**Face Recognition Based Attendance Management System**



Session 2022-2026

**SUBMITTED BY:**

Hadia Moosa 2022-CS-119

Momina Rashad 2022-CS-137

Shahneela Iqbal 2022-CS-159

**SUPERVISED BY :**

Sir Samyan Wahla

DEPARTMENT OF COMPUTER SCIENCE

**UNIVERSITY OF ENGINEERING AND TECHNOLOGY , Lahor**

Table of Contents

[1. Abstract 7](#_Toc185196187)

[2. Introduction 7](#_Toc185196188)

[2.1 Problem Statement 7](#_Toc185196189)

[2.2 Objective 8](#_Toc185196190)

[2.2 Scope 8](#_Toc185196191)

[3. Project Description 9](#_Toc185196192)

[3.1 Project Methodology 9](#_Toc185196193)

[3.2 Project Rationale 9](#_Toc185196194)

[3.2.1 Operational Inefficiencies 9](#_Toc185196195)

[3.2.2 Security and Fraud Prevention 9](#_Toc185196196)

[3.2.3 Adapting to User Needs 9](#_Toc185196197)

[3.2.4 Scalability and Centralized Management 9](#_Toc185196198)

[3.2.5 Compliance with Post-Pandemic Standards 10](#_Toc185196199)

[3.3 Product Features 10](#_Toc185196200)

[3.3.1 Face Detection 10](#_Toc185196201)

[3.3.2 Face Recognition 10](#_Toc185196202)

[3.3.3 Attendance Logging 10](#_Toc185196203)

[3.3.4 Admin Dashboard 11](#_Toc185196204)

[3.3.5 User Management 11](#_Toc185196205)

[4. Technology stack 11](#_Toc185196206)

[5. Constraints 12](#_Toc185196207)

[Camera Quality 12](#_Toc185196208)

[Processing Power 12](#_Toc185196209)

[Data Privacy Laws 12](#_Toc185196210)

[6. Use Cases 12](#_Toc185196211)

[6.1 Admin Login 12](#_Toc185196212)

[Primary Actor 12](#_Toc185196213)

[Preconditions 12](#_Toc185196214)

[Success Guarantee 12](#_Toc185196215)

[Main Success Scenario 12](#_Toc185196216)

[Extensions (Alternatives) 12](#_Toc185196217)

[6.1a Invalid credentials 12](#_Toc185196218)

[Trigger 12](#_Toc185196219)

[System Response: 13](#_Toc185196220)

[6.2 Admin Signup 13](#_Toc185196221)

[Primary Actor 13](#_Toc185196222)

[Preconditions 13](#_Toc185196223)

[Success Guarantee: 13](#_Toc185196224)

[Main Success Scenario 13](#_Toc185196225)

[Extensions: 13](#_Toc185196226)

[6.2c Data validation Failure 13](#_Toc185196227)

[Trigger 13](#_Toc185196228)

[System Response 13](#_Toc185196229)

[6.2c. Duplicate email 13](#_Toc185196230)

[Trigger: 13](#_Toc185196231)

[System Response: 14](#_Toc185196232)

[6.3 Register Student (By Admin) 14](#_Toc185196233)

[Primary Actor 14](#_Toc185196234)

[Preconditions 14](#_Toc185196235)

[Success Guarantee 14](#_Toc185196236)

[Main Success Scenario 14](#_Toc185196237)

[Extensions 14](#_Toc185196238)

[6.3e Data validation issues 14](#_Toc185196239)

[Trigger 14](#_Toc185196240)

[System Response 14](#_Toc185196241)

[6.4 Student Management (CRUD Operations) 14](#_Toc185196242)

[Primary Actor 15](#_Toc185196243)

[Main Success Scenario 15](#_Toc185196244)

[Trigger 15](#_Toc185196245)

[System response 15](#_Toc185196246)

[A. Update 15](#_Toc185196247)

[System response: 15](#_Toc185196248)

[B. Delete 15](#_Toc185196249)

[System response: 15](#_Toc185196250)

[C. View 15](#_Toc185196251)

[System response: 15](#_Toc185196252)

[Extensions 16](#_Toc185196253)

[6.4.2Ai., 6.4Bi , 6.4Ci : If a student is not found during search 16](#_Toc185196254)

[Trigger 16](#_Toc185196255)

[System Response 16](#_Toc185196256)

[6.4.2Aiii: Validation error 16](#_Toc185196257)

[Trigger 16](#_Toc185196258)

[System Response 16](#_Toc185196259)

[6.5 Generate Attendance Report (CSV Format) 16](#_Toc185196260)

[Primary Actor 16](#_Toc185196261)

[Main Success Scenario 16](#_Toc185196262)

[Extensions: 17](#_Toc185196263)

[6.5b. No attendance data record 17](#_Toc185196264)

[Trigger 17](#_Toc185196265)

[System Response 17](#_Toc185196266)

[6.6 Take Attendance with Face Detection 17](#_Toc185196267)

[Primary Actor 17](#_Toc185196268)

[Main Success Scenrio 17](#_Toc185196269)

[Extensions: 17](#_Toc185196270)

[6.6b. Face not recognized: 17](#_Toc185196271)

[Trigger 17](#_Toc185196272)

[System Response 17](#_Toc185196273)

[7. Functional Requirements 18](#_Toc185196274)

[7.1 User Authentication 18](#_Toc185196275)

[Sign Up 18](#_Toc185196276)

[Log In 18](#_Toc185196277)

[Log Out 18](#_Toc185196278)

[7.2 Face Recognition 18](#_Toc185196279)

[Face Registration 18](#_Toc185196280)

[Face Recognition for Attendance 19](#_Toc185196281)

[Face Detection with Mask and Glasses 19](#_Toc185196282)

[7.3 Attendance Management 19](#_Toc185196283)

[Automatically Attendance Marking 19](#_Toc185196284)

[Manual Attendance Entering (Admin) 19](#_Toc185196285)

[7.4 Reports and Dashboards 20](#_Toc185196286)

[View Attendance Reports (Admin) 20](#_Toc185196287)

[View Attendance Reports (User) 20](#_Toc185196288)

[7.5 Security Management 20](#_Toc185196289)

[Role Based Access Control 20](#_Toc185196290)

[8. System Architecture 21](#_Toc185196291)

[8.1 User Interface subsystem 21](#_Toc185196292)

[Student Management(CRUD) Subsystem 21](#_Toc185196293)

[Student Interface Subsystem 22](#_Toc185196294)

[9. Methadology 22](#_Toc185196295)

[9.1 Data Collection 22](#_Toc185196296)

[9.2 Implementaion 22](#_Toc185196297)

[Pre Processing 22](#_Toc185196298)

[9.2.1 Overfitting 22](#_Toc185196299)

[9.2.2 Under Sampling 23](#_Toc185196300)

[9.2.3 Over Sampling 23](#_Toc185196301)

[9.2.4 Confusion Matrix 23](#_Toc185196302)

[10. App Development 24](#_Toc185196303)

[10.1 UI/UX design 24](#_Toc185196304)

[10.2 Front-end development 24](#_Toc185196305)

[10.3 Back-end development 24](#_Toc185196306)

[10.4 Database design 25](#_Toc185196307)

[10. 5 API integration 25](#_Toc185196308)

[11. Implementation 26](#_Toc185196309)

[11.1 System Interface 26](#_Toc185196310)

[11.2 Backend Workflows 29](#_Toc185196311)

[11.3 Code Overview 29](#_Toc185196312)

[12. Limitations 33](#_Toc185196313)

[12.1 Failure to Account for Edge Cases 33](#_Toc185196314)

[9.2 Insufficient Error Reporting 33](#_Toc185196315)

[9.3 Ignoring Multi-Device Compatibility 33](#_Toc185196316)

[9.4 Underestimating the Impact of Poor Data Quality 33](#_Toc185196317)

[9. 5 Neglecting Real-World Environmental Factors 33](#_Toc185196318)

[13. Results and Discussion 33](#_Toc185196319)

[10.1 Summary of Results 33](#_Toc185196320)

[10.2 Real-world Applicability 34](#_Toc185196321)

[14. Conclusion 34](#_Toc185196322)

[15. Future Work 34](#_Toc185196323)

[16. References 35](#_Toc185196324)

**Table of Figures**

[Figure 1 Confusion Matrix 23](file:///F:\SEM%205\AI(theory)\final%20report.docx#_Toc185197421)

[Figure 2 Database Overvew 25](#_Toc185197422)

[Figure 3 System First Screen 26](file:///F:\SEM%205\AI(theory)\final%20report.docx#_Toc185197423)

[Figure 4 Sign Up Page 26](#_Toc185197424)

[Figure 5Sign In Page 26](#_Toc185197425)

[Figure 6 Admin Dashboard 27](#_Toc185197426)

[Figure 7 View Student List 27](#_Toc185197427)

[Figure 8 Register Student 27](#_Toc185197428)

[Figure 9 View Attendance Records 28](file:///F:\SEM%205\AI(theory)\final%20report.docx#_Toc185197429)

[Figure 10 User Dashboard 28](#_Toc185197430)

**Final Project Report**

# Abstract

This project presents an AI-powered Facial Recognition Attendance System to automate the attendance-taking process in institutions. Traditional attendance management systems often suffer from inefficiencies, errors, and challenges in accurately tracking attendance under varying conditions, such as the presence of masks, glasses, or facial expressions. These conventional methods, whether paper-based or biometric, are increasingly inadequate in meeting the demands for a more reliable, scalable, and adaptable solution. As organizations strive to enhance operational efficiency, the need for an advanced, automated system for attendance tracking has become paramount. This project introduces a Facial Recognition-Based Attendance System designed to address these limitations. By leveraging cutting-edge deep learning algorithms, including the OpenCV DNN module for real-time face detection and the face\_recognition library for encoding and identification, the system ensures highly accurate attendance logging. It effectively handles diverse conditions, such as mask-wearing and varying head orientations, offering a robust solution for real-world applications. The system integrates with a MySQL database to store and manage attendance data, ensuring secure, accessible, and reliable record-keeping. A user-friendly interface, developed using Tkinter, allows both administrators and users to seamlessly register and mark attendance. With its ability to operate efficiently in dynamic environments, this system offers a contactless and scalable solution that enhances accuracy and inclusivity. By catering to diverse user conditions, it ensures a broad application range, from educational institutions to corporate settings, fostering a more efficient and inclusive approach to attendance management.

# **Introduction**

2.1 Problem Statement

The purpose of this project is to design an efficient way of marking attendance through a face recognition mechanism as the .traditional attendance management systems often suffer from inefficiencies, errors whether paper-based or biometric, are increasingly inadequate in meeting the demands. One of the primary challenges is ensuring accurate facial recognition in scenarios where users are wearing masks or glasses, which poses a significant barrier to conventional systems. This issue is further exacerbated by the need to accommodate current health and safety protocols.

## 2.2 Objective

The objective of an this project is to automate the attendance-taking process, improving efficiency, accuracy, and user experience. By leveraging advanced AI and facial recognition technologies, the system aims to eliminate manual attendance logging, replacing it with a seamless, automated process. Real-time face detection allows individuals to be automatically identified as soon as they enter a monitored area, with attendance logged instantly. This ensures a quick, error-free process, enhancing reliability and minimizing delays. The system's accuracy is leverages through deep learning algorithms that ensure precise recognition under varying conditions. Additionally, the system should provide a user-friendly interface for both administrators and individuals, offering simple attendance tracking and access to reports. Security and privacy are also critical, with encrypted facial data storage and compliance with data protection regulations, ensuring user data is handled responsibly. To further enhance its utility, the system can generate detailed attendance reports. This AI-based solution not only saves time but also ensures that attendance management is more accurate, reliable, and secure while providing a hassle-free experience for all users.

## 2.2 Scope

The system is designed to be a versatile and scalable solution, suitable for a wide range of organizations, including educational institutions (such as schools, colleges, and universities), corporate offices, and healthcare facilities like hospitals. Its primary function is to automate the process of attendance tracking for both staff members and students, significantly reducing the time and effort required for manual attendance management.The application is built to be adaptable, meaning it can be seamlessly integrated into various organizational structures and workflows. It is capable of supporting multiple use cases and different types of users, such as administrators, teachers, students, and employees, with each user role having specific access and functionality. This customization allows the system to cater to the unique needs of each institution, whether it’s tracking student attendance in a classroom, employee presence in an office, or patient attendance in a hospital setting.Moreover, the system will provide flexibility with user role-based customization. Administrators can have full access to manage the entire system, including adding or removing users, monitoring attendance data, generating reports, while students or employees can view their attendance records and mark attendance. This role-based access enhances security and ensures that each user has appropriate access to the system based on their responsibilities.

# **Project Description**

## **3.1 Project Methodology**

The project will be divided into phases, with each including requirements gathering, design, implementation, testing, and deployment. Regular stakeholder meetings and feedback sessions will ensure that the business objectives and user expectations are aligned. Agile approaches, such as continuous integration and sprint-based development, will improve flexibility and responsiveness to changing needs.

## 3.2 Project Rationale

The decision to implement an AI-based Face Recognition Attendance System stems from the growing demand for efficient, secure, and automated attendance solutions. The project addresses critical operational challenges while offering scalable and adaptable technology to meet the diverse needs of various institutions. Key motivations include:

### 3.2.1 Operational Inefficiencies

Traditional attendance systems, whether manual or card-based, are prone to errors, time delays, and fraudulent practices such as proxy attendance. An AI-driven system utilizing face recognition eliminates these inefficiencies by providing a fast, reliable, and contactless method of attendance tracking.

### 3.2.2 Security and Fraud Prevention

Face recognition technology adds an additional layer of security by ensuring that attendance is marked only for authenticated individuals. This eliminates issues like buddy punching, thereby improving accuracy and accountability.

### 3.2.3 Adapting to User Needs

Modern organizations require attendance systems that are seamless and easy to use. By integrating AI with real-time face recognition, this system caters to the evolving needs of schools, colleges, universities, corporate offices, and hospitals. It also offers customizable features for different roles, ensuring a versatile and user-friendly experience.

### 3.2.4 Scalability and Centralized Management

The system is designed to handle operations across single or multiple branches, enabling centralized attendance record management. This scalability ensures that the system can grow with the organization’s needs, making it suitable for both small-scale and large-scale deployments.

### 3.2.5 Compliance with Post-Pandemic Standards

With the emphasis on contactless solutions in a post-pandemic world, the system offers a hygienic alternative to fingerprint-based or manual attendance systems.

## 3.3 Product Features

### 3.3.1 Face Detection

**Description**

Utilizes advanced deep learning algorithms, specifically OpenCV DNN, to detect faces in real-time video streams. This ensures that attendance is recorded quickly and accurately.

**Functionality**

* Detects faces in live video streams or uploaded images.
* Handles single faces at a time to maintain scalability and accuracy.

### 3.3.2 Face Recognition

**Description**

Implements robust recognition capabilities to identify users' faces with high precision. The system remains effective under challenging real-world conditions, including when users wear masks, glasses, or are slightly angled.

**Functionality**

* Matches detected faces against stored user profiles with high accuracy.
* Supports real-time recognition for seamless attendance marking.
* Accurately identifies users even when their faces are tilted up to 45 degrees left or right.

### 3.3.3 Attendance Logging

**Description**

Automatically logs attendance records into a .csv file format, providing a reliable and structured method for data storage.

**Functionality**

* Each entry includes a user ID, name, timestamp, and attendance count.
* Enables easy retrieval, analysis, and reporting of attendance data.
* Facilitates integration with external systems for extended use cases like payroll or academic performance tracking.

### 3.3.4 Admin Dashboard

**Description**

A comprehensive interface designed for administrators to manage user profiles, attendance records, and reports effectively.

**Functionality**

* Allows administrators to add, update, or remove user profiles.
* Provides access to detailed attendance logs.
* Offers intuitive tools for managing attendance across multiple locations or branches.

### 3.3.5 User Management

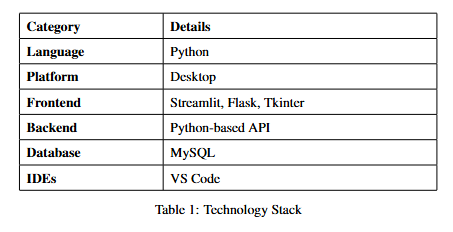
**Description**

Empowers users to mark their attendance independently and review their attendance records.

**Functionality**

* Users can mark attendance in real-time using facial recognition.
* Provides a personalized dashboard where users can view their attendance history, including dates, timestamps, and attendance count.
* Ensures data accuracy by linking attendance to unique user profiles.

# Technology stack



# Constraints

## ****Camera Quality****

The system will require a high-quality camera to ensure clear and accurate image capture, which is crucial for effective face detection and recognition.

## Processing Power

Deep learning models will be optimized to perform real-time detection, demanding a robust processing capability to maintain high performance during operation.

## Data Privacy Laws

The system will incorporate security measures to protect sensitive information related to admin and students, including encryption protocols and secure data storage practices. Ensuring the confidentiality and integrity of personal data will be a top priority.

# Use Cases

## Admin Login

Primary Actor: Admin

Preconditions: Admin must registered in the system and have valid login credentials.

Success Guarantee: Admin will be successfully log in and will have access to system’s functionalities of admin dashboard.

### Main Success Scenario

1. Admin opens the application and navigates to login page .
2. Then Enter credentials like username and password.
3. Then system validates the entered credentials.
4. If they are valid, system gives access and redirects Admin to dashboard.
5. System adds message “Logged In successfully”.

### Extensions (Alternatives)

### 6**.1a Invalid credentials**

Trigger:

* Admin enters incorrect username or password.

### System Response:

* System will display an error message that “Please Re-enter all credentials”
* Admin will re-enter correct credentials and try again.

## Admin Signup

Primary Actor: Admin

Preconditions: Admin have an access to create an account.

Success Guarantee: Admin successfully registered and now after this can login.

### Main Success Scenario

1. Admin navigates to signup page.
2. Admin enter required information like name, email , password.
3. System will check validations and create new account.
4. System will display message “Account created successfully”.
5. It will redirect you to login page.

### Extensions:

### 6.2c Data validation Failure

Trigger:

* Admin submits the form with missing or incorrect data (e.g., invalid email format, weak password).

System Response:

* The system will show failure in signing up due to missing or incorrect data (e.g email format , password strength , specific characters)
* Admin then correct the data and re-submit signup form.

### 6.2c. Duplicate email

### Trigger:

* Admin enters an email address that is already registered in the system.

### System Response:

* System shows an error message: "Email is already registered. Please use a different email."
* Admin enters a unique email and re-submits.

## Register Student (By Admin)

Primary Actor: Admin

Preconditions: Admin must be logged in.

Success Guarantee: Student is registered in the system.

Main Success Scenario

1. Admin selects the option “Register Student” from dashboard.
2. Admin fill the Student requirement details that student will tell.
3. Admin uploads the student’s face photo for future recognition.
4. System validates and stores the student’s details along with the face photo.
5. System confirms the successful registration of the student.

### Extensions

### 6.3e Data validation issues

Trigger

* Admin submits the form with missing, incorrect, or duplicate information (e.g., missing student ID, duplicate ID).

### System Response

* System shows an error message: "Please fill all required fields."
* If duplicate entry, system shows: "Student ID already exists."
* Admin corrects the information and re-submits the form.

## ****Student Management (CRUD Operations)****

Primary Actor: Admin  
Preconditions: Admin must be logged in.  
Success Guarantee: Admin can update, view, or delete student records.

### Main Success Scenario

1. **Admin navigates to the Student Management module**:

Trigger: Admin clicks on the "Student Management" button in the system dashboard.

System response: The system displays the available options: Update, View, and Delete.

1. Admin selects options

### Update

1. Admin selects the "Update" option and searches for a student by ID or name.

### System response:

1. The system retrieves the student's current details and displays them in an editable form.
2. Admin edits the fields that need to be updated (e.g., contact details, class, etc.).
3. Admin then clicks on "Save Changes".
4. The message will be displayed “Changes Saved Successfully!”

### Delete

1. Admin selects "Delete" and searches for a student by name or ID.

### System response:

1. The system retrieves the student’s details and asks for confirmation before deleting.
2. Admin confirms the deletion.
3. The system shows a message like "Student deleted successfully" and updates the list of students.

### View

1. Admin selects "View" and searches for a student by ID or name.

### System response:

1. The system displays the student's details (e.g., name, class, contact, face photo) in a read-only format.
2. Admin can navigate back to the list of students or perform another action like update or delete.

### Extensions

### 6**.4.2Ai.,** 6**.4Bi ,** 6**.4Ci** : If a student is not found during search

Trigger: Admin tries to search for a student by StudentId or name, but Id does not exist in system

System Response:

* The system shows an error message: "Student not found."
* Admin can re-enter the correct ID or search by name.

### 6**.4.2Aiii**: Validation error

Trigger: Admin tries to submit the form with missing or incorrect data (e.g., blank required fields, invalid student ID format).

### System Response

* The system highlights the invalid fields and displays error messages for each (e.g., "Student ID must be unique", "Name cannot be blank").
* Admin corrects the data and re-submits the form.

## Generate Attendance Report (CSV Format)

Primary Actor: Admin  
Preconditions: Admin must be logged in.  
Success Guarantee: Attendance report is generated and available for download.

### ****Main Success Scenario****

1. Admin navigates to the "Generate Attendance Report" module.
2. Admin selects the desired parameters (e.g class ).
3. System compiles the data into a CSV file.
4. Admin clicks "Download", and the file is saved on their local machine.

### ****Extensions****:

### 6****.5b. No attendance data record****

Trigger:

* Admin selects a class that contains no attendance data.

System Response:

* System shows a message: "No attendance data found for the selected class."
* Admin can modify the search criteria or exit the module.

## ****Take Attendance with Face Detection****

Primary Actor: Admin, System  
Preconditions: Students must be registered with a valid photo in the system.  
Success Guarantee: Attendance is recorded for each student based on face detection.

### Main Success Scenrio

1. Admin selects the "Take Attendance" module.
2. Students stand in front of the camera.
3. The system detects and identifies the student's face.
4. The system marks the student as present.
5. System stores the attendance data for future reporting.

### ****Extensions****:

### 6.****6b. Face not recognized:****

### ****Trigger****

* The system is unable to detect or match the student's face with the stored photo.

### ****System Response****

* System shows a message: "Face not recognized."
* Admin can either retry face detection or manually mark the student as present.

# Functional Requirements

## 7.1 User Authentication

### Sign Up

**Description**: User can create account by entering email, name, password and contact number.

**Input**:Username, email, contact number and password**.**

**Process**: System validates the inputs and check if this already exists or not and store the information in the database.

**Output**: A confirmation message if successful or error message if any information is incomplete or invalid.

### Log In

**Description**: User can log in to the system by entering their username and password.

**Input**: Username and password.

**Process**: System checks that if this user present in the database and then check the password with the password in the database.

**Output**: If successful the user will be redirected to its dashboard if not this shows the error message.

### Log Out

**Description**: User can log out from the system.

**Input**: Click on the exit button.

**Process**: System invalidate the current session and redirect them to the login page.

**Output**: A log out message will be shown.

## Face Recognition

### Face Registration

**Description**: User can register its bear and masked face by capturing images through camera attached with system.

**Input**: Images captured from camera.

**Process**: System extracts the features of face using algorithm like OpenCv DNN and store it in database.

**Output**: A confirmation message will be shown.

### Face Recognition for Attendance

**Description**: System recognize user from face and marks its attendance automatically.

**Input**: Real time image from camera.

**Process**: System match real time image with the images stored in the database and confirms its identity.

**Output**: Attendance will be entered and confirmation message will be shown.

### Face Detection with Mask and Glasses

**Description**: System recognize user face even if they are wearing glasses and masks.

**Input**: Real time image from camera.

**Process**: System adjusts the algorithm to detect the partial features of face and match it with the images in the database.

**Output**: Attendance will be entered and confirmation message will be shown.

## Attendance Management

### Automatically Attendance Marking

**Description**: System automatically enter attendance by detecting user’s face when click on Take Attendance module.

**Input**: Image captured from camera.

**Process**: System match real time image with the images stored in the database for confirmation of its identity and marks its attendance with date and time in the database.

**Output**: A confirmation message will be shown if attendance marks successfully and error message if it fails to recognize.

### Manual Attendance Entering (Admin)

**Description**: Admin can manually enter user attendance and adjust records of users.

**Input**: User Id, date and time for attendance entering.

**Process**: Admin can select the user, inputs the date and time and update the attendance record in the database.

**Output**: A confirmation message will be shown if database updated successfully and error message if it fails to update.

## Reports and Dashboards

### View Attendance Reports (Admin)

**Description**: Admin can view the attendance report of the user of specific section.

**Input**: Section selection.

**Process**: System retrieve the data of that specific section and generate report of that record.

**Output**: A report will be shown and available for download.

### View Attendance Reports (User)

**Description**: User can view the attendance report of specific section.

**Input**: Section selection.

**Process**: System retrieve the data of that specific section and generate report of that record.

**Output**: A report will be shown to the user.

## Security Management

### Role Based Access Control

**Description**: System ensure that only authorized user (admin) can have access to certain functionalities like manual attendance entry.

**Input**: Log in as a admin.

**Process**: System checks if information is correct or not.

**Output**: Open the admin dashboard.

# System Architecture

The **face recognition-based student management and attendance system** is designed as a modular, distributed system that integrates face detection technology with a comprehensive student management platform. The system is primarily divided into front-end and back-end components, each with specific responsibilities and functionalities. The front-end interacts with users (admin and students) to gather data, such as face images, while the back-end processes these inputs, performs face recognition, manages the student database, and generates attendance reports.

**Subsystems and Components**

## 8.1 User Interface subsystem

The User Interface(UI) is the part of system through which user interact with system. In a face recognition-based student management and attendance system, the user interface must be intuitive, user-friendly, and responsive, designed to support different user roles such as Admin, and Students for profile view.

**Components:**

* Dashboard: The admin dashboard should present a clear summary of key features such as total students, daily attendance record.
* Login/Signup Page: The UI includes login and signup pages with proper validation and authentication mechanisms. Admin must be login using secure credentials.
* Student Registration: The UI allow Admin to register student with proper details like name, ID, class , contact number, and also upload student face photos for training.

### Student Management(CRUD) Subsystem

Handles CRUD (Create, Read, Update, Delete) operations for students, including adding new students, updating records, and removing students

**Components:**

* Update: In this UI, first of all admin search a student by StudentID and edit their necessary information.
* Delete: Confirmation message when deleting student from system.
* View: A page from where we can view all students details, including photos that can only be seen by admin.
* Attendance Reports : A UI to select classes to generate attendance reports. Includes a “DOWNLOAD REPORT” button.

**8.2 Face Recognition for Attendance Subsystem:**

This interface will display camera in real-time as system scans students faces and mark attendance.

**Components:**

* **Image Capture Module**: Uses the laptop camera to capture student faces.
* **Face Detection Module**: Detects faces in the captured image.
* **Face Recognition Module**: Identifies faces by matching them

## Student Interface Subsystem

This system where student can view its profile, mark attendance without admin and view its attendance report.

**Components:**

* Profile View: Students have access to view their profiles and attendance record.
* Self- Attendance: The UI allow students to mark attendance through face recognition without admin intervention.
* Design Considerations
* Responsive Design: UI works across multiple devices including dekstops, etc.
* Accessibility: UI for accessibility (e.g., high contrast mode, screen readers) to accommodate different user needs.

# Methadology

## 9.1 Data Collection

For this project dataset is collected from Kaggle that has contains two categories with mask and without mask. In with mask category all faces with masks are present and in without mask category all faces without mask are present.

## 9.2 Implementaion

## Pre Processing

For Pre Processing before made it for training images in the dataset

Convert all images to grayscale or normalize them to ensure uniformity.

* Resize the images to a fixed size (e.g., 128x128 pixels) to ensure compatibility with the recognition model.
* Use data augmentation techniques to increase dataset diversity (e.g., rotation, scaling, and flipping).

### 9.2.1 Overfitting

Overfitting happens when the model learns the noise or random patterns in the training data, making it perform well on the training set but poorly on unseen data. In this project, overfitting could arise if the model is exposed to excessive training on specific images, making it overly dependent on those features.

**Example**

If the model is trained solely on high-quality studio images, it may fail to detect faces captured in real-world settings, such as low-resolution webcam feeds.

### 9.2.2 Under Sampling

Under sampling involves reducing the number of samples in overrepresented classes to balance the dataset. For this project, this might mean reducing the number of images for certain users if they dominate the dataset but there is no need to do this in our project as it is mainly not overfitting.

**Pros**: Balances the dataset, avoiding model bias.

**Cons**: May lead to loss of valuable information.

### 9.2.3 Over Sampling

Over sampling is the process of increasing the number of samples in underrepresented classes to balance the dataset. In this project, duplicate or generate synthetic images for users with fewer face samples are implemented.

**Pros**: Improves model performance for underrepresented cases.

**Cons**: May lead to overfitting if synthetic data is not diverse enough.

### 9.2.4 Confusion Matrix

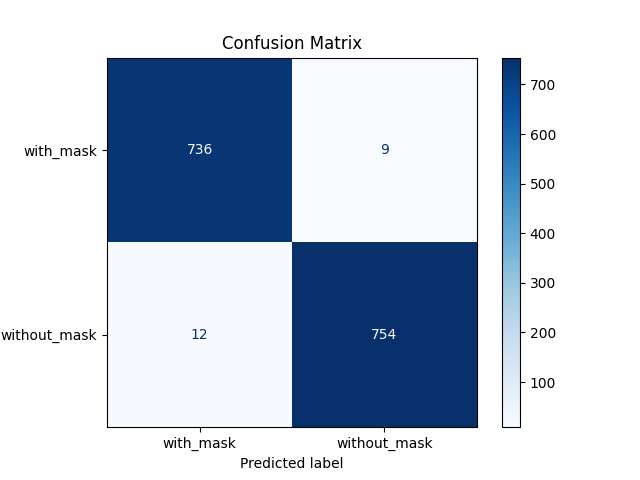
A confusion matrix is a tool to evaluate the performance of a classification model. It provides a summary of prediction results by comparing true labels to predicted labels. Here is a snapshot of confusion matrix.

Figure 1 Confusion Matrix

# App Development

The app development process involved a structured approach to ensure the final application is user-friendly, efficient, and scalable. It incorporated a robust combination of UI/UX design principles, modern front-end frameworks, a reliable back-end system, and a well-structured database. Each component was carefully integrated to create a seamless experience for the users while ensuring all functionalities operate efficiently.

10.1 UI/UX design

It was centered around user-friendliness and accessibility. Wireframes were designed to map out the app's structure, focusing on intuitive navigation and clear visual hierarchies. Elements such as buttons, forms, and visual cues were positioned strategically to enhance user interaction and minimize confusion. The design followed principles of simplicity and responsiveness, ensuring compatibility with devices of varying screen sizes. The goal was to provide users with a visually appealing interface that simplifies the task of managing attendance and accessing related information.

## 10.2 Front-end development

For this, the app utilized HTML, CSS, and JavaScript to create an interactive and responsive user-facing interface. HTML provided the structural foundation of the app, CSS ensured visual styling and layout consistency, and JavaScript introduced interactivity, such as dynamic content updates and real-time validation. These technologies enabled the creation of a responsive design that functions seamlessly across different platforms, offering a rich user experience.

## 10.3 Back-end development

It was handled using Flask, a lightweight Python framework known for its simplicity and flexibility. Flask managed the core application logic, including user authentication, attendance management workflows, and server-side processing. It also handled API requests and integrated seamlessly with the database. Flask's modular design allowed for efficient scaling, ensuring the app could handle increasing user demands without performance degradation.

## 10.4 Database design

It relied on MySQL, a robust relational database management system. The database was structured with tables for users, attendance records, and face recognition data. Relationships between tables were carefully designed to ensure efficient data retrieval and consistency. For example, a "Users" table stored information about users, while an "Attendance" table tracked attendance logs linked to specific users. The database design prioritized fast query execution to support real-time data retrieval during attendance checks.

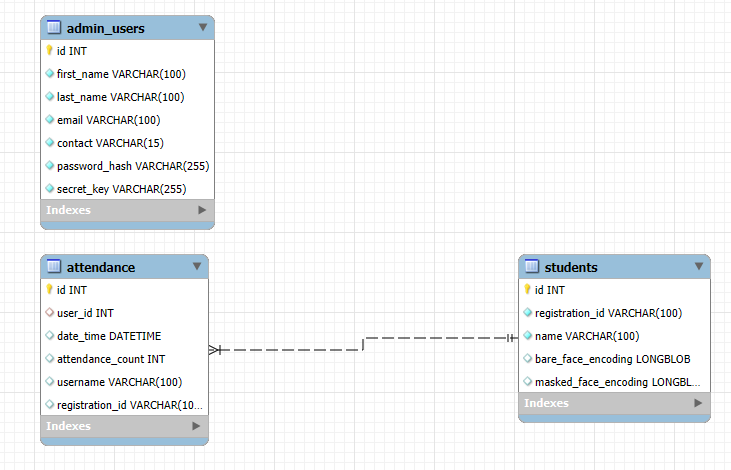


Figure 2 Database Overvew

## 10. 5 API integration

It played a crucial role in enhancing the app’s functionality. Flask APIs connected the front-end with the back-end and database, enabling smooth communication between components. While third-party APIs for additional functionality were not extensively used in this system, the modular nature of Flask ensures future scalability. For instance, APIs for advanced analytics or integration with external systems, such as school management platforms, can be easily incorporated if needed.

# ****Implementation****

## ****System Interface****

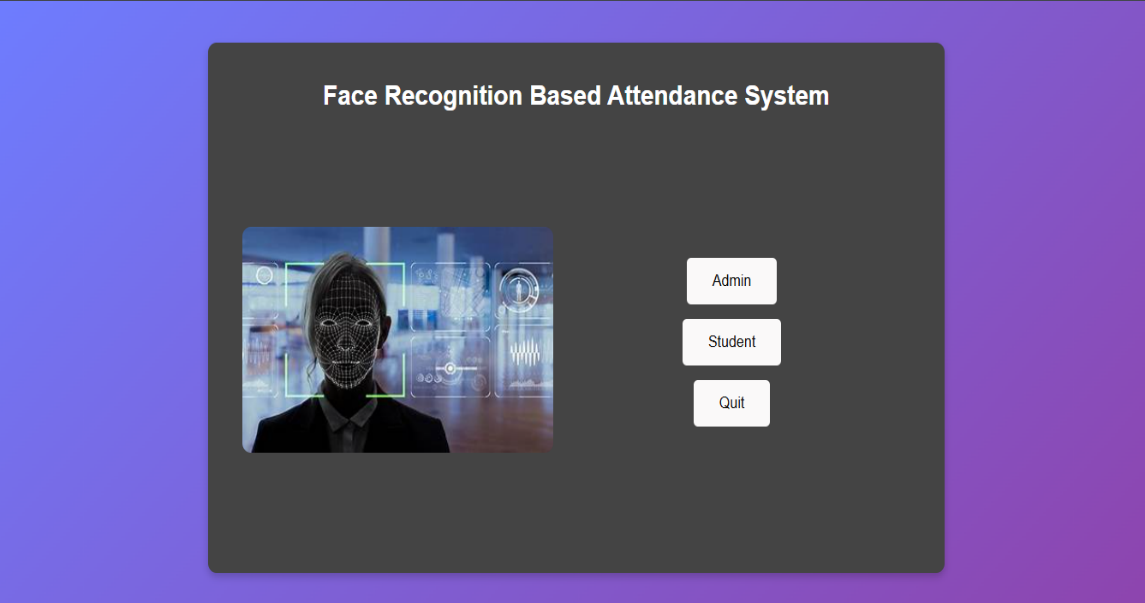


Figure 3 System First Screen

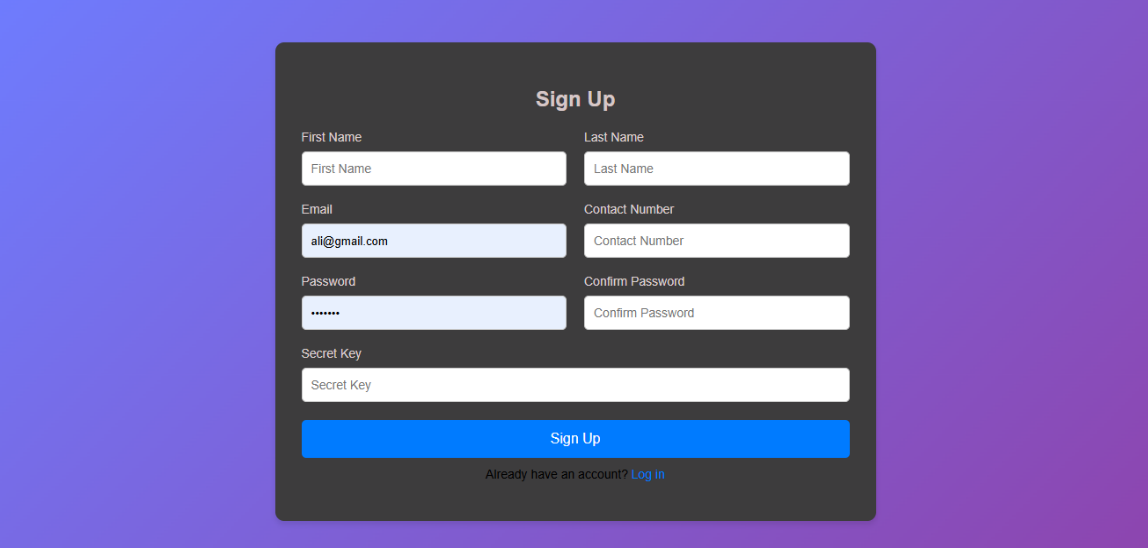


Figure 4 Sign Up Page

Figure 5Sign In Page

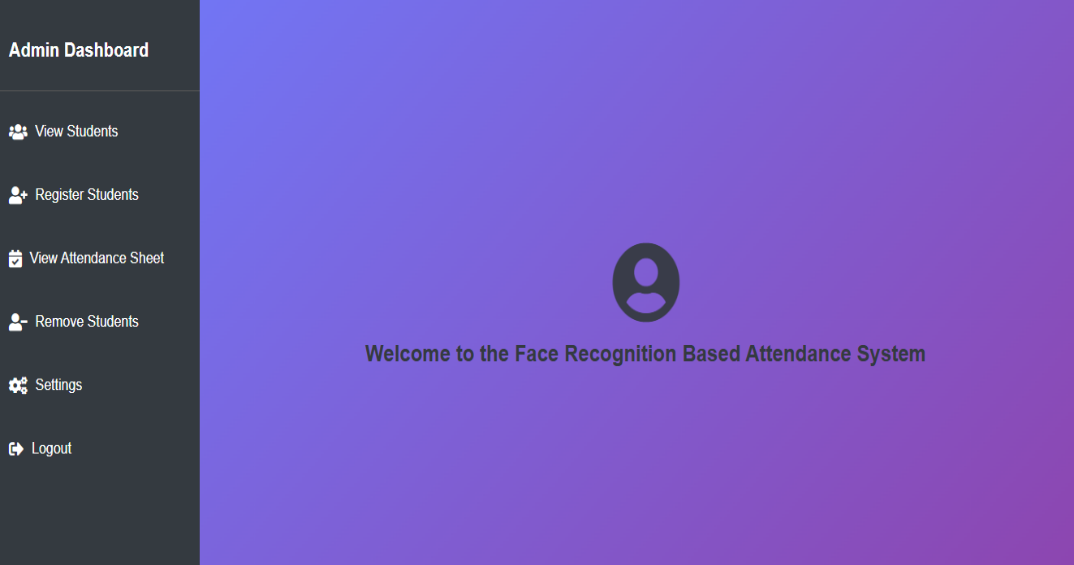


Figure 6 Admin Dashboard



Figure 7 View Student List

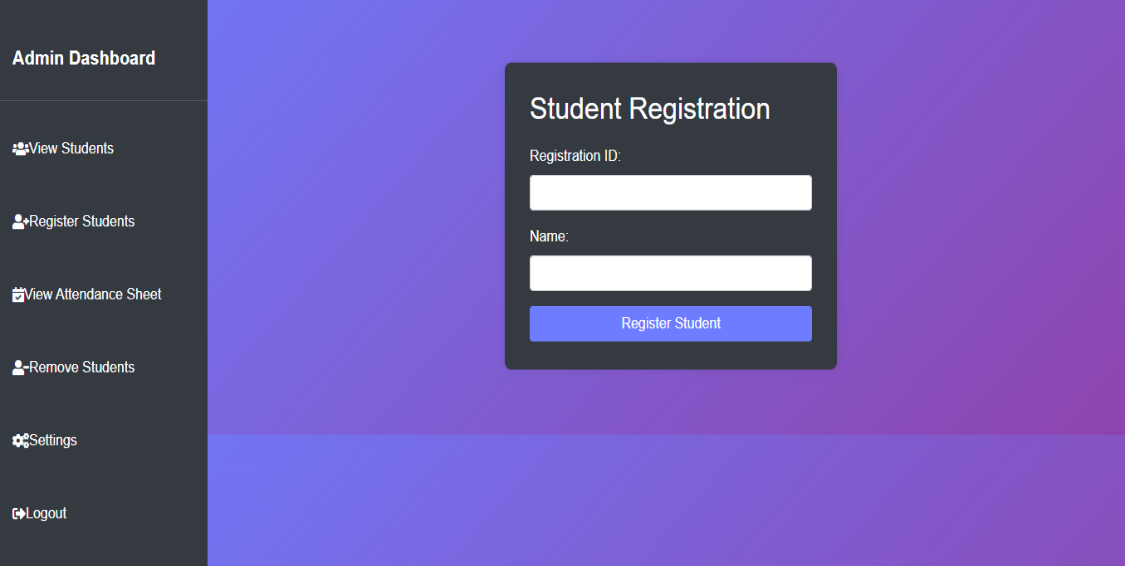


Figure 8 Register Student

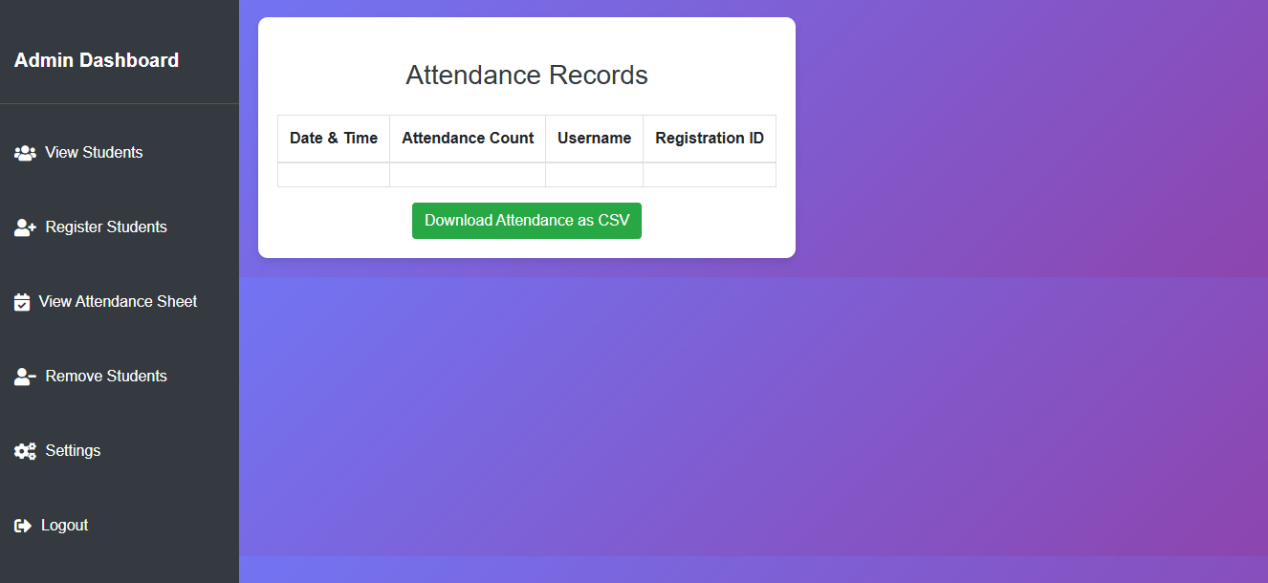


Figure 9 View Attendance Records

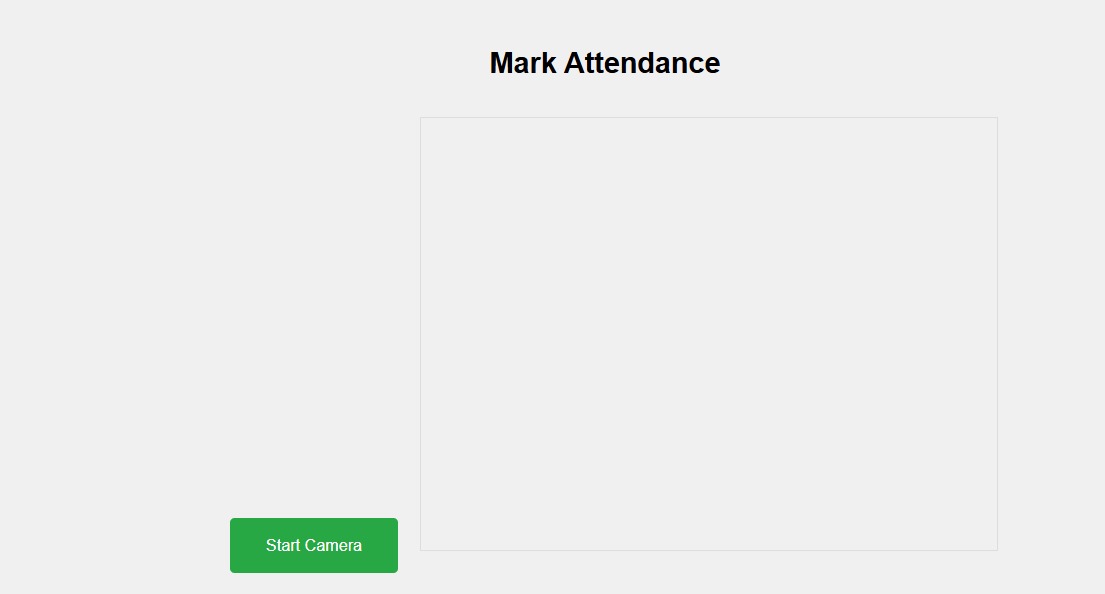
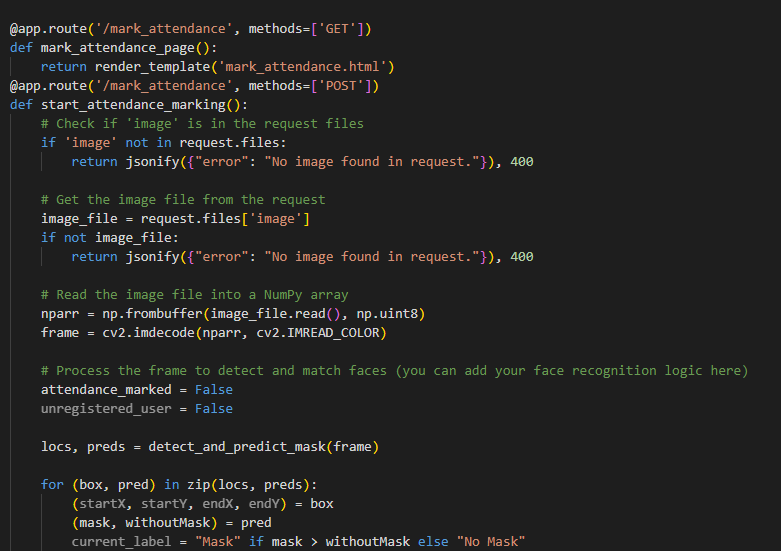


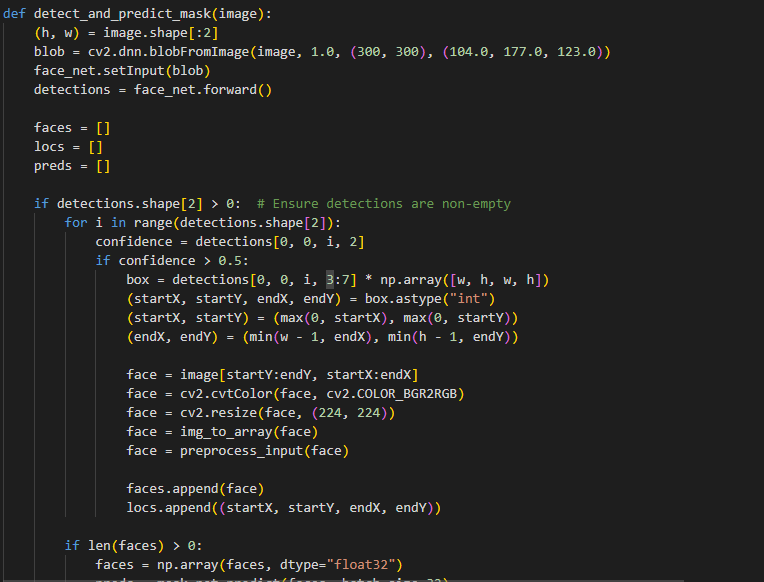
Figure 10 User Dashboard

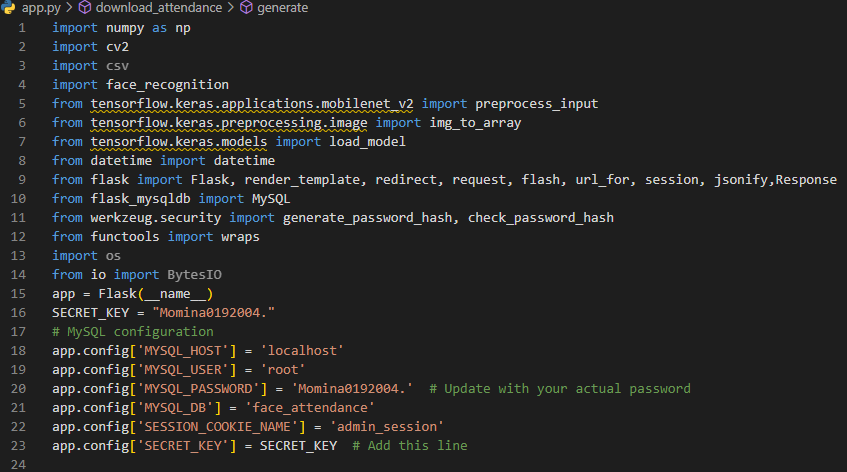
## ****11.2 Backend Workflows****

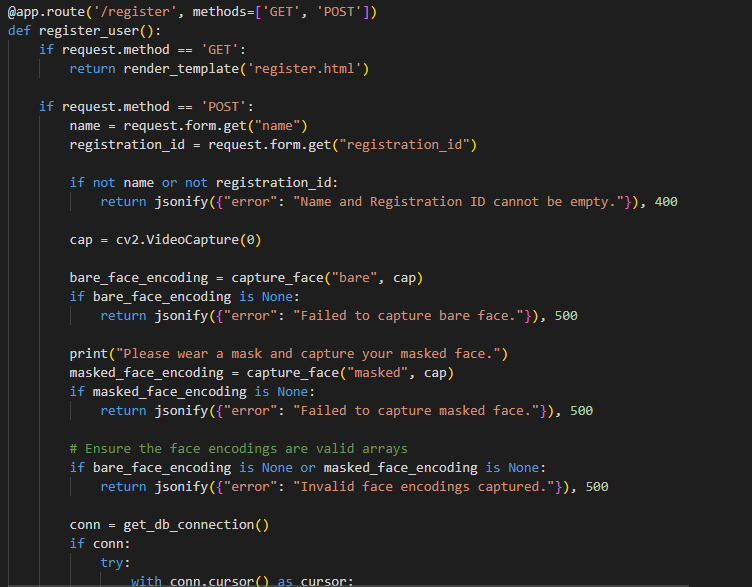
The backend flow for training data begins with the collection of facial images, captured through a camera for bear face and for mask ones taken from kaggle or uploaded by the admin via the dashboard. These images are organized systematically, with each student’s photos stored in separate directories to facilitate identification. The collected data undergoes preprocessing to ensure consistency, including resizing images to a standard resolution, converting them to grayscale or normalized color formats, and applying data augmentation techniques such as rotation and flipping to increase dataset variability. For dataset of mask we use DNN algorithm. Unique facial features are then extracted from each image using advanced recognition techniques like Local Binary Patterns Histograms (LBPH) or deep learning-based methods, which encode these features into numerical vectors known as facial embeddings. These embeddings are used to train a machine learning classifier, such as Support Vector Machines (SVM) or K-Nearest Neighbors (KNN), to map embeddings to the corresponding student identities. The trained classifier and embeddings are securely stored, enabling the system to compare live facial data with stored information for accurate recognition. Additionally, the system allows for periodic retraining to incorporate new students, ensuring long-term accuracy and adaptability of the model.

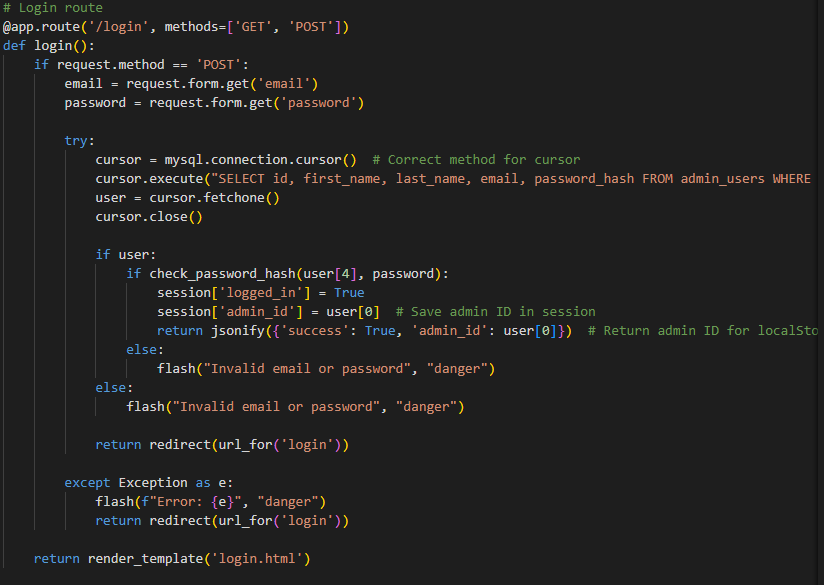
## 11.3 **Code Overview**

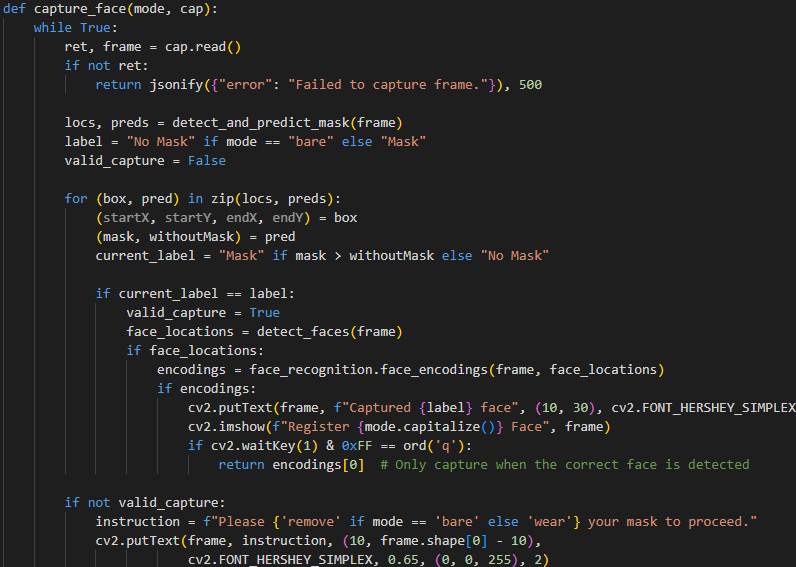


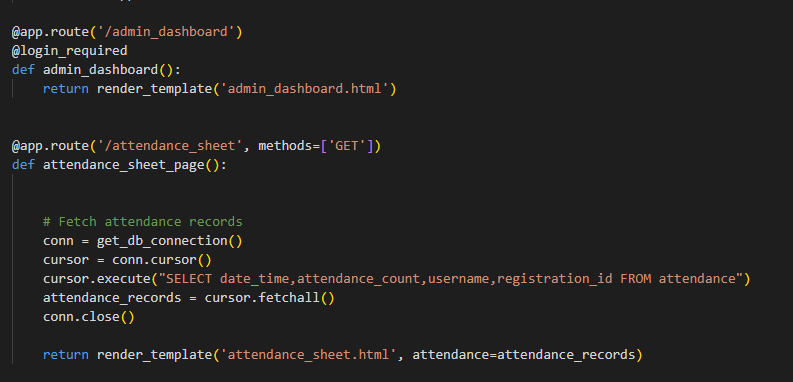




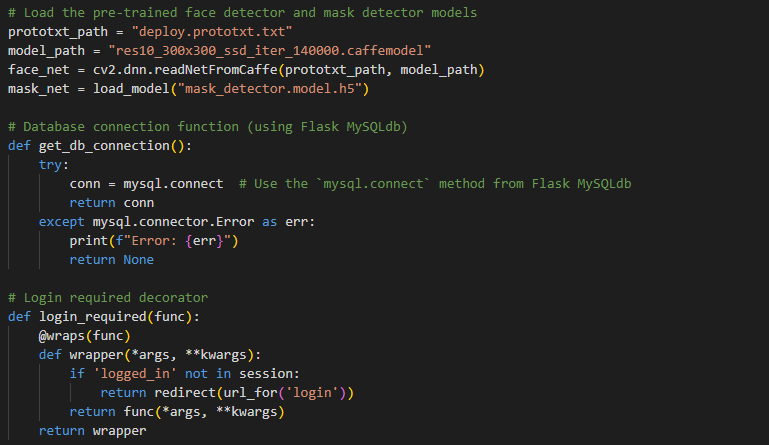


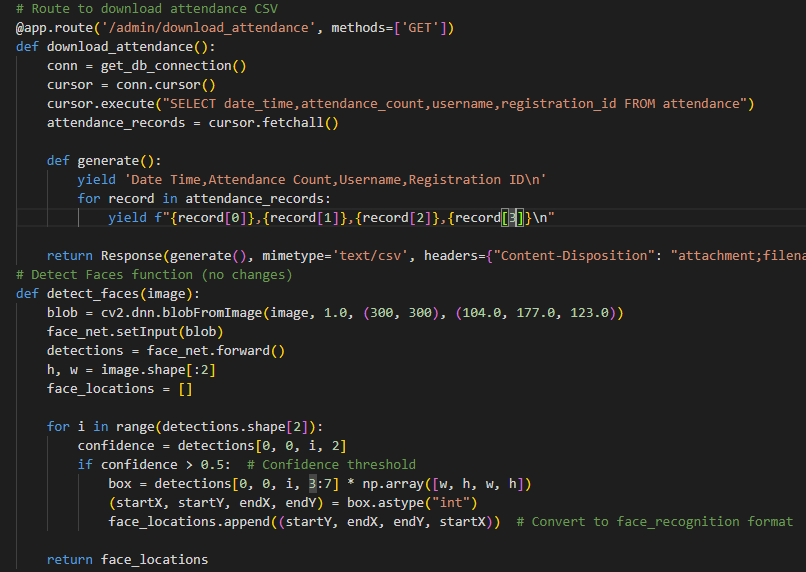














# Limitations

## ****Failure to Account for Edge Cases****

Face recognition systems may struggle in low-light conditions or when students have covered faces. These scenarios must be considered.

## 9.2 Insufficient Error Reporting

Systems may fail without giving informative error messages. Ensure that every failure in face recognition (e.g., unable to detect a face, network failure) provides meaningful feedback to users so they can take corrective action.

## 9.3 Ignoring Multi-Device Compatibility

While the system might be designed for use on a laptop with a built-in camera, there may be cases where administrators need to access the system on tablets or mobile phones. If multi-device support and adaptability to different screen sizes aren’t considered, the system could have limited usability.

## 9.4 Underestimating the Impact of Poor Data Quality

The face recognition system relies on the quality of input data . There should be clear, well-lit face images. If the requirements do not include conditions for the quality of student photos or how to handle poor-quality images, the system’s effectiveness can be severely affected.

## 9. 5 Neglecting Real-World Environmental Factors

Environmental conditions like lighting, camera angles, or even classroom setups can significantly impact system performance. Requirements that fail to address how the system should perform under these special conditions will lead to inaccuracies in attendance tracking

# Results and Discussion

## 10.1 Summary of Results

The Face Recognition Attendance System demonstrates a significant improvement over traditional attendance management methods by leveraging state-of-the-art facial recognition technology. The system efficiently captures facial data, compares it with stored encodings, and marks attendance with high accuracy. Its ability to recognize both bare and masked faces enhances its adaptability, especially in post-pandemic environments. The system processes attendance swiftly, significantly reducing the time required compared to manual methods. Additionally, it ensures minimal errors in attendance recording, offering a reliable and automated solution for managing attendance records.

## 10.2 Real-world Applicability

The developed system has extensive potential applications in schools, universities, offices, and other organizations where attendance tracking is critical. By automating the process, it eliminates common issues such as proxy attendance and human error in manual records. For educational institutions, the system ensures accurate tracking of student attendance, aiding in compliance with regulatory requirements. In workplaces, it enhances employee monitoring and attendance reporting, which can lead to improved productivity and resource management. Furthermore, the system’s adaptability to use cases requiring masked face detection makes it particularly relevant for healthcare facilities, public spaces, and environments where hygiene is a priority. This project addresses the growing need for secure and efficient attendance systems by providing a scalable, real-time solution to prevent fraudulent practices and improve accountability.

# Conclusion

The Face Recognition Attendance System provides an efficient, secure, and contactless solution for modern attendance management. By leveraging OpenCV DNN, the face\_recognition library, and MySQL, the system successfully implemented facial detection and authentication while streamlining the attendance process. Despite challenges like lighting variations and masked face recognition, the system has proven its potential to replace traditional methods. Its scalability, accuracy, and integration with biometric technology highlight its suitability for real-world applications, setting the foundation for future advancements.

# Future Work

Future improvements include adopting advanced deep learning models like YOLO or Faster R-CNN for enhanced accuracy and reliability, especially for masked face recognition. Expanding functionality through multi-modal biometrics, cloud integration, and real-time reporting would increase scalability and usability. Security upgrades, such as encrypted data handling, and cross-platform support can further optimize the system. With these advancements, the system can evolve into a robust, scalable solution for modern attendance management.

# References

|  |  |
| --- | --- |
| [1] | C. Larman, "USE CASES," in *UML and Patterns.book*, Thursday, September 16, 2004, p. 40. |
| [2] | C. A. h5. Davis, *IEEE Guide to Software Requirements Specifications ,* no. Febraury 10, 1984, p. 26, September 30,1983 . |
| [3] | P. P. a. K. K. Mohanty, "AI-based techniques for real-time face recognition-based attendance system-A comparative study," *th International Conference on Electronics, Communication and Aerospace Technology (ICECA),* p. 1034–1039, 2020. |
| [4] | T. A. T. S. M. a. R. K. KP Naveen Reddy, "AI-based attendance monitoring system," *International Journal of Innovative Technology and Exploring Engineering,* vol. 9, no. 2S, p. 592–597, 2019. |
| [5] | . W. F. C. L. Ankur Singh Bist, "Design of face recognition AttendX for recording student attendance data based on artificial intelligence technology," *Solid State Technology,* vol. 63, no. 2S, p. 4505–4518, 2020. |