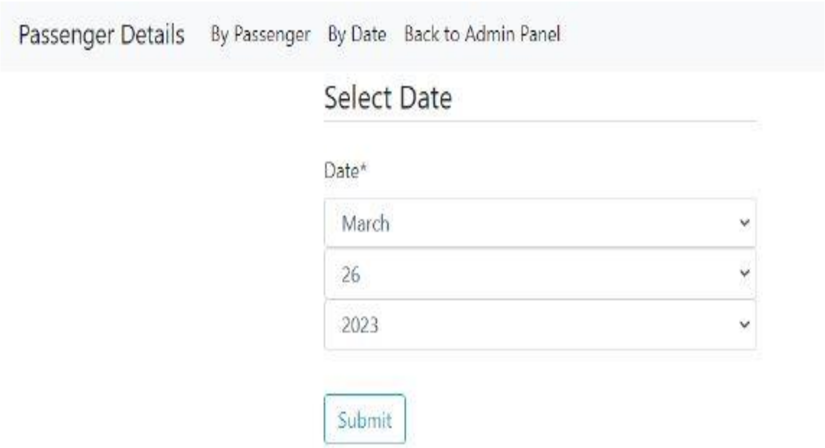
2023

Date to*

April

1

2023



Select Date

Date*

March

26

2023

CHAPTER-6

Software Requirements Specification

for

FPay As You Go

Version 1.0 approved

Prepared by

Sanjana Rajeshwar 20BCE0782

Shreya Thaplyal 20BCE0814

Arnav Srivastava 20BCE0831

Danish Ansari 20BCE0932

Vellore Institute of Technology, Vellore

05/02/2023

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

Review 1	08/02/2022	Initial Commits	1.0

1. Introduction

1.1 Purpose

Our concept presents a simple and efficient facial recognition-based payment approach to ensure that all shuttle boarders pay, and that the shuttle services do not suffer any losses

1.2 Document Conventions

The document uses the following conventions:

- Entire document should be justified.
- Convention for Main title
  - Font face: Times New Roman
  - Font style: Bold
  - Font Size: 18
- Convention for Subtitle
  - Font face: Times New Roman
  - Font style: Bold
  - Font Size: 14
- Convention for body
  - Font face: Arial
  - Font style: Italics
  - Font Size: 12

1.3 Intended Audience and Reading Suggestions

This project is a prototype for the college students (VITians); however, it is not restricted within the college premises. This has been implemented under the guidance of college professors. This project is useful for the shuttle service, students, and anyone else who utilizes the services.

1.4 Product Scope

Students often get off the shuttles without paying when they are in a hurry to reach their classes since it is a hustle to make payments using cash for an amount as small as 15 rupees or wait for the mobile data to work to make use of UPI.

There is no way for the shuttle drivers to keep a track of whether the student had paid before leaving to prevent such loss for the shuttle services, we propose a facial recognition-based payment system that will capture the pictures of boarders as they enter the shuttle and keep a detailed account of the payments due.

To ensure that everyone using the shuttle pays and that there are no losses for the shuttle services, our solution provides a simple and reliable facial recognition-based payment method.

1.5 References

- *R.S Pressman, Software Engineering: A Practitioner's Approach, Mc-Graw-Hill, Edition-7e (2010).*
- *L. Li, X. Mu, S. Li and H. Peng, "A Review of Face Recognition Technology," in IEEE Access, vol. 8, pp. 139110-139120, 2020, doi: 10.1109/ACCESS.2020.3011028.*
- *M. Arsenovic, S. Sladojevic, A. Anderla, and D. Stefanovic, "FaceTime - Deep learning based face recognition rides system," SISY 2017 - IEEE 15th Int. Symp. Intell. Syst. Informatics, Proc., pp. 53–57, 2017.*
- *S.Milan, H.Vaclav, and B.Roger, Image Processing, Analysis, and Machine Vision, 4th ed. Cengage Learning, 2014*

2. Overall Description

2.1 Product Perspective

FPay As You Go is a replacement for an existing system.

The current system involves payments for shuttle services through cash or UPI.

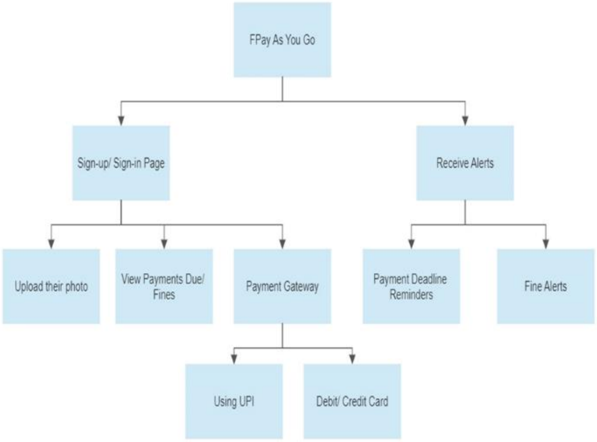
Using cash for an amount as small as 15 rupees or waiting for the mobile data to work to make use of UPI, are both time consuming and inefficient processes.

These are also not a very secure method as they do not insure that every passenger has paid for their ride.

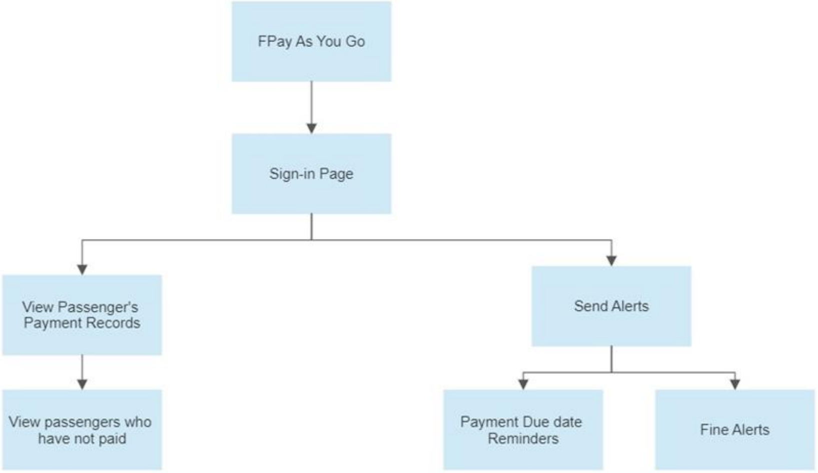
So, this project offers an easy and effective facial recognition-based payment method to make sure that everyone using the shuttle pays and that there are no losses for the shuttle services.

User-Journey (Web-App Flow Diagram):

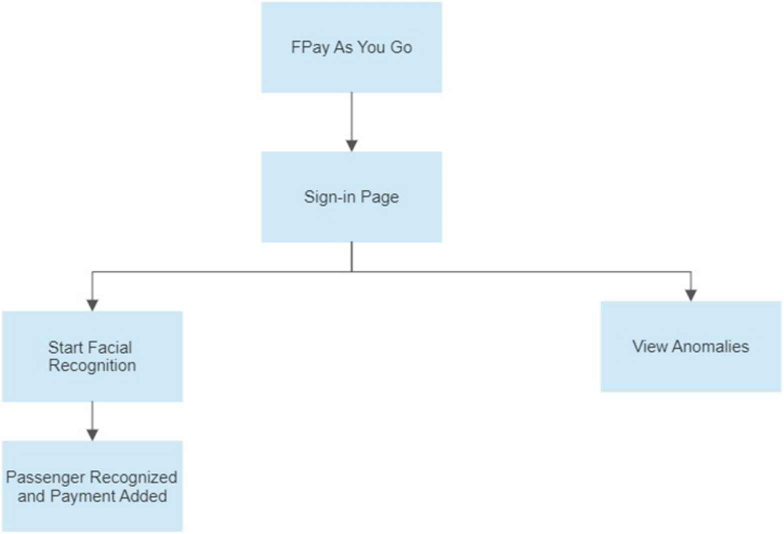
For Passenger:



For Admin


FPay As You Go
Sign-in Page
View Passenger's Payment Records
View passengers who have not paid
Send Alerts
Payment Due date Reminders
Fine Alerts

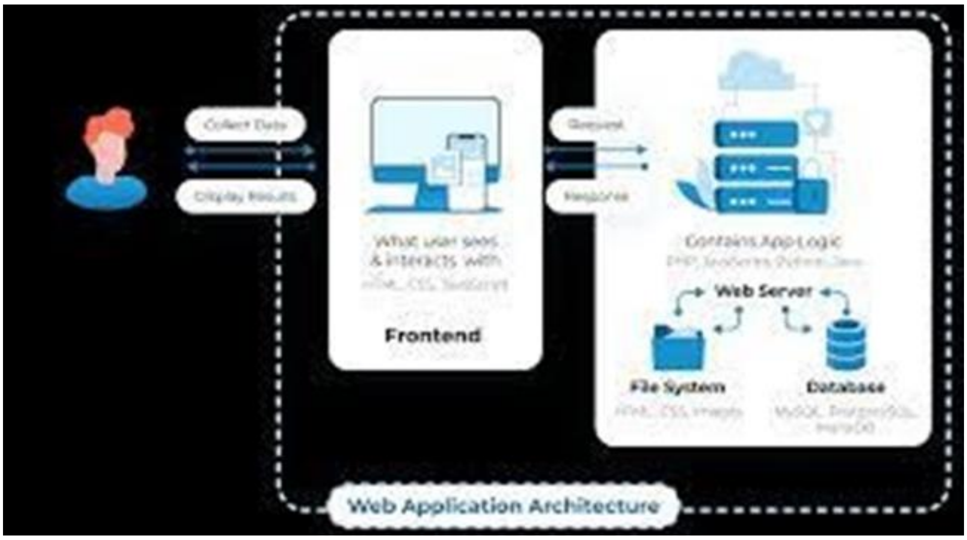
For Driver



2.2 Product Functions

1. Login
   a. Sign-up
   b. Sign-in
   c. Forgot Password
2. Facial Recognition
   a. Passengers Upload Image which are stored in Database.
   b. Driver Starts Recognition Process
   c. SVM + HOG Model recognizes Passenger.
   d. Payment is added to passenger Login.
3. Payment
   a. Passengers can view Payment Due
   b. Make Payment using Payment Gateway
   c. Students receive due/fine alerts.

4. Report Generation
    a. Admin Views Payment Records



2.3 User Classes and Characteristics

User Classes of our project are differentiated on the basis of subset of product functions used.

The classes are as follows:

1. Passenger
    a. Upload Image
    b. View Payment Due
    c. Pay Using Payment Gateway
2. Driver
    a. Start Facial Recognition
    b. View Anomalies
3. Admin
    a. View Passenger Payment Records
    b. Train Images

2.4 Operating Environment

H/W Platform

Core i3, i5, i7 processor
2GB Ram.
2GB of hard disk space in terminal machines
256GB hard disk space in Server Machine
a cost-efficient high-quality camera like Wyze Cam v3
Wired Security Camera and an inexpensive display device.

S/W Requirement

Windows 7 or above operating system, macOS
IDEs (like VS Code)
SQLite
Sklearn, OpenCV library

2.5 Design and Implementation Constraints

Synchronization: uses USB 2.0, connects only to Windows 7 and above, macOS
Memory: device will have 2GB internal hard drive. Device will have a SD card slot, and the software must be able to read and write to that slot.
Technology Tools: VS Code, SQLite, Bootstrap Studio.
Language Requirements: Python – OpenCV & SklearnD HTML, CSS, JavaScript, SQL.

2.6 User Documentation

Guidelines for Passengers

- Upon visiting the portal, the passengers can login using their valid credentials
- If they forgot their password, make use of the forgot password option
- If not registered, the user can register using the sign-up option
- Then they are supposed to upload their own image, which will be used to train the model.
- While taking the image, ensure that there is proper lighting and that the face is clearly visible
- Using the view payments option, the user can view the due amount
- Then making use of the payment gateway, they can pay the total amount
- They can pay through credit/debit card or UPI.
- The amount should be paid within the set due date, or else they would be liable for fine/s
- Make sure to provide the correct email, in order to receive the reminders and alerts

Guidelines for Drivers

- Make use of the assigned credentials to access the system
- Start the recognition process on time, i.e. before the passengers start boarding
- During the boarding process, keep a track of anomalies and ask them to pay beforehand
- Don’t leave the system running unattended

Guidelines for Admin

- Keep a track of the payment records
- Send alerts/reminders on time
- Ensure there are no inconsistencies in the database

2.7 Assumptions and Dependencies

- Each user must have valid credentials.
- Server must be running for the system to function.
- Users must log in to the system to access any feature.
- Passenger is uploading their own photo.

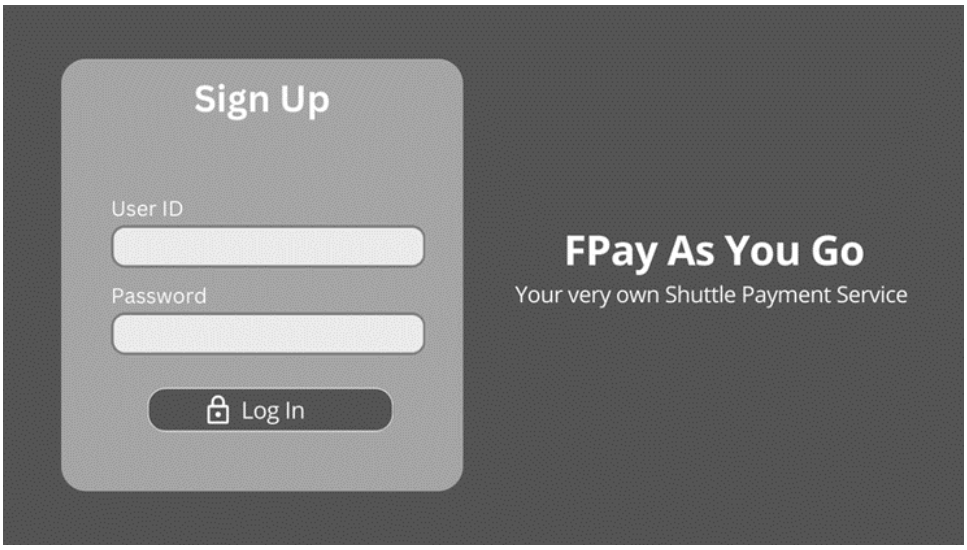
- Facial Recognition System is quick and accurate.
- Payment Gateway Services is always active.

3. External Interface Requirements

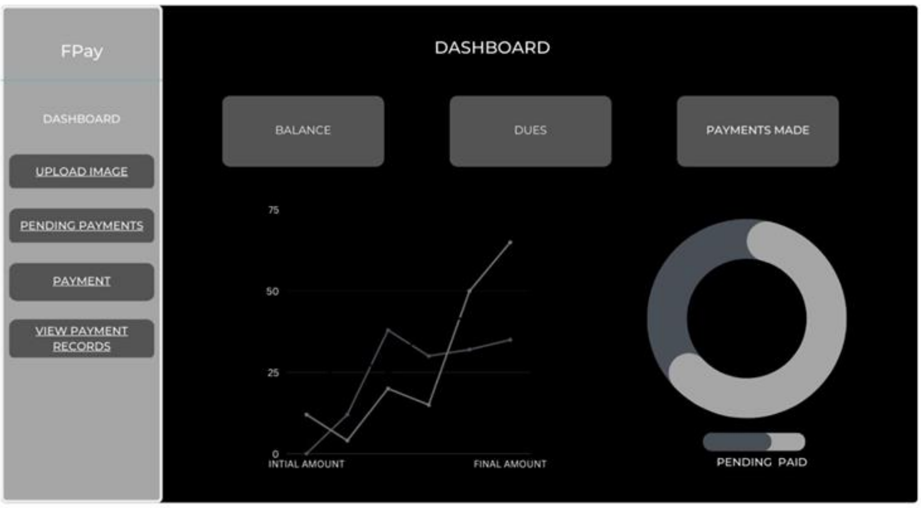
3.1 User Interfaces

A few sample snippets of our software's UI are as follows:

Login Page:



Passenger Dashboard Interface:



3.2 Hardware Interfaces

Laptop/Desktop PC/Mobile/Tablets-To access the website and its features.

3.3 Software Interfaces

Following are the software interfaces:

- Databases are managed using MongoDB Compass.

- Operating System like Windows 7 or higher, MacOS, Linux, etc.
- Tools like Visual Studio Code to implement python code.
- Libraries like Pandas, Numpy, DLib, OpenCV, etc.
- Integrated Commercial component is payment gateway (example – RazorPay).

Data items or messages are as follows:

- Images: Take as input to train and test the facial recognition model.
- Passenger Identity: Output from the facial recognition system.
- Payment: Shuttle ride payments paid by the passenger.
- Alerts/ Reminders: Received by the passengers.

3.4 Communications Interfaces

NIC (Network Interface Card) – It is a computer hardware component that allows a computer to connect to a network. NICs may be used for both wired and wireless connections.

TCP/IP protocol- Internet service provider to access and share information over the Internet

Ethernet Communications Interface- Ethernet is a frame-based computer network technology for local area networks (LANs)

4. System Features

4.1 Login

4.1.1 Description and Priority

Description: A user-friendly platform for users (passengers, drivers and admin) to sign up/ sign in. Authentication and forgot password features are also provided.
Priority: High
Rating
Risk: 4
Cost: 2
Benefit: 6

4.1.2 Stimulus/Response Sequences

Visit portal ->visit login page -> sign up (for new users) -> login (for existing users)

4.1.3 Functional Requirements

REQ-1: User should input valid credentials.

REQ-2: Field Validations

Anticipated Errors:

Forgot Password Feature

4.2 Facial Recognition

4.2.1 Description and Priority

Description: It is decomposed it Facial Recognition model training, Process Input Picture and Passenger Identification. Students will login and upload their images to train the facial recognition model. The driver can start the facial recognition system when passengers are boarding i.e., the camera will take pictures of passengers and facial recognition system will identify the passenger using Passenger Details datastore and the payment will be reflected in the identified passenger's login account.
Priority: High
Rating
Risk: 8
Cost: 7
Benefit: 8

4.2.2 Stimulus/Response Sequences

For Student

Visit portal -> login to their account ->Upload Image

For Driver

Visit portal -> login to their account ->Start Facial Recognition-> View Anomalies

4.2.3 Functional Requirements

REQ-1: Correct Input Image

REQ-2: Timely Updates on Server.

REQ-3: Camera is functioning.

Anticipated Errors:

Facial Recognition Failed – View anomalies feature.
Incorrect Input Image

4.3 Payment System

4.3.1 Description and Priority

Description: It is decomposed into Make Payment and View payment Summary processes with Payment confirmation as its output. Once the payment is added, the passengers can login to their accounts and view the payments due. The passengers can pay using the payment gateway and subsequently, their payment will be updated in the Payment details datastore, and they will receive payment confirmation.
Priority: High
Rating
Risk: 7
Cost: 3
Benefit: 8

4.3.2 Stimulus/Response Sequences

For Admin

Visit portal ->visit login page -> View Passenger Payment Records-> Send Alerts

For Passenger

Visit portal ->visit login page -> Check for due payments-> Visit the Payment Page -> Use Payment Gateway-> Receive Confirmation

4.3.3 Functional Requirements

REQ-1: Payment Gateway should be active.

REQ-2: Alerts should be sent on time.

Anticipated Errors:

Payment Gateway Server Down

4.4 Notes

4.4.1 Description and Priority

Description: It is decomposed into Generate Reports and View Reports processes. Reports are generated because of this process. The passengers can view payment receipts and relevant reports on their payments and drivers can view reports on who has paid and who haven't.
Priority: Low
Rating
Risk: 1
Cost: 2
Benefit: 5

4.4.2 Stimulus/Response Sequences

Visit portal ->visit login page -> Check for due payments-> Visit the Payment Page -> Use Payment Gateway-> Receive Confirmation

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Response Time: The system is able to recognize the passenger within a few seconds.
Capacity: The system can support multiple simultaneous logins at once.
User-Interface: The user interface acknowledges within a few seconds.

5.2 Safety Requirements

Due to a virus or OS error, the database could crash at any given time. As a result, taking a database backup is necessary to prevent database loss. In the event of a power supply breakdown, suitable UPS/inverter facilities should be available.

Safety Certifications

ISO/IEC 12207:2008

5.3 Security Requirements

User Identification: The system needs the passenger to recognize herself or himself using valid credentials.
Privacy: Data Leak can result in privacy violation.

Security and Privacy Certifications:

ISO/IEC 27001:2013

5.4 Software Quality Attributes

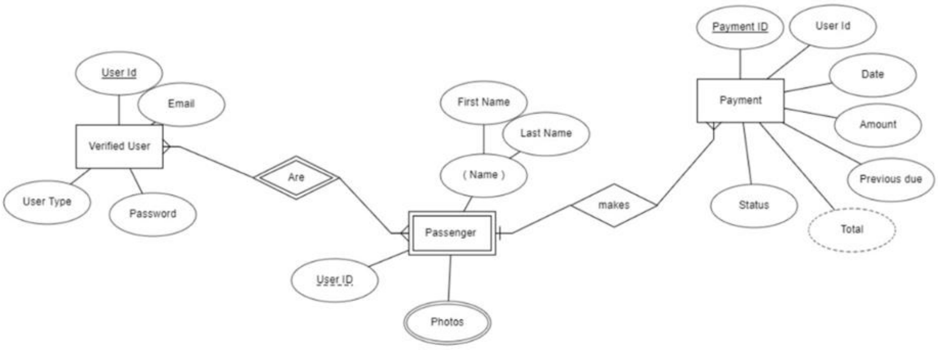
Usability: The system can be used again and again without any malfunctions or distortion.
Availability: The system shall be always available for use.
Correctness: The system will be free of bugs and errors to fulfill the desired requirements of the users.
Maintainability: The system will be maintained and updated with new features regularly to incorporate new user needs.
Portability: The software is operable on multiple OS.
Testability: The software will be tested using various testing techniques (manual, automated, etc) and test cases.

Appendix A: Glossary

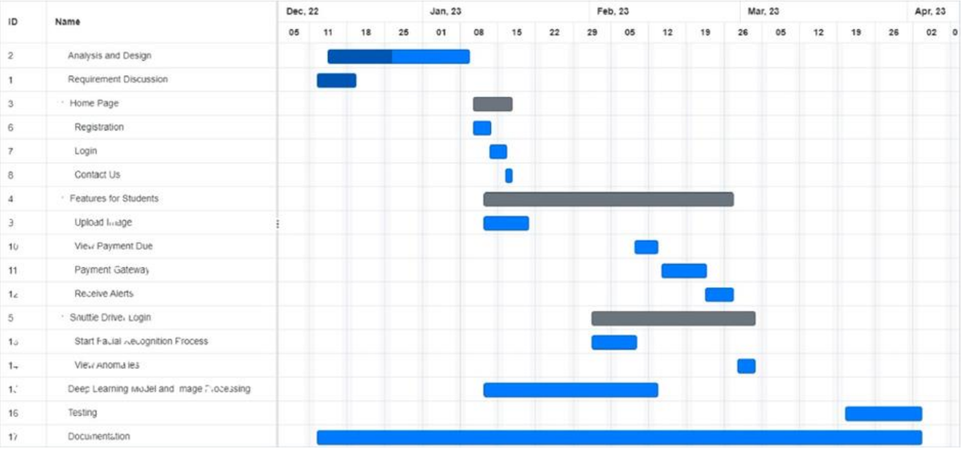
Acronyms	***Abbreviations***
SSMS	*SQL Server Management Studio*
JS	*JavaScript*
WBS	*Work Breakdown Structure*
Req	*Requirements*
GB	*Giga Byte*
SRS	*Software Requirement Specifications*
IDE	*Integrated Development Environment*
HTML	*Hyper Text Markup Language*
CSS	*Cascading Style Sheets*
OS	*Operating System*
ISO	*International Organization for Standardization*
TCP	*Transmission Control Protocol*
IP	*Internet Protocol*
OPT	*Optional*

Appendix B: Analysis Models

Entity Relation Diagram:



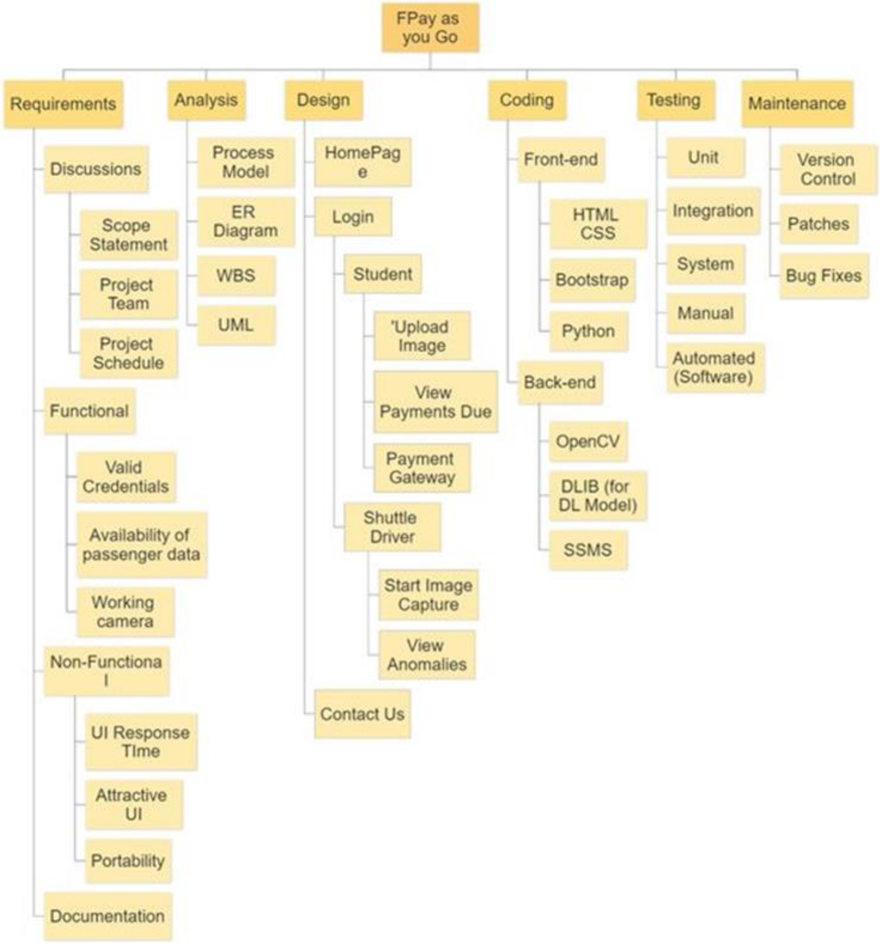
Project Scheduling Gantt Chart:



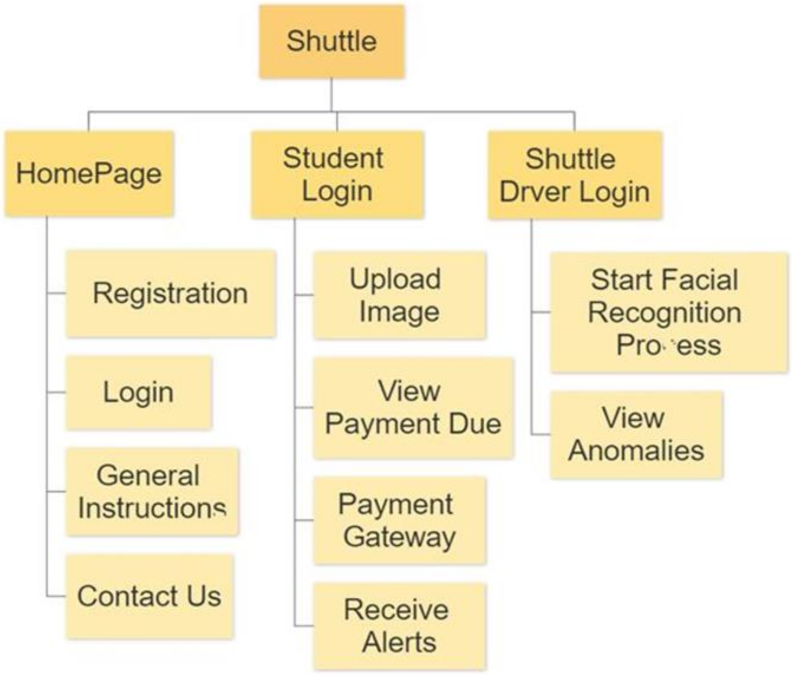
ID	Name
2	Analysis and Design
1	Requirement Discussion
3	Home Page
6	Registration
7	Login
8	Contact Us
4	Features for Students
3	Upload Image
10	View Payment Due
11	Payment Gateway
12	Receive Alerts
5	Shuttle Driver Login
13	Start Facial Recognition Process
14	View Anomalies
15	Deep Learning Model and Image Processing
16	Testing
17	Documentation

Work Breakdown Structure: -

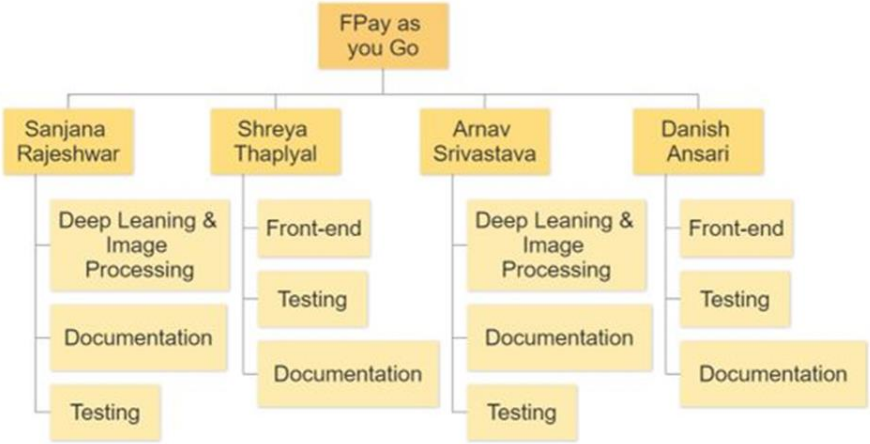
Phase:



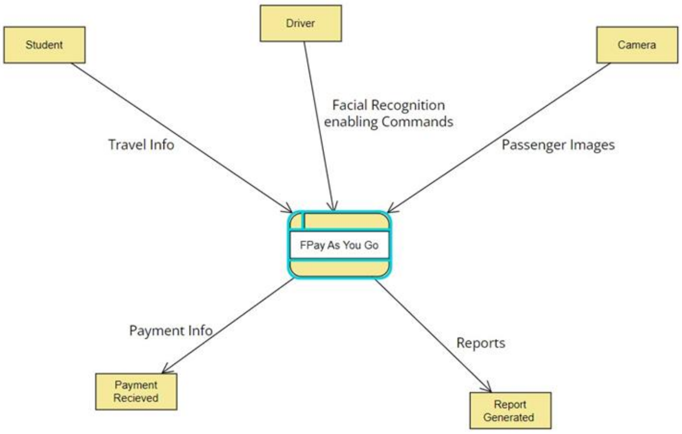
Deliverable:



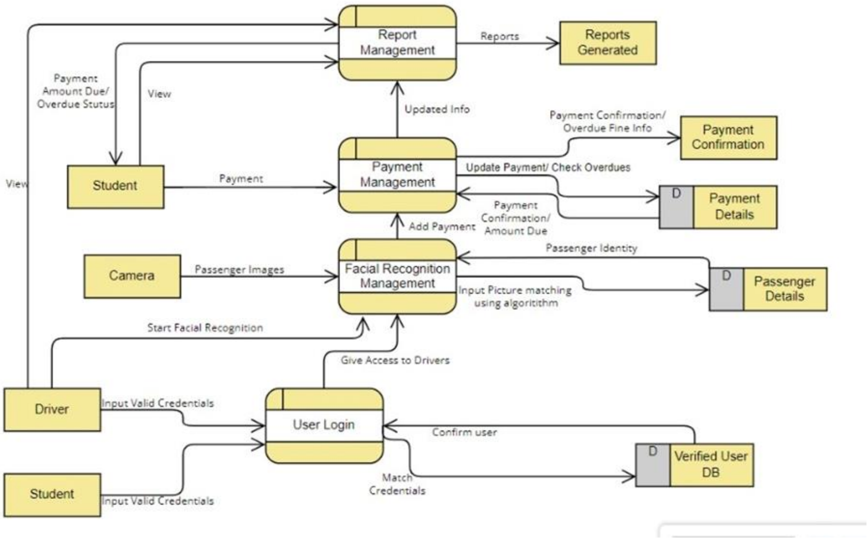
Responsibility:



Context Level Diagram:

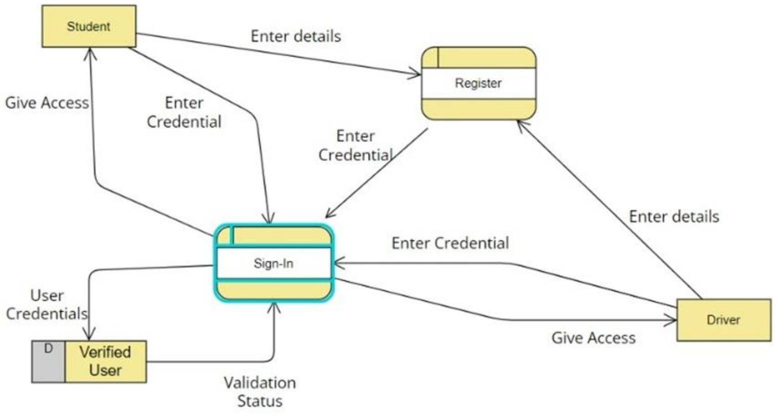


Data Flow Diagram Level 0:

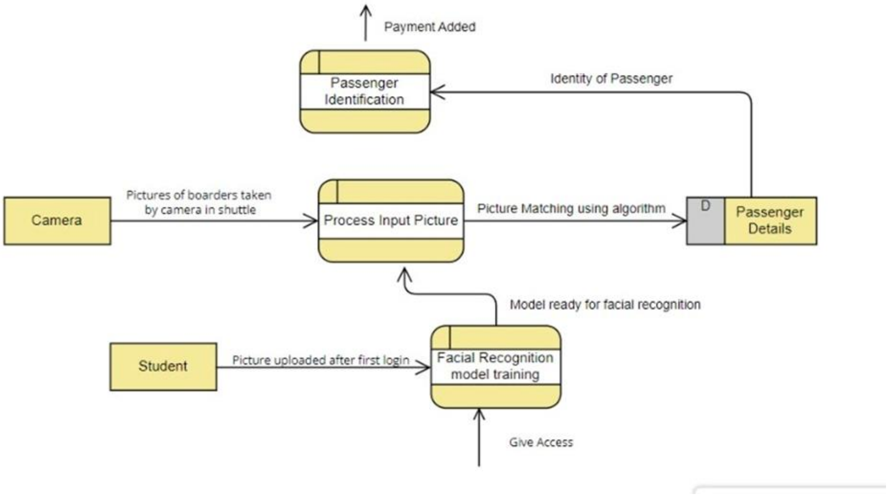


DFD Level 1:

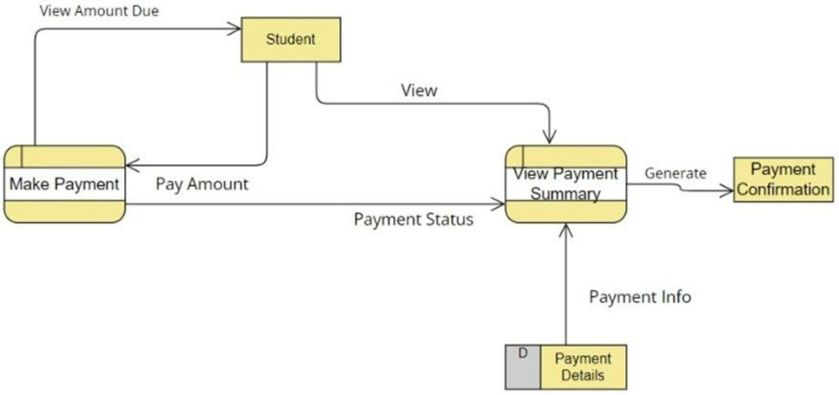
Login



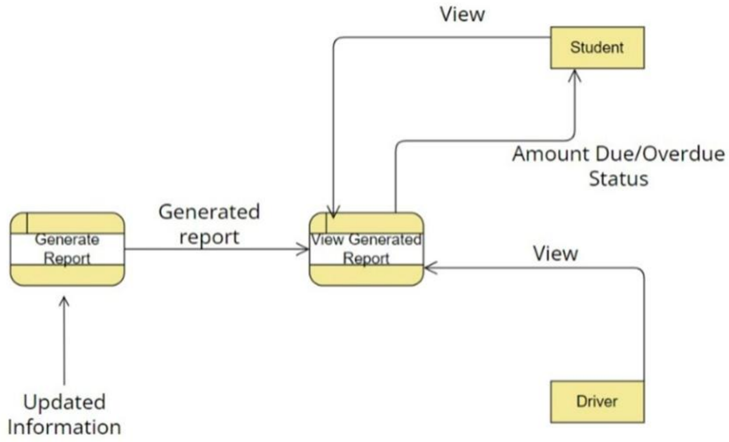
Facial Recognition Management


Payment Added
Passenger Identification
Identity of Passenger
Camera
Pictures of boarders taken by camera in shuttle
Process Input Picture
Picture Matching using algorithm
D
Passenger Details
Model ready for facial recognition
Student
Picture uploaded after first login
Facial Recognition model training
Give Access

Payment System:

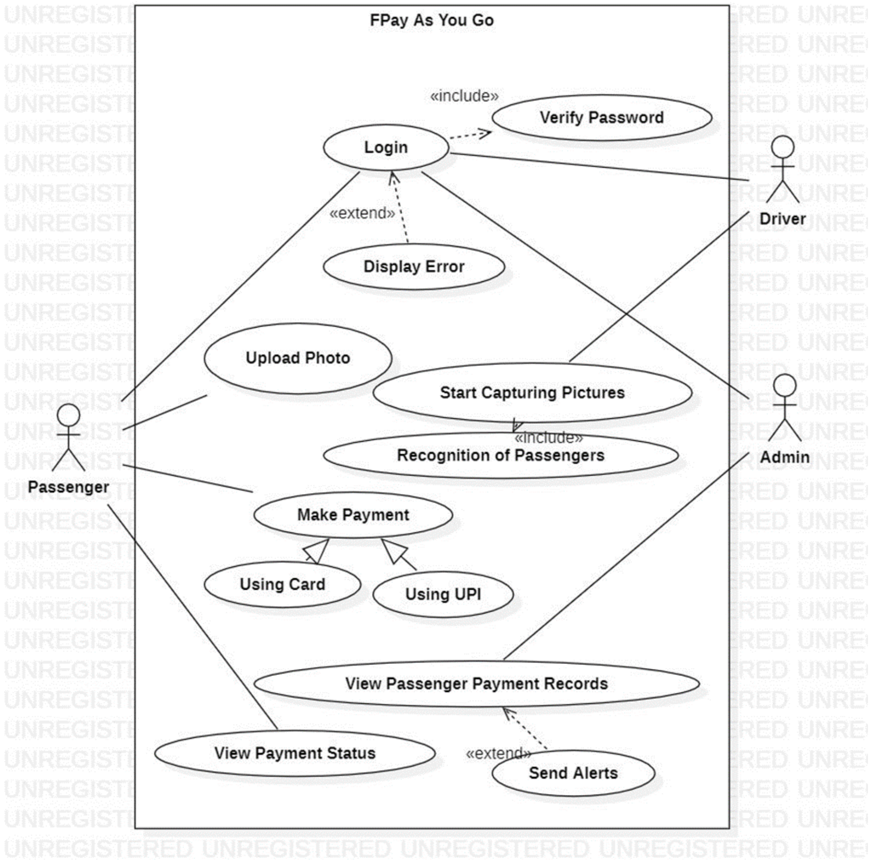


Reports Management System:



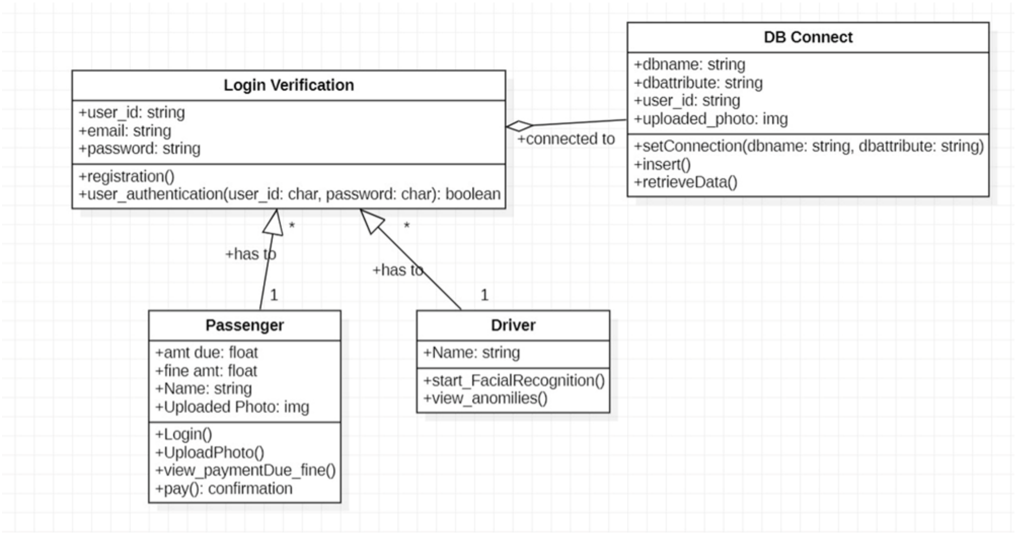
CHAPTER 7

USE CASE DIAGRAM:

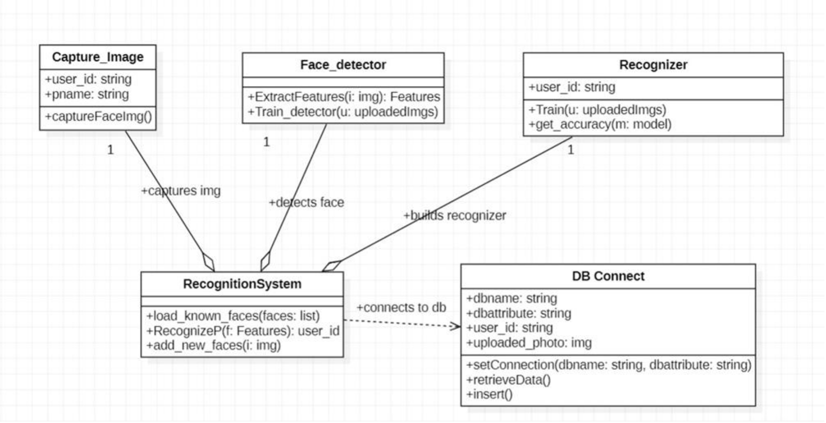


CLASS DIAGRAMS:

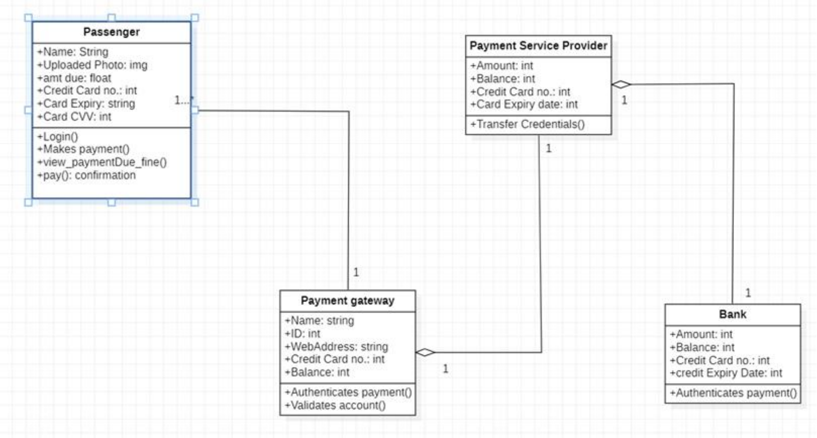
LOGIN SYSTEM:



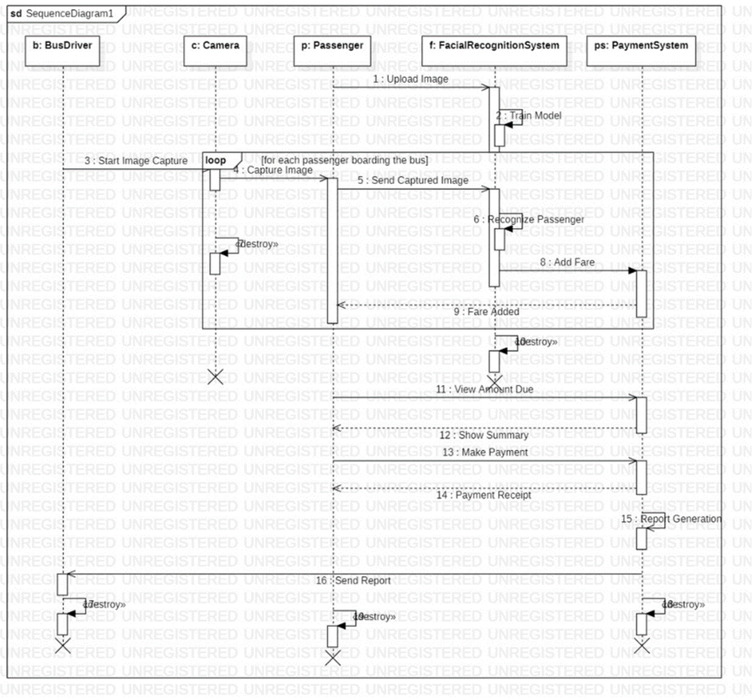
FACIAL RECOGNITION SYSTEM:



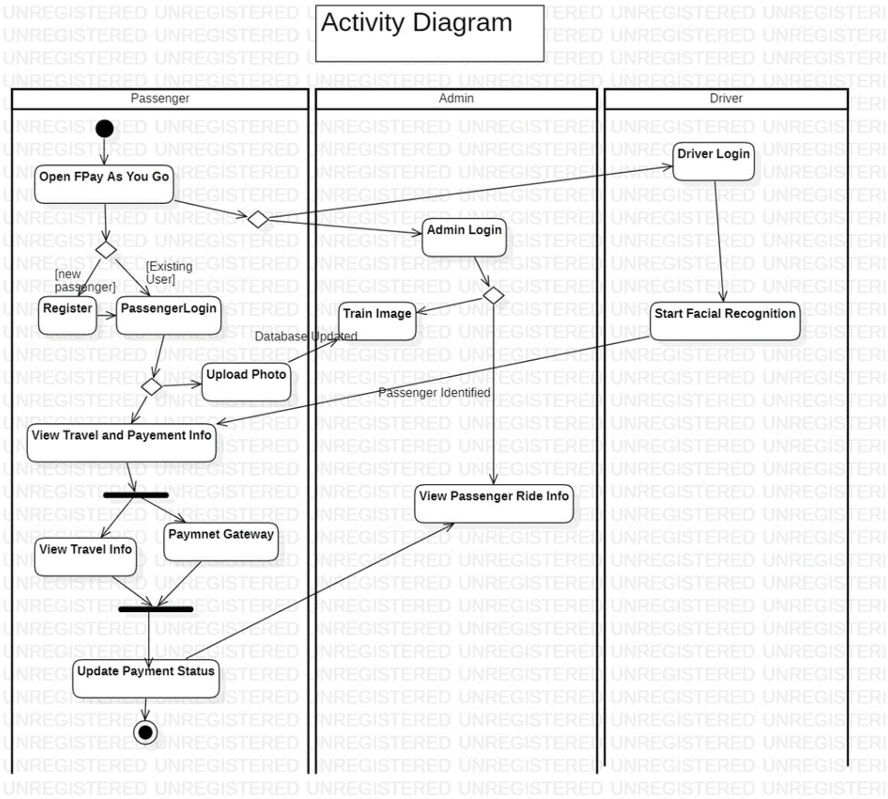
PAYMENT SYSTEM:



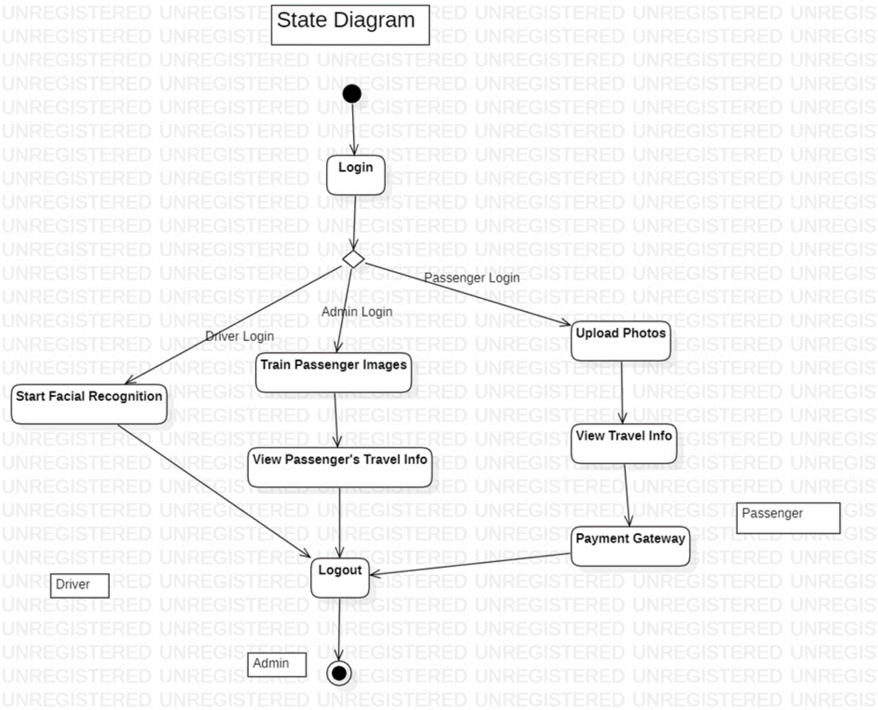
SEQUENCE DIAGRAM:



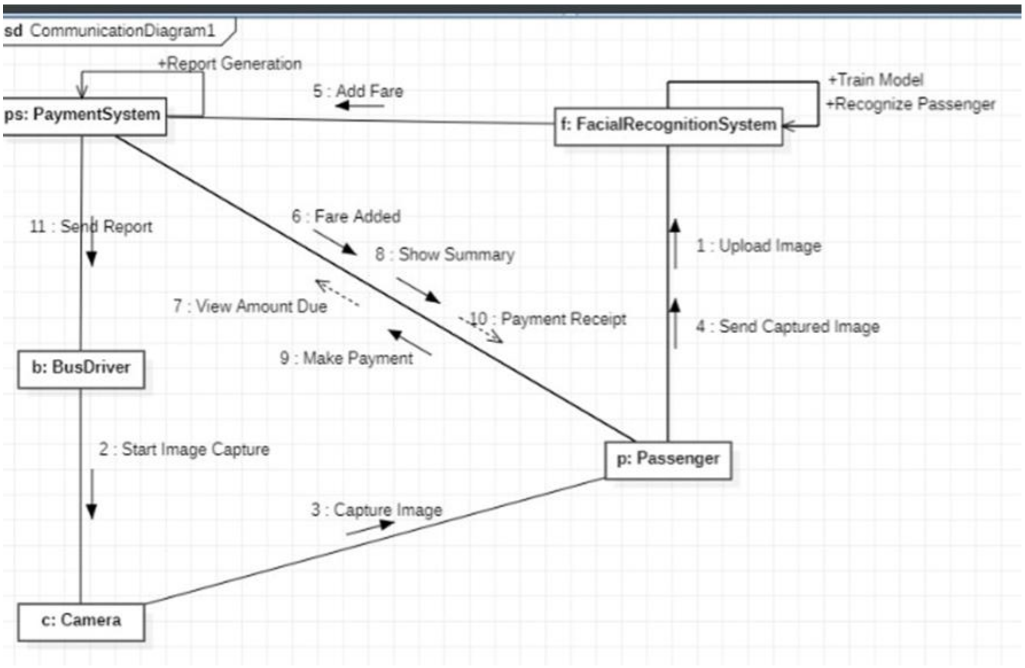
ACTIVITY DIAGRAM:



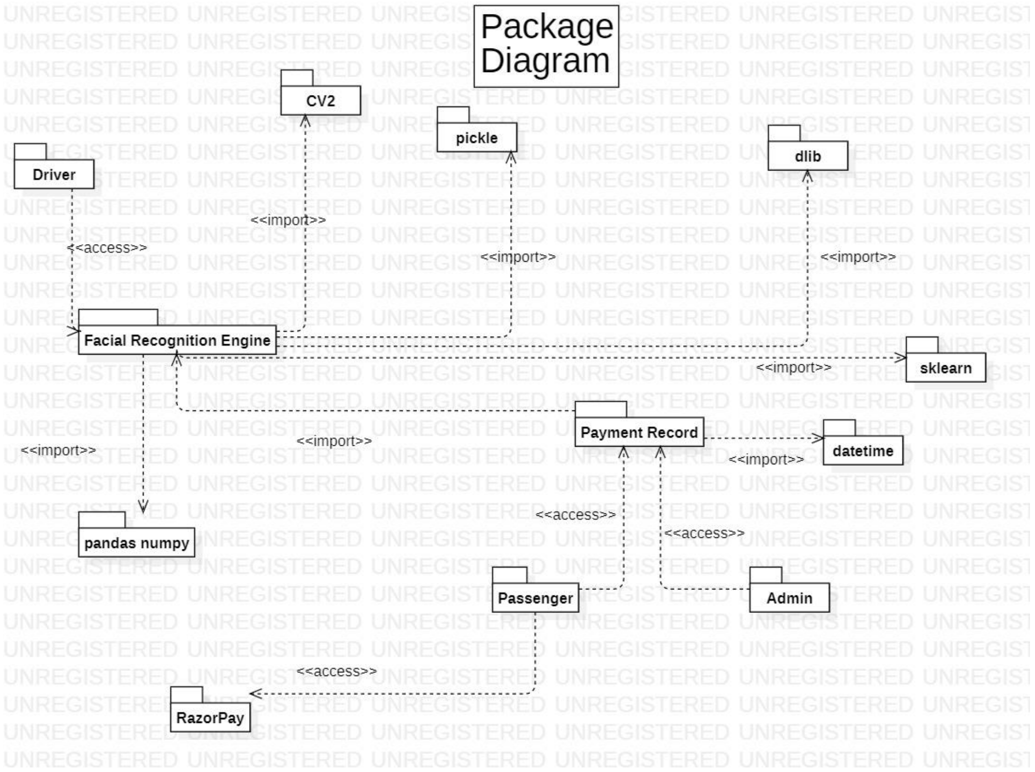
STATE DIAGRAM:



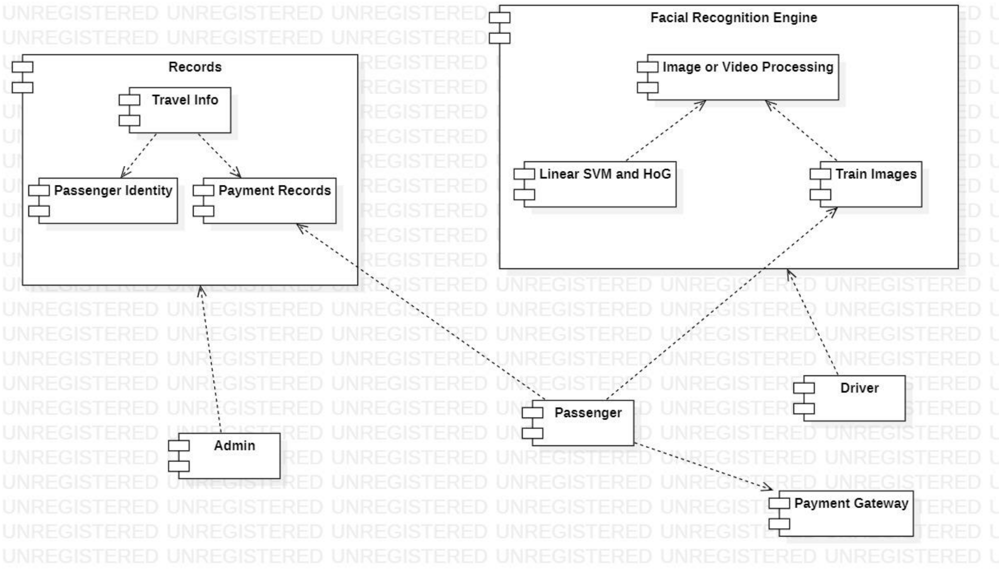
COLLABORATION DIAGRAM:



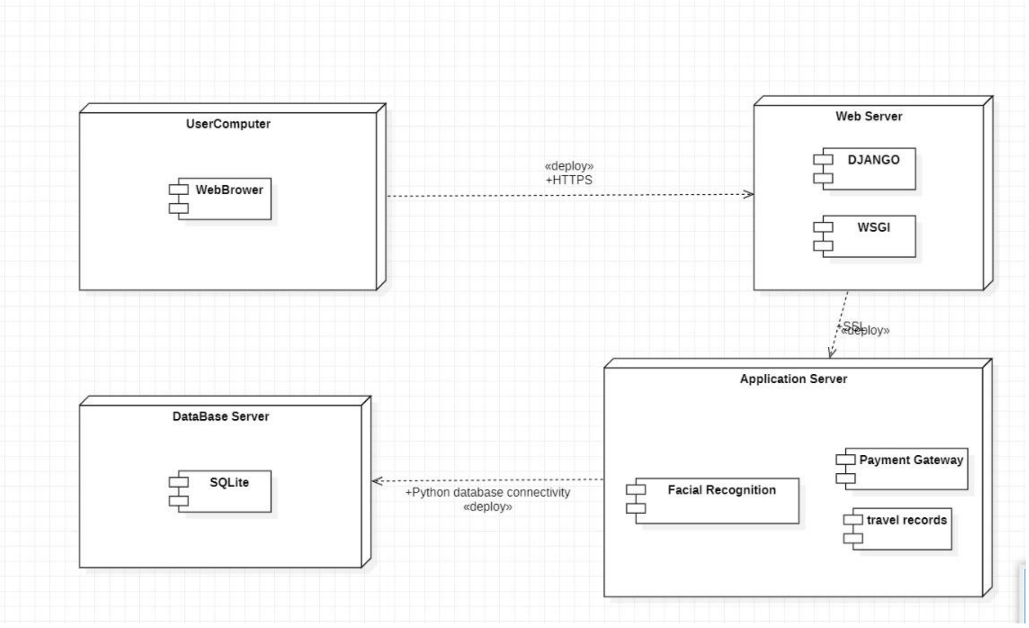
PACKAGE DIAGRAM:



COMPONENT DIAGRAM:



DEPLOYMENT DIAGRAM:



CHAPTER 8: SOURCE CODE

Models.py

```
from django.db import models
from django.contrib.auth.models import User

import datetime
# Create your models here.


class Present(models.Model):
   user=models.ForeignKey(User,on_delete=models.CASCADE)
   date = models.DateField(default=datetime.date.today)
   present=models.BooleanField(default=False)
   total_due = models.IntegerField(default=15)
   email=models.CharField(max_length=100)
   # user_id = models.IntegerField()
   paid = models.BooleanField(default=False)
   payment_name = models.CharField(max_length=50)
   pg_amt = models.IntegerField()

class Time(models.Model):
   user=models.ForeignKey(User,on_delete=models.CASCADE)
   date = models.DateField(default=datetime.date.today)
   time=models.DateTimeField(null=True,blank=True)
   out=models.BooleanField(default=False)
   total_due = models.IntegerField(default=15)

class payment(models.Model):
   payment_name=models.CharField(max_length=50)
   # user=models.ForeignKey(User,on_delete=models.CASCADE)
   total_due = models.IntegerField( blank=True)
   email=models.CharField(max_length=100)
   # user_id = models.IntegerField()
   paid = models.BooleanField(default=False)
```

Manage.py

```
#!/usr/bin/env python
"""Django's command-line utility for administrative tasks."""
import os
import sys


def main():
    os.environ.setdefault('DJANGO_SETTINGS_MODULE',
'payment_system_facial_recognition.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()
```

Views.py

```
import datetime
import math
import os
import pickle
import time
import svm

import numpy as np
import numpy.core.multiarray
import cv2
import dlib
import face_recognition
import imutils
import matplotlib as mpl
# import mpld3
import matplotlib.pyplot as plt

import pandas as pd
import razorpay
import seaborn as sns
from django.conf import settings
from django.contrib import messages
from django.contrib.auth.decorators import login_required
from django.contrib.auth.models import User
from django.core.mail import send_mail
from django.shortcuts import redirect
from django.shortcuts import render
from django.template.loader import render_to_string
from django.views.decorators.csrf import csrf_exempt
from django_pandas.io import read_frame
from face_recognition.face_recognition_cli import image_files_in_folder
from imutils import face_utils
from imutils.face_utils import FaceAligner
from imutils.video import VideoStream
from matplotlib import rcParams
from pandas.plotting import register_matplotlib_converters

from sklearn.manifold import TSNE
from sklearn.preprocessing import LabelEncoder
from sklearn.svm import SVC
from users.models import Present, Time



from datetime import date
```

```
# from datetime import datetime
from datetime import timedelta


from .forms import usernameForm, DateForm, UsernameAndDateForm, DateForm_2
from users.models import donate

mpl.use('Agg')

#utility functions:

def fine(present):
   today = datetime.date.today()
   fineamt=0

 for person in present:
      user = User.objects.get(username=person)
      try:
         qs = Present.objects.get(user=user)
         # , date = present[datetime.date]
      except:
         qs = None
      # if qs is None:
      #  if present[person] == True:
      #     a = Present(user=user, date=today, present=True)
      #     a.save()
      #  else:
      #     a = Present(user=user, date=today, present=False)
      #     a.save()
      # else:
      if qs is not None:
         if present[person] == True:
            if today-qs.date >7:
               Begindate = datetime.strptime(qs.date, "%Y-%m-%d")
               enddate = Begindate + timedelta(days=7)
               fineamt=5*(today-timedelta(enddate))
            qs.total_due+=fineamt
            # qs.present = True
            qs.save(update_fields=['total_due'])
      # if present[person] == True:
      #  a = Time(user=user, date=today, time=time, out=False)
      #  a.save()








def check_validity_times(times_all):

   if(len(times_all)>0):
      sign=times_all.first().out
   else:
      sign=True
   times_in=times_all.filter(out=False)
   times_out=times_all.filter(out=True)
   if(len(times_in)!=len(times_out)):
      sign=True
   break_hourss=0
   if(sign==True):
         check=False
         break_hourss=0
         return (check,break_hourss)
   prev=True
   prev_time=times_all.first().time

   for obj in times_all:
      curr=obj.out
      if(curr==prev):
         check=False
         break_hourss=0
         return (check,break_hourss)
      if(curr==False):
         curr_time=obj.time
         to=curr_time
         ti=prev_time
         break_time=((to-ti).total_seconds())/3600
         break_hourss+=break_time


      else:
         prev_time=obj.time

      prev=curr

   return (True,break_hourss)


def convert_hours_to_hours_mins(hours):

  h=int(hours)
   hours-=h
   m=hours*60
   m=math.ceil(m)
   return str(str(h)+ " hrs " + str(m) + "  mins")
```

```
def total_number_passenger():
   qs=User.objects.all()
   return (len(qs) -1)
   # -1 to account for admin



def passenger_present_today():
   today=datetime.date.today()
   qs=Present.objects.filter(date=today).filter(present=True)
   return len(qs)


 # Create your views here.
def home(request):

   return render(request, 'recognition/home.html')

@login_required
def dashboard(request):
   if(request.user.username=='admin'):
      print("admin")
      return render(request, 'recognition/admin_dashboard.html')
   else:
      print("not admin")

      return render(request,'recognition/passenger_dashboard.html')

@login_required
def add_photos(request):
   # if request.user.username!='admin':
   #  return redirect('not-authorised')
   if request.method=='POST':
      form=usernameForm(request.POST)
      data = request.POST.copy()
      username=data.get('username')
      if username_present(username):
         create_dataset(username)
         messages.success(request, f'Dataset Created')
         return redirect('add-photos')
      else:
         messages.warning(request, f'No such username found. Please register passenger 
first.')
         return redirect('dashboard')


   else:


        form=usernameForm()
         return render(request,'recognition/add_photos.html', {'form' : form})


@login_required
def mark_your_rides(request):
   if request.user.username.startswith('admin'):
      return redirect('not-authorised')



  detector = dlib.get_frontal_face_detector()

  predictor =
dlib.shape_predictor('face_recognition_data/shape_predictor_68_face_landmarks.dat')
#Add path to the shape predictor ######CHANGE TO RELATIVE PATH LATER
   svc_save_path="face_recognition_data/svc.sav"




  with open(svc_save_path, 'rb') as f:
         svc = pickle.load(f)
   fa = FaceAligner(predictor , desiredFaceWidth = 96)
   encoder=LabelEncoder()
   encoder.classes_ = np.load('face_recognition_data/classes.npy')


   faces_encodings = np.zeros((1,128))
   no_of_faces = len(svc.predict_proba(faces_encodings)[0])
   count = dict()
   present = dict()
   log_time = dict()
   start = dict()
   for i in range(no_of_faces):
      count[encoder.inverse_transform([i])[0]] = 0
      present[encoder.inverse_transform([i])[0]] = False



  vs = VideoStream(src=0).start()

  sampleNum = 0

  while(True):

     frame = vs.read()

     frame = imutils.resize(frame ,width = 800)

     gray_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

     faces = detector(gray_frame,0)




     for face in faces:
         print("INFO : inside for loop")
         (x,y,w,h) = face_utils.rect_to_bb(face)

         face_aligned = fa.align(frame,gray_frame,face)
         cv2.rectangle(frame,(x,y),(x+w,y+h),(0,255,0),1)


        (pred,prob)=predict(face_aligned,svc)



        if(pred!=[-1]):

           person_name=encoder.inverse_transform(np.ravel([pred]))[0]
            pred=person_name
            if count[pred] == 0:
               start[pred] = time.time()
               count[pred] = count.get(pred,0) + 1

            if count[pred] == 4 and (time.time()-start[pred]) > 1.2:
                count[pred] = 0
            else:
            #if count[pred] == 4 and (time.time()-start) <= 1.5:
               present[pred] = True
               log_time[pred] = datetime.datetime.now()
               count[pred] = count.get(pred,0) + 1
               print(pred, present[pred], count[pred])
            cv2.putText(frame, str(person_name)+ str(prob), (x+6,y+h-6),
cv2.FONT_HERSHEY_SIMPLEX,0.5,(0,255,0),1)

         else:
            person_name="unknown"
            cv2.putText(frame, str(person_name), (x+6,y+h-6),
cv2.FONT_HERSHEY_SIMPLEX,0.5,(0,255,0),1)



        #cv2.putText()
         # Before continuing to the next loop, I want to give it a little pause
         # waitKey of 100 millisecond
         #cv2.waitKey(50)

      #Showing the image in another window
      #Creates a window with window name "Face" and with the image img
      cv2.imshow("Mark Rides - In - Press q to exit",frame)
      #Before closing it we need to give a wait command, otherwise the open cv wont
work
      # @params with the millisecond of delay 1
      #cv2.waitKey(1)
      #To get out of the loop
      key=cv2.waitKey(50) & 0xFF
      if(key==ord("q")):
         break

  #Stoping the videostream
   vs.stop()
   # destroying all the windows
   cv2.destroyAllWindows()
   update_rides_in_db_in(present)
   # fine(present)
   return redirect('home')





def mark_your_rides_out(request):



  detector = dlib.get_frontal_face_detector()

  predictor =
dlib.shape_predictor('face_recognition_data/shape_predictor_68_face_landmarks.dat')
#Add path to the shape predictor ######CHANGE TO RELATIVE PATH LATER
   svc_save_path="face_recognition_data/svc.sav"





  with open(svc_save_path, 'rb') as f:
         svc = pickle.load(f)
   fa = FaceAligner(predictor , desiredFaceWidth = 96)
   encoder=LabelEncoder()
   encoder.classes_ = np.load('face_recognition_data/classes.npy')


   faces_encodings = np.zeros((1,128))
   no_of_faces = len(svc.predict_proba(faces_encodings)[0])
   count = dict()
   present = dict()
   log_time = dict()
   start = dict()
   for i in range(no_of_faces):
      count[encoder.inverse_transform([i])[0]] = 0
      present[encoder.inverse_transform([i])[0]] = False



  vs = VideoStream(src=0).start()

  sampleNum = 0

  while(True):

     frame = vs.read()

     frame = imutils.resize(frame ,width = 800)
     gray_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

     faces = detector(gray_frame,0)




     for face in faces:
         print("INFO : inside for loop")
         (x,y,w,h) = face_utils.rect_to_bb(face)

         face_aligned = fa.align(frame,gray_frame,face)
         cv2.rectangle(frame,(x,y),(x+w,y+h),(0,255,0),1)


        (pred,prob)=predict(face_aligned,svc)




        if(pred!=[-1]):

           person_name=encoder.inverse_transform(np.ravel([pred]))[0]
            pred=person_name
            if count[pred] == 0:
               start[pred] = time.time()
               count[pred] = count.get(pred,0) + 1

            if count[pred] == 4 and (time.time()-start[pred]) > 1.5:
                count[pred] = 0
            else:
            #if count[pred] == 4 and (time.time()-start) <= 1.5:
               present[pred] = True
               log_time[pred] = datetime.datetime.now()
               count[pred] = count.get(pred,0) + 1
               print(pred, present[pred], count[pred])
            cv2.putText(frame, str(person_name)+ str(prob), (x+6,y+h-6),
cv2.FONT_HERSHEY_SIMPLEX,0.5,(0,255,0),1)

         else:
            person_name="unknown"
            cv2.putText(frame, str(person_name), (x+6,y+h-6),
cv2.FONT_HERSHEY_SIMPLEX,0.5,(0,255,0),1)




        #cv2.putText()
         # Before continuing to the next loop, I want to give it a little pause
         # waitKey of 100 millisecond
         #cv2.waitKey(50)

      #Showing the image in another window
      #Creates a window with window name "Face" and with the image img
      cv2.imshow("Mark Rides- Out - Press q to exit",frame)
      #Before closing it we need to give a wait command, otherwise the open cv wont
work
      # @params with the millisecond of delay 1
      #cv2.waitKey(1)
      #To get out of the loop
      key=cv2.waitKey(50) & 0xFF
      if(key==ord("q")):
         break

  #Stoping the videostream
   vs.stop()

   # destroying all the windows
   cv2.destroyAllWindows()
   update_rides_in_db_out(present)
   return redirect('home')





@login_required
def train(request):
   # if request.user.username!='admin':
   #  return redirect('not-authorised')

   training_dir='face_recognition_data/training_dataset'



  count=0
   for person_name in os.listdir(training_dir):
      curr_directory=os.path.join(training_dir,person_name)
      if not os.path.isdir(curr_directory):
         continue
      for imagefile in image_files_in_folder(curr_directory):
         count+=1

   X=[]
   y=[]
   i=0


   for person_name in os.listdir(training_dir):
      print(str(person_name))
      curr_directory=os.path.join(training_dir,person_name)
      if not os.path.isdir(curr_directory):
         continue
      for imagefile in image_files_in_folder(curr_directory):
         print(str(imagefile))
         image=cv2.imread(imagefile)
         try:
            X.append((face_recognition.face_encodings(image)[0]).tolist())
```

```
           y.append(person_name)
            i+=1
         except:
            print("removed")
            os.remove(imagefile)




  targets=np.array(y)
   encoder = LabelEncoder()
   encoder.fit(y)
   y=encoder.transform(y)
   X1=np.array(X)
   print("shape: "+ str(X1.shape))
   np.save('face_recognition_data/classes.npy', encoder.classes_)
   svc = SVC(kernel='linear',probability=True)
   svc.fit(X1,y)
   svc_save_path="face_recognition_data/svc.sav"
   with open(svc_save_path, 'wb') as f:
      pickle.dump(svc,f)


  vizualize_Data(X1,targets)

  messages.success(request, f'Training Complete.')

   return render(request,"recognition/train.html")


@login_required
def not_authorised(request):
   return render(request,'recognition/not_authorised.html')



@login_required
def view_rides_home(request):
   total_num_of_emp=total_number_passenger()
   emp_present_today=passenger_present_today()
   this_week_emp_count_vs_date()
   last_week_emp_count_vs_date()
   return render(request,"recognition/view_rides_home.html", {'total_num_of_emp' :
total_num_of_emp, 'emp_present_today': emp_present_today})


@login_required
def view_rides_date(request):
   if request.user.username!='admin':
      return redirect('not-authorised')
   qs=None
   time_qs=None
   present_qs=None
   if request.method=='POST':
      form=DateForm(request.POST)
      if form.is_valid():
         date=form.cleaned_data.get('date')
         print("date:"+ str(date))
         time_qs=Time.objects.filter(date=date)
         present_qs=Present.objects.filter(date=date)
         if(len(time_qs)>0 or len(present_qs)>0):
            qs=hours_vs_employee_given_date(present_qs,time_qs)


            return render(request,'recognition/view_rides_date.html', {'form' :
form,'qs' : qs })
         else:
            messages.warning(request, f'No records for selected date.')
            return redirect('view-rides-date')









  else:


        form=DateForm()
         return render(request,'recognition/view_rides_date.html', {'form' : form,
'qs' : qs})


@login_required
def view_rides_passenger(request):
   if request.user.username!='admin':
      return redirect('not-authorised')
   time_qs=None
   present_qs=None
   qs=None

   if request.method=='POST':
      form=UsernameAndDateForm(request.POST)
      if form.is_valid():
         username=form.cleaned_data.get('username')
         if username_present(username):

           u=User.objects.get(username=username)

           time_qs=Time.objects.filter(user=u)
            present_qs=Present.objects.filter(user=u)
            date_from=form.cleaned_data.get('date_from')
            date_to=form.cleaned_data.get('date_to')
           if date_to < date_from:
               messages.warning(request, f'Invalid date selection.')
               return redirect('view-rides-employee')
            else:



time_qs=time_qs.filter(date__gte=date_from).filter(date__lte=date_to).order_by('-
date')

present_qs=present_qs.filter(date__gte=date_from).filter(date__lte=date_to).order_by('
-date')

              if (len(time_qs)>0 or len(present_qs)>0):
                  qs=hours_vs_date_given_employee(present_qs,time_qs,admin=True)
                  return render(request,'recognition/view_rides_employee.html',
{'form' : form, 'qs' :qs})
               else:
                  #print("inside qs is None")
                  messages.warning(request, f'No records for selected duration.')
                  return redirect('view-rides-employee')







        else:
            print("invalid username")
            messages.warning(request, f'No such username found.')
            return redirect('view-rides-employee')


   else:


        form=UsernameAndDateForm()
         return render(request,'recognition/view_rides_employee.html', {'form' : form,
'qs' :qs})





@login_required
def view_my_rides_passenger_login(request):
   if request.user.username=='admin':
      return redirect('not-authorised')
   qs=None
   time_qs=None
   present_qs=None
   if request.method=='POST':
      form=DateForm_2(request.POST)
      if form.is_valid():
         u=request.user
         time_qs=Time.objects.filter(user=u)
         present_qs=Present.objects.filter(user=u)
         date_from=form.cleaned_data.get('date_from')
         date_to=form.cleaned_data.get('date_to')
         if date_to < date_from:
               messages.warning(request, f'Invalid date selection.')
               return redirect('view-my-rides-passenger-login')
         else:



time_qs=time_qs.filter(date__gte=date_from).filter(date__lte=date_to).order_by('-
date')

present_qs=present_qs.filter(date__gte=date_from).filter(date__lte=date_to).order_by('
-date')

              if (len(time_qs)>0 or len(present_qs)>0):
                  qs=hours_vs_date_given_employee(present_qs,time_qs,admin=False)
                  return
render(request,'recognition/view_my_rides_passenger_login.html', {'form' : form, 'qs'
:qs})
               else:

                 messages.warning(request, f'No records for selected duration.')
                  return redirect('view-my-rides-employee-login')
   else:


        form=DateForm_2()
         return render(request,'recognition/view_my_rides_employee_login.html',
{'form' : form, 'qs' :qs})

def amt(present):
   pg_amt = 0
   form = Present.objects.all()
   context = {'form': form}
   for person in form:
      user = User.objects.get(username=person)
      try:
         qs = Present.objects.get(user=user)
      except:
         qs = None


      if qs is not None:
         pg_amt += qs.total_due

   return pg_amt




def pg(request):
   if request.method == "POST":
      # for person in request:
      #  user = User.objects.get(username=person)
      #  try:
      #     qs = Present.objects.get(user=user)
      #  except:
      #     qs = None

      amount=int(126)*100

      name = request.POST.get('name')
      # amount = int(request.POST.get('amount')) * 100
      # email = request.POST.get('email')
      client = razorpay.Client(auth=("rzp_test_ifqXZb84qSL1CP",
"IwSyyaBvXh300nlqM0kqb0ow"))
      payment = client.order.create({'amount': amount, 'currency': 'INR',
                              'payment_capture': '1'})

      info = Present(payment_name=name, total_due=amount)
      # , email = email
      #  , user_id = payment['id']
      info.save()

      return render(request, 'recognition/index.html', {'payment': payment})
   return render(request, 'recognition/index.html')


   return redirect(dashboard)
```

Urls.py

```
The `urlpatterns` list routes URLs to views. For more information please see:
    https://docs.djangoproject.com/en/2.2/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
from django.contrib.auth import views as auth_views
from recognition import views as recog_views
from users import views as users_views

urlpatterns = [
    path('admin/', admin.site.urls),
     path('', recog_views.home, name='home'),

     path('dashboard/', recog_views.dashboard, name='dashboard'),
      path('train/', recog_views.train, name='train'),
    path('add_photos/', recog_views.add_photos, name='add-photos'),


path('login/',auth_views.LoginView.as_view(template_name='users/login.html'),name='log
in'),

path('logout/',auth_views.LogoutView.as_view(template_name='recognition/home.html'),na
me='logout'),
     path('register/', users_views.register, name='register'),
     path('mark_your_rides', recog_views.mark_your_rides ,name='mark-your-rides'),

      path('view_attendance_home', recog_views.view_attendance_home ,name='view-
attendance-home'),

       path('view_attendance_date', recog_views.view_attendance_date ,name='view-
attendance-date'),
        path('view_rides_passenger', recog_views.view_rides_passenger ,name='view-
rides-passenger'),
         path('view_my_rides ', recog_views.view_my_rides_passenger_login ,name='view-
my-rides-passenger-login'),
       path('not_authorised', recog_views.not_authorised, name='not-authorised'),
    path('pg',recog_views.pg,name='pg'),
path('success',recog_views.success,name='success'),
    # path('fine',recog_views.fine,name='fine')

]
```

Apps.py

```
from django.apps import AppConfig


class RecognitionConfig(AppConfig):
    name = 'recognition'
```

Base.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Razorpay</title>
    <!-- CSS only -->
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
integrity="sha384-JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZOnIxN0hoP+VmmDGMN5t9UJ0Z"
crossorigin="anonymous">

</head>
<body>
```

```
   {% block content %}{% endblock %}




   <!-- JS, Popper.js, and jQuery -->
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj" crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js" integrity="sha384-9/reFTGAW83EW2RDu2S0VKaIzap3H66lZH81PoYlFhbGU+6BZp6G7niu735Sk7lN" crossorigin="anonymous"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js" integrity="sha384-B4gt1jrGC7Jh4AgTPSdUtOBvfO8shuf57BaghqFfPlYxofvL8/KUEfYiJOMMV+rV" crossorigin="anonymous"></script>

</body>
</html>
```

Add_photos.html

```
{% load static %}
{% load crispy_forms_tags %}

<!DOCTYPE html>
<html>
<head>
     <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">

   <style>
    body{
      background: url('{% static "recognition/img/bg_image2.png"%}') no-repeat center center fixed;
      background-size: cover;

   }

  </style>



</head>
<body>
```

```
<div class="col-lg-12" style="background: rgba(0,0,0,0.6);max-height: 20px ; padding-
top:1em;padding-bottom:3em;color:#fff;border-radius:10px;-webkit-box-shadow: 2px 2px
15px 0px rgba(0, 3, 0, 0.7);
-moz-box-shadow:    2px 2px 15px 0px rgba(0, 3, 0, 0.7);
box-shadow:         2px 2px 15px 0px rgba(0, 3, 0, 0.7); margin-left:auto; margin-
right: auto; ">

  <a href="{% url 'dashboard' %}"><h5 class="text-left">  Back</h5></a>
</div>

 <div class="col-lg-4" style="background: rgba(0,0,0,0.6);margin-top:150px ; padding-
top:1em;padding-bottom:3em;color:#fff;border-radius:10px;-webkit-box-shadow: 2px 2px
15px 0px rgba(0, 3, 0, 0.7);
-moz-box-shadow:    2px 2px 15px 0px rgba(0, 3, 0, 0.7);
box-shadow:         2px 2px 15px 0px rgba(0, 3, 0, 0.7); margin-left:auto; margin-
right: auto; ">


 <form method="POST" >
      {% csrf_token %}
      <fieldset class="form-group">
        <legend class="border-bottom mb-4"> Enter Username </legend>
        {{form| crispy}}
      </fieldset>

      <div class="form-group">
        <button class="btn btn-outline-info" type="submit"> Submit</button>
      </div>
    </form>
</div>
<div class="col-lg-12" style="padding-top: 100px;">
 {% if messages %}
      {% for message in messages%}
      <div class="alert alert-{{message.tags}}" > {{message}}
      </div>
      {%endfor %}

    {%endif%}
  </div>



  <!-- Optional JavaScript -->
    <!-- jQuery first, then Popper.js, then Bootstrap JS -->
    <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-
KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
crossorigin="anonymous"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
crossorigin="anonymous"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl"
crossorigin="anonymous"></script>
</body>
</html>
```

Admin_dashboard.html

```
{% load static %}


<!DOCTYPE html>
<html>
<head>
     <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
crossorigin="anonymous">

   <style>
    body{
      background: url('{% static "recognition/img/bg_image2.png"%}') no-repeat center
center fixed;
      background-size: cover;

   }

  </style>

<script  type='text/javascript'
src='https://ajax.googleapis.com/ajax/libs/jquery/2.1.1/jquery.min.js'>
$(document).ready(function(){
  $("#train").click(function(){
    alert("Training begun. This may take upto 5 minutes. Please wait.");
  });
});
</script>

</head>
<body>
   <div class="col-lg-12" style="background: rgba(0,0,0,0.6);margin-top:3em;margin-
bottom:5em;padding-top:1em;padding-bottom:3em;color:#fff;border-radius:10px;-webkit-
box-shadow: 2px 2px 15px 0px rgba(0, 3, 0, 0.7);
-moz-box-shadow:    2px 2px 15px 0px rgba(0, 3, 0, 0.7);
box-shadow:         2px 2px 15px 0px rgba(0, 3, 0, 0.7);">
    <div class="col-sm-12">
       <a href="{% url 'logout' %}"><h5 class="text-right"><i class="fa fa-user-
circle" aria-hidden="true"></i> Logout</h5></a>
      <h1 class="text-center section-title" style="margin-
bottom:2em">Welcome,admin</h1>
    </div>
     <style>
      h4{
        margin-bottom: 1.5em;
        padding-top: 30px;
      }
      img{
        border-radius:50%;
        -webkit-transition: all 0.3s ease-in-out;
    -moz-transition: all 0.3s ease-in-out;
     transition: all 0.3s ease-in-out;
      }
      .center{
      margin: auto;
  width: 50vh;
<!--  border: 3px solid green;-->
  padding: 10px;
  text-align:center;
      }
      img:hover{
        -webkit-box-shadow: 2px 2px 21px 0px rgba(0, 3, 0, 0.91);
        -moz-box-shadow:    2px 2px 21px 0px rgba(0, 3, 0, 0.91);
        box-shadow:         2px 2px 21px 0px rgba(0, 3, 0, 0.91);
        border:2px solid #fff;
      }
      h3{
        margin-bottom: 1.3em;
      }
      a{
        color:inherit
      }
      a:hover{
        color:inherit;
        text-decoration: none;
      }
      /*
      .section-title:after {
          content:' ';display:block;margin:0 auto;width:100px;margin-top:
6px;border:2px solid #d0d0d0;border-radius:4px;
          -webkit-border-radius:4px;
          -moz-border-radius:4px;
          box-shadow:inset 0 1px 1px rgba(0, 0, 0, .05);
          -webkit-box-shadow:inset 0 1px 1px rgba(0, 0, 0, .05);
          -moz-box-shadow:inset 0 1px 1px rgba(0, 0, 0, .05);
          margin-bottom:1em;
        }
        */
        .att{
          padding-right: 150px;
        }
        .lgn{
          padding-right: 80px;
        }
    </style>




<!--      <div class="row">-->
```

```
<!--           <div class="col-md-3">-->
<!--          <a href="{%url 'register' %}"><img src="{% static
'recognition/img/register.png' %}" class="img-responsive"
style="width:300px;height:300px ;" /></a>-->

<!--        </div>-->
<!--          <div class="col-md-3">-->
<!--          <a href="{%url 'add-photos'%}"><img src="{% static
'recognition/img/addphotos.png' %}" class="img-responsive"
style="width:300px;height:300px ;" /></a>-->
<!--          -->
<!--        </div>-->
<!--           <div class="col-md-3">-->
<!--          <a href="{%url 'train' %}" id="train" ><img src="{% static
'recognition/img/train.jpeg' %}" class="img-responsive"
style="width:300px;height:300px ;" /></a>-->
<!--          -->
<!--        </div>-->
<!--        <div class="col-md-3">-->
<!--          <a href="{%url 'view-attendance-home' %}"><img src="{% static
'recognition/img/reports.png' %}" class="img-responsive"
style="width:300px;height:300px"/></a>-->

<!--        </div>-->

<!--<div class="row " >-->
<!--            <div class="col-md-2">-->
<!--        </div>-->
<!--    <div class="col-md-2">-->
<!--        </div>-->
<!--               <div class="col-md-4">-->
<!--          <a href="{%url 'train' %}" id="train" ><img src="{% static
'recognition/img/train.jpeg' %}" class="img-responsive"
style="width:300px;height:300px ;" /></a>-->
<!--        </div>-->
<!--    <h4 class="text-center lgn">Train Image</h4>-->
<!--    <div class="center">-->
<!--          <a href="{%url 'view-attendance-home' %}"><img src="{% static
'recognition/img/reports.png' %}" class="img-responsive"
style="width:300px;height:300px"/></a>-->
<!--<h4 class="text-center lgn">View Attendance Reports </h4>-->
<!--        </div>-->

<!--    <div class="col-md-2">-->
<!--        </div>-->
<!--    <div class="col-md-2">-->
<!--        </div>-->



    <div class="row" style=" display:flex; float:none; margin-left:10px">
          <div class="col-md-3">

          </div>
          <div class="col-md-3">
          <a href="{%url 'train'%}"><img src="{% static 'recognition/img/train.jpeg'
%}" class="img-responsive" style="width:300px;height:300px ;" /></a>
        </div>
        <div class="col-md-3">
          <a href="{%url 'view-attendance-home' %}"><img src="{% static
'recognition/img/reports.png' %}" class="img-responsive"
style="width:300px;height:300px"/></a>
        </div>

          <div class="col-md-3">

          </div>
       </div>

<!--      </div>-->

         <div class="row" style=" display:flex; float:none; margin-left:50px">
        <div class="col-md-3">

          </div>
        <div class="col md-3" >
             <h4 >Train Passenger Images</h4>
        </div>
<!--      <div class="col md-6" >-->
<!--             <h4 >View My Attendance</h4>-->
<!--        </div>-->
        <div class="col md-3" >
             <h4 >View Passenger Statistics </h4>
        </div>
      <div class="col-md-3">

          </div>
</div>

<!--  <div class="row">-->



<!--        <div class="col md-3">-->
<!--   <h4 class="text-center att">Register New Employees</h4>-->
<!--        </div>-->
<!--          <div class="col md-3">-->
<!--   <h4 class="text-center att">Add Photos</h4>-->
<!--        </div>-->
<!--            <div class="col md-3">-->
<!--   <h4 class="text-center att">Train</h4>-->
<!--        </div>-->
<!--        <div class="col md-3">-->
<!--            <div class="col-md-3">-->
<!--          <a href="{%url 'view-attendance-home' %}"><img src="{% static
'recognition/img/reports.png' %}" class="img-responsive"
style="width:300px;height:300px"/></a>-->
<!--             <h4 class="text-center lgn">View Attendance Reports </h4>-->
<!--        </div>-->
<!--     </div>-->
```

```
</div>

 {% if messages %}
      {% for message in messages%}
      <div class="alert alert-{{message.tags}}" > {{message}}
      </div>
      {%endfor %}

    {%endif%}


</div>




<!-- Optional JavaScript -->
    <!-- jQuery first, then Popper.js, then Bootstrap JS -->
    <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN" crossorigin="anonymous"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js" integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q" crossorigin="anonymous"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl" crossorigin="anonymous"></script>
</body>
</html>
```

Employee_dashboard.html

```
{% load static %}


<!DOCTYPE html>
<html>
<head>
     <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">

   <style>
    body{
      background: url('{% static "recognition/img/bg_image2.png"%}') no-repeat center center fixed;
      background-size: cover;
   }

  </style>



</head>
<body>
   <div class="col-lg-12" style="background: rgba(0,0,0,0.6);margin-top:3em;margin-bottom:5em;padding-top:1em;padding-bottom:3em;color:#fff;border-radius:10px;-webkit-box-shadow: 2px 2px 15px 0px rgba(0, 3, 0, 0.7);
-moz-box-shadow:    2px 2px 15px 0px rgba(0, 3, 0, 0.7);
box-shadow:         2px 2px 15px 0px rgba(0, 3, 0, 0.7);">
    <div class="col-sm-12">
       <a href="{% url 'logout' %}"><h5 class="text-right"><i class="fa fa-user-circle" aria-hidden="true"></i> Logout</h5></a>
      <h1 class="text-center section-title" style="margin-bottom:2em">Welcome, {{user.username}}</h1>
    </div>
     <style>
      h4{
        margin-bottom: 1.5em;
        padding-top: 30px;

      }
      img{
        border-radius:50%;
        -webkit-transition: all 0.3s ease-in-out;
    -moz-transition: all 0.3s ease-in-out;
     transition: all 0.3s ease-in-out;
      }
      img:hover{
        -webkit-box-shadow: 2px 2px 21px 0px rgba(0, 3, 0, 0.91);
        -moz-box-shadow:    2px 2px 21px 0px rgba(0, 3, 0, 0.91);
        box-shadow:         2px 2px 21px 0px rgba(0, 3, 0, 0.91);
        border:2px solid #fff;
      }
      h3{
        margin-bottom: 1.3em;
      }
      a{
        color:inherit
      }
      a:hover{
        color:inherit;
        text-decoration: none;
      }
      /*
      .section-title:after {
          content:' ';display:block;margin:0 auto;width:100px;margin-top:6px;border:2px solid #d0d0d0;border-radius:4px;
          -webkit-border-radius:4px;
          -moz-border-radius:4px;
          box-shadow:inset 0 1px 1px rgba(0, 0, 0, .05);
          -webkit-box-shadow:inset 0 1px 1px rgba(0, 0, 0, .05);
     -moz-box-shadow:inset 0 1px 1px rgba(0, 0, 0, .05);
     margin-bottom:1em;
    }
    */

  </style>




   <div class="row" style=" display:flex; float:none; margin-left:10px">
     <div class="col-md-3">

     </div>
     <div class="col-md-3">
     <a href="{%url 'add-photos'%}"><img src="{% static
'recognition/img/addphotos.png' %}" class="img-responsive"
style="width:300px;height:300px ;" /></a>
    </div>
    <div class="col-md-3">
     <a href="{%url 'view-my-attendance-employee-login' %}"><img src="{% static
'recognition/img/reports.png' %}" class="img-responsive"
style="width:300px;height:300px"/></a>
    </div>

     <div class="col-md-3">

     </div>

<!--       <div class="col-md-4">-->
<!--       <a href="{%url 'train' %}" id="train" ><img src="{% static
'recognition/img/train.jpeg' %}" class="img-responsive"
style="width:300px;height:300px ;" /></a>-->
<!--      </div>-->



   </div>




<!--     </div>-->
 <div class="row" style=" display:flex; float:none; margin-left:90px">
    <div class="col-md-3">

     </div>
    <div class="col md-3" >
       <h4 >Upload Photo</h4>
    </div>
<!--     <div class="col md-6" >-->
<!--          <h4 >View My Attendance</h4>-->
<!--      </div>-->
```

```
        <div class="col md-3" >
             <h4 >Travel and Payment Info</h4>
        </div>
      <div class="col-md-3">

          </div>
</div>

 {% if messages %}
      {% for message in messages%}
      <div class="alert alert-{{message.tags}}" > {{message}}
      </div>
      {%endfor %}

    {%endif%}


</div>




<!-- Optional JavaScript -->
    <!-- jQuery first, then Popper.js, then Bootstrap JS -->
    <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-
KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
crossorigin="anonymous"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
crossorigin="anonymous"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl"
crossorigin="anonymous"></script>
</body>
</html>
```

Home.html

```
{% load static %}


<!DOCTYPE html>
<html>

<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
crossorigin="anonymous">
```

```
    <style>
        body {
            background: url('{% static "recognition/img/bg_image2.png"%}') no-repeat
center center fixed;
            background-size: cover;
        }
    </style>




</head>

<body>

    <div class="col-lg-12" style="background: rgba(0,0,0,0.6);margin-top:3em;margin-
bottom:5em;padding-top:1em;padding-bottom:3em;color:#fff;border-radius:10px;-webkit-
box-shadow: 2px 2px 15px 0px rgba(0, 3, 0, 0.7);
-moz-box-shadow:    2px 2px 15px 0px rgba(0, 3, 0, 0.7);
box-shadow:         2px 2px 15px 0px rgba(0, 3, 0, 0.7);">
        <div class="col-sm-12">

            <h1 class="text-center section-title" style="margin-bottom:2em">FPay As
You Go</h1>
        </div>
        <style>
            h4 {
                margin-bottom: 1.5em;
                padding-top: 30px;
            }

           img {
                border-radius: 50%;
                -webkit-transition: all 0.3s ease-in-out;
                -moz-transition: all 0.3s ease-in-out;
                transition: all 0.3s ease-in-out;
            }

           img:hover {
                -webkit-box-shadow: 2px 2px 21px 0px rgba(0, 3, 0, 0.91);
                -moz-box-shadow: 2px 2px 21px 0px rgba(0, 3, 0, 0.91);
                box-shadow: 2px 2px 21px 0px rgba(0, 3, 0, 0.91);
                border: 2px solid #fff;
            }

           h3 {
                margin-bottom: 1.3em;
            }

           a {
                color: inherit
            }

           a:hover {
                color: inherit;
        text-decoration: none;
      }
      /*
   .section-title:after {
      content:' ';display:block;margin:0 auto;width:100px;margin-top:
6px;border:2px solid #d0d0d0;border-radius:4px;
      -webkit-border-radius:4px;
      -moz-border-radius:4px;
      box-shadow:inset 0 1px 1px rgba(0, 0, 0, .05);
      -webkit-box-shadow:inset 0 1px 1px rgba(0, 0, 0, .05);
      -moz-box-shadow:inset 0 1px 1px rgba(0, 0, 0, .05);
     margin-bottom:1em;
    }
    */
    </style>

    <div class="col-lg-12" style="padding-top: 100px;">
      {% if messages %} {% for message in messages%}
      <div class="alert alert-{{message.tags}}"> {{message}}
      </div>
      {%endfor %} {%endif%}
    </div>


    <div class="row">


      <div class="col-md-4" style="padding-left: 220px">
        <a href='{%url "mark-your-attendance" %}'><img src="{% static
'recognition/img/driver1.png' %}" class="img-responsive"
style="width:300px;height:300px ;" /></a>

      </div>

<!--        <div class="col-md-4" style="padding-left: 220px">-->
<!--          <a href='{%url "mark-your-attendance-out" %}'><img src="{% static
'recognition/img/exit.png' %}" class="img-responsive" style="width:300px;height:300px
;" /></a>-->

<!--        </div>-->

      <div class="col-md-4" style="padding-left: 220px">
        <a href='{%url "login" %}'><img src="{% static
'recognition/img/passenger1.png' %}" class="img-responsive"
style="width:300px;height:300px ;" /></a>

      </div>

      <div class="col-md-4" style="padding-left: 220px">
        <a href='{%url "login" %}'><img src="{% static
'recognition/img/admin3.png' %}" class="img-responsive"
style="width:300px;height:300px" /></a>

<!--        <div class="col-md-4" style="padding-left: 220px">-->
<!--          <a href='{%url "login" %}'><img src="{% static
```

```
'recognition/img/login.png' %}" class="img-responsive"
style="width:300px;height:300px" /></a>-->


            </div>




        </div>
        <div class="row">


            <div class="col md-3" style="padding-left: 70px">
                <h4 class="text-center ">Driver Login</h4>
            </div>
<!--            <div class="col md-3" style="padding-left: 70px">-->
<!--                <h4 class="text-center ">Mark Your Attendance - Out </h4>-->
<!--            </div>-->


            <div class="col md-3" style="padding-left: 70px">
                <h4 class="text-center ">Passenger Log In </h4>
            </div>
            <div class="col md-3" style="padding-left: 70px">
                <h4 class="text-center ">Admin Log In </h4>
            </div>


        </div>

    </div>










    <!-- Optional JavaScript -->
    <!-- jQuery first, then Popper.js, then Bootstrap JS -->
    <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-
KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
crossorigin="anonymous"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
crossorigin="anonymous"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl"
crossorigin="anonymous"></script>
</body>
```

```
</html>
```

Index.html

```
{% load static %}
{% block content %}



{% if not payment %}
<div id="app">
<div class="col-6 pt-5  mx-auto">
<div class="text-center">
<img src="{% static 'images/d.png' %}"
style="height:200px" class="img-fluid img-reponsive text-center">
<br>
<h4>Help Rebuild lives of Children</h4>
</div>

    <form class="m" method="post" >
        {% csrf_token %}
        <div class="form-group pt-4">
            <label for="exampleInputEmail1">Your name</label>
            <input  v-model="name" name="name" type="text" class="form-control"
             id="exampleInputEmail1" aria-describedby="emailHelp">

         </div>
        <div class="form-group">
          <label for="exampleInputPassword1">Enter Email</label>
          <input type="email"  class="form-control" name="email"
           id="">
        </div>

        <div class="form-group">
          <label for="exampleInputPassword1">Enter Amount</label>
          <input type="number"  class="form-control" name="amount"
           id="exampleInputPassword1">

       </div>

        <button class="btn btn-success btn-block" type="submit">Submit</button>
        </form>
{% endif %}

{% if payment %}
        <div class="container mx-auto text-center mt-4" v-if="name && amount">

            <form action="{% url 'success' %}" method="POST">
                <script
                    src="https://checkout.razorpay.com/v1/checkout.js"
                    data-key="rzp_test_ifqXZb84qSL1CP"
                    data-amount="{{payment.amount}}"
                    data-currency="INR"
                    data-order_id="{{ payment.id }}"
                    data-buttontext="Pay with Razorpay"
                    data-name="Donate Corp"
                    data-description="Thanks for donating money."
                    data-image="https://example.com/your_logo.jpg"
                    data-prefill.name="Manik Sharma"
                    data-prefill.email="5@gmil.com"
                    data-prefill.contact="333333333"
                    data-theme.color="#F37254"
                >
                </script>
                <input type="hidden" custom="Hidden Element" name="hidden">
                </form>
            </div>
{% endif %}
       </div>




</div>



   <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>

    <script>
        var app = new Vue({
          delimiters: ['[[', ']]'],
          el: '#app',
          data: {
              message: 'Hello Vue!',
              name : null,
              amount : null

         },
          methods: {
              greet: function(name) {
                  console.log('Hello from ' + name + '!')
              },
              get : function(){
                console.log(this.name , this.amount)
              }
          },
          watch : {
            amount(){
                console.log(this.amount)
            }
          }
        });
      </script>


<style>
body{
    background :#f5f6f8;
}
```

```
</style>
{% endblock %}
```

Not_authorised.html

```
{% load static %}


<!DOCTYPE html>
<html>
<head>
     <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJ1SAwiGgFAW/dAiS6JXm" crossorigin="anonymous">

   <style>
    body{
      background: url('{% static "recognition/img/bg_image2.png"%}') no-repeat center center fixed;
      background-size: cover;

   }

  </style>




</head>
<body >
<div class="col-lg-12" style="background: rgba(0,0,0,0.6);max-height: 20px ; padding-top:1em;padding-bottom:3em;color:#fff;border-radius:10px;-webkit-box-shadow: 2px 2px 15px 0px rgba(0, 3, 0, 0.7);
-moz-box-shadow:    2px 2px 15px 0px rgba(0, 3, 0, 0.7);
box-shadow:         2px 2px 15px 0px rgba(0, 3, 0, 0.7); margin-left:auto; margin-right: auto; ">

  <a href="{% url 'dashboard' %}"><h5 class="text-left">  Dashboard</h5></a>
</div>

<div class="alert alert-danger" role="alert" style=" margin-top: 50px;  max-height: 50px;">
  <h3>Sorry, {{user.username}}. You are not authorised to view this page.</h3>
</div>
```

```
<!-- Optional JavaScript -->
    <!-- jQuery first, then Popper.js, then Bootstrap JS -->
    <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-
KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
crossorigin="anonymous"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
crossorigin="anonymous"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl"
crossorigin="anonymous"></script>
  </body>
  </html>
```

Success.html

```
<html></html>

<div class="container mx-auto mt-5 pt-4 text-center">

    <img src="https://encrypted-
tbn0.gstatic.com/images?q=tbn%3AANd9GcTgyobPWtA8sK4FUdJ7v2mVN1k1XYUwsy1q8A&usqp=CAU">


   <h2 class="mt-3 pt-4">Your payment has been received!!</h2>



</div>
{% endblock %}
```

Train.html

```
{% load static %}


<!DOCTYPE html>
<html>
<head>
     <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
crossorigin="anonymous">

   <style>
    body{
      background: url('{% static "recognition/img/bg_image2.png"%}') no-repeat center
center fixed;
      background-size: cover;
   }

  </style>



</head>
<body>

<div class="col-lg-12" style="background: rgba(0,0,0,0.6);max-height: 20px ; padding-
top:1em;padding-bottom:3em;color:#fff;border-radius:10px;-webkit-box-shadow: 2px 2px
15px 0px rgba(0, 3, 0, 0.7);
-moz-box-shadow:    2px 2px 15px 0px rgba(0, 3, 0, 0.7);
box-shadow:         2px 2px 15px 0px rgba(0, 3, 0, 0.7); margin-left:auto; margin-
right: auto; ">

  <a href="{% url 'dashboard' %}"><h5 class="text-left">  Back</h5></a>
</div>

 <div class="col-lg-4" style="background: rgba(0,0,0,0.6);margin-top:150px ; padding-
top:1em;padding-bottom:3em;color:#fff;border-radius:10px;-webkit-box-shadow: 2px 2px
15px 0px rgba(0, 3, 0, 0.7);
-moz-box-shadow:    2px 2px 15px 0px rgba(0, 3, 0, 0.7);
box-shadow:         2px 2px 15px 0px rgba(0, 3, 0, 0.7); margin-left:auto; margin-
right: auto; ">




<img src="{% static 'recognition/img/training_visualisation.png' %}"
style="width:900px; margin-right: auto; margin-left: auto;"/>
</div>
<div class="col-lg-12" style="padding-top: 100px;">
 {% if messages %}
      {% for message in messages%}
      <div class="alert alert-{{message.tags}}" > {{message}}
      </div>
      {%endfor %}

    {%endif%}
  </div>



  <!-- Optional JavaScript -->
    <!-- jQuery first, then Popper.js, then Bootstrap JS -->
    <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-
KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
crossorigin="anonymous"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
crossorigin="anonymous"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl"
```

```
crossorigin="anonymous"></script>
</body>
</html>
```

View_attendance_date.html

```
{% load static %}
 {% load crispy_forms_tags %}






<!DOCTYPE html>
<html>
<head>    <title></title>
      <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
crossorigin="anonymous">

</head>
<body>

  <nav class="navbar navbar-expand-lg navbar-light bg-light">
  <a class="navbar-brand" href="{%url 'view-attendance-home' %}">Passenger Details</a>
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse" id="navbarNav">
    <ul class="navbar-nav">

     <li class="nav-item active">
        <a class="nav-link" href="{%url 'view-attendance-employee' %}">By
Passenger</a>
      </li>
        <li class="nav-item active">
        <a class="nav-link" href="{% url 'view-attendance-date' %}">By Date</a>
      </li>




         <li class="nav-item active"">
        <a class="nav-link" href="{% url 'dashboard' %}">Back to Admin Panel</a>
      </li>

   </ul>
  </div>
</nav>



<div class="container">
  <div style="width: 400px">
 <form method="POST" >
      {% csrf_token %}
      <fieldset class="form-group">
        <legend class="border-bottom mb-4"> Select Date </legend>
        {{form| crispy}}
      </fieldset>

      <div class="form-group">
        <button class="btn btn-outline-info" type="submit" value="Create">
Submit</button>
      </div>
    </form>



</div>
{%if qs%}
<table class="table"  style="margin-top: 5em;">
    <thead class="thead-dark">
    <tr>
        <th scope="col">Date</th>

        <th scope="col" >Passenger</th>
        <th scope="col">Payment Due</th>
        <th scope="col">Time in</th>
<!--        <th scope="col">Time out </th>-->
<!--         <th scope="col">Hours </th>-->
<!--         <th scope="col"> Break Hours </th>-->
    </tr>
</thead>
<tbody>
    {% for item in qs %}
    <tr>
           <td>{{ item.date }}</td>
        <td>{{ item.user.username}}</td>

         {% if item.present %}
        <td> Rs. 15 </td>
        {% else %}
        <td> Rs. 0 </td>
        {% endif %}
        {% if item.time_in %}
        <td>{{ item.time_in }}</td>
       {% else %}
       <td> - </td>
       {% endif %}
<!--         {% if item.time_out %}-->
<!--        <td>{{ item.time_out }}</td>-->
<!--       {% else %}-->
<!--       <td> - </td>-->
<!--       {% endif %}-->
```

```
<!--       <td> {{item.hours}}</td>-->
<!--          <td> {{item.break_hours}}</td>-->
    </tr>
    {% endfor %}
</tbody>
</table>

<!-- <div class="card" style=" margin-top: 5em; margin-bottom: 10em;">-->
<!--  <img class="card-img-top" src="{% static
'recognition/img/attendance_graphs/employee_login/1.png'%}" alt="Card image cap">-->
<!--  <div class="card-body">-->
<!--    <p class="card-text" style="text-align: center;">Number of hours worked each
day.</p>-->
<!--  </div>-->
<!--</div>-->


{% endif %}



 {% if messages %}
      {% for message in messages%}
      <div class="alert alert-{{message.tags}}" > {{message}}
      </div>
      {%endfor %}

    {%endif%}


</div>



  <!-- Optional JavaScript -->
    <!-- jQuery first, then Popper.js, then Bootstrap JS -->
    <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-
KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
crossorigin="anonymous"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
crossorigin="anonymous"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl"
crossorigin="anonymous"></script>
</body>
</html>
```

View_Attendance_Employee.html

```
{% load static %}
 {% load crispy_forms_tags %}
```

```
<!DOCTYPE html>
<html>
<head>
     <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
crossorigin="anonymous">



</head>
 <body>

   <nav class="navbar navbar-expand-lg navbar-light bg-light">
  <a class="navbar-brand" href="{%url 'view-attendance-home' %}">Passenger Details</a>
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse" id="navbarNav">
    <ul class="navbar-nav">

     <li class="nav-item active">
        <a class="nav-link" href="{%url 'view-attendance-employee' %}">By
Passenger</a>
      </li>
        <li class="nav-item active">
        <a class="nav-link" href="{% url 'view-attendance-date' %}">By Date</a>
      </li>




         <li class="nav-item active">
        <a class="nav-link" href="{% url 'dashboard' %}">Back to Admin Panel</a>
      </li>

   </ul>
  </div>
</nav>



<div class="container">
  <div style="width:400px;">

 <form method="POST" >
      {% csrf_token %}
      <fieldset class="form-group">
        <legend class="border-bottom mb-4"> Select Username And Duration </legend>
        {{form| crispy}}
      </fieldset>

      <div class="form-group">
        <button class="btn btn-outline-info" type="submit"> Submit</button>
      </div>
    </form>

</div>

{%if qs%}
<table class="table"  style="margin-top: 5em;">
    <thead class="thead-dark">
    <tr>
        <th scope="col">Date</th>
        <th scope="col" >Passenger</th>
        <th scope="col">Ride Fare</th>
        <th scope="col">Time in</th>
<!--        <th scope="col">Time out </th>-->
<!--         <th scope="col">Hours </th>-->
<!--          <th scope="col"> Break Hours </th>-->
    </tr>
</thead>
<tbody>
    {% for item in qs %}
    <tr>
           <td>{{ item.date }}</td>
        <td>{{ item.user.username}}</td>

         {% if item.present %}
        <td> Rs. 15 </td>
        {% else %}
        <td> Rs. 0 </td>
        {% endif %}
        {% if item.time_in %}
        <td>{{ item.time_in }}</td>
       {% else %}
       <td> - </td>
       {% endif %}
<!--         {% if item.time_out %}-->
<!--        <td>{{ item.time_out }}</td>-->
<!--       {% else %}-->
<!--       <td> - </td>-->
<!--       {% endif %}-->
<!--       <td> {{item.hours}}</td>-->
<!--          <td> {{item.break_hours}}</td>-->
    </tr>
    {% endfor %}
</tbody>
</table>

<!-- <div class="card" style=" margin-top: 5em; margin-bottom: 10em;">-->
<!--  <img class="card-img-top" src="{% static
'recognition/img/attendance_graphs/hours_vs_date/1.png'%}" alt="Card image cap">-->
<!--  <div class="card-body">-->
```

```
<!--    <p class="card-text" style="text-align: center;">Number of hours worked each
day.</p>-->
<!--  </div>-->
<!--</div>-->

{% endif %}


 {% if messages %}
      {% for message in messages%}
      <div class="alert alert-{{message.tags}}" > {{message}}
      </div>
      {%endfor %}

    {%endif%}

</div>

  <!-- Optional JavaScript -->
    <!-- jQuery first, then Popper.js, then Bootstrap JS -->
    <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-
KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
crossorigin="anonymous"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
crossorigin="anonymous"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl"
crossorigin="anonymous"></script>





</body>
 </html>
```

View_attendance_home.html

```
{% load static %}


<!DOCTYPE html>
<html>
<head>
     <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
crossorigin="anonymous">
```

```
</head>
 <body>


  <nav class="navbar navbar-expand-lg navbar-light bg-light">
  <a class="navbar-brand" href="{%url 'view-attendance-home' %}">Admin View</a>
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse" id="navbarNav">
    <ul class="navbar-nav">

     <li class="nav-item active">
        <a class="nav-link" href="{%url 'view-attendance-employee' %}">By
Passenger</a>
      </li>
        <li class="nav-item active">
        <a class="nav-link" href="{% url 'view-attendance-date' %}">By Date</a>
      </li>




         <li class="nav-item active">
        <a class="nav-link" href="{% url 'dashboard' %}">Back to Admin Panel</a>
      </li>

   </ul>
  </div>
</nav>






  <div class="card" style="margin-top: 2em; margin-left: 2em; margin-right: 2em;
margin-bottom: 2em;">
    <div class="card-body">



<h2> Today's Statistics </h2>
   <div class="row" style="margin-left: 12em">
<div class="card" style="width: 20em; background-color: #338044; text-align : center;
margin-left: 5em; margin-top: 5em; color: white;">
  <div class="card-body">
    <h5 class="card-title"> <b>Total Number Of Registered Passengers</b></h5>

   <p class="card-text" style="padding-top: 1em; font-size: 28px;">
<b>{{total_num_of_emp }}</b></p>

 </div>
```

```
</div>
<div class="card" style="width: 20em; background-color: #80335b; text-align : center;
margin-left: 5em; margin-top: 5em; color: white;">
  <div class="card-body">
    <h5 class="card-title"> <b> Number of Riders today</b></h5>

   <p class="card-text" style="padding-top: 1em; font-size: 28px;"><b>
{{emp_present_today }}</b></p>

 </div>
</div>

</div>



</div>
</div>











 <div class="card" style="margin-top: 2em; margin-left: 2em; margin-right: 2em;
margin-bottom: 2em;">
    <div class="card-body">




<!--  <div class="row" >-->
<!--<div class="col-md-6">-->
<!--<h2> Last Week </h2>-->
<!--    <div class="card" style="width: 50em;">-->
<!--  <img class="card-img-top" src="{% static
'recognition/img/attendance_graphs/last_week/1.png'%}" alt="Card image cap">-->
<!--  <div class="card-body">-->
<!--    <p class="card-text" style="text-align: center;">Number of employees present
each day</p>-->
<!--  </div>-->
<!--</div>-->







<!--</div>-->
<!--<div class="col-md-6">-->
<!--  <h2> This Week </h2>-->
<!-- <div class="card" style="width: 50em;">-->
<!--  <img class="card-img-top" src="{% static
'recognition/img/attendance_graphs/this_week/1.png'%}" alt="Card image cap">-->
```

```
<!--  <div class="card-body">-->
<!--    <p class="card-text" style="text-align: center;">Number of employees present 
each day</p>-->
<!--  </div>-->
<!--</div>-->


</div>
</div>
</div>
</div>



  <!-- Optional JavaScript -->
    <!-- jQuery first, then Popper.js, then Bootstrap JS -->
    <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-
KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN" 
crossorigin="anonymous"></script>
    <script 
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js" 
integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q" 
crossorigin="anonymous"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" 
integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl" 
crossorigin="anonymous"></script>

 </body>
 </html>
```

View_my_attendance_employee_login.html

```
{% load static %}
 {% load crispy_forms_tags %}


<!DOCTYPE html>
<html>
<head>
     <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet" 
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" 
integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJ1SAwiGgFAW/dAiS6JXm" 
crossorigin="anonymous">
    <style>
        .check{
   background: none;
   color: inherit;
   border: none;
   padding: 0;
   font: inherit;
   cursor: pointer;
   outline: inherit;
}

    </style>



</head>
<body>

   <nav class="navbar navbar-expand-lg navbar-light bg-light">
  <a class="navbar-brand" href="{%url 'view-my-attendance-employee-login' %}">Travel
and Payment Info</a>
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse" id="navbarNav">
    <ul class="navbar-nav">





        <li class="nav-item active">
        <a class="nav-link" href="{% url 'dashboard' %}">Back to Dashboard</a>
      </li>

   </ul>
  </div>
</nav>



   <div class="container">
  <div style="width:400px;">

 <form method="POST" >
      {% csrf_token %}
      <fieldset class="form-group">
<!--          <h1 style="color:white">ajajajaaj</h1>-->
        <legend class="border-bottom mb-4"> Select Duration </legend>
        {{form| crispy}}
      </fieldset>

      <div class="form-group" style="display:flex">
        <button class="btn btn-outline-info" type="submit"> Submit</button>

      </div>
    </form>
    <button class="check"
style="background:white;color:white;width:100%;height:100px;border:none;outline:none"
onclick="funn()"></button>
</div>
```

```
    <script>
        function funn()
{
var x=document.getElementById('tt').rows[1].cells.namedItem("ps").innerHTML="Paid";
var x=document.getElementById('tt').rows[1].cells.namedItem("ps").style.color="green";
var x=document.getElementById('tt').rows[2].cells.namedItem("ps").innerHTML="Paid";
var x=document.getElementById('tt').rows[3].cells.namedItem("ps").innerHTML="Paid";
var x=document.getElementById('tt').rows[4].cells.namedItem("ps").innerHTML="Paid";
var x=document.getElementById('tt').rows[2].cells.namedItem("ps").style.color="green";
var x=document.getElementById('tt').rows[3].cells.namedItem("ps").style.color="green";

var x=document.getElementById('tt').rows[4].cells.namedItem("ps").style.color="green";
}
    </script>
{%if qs%}
<table id ="tt" class="table"  style="margin-top: 5em;">
    <thead class="thead-dark">
    <tr>
        <th scope="col">Date</th>

       <th scope="col"  >Passenger</th>
        <th scope="col">Ride Fair</th>
        <th scope="col">Day in</th>
        <th scope="col">Time in</th>
<!--        <th scope="col">Fine</th>-->
        <th scope="col">Amount Due</th>
        <th scope="col">Payment Status</th>

<!--        <th scope="col">Time out </th>-->
<!--         <th scope="col">Hours </th>-->
<!--         <th scope="col"> Break Hours </th>-->
    </tr>
</thead>

<tbody>
<!--    <% count = 0 %>-->
    {% for item in qs %}
    <tr>
<!--        <% count+=1 %>-->
           <td>{{ item.date }}</td>
        <td>{{ item.user.username}}</td>

        {% if item.present %}
        <td> Rs. 15 </td>
        {% else %}
        <td> Rs. 15 </td>
        {% endif %}
        {% if item.date %}
        <td>{{ item.date }}</td>
       {% else %}
       <td> - </td>
       {% endif %}



        {% if item.time_out %}
        <td>{{ item.time_out }}</td>
       {% else %}
       <td> {{ item.time_in }} </td>
       {% endif %}

         {% if item.total_due %}
        <td>{{ item.total_due }}</td>
       {% else %}
       <td> - </td>
       {% endif %}
        <td id="ps">
<!--        <!<div class="sta">-->
            <p id="stat" style="color:Red; display:block">Unpaid</p>
            <p id="stat1" style="color:Green; display:none">Paid</p>
<!--        </div>-->
        </td>

<!--       <td> {{item.hours}}</td>-->
<!--          <td> {{item.break_hours}}</td>-->
    </tr>
    {% endfor %}
</tbody>
</table>
<!--<table id="stat3">-->
<!--    <th>Payment Status</th>-->
<!--    <tr>Paid</tr>-->
<!--    <tr>Paid</tr>-->
<!--    <tr>Paid</tr>-->
<!--</table>-->

<!-- <div class="card" style=" margin-top: 5em; margin-bottom: 10em;">-->
<!--  <img class="card-img-top" src="{% static
'recognition/img/attendance_graphs/employee_login/1.png'%}" alt="Card image cap">-->
<!--  <div class="card-body">-->
<!--    <p class="card-text" style="text-align: center;">Number of hours worked each
day.</p>-->
<!--  </div>-->
<!--</div>-->



{% endif %}




{% if not payment %}
<div id="app">
<div class="col-6 pt-5  mx-auto">
<div class="text-center">
<br>
<h4>Welcome to the Payment Gateway</h4>
    <p style="text-align:center">Please enter your full name and click on pay now to
proceed with the payment</p>
</div>

    <form class="m" action="{% url 'pg' %}" method="POST" >
        {% csrf_token %}
        <div class="form-group pt-4">
            <label for="exampleInputEmail1">Enter Your Name</label>
            <input  v-model="name" name="name" type="text" class="form-control"
             id="exampleInputEmail1" aria-describedby="emailHelp">

          </div>
<!--        <div class="form-group">-->
<!--          <label for="exampleInputPassword1">Enter Your Email</label>-->
<!--          <input type="email"  class="form-control" name="email"-->
<!--           id="">-->
<!--        </div>-->

        <div class="form-group">

<!--          <label for="exampleInputPassword1">Enter Amount</label>-->

<!--          <input type="number"  class="form-control" name="amount" value="<%
(count-1)*15+35 %>"-->
<!--           id="exampleInputPassword1">-->

        </div>

        <button class="btn btn-success btn-block"  type="submit">Pay Now</button>

        </form>

{% endif %}

{% if payment %}
        <div class="container mx-auto text-center mt-4" v-if="name && amount">

            <form action="{% url 'success' %}" method="POST">
                <script
                    src="https://checkout.razorpay.com/v1/checkout.js"
                    data-key="rzp_test_ifqXZb84qSL1CP"
                    data-amount="{{payment.amount}}"
                    data-currency="INR"
                    data-order_id="{{ payment.id }}"
                    data-buttontext="Pay with Razorpay"
                    data-name="Donate Corp"
                    data-description="Thank You."
                    data-image="https://example.com/your_logo.jpg"
                    data-prefill.name="Fpay As You Go"
                    data-prefill.email="a@gmail.com"
                    data-prefill.contact="919211302420"
                    data-theme.color="#F37254"
                ></script>
                <input type="hidden" custom="Hidden Element" name="hidden">
                </form>
            </div>
{% endif %}
        </div>

</div>
```

```html
</div>

  <!-- Optional JavaScript -->
    <!-- jQuery first, then Popper.js, then Bootstrap JS -->
    <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
crossorigin="anonymous"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
crossorigin="anonymous"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl"
crossorigin="anonymous"></script>



   </body>
</html>
```

Forms.py

```python
from django.forms import ModelForm
from django.contrib.auth.models import User
from django import forms
#from django.contrib.admin.widgets import AdminDateWidget

class usernameForm(forms.Form):
   username=forms.CharField(max_length=30)




class DateForm(forms.Form):
   date=forms.DateField(widget = forms.SelectDateWidget(empty_label=("Choose Year",
"Choose Month", "Choose Day")))



class UsernameAndDateForm(forms.Form):
   username=forms.CharField(max_length=30)
   date_from=forms.DateField(widget = forms.SelectDateWidget(empty_label=("Choose
Year", "Choose Month", "Choose Day")))
   date_to=forms.DateField(widget = forms.SelectDateWidget(empty_label=("Choose Year",
"Choose Month", "Choose Day")))



class DateForm_2(forms.Form):
   date_from=forms.DateField(widget = forms.SelectDateWidget(empty_label=("Choose
Year", "Choose Month", "Choose Day")))
   date_to=forms.DateField(widget = forms.SelectDateWidget(empty_label=("Choose Year",
"Choose Month", "Choose Day")))
```

Chapter 9: Conclusion and Future Enhancements

- Making use of the HOG SVM classifier, we developed an efficient Facial recognition-based payment
- system for the shuttle services.
- The application accurately identifies the passenger, adds the travel fare to their account, and maintains all passenger travel records which can be accessed by the admin.
- It has an integrated payment gateway (RazorPay) to provide an easy-to-use payment interface.
- An efficient solution was provided for the problem stated.

Our application can be enhanced and refined in the future as well, some of the future enhancements are listed below:

- Make a mobile app, as they have better accessibility and thus would attract more customers.
- Improve the processing speed, the main aim is to reduce the time required for training the model while making sure to not affect its accuracy
- Multiple payment gateways can be integrated for user convenience.
- This project can utilize VIT database to ensure valid database entries.