Tribhuvan University

Faculty of Humanities and Social Sciences

A Project Report On

Student Attendance Management System

Submitted to:

Department of Computer Application

Nepal Kasthamandap College, Kalanki

In partial fulfillment of the requirements for the Bachelor in Computer Application

Submitted by:

Nischal Bhattarai [Reg no: 6-2-144-12-2021]

July 2025

Under the Supervision of:

Shyam Maharjan

Tribhuvan University

Faculty of Humanities and Social Sciences

Nepal Kasthamandap College

Kalanki, Kathmandu

Bachelor in Computer Application (BCA)

Supervisor's Recommendation

I hereby recommend that this project prepared under my supervision by **Nischal Bhattarai** entitled "**Student Attendance Management System**" in the Partial Fulfillment of requirement for the degree of Bachelor in Computer Application is recommended for that final evaluation.

Shyam Maharjan

Project Supervisor

BCA Department

Nepal Kasthamandap College

Tribhuvan University

Faculty of Humanities and Social Sciences

Nepal Kasthamandap College

Kalanki, Kathmandu

Bachelor in Computer Application (BCA)

Letter of Approval

This is certified that this project prepared by Nischal Bhattarai entitled "Student Attendance Management System" in the Partial Fulfillment of requirement for the degree of Bachelor in Computer Application has been evaluated. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

_____   _____

Syham Maharjan   Er. Sujan Poudel

Supervisor   Program Coordinator

BCA Department   Nepal Kasthamandap College

Nepal Kasthamandap College

_____   _____

Internal Examiner   External Examiner

Roshan Poudel   Ananda Kc

BCA Department   R.R Campus

Abstract

It is a Student Attendance Management System with facial recognition features to make attendance automatic, convenient, and faster. The students are able to log in by utilizing a webcam that checks their face to confirm their presence. The system has a complete admin panel through which the administrators can manage students, create new student records, manage classes, set timetables, and check attendance reports. Students are able to view their own attendance records from their individual dashboards, which makes it transparent and readily accessible. The system aims at enhancing accuracy, reducing manual effort, as well as preventing attendance cheating.

Keywords: Student Attendance Management System, Facial Recognition, Face Identification. Automated Attendance, Webcam Attendance, Admin Dashboard, Student Portal, Digital Attendance System

Acknowledgement

I will like to thank our supervisor **Mr. Shyam Maharjan** a lot. He provided us with the opportunity to complete the project Student Attendance Management System. Through this project, I gained knowledge on new technologies such as facial recognition and web-based systems. I am very grateful to him for his valuable suggestions and guidance. I would like to thank our BCA Program Coordinator, **Er. Sujan Poudel**, for guiding us to become professionals and good human beings. Without his support, it was very difficult to complete this project. I would like to thank our friends and families for always motivating us and assisting us in completing this project within the given time. I also want to thank everyone who assisted in any way to make this project a success.

Yours Sincerely,

Nischal Bhattarai

Table of Contents

List of Figures

List of Tables

List of Listings

List of Abbreviations

Abbreviation	Description
CSS	Cascading Style Sheet
DBMS	Database Management System
SAMS	Student Attendance Management System
HTML	Hypertext Markup Language
XAAMP	X-platform, Apache, MySQL, PHP,Perl
JS	JavaScript
MySQL	Structured Query Language

Chapter-1:Introduction

1.1. Introduction

Traditional method of attendance marking is time-consuming activity in majority of the schools and colleges. It is also extra workload to the faculties who have to mark attendance by roll calling student's names which will take about 5 minutes of entire session. It is time consuming. There are some chances of proxy attendance. Therefore, many institutes have started utilizing many other means for marking attendance like utilization of Radio Frequency Identification (RFID) iris scan, fingerprint scan [1] etc. These systems are queue based and can be time-consuming as well as intrusive in nature. Face recognition has established a significant biometric characteristic, which can be readily attainable and non-obtrusive. Face recognition dependent systems are comparatively insusceptible to different facial expression. Face recognition system involves two categories: verification and face identification. Face verification is a 1:1 matching process, it is a face image compared to the template face images and whereas is a 1: N problems which is compared a query face image. In this paper, I have put forward a system that identifies students' faces from live streaming video of classrooms and attendance will be marked if the identified face exists in the database. The new system will be faster than the traditional method.

1.2 Problem Statements

In educational institutes, accurate attendance records of students have a big role while tracking performance, discipline, and school requirements. Dependance over old register or excel attendance records is time-wasting, error-prone, and time-consuming and Inefficient Traditional Methods. Some of the lists are pointed out below.

- No Real-Time Attendance Monitoring
- Possibility of False or Manipulated Attendance
- Difficulty in Generating Reports of student

1.3 Objectives

The primary objectives of a student attendance management system using facial recognition are to reduce attendance marking time, enhance accuracy, eliminate proxy attendance, and ensure a non-intrusive, efficient solution for educational institutions. Here are the key objectives:

- To develop a face recognition-based attendance system.

- To Minimize the time required for attendance marking.

- To Enhance accuracy and reduce proxy attendance.

1.4 Scope and Limitations
1.4.1 Scope
Some Scope of SAMS are as follow.

- It allows to take attendance of student digitally.

- Student can view attendance history.

- Student can view announcement and routine.

- Admin can manage student.

- Admin can manage classes, routines, and announcements.

1.4.2 Limitations
Some Limitations of SAMS are as follow.

- Proxy attendance

- Low accuracy under poor lighting or occlusion

- Limited Users Roles

- No GPS or Location Verification

1.5 Development Methodology

I have used Agile methodology while building the Student Attendance Management System.

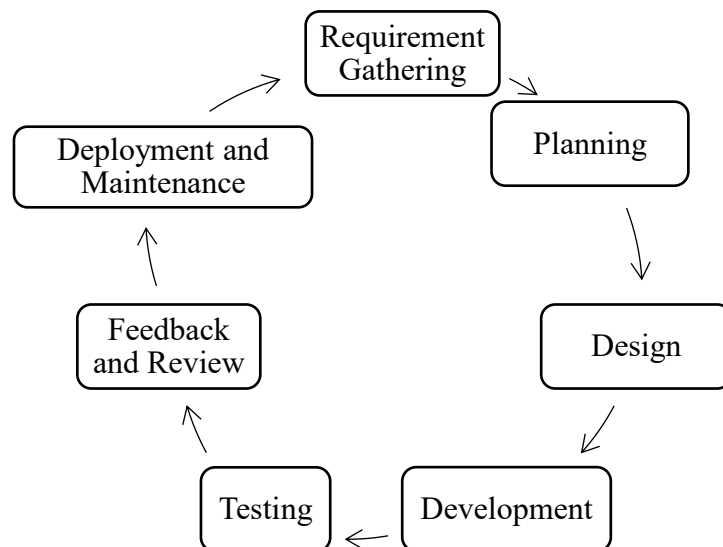This approach helped streamlined development through iterative process and feedback.

Figure 1.1 Agile Method for SAMS

The following illustration is a representation of the different phases of the Agile methodology applied in this system:

1. Requirement Gathering: - Collected initial ideas and needs from stakeholders like students, faculty, and admin.
2. Planning: - Broke down features into user stories (e.g., face detection, marking attendance, generating reports).
3. Design: - Designed basic UI for login, dashboard, and attendance tracking.
4. Development (Iterations/Sprints): - Developed the system in small functional increments, adding and testing features sprint by sprint.
5. Testing: - Regularly tested modules like login, attendance marking, and reporting after each sprint.
6. Feedback & Review: - Collected feedback after each iteration and made necessary improvements.
7. Deployment & Maintenance: - Deployed a working version and continuously improved the system based on user input.

1.6 Report Organization
Introduction:

This chapter provides an overview of the Student Attendance Management System, highlighting its purpose, key objectives, and the problems it aims to solve through automation using facial recognition.

Background Study and Literature Review:

This chapter includes previous studies and research done in the area of automated attendance systems using facial recognition, along with technologies and tools used in similar systems.

System Analysis and Design:

This section covers the system's functional and non-functional requirements, feasibility study, ER diagram, DFD, Gantt chart, database schema, and the overall design and architecture of the system.

Implementation and Testing:

This chapter explains the tools and technologies used for implementation, development methodology, system components, and the results of unit and system testing with defined test cases.

Conclusion and Future Recommendation:

This final chapter summarizes the outcomes and lessons learned during the project. It also discusses the current system's limitations and suggests improvements for future development.

Chapter-2: Background Study and Literature Review

2.1. Background Study

Automated attendance systems using facial detection and recognition technology have been developed and implemented in various environments such as educational institutions, workplaces, and event venues. These systems utilize biometric data, specifically facial features, to identify individuals and monitor their attendance, offering higher accuracy and efficiency compared to traditional approaches like sign-in sheets or roll calls. One proposed system for automated attendance tracking in classroom settings captures student's images upon entering the classroom and employs a facial recognition algorithm to identify and record their attendance. This system is designed to be user-friendly and efficient, utilizing commonly available devices. By continuously analyzing facial information, this method aims to enhance the accuracy and efficiency of existing attendance tracking methods [2].

Facial detection and recognition technology has the potential to transform conventional attendance tracking methods by offering higher accuracy and efficiency. However, there are potential challenges and concerns associated with using this technology, such as privacy concerns and the potential for bias in the recognition algorithms. These concerns must be addressed to ensure the ethical and responsible use of this technology. Overall, automated attendance systems utilizing facial detection and recognition technology represent a significant advancement toward streamlined and intelligent attendance management in various environments. Further research and development are needed to address potential challenges and concerns associated with this technology and ensure its ethical and responsible use [3].

2.2. Literature Review

Face recognition is a technology created by Woodrow Wilson Bledsoe in 1966 that works to match human faces through digital images or video footage through a facial database. Face recognition became an idea to allow computers to find and recognize human faces quickly and precisely. Many algorithms have been developed to improve the performance of face recognition [4].

A student's attendance at his institution has an impact on his overall academic success. Calling students' names or getting their signatures on paper are the two main traditional ways to take attendance. They were both more ineffective and time-consuming [5].

This proposed system uses facial recognition technology to create an automated system that tracks students' attendance instead of the traditional ways. This work's primary goal is to make the system for managing and recording attendance effective, time-efficient, straightforward, and simple [6].

In order to take attendance, students' faces are recognized using face biometrics that are based on a high-definition monitor. They took more time and were more challenging. video in addition to other information technology [7].

In this facial recognition project, a computer system will locate faces. can accurately and swiftly identify faces in images or videos captured by a surveillance system. Numerous methods and tactics have been developed to improve facial recognition performance [8].

Therefore, in order to help instructors automatically maintain attendance data, a computer-based student attendance management system is needed. The attendance system is a typical example of this transition, starting from the traditional signature on a paper sheet to face recognition [9].

The principal reason this system has been put forward is to improve the traditional attendance system of various universities to avoid the misuse of time and assets. The pointing-sides of automation world have forced an idea of switching from standard attendance to the digital system by using face detection and recognition methods. The major reason of building this system is to improve the adaptability and performance of the attendance system procedure besides reducing the long-term time load, work and disposables used. The main purpose of the Student Attendance markup structure is to perform, adding and manipulating attendance notes of an individual, automatic calculation on number of presented and absentees based on subject and affability of the class and then generates the automated document or spreadsheet [10].

Using the concept of open computer vision, this idea is entirely dependent on the general-purpose language known as Python. The LBPH model was utilized for face recognition and Haar cascade for face detection. After individual student training, the system creates a spreadsheet that shows the number of pupils in the classroom together with a live video or image capture [11].

Teachers can utilize smart cards that track attendance data to keep tabs on their students. By matching biographical information to student identity card entries, biometric technology

enables schools to confirm student attendance. School administrators may keep accurate attendance records for every student with the use of an effective attendance system. However, when only a small percentage of the student body at the school has the necessary IDs, no system functions well. [12].

The Attendance Management System using Haar cascade and OpenCV is a Python-based project that aims to automate the process of attendance management in various settings such as schools, colleges, or workplaces. The system utilizes computer vision techniques, specifically Haar cascade and OpenCV, to detect and recognize faces from images or video streams, enabling accurate and efficient attendance tracking [13].

Chapter-3: System analysis and design

3.1 System Analysis

This chapter describes the different fact-finding techniques used to achieve the goals and objectives of the project, such as Population of the Study, Data Collection and Analysis, System Analysis, System Design and Implementation, Testing, and Validation.

3.1.1 Requirement Analysis

Project has the following functional and non-functional requirements.

i. Functional Requirement

For User:

- The system should allow user to register.
- The system should manage user email and password.
- The system should allow user to get attendance.
- The system should allow user to view attendance

For Admin:

- The system should allow the system administrator to login and logout from the system.
- The system should allow monitoring the system user's data.
- The system should allow admin to add, manage student.
- The system should allow admin to add announcement, make changes in classes.
- The system should allow admin to add make a attendance report of a student.
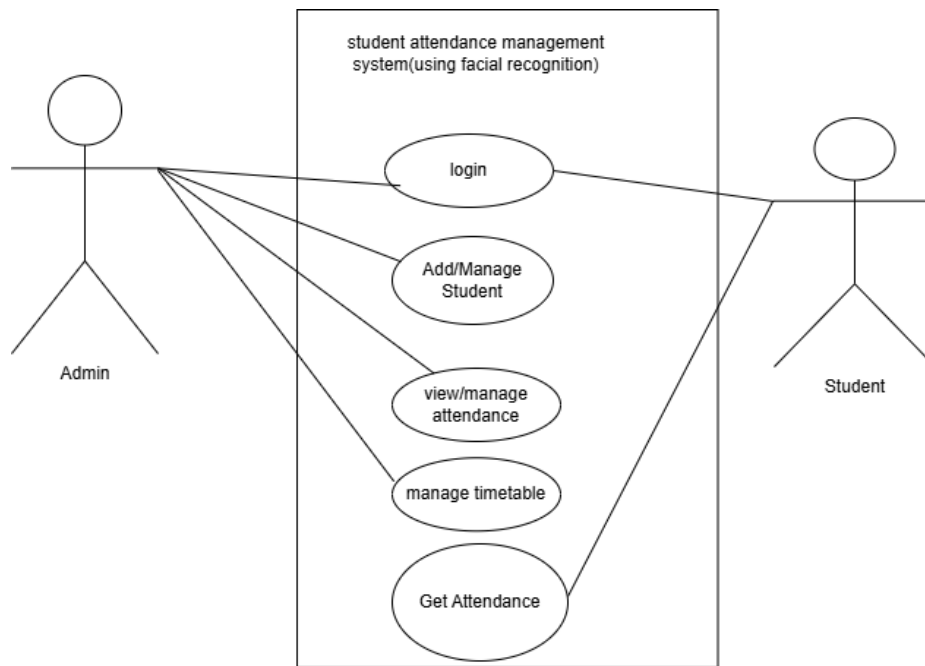- The system should allow admin to manage routine of a student semester wise.

Figure 3.1 Use Case Diagram for SAMS

In this figure 3.1 it shows the use case diagram for admin. In this project admin can manage login view/manage attendance manage timetable and add student. In student part they can login the page and capture the attendance.

ii. Non-Functional Requirement

• Usability: The system provides an intuitive user interface, ensuring easy access for students.

• Portability: The website is portable as it is an online website running across the net.

• Flexibility: It can be modified or expanded to accommodate new features, such as integrating additional biometric authentication methods, supporting multiple campuses, or adapting to different academic policies. Additionally, the system can handle varying numbers of students and staff without significant changes to its core functionality, making it scalable and efficient.

3.1.2 Feasibility Analysis

1. Economic Feasibility

This project is economically feasible as it eliminates the need for manual attendance registers, reducing paper usage and administrative workload. The investment in a computer system with a camera is a one-time cost, significantly lowering operational expenses. Additionally,

open-source technologies such as Python, OpenCV, and MySQL are used, making the project cost-effective.

2. Technical Feasibility

The proposed system is technically feasible as it utilizes readily available and widely used technologies such as:

- Operating System: Windows
- Programming Language: Python, HTML, CSS, JS, PHP
- Database: MySQL
- AI & Machine Learning
  - OpenCV: - Utilized for image processing and face detection.
  - Face Recognition: - Used for comparing and identifying student faces for accurate attendance marking.

3. Operational Feasibility

The system is operationally feasible because it enhances security, eliminates proxy attendance, and automates record-keeping, making the attendance process more efficient and reliable. It does not require extensive training, as the interface is user-friendly and easy to navigate. Teachers and administrators can manage attendance records with minimal effort, reducing errors and increasing overall efficiency.

4. Schedule Feasibility

Table 3.1 Gantt Chart

Task/Date	12th February	17th March	25th April	6th June	20th July
planning	�				
Requirement gathering	▮▮				
Sprint 1-Basic UI and DB Setup		▮			
Testing Sprint 1			▮		
Sprint2-Facial Recognition Module			▮		
Testing Sprint 2				▮	
Sprint3- Attendance Reports and UI Enhancement				▮	
Final Testing and Deployment					▮
User Feedback and Continuous Improvement					▮
Documentation	▮▮▮▮▮▮▮▮▮▮				

A Gannt Chart illustrates a project schedule. In this Gantt chart, each row represents a task involved in the development of the student attendance management system. The start date and end date for each task is shown, as well as the duration. The Gantt chart visualizes the timeline for each task and shows the overlap between the separate phases of the project.
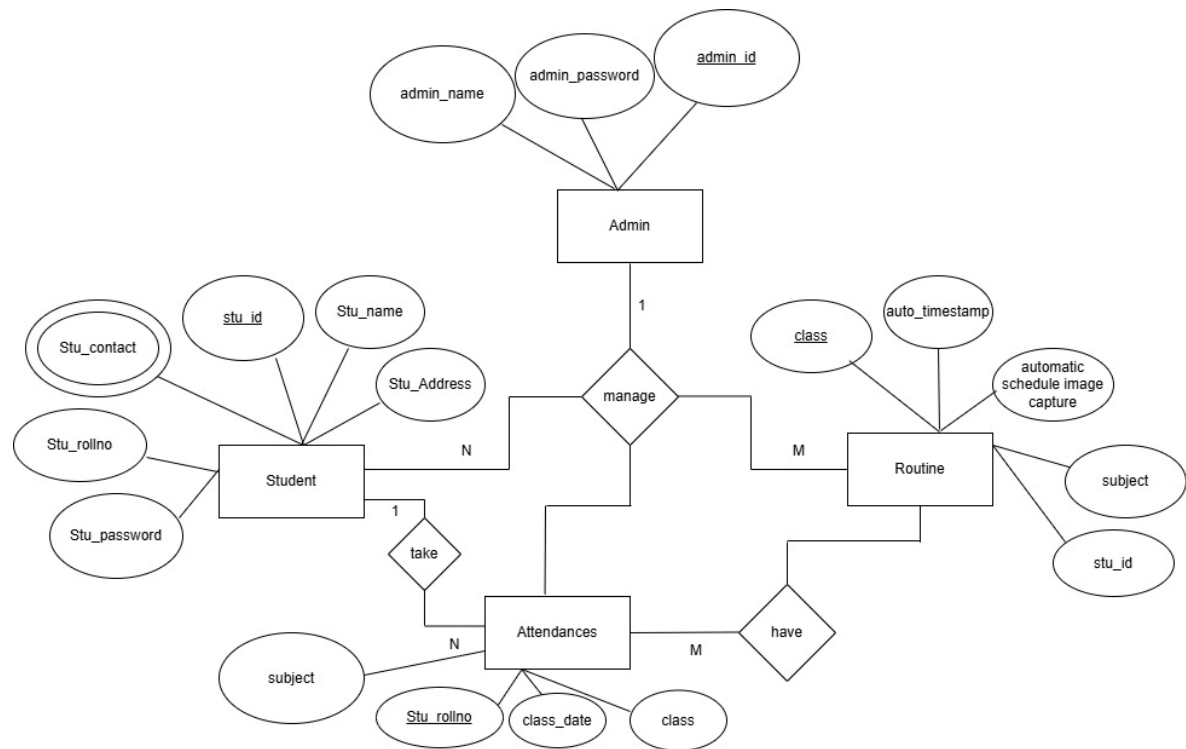
3.1.3 E-R Diagram

Figure 3.2 ER Diagram for SAMS

This ER diagram is for a Student Attendance Management System, where an admin manages both students and professors. The professors take attendance of students in various subjects and classes. Student details are name, roll number, contact, and address, while professors log in using their own login credentials. The attendance records have information of which student attended a class on what date. The system offers efficient tracking of students' attendance, allowing the admins to monitor activities.
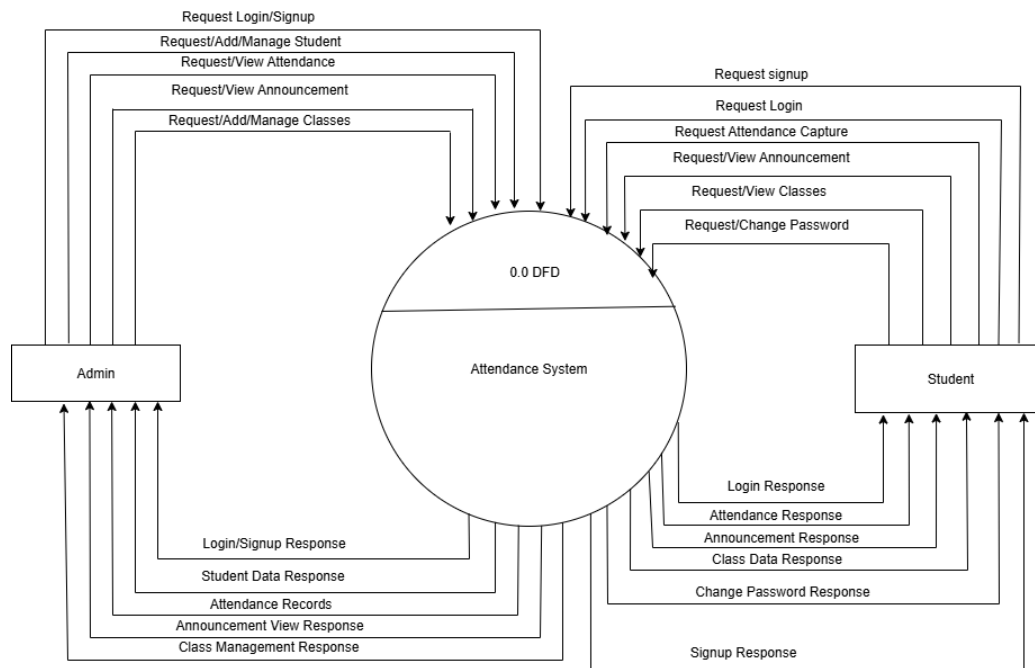
3.1.4 Process Modeling

Figure 3.3 Level 0 DFD for SAMS

The image represents a 0-level Data Flow Diagram (DFD) for a Student Attendance Management System at the Context Level. It gives an overview of the entire system, showing how external entities interact with the system. This is the main system that processes attendance data. Student (Left side): Interacts with the system by logging in and providing attendance. Admin (Right side): Logs in and manages/view attendance records.

In this Level 1 DFD, I have seven main processes that make the Student Management System work smoothly. These are: Registration (1.0), which allows new students to sign up and save their information securely in the d_users database, and Login (2.0), which verifies user details and gives them access to the system. The Announcement (3.0) process lets staff or admins post important messages for students, while Classes (4.0) handles viewing and updating class schedules and information. The Attendance (5.0) process makes it easy to record and review student attendance, making sure this data is always accurate and available. The Students (6.0) process takes care of all student-related information making it simple to view, add, or update profiles. Meanwhile, the Change Password (7.0) process allows students to securely reset or update their account passwords when needed. The system has two external entities Student and Admin and a single data store called d_users, which contains all user and student data.

Together, these elements create a well-structured, easy-to-use, and secure Student Management System for both students and administrators.
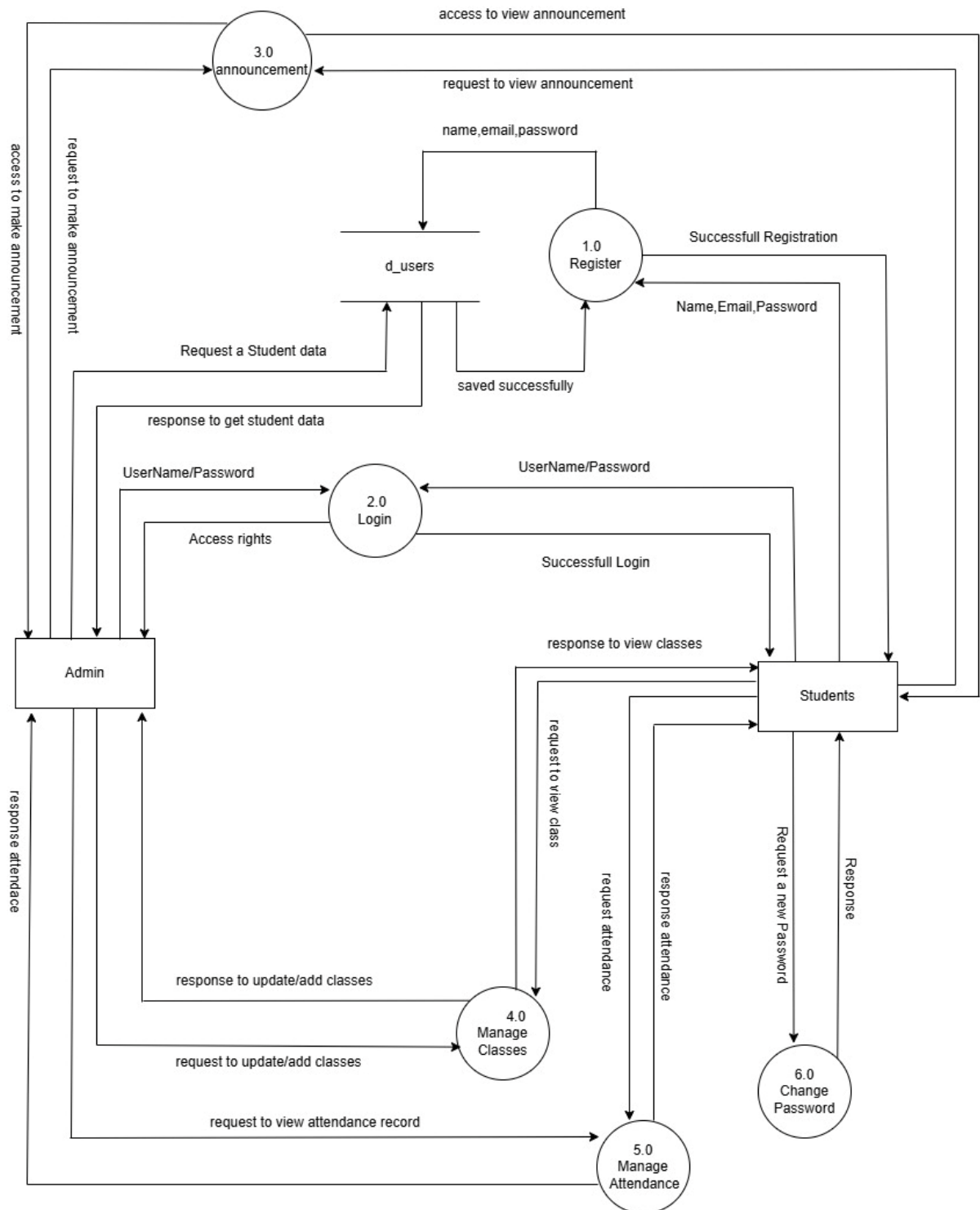

Figure 3.4 Level 1 DFD for SAMS

3.2 System Design

To realize the different functional requirement of the system in graphical form, different design diagram of the system has been prepared which are as follow:

3.2.1 Architectural Design

For the system, three tier architecture is used which includes user interface, web server and database. In architectural design, basic structure of the system is shown:
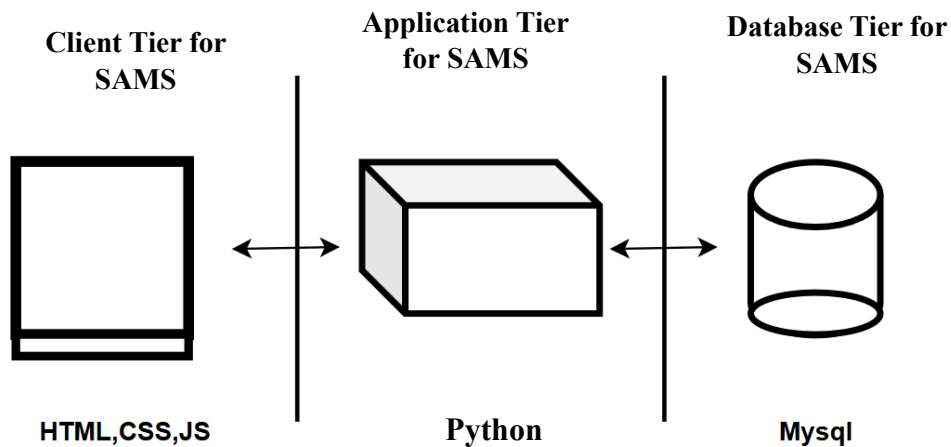
Client Tier for SAMS   **Application Tier for SAMS**   **Database Tier for SAMS**

HTML,CSS,JS   **Python**   **Mysql**

Figure 3.5 Architectural Design for SAMS

The system follows a three-tier architecture. The Client Tier is developed using HTML, CSS, and JavaScript, which provides the user interface. The Application Tier uses Python to handle the core logic and communication between the client and the database. The Database Tier uses MySQL to store and manage data such as student details, attendance records, and class information.

3.2.2 System Flowchart
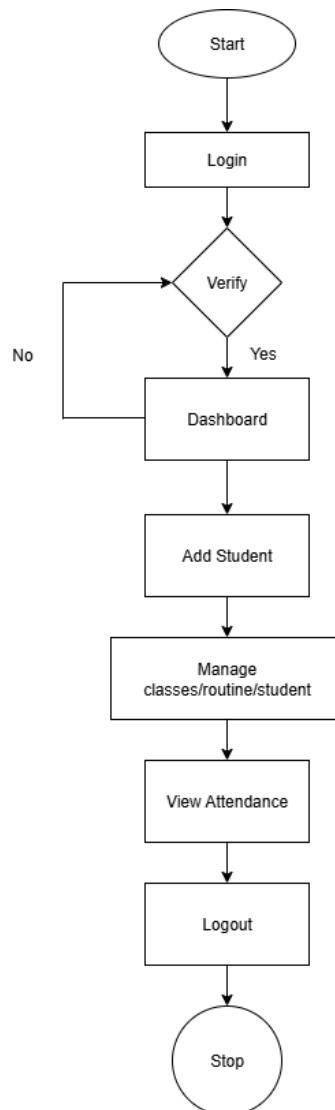
Admin module flowchart

Figure 3.6 Flowchart for Admin for SAMS

The admin flowchart outlines the process for managing an admin system. It begins with the admin initiating the process by clicking "Start". The admin then proceeds to the "Login" step, where credentials are entered. Following this, the system attempts to "Verify" the login details. If the verification fails ("No"), the admin is directed back to the "Login" step to try again. If the verification is successful ("Yes"), the admin is granted access to the "Dashboard". From the dashboard, the admin can perform several key tasks: "Add Student", "Manage classes/routine/student", and "View Attendance". After completing all necessary tasks, the admin chooses to "Logout", bringing the process to an "Stop".
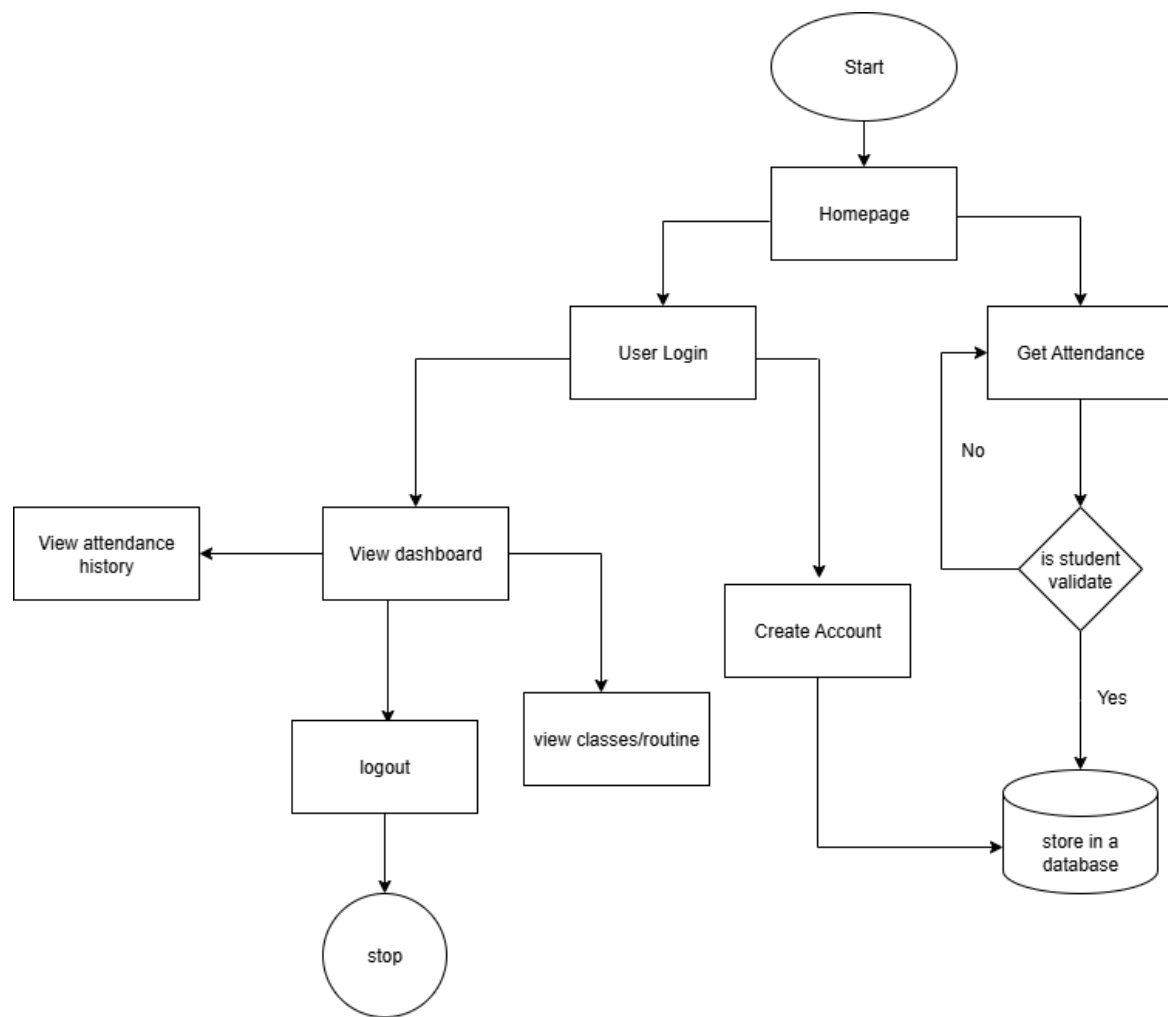
Client module flowchart

Figure 3.7 Flowchart for User for SAMS

The student flowchart illustrates the process for students interacting with the system. It begins with the student clicking "Start," which leads to the "Homepage." From the homepage, students have two main options: "User Login" or "Get Attendance."

If the student chooses "User Login," they can proceed to "View dashboard." From the dashboard, they can "View attendance history" or "view classes/routine." Alternatively, if they don't have an account, they can "Create Account," which then leads to storing information "Store in a database." After completing their tasks via the dashboard, students can "logout," bringing the process to a "stop."If the student chooses "Get Attendance" from the homepage, the system performs an "is student Validate" check. If the student is not validated ("No"), they are directed back to "Get Attendance." If the student is validated ("Yes"), their attendance information is then "Store in a database."

17

3.2.3 Database Schema Design

The figure below is database schema design for Student attendance management system. Database schema design is used to show basics structure of the system.
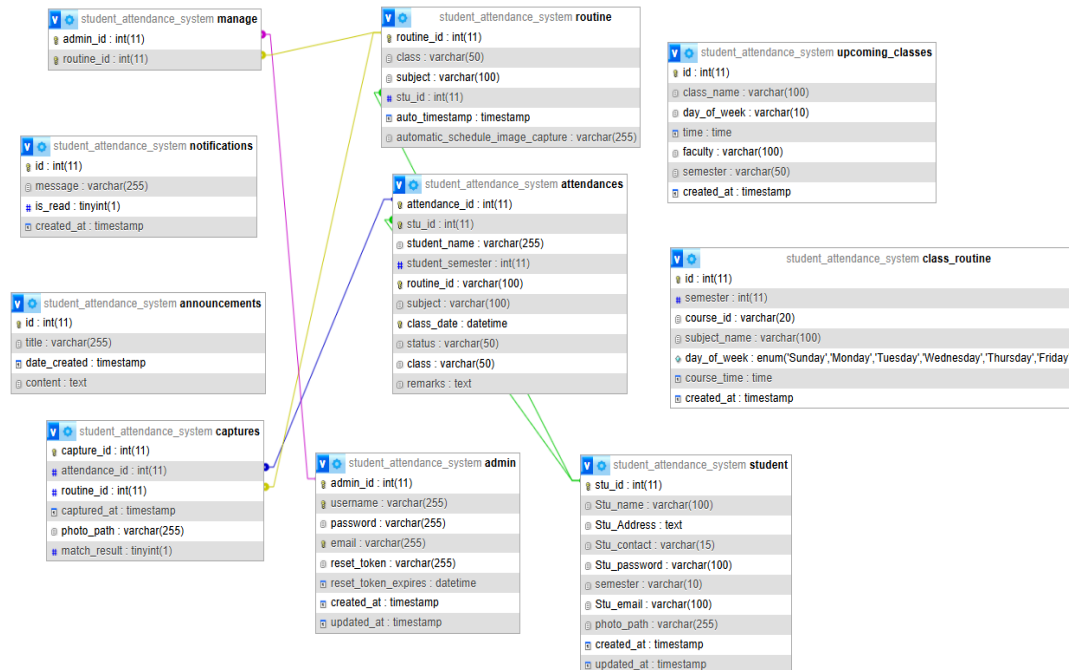

Figure 3.8 Database Schema Design For SAMS

3.2.4 Wireframing

Wireframing is the process of developing a visual blueprint or guide for a software interface, app, or website. It helps in organizing the structure, functionality, and layout earlier than the actual design and development process. In the Student Attendance Management System, wireframing helps in planning the layout and structure of the main pages such as the attendance marking page, student details page, and admin dashboard. It visually outlines the placement of key elements like student lists, attendance buttons, search bars, filters (e.g., date, class, subject), and attendance history sections. This helps ensure a smooth user experience, clear navigation, and well-organized functionality before the design and development begin.
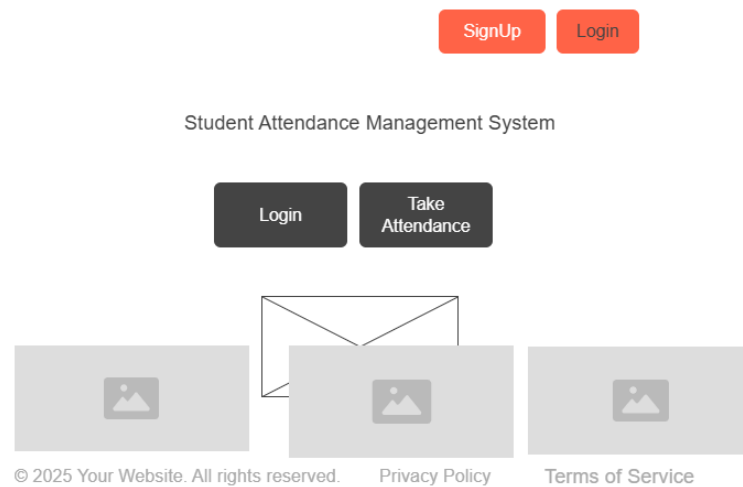
18

Home page:

Figure 3.9 wireframing design of Homepage

This is the homepage wireframe of the SAMS featuring options for Signup, Login, and Take Attendance. It includes a logo section, central navigation buttons, and a footer with policy links.

User login page:

This is a wireframe of login page where user can enter username and password to login the system. If user is not registered there is a signup option for it.
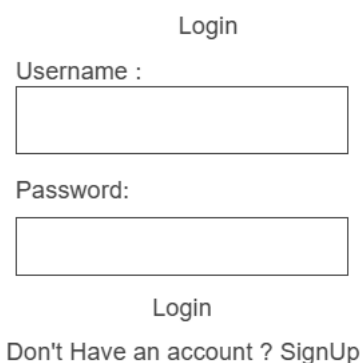

Figure 3.10 Wireframing design of Login page

Admin Dashboard page:

In this wireframing of admin dashboard admin can manage student, manage announcement, manage attendance reports and manage routine.

Figure 3.11 Wireframing of Admin page

Student Dashboard page:

In the wireframing of student dashboard student can view attendance history, view announcement made by admin, view classes and access to their user profile.
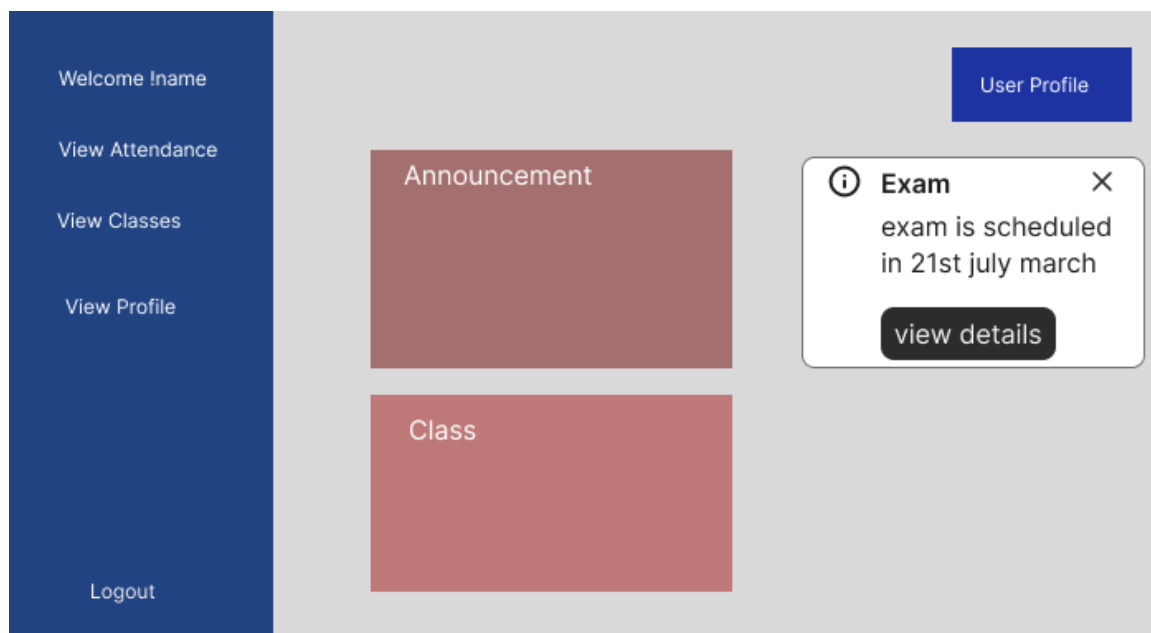

Figure 3.12 Wireframing of User page

3.2.5 Interface Design (UI Interface/Interface Structure Diagrams)

The application's user interface was created with Figma. It is a free online user interface tool that can be used for creation, teamwork, prototyping, and handoff.

Homepage:

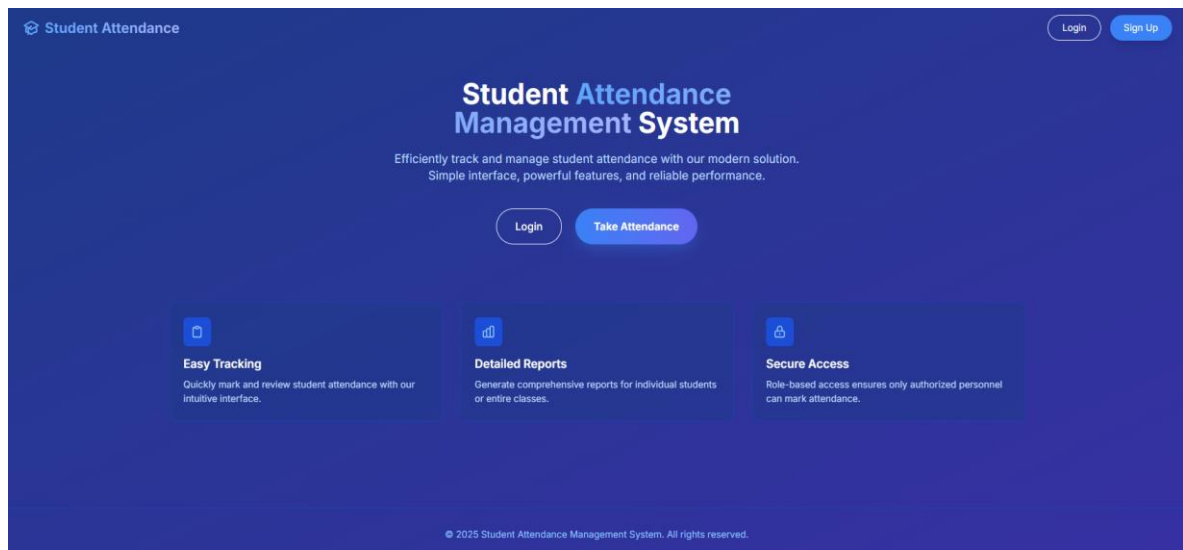This is a interface design of home page. Here admin and user can login. user can take attendance.

Figure 3.13 Interface design of home page

User login page:

In this login page interface design user and admin can login to access dashboard. User can also register to signup page if he is a new user in the system.
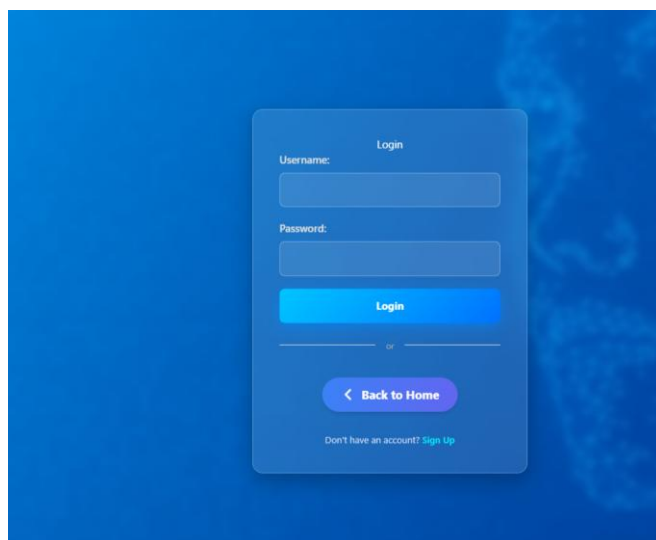

Figure 3.14 Interface design of Login page

User Dashboard page:

This is the interface design of user dashboard where they can access to view attendance, classes, routine etc.
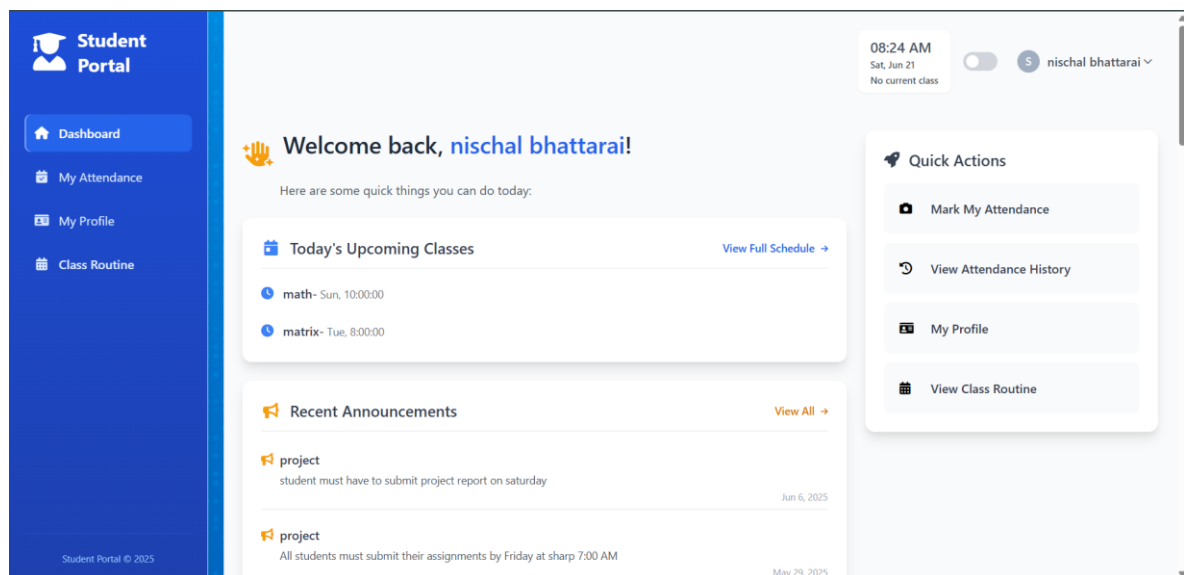

Figure 3.15 Interface design of User dashboard

Admin Dashboard:

This is the interface design of admin dashboard where admin can access to assign classes to student, manage student, make announcement, make attendance report etc.
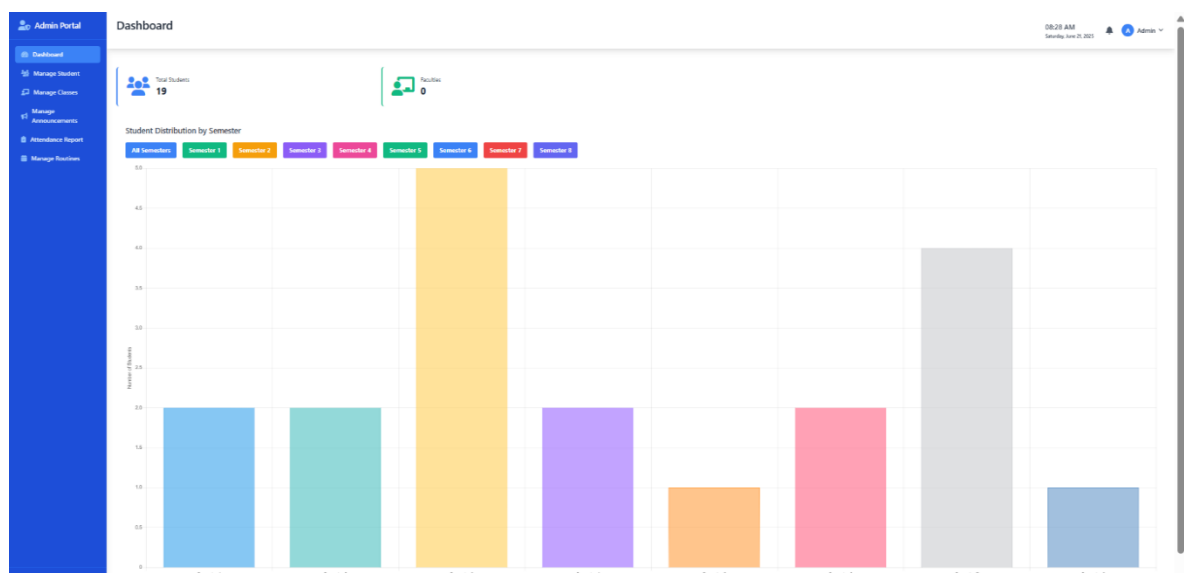

Figure 3.16 Interface design of Admin Dashboard

3.3 Building a Student Attendance Mechanism

The Student Attendance Management System in this project uses a face detection-based mechanism that leverages either OpenCV's Haar Cascade Classifier and HOG (Histogram of Oriented Gradients) with Dlib/face_recognition library to accurately detect student faces and mark attendance. This automated approach reduces manual errors and accelerates the process, ensuring only genuine students are marked present.

Overview of Attendance Mechanism

The system implements the following key steps:

1. **Image/Video Capture:** The system captures a live camera feed or processes an uploaded image for attendance marking.

2. **Preprocessing:** The captured image is converted into grayscale for better accuracy and faster processing.

3. **Face Detection (HOG, Haar Cascade):** The Haar Cascade Classifier is applied to detect faces from the preprocessed image.

4. **Face Recognition (Optional):** If implemented, the detected face is compared against stored student images in the database for identity verification.

5. **Attendance Marking:** Once the student's face is validated, the system updates their attendance record in the **MySQL database**, marking them as present for the respective class/date.

Implementation of Algorithms

1. **Face Detection Method (face_detection.py)**

   This method detects faces from the captured image/video using OpenCV's Haar Cascade Classifier, face_encodings.

   - Convert the image into grayscale.
   - Apply the detectMultiScale() method to identify face regions.
   - Extract bounding boxes around detected faces.

2. **Attendance Marking Method (mark_attendance.py)**

   After detecting or recognizing a student:

   - Retrieve the student ID associated with the detected face.
   - Mark the student as Present in the database.
   - If the student is already marked, skip duplicate entries.

Algorithm flow:

Capture Image/Video → Preprocess (Grayscale) → Face Detection

→ (Optional) Face Recognition → Mark Attendance in Database
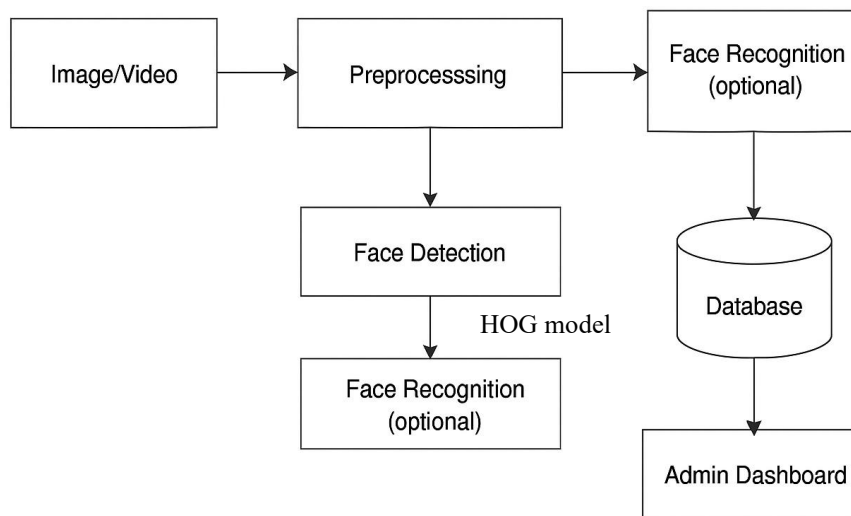
System Architecture Design

Figure 3.17 System architecture design for SAMS

The process of working they are as follows:

1.Live camera feed or uploaded image is input into the system.

2.Face Detection is carried out using HOG and Haar cascade model.

3.Detected faces are matched with stored student data (if recognition is enabled).

4.Attendance is updated in the MySQL database.

5.The admin dashboard can then display attendance history and reports.

Fallback Handling:

If no face is detected or recognition fails:

• The system prompts the user for another image capture.

• Manual attendance marking (admin feature) can be performed as a backup.

Chapter-4: Implementation and Testing

4.1 Implementation

To develop the software some standard tools that are very essential and common are used which are mentioned as below.

4.1.1 Tools Used

Front End Tool

Front end of this application is designed using HTML, CSS and JavaScript(S). The application is built utilizing the most popular combination of web technology to develop the Student Attendance Management System.

Back End and Database Tool

Back end of this application is built with Phyton and MySQL is used for managing database. Our application is hosted in Apache server with the help of XAMPP software.

XAMPP is an abbreviation for cross-platform, Apache, MySQL, Phyton, on a local web server on your computer. This simple and lightweight solution works on Windows, Linux and Mac as well.

Documentation Tools

1. **MS Office**

This is used for writing and editing the documentation of System Attendance Management System.

2. **Draw.io**

This is used to generate diagrams for system analysis and design of. Diagrams were created using this tool in order to save time System Attendance Management System.

4.1.2 Implementations Details of Module

The system's modules are shown below:

- **User Registration Module**

```python
import sqlite3
db = sqlite3.connect('attendance.db'); c = db.cursor()
c.execute('CREATE TABLE IF NOT EXISTS users(name TEXT, pass TEXT)')
def register(name, pw): c.execute('INSERT INTO users VALUES(?,?)', (name, pw)); db.commit()
```

Listing 4.1: Code snippet to user registration

- **User Login**

```python
def login(name, pw):
    user = c.execute('SELECT * FROM users WHERE name=? AND pass=?', (name, pw)).fetchone()
    return True if user else False
```

Listing 4.2 Code snippet to User login

- **Capture Image**

```python
import cv2
def capture_img(file_name):
    cam = cv2.VideoCapture(0); ret, frame = cam.read()
    if ret: cv2.imwrite(file_name, frame)
    cam.release()
```

Listing 4.3 Code snippet to Capture Image

- **Redirect Dashboard**

```python
def dashboard():
    users = c.execute('SELECT name FROM users').fetchall()
    for u in users: print(f"User: {u[0]} | Attendance: Present/Absent")
```

Listing 4.4 Code snippet to Redirect Dashboard

- **Attendance Algorithm**

```python
def detect_faces(image):
    face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
    gray = cv2.cvtColor(cv2.imread(image), cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray, 1.3, 5)
    print(f"{len(faces)} face(s) detected")
```

Listing 4.5 Code snippet for attendance algorithm

- **Mark Attendance**

```python
def mark_attendance(name):
    c.execute('CREATE TABLE IF NOT EXISTS attendance(name TEXT, status TEXT)')
    c.execute('INSERT INTO attendance VALUES(?, ?)', (name, "Present")); db.commit()
```

Listing 4.6 Code snippet for Mark Attendance

- **View Attendance**

```python
def view_attendance():
    rows = c.execute('SELECT * FROM attendance').fetchall()
    for r in rows: print(f"{r[0]} - {r[1]}")
```

Listing 4.7 Code snippet for View Attendance

4.2 Testing

Software testing is an investigation conducted to provide stakeholders with the information about the quality of the product or services under test. It is the process of executing software with the intent of finding bugs and to ensure that it satisfies the specified requirements After the development of the system, I have tested whether it meets the requirement or not in different phases. The functionality has also been tested taking some test cases.

4.2.1 Unit Testing

Unit testing in the Student Attendance Management System is a vital step to ensure the correctness and reliability of individual components. This process involves designing and executing test cases for specific modules such as user authentication, facial recognition, attendance marking, and dashboard functionalities. By conducting unit tests, I validate that each module operates as intended in isolation. Comprehensive unit testing helps identify and resolve bugs at an early stage, ensuring a stable foundation for the entire system.

4.2.2 System Testing

System testing plays a key role in verifying the proper integration and overall functionality of all modules within the Student Attendance Management System. This phase involves testing the system as a whole to ensure seamless interaction between components like the admin panel, student dashboard, attendance logging, and database operations. The objective

is to confirm that the entire system works cohesively to deliver accurate and efficient performance under real-world conditions.

4.2.3 Test Cases

Test cases for system testing cover a broad range of functionalities essential to the attendance system. Examples include verifying login and registration processes to ensure secure user access, checking the functionality of facial recognition to validate attendance marking, and ensuring that attendance data is accurately stored and displayed in the dashboard. System testing was performed to guarantee a reliable, user-friendly experience for both students and administrators, enabling efficient attendance tracking and management.

The software was tested under various scenarios to validate its robustness. Some of the key test cases designed to assess the correctness and reliability of the system are listed and discussed in detail in the following test table.

Student Registration

Table 2 Test Cases for Student Registration for SAMS

S.N	Test Name	Input	Expected outcome	Actual Outcome	Test Result
1.	Open Application	http://127.0.0.1:5000/signUp	At form page	At form page	Pass
2.	Enter Invalid Username, email, address, choose user, password, confirm password and click register button	Username = Nischal Bhattarai E-mail= nischalbhattarai@gmail.com Password = adapter@123 Confirm password =	Enter Confirm password	Registration Failed	Pass

| 3. | Enter Valid Username, email, address, choose user, password, confirm password and click register button | Username = Nischal Bhattarai E-mail= nischalbhattarai@gmail.com Password = adapter@123 Confirm password =adapter@123 | Registration successful and redirect to home page | Registration successful and redirect to home page | Pass |

Student Login

<p align="center">Table 3 Test case of Student login for SAMS</p>

S. N	Test Name	Input	Expected Output	Actual Outcome	Test Result
1.	Open Application	http://127.0.0.1:5000	At Form Page	At Form page	Pass
2.	Enter Username and Invalid password and click login button	UserName: Nischal Bhattarai Password: 123	Login Failed And refresh the fields in same page	Login failed	Pass
3.	Enter Valid Username and Password and click login button	UserName: Nischal Bhattarai Password: adapter@123	Login Successful and redirect to home page	Redirect to home page	Pass

Admin Login

<p align="center">Table 4 Test case of Admin Login for SAMS</p>

S. N	Test Name	Input	Expected Output	Actual Outcome	Test Result
1.	Open Application	http://127.0.0.1:5000	At Form Page	At Form page	Pass
2.	Enter Username and Invalid password and click login button	UserName:admin Password: 123	Login Failed And refresh the fields in same page	Login failed	Pass
3.	Enter Valid Username and Password and click login button	UserName: admin Password: admin@123	Login Successful and redirect to home page	Redirect to admin dashboard	Pass

4.3 Result an analysis

The Student Attendance Management System has been successfully designed and implemented as a complete platform for automating attendance and improving academic record management. The system uses an image capture and face detection mechanism (OpenCV's Haar Cascade Classifier) to identify students and mark attendance automatically, reducing manual effort and errors.

The platform is role-based with two user types: Admin and Student.

- Admin: Can manage students, manage classes, manage routines, post announcements, view attendance reports, and oversee the entire system.
- Student: Can mark attendance via face detection, view assigned classes and announcements, and check attendance history.

The system is built with a front-end in HTML, CSS, and JavaScript and a back-end in Python (Flask/Django) with MySQL for secure data storage and smooth operations. This architecture

ensures efficient data handling, modularity, and scalability for multiple classes and large student data sets.

From an analytical perspective, the face detection-based attendance module is the most impactful feature, allowing students to be identified through captured images and marked present without manual roll calls. This proxy attendance mechanism eliminates delays and improves accuracy. The admin dashboard allows teachers or administrators to monitor attendance records in real-time and generate attendance reports, while students benefit from easy access to class schedules, announcements, and attendance history.

The system has demonstrated strong performance, handling multiple students and classes simultaneously without lag. The role-based access control further enhances security and accountability by ensuring that admins and students only access features relevant to their roles.

Areas for improvement:

- Upgrade the face detection module to advanced face recognition algorithms (deep learning) for improved accuracy and to prevent impersonation.
- Add attendance analytics such as monthly summaries, low attendance alerts, and graphical dashboards.
- Integrate SMS/Email notifications for students and parents when attendance falls below a threshold.
- Build a mobile application for improved accessibility and push notifications.

In summary, the Student Attendance Management System effectively meets its objectives by automating attendance through face detection and streamlining academic processes for both admins and students. With its modular design, secure database structure, and user-friendly interface, it offers a reliable solution for institutions. Future updates like mobile app support, AI-powered recognition, and enhanced analytics will further strengthen its capability as a complete attendance management platform.

Chapter 5: Conclusion and Recommendation System

5.1 Outcome and Lesson Learnt

Every project helps in learning and enhancing the knowledge in the respective field. This project also helped in learning different aspects. In the project, ihave learned lots of problem-solving skills and learn things like finding the solution on our own, proper use of guidelines, communication and writing skills and management of team.

- **Problem Solving Skills**

  From this project, I have learned lots of problem-solving skills and also learned to recognize different errors occur in this system and solve it.

- **Writing Skills**

  We've learned how to create project proposals and documentation, and we're skilled in using various case tools for different types of diagrams, such as use case diagrams, schema diagrams, data flow diagrams, and ER diagrams.

- **User Interface Design Matters**

  Wireframing and creating a user-friendly interface showed us that a clean layout and clear navigation significantly improve usability for both admins and students.

- **Practical Use of Face Detection**

  Implementing OpenCV's Haar Cascade for image-based attendance helped us understand real-time image processing, the importance of preprocessing (e.g., grayscale conversion), and challenges like lighting and camera angles.

- **Scalability and Future Enhancements**

  I learnt the importance of modular coding and flexible design to accommodate features like analytics, notifications, and mobile app integration in the future.

5.2 Conclusion

In conclusion, the Student Attendance Management System has been thoroughly tested through both unit and system testing approaches to ensure its accuracy, reliability, and overall performance. Unit testing helped verify the correct functionality of individual modules such as user authentication, facial recognition, and attendance logging. System testing ensured that all components integrate seamlessly and work together as expected in a real-world environment.

The test cases designed and executed covered various scenarios, confirming that the system performs well under different conditions and handles errors gracefully. As a result, the system delivers efficient solution for managing student attendance, reducing manual effort, minimizing errors, and enhancing overall administrative efficiency. The successful testing outcomes validate the readiness of the system for deployment in educational institutions.

5.3 Future Recommendations

The system can be developed more advanced by adding some more features. Some of the future recommendations for this System are:

- Mobile application support
- Geolocation validation on attendance
- Advanced face recognition
- Automated Announcements & Class Updates

References

[1] "ResearchGate,"[Online].Available:
https://www.researchgate.net/publication/349327088_Implementing_Student_Attend
ance_System_Using_Fingerprint_Biometrics_for_Kolej_Universiti_Poly-
Tech_Mara.

[2] G. Kumar, "An automated attendance system using facial detection and recognition technology," *Apex Journal of Business and Management,* vol. 1, no. 1, pp. 103--120, 2023.

[3] Rabu, The design and implementation of student attendance tracking system using QR code card, National Defence University Publishing House, 2019.

[4] R. a. Dixit, "Comparative analysis of human face recognition by traditional methods and deep learning in real-time environment," in *2020 IEEE 9th international conference on communication systems and network technologies*, IEEE, 2020, pp. 66--71.

[5] U. A. a. Priya, "Development of a student attendance management system using RFID and face recognition," *International Journal of Advance Research in Computer Science and Management Studies,* vol. 2, pp. 109--119, 2014.

[6] C. a. Chanchal, "Class attendance management system using facial recognition," in *ITM Web of Conferences*, EDP Sciences, 2020, p. 2001.

[7] A. a. Sharma, "Attendance Management System Based on Face Recognition Using Haar-Cascade," in *2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering*, IEEE, 2022, pp. 1792--1976.

[8] R. a. Tisha, "A real-time face recognition smart attendance system with haar cascade classifiers," in *2021 third international conference on inventive research in computing applications*, IEEE, 2021, pp. 1417--1425.

[9] W. Adebayo, "Class attendance management system using face recognition," in *2018 7th International conference on computer and communication engineering* , IEEE, 2018, pp. 93--98.

[10] S. a. Mani, "Smart attendance system using OPENCV based on facial recognition," *Int. J. Eng. Res. Technol,* vol. 9, no. 3, pp. 54--59, 2020.

[11] Hasan, "Face detection and recognition using opencv," *Journal of Soft Computing and Data Mining,* vol. 2, no. 2, pp. 86--97, 2021.

[12] A. a. Samal, "Automated Attendance System Based on Face Recognition Using Opencv," in *2023 9th International Conference on Advanced Computing and Communication Systems*, IEEE, 2023, pp. 2256--2259.

[13] B. a. Rawat, "Attendance management system using Python (Haar cascade and Open-CV)," in *International Conference on Green Energy, Computing and Intelligent Technology*, IET, 2023, pp. 438--444.

[14] "GeeksforGeeks,"[Online].Available:https://www.geeksforgeeks.org/what-is-agile-methodology/.

[15] GeeksforGeeks,"GeeksforGeeks,"2023.[Online].Available: https://www.geeksforgeeks.org/python-haar-cascades-for-object-detection/.

Appendices

Frontend Overview

- **Login Page**

- **Registration Page**

- **Student Dashboard**

- **My Profile**

- **Admin Dashboard**

- **Manage Student**

- **Manage Routine Semester wise**

- **Notifications**

Sample Code

Login Code:

```python
from flask import Flask, request, jsonify

app = Flask(__name__)

@app.route('/login', methods=['POST'])
def login():
    data = request.json
    username = data.get('username')
    password = data.get('password')
    if username == 'admin' and password == 'secret':
        return jsonify({'message': 'Login successful'}), 200
    return jsonify({'message': 'Invalid credentials'}), 401
```

Add Student:

```python
from flask import Flask, request, jsonify

app = Flask(__name__)
students = []

@app.route('/add_student', methods=['POST'])
def add_student():
    data = request.json
    student = {
        'id': data.get('id'),
        'name': data.get('name'),
        'email': data.get('email')
    }
    students.append(student)
    return jsonify({'message': 'Student added', 'student': student}), 201
```

Mark Attendance:

```python
from flask import Flask, request, jsonify

app = Flask(__name__)
attendance_records = []

@app.route('/mark_attendance', methods=['POST'])
def mark_attendance():
    data = request.json
    record = {
        'student_id': data.get('student_id'),
        'date': data.get('date'),
        'status': data.get('status')  # Present/Absent
    }
    attendance_records.append(record)
    return jsonify({'message': 'Attendance marked', 'record': record}), 201
```

Get Student Attendance:

```python
from flask import Flask, request, jsonify

app = Flask(__name__)
attendance_records = [
    # Example: {'student_id': 1, 'date': '2025-06-21', 'status': 'Present'}
]

@app.route('/get_attendance/<int:student_id>', methods=['GET'])
def get_attendance(student_id):
    records = [r for r in attendance_records if r['student_id'] == student_id]
    return jsonify({'student_id': student_id, 'attendance': records}), 200
```

List of All Student:

```python
from flask import Flask, jsonify


app = Flask(__name__)

students = [

# Example: {'id': 1, 'name': 'Nischal', 'email': 'nischal@gmail.com'}

]


@app.route('/students', methods=['GET'])

def list_students():

return jsonify({'students': students}), 200
```