SmartAttendance.Pi

(An Automatic Face-based Attendance System for Smart Classroom)

A project submitted

in fulfillment of the requirement for the

award of

Master of Computer Applications

by:

Aditya Maurya	2023CA09
Shishupal Sahu	2023CA90
Somenath Singh	2023CA99

Under the guidance of

Dr. Shailendra Shukla

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MOTILAL NEHRU NATIONAL INSTITUTE OF TECHNOLOGY

ALLAHABAD

UNDERTAKING

We declare that the work presented in this report titled **“SmartAttendance.Pi (**An Automatic Face-Based Attendance System for Smart Classroom**)”** submitted to the Computer Science and Engineering Department, Motilal Nehru National Institute of Technology Allahabad, Prayagraj, for the award of the Master of Computer Applications degree, is our original work. In case this undertaking is found incorrect, we accept that our degree may be unconditionally withdrawn.

May 2025

Allahabad

Aditya Maurya
Shishupal Sahu
Somenath Singh

CERTIFICATE

This is to certify that **Aditya Maurya, Shishupal Sahu, and Somenath Singh** have successfully carried out the completion of the project entitled **“SmartAttendance.Pi (**An Automatic Face-Based Attendance System Smart Classroom**)”** under my supervision during session 2024-25 and all the requirements of the project have been met.

May 2025

Dr. Shailendra Shukla

Computer Science and

Engineering Dept.

MNNIT Allahabad

ACKNOWLEDGMENT

We would like to give our sincere thanks to our mentor, **Dr. Shailendra Shukla,** who guided us throughout our project **“SmartAttendance.Pi (**An Automatic Face-Based Attendance System for Smart Classroom**)”** in every possible way with his invaluable advice and valuable suggestions. Our project would not have been possible without his constant guidance, support, and encouragement that helped us complete our project on time. We feel honored and privileged to work under him.

Aditya Maurya
Shishupal Sahu
Somenath Singh

Contents

Abstract

Our project "**SmartAttendance.Pi** (An Automatic Face-Based Attendance System for Smart Classroom)" is an intelligent and automated platform developed to modernize the process of attendance management using facial recognition technology. Upon registration and login, users gain access to a range of functionalities designed to simplify and secure the attendance-taking process in institutions and workplaces.

"**SmartAttendance.Pi** (An Automatic Face-Based Attendance System for Smart Classroom)" enables users to register individuals by capturing and storing their facial data. During attendance sessions, the system utilizes real-time face detection and recognition to automatically identify and mark present individuals. It maintains detailed logs with date and time stamps, allowing for accurate and transparent attendance records. The platform also includes features for monitoring daily attendance summaries and generating reports.

Overall, this project represents an innovative and practical application of artificial intelligence in the field of education and workforce management. By replacing traditional attendance methods with a fast, contactless, and error-free solution, this system enhances efficiency and promotes a tech-driven approach to daily operations.

Chapter 1

Introduction

1.1 Motivation

The motivation behind our project "**SmartAttendance.Pi (**An Automatic Face-Based Attendance System for Smart Classroom**)**" is to streamline and modernize the traditional attendance process using the power of facial recognition technology. In educational institutions and workplaces, manual attendance systems are often time-consuming, prone to errors, and susceptible to proxy marking. By introducing an automated, contactless solution built on a Raspberry Pi and open-source tools, we aim to offer a more efficient, accurate, and hygienic alternative that ensures data privacy and is accessible even in low-resource environments.

The proposed **"SmartAttendance.Pi (**An Automatic Face-Based Attendance System for Smart Classroom**)"** system addresses a critical gap in the market by offering a cost-effective, offline, and privacy-focused alternative to existing attendance solutions. Many competing systems, such as BioEnable/Mantra, MyClassCampus, and CamAttendance, come with significant financial barriers, making them impractical for low-resource educational institutions or small businesses. For instance, BioEnable/Mantra requires a high initial setup cost (₹6,000–₹15,000) along with annual maintenance charges (AMC) of ₹2,000–₹5,000, while MyClassCampus operates on a SaaS model with recurring monthly fees (₹50–₹200 per student), which can become prohibitively expensive for larger groups. Similarly, CamAttendance, designed for enterprises, demands an exorbitant upfront investment (₹25,000–₹60,000 per device) and ongoing monthly charges (₹500–₹1,000). Even API-based solutions like FaceTagr/Staqu incur variable costs (₹0.50–₹2.00 per scan or ₹5,000–₹50,000 monthly), which escalate with usage. In contrast, **"SmartAttendance.Pi (**An Automatic Face-Based Attendance System for Smart Classroom**)"** eliminates these financial burdens with a one-time cost of just ₹8,000–₹10,000, no recurring fees, and minimal internet dependency. This affordability, combined with its portability and offline functionality, makes it an ideal solution for budget-conscious settings where existing options are simply too costly.

1.2 Objective

The objective of our project, "**SmartAttendance.Pi (**An Automatic Face-Based Attendance System for Smart Classroom**)**" is to develop a compact and cost-effective system capable of identifying individuals in real time and marking their attendance automatically. The system leverages Dlib-based 128D face encoding for reliable recognition and supports various features such as student registration, course-wise XLS attendance generation, voice announcements, and email-based reporting. Our goal is to minimize manual intervention, reduce infrastructure costs, enhance data security, and make the process of attendance management seamless and efficient — even in offline scenarios.

A key objective of **SmartAttendance.Pi** is to provide institutions with a high-quality attendance system at a fraction of the cost of commercial alternatives. Unlike enterprise-grade solutions that require expensive hardware (₹25,000–₹60,000 per device) or recurring SaaS subscriptions (₹50–₹200 per user monthly), our system is built on affordable Raspberry Pi hardware with a one-time cost of just ₹8,000–₹10,000. By eliminating licensing fees, cloud dependencies, and proprietary software, we significantly reduce both initial and long-term expenses. This makes advanced facial recognition technology accessible to schools, small businesses, and rural institutions that would otherwise be priced out of the market. The open-source foundation further ensures scalability and customization without additional financial burden, delivering premium functionality at an economical price point.

In this report, we will provide a detailed overview of the Automatic Face-based Attendance System, including its features, system architecture, implementation details, and real-world applications. We will also highlight the comparative advantages of our system over existing solutions and suggest future enhancements to expand its capabilities further.

Chapter 2

Proposed Work

The SmartAttendance.Pi project aims to revolutionize attendance management in educational and workplace settings by introducing an automated, contactless, and cost-effective facial recognition system. Built on Raspberry Pi and open-source technologies, this solution eliminates the inefficiencies of manual attendance tracking while addressing the high costs and privacy concerns of existing commercial systems. The project follows a structured development approach—requirement gathering, system design, development, testing, and maintenance—to ensure a robust and user-friendly product. By leveraging Dlib’s 128D facial encoding, the system provides real-time, offline attendance marking with features like Excel report generation, voice confirmation, and email notifications. This report details the system's architecture, data processing, and predictive model, highlighting its advantages over traditional methods.

1. Requirement Gathering: This involves identifying the essential requirements and features of the system, such as face recognition using 128D Dlib facial features, real-time attendance marking, offline functionality, course-wise attendance support, and voice-based confirmation. The device should be lightweight, cost-effective, and able to function without internet connectivity, except during email reporting.

2. Design and architecture: This involves designing the system’s hardware and software components. The hardware includes a Raspberry Pi-based compact setup integrated with a camera module. The software architecture includes modules for student registration, face recognition, data storage in Excel files, and user interface interactions. The system must be designed to operate offline and ensure data privacy through local storage.

3. Development involves implementing the system design using Python along with open-source libraries like Dlib for facial recognition. The interface is built with a basic UI suitable for non-technical users, and local file handling is used for storing facial features and Excel-based attendance records. Email functionalities are implemented using SMTP protocols for report sharing.

4. Testing: This involves testing all core functionalities of the system including face detection and recognition accuracy, Excel file generation, voice announcements, and email dispatch.

5. Maintenance and updates: This involves monitoring the website's performance, fixing bugs, and updating the website's features and functionality to ensure that it remains relevant and up-to-date.

2.1 Data Collection

Collecting relevant data is one of the most important aspects of building the Automatic Face-based Attendance System. The system requires accurate facial data of students to perform reliable recognition during attendance sessions.

Facial data is collected during the student registration phase, where multiple images are captured from different angles to ensure robustness in real-time detection. These images are then processed to extract 128-dimensional facial features using the Dlib library. Only the extracted facial encodings are stored locally on the device to ensure privacy and reduce storage usage.

2.2 Data Pre-processing:

1. Data pre-processing is a critical step in building the Automatic Face-based Attendance System, as it ensures that the facial data used for recognition is clean, efficient, and reliable for accurate real-time performance. The following are the common data pre-processing steps involved
2. Image Cleaning: This step involves validating the captured images during registration. Blurry or improperly lit images are discarded to maintain a clean dataset. Ensuring consistent lighting and resolution during image capture improves recognition accuracy.

3. Feature Extraction: Instead of storing raw images, the system extracts 128-dimensional facial encodings using the Dlib face recognition library. These encodings serve as a compact and unique representation of each student's face and are used for future matching.

4. Data Transformation: The extracted facial features are converted into a standardized format suitable for comparison during attendance sessions. This format is structured and stored locally for fast retrieval.

5. Data Reduction: To ensure the system remains lightweight, the original face images are automatically deleted after feature extraction. Only the essential numerical encodings are stored, reducing memory usage and enhancing data privacy

2.3 Prediction Model

The Automatic Face-based Attendance System relies on a facial recognition model that uses Dlib's 128D facial feature encoding to identify and verify individuals with high accuracy. Unlike traditional authentication systems, this model enables contactless, real-time recognition suitable for classroom and institutional environments.

Dlib’s face recognition model is based on deep metric learning, where facial features are mapped into a 128-dimensional vector space. The key advantage of this approach is that it does not require re-training for every new face. Instead, faces are compared based on the distance between their feature vectors.

Here is a high-level overview of how the face recognition model functions:

1. Data Preparation: The first step involves capturing clear images of students from multiple angles during registration. These images are used to generate 128-dimensional facial feature vectors using a pre-trained Dlib model.

2. Feature Extraction: The model detects the face within each image and calculates a unique 128D feature vector. These vectors serve as a compact numerical identity for each student and are saved locally.

3. Recognition Process: During attendance, the system captures a real-time image of a student and extracts its 128D vector. It then compares this vector with the registered vectors using Euclidean distance. A match is declared if the distance falls below a pre-defined threshold.

4. Efficiency and Accuracy: The Dlib model provides reliable accuracy while being lightweight enough to run on devices like Raspberry Pi. It offers fast processing and high recognition rates under varied lighting and pose conditions.

5. Privacy and Optimization: To enhance privacy, the raw images are deleted after feature extraction. Only the numerical encodings are used for matching, ensuring the system remains secure and lightweight.

Chapter 3

System Design

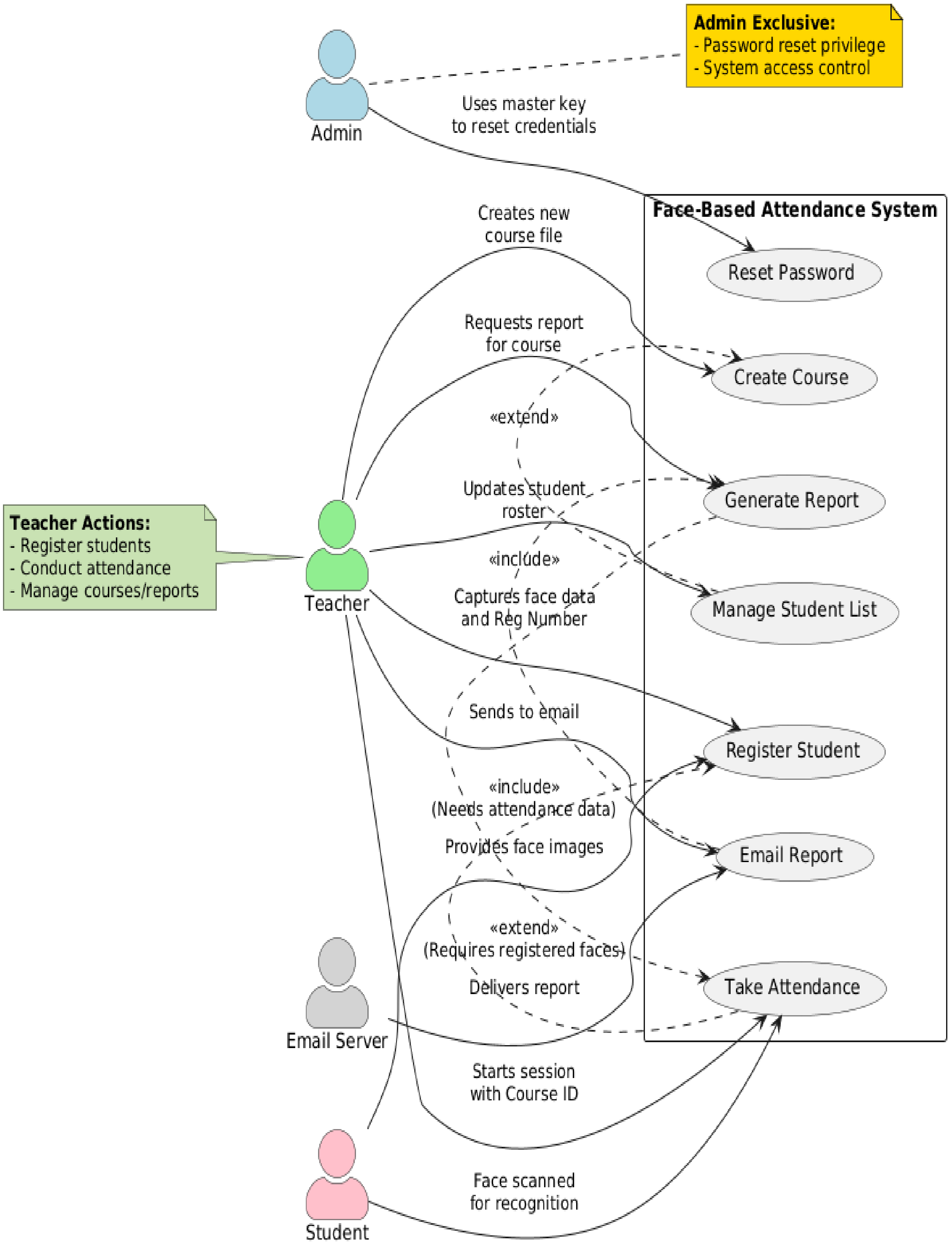
3.1 USE CASE DIAGRAM

A use case diagram is a visual representation of the functional requirements of a system, showing the interactions between the system and its users.

the functionalities of an **Admin Exclusive system**, primarily focusing on a Face-Based Attendance System. The admin has exclusive privileges such as password reset and system access control, including a master key to reset user credentials. A significant feature is the ability to create a new course file, which involves generating reports for courses. This process includes updating the student roster, capturing face data and registration numbers, and sending reports via email. Additionally, it requires attendance data and registered faces to deliver comprehensive reports. The system initiates a session using a Course ID and uses face scanning for recognition.

For **teachers**, the system allows actions like registering students, conducting attendance, and managing courses/reports. The Face-Based Attendance System encompasses several key functions: resetting passwords, creating courses, generating reports, managing student lists, registering students, emailing reports, and taking attendance. Overall, the system streamlines administrative and teaching tasks by integrating face recognition technology for efficient attendance tracking and reporting.

Here is a use case diagram for the Automatic Face-based Attendance System:



3.2 ER Diagram

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system.

The database schema of a Face-Based Attendance System, which consists of three key tables. The COURSE table stores the unique Course ID, serving as the primary identifier for each course in the system. The STUDENT table contains essential student details, including their registration number and a 128-dimensional facial feature vector, which is used for facial recognition and identity verification.

The ATTENDANCE table records attendance data by linking students to their respective courses. It includes fields such as the registration number, course_id, date, and status (e.g., present or absent). This structured approach ensures seamless integration of facial recognition technology with attendance tracking, enabling efficient and automated management of student attendance records. The relational design allows for quick retrieval and analysis of attendance data while maintaining accuracy and consistency across the system.

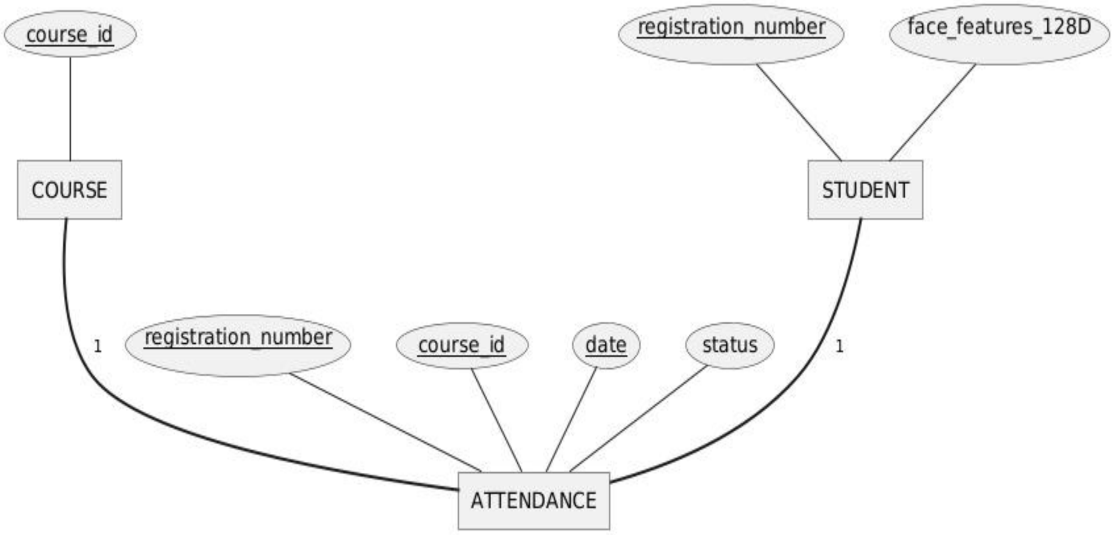


Fig 4.2 ER Diagram

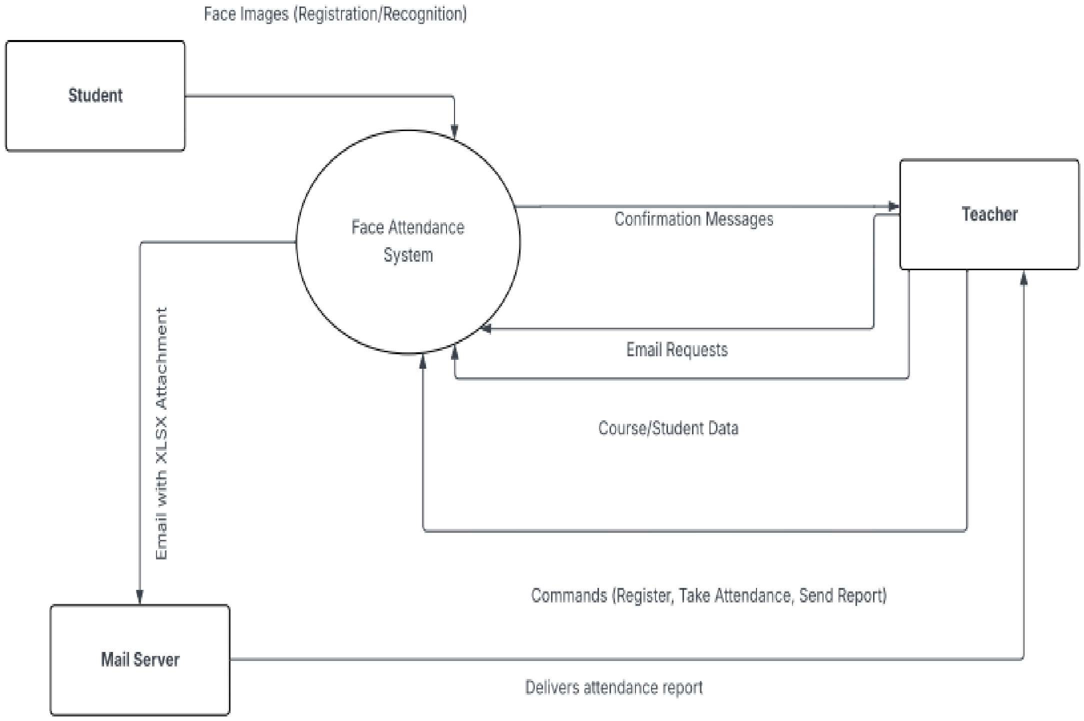
3.3 DFD Diagram

DFD stands for Data Flow Diagram, which is a graphical representation of how data flows within a system.

There are different levels of DFDs, ranging from the high-level context diagram to detailed diagrams that show the internal workings of a system.

3.3.1 LEVEL 0:

DFD Level 0 is also called a Context Diagram. It is a basic overview of the whole system or process being analyzed or modeled.


Face Images (Registration/Recognition)
Student
Face Attendance System
Confirmation Messages
Teacher
Email Requests
Course/Student Data
Email with XLSX Attachment
Commands (Register, Take Attendance, Send Report)
Mail Server
Delivers attendance report

Data Flow Diagram (Level 0)

3.3.2 LEVEL 1:

A level 1 DFD notates each of the main sub-processes that together form the complete system. We can think of a level 1 DFD as an “exploded view” of the context diagram.

The image illustrates a Face-Based Attendance System that uses 1280D facial recognition to automate student identification and attendance tracking. Admins control the system through a Master Key, enabling functions like password resets, student registration, and course management via unique Course IDs. The system logs attendance data in real-time, processes it into present/absent records, and automatically generates XLSX reports for email distribution. Secure authentication ensures only authorized access, while the facial feature database (Textures_all.csv) maintains accuracy. This integrated solution streamlines attendance management with efficiency and security.

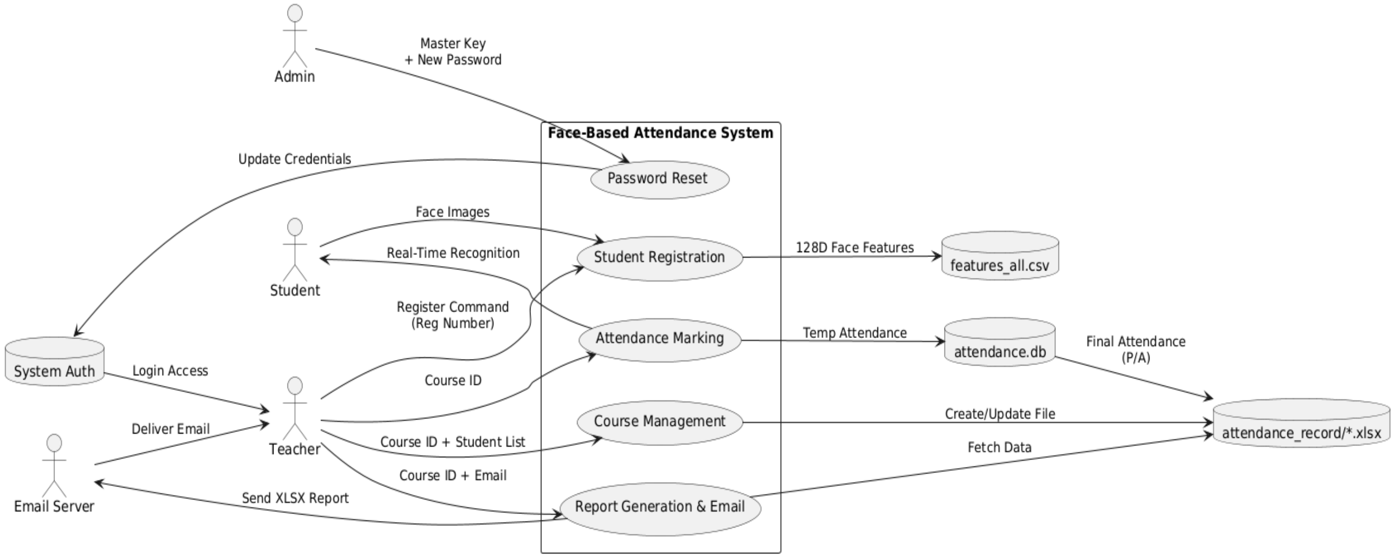


Fig 4.4 First Level DFD

3.3.3 LEVEL 2:

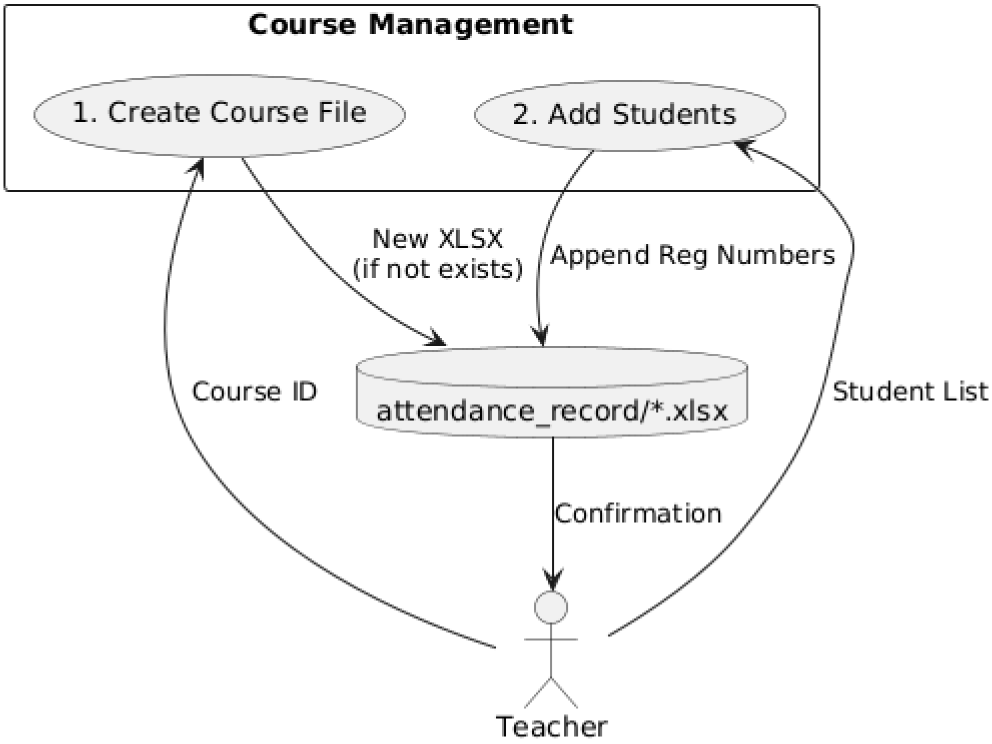


Figure1: Course Creation

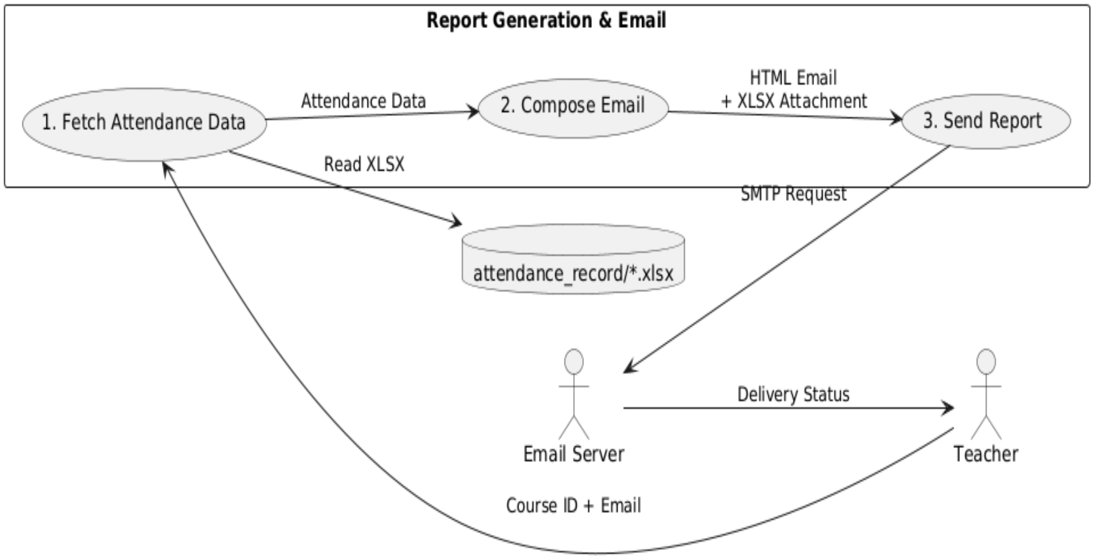


Figure2: Email Sending

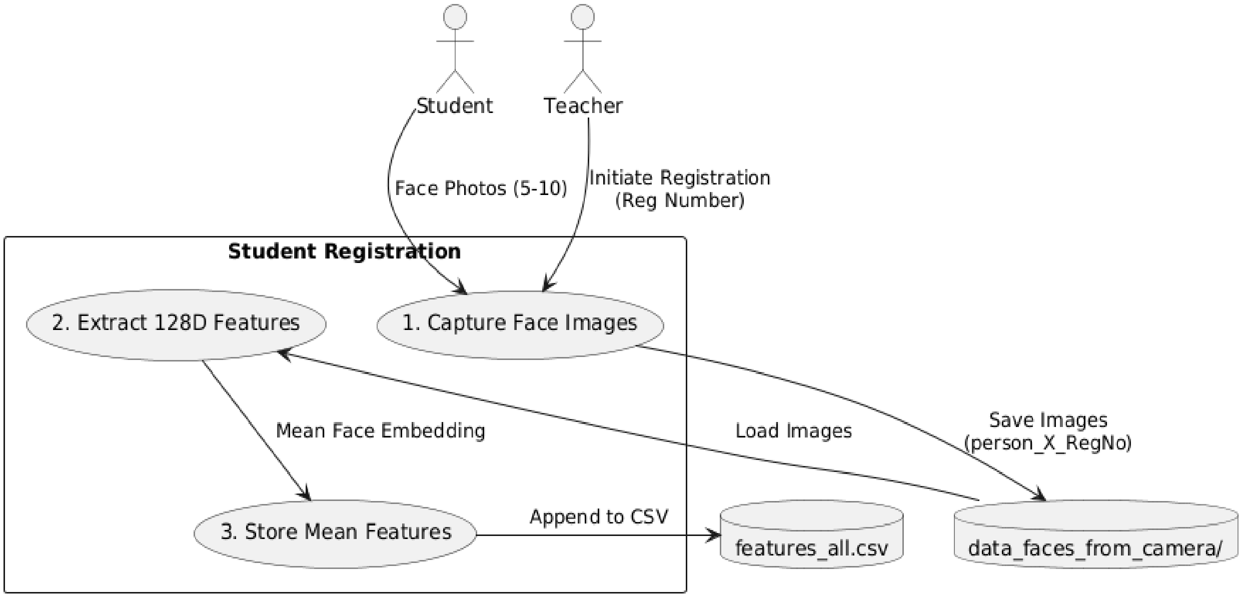


Figure3: Register Student

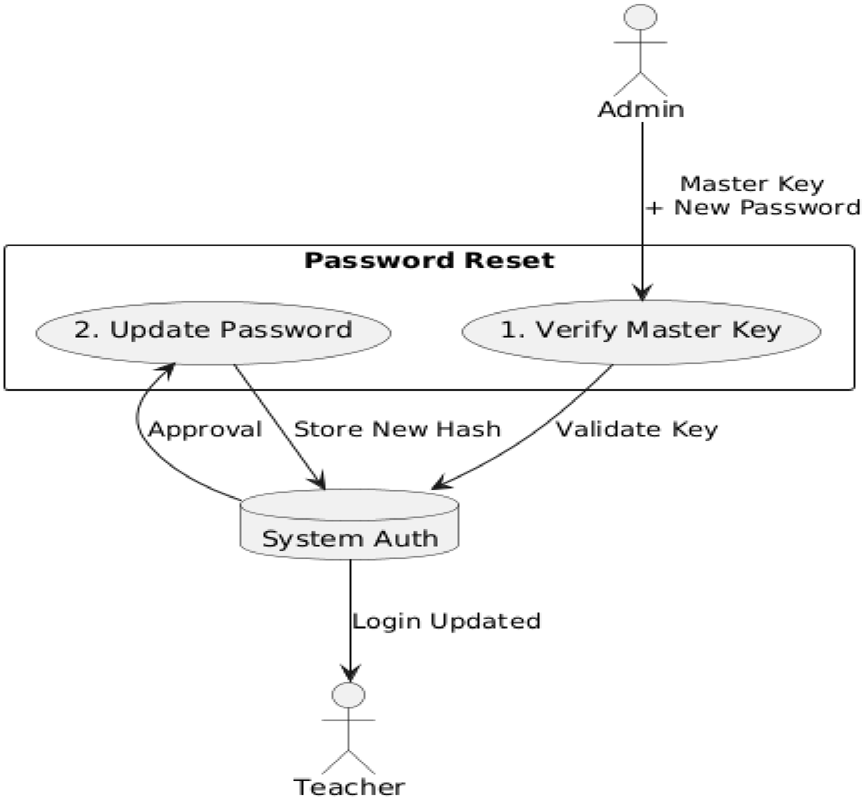


Figure4: Password Reset

.

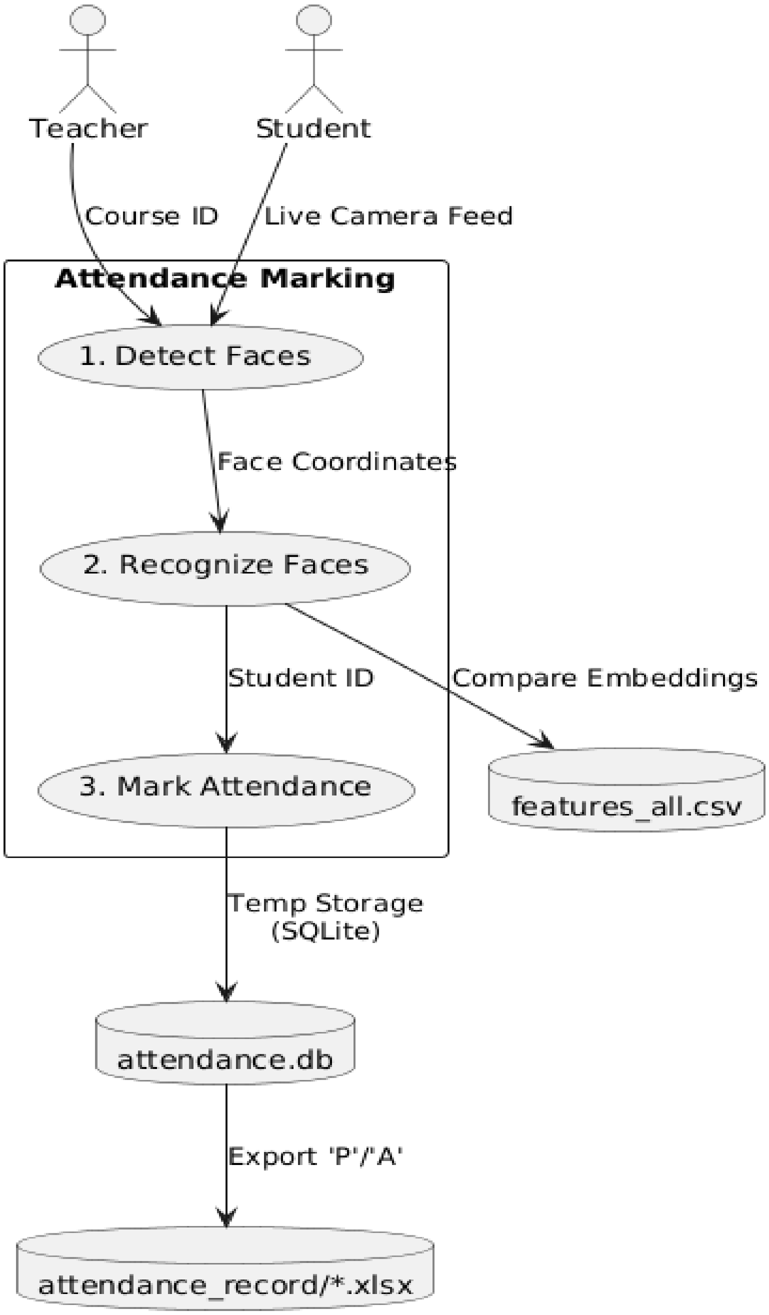


Figure5: Mark Attendance

Chapter 4

Implementation

The implementation of the **SmartAttendance.Pi (**An Automatic Face-Based Attendance System for Smart Classroom**)** involves several components, including user interface development, facial recognition model integration, and attendance data handling.

4.1 Front-end Development:

The front-end of the system is developed using Tkinter, Python's standard GUI library. It provides a lightweight and intuitive interface that allows users (teachers/admins) to: Register students by capturing face images, Take attendance using real-time face recognition, Send attendance files via email The interface is designed with accessibility in mind, making it suitable for non-technical users and allowing smooth operation on low-resource hardware like Raspberry Pi.

.

4.2 Back-end Development:

The back-end logic of the system is written entirely in Python, utilizing several key libraries: Dlib for facial feature extraction (128D encodings), OpenCV for real-time image and video frame capture, Pandas, OpenPyXL, and xlrd for handling attendance data in ".xls/.xlsx formate".

Other back-end features include:

- Voice announcement of present students using pyttsx3
- Email sending functionality via smtplib and MIME libraries
- Local database management using SQLite for efficient course and registration tracking

4.3 Face Recognition Implementation:

The core facial recognition functionality is implemented using the Dlib face recognition library, which relies on deep metric learning. Key steps in this module include:

- Capturing multiple face angles during student registration
- Extracting 128-dimensional facial embeddings and storing them locally
- Comparing real-time captured encodings with stored ones using Euclidean distance for identity matching

Chapter 5

Software and System Requirements

The technical requirements for the proposed **Automatic Face Attendance System** are feasible and should not be difficult to acquire. This system is designed to function offline and is especially suited for low-connectivity environments. Below are the detailed software and system requirements for the project.

5.1 Software Requirements

The technologies and tools required for this project are as follows:

- Python 3.x : The core programming language used for the development of the system.

- OpenCV :A computer vision library used for capturing images, processing video streams, and face detection.

- Dlib : A library for machine learning and face recognition. Specifically used for facial feature extraction using 128-dimensional vectors.

- Tkinter :A library for creating the graphical user interface (GUI) for easy interaction with the system.

- NumPy; A library for numerical operations, primarily used for handling arrays and matrices related to image processing.

- Pillow (PIL) :A Python Imaging Library for image processing, manipulation, and conversion.

- pyttsx3 :A text-to-speech library used for announcing the attendance or names of students marked present.

- python-dotenv : Used to manage environment variables, particularly for storing sensitive data like email credentials.

- Pandas :A data manipulation library used for managing attendance data and exporting it to .xls files.

- Openpyxl : A library to read and write Excel files, which is crucial for storing the attendance records in an organized manner.

- Smtplib : A built-in Python library to send emails, used for emailing the attendance reports.

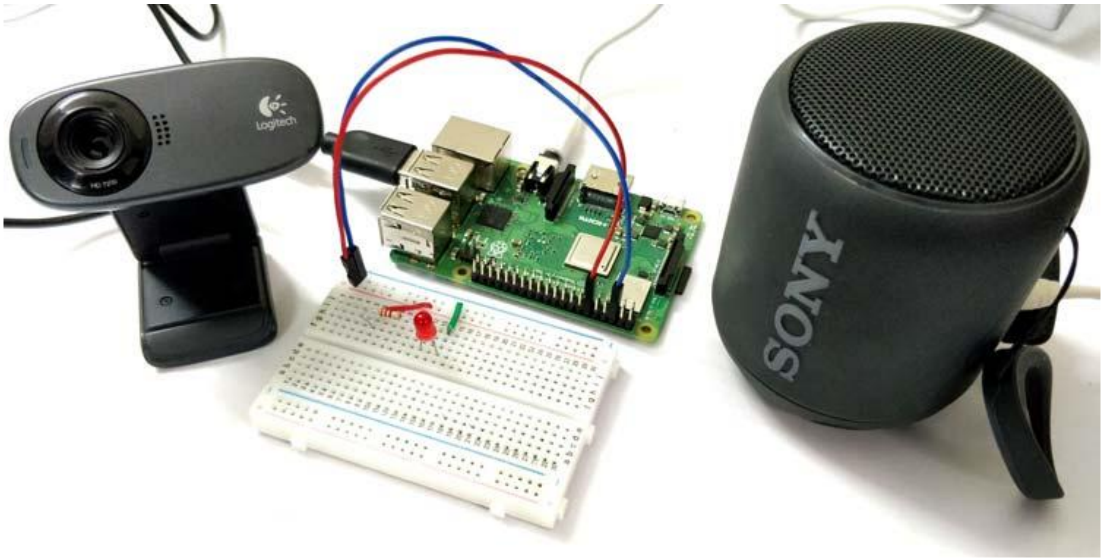
- sqlite3 : A lightweight database engine for storing student data and attendance logs locally.

- Xlrd :A library for reading Excel files, used to process and analyze attendance records stored in .xls format.

- Logging : A built-in Python library for tracking system logs and errors.

- Datetime For handling date and time, especially for adding timestamps to the attendance records.

5.2 System Requirements



To run the Automatic Face Attendance System effectively, the system should meet the following hardware and software requirements:

1. **Operating System**:
   - Linux (preferred, especially for Raspberry Pi deployment).
   - Windows 10 or above
   - macOS
2. **Processor**:
   - Intel Core i3 or higher, AMD, or Apple M1 chip
   - Raspberry Pi 4 Model B or higher for optimal performance
3. **RAM**:

Minimum 2 GB RAM (Recommended: 4 GB or higher for better performance)

4. **Storage**:
   - Minimum 2 GB of free disk space for system and software installation
   - Additional storage may be required depending on the number of students and data processing needs

5. **Web Browser**:
    - Chrome (preferred for any web-based user interface or email setup)
    - Firefox or Edge can be used as alternatives.
6. **Camera**:

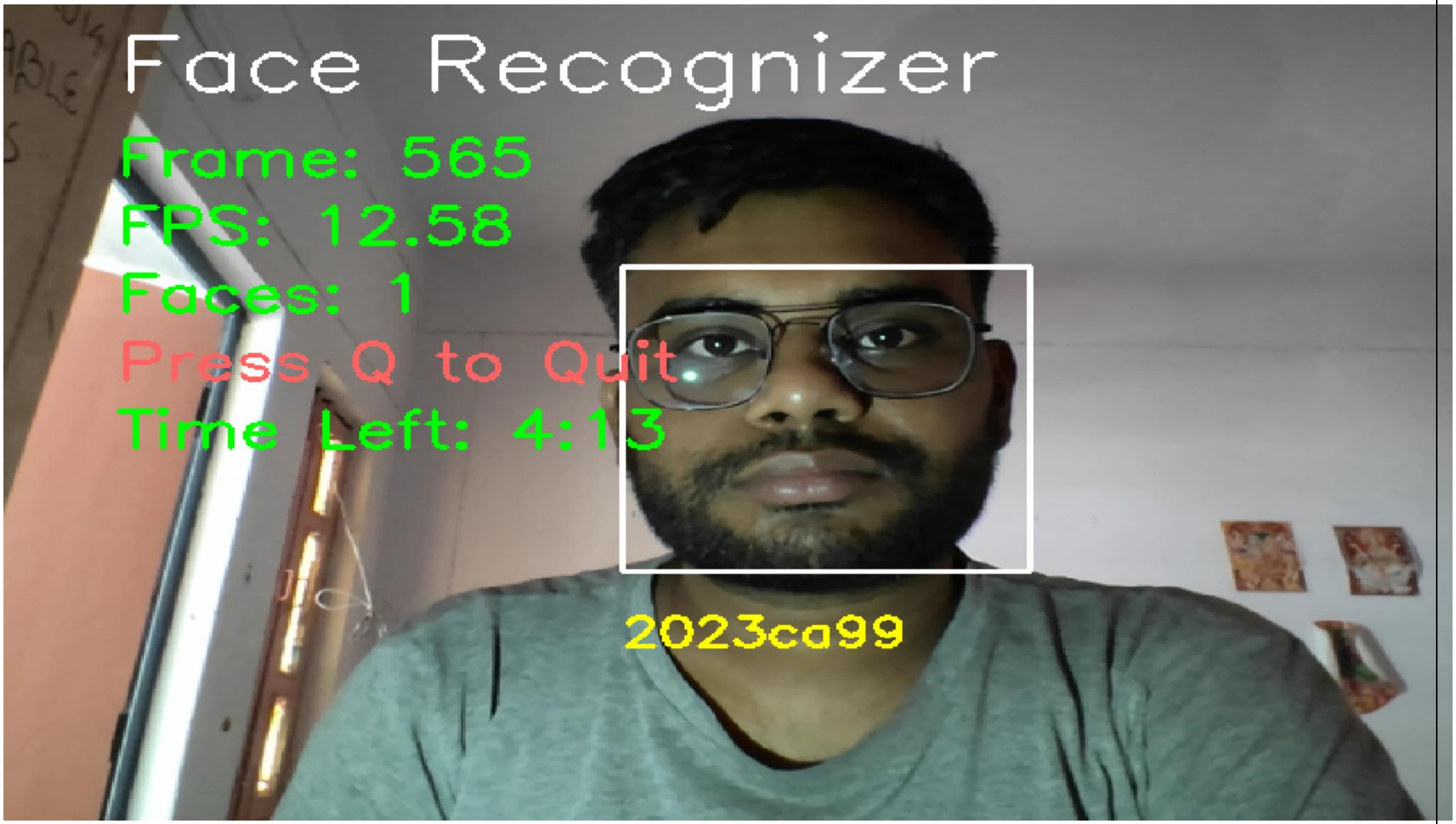
A USB or Raspberry Pi-compatible camera for face recognition.

7. **Internet Connection:**
    - Offline functionality: Works offline for face recognition and attendance marking.
    - Internet (Optional): Only needed for sending emails with attendance files.
8. **Other Hardware**:
    - Raspberry Pi 4 Model B (recommended) for portability, cost-effectiveness, and offline operations.
    - Display: A basic monitor or screen for displaying the UI.

Chapter 6

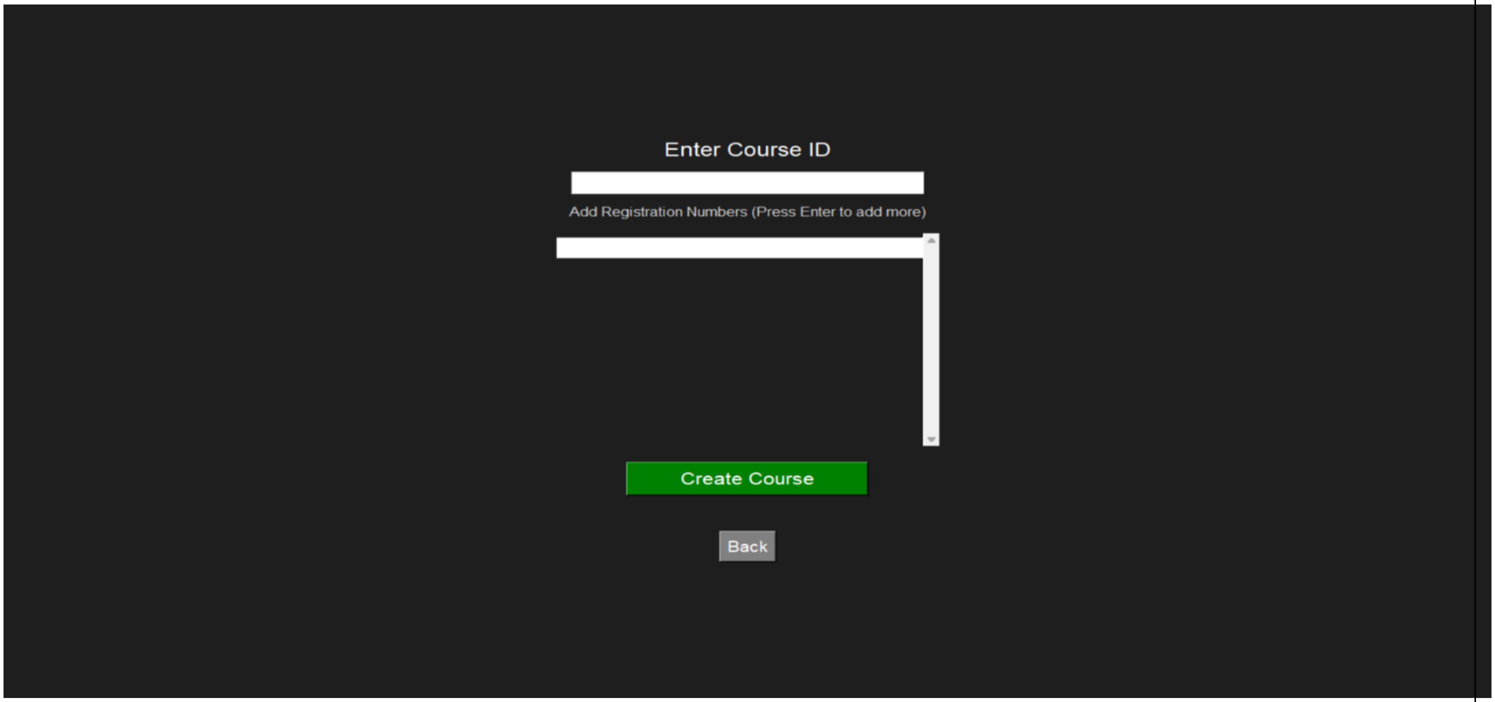
6.1 Screenshots (Frontend and Backend)

6.1.1 Attendance page

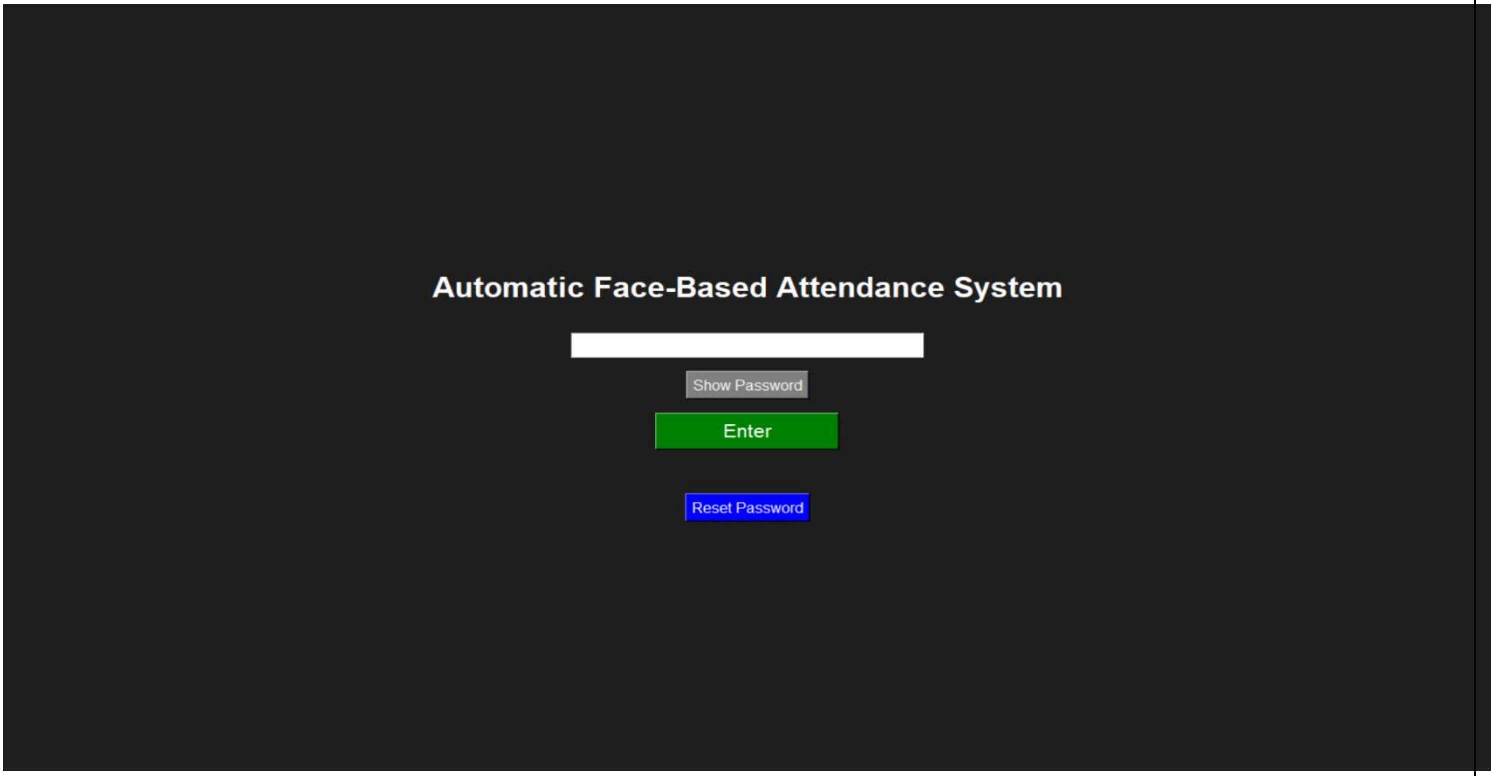


```
Face Recognizer
Frame: 565
FPS: 12.58
Faces: 1
Press Q to Quit
Time Left: 4:13
2023ca99
```

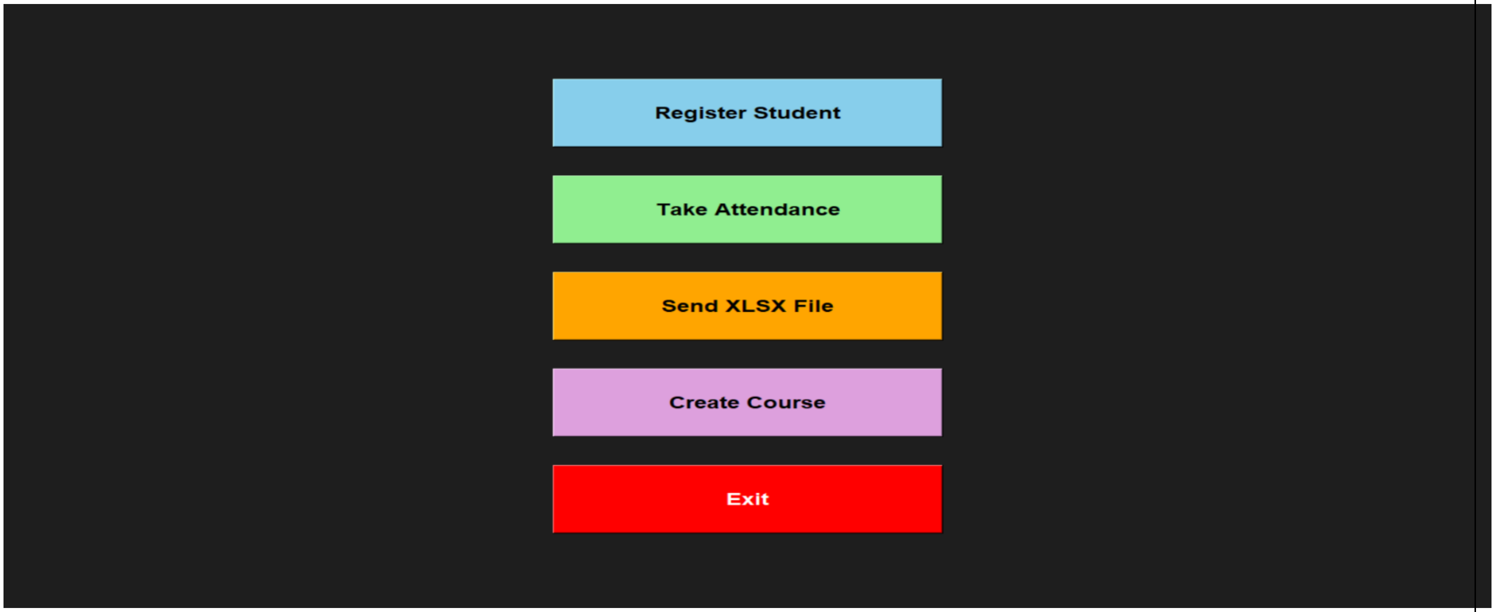
6.1.2 create_course



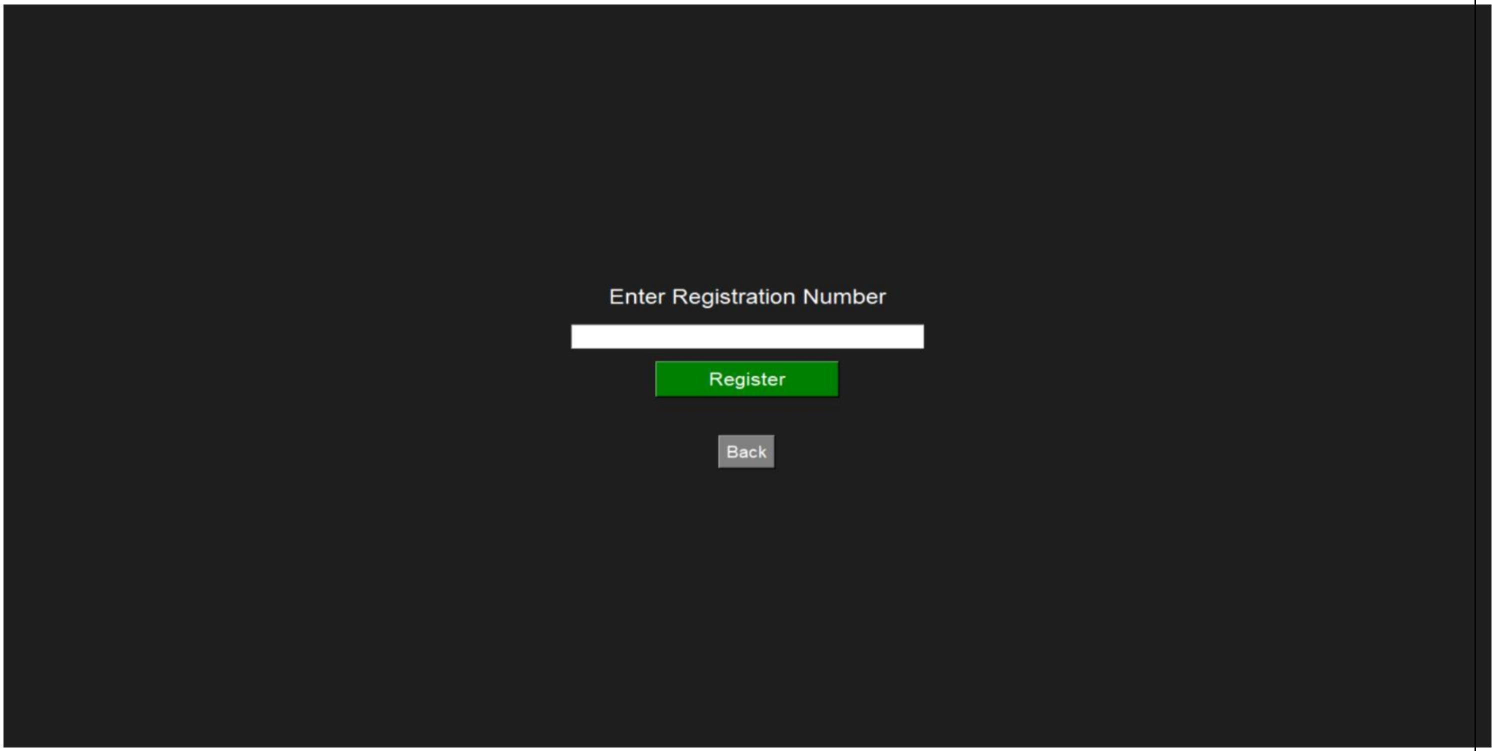
6.1.3 home_page



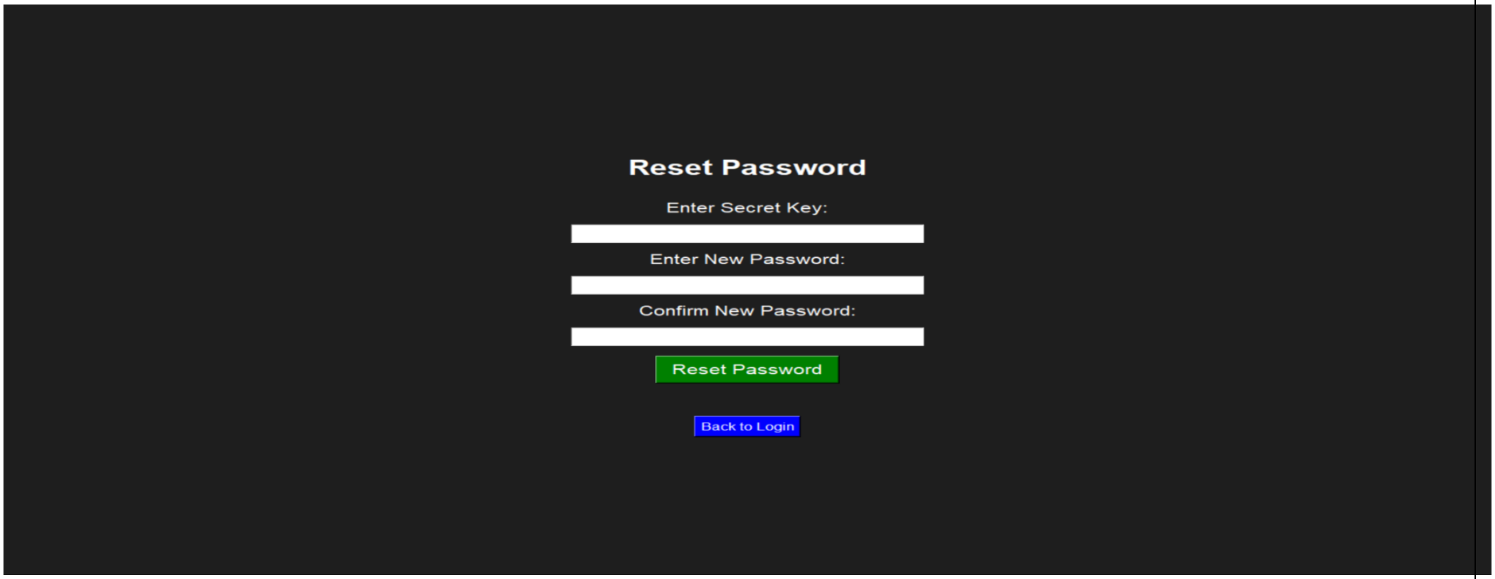
6.1.4 Main Page



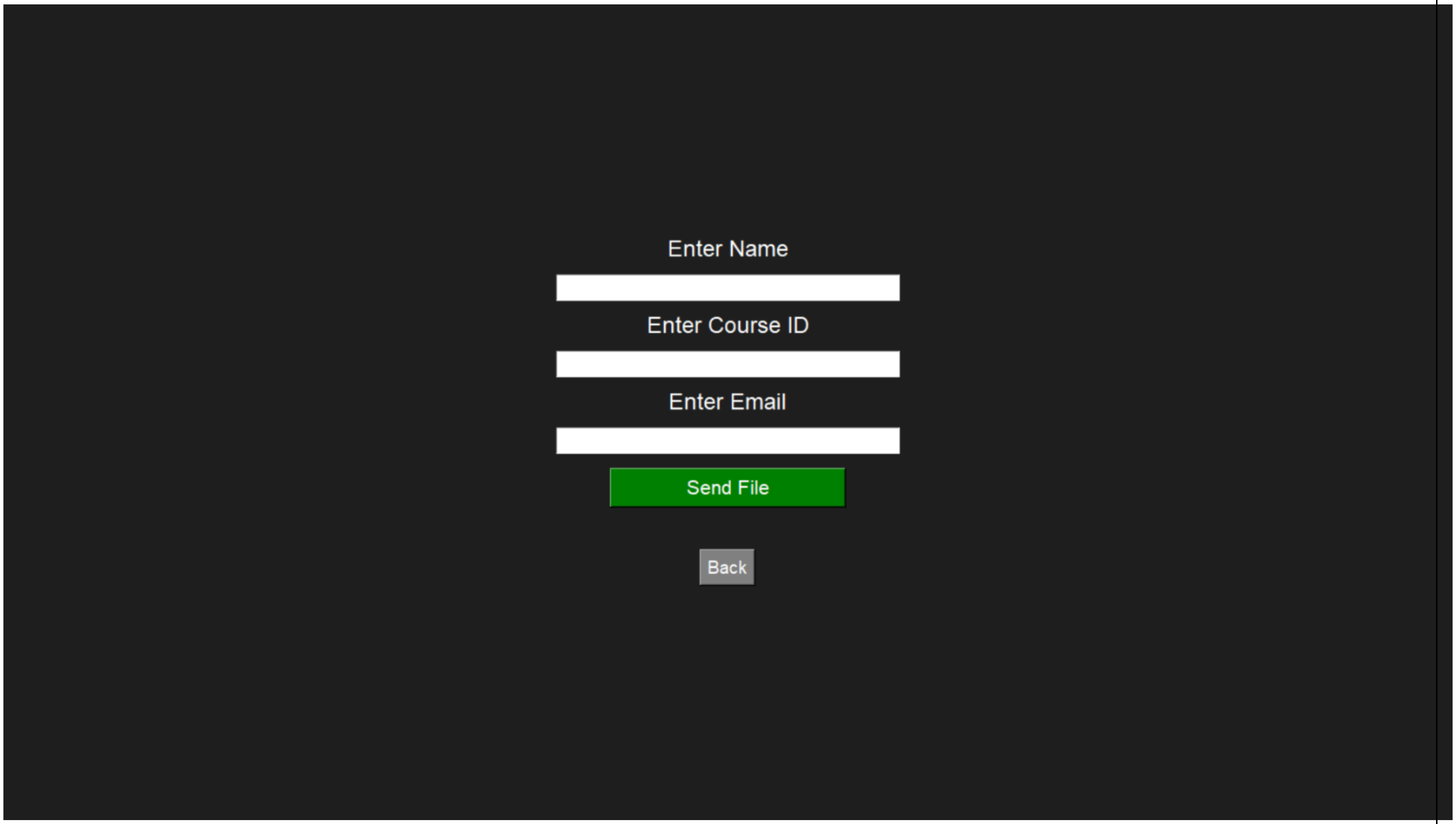
6.1.5 Register Student



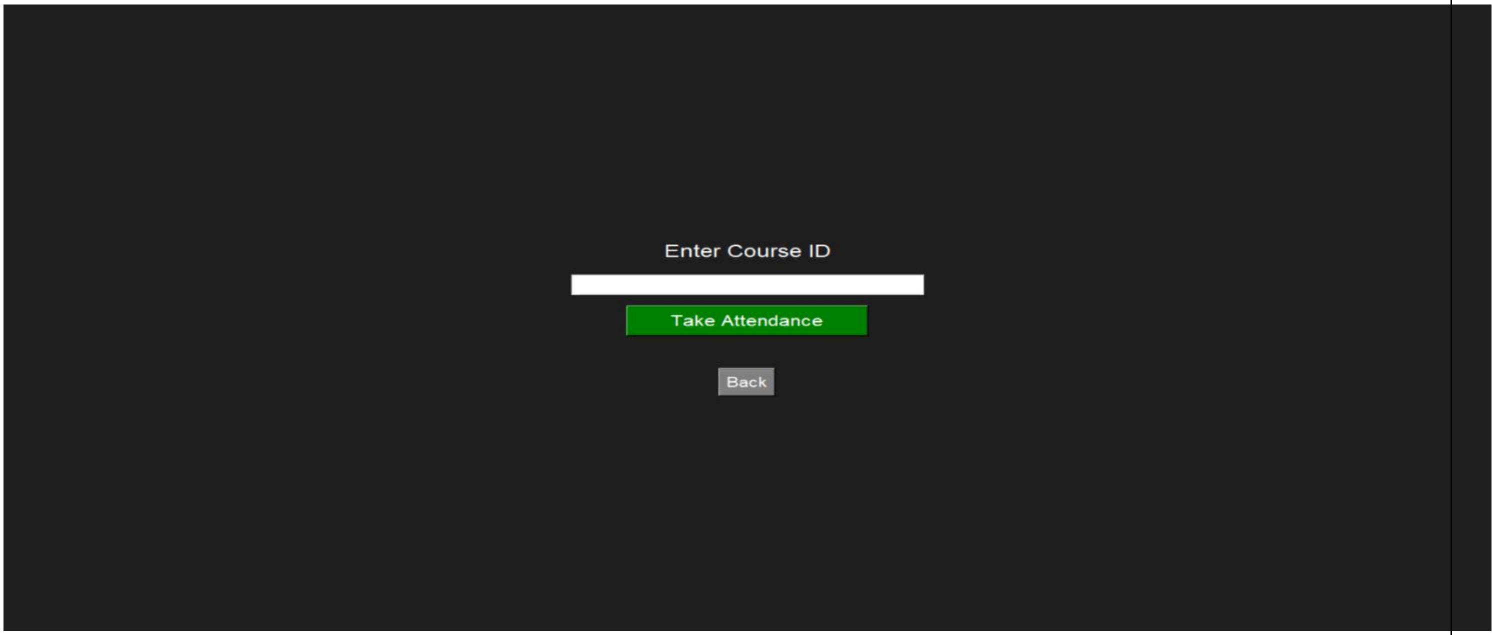
6.1.6 Reset Password



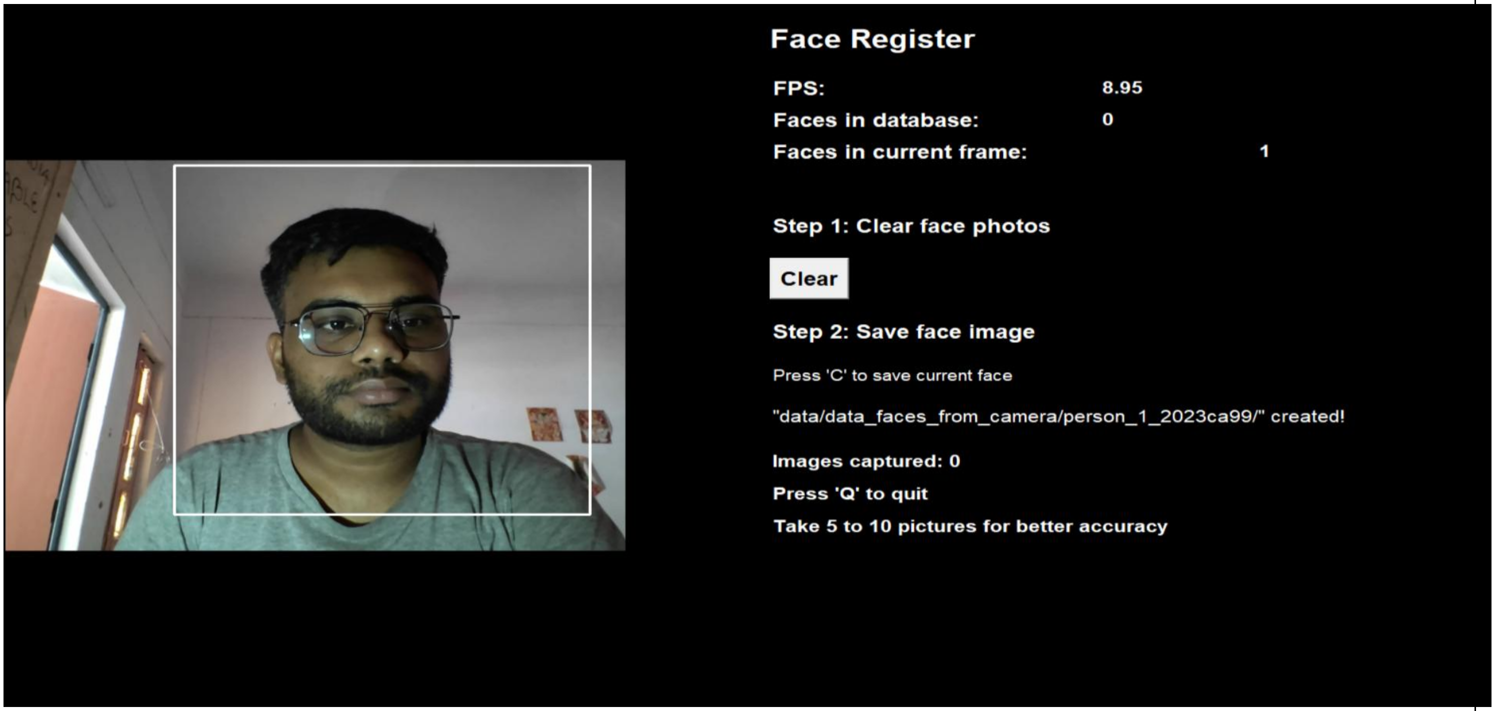
6.1.7 Sending Mail



6.1.8 Starting Attendance



6.1.9 Student_Register_image



Face Register

FPS: 8.95

Faces in database: 0

Faces in current frame: 1

Step 1: Clear face photos

Step 2: Save face image

Press 'C' to save current face

"data/data_faces_from_camera/person_1_2023ca99/" created!

Images captured: 0

Press 'Q' to quit

Take 5 to 10 pictures for better accuracy

Chapter 7

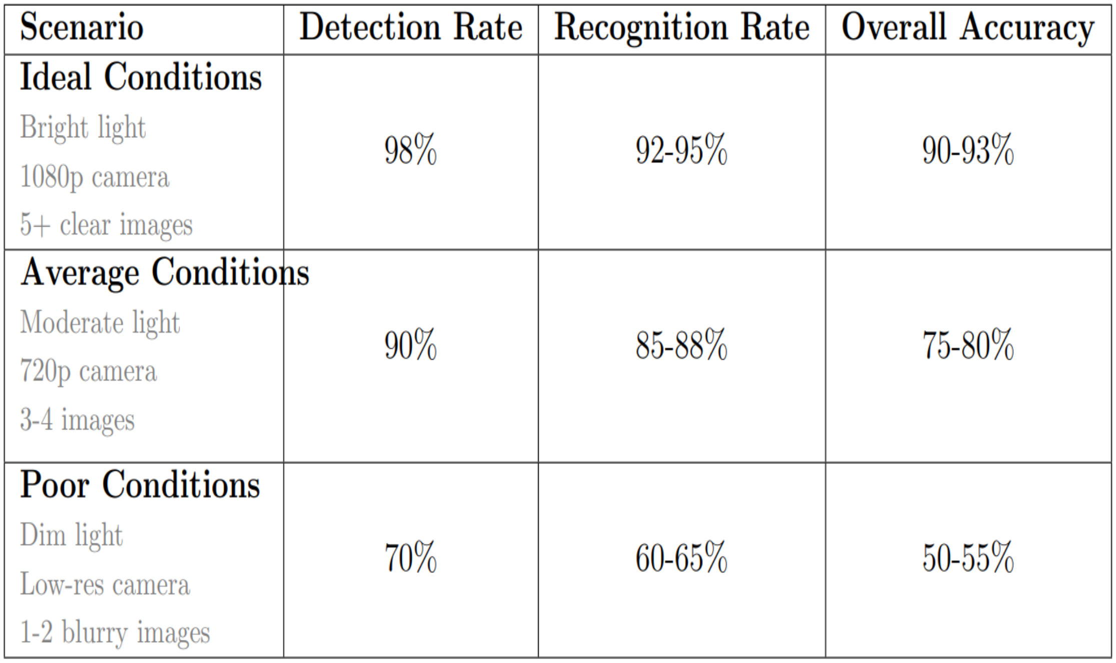
Comparison with Other Market Solutions

Feature	Our System	Biometric Devices (e.g., Fingerprint)	RFID Attendance	Cloud Face Attendance Apps
Touchless	Yes	No	Yes	Yes
Face-Based	Yes (Dlib-128D)	No	No	Yes
Offline Use	Full support	Yes	Yes	Needs Internet
Course-wise XLS Generation	Built-in	No	No	Some
Custom Voice Announcement	Yes	No	No	Yes
Student Registration UI	Simple UI	Manual setup	Tag-only	Web-based
Cost-Effective	Very Low (Pi-based)	High	Medium	Monthly fee

Chapter 8

Accuracy, Conclusion and Future Work

8.1 Accuracy and Conclusion



Scenario	Detection Rate	Recognition Rate	Overall Accuracy
Ideal Conditions Bright light 1080p camera 5+ clear images	98%	92-95%	90-93%
Average Conditions Moderate light 720p camera 3-4 images	90%	85-88%	75-80%
Poor Conditions Dim light Low-res camera 1-2 blurry images	70%	60-65%	50-55%

Accuracy Table of Our SmartAttendance.Pi

In conclusion, the "Automatic Face Attendance System" presents a modern, privacy-focused, and efficient solution for attendance management in classroom environments. By integrating facial recognition technology with offline capabilities, the project successfully addresses common challenges faced in manual and semi-digital attendance systems, particularly in areas with limited internet connectivity.

Throughout the development of this system, we have leveraged powerful libraries and tools such as OpenCV, dlib, Tkinter, and pandas to design an end-to-end application that supports student registration, facial feature extraction, attendance tracking, and report generation. Furthermore, features such as voice announcements, course-wise record keeping, and automatic emailing of attendance sheets enhance the practicality and usability of the system.

The user-friendly graphical interface built using Tkinter ensures ease of access for educators with limited technical experience, while the system's offline-first architecture supports deployment in resource-constrained institutions or remote locations. By using Raspberry Pi as a target platform, the solution remains portable, cost-effective, and scalable.

Ultimately, this project not only reduces the administrative burden of taking attendance but also increases accuracy, saves time, and provides a secure and privacy-aware method of student tracking.

8.2 Future Work

While the system already offers a robust feature set, several improvements and expansions are planned for future versions:

- Real-Time Face Recognition: Enhance performance by integrating real-time face recognition with improved latency and multi-threading support.
- Attendance Analytics Dashboard: Develop an analytics interface showing trends, student-wise attendance graphs, and alerts for irregular attendance patterns.
- Mobile App Integration: Create a mobile application version to allow teachers to initiate attendance from their smartphones or tablets.
- Cloud Backup (Optional): Provide optional cloud synchronization to back up attendance records securely, with admin-level access controls.
- Multi-Language Support: Extend the system to support regional languages in both GUI and voice announcements.
- Admin Panel and Authentication: Introduce user roles and login systems to restrict access and enhance security.
- Integration with Biometric Devices: Add support for additional identification methods like RFID or fingerprint readers for hybrid authentication.
- Model Optimization: Improve face recognition accuracy and speed, especially in varied lighting or crowded environments.
- Notification System: Enable automatic SMS/email notifications to guardians for absentees or late arrivals.

References

[1] “OpenCV (Open Source Computer Vision Library)” [Online]. Available: https://opencv.org/
[2] “dlib C++ Library” [Online]. Available: http://dlib.net/
[3] “Python” [Online]. Available: https://www.python.org/
[4] “Tkinter GUI Programming” [Online]. Available: https://docs.python.org/3/library/tk.html
[5] “Pandas - Python Data Analysis Library” [Online]. Available: https://pandas.pydata.org/
[6] “NumPy” [Online]. Available: https://numpy.org/
[7] “Pillow (PIL Fork)” [Online]. Available: https://python-pillow.org/
[8] “OpenPyXL - Excel for Python” [Online]. Available: https://openpyxl.readthedocs.io/
[9] “pyttsx3 - Text to Speech Conversion Library in Python” [Online]. Available: https://pypi.org/project/pyttsx3/
[10] “Python-dotenv” [Online]. Available: https://pypi.org/project/python-dotenv/
[11] “GitHub” [Online]. Available: https://github.com/
[12] “Stack Overflow” [Online]. Available: https://stackoverflow.com/
[13] “GitHub Repository” [Online]. Available: https://github.com/itsnarutouzumaki/automatic_attendance_system