

# STUDENT PRESENCE MONITORING SYSTEM

A

## PROJECT REPORT

*Submitted by*

**Dhvi Shah(235490694116)**

**Sahil Vaghela(235490694137)**

**Jay Makwana(235490694051)**

*In partial fulfillment for the award of the degree of*

**MASTER OF COMPUTER APPLICATIONS**

*In*

Department of Computer Applications

**R.B. INSTITUTE OF MANAGEMENT & STUDIES**

**Opp. Hirawadi BRTS Thakkarbapa Nagar ,  
Hirawadi, Ahmedabad, Gujarat 382350**



**Gujarat Technological University, Ahmedabad**

**[April,2025]**

# CERTIFICATE

This is to certify that the project report submitted along with the project entitled **Student Presence Monitoring System** has been carried out by **Dhviy Shah(235490694116)** , **Sahil Vaghela (235490694137)** , **Jay Makwana (235490694051)** under my guidance in partial fulfilment for the degree of Master of Computer Application, 4<sup>th</sup> Semester of Gujarat Technological University, Ahmadabad during the academic year 2024-25.

Prof. **Pooja Bhadauriya**

**Internal Guide**

Prof. Hiral Rathod

Head of the Department



## **R.B. INSTITUTE OF MANAGEMENT & STUDIES**

**Opp. Hirawadi BRTS Thakkarbapa Nagar ,  
Hirawadi, Ahmedabad, Gujarat 382350**

### **DECLARATION**

We hereby declare that the Internship report submitted along with the Internship entitled **Student Presence Monitoring System** submitted in partial fulfilment for the degree of Master of Computer Application to Gujarat Technological University, Ahmedabad, is a Bonafede record of original project work carried out by me Student Presence Monitoring System under the supervision of **Prof. Pooja Bhadauriya** and that no part of this report has been directly copied from any students' reports or taken from any other source, without providing due reference.

Name of the students	Sign of Student
Dhvi Shah	
Sahil Vaghela	
Jay Makwana	

Date: 24/04/2025

## **TO WHOM IT MAY CONCERN**

This is to certify that **Shah Dhvij Amitkumar**, a student of **R. B. Institute of Management Studies** has successfully completed his internship in the field of **Python** from 3<sup>rd</sup> January 2025 to 24<sup>th</sup> April 2025. He has worked on project entitled **Face Reorganization Attendance System**. During the period of his internship program with us, he had been exposed to different processes and was found diligent, hardworking and inquisitive.

We wish him every success in his life and career.

For **TuskerAI Pvt. Ltd.**





Date: 24/04/2025

## **TO WHOM IT MAY CONCERN**

This is to certify that **Vaghela Sahil Sandipbhai**, a student of **R. B. Institute of Management Studies** has successfully completed his internship in the field of **Python** from 3<sup>rd</sup> January 2025 to 24<sup>th</sup> April 2025. He has worked on project entitled **Face Reorganization Attendance System**. During the period of his internship program with us, he had been exposed to different processes and was found diligent, hardworking and inquisitive.

We wish him every success in his life and career.

For TuskerAI Pvt. Ltd.





## **Acknowledgement**

I am pleased to take this opportunity to express my deep gratitude to all those who contributed to the successful completion of my project titled “**Student Presence Monitoring System**”.

First and foremost, I would like to extend my sincere thanks to my respected guide, **Prof. Pooja Bhadauriya**, for their invaluable guidance, consistent support, and encouragement throughout the course of this project. Their insights and suggestions were instrumental in shaping the direction and outcome of this work.

I would also like to thank the faculty members of the **Department of Computer Applications, R B Institute Of Management Studies**, for their academic support and for providing the knowledge foundation required for this project.

I am grateful to my peers and friends for their motivation and assistance, and for creating a collaborative environment that helped me stay focused and inspired.

Lastly, I would like to express my heartfelt appreciation to my family for their unwavering support, patience, and confidence in me during every phase of this project.

This acknowledgement would be incomplete without recognizing all those who, directly or indirectly, helped me in bringing this project to its final form.

## **Abstract**

A Student Presence Monitor System project aims to develop a software application that digitally tracks and manages student attendance within an educational institution automates processes. Such as Student enrollment , Attendance Tracking Allowing Administrators to record monitor and analyze student presence in classes.

<b>Sr. No.</b>	<b>Table of Contents</b>	<b>Page No.</b>
	Acknowledgement.....	i
	Abstract.....	ii
	Table of Content.....	iii
<b>1</b>	<b>Introduction.....</b>	<b>1</b>
	1.1 Existing System.....	2
	1.2 Need for the New System.....	3
	1.3 Objective of the New System.....	4
	1.4 Problem Definition.....	5
	1.5 Core Components.....	6
	1.6 Project Profile.....	7
	1.7 Assumptions and Constraints.....	8
	1.8 Advantages and Limitations.....	10
<b>2</b>	<b>Requirement Determination Analysis.....</b>	<b>11</b>
	2.1 Requirement Determination.....	12
	2.2 Targeted Users.....	12
<b>3</b>	<b>System Design.....</b>	<b>13</b>
	3.1 Use Case Diagram.....	14
	3.2 Class Diagram.....	15
	3.3 Interaction Diagram.....	16
	3.4 Activity Diagram.....	18
	3.5 Data Dictionary.....	20
<b>4</b>	<b>Development.....</b>	<b>22</b>
	4.1 Coding Standards.....	23
	4.2 Screenshots.....	26
<b>5</b>	<b>Agile Documentation.....</b>	<b>46</b>
	5.1 Agile Project Charter.....	47



5.2 Agile Roadmap / Schedule.....	48
5.3 Agile Project Plan.....	49
5.4 Agile User Story (Minimum 3 Tasks).....	50
5.5 Agile Release Plan.....	51
5.6 Agile Sprint Backlog.....	52
5.7 Agile Test Plan.....	53
5.8 Earned-value and Burn Charts.....	54
<b>6 Proposed Enhancement.....</b>	<b>55</b>
<b>7 Conclusion.....</b>	<b>56</b>
<b>8 Bibliography.....</b>	<b>57</b>

# **1. Introduction**

## **1.1 Existing System**

- In the past, educational institutions primarily relied on manual attendance systems to track student presence.
- The most common method was the manual attendance register , where teachers marked attendance in a physical logbook.
- Some institutions later transitioned to Excel-based attendance, where records were maintained in spreadsheets

## **1.2 Need for the New System**

- Manual systems, while simple, lacked accuracy and were difficult to manage at scale.
- Excel-based records still required manual input, which introduced chances of human error and duplication.
- Paper-based sign-in sheets could be easily manipulated, lost, or damaged.
- Analyzing trends or generating reports was tedious and required extra effort.

## **1.3 Objective of the New System**

- It aims to ensure accurate and real-time tracking of student attendance using face recognition technology.
- The system is designed to reduce manual work for teachers and administrative staff.
- It provides easy data storage, quick retrieval, and report generation capabilities.
- Ultimately, the goal is to create a secure, efficient, and user-friendly system that improves attendance management in educational institutions.



## **1.4 Problem Definition**

- Manual registers and sign-in sheets lack accuracy, are difficult to manage, and are susceptible to manipulation or loss.
- Even Excel-based tracking requires significant manual effort, making data analysis and report generation tedious.
- These challenges often result in poor attendance management and limited insights into student presence.
- A Student Presence Monitor System project aims to develop a software application that digitally tracks and manages student attendance within an educational institution.
- The system automates processes such as student enrollment and attendance tracking.
- It allows administrators to record, monitor, and analyze student presence in classes.

## **1.5 Core Components**

### **Admin**

1. Login : Admins access a secure login portal to enter the admin dashboard. This ensures that only authorized personnel can manage system functionalities like student profiles, attendance, and scheduling.
2. Student Management: Admins can add, update, view, or remove student records. This includes managing student details, assigned courses, and associated profile images for face recognition purposes.
3. Attendance Monitoring: Admins can oversee and manage real-time attendance records collected through face recognition. They can view daily, weekly, or course-wise attendance and export reports if needed.
4. Course Management : Admins can create and manage course information including course names, codes, instructors, and enrolled students. Courses are directly linked to the attendance tracking module.
5. Timetable Scheduling: Admins can generate and update class schedules. The timetable integrates with attendance tracking to ensure sessions are monitored accurately.

### **Student**

1. Automatic Attendance Capture : Students' attendance is captured automatically using face recognition technology when they are present during scheduled class times. This ensures a contactless, error-free, and real-time attendance recording system.
2. View Timetable: Students can access their personalized class schedules. The timetable provides details such as course name, date, time, and location, helping students stay organized and punctual.

## **1.6 Project Profile**

<b>PROJECT NAME</b>	Student Presence Monitoring System
<b>FRONT-END</b>	HTML , CSS , JavaScript
<b>FRAMEWORK</b>	Flask
<b>BACK-END</b>	Python
<b>DATABASE</b>	SQLYOG

## **1.7 Assumptions and Constraints**

### **Assumptions**

- All students are registered in the system with accurate personal details, including valid student IDs and facial images for recognition.
- The classroom has the necessary hardware setup, including cameras with good resolution and adequate lighting for effective facial recognition.
- Administrative staff and faculty have the necessary technical skills to use the system effectively, including registering students and managing attendance data.

## **Constraint**

- Facial recognition accuracy may be affected by factors like poor lighting conditions, low-quality cameras, or students' appearance changes (e.g., masks, hairstyles, or accessories).
- If the system depends on cloud-based services, its functionality may be limited or unavailable during network downtime or if used in an offline environment.
- The system's performance may be limited by environmental factors such as lighting, classroom layout, or the number of students present in the class.



## **1.8 Advantage and Limitations of the Proposed System**

### **Advantages**

- Automated Attendance Tracking
- Reduces manual effort and paperwork through facial recognition and digital records.
- Real-Time Reporting
- Generates instant attendance reports, helping in faster decision-making.
- Notifies administrators about frequently absent students for early intervention.
- Minimizes errors and manipulations common in manual attendance system.

### **Limitations**

- Requires good-quality cameras, stable internet (if cloud-based), and capable hardware.
- Recognition Issues
- Accuracy can drop due to poor lighting, face masks, or student appearance changes.
- Initial Setup Cost
- Installation of hardware and integration of software can be expensive initially.
- Privacy Concerns

## **2. Requirement Determination & Analysis**

## **2.1 Requirement Determination**

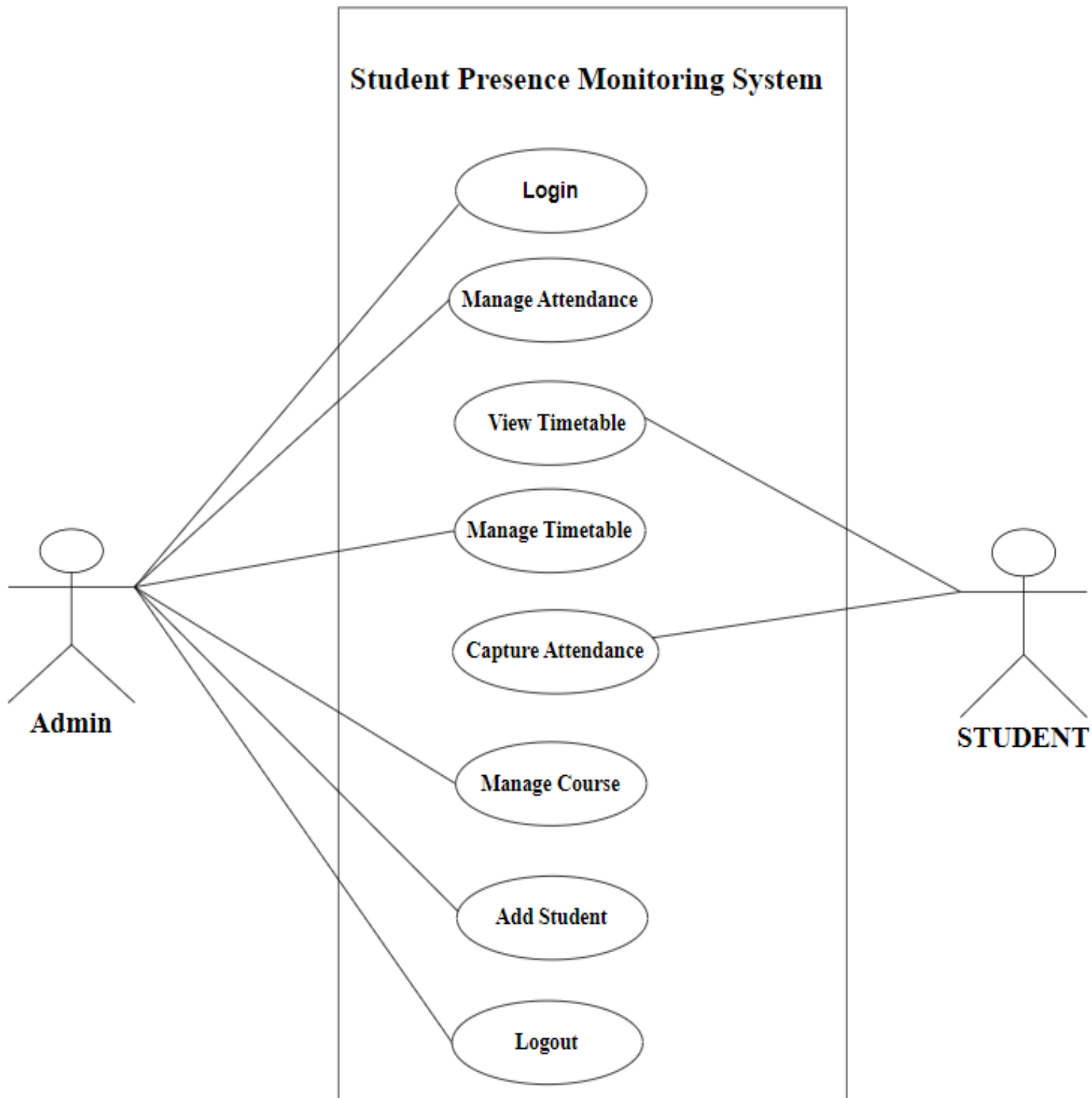
- **Student Enrollment:-** Admin can register students with details like name, student ID, course, and facial image.
- **Facial Recognition Attendance:-** The system detects and marks attendance based on student face recognition.
- **Attendance Records:-** Automatically stores attendance data with date and time for each student.
- **Report Generation:-** Generates daily, weekly, and monthly attendance reports.

## **2.2 Targeted Users**

- **Admin:** Admins are responsible for managing the overall system. They handle student enrollment, monitor real-time attendance records, and analyze attendance data. The system enables admins to maintain accurate records, reduce manual errors, and ensure smooth institutional operations.
- **User:** Can access a range of features including emergency dialing, hospital and medical shop locators, AI chatbot assistance, medical history upload, and NFC data sharing. Users also receive real-time notifications for nearby facilities and updates.

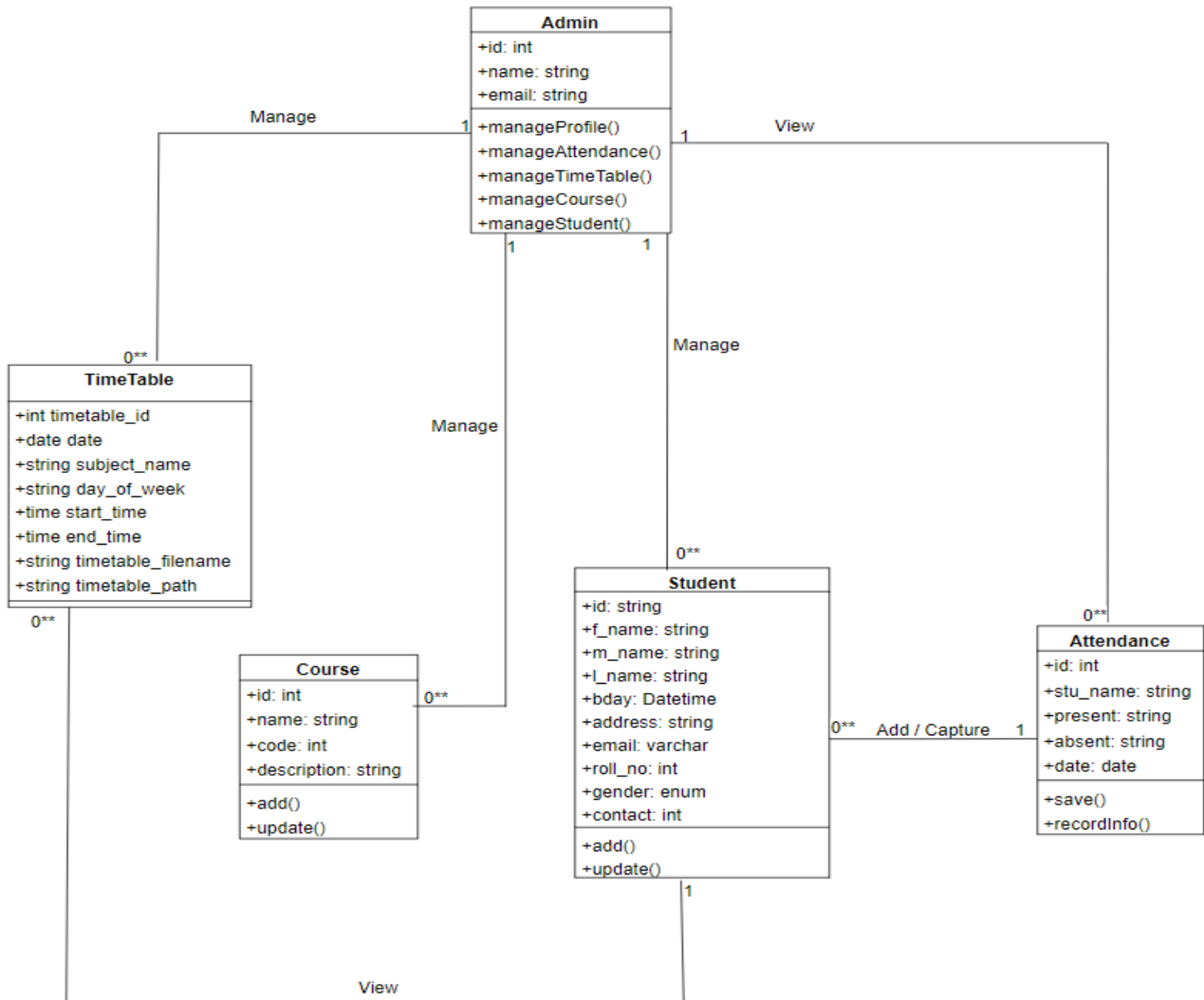
### **3. System Design**

### 3.1 Use Case Diagram



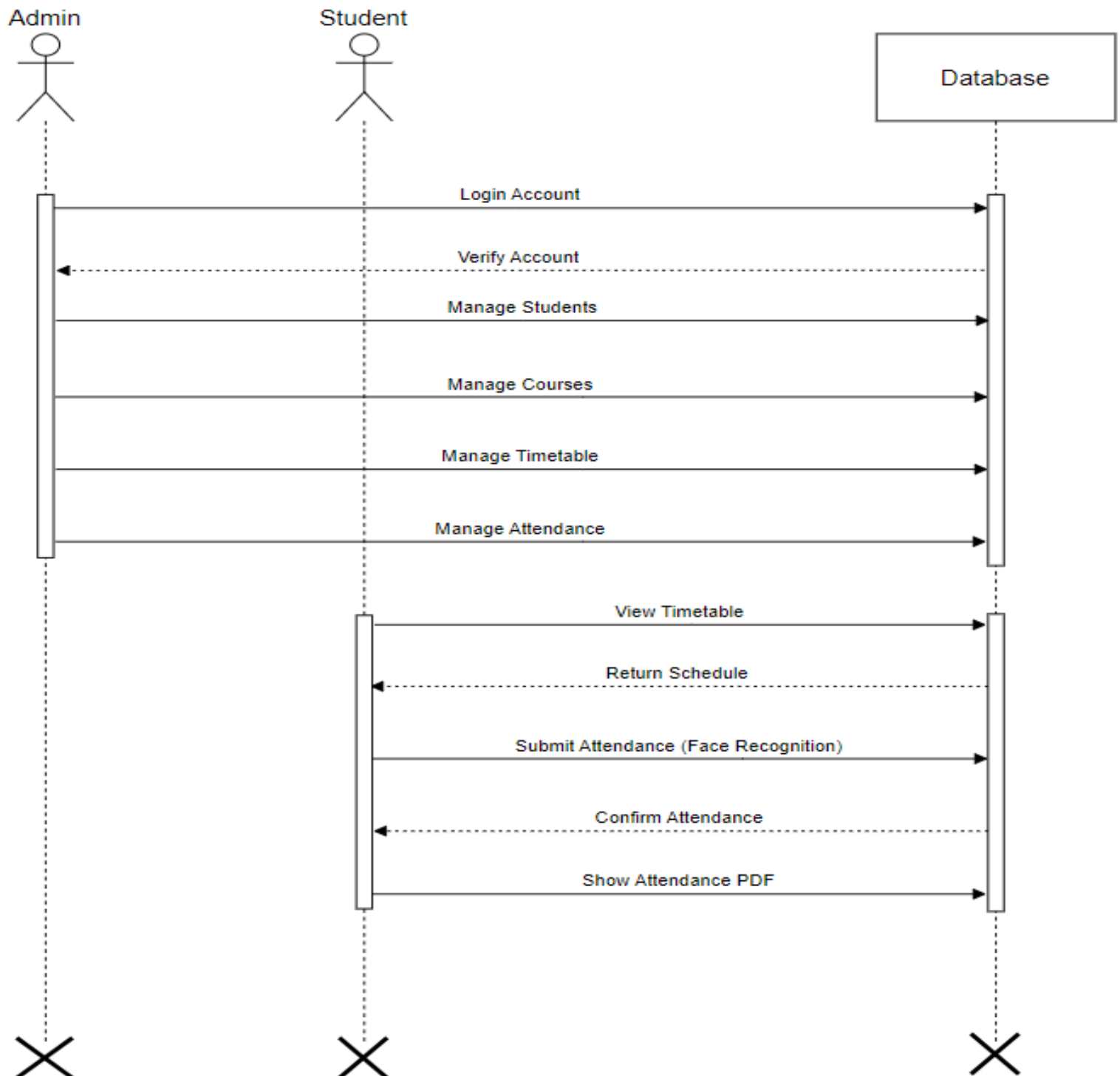


## 3.2 Class Diagram



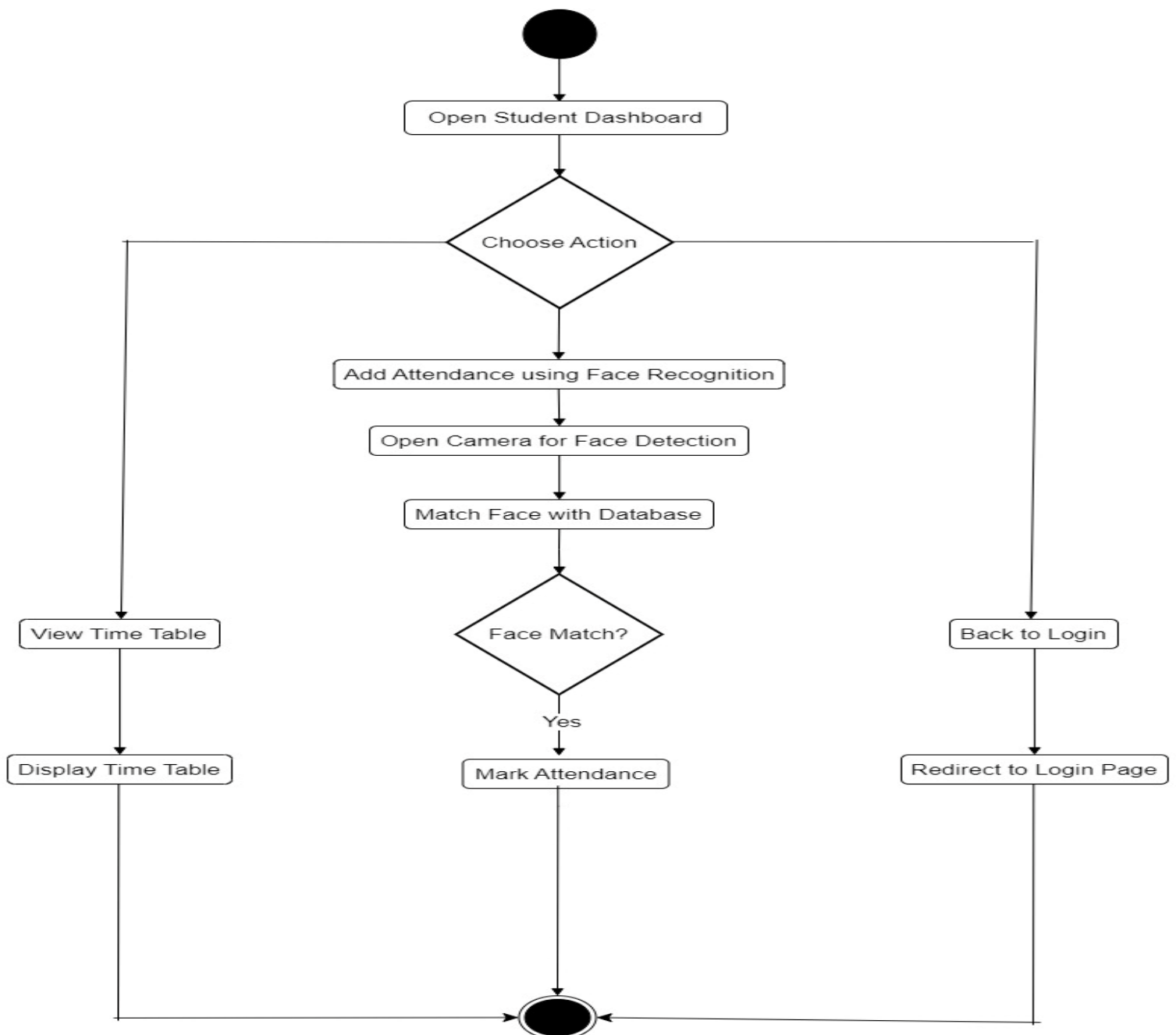
### 3.3 Interaction Diagram

#### Interaction Diagram Admin And User

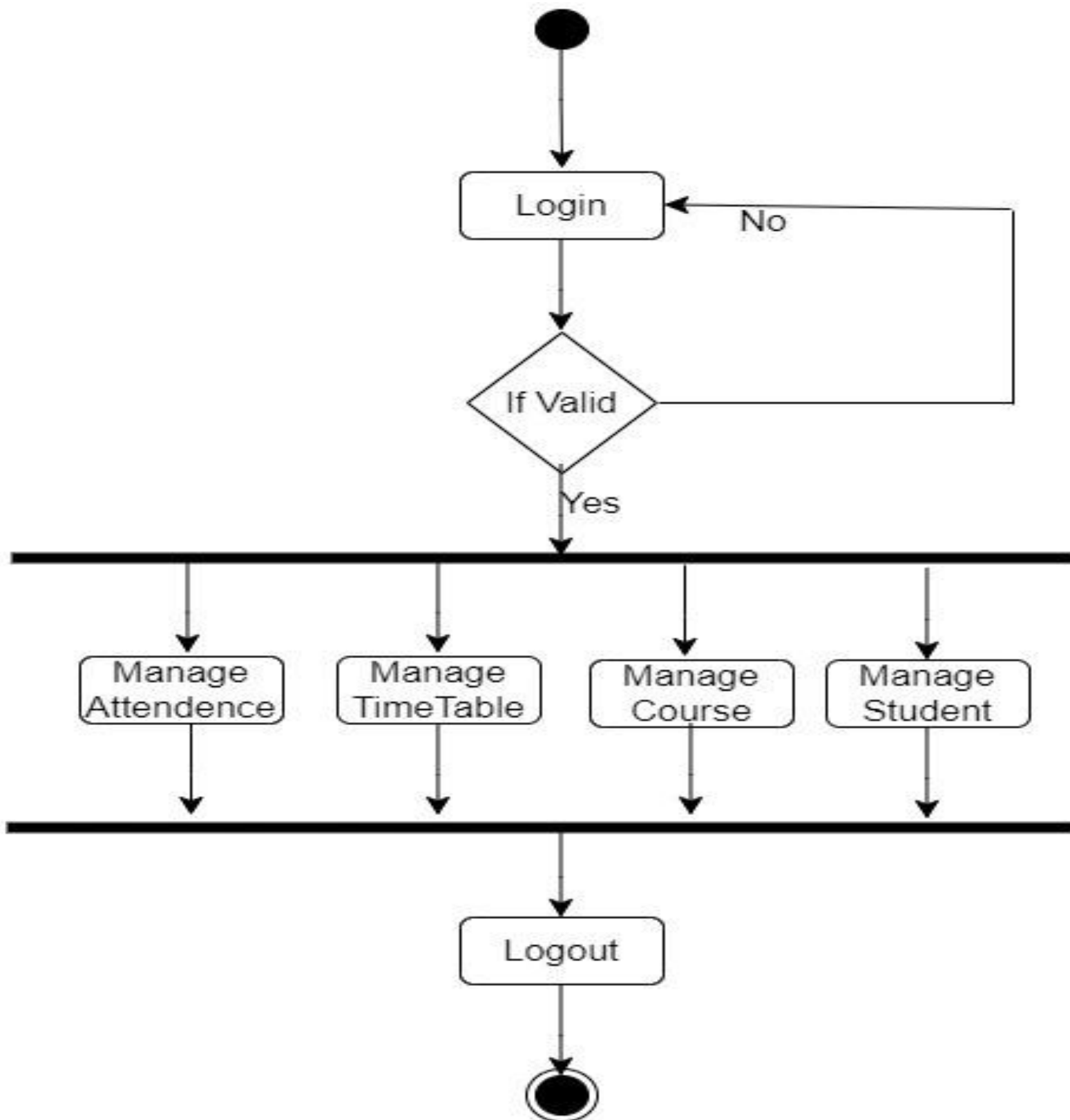


## 3.4 Activity Diagram

### Activity Diagram User



## Activity Diagram Admin



## **3.5 Data Dictionary**

**1. Table Name : Student**

**Primary Key :Student \_Id**

**Description : It Store all Details about Student.**

<b>Column</b>	<b>Data Type</b>	<b>Constraint</b>	<b>Description</b>
<b>Student_Id</b>	Int(11)	Primary Key	It store student id
<b>First_name</b>	varchar(100)	Not Null	It store first name
<b>Last_name</b>	varchar(50)	Not Null	It store last name
<b>Birth_date</b>	Date	Not null	It store birth date
<b>Address</b>	Varchar(100)	Not null	It store address
<b>Email_id</b>	Varchar(25)	Not null	It store email id
<b>Roll_no</b>	Int(10)	Not null	It store roll no
<b>Gender</b>	Enum(10)	Not null	It store gender
<b>Contact</b>	Int(10)	Not null	It store contact
<b>Course_name</b>	Varchar(100)	Foreign key	It store course name
<b><u>Image_file_name</u></b>	varchar(255)	Not Null	Store name of Image file Name
<b><u>Image_path</u></b>	varchar(255)	Not Null	Store Image file path

**2. Table Name : Admin**

**Primary Key :Admin \_Id**

**Description : It Store all Details about Admin**

<b>Column</b>	<b>Data Type</b>	<b>Constraint</b>	<b>Description</b>
<b>Admin_Id</b>	Int(11)	Primary Key	It store admin id
<b>Admin_Name</b>	varchar(100)	Not Null	It store admin name
<b>Email_id</b>	varchar(50)	Not Null	It store email id



### 3. Table Name : Timetable

**Primary Key :Timetable \_Id**

**Description : It Store all Details about Timetable**

Column	Data Type	Constraint	Description
Timetable_id	Int(11)	Primary Key	Unique identifier for each timetable entry
Date	Date	Not Null	It Store date
<u>Subject_name</u>	varchar(255)	Not Null	Store subject name
Day_of_week	Varchar(10)	Not Null	Represents the day of the week (e.g., Monday, Tuesday)
Start_time	Time	Not Null	Time when the scheduled event begins
End_time	Time	Not Null	Time when the scheduled event ends
<u>timetable_filename</u>	varchar(255)	Not Null	Store name of time table file
<u>timetable_path</u>	varchar(255)	Not Null	Store Excel file path

### 4. Table Name : Course

**Primary Key :Course \_Id**

**Description : It Store all Details about Course**

Column	Data Type	Constraint	Description
Course_Id	Int(11)	Primary key	It store course id
Course_name	Varchar(50)	Not Null	It store course name
Course_code	int(10)	Not Null	It store course code
Course_description	text	Not Null	It store course description

**5. Table Name : Attendance**

**Primary Key :Attendance \_Id**

**Description : It Store all Details about Attendance.**

<b>Column</b>	<b>Data Type</b>	<b>Constraint</b>	<b>Description</b>
<b>Attendance_Id</b>	Int	Primary key	It store attendance id
<b>Attendance_Date</b>	Varchar(50)	Not Null	It store attendance date
<b>Attendance_Time</b>	time	Not Null	It store attendance time
<b>Attendance_Status</b>	int	Not Null	It store attendance status
<b>Attendance_Student_id</b>	int	Foreign key	It store attendance student id

## **4. Development**

## 4.1 Coding Standards

To ensure a consistent and maintainable code base, the following coding standards will be adhered to:

- Meaningful variable and function names.
- Proper indentation and formatting of code.
- Comments for complex logic and important sections of code.
- Utilization of object-oriented principles to improve code structure.
- Code :-

```
1  import cv2
2  import face_recognition
3  import os
4  import numpy as np
5  from datetime import datetime
6
7
8  1 usage
9  def load_known_faces_from_folder(folder_path):
10     known_face_encodings = []
11     known_face_names = []
12     for image_name in os.listdir(folder_path):
13         image_path = os.path.join(folder_path, image_name)
14         if image_path.lower().endswith(('.png', '.jpg', '.jpeg')):
15             image = face_recognition.load_image_file(image_path)
16             face_encoding = face_recognition.face_encodings(image)
17             if face_encoding:
18                 known_face_encodings.append(face_encoding[0])
19                 known_face_names.append(image_name)
20     return known_face_encodings, known_face_names
21
```

```

12 | usage
13 | def convert_to_date(date_value):
14 |     if isinstance(date_value, str):
15 |         return pd.to_datetime(date_value).date()
16 |     elif hasattr(date_value, 'date'):
17 |         return date_value.date()
18 |     elif pd.isna(date_value):
19 |         raise ValueError("Date value cannot be empty")
20 |     return date_value
21 |
22 | 3 usages
23 | def parse_excel_to_dict(source):
24 |     df = pd.read_excel(source)
25 |
26 |     df['date'] = df['date'].apply(convert_to_date)
27 |     df['start_time'] = df['start_time'].apply(convert_to_time)
28 |     df['end_time'] = df['end_time'].apply(convert_to_time)
29 |
30 |     data_dict = df.to_dict(orient='records')
31 |     return data_dict
32 |
33 | /admin/home 1 usage
34 | @app.route('/admin/home')
35 | @login_required('admin')
36 | <> def admin_home():
37 |     student_dao = StudentDAO()
38 |     attendance_dao = AttendanceDAO()
39 |     count_student = student_dao.count_student()
40 |     count_attendance = attendance_dao.count_attendance()
41 |     return render_template(template_name_or_list='admin/index.html',
42 |                             count_student=count_student,
43 |                             count_attendance=count_attendance)
44 |
45 | /home
46 | @app.route('/home')
47 | <> def home():
48 |     student_dao = StudentDAO()
49 |     attendance_dao = AttendanceDAO()
50 |     count_student = student_dao.count_student()
51 |     count_attendance = attendance_dao.count_attendance()
52 |     return render_template(template_name_or_list='admin/index.html',
53 |                             count_student=count_student,
54 |                             count_attendance=count_attendance)

```

```

1 > import ...
4
11 usages
5 class StudentDAO:
    1 usage
6     def insert_student(self, student_vo):
7         db.session.add(student_vo)
8         db.session.commit()
9
    1 usage
10    def view_students(self):
11        student_vo_list = db.session.query(StudentV0, CourseV0).join(CourseV0, StudentV0.course_id==CourseV0.co
12        # student_vo_list = StudentV0.query.all()
13        print(student_vo_list, "*****")
14        return student_vo_list
15
    1 usage
16    def delete_student(self, student_vo):
17        student_vo_list = StudentV0.query.get(student_vo.student_id)
18        db.session.delete(student_vo_list)
19        db.session.commit()
20        return student_vo_list

```

```

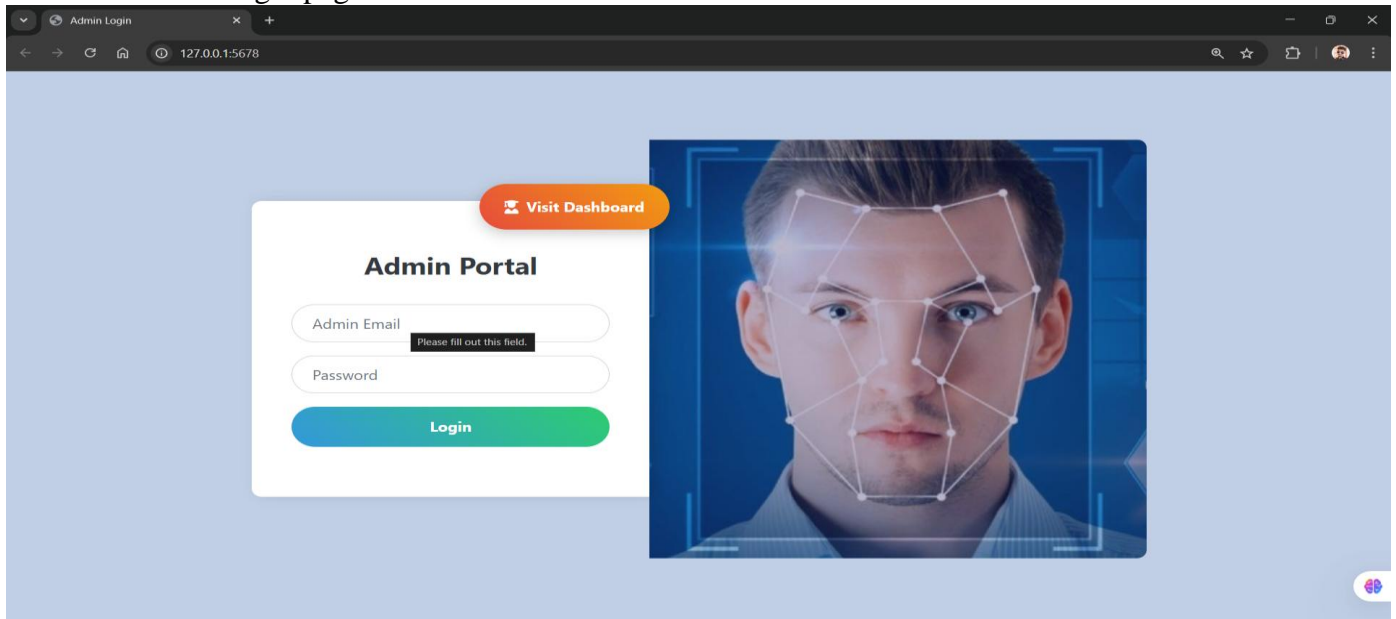
1 from base import db
2
12 usages
3 class LoginV0(db.Model):
4     __tablename__ = 'login_table'
5     login_id = db.Column('login_id', db.Integer, primary_key=True,
6                         autoincrement=True)
7     login_username = db.Column('login_username', db.String(255),
8                               nullable=False)
9     login_password = db.Column('login_password', db.String(255),
10                              nullable=False)
11     login_role = db.Column('login_role', db.String(100), nullable=False)
12     login_status = db.Column('login_status', db.Boolean, nullable=False)
13
    1 usage (1 dynamic)
14    def as_dict(self):
15        return {
16            'login_id': self.login_id,
17            'login_username': self.login_username,
18            'login_password': self.login_password,
19            'login_role': self.login_role,
20            'login_status': self.login_status,
21        }

```

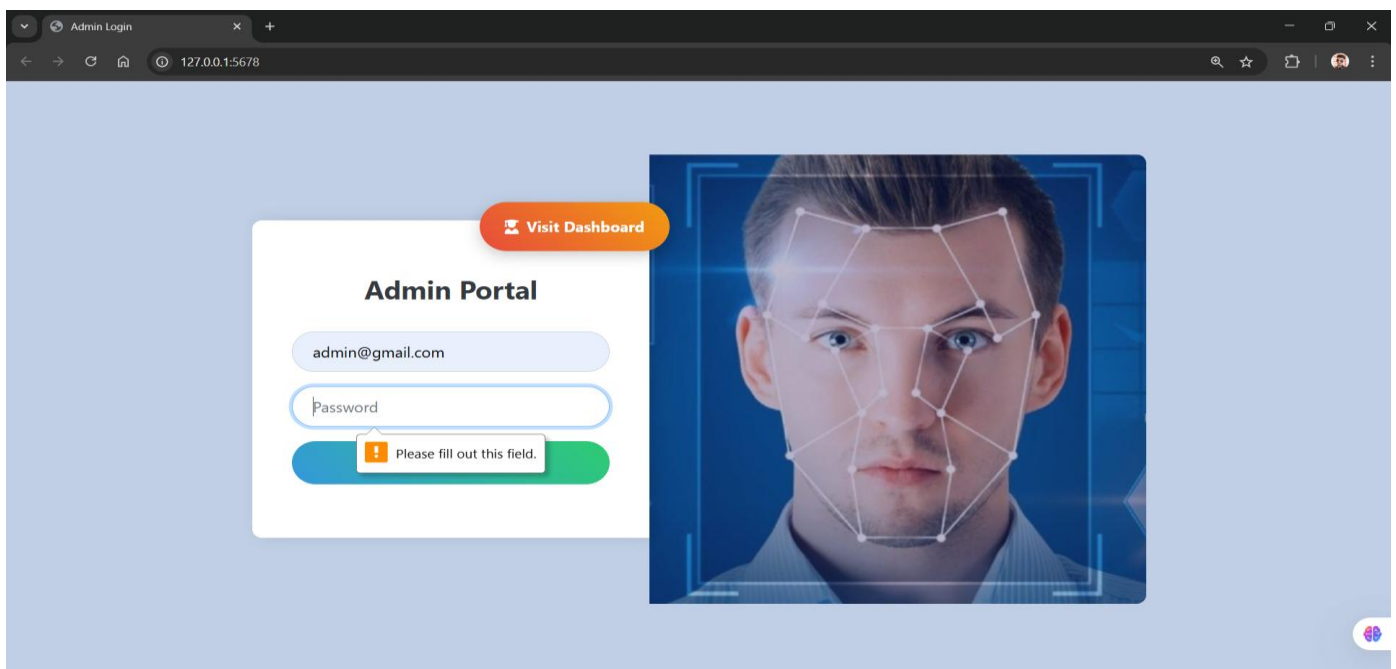
## 4.2 Screen Shots

### Login Page for Admin :

- This is the login page for the Admin

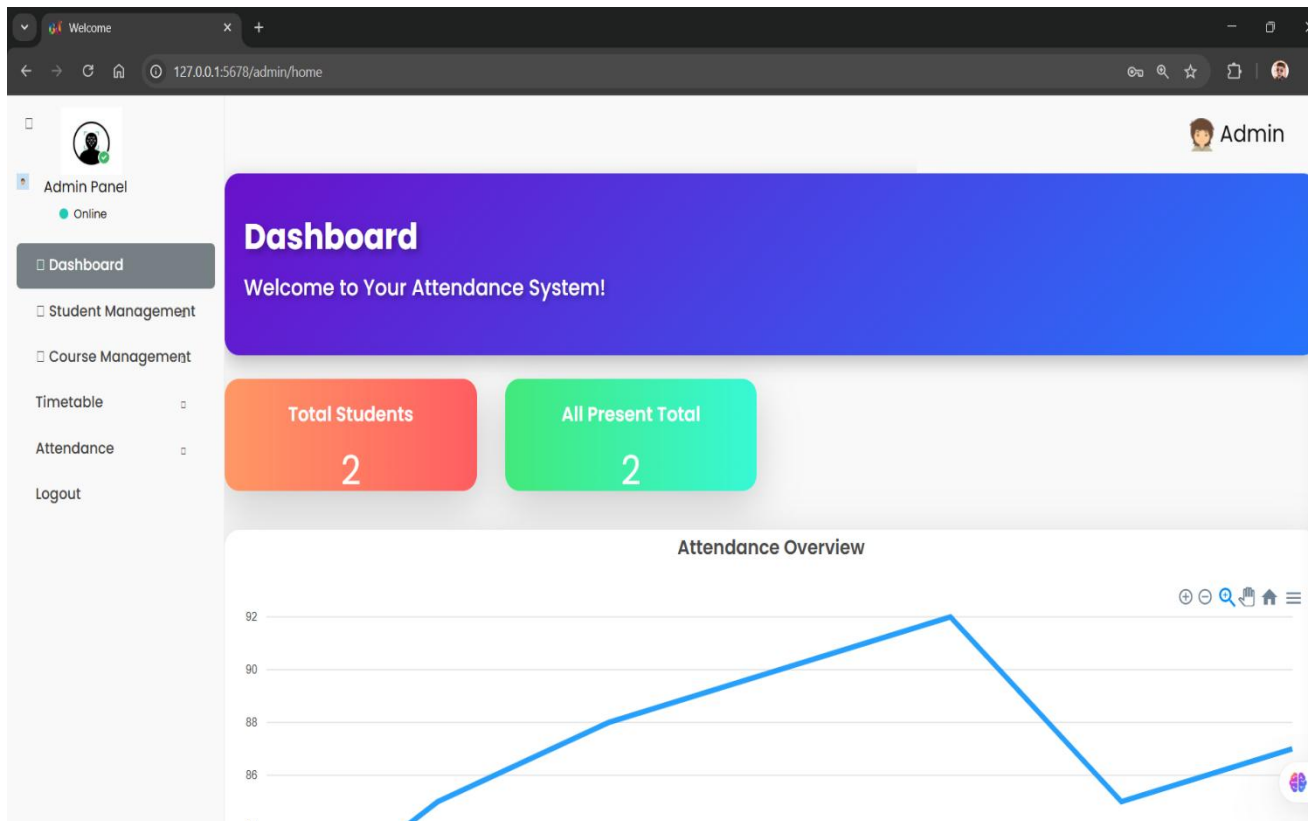


- This are the login page validations where all the fields are required.



## Admin Dashboard :

- This page is admin dashboard.





- Add student at admin side

**Add Student**

Roll No \*  
1

First Name \*  
Dhvij

Last Name \*  
Shah

Gender \*  
☒ Male 
 ☐ Female

Date Of Birth \*  
03-07-2002

Email Id \*  
dhvij52@gmail.com

Course Name  
BCA

Mobile No \*  
9998422383

Address \*  
Ahmedabad

Image\*  
 Dhvij.jpg

- View student at admin side

**Student Data**

Show 10 entries Search:

Roll No	First Name	Last Name	Gender	Date of Birth	Course	Email	Mobile	Address	Action
1	Dhvij	Shah	Male	2002-07-03	BCA	dhvij52@gmail.com	9998422384	Ahmedabad	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
2	Sahil	Vaghela	Male	2000-04-19	BCA	sahil@gmail.com	9985232112	Vadodara	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

Showing 1 to 2 of 2 entries Previous 1 Next

- Admin add excel sheet time table

Admin Panel  
Online

Dashboard

Student Management

Course Management

Timetable

Attendance

Logout

### Add TimeTable

File\*

Choose File Book1.xlsx

Submit

© 2025 Student-Attendance Teame. All Rights Reserved.

- Admin can view time table

### Timetable Data

Add Timetable

Show 10 entries

Search:

Date	Day	Start Time	End Time	Subject	Actions
2025-04-01	Monday	09:00	10:00	java	
2025-04-02	Tuesday	11:00	12:00	python	
2025-04-03	Wednesday	12:00	01:00	HTML	
2025-04-04	Thursday	01:00	02:00	Operating System	
2025-04-05	Friday	02:00	03:00	PHP	
2025-04-06	Saturday	03:00	04:00	Java Script	

Showing 1 to 6 of 6 entries

Previous 1 Next

- Admin can add course

**Add Course**

**Course Name \***

BCA

**Course Description \***

BCA stands for Bachelor of Computer Applications. It's a three-year undergraduate degree program that focuses on computer applications and software development.

Submit

© 2025 Student-Attendance Teame. All Rights Reserved.

- Admin can view course

**Course Data**

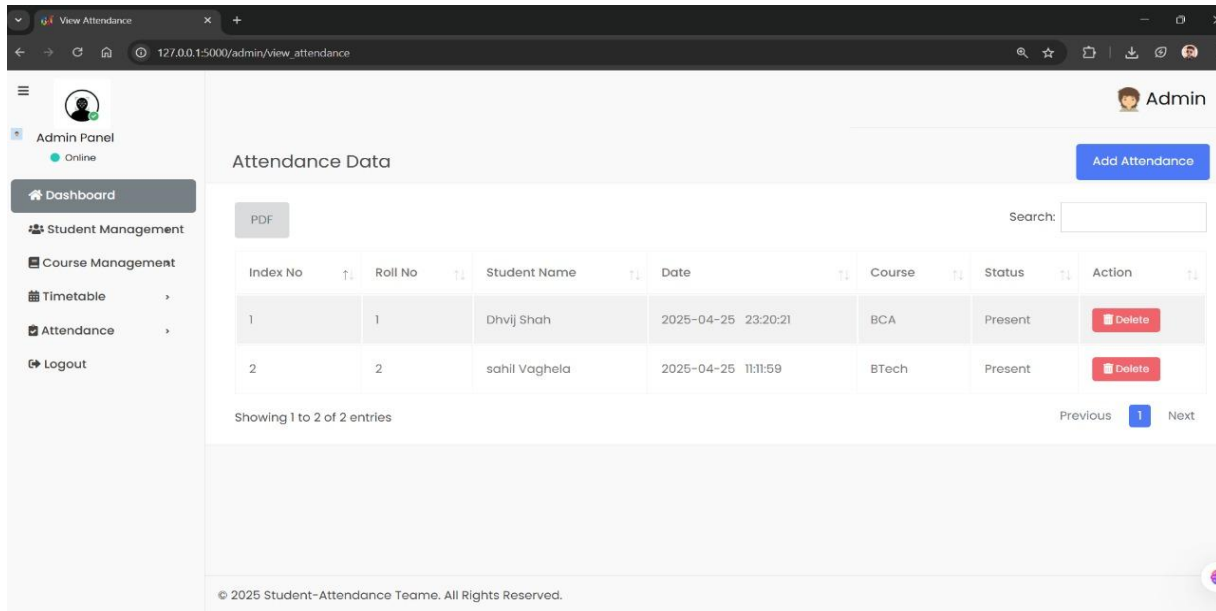
Show 10 entries Search:

#	Course Name	Course Description	Action
1	BCA	BCA stands for Bachelor of Computer Applications. It's a three-year undergraduate degree program that focuses on computer applications and software development.	

Showing 1 to 1 of 1 entries Previous 1 Next

© 2025 Student-Attendance Teame. All Rights Reserved.

- Admin can manage attendance



Admin Panel

Attendance Data

PDF

Search:

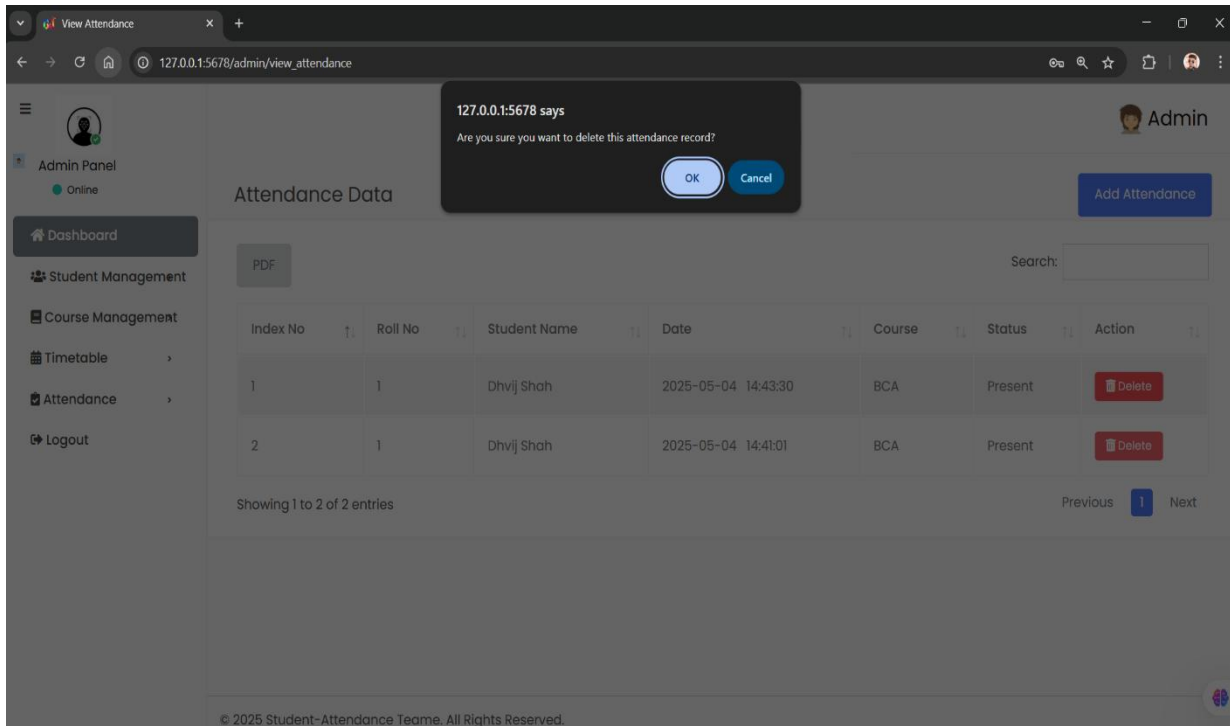
Index No	Roll No	Student Name	Date	Course	Status	Action
1	1	Dhvi Shah	2025-04-25 23:20:21	BCA	Present	Delete
2	2	sahil Vaghela	2025-04-25 11:11:59	BTech	Present	Delete

Showing 1 to 2 of 2 entries

Previous 1 Next

© 2025 Student-Attendance Teame. All Rights Reserved.

- Admin can delete the attendance



Admin Panel

Attendance Data

PDF

Search:

Index No	Roll No	Student Name	Date	Course	Status	Action
1	1	Dhvi Shah	2025-05-04 14:43:30	BCA	Present	Delete
2	1	Dhvi Shah	2025-05-04 14:41:01	BCA	Present	Delete

Showing 1 to 2 of 2 entries

Previous 1 Next

© 2025 Student-Attendance Teame. All Rights Reserved.

- Admin can edit the timetable

The screenshot shows a web browser window with the URL `127.0.0.1:5678/admin/edit_timetable?timetableId=1`. The page title is "Edit Timetable". On the left, there is a sidebar menu with the following items: "Dashboard", "Student Management", "Course Management", "Timetable", "Attendance", and "Logout". The "Timetable" item is currently selected. The main content area contains a form with the following fields:
 

- Date:** A text input field containing "01-04-2025" with a calendar icon on the right.
- Day of Week:** A dropdown menu showing "Monday".
- Start Time:** A text input field containing "09:00" with a clock icon on the right.
- End Time:** A text input field containing "10:00" with a clock icon on the right.
- Subject Name:** A text input field containing "C++".

 At the bottom of the form are two buttons: "Update" (in blue) and "Cancel" (in grey). A "Back to List" button is located in the top right corner of the main content area.

- Admin can edit the course

The screenshot shows a web browser window with the URL `127.0.0.1:5678/admin/edit_course?courseId=1`. The page title is "Edit Course". On the left, there is a sidebar menu with the following items: "Admin Panel", "Dashboard", "Student Management", "Course Management", "Timetable", "Attendance", and "Logout". The "Course Management" item is currently selected. The main content area contains a form with the following fields:
 

- Course Name \*:** A text input field containing "BCA".
- Course Description \*:** A text area containing "BCA stands for Bachelor of Computer Applications. It's a three-year undergraduate degree program that focuses on computer applications and software development."

 Below the form is a "Submit" button. In the top right corner, there is a user profile icon and the name "Admin". At the bottom of the page, there is a footer that reads "© 2025 Student-Attendance Teame. All Rights Reserved."

- Admin can delete the student

The screenshot shows the 'Student Data' management page. A confirmation dialog is displayed in the center, asking 'Are you sure you want to delete this student?' with 'OK' and 'Cancel' buttons. The dialog is titled '127.0.0.1:5678 says'. The background interface includes a sidebar with navigation options: Dashboard, Student Management, Course Management, Timetable, Attendance, and Logout. The main content area shows a table of student data with columns: Roll No, First Name, Last Name, Gender, Date of Birth, Course, Email, Mobile, Address, and Action. Two students are listed: Dhvij Shah (Male, 2002-07-03, BCA) and Sahil Vaghela (Male, 2000-04-19, BCA). The 'Action' column contains edit and delete icons for each student.

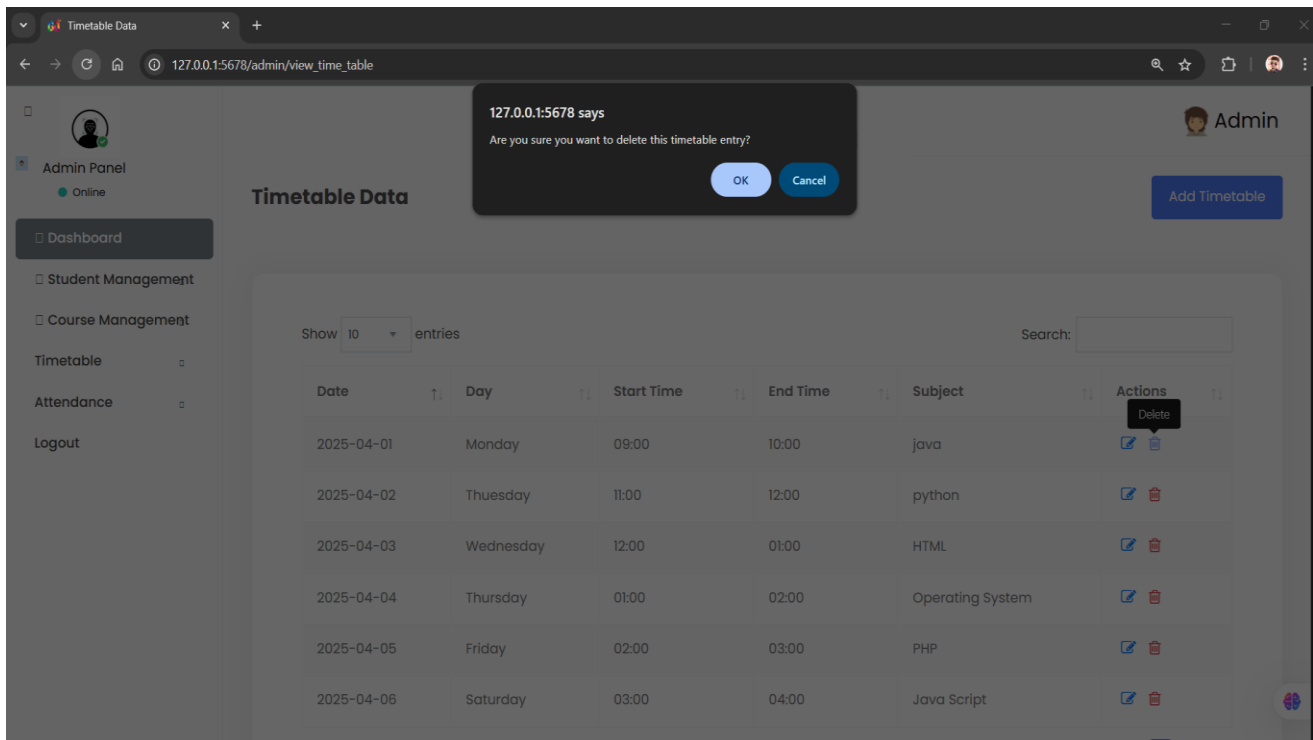
Roll No	First Name	Last Name	Gender	Date of Birth	Course	Email	Mobile	Address	Action
1	Dhvij	Shah	Male	2002-07-03	BCA	dhvij52@gmail.com	9998422384	Ahmedabad	[Edit] [Delete]
2	Sahil	Vaghela	Male	2000-04-19	BCA	sahil@gmail.com	9985232112	Vadodara	[Edit] [Delete]

- Admin can delete the course

The screenshot shows the 'Course Data' management page. A confirmation dialog is displayed in the center, asking 'Are you sure you want to delete this course?' with 'OK' and 'Cancel' buttons. The dialog is titled '127.0.0.1:5678 says'. The background interface includes a sidebar with navigation options: Dashboard, Student Management, Course Management, Timetable, Attendance, and Logout. The main content area shows a table of course data with columns: #, Course Name, Course Description, and Action. One course is listed: BCA (Bachelor of Computer Applications). The 'Action' column contains edit and delete icons for the course.

#	Course Name	Course Description	Action
1	BCA	BCA stands for Bachelor of Computer Applications. It's a three-year undergraduate degree program that focuses on computer applications and software development.	[Edit] [Delete]

- Admin can delete the timetable

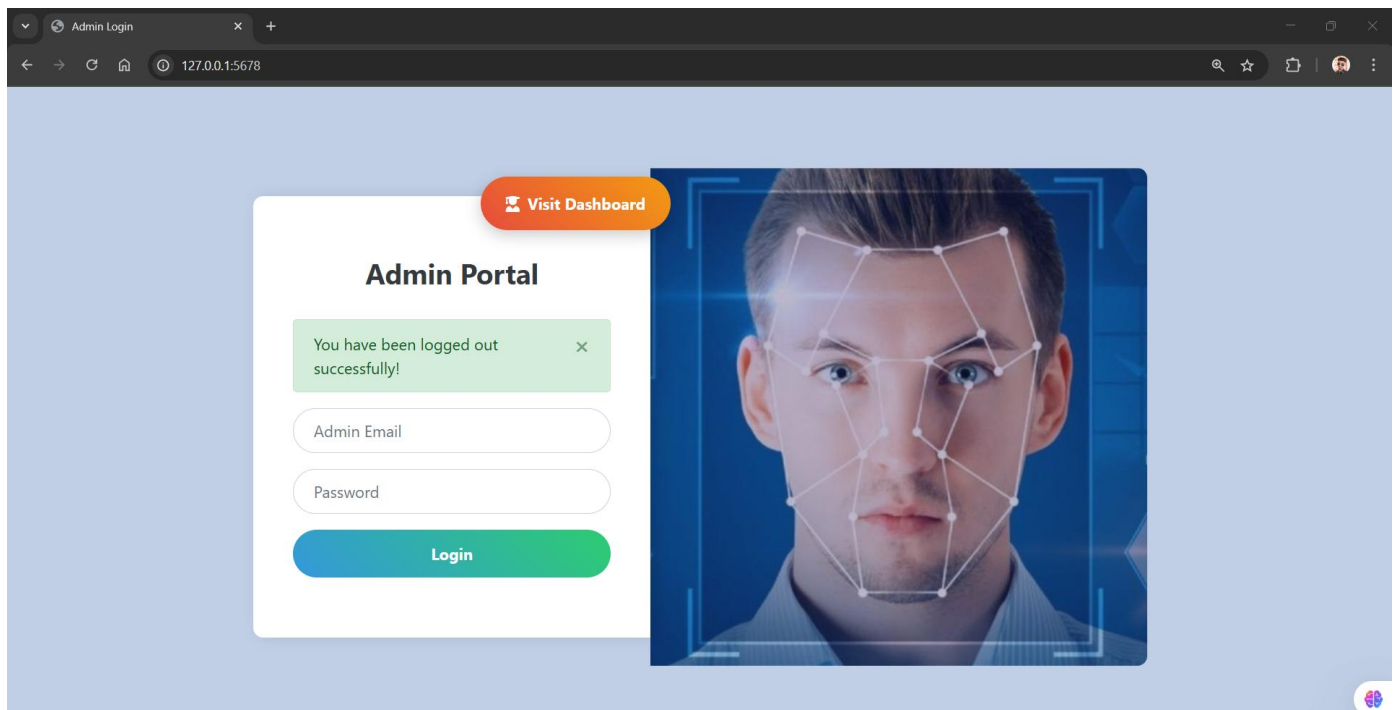


- See attendance reports via pdf

### Attendance Report

Index No	Roll No	Student Name	Date	Course	Status
1	1	Dhvij Shah	2025-04-25 23:20:21	BCA	Present
2	2	sahil Vaghela	2025-04-25 11:11:59	BTech	Present
3	3	Jay Makwana	2025-04-26 10:08:57	BCA	Present

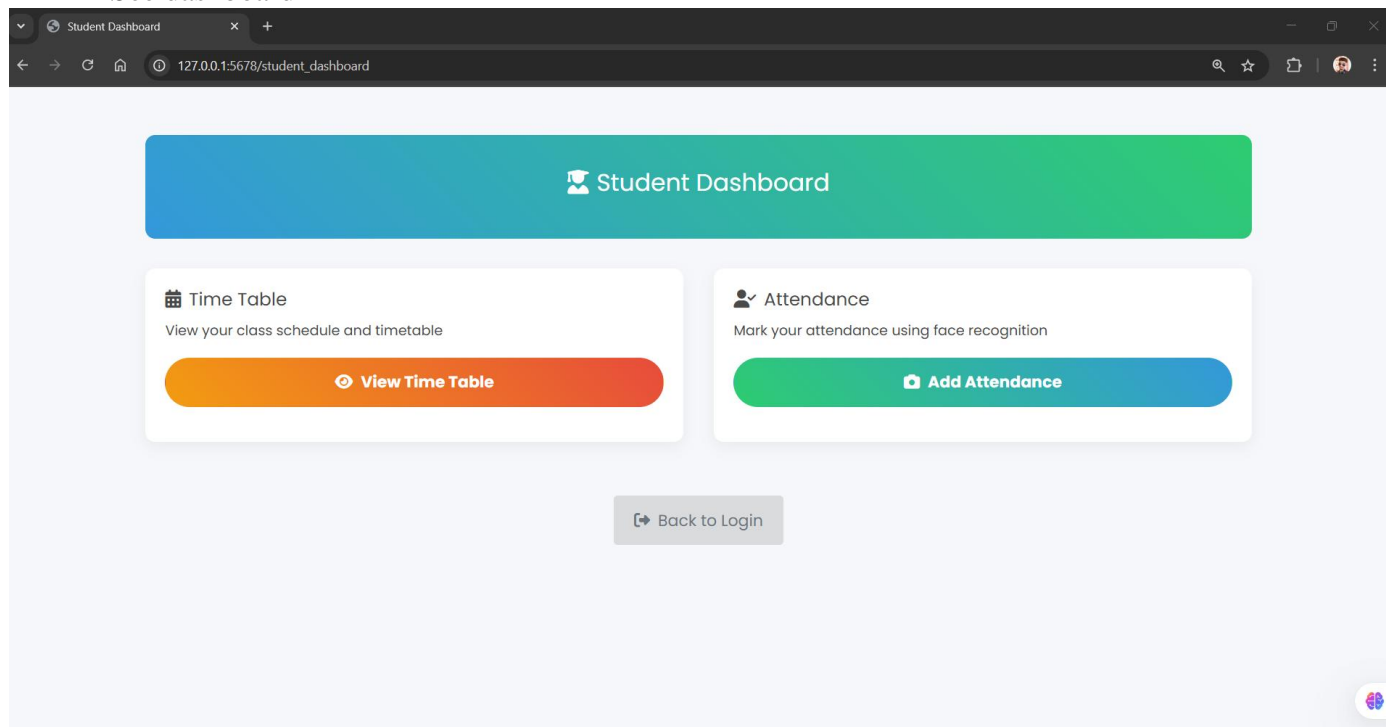
- Using this log out admin will logged out.



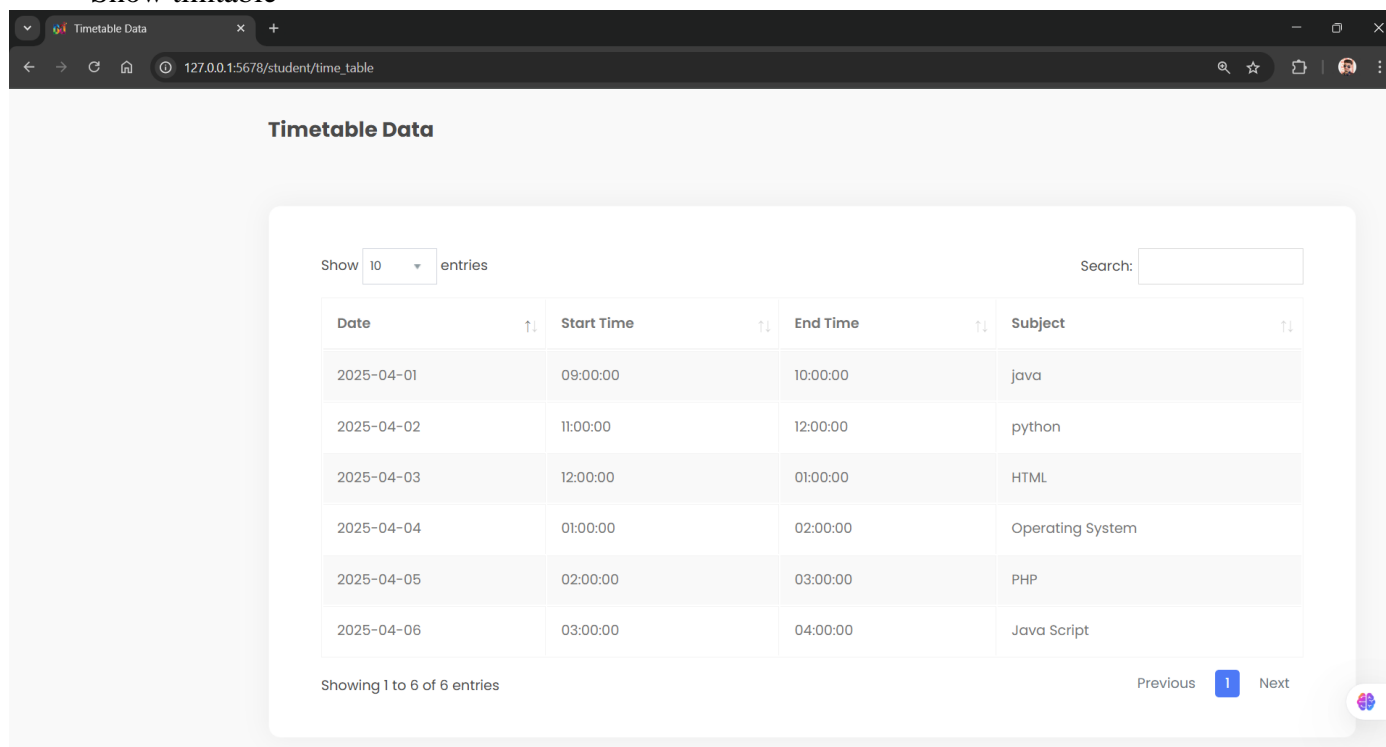


# Student Side

- See dashboard



- Show timetable



- Student can capture Attendance








## **5. Agile Documentation**

## 5.1 Agile Project Charter

Project Charter				
Project Title	Student Presence Monitoring System			
Project Start Date	01/02/2025	Project End Date	27/04/2025	
Business Need				
<p>The <b>Student Presence Monitoring System</b> is a smart attendance management platform that enables educational institutions to track and manage student attendance efficiently using face recognition. It provides a modern, accurate, and automated solution for administrators and faculty, enhancing institutional productivity, reducing manual workload, and ensuring a seamless user experience.</p>				
Project Scope				
<ul style="list-style-type: none"> <li><input checked="" type="checkbox"/> Develop a robust <b>student panel</b> for real-time attendance viewing, personal attendance history, and profile management.</li> <li><input checked="" type="checkbox"/> Implement a secure <b>admin panel</b> to manage student records, monitor attendance logs, and generate reports.</li> <li><input checked="" type="checkbox"/> Provide features such as <b>face recognition, daily/weekly/monthly reports, and automatic notifications</b> to students and parents.</li> <li><input checked="" type="checkbox"/> Ensure smooth and secure authentication, with role-based access for students and administrators.</li> </ul>				
Objective				
<p>To design and develop an intelligent and interactive attendance monitoring system that uses facial recognition for accurate presence tracking, supports administrative and academic needs, and provides a user-friendly interface for both students and administrators.</p>				

## 5.2 Agile Roadmap/Schedule

	Month 1	Month 2	Month 3	Month 4
	03-Jan-25	15-Feb-25	10-Mar-25	24-Apr-25
	Version 1	Version 2	Version 3	Version 4
	Started our Application Designing	Started Making Modules	Started Testing	Make Documentation
	Completed our website Designing	Completed Making Modules	Completed Testing	Completed Documentation
	25%	30%	85%	100%

### **5.3 Agile Project Plan**

Task Name	Priority	Owner	Start Date	End Date	Status
Sprint - 1 - Requirements Mapping and Design a System Flow					
Gather Requirements and User Stories	High	Dhvij	03-Jan-25	04-Jan-25	Completed
Design User Interface	High	Sahil	06-Jan-25	25-Jan-25	Completed
Define Technical Architecture	High	Jay	26-Jan-25	05-Feb-25	Completed
Sprint - 2 - Development & Coding					
Implement Admin Authentication	High	Dhvij, Sahil, Jay	10-Feb-25	09-Mar-25	Completed
Develop Student Management	High	Dhvij, Sahil, Jay	10-Mar-25	14-Mar-25	Completed

## 5.4 Agile User Story

User Story	Role	Reason/Goal	User Story
As an <b>admin</b> , I want to add, update, and delete student profiles, so that the database remains accurate.	Administrator	Effective student management	As an <b>admin</b> , I want to add, update, and delete student profiles, so that the database remains accurate
As a <b>student</b> , I want to receive alerts if my attendance is low, so that I can take corrective action early.	Student	Encourage better attendance responsibility.	As a <b>student</b> , I want to receive alerts if my attendance is low, so that I can take corrective action early.
As an <b>admin</b> , I want to generate daily and monthly attendance reports, so that I can monitor overall student attendance.	Administrator	Monitor and analyze student attendance trends.	As an <b>admin</b> , I want to generate daily and monthly attendance reports, so that I can monitor overall student attendance.

## **5.5 Agile Release Plan**

Sprint	Task	Start	End	Duration	Release Date	Goal
1	Requirement Gathering and Analysis	03-jan-25	04-jan-25	1 Days	06-jan-25	Define clear project requirements and scope
2	UI/UX Design	08-jan-25	10-Feb-25	32 Days	11-Feb-25	Deliver a user-friendly design for Students
3	Backend Development and Integrations	12-Feb-25	10-Mar-25	28 Days	11-Mar-25	Build a robust and scalable backend.
4	Feature Integration and Testing	12-Mar-25	29-Mar-25	18 Days	30-Mar-25	Integrate features and perform internal testing.
5	Deployment	11-Apr-25	21-Apr-25	10 Days	22-Apr-25	Resolve issues and release the platform.



## **5.6 Agile Sprint**

Task ID	Task	Estimated Effort (hours)	Planned Effort (hours)	Status
1	Requirement Analysis	50	50	Complete
2	UI/UX Wireframe Development	70	70	Complete
3	Backend Development	220	220	Complete
4	Frontend Development	120	120	Complete
5	User Testing	70	70	Complete
6	Debugging	60	60	Complete

## **5.7 Agile Test Plan**

<b>Test</b>	<b>Action</b>	<b>Expected Results</b>	<b>Actual Results</b>	<b>Pass?</b>
1	Admin Login	Should be able to Login	Admin can register	✓
2	Face Recognition Attendance	System should recognize student's face and mark attendance.	Face recognized and attendance marked.	✓
3	Generate Attendance Report	Admin should generate daily/monthly attendance reports.	Reports generated accurately.	✓
4	Update Student Details	Admin should be able to edit/update student information	Student details updated successfully.	✓
5	Delete Student Profile	Admin should be able to delete a student profile	Student profile deleted.	✓

## **6. Proposed Enhancement**

To improve efficiency, accuracy, and user experience, the proposed system introduces the following enhancements over traditional attendance methods:

### **1. Facial Recognition-Based Attendance**

- Automatically detects and marks student presence using real-time facial recognition.
- Eliminates the need for manual roll calls or biometric punching.

### **2. Digital Student Enrollment**

- Admins can register students with photo and academic details.
- Ensures a secure and verifiable identity database for recognition.

### **3. Real-Time Attendance Monitoring**

- Attendance is tracked and updated instantly as students are detected.

### **4. Analytics and Reporting**

- Generates attendance reports (daily, weekly, monthly).

## **7. Conclusion**

- The **Student Presence Monitoring System** offers an innovative and efficient approach to attendance management by integrating facial recognition technology with a user-friendly web interface. It effectively addresses the limitations of traditional attendance methods by automating the process, reducing administrative workload, and ensuring accurate and real-time attendance tracking.
- With distinct panels for students and administrators, the system supports seamless interaction, secure data management, and insightful reporting. Its modular and scalable design makes it adaptable to various educational institutions, contributing to enhanced productivity, transparency, and student engagement.
- Overall, this system not only streamlines daily attendance operations but also lays the foundation for future advancements in smart education systems.

## **8. Bibliography**

### 1.opencv python tutorials

- [https://docs.opencv.org/4.x/d6/d00/tutorial\\_py\\_root.html](https://docs.opencv.org/4.x/d6/d00/tutorial_py_root.html)

### 2.Flask Documentation:-

- <https://flask.palletsprojects.com/en/stable/>

