PROJECT REPORT

on

Student Attendance Management System

Submitted in partial fulfillment of the requirements
for the award of degree

MASTER OF COMPUTER APPLICATIONS

of

KLE TECHNOLOGICAL UNIVERSITY

By

Mr. Steven Manchala
(SRN: 01FE21MCA054)

DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS
KLE TECHNOLOGICAL UNIVERSITY
Vidyanagar, Hubballi - 580031 Karnataka.
August- 2023

A Project Report on

Student Attendance Management System

Submitted in partial fulfillment of the requirements
for the award of degree

MASTER OF COMPUTER APPLICATIONS

of

KLE TECHNOLOGICAL UNIVERSITY

By

Mr. Steven Manchala
(SRN: 01FE21MCA0054)

Under the guidance of

Dr. Ashok Chikaraddi

DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS
KLE TECHNOLOGICAL UNIVERSITY
Vidyanagar, Hubballi - 580031 Karnataka.
August- 2023

DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS
KLE TECHNOLOGICAL UNIVERSITY

CERTIFICATE

This is to certify that the project work entitled "**Student Attendance Management System**" submitted in partial fulfillment of the requirements for the award of degree of Master of Computer Applications of KLE Technological University, Hubballi, Karnataka, is a result of the bonafide work carried out by **Mr. Steven Manchala (SRN: 01FE21MCA054)** during the academic year 2022-2023.

Dr. Ashok Chikaraddi	**Dr. P.R.Patil, Ph.D.**	**Dr. B. S. Anami**
Project Guide, Assistant Professor, MCA Dept	Professor, Head of Department MCA Dept	Registrar, KLE Technological **University**

<u>Viva-Voce Examination</u>

Name of the Examiners   Signature with Date

1. ______________________________   ________________

2. ______________________________   ________________

ACKNOWLEDGEMENT

Every successful completion of any undertaking would be complete only after we remember and thank the almighty, the parents, the teachers and the personalities, who directly or indirectly helped and guided during the execution of that work. The success of this work is equally attributed to all well-wishers who have encouraged and guided throughout the execution.

I express my deepest gratitude to Internal Guide **Dr. Ashok Chikaraddi,** for their guidance and assistance throughout the project with great interest.

I avail this opportunity to express deepest gratitude to **Dr. P. R. Patil, Professor & HoD** for encouraging us with his valuable tips.

I am grateful to **Dr. B. S. Anami,** Registrar for his blessings. And we are grateful to **K L E Technological University Hubballi** which has given us a bright future. I would like to thank all the people for their guidance and valuable suggestions throughout the project.

I also thank all teaching and non-teaching staff members of the MCA department for their invaluable cooperation.

I would like to thank "**Pravinya Information Technology Service**" for their guidance and valuable suggestions throughout the project.

Finally, I would like to express our sincere thanks to our **Parents and Friends** for their enormous encouragement and all other who have extended their helping hands towards completion of our project.

Steven Manchala

ABSTRACT

The Student Attendance Management System is a digital solution designed to optimize and streamline the process of recording and tracking attendance in educational institutions. This system offers an automated and efficient approach to attendance management, eliminating the traditional manual methods and associated challenges. With a user-friendly interface accessible to students, teachers, administrators, and parents, the system simplifies the attendance recording process, allowing for both manual and automated attendance marking based on predefined schedules. The system also generates comprehensive attendance reports, aiding in monitoring attendance trends, identifying patterns, and making data-driven decisions. In addition, the system can send timely notifications to relevant stakeholders, fostering better communication and engagement. The future enhancement of incorporating advanced analytics and predictive insights promises to further elevate the system's capabilities by providing educators with actionable information to proactively address attendance issues and implement personalized interventions. Overall, the Student Attendance Management System enhances administrative efficiency, improves student engagement, and contributes to a more effective and data-informed educational environment.

INDEX

List of Figures

List of Tables

Chapter 1

INTRODUCTION

The "Student Attendance Management System"(SAMS) is a specialized software solution meticulously crafted to streamline the process of monitoring and recording student attendance in a college environment. This innovative system empowers designated faculty members to efficiently mark daily attendance for students under their purview. By assigning exclusive usernames and passwords based on their respective subjects, educators are granted controlled access to the system, ensuring data security and accuracy.

The SAMS's functionalities include easy attendance marking, real-time reporting, and notifications for students and teachers. This simplicity and automation not only make the system user-friendly but also contribute to accurate attendance records. As this report unfolds, we will explore how the SAMS optimizes attendance management, supports efficient communication, and makes administrative tasks smoother in today's tech-savvy educational environments.

In essence, the "Student Attendance Management System" not only simplifies the often complex task of attendance tracking but also promotes a data-driven approach to academic management, benefiting both educators and students alike.

1.1 Literature survey:

The Existing system is a manual entry for the students. Here the attendance will be carried out in the hand written registers. It will be a tedious job to maintain the record for the user. The human effort is more here. The retrieval of the information is not as easy as the records are maintained in the hand written registers. This application requires correct feed on input into the respective field. Suppose the wrong inputs are entered, the application resist to work. So the user finds it difficult to use.

i. IOT Based Cloud Integrated Smart Classroom and Sustainable Campus

This paper proposed an idea of recording attendance using face recognition technique and storing the data using IOT. In this method Arduino is used as a microcontroller. Cameras are used to detect the face of an individual or group of pupils. Based on the information that is stored in prior, the faces are recognized and the attendance is recorded and the database is obtained. This method provides better results in short span of time but fails to produce most accurate results. There are some chances of some errors.

ii. Attendance Management System through Fingerprint

This paper proposed an idea of recording attendance using biometrics (fingerprint) for tracking attendance and storing the data using LAN. This paper provides a brief description about the usage, accessibility, accuracy, affordability and acceptance of biometric (fingerprint verification) system. In this system the data is fetched from the individual in the form of fingerprint and then it is verified with the data that was stored in prior and marks the attendance of an individual. Finally the database is also obtained. This method provides high accuracy results and consumes less time but it is not cost-effective.

iii. Student attendance system in classroom using face recognition technique

Here this paper gives an idea of recording attendance using face recognition technique. Also this paper provides a detailed description about the results and its analysis obtained from this method. Faces are recognized using cameras and the verification is done. Then the attendance is marked. This method is suitable only for moderate number of people and the results obtained are nearly 87% accurate. This method fails to recognize people in bulk quantity and causes error in results.

iv. Bluetooth Based Attendance Management

The instructors in universities and colleges take the attendance manually either by calling out individual's name or by passing around an attendance sheet for student's signature to confirm his/her presence. Using these methods is both cumbersome and time-consuming. Therefore a method of taking attendance using instructor's mobile telephone has been presented in this paper which is paperless, quick, and accurate. An application software installed in the instructor's mobile telephone enables it to query students' mobile telephone via Bluetooth connection and, through transfer of students' mobile telephones' Media Access Control (MAC) addresses to the instructor's mobile telephone, presence of the student can be confirmed.

v. Web-based laboratory attendance system by integrating RFID-ARDUINO technology

The proposed system aims to manage student’s attendance recording and provides the capabilities of tracking student absentee as well, supporting information services include students grading marks, daily timetable, lectures time and classroom numbers, and other student-related instructions provided by faculty department staff. Based on the results, the proposed attendance and information system is time-effective and it reduces the documentation efforts as well as, it does not have any power consumption. Besides, student’s attendance RFID based systems

1.2 Motivation:

The challenges in manual attendance systems include inaccuracies due to proxy attendance, time-consuming data entry, and difficulty in generating comprehensive attendance reports. These challenges underscore the motivation to develop an automated Student Attendance Management System. Automation addresses these issues by providing real-time data, reducing errors, and offering efficient data analysis for informed decision-making**.**

1.3 Objectives

The objective of this report is to comprehensively examine and evaluate the implementation, functionality, and impact of the Student Attendance Management System in educational institutions. The report aims to analyze how the system improves attendance tracking, enhances communication, and streamlines administrative processes, ultimately contributing to a more efficient and engaging educational environment.

1.4 Problem definition

The Student Attendance Management System addresses the challenge of accurately recording attendance in educational institutions and workplaces. The traditional approach of using manual methods, such as paper registers or spreadsheets, is cumbersome and error-prone. This leads to inefficiencies, inaccuracies, and difficulties in generating meaningful attendance reports. With the Student Attendance Management System, this problem is streamlined. The system automates the process of tracking attendance, reducing the time and effort required for teachers and administrators. It ensures accurate data entry, enables easy retrieval of historical attendance records, and simplifies the creation of comprehensive reports. By replacing manual processes with digital automation, the system enhances data accuracy, facilitates better communication with stakeholders, and ultimately contributes to improved decision-making based on reliable attendance insights.

Chapter 2

PROPOSED SYSTEM

The proposed Student Attendance Management System will modernize attendance tracking in institution. It uses automation for accurate recording, provides easy interaction, and offers real-time reporting and notifications for better communication. This system aims to improve administrative efficiency and student engagement.

2.1 Proposed System Description:

In this system, admin can login and check the already added data like department, staff, course, student, timetable, assign course and along with checking admin can also add the new data if needed, and admin can also perform the necessary actions like changing the status of the field also can delete the data, and can also change the details of the fields which are already filled while creating table.

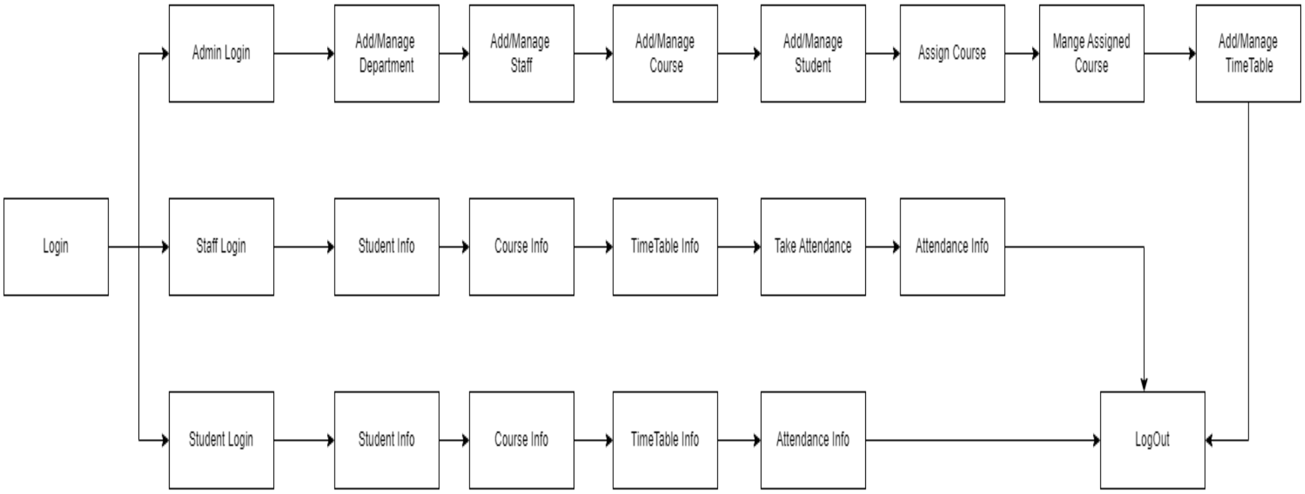


Fig 2.1 Block Diagram

2.2 Description of Target users

The Student Attendance Management System is designed to cater to a diverse range of users within the educational ecosystem. These users include:

- Teachers/Staff: The system provides teachers with a streamlined and efficient method to mark and manage student attendance. They can mark attendance for their respective classes, view historical attendance data, and receive real-time insights into student attendance patterns.
- Students: Students have the ability to access their own attendance records through the system. This feature empowers them to monitor their attendance performance, identify trends, and take proactive measures to improve their attendance if necessary.
- Administrators: Administrators play a pivotal role in overseeing the overall functionality of the system. They have access to administrative controls, enabling them to manage user accounts, configure system settings, and generate comprehensive attendance reports. This empowers administrators to make informed decisions based on attendance data

2.3 Advantages

The proposed system offers various benefits and uses:

- **Efficiency:** Faster and easier attendance marking for teachers.
- **Accurate Records:** Reliable digital records compared to manual paper-based systems.
- **Convenience:** Students and parents can access attendance records remotely.
- **Reports:** Automated generation of attendance reports for different periods.
- **Communication:** Improved communication between teachers and students.
- **Resource Savings:** Reduced paper usage, promoting environmental awareness.

2.4 Scope

The scope of the Student Attendance Management System encompasses various aspects and functionalities that contribute to streamlining attendance tracking and enhancing communication within educational institutions. The system aims to address the limitations of traditional manual attendance methods by introducing automation and digitalization

CHAPTER 3

SOFTWARE REQUIREMENT SPECIFICATION

3.1 Overview

The Software Requirements Specification (SRS) for the Student Attendance Management System outlines the comprehensive set of functional and non-functional requirements that the system must adhere to. It serves as a detailed roadmap for the development team and stakeholders, ensuring a clear understanding of the system's scope, features, and constraints. The SRS provides a concise overview of the system's purpose, its intended users (students, staff, administrators), and the problems it aims to address (manual attendance tracking, communication gaps). It outlines the key features such as attendance recording, reporting, and notifications, while also specifying user roles, access controls, and security measures. Additionally, the SRS details the system's performance expectations, scalability, and compatibility requirements, as well as considerations for data privacy and regulations. By providing a comprehensive snapshot of the project, the SRS acts as a foundation for effective communication, development, testing, and ultimately, the successful implementation of the Student Attendance Management System.

3.2 Requirement Specification

The Requirement Specification for the Student Attendance Management System delineates the detailed functional and non-functional prerequisites that the system must fulfill. It encapsulates the purpose of the system, its intended users, and the problems it aims to resolve. This document expounds upon vital features such as attendance marking, reporting, notifications, and leave management, outlining user roles, authentication mechanisms, and data protection measures. It elucidates the system's performance expectations, scalability, and compatibility requisites. Security considerations, data privacy, and regulatory compliance are also highlighted. In essence, the Requirement Specification serves as a comprehensive guide, forming the basis for seamless communication, development, and successful deployment of the Student Attendance Management System.

3.2.1 Functional Requirements:

Functional requirements refer to the specific features or functions that a software system must have to meet the needs of its users. These requirements describe what the software must do, and often include things like user interface design, data storage and retrieval, and other core functionalities

3.2.1.1 Login

The user can login or sign in to his account by giving the credentials like email, password.

Input: Enter email/username and password.

Processing: Verify the user

Output: Admin will enter the Homepage.

3.2.1.2 Department Management

Input: Enter Department information to be created

Processing: Verify the information and add it to the database

Output: Department has been created

3.2.1.3 Staff Management

Input: Enter Staff information to be created

Processing: Verify the information and add it to the database

Output: Staff has been created

3.2.1.4 Student Management

Input: Enter Student information to be created

Processing: Verify the information and add it to the database

Output: Student has been created

3.2.1.5 Course Management

Input: Enter Course information to be created

Processing: Verify the information and add it to the database

Output: Course has been created

3.2.1.6 Timetable Management

Input: Enter Timetable information to be created

Processing: Verify the information and add it to the database

Output: Timetable has been created

3.2.1.6.1 Logout

The admin can logout anytime from the account.

3.3.2 Use Case Diagrams

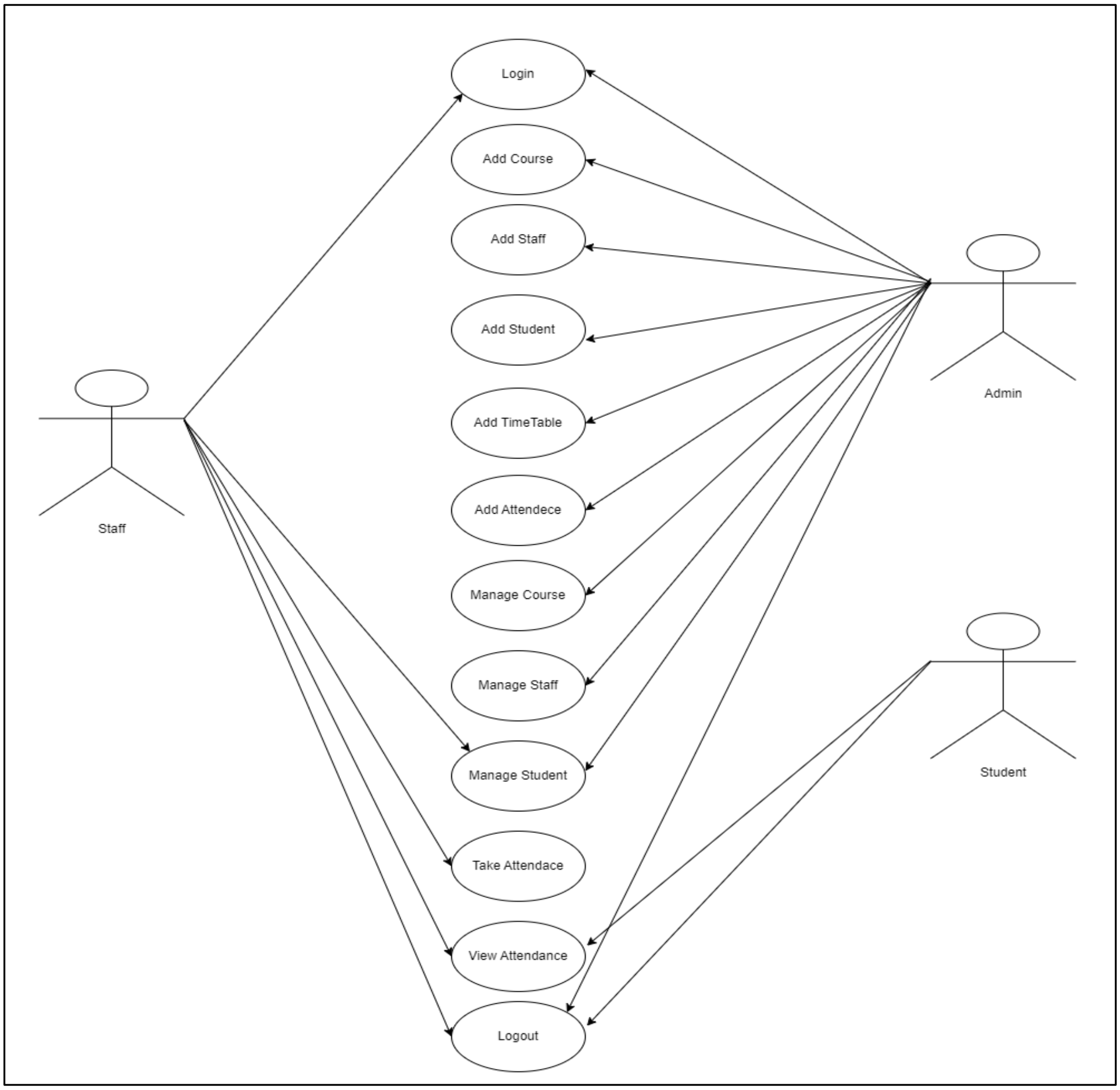


Fig 3.1 Use Case Diagram

3.2 Use case description

A use case diagram is a visual representation of the functional interactions between different actors (users or external entities) and the system they are interacting with. In the context of an Student Attendance Management System, let's create a use case diagram that includes three main actors: Admin, Staff, and Student

i. **Admin:**

- The admin actor interacts with the system to manage user accounts, configure system settings, and generate attendance reports for all classes and subjects.
- Admin can access functionalities like user management, system configuration, and report generation.

ii. **Staff:**

- The staff actor interacts with the system to mark attendance for their respective Classes and subjects.
- Staff can also view attendance records for the classes they are responsible for.
- This actor has access to functionalities related to attendance recording and viewing.

iii. **Student:**

- The student actor interacts with the system to view their own attendance records.
- Students can access their personal attendance history to monitor their attendance performance.
- This actor's interactions focus on viewing their attendance records.

3.3.3 Nonfunctional Requirements

Here is a summary of the non-functional requirements for the Student Attendance Management System:

1. Usability:

- The system's interface must be intuitive and user-friendly.
- Users should require minimal training to navigate and use the system effectively.

2. Performance:

- The system must handle a substantial number of concurrent users during peak periods.
- Attendance marking and data retrieval processes must be swift to avoid delays.

3. Scalability:

- The system should be designed to accommodate increasing data volumes and user traffic as the institution expands.

4. Security:

- Sensitive attendance data must be securely transmitted and stored.
- Robust authentication and authorization mechanisms should be implemented.
- Measures must be in place to prevent unauthorized access and data breaches.

5. Reliability:

- The system must be consistently available and accessible, with minimal downtime.

6. Compatibility:

- The system should function smoothly across various devices and preferences.

7. Maintainability:

- The system's codebase should be well-organized and properly documented, facilitating ease of maintenance.
- Updates and enhancements should be implementable without causing significant disruptions.

3.4 Software and Hardware requirements

3.4.1 Software requirements

- **Front End:**

  HTML, CSS, JAVASCRIPT, BOOTSTRAP

- **Frame Work:**

  Python 3.11-Django

- **Back End**:

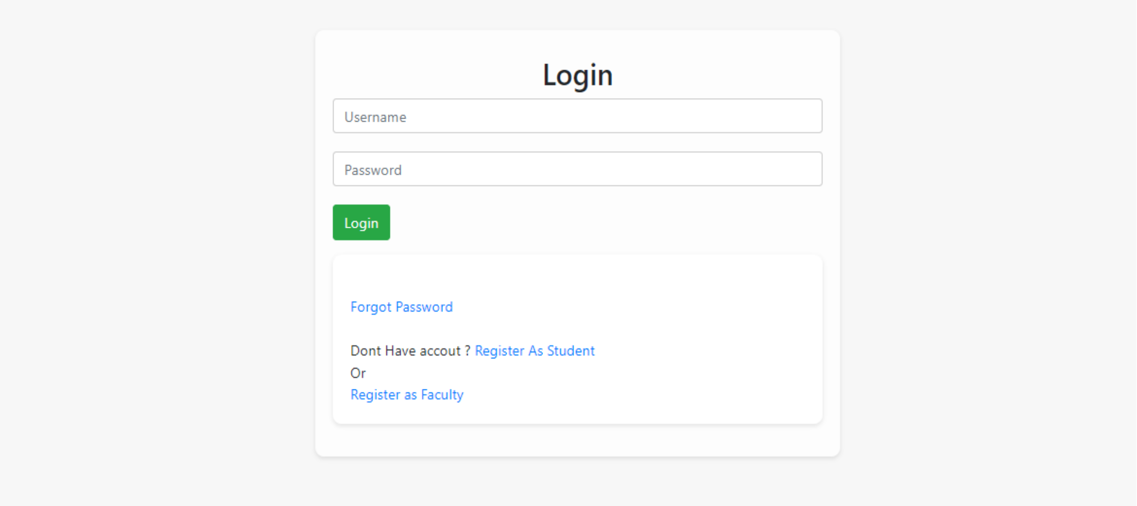
  MySQL Workbench

3.4.2 Hardware requirements

- **Minimum RAM**:-4GB
- **Hard Disk**:-128 GB
- **Processor**:-Intel Pentium 4(1.50 GHZ) or above

3.5 GUI OF PROPOSED SYSTEM:

i. **Login :** Admin ,staff or student need to enter their login credentials to login

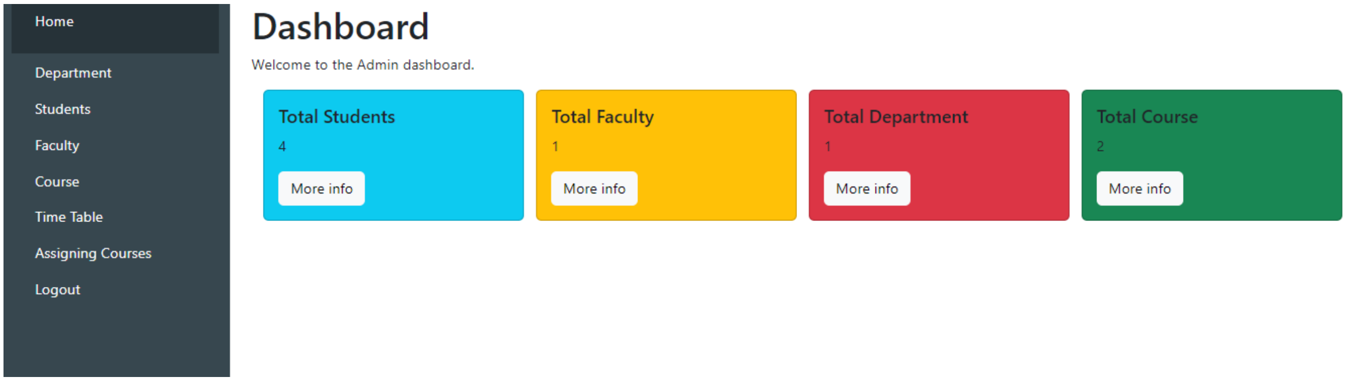


Dont Have accout ? Register As Student

Fig 3.2 Login Page

Fig 3.2 Login Page shows that the system requires login credentials for access, catering to admins, staff, and students, ensuring secure authentication and personalized interactions.

ii. **Admin Dashboard**

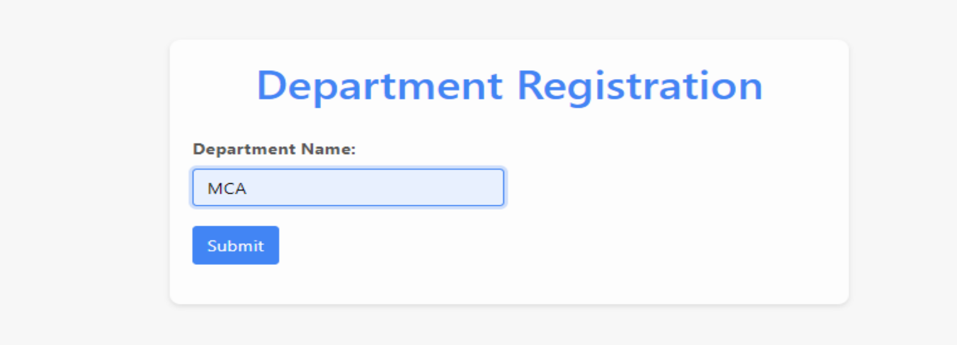


Welcome to the Admin dashboard.

Fig 3.3 Admin Dashboard

Fig 3.3 Admin Dashboard shows that the admin dashboard offers a centralized hub for administrators to manage attendance records, user permissions, and system settings efficiently and effectively.

iii. **Create Department:** Admin can create department



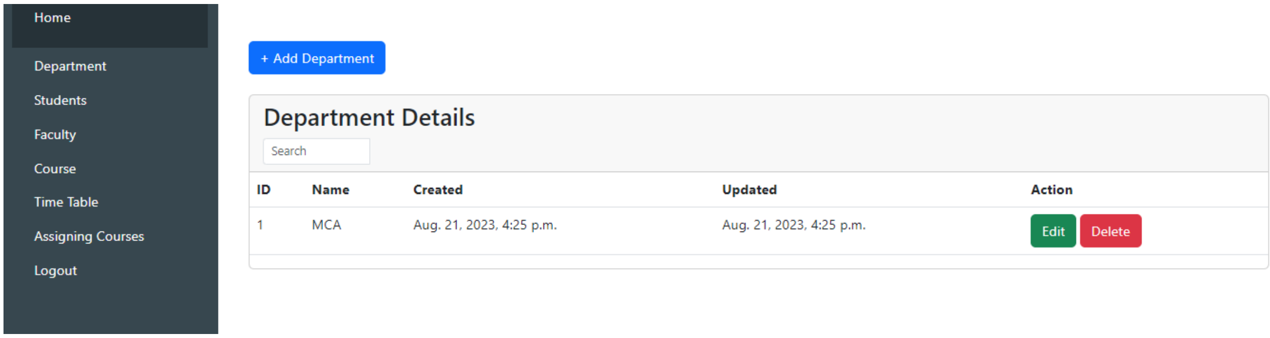
Department Registration

Department Name: MCA

Fig 3.4 Add Department

Fig 3.4 Add Department shows that the option to create departments is available, allowing administrators to organize and manage different academic divisions efficiently within the system.

iv. **Manage Department**: Admin can manipulate existing department information



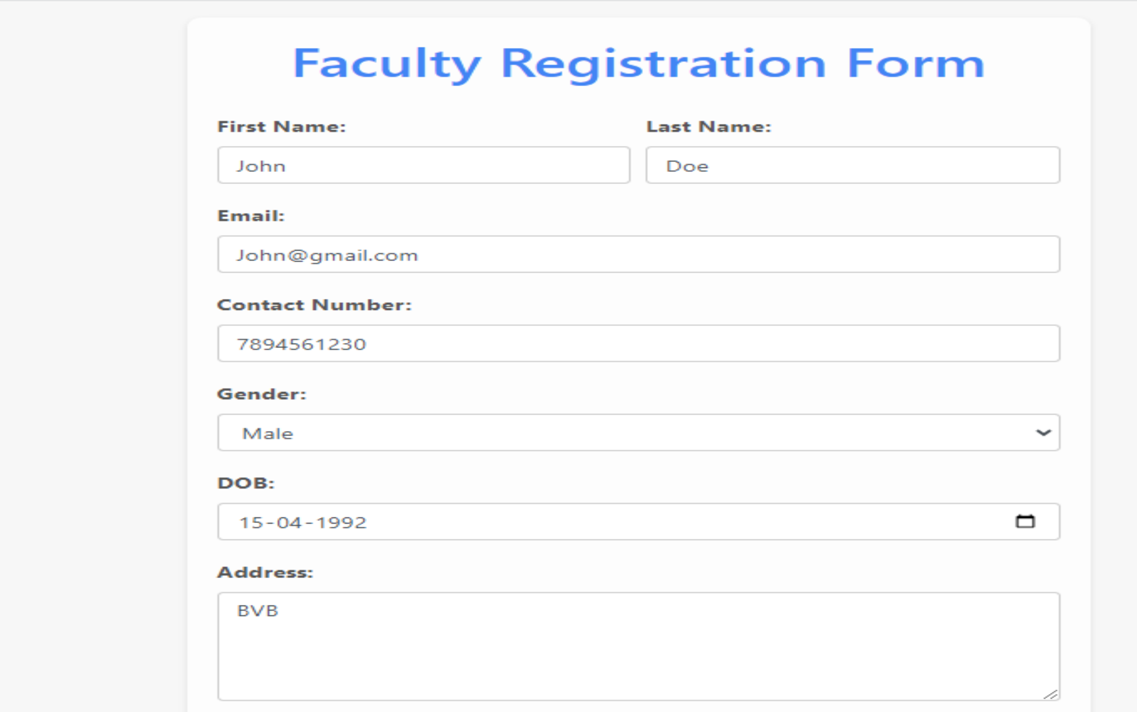
Department Details

ID	Name	Created	Updated	Action
1	MCA	Aug. 21, 2023, 4:25 p.m.	Aug. 21, 2023, 4:25 p.m.	Edit Delete

Fig 3.5 Manage Department

Fig 3.5 Manage Department shows that the system facilitates department management, allowing efficient organization and oversight of academic departments, roles, and responsibilities**.**

vi. Create Faculty: Admin can create Faculty



Faculty Registration Form

First Name: John

Last Name: Doe

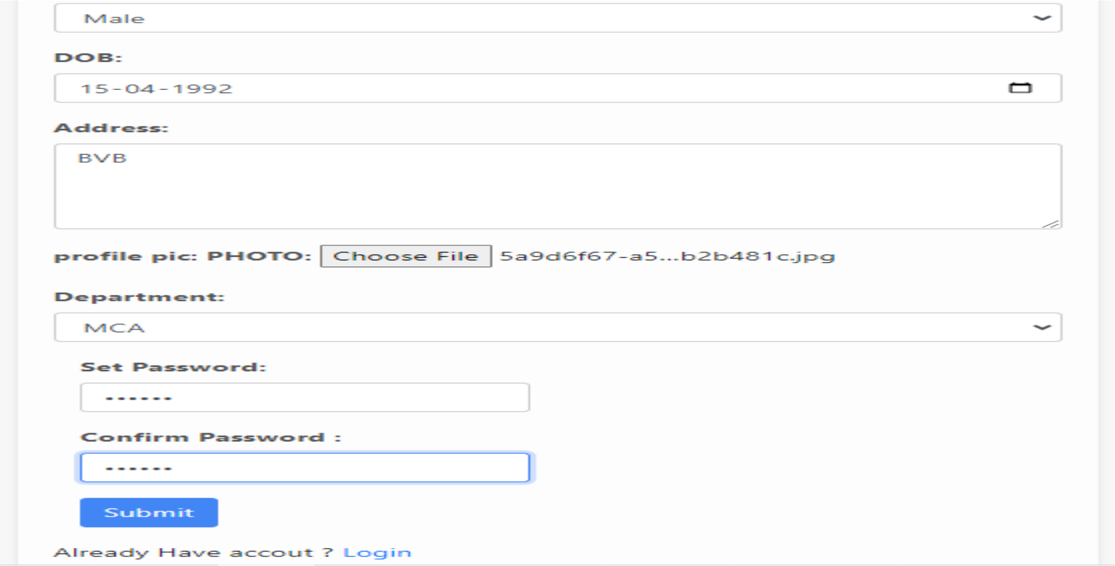
Email: John@gmail.com

Contact Number: 7894561230

Gender: Male

DOB: 15-04-1992

Address: BVB



Male

DOB: 15-04-1992

Address: BVB

profile pic: PHOTO: Choose File 5a9d6f67-a5...b2b481c.jpg

Department: MCA

Set Password: ······

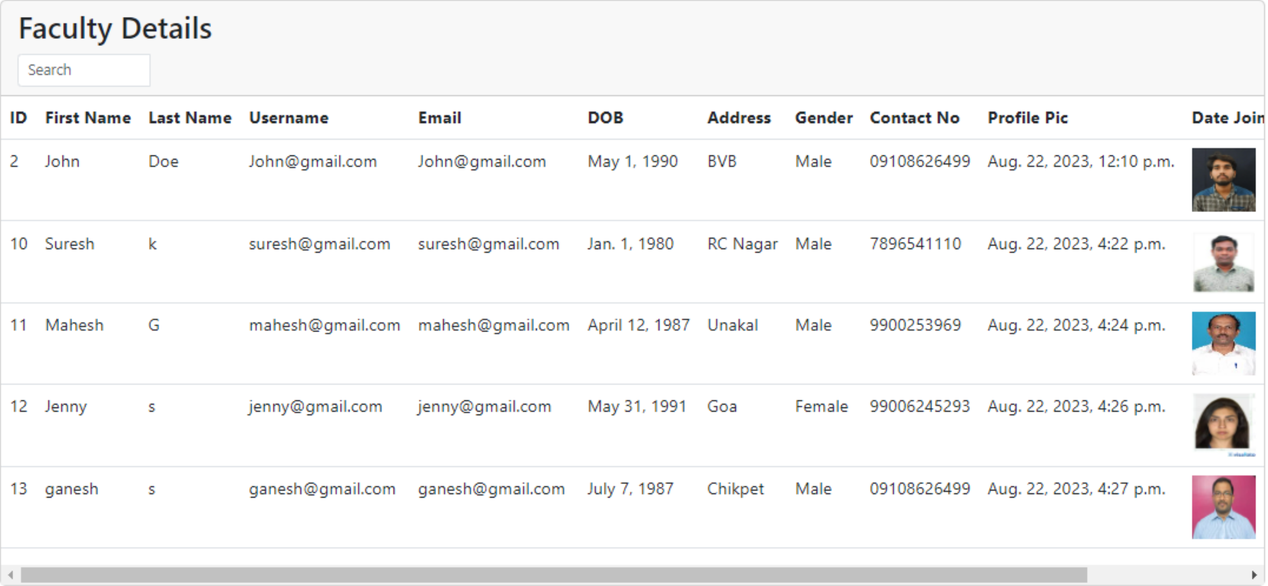
Confirm Password : ······

Already Have accout ? Login

Fig 3.6 Add Faculty

Fig 3.6 Add Faculty shows that the option to create faculty is available, allowing administrators to organize and manage different academic divisions efficiently within the system.

vii. Manage Faculty: Admin can manipulate existing faculty information



Faculty Details

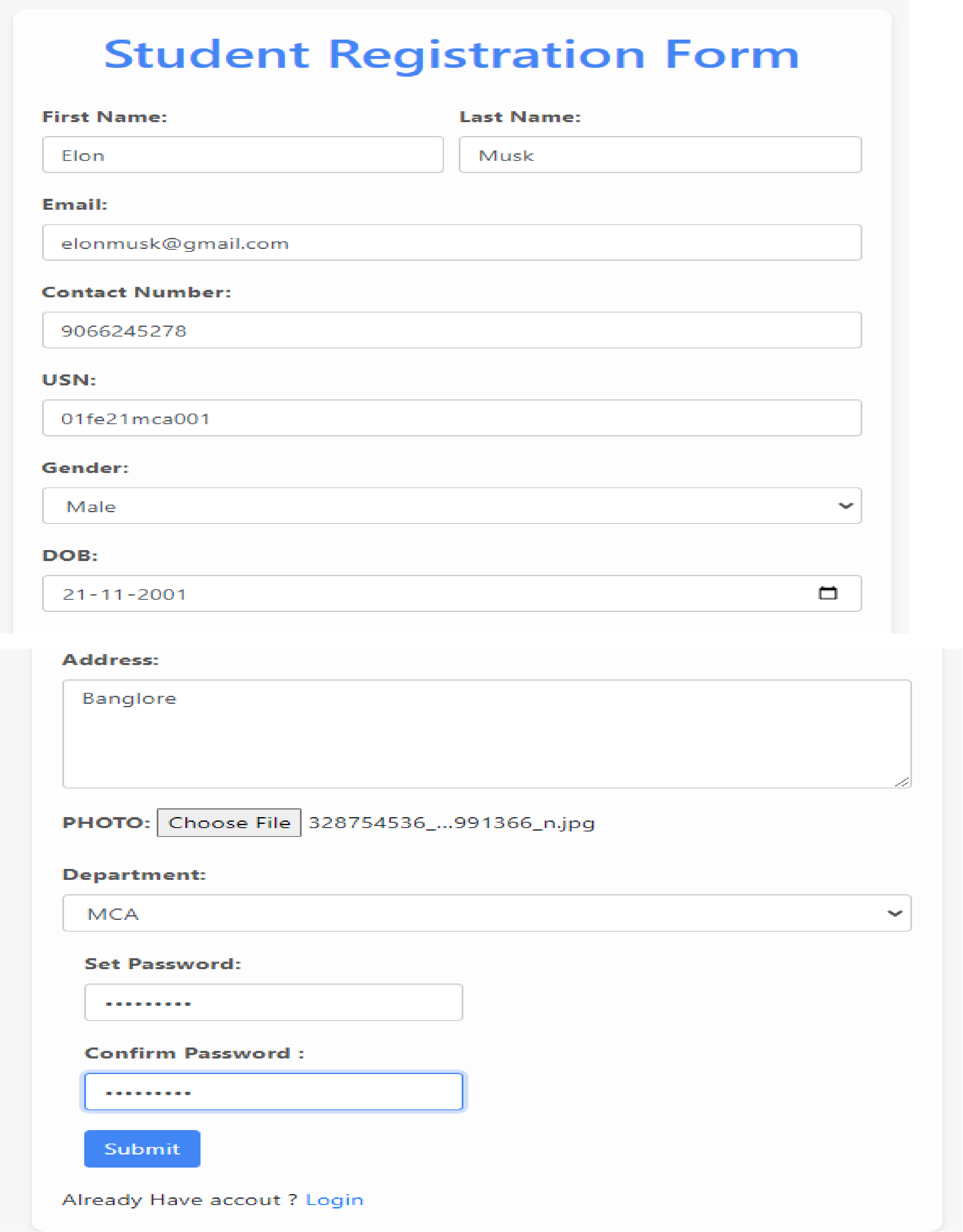
Search

ID	First Name	Last Name	Username	Email	DOB	Address	Gender	Contact No	Profile Pic	Date Join
2	John	Doe	John@gmail.com	John@gmail.com	May 1, 1990	BVB	Male	09108626499	Aug. 22, 2023, 12:10 p.m.	
10	Suresh	k	suresh@gmail.com	suresh@gmail.com	Jan. 1, 1980	RC Nagar	Male	7896541110	Aug. 22, 2023, 4:22 p.m.	
11	Mahesh	G	mahesh@gmail.com	mahesh@gmail.com	April 12, 1987	Unakal	Male	9900253969	Aug. 22, 2023, 4:24 p.m.	
12	Jenny	s	jenny@gmail.com	jenny@gmail.com	May 31, 1991	Goa	Female	99006245293	Aug. 22, 2023, 4:26 p.m.	
13	ganesh	s	ganesh@gmail.com	ganesh@gmail.com	July 7, 1987	Chikpet	Male	09108626499	Aug. 22, 2023, 4:27 p.m.	

Fig 3.7 Manage Faculty

Fig 3.7 Manage Faculty shows that the system facilitates faculty management, allowing efficient organization and oversight of academic departments, roles, and responsibilities**.**

viii. Create Student: Admin can create Student



Student Registration Form

First Name: Elon

Last Name: Musk

Email: elonmusk@gmail.com

Contact Number: 9066245278

USN: 01fe21mca001

Gender: Male

DOB: 21-11-2001

Address: Banglore

PHOTO: Choose File 328754536_...991366_n.jpg

Department: MCA

Set Password: ·········

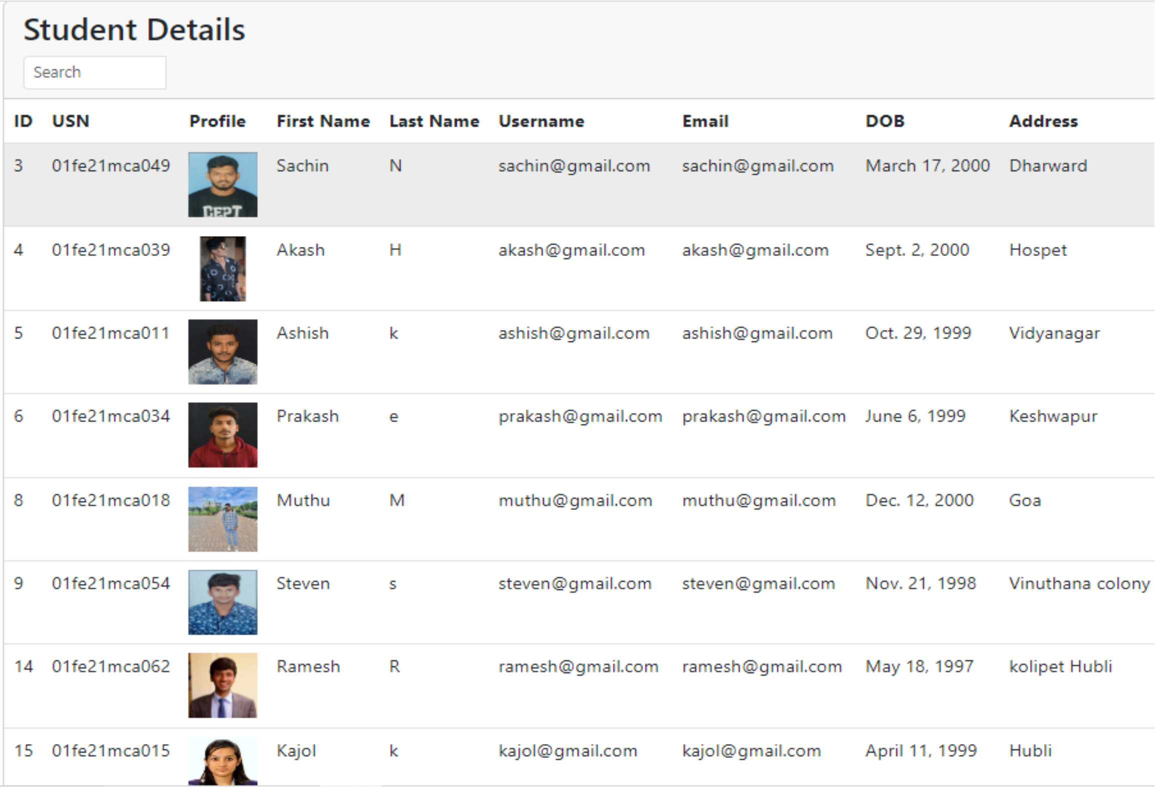
Confirm Password : ·········

Already Have accout ? Login

Fig 3.8 Add Student

Fig 3.8 Add Student shows that the option to create student is available, allowing administrators to organize and manage different academic divisions efficiently within the system.

ix. Manage Student: Admin can manipulate existing Student information



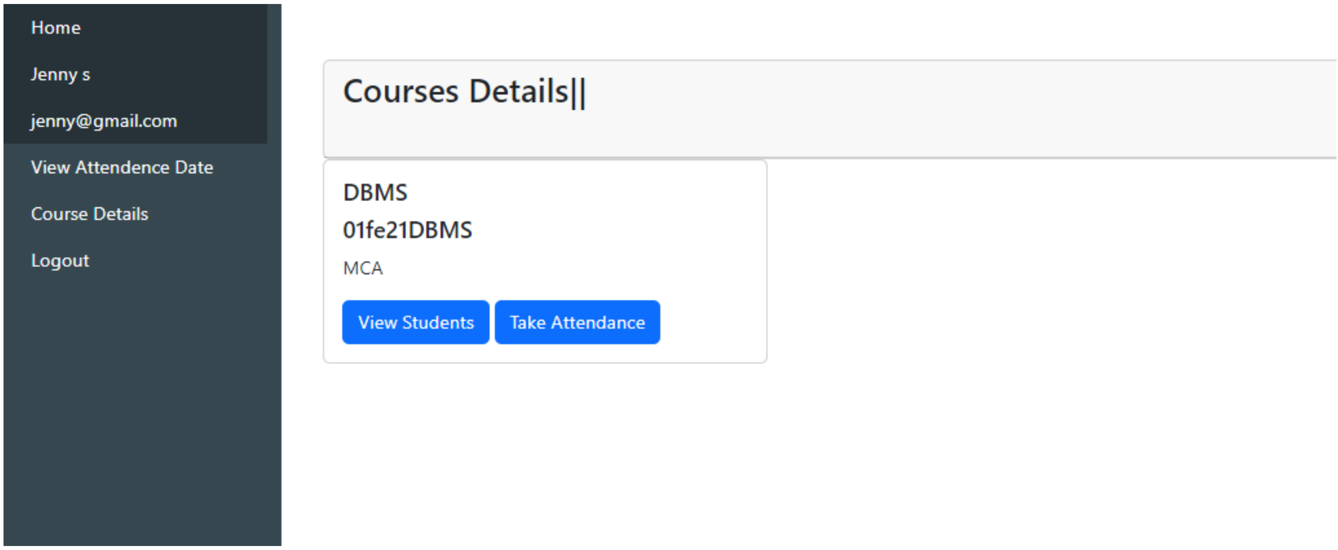
Student Details

ID	USN	Profile	First Name	Last Name	Username	Email	DOB	Address
3	01fe21mca049		Sachin	N	sachin@gmail.com	sachin@gmail.com	March 17, 2000	Dharward
4	01fe21mca039		Akash	H	akash@gmail.com	akash@gmail.com	Sept. 2, 2000	Hospet
5	01fe21mca011		Ashish	k	ashish@gmail.com	ashish@gmail.com	Oct. 29, 1999	Vidyanagar
6	01fe21mca034		Prakash	e	prakash@gmail.com	prakash@gmail.com	June 6, 1999	Keshwapur
8	01fe21mca018		Muthu	M	muthu@gmail.com	muthu@gmail.com	Dec. 12, 2000	Goa
9	01fe21mca054		Steven	s	steven@gmail.com	steven@gmail.com	Nov. 21, 1998	Vinuthana colony
14	01fe21mca062		Ramesh	R	ramesh@gmail.com	ramesh@gmail.com	May 18, 1997	kolipet Hubli
15	01fe21mca015		Kajol	k	kajol@gmail.com	kajol@gmail.com	April 11, 1999	Hubli

Fig 3.9 Manage Student

Fig 3.9 Manage Student shows that the system facilitates student management, allowing efficient organization and oversight of academic departments, roles, and responsibilities.

x. Staff Dashboard : Staff Dashboard



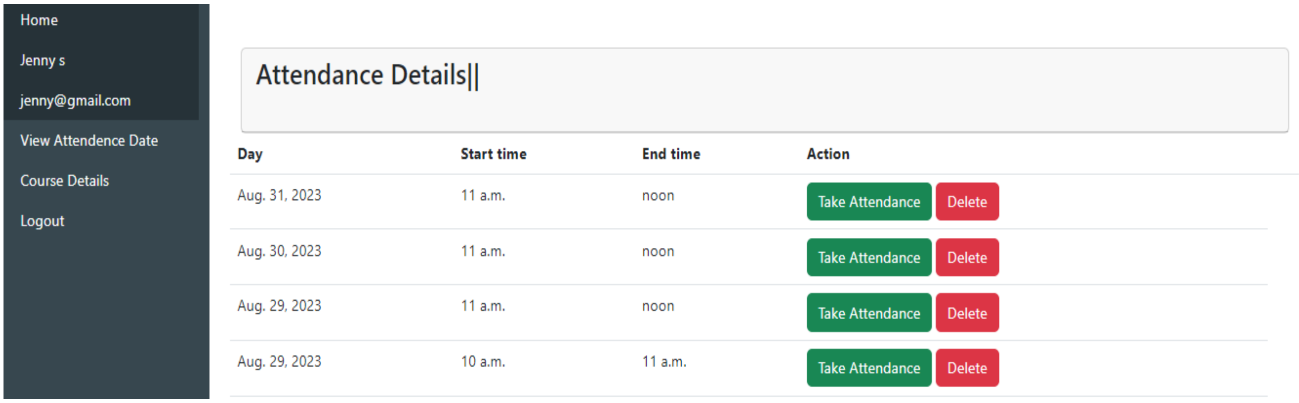
Courses Details||

DBMS
01fe21DBMS
MCA

Fig 3.10 Staff Dashboard

Fig 3.10 Staff Dashboard shows the staff dashboard offers a dedicated interface for staff members, providing quick access to attendance-related tasks, notifications, and reports.

xi. Attendance Details: Staff can Take Attendance as per allotted Date and Time



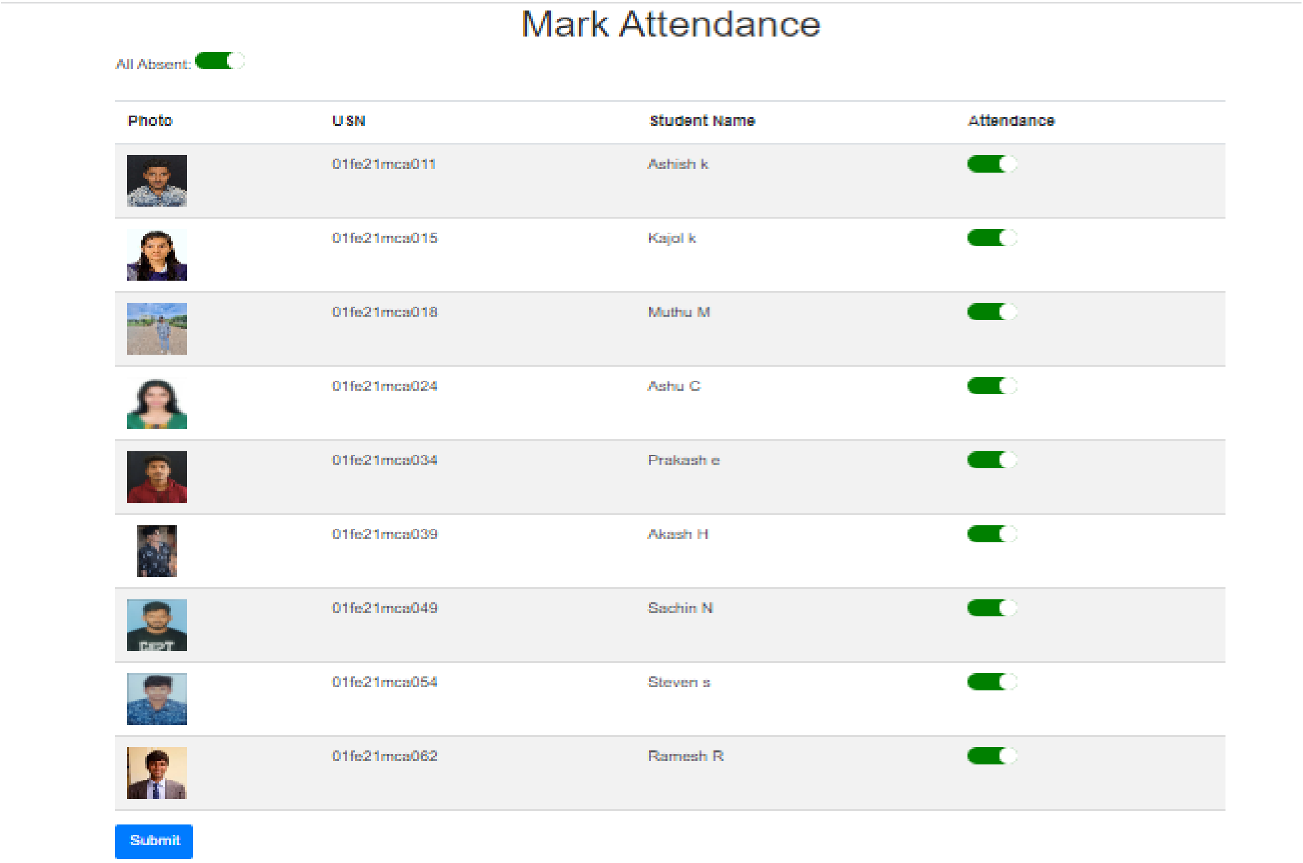
Attendance Details||

Day	Start time	End time	Action
Aug. 31, 2023	11 a.m.	noon	Take Attendance Delete
Aug. 30, 2023	11 a.m.	noon	Take Attendance Delete
Aug. 29, 2023	11 a.m.	noon	Take Attendance Delete
Aug. 29, 2023	10 a.m.	11 a.m.	Take Attendance Delete

Fig 3.11 Staff Attendance Details

Fig 3.11 Staff Attendance Details shows that the Staff members can easily mark attendance within specified schedules, ensuring accurate tracking of student presence based on allocated dates and times.

xii. Take Attendance: Staff can Take Attendance



Mark Attendance

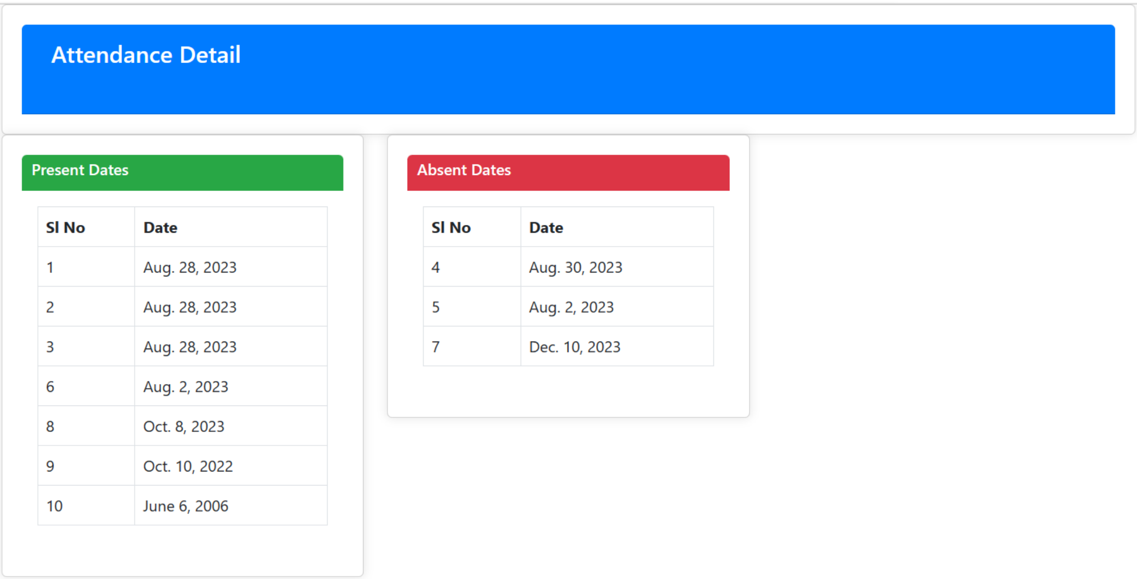
All Absent: ●

Photo	USN	Student Name	Attendance
	01fe21mca011	Ashish k	●
	01fe21mca015	Kajol k	●
	01fe21mca018	Muthu M	●
	01fe21mca024	Ashu C	●
	01fe21mca034	Prakash e	●
	01fe21mca039	Akash H	●
	01fe21mca049	Sachin N	●
	01fe21mca054	Steven s	●
	01fe21mca062	Ramesh R	●

Fig 3.12 Mark Attendance

Fig 3.12 Mark Attendance shows that the Staff members have the capability to record attendance, providing a convenient and straightforward method for tracking student participation.

xiii. Student Attendance Detail: Student can Check their Attendance



Attendance Detail

Present Dates

Sl No	Date
1	Aug. 28, 2023
2	Aug. 28, 2023
3	Aug. 28, 2023
6	Aug. 2, 2023
8	Oct. 8, 2023
9	Oct. 10, 2022
10	June 6, 2006

Absent Dates

Sl No	Date
4	Aug. 30, 2023
5	Aug. 2, 2023
7	Dec. 10, 2023

Fig 3.13 Student Attendance Detail

Fig 3.13 Student Attendance Detail shows that the Students can easily view their attendance details, providing them with quick access to their attendance records and promoting personal accountability.

CHAPTER 4

SYSTEM DESIGN

The Student Attendance Management System's design is a well-structured arrangement that brings together different parts for smooth attendance tracking and handling. It includes a smart database setup for storing attendance and user details, and an easy interface for students, teachers, and others to mark attendance and view reports. Security measures are built in for data safety, and the design is ready to grow as more people use it. This design ensures the system works well, is user-friendly, and keeps information secure.

4.1 Architecture of the system

The architecture of the Student Attendance Management System (SAMS) is designed to ensure efficient data flow, modularity, and scalability. Student Attendance Management System follows a three-tier architecture, separating the application into presentation, logic, and data layers.

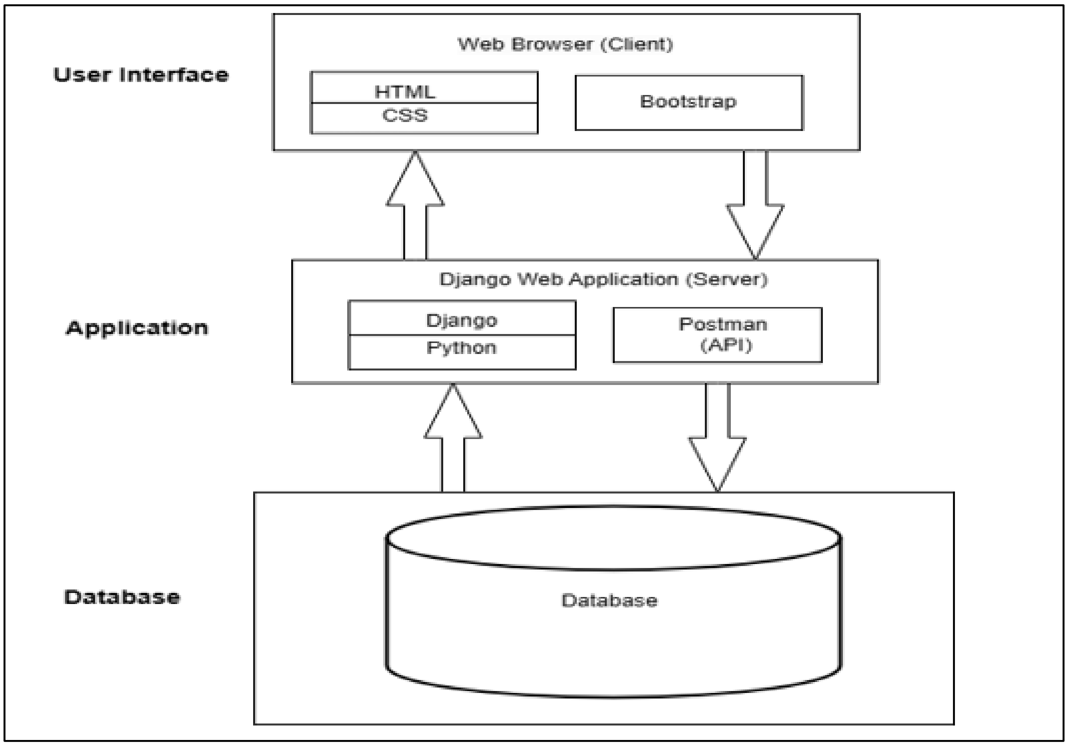


Fig 4.1 Architecture Diagram

4.2 Level 0 DFD :

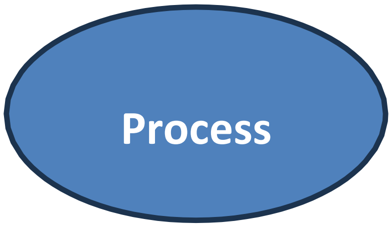
A Data Flow Diagram (DFD) is a graphical representation that describes how data flows through a system. It illustrates the processes, data sources, data destinations, data stores, and the flow of data between them. DFDs are used to visualize the flow of information within a system, making it easier to understand the interactions between different components.

Level 0 DFD, also known as Context Diagram, is the highest-level view of a system's data flow. It provides an overview of the entire system and shows how external entities interact with the system. Level 0 DFD represents the boundaries of the system, without delving into the internal processes or details. The four essential building blocks of a data flow diagrams are :

- Process
- Data flow
- Store
- Terminator

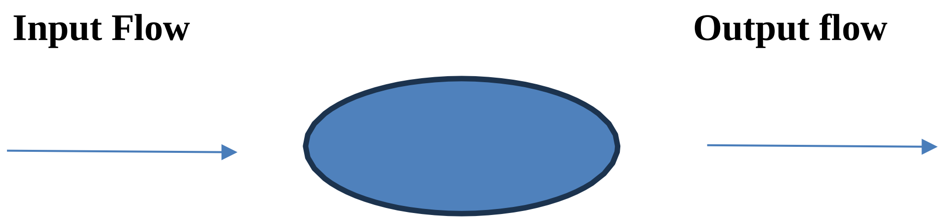
The Process:

A process represents a specific function, action, or transformation that occurs within the system. It takes input data, performs some processing, and produces output data. Processes in DFDs are typically labeled with a verb-noun phrase that describes the action being performed.



Data Flow:

Data flows are arrows that show the movement of data between processes, data stores, and external entities. They represent the path that data takes as it moves through the system. Data flows are labeled toindicate the type of data being transferred, providing clarity on the information being exchanged.



- **Two data packers**

4.2 Context Level DFD

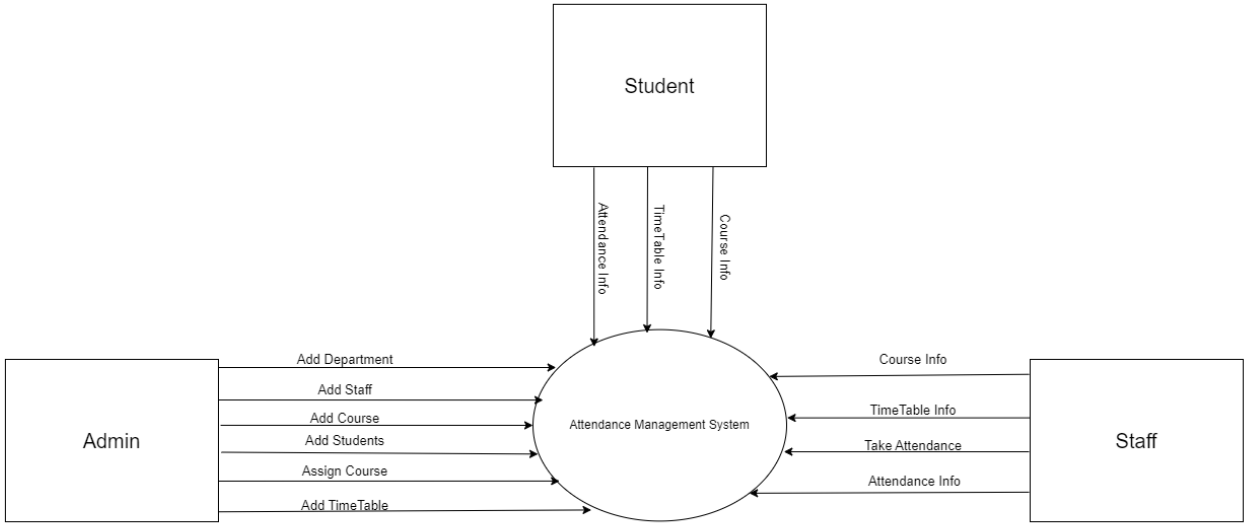


Fig 4.2 Context Level DFD Diagram

Fig 4.2 Context Level DFD Diagram shows that the Context Level DFD Diagram offers a top-level view of the Student Attendance Management System, showcasing interactions with users and external elements, providing a clear boundary and data flow depiction.

4.3 Detailed DFD for the proposed system:

A Detailed Data Flow Diagram (DFD) is like a detailed map that shows exactly how information moves within a system. It breaks down each step, showing where data comes from, where it goes, and how it changes along the way. This helps people understand exactly what's happening inside the system and is useful for designing and improving how things work. It helps analysts and developers understand system intricacies, making it valuable for in-depth system analysis and design

Level 0 DFD Diagram

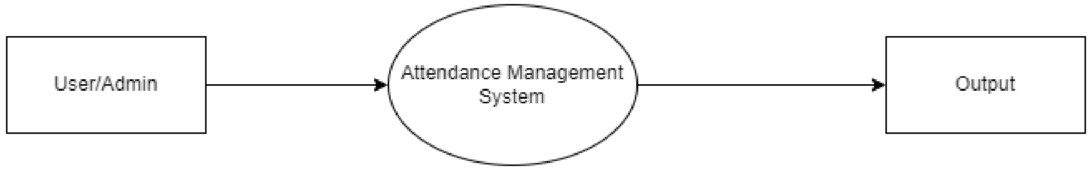


Fig 4.3 Level 0 DFD Diagram

Fig 4.3 illustrates a Level 0 Data Flow Diagram (DFD) for the Student Attendance Management System, showcasing the high-level overview of data interactions between external entities and the core system processes.

Level 1 DFD Diagram

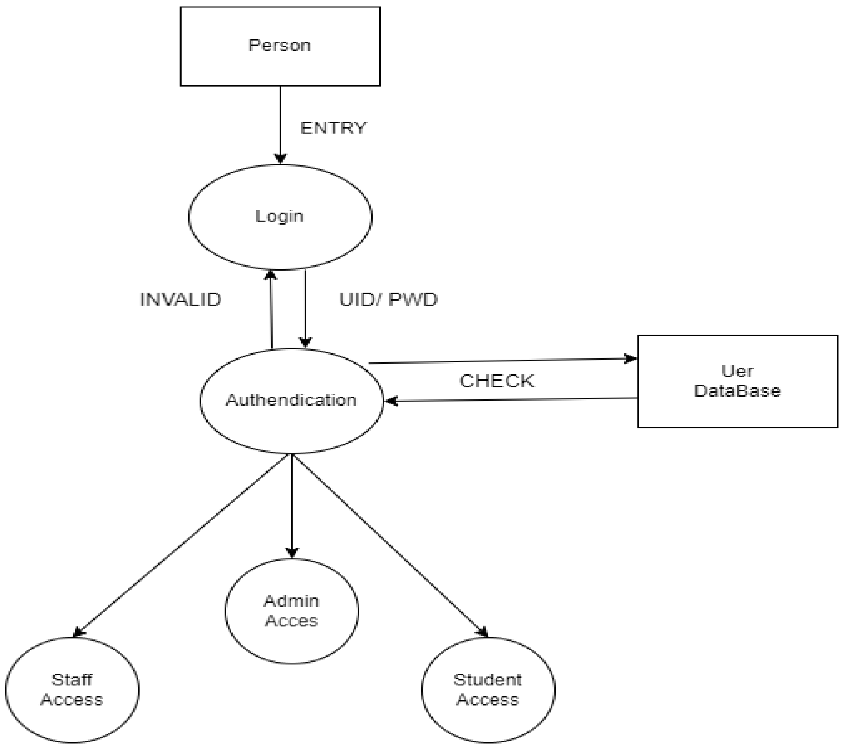


Fig 4.4 Level 1 DFD Diagram

Fig 4.4 presents a Level 1 Data Flow Diagram illustrating the flow of data and processes within the Student Attendance Management System, offering a visual representation of its core functionalities and interactions.

Level 2 DFD Diagram

ADMIN MODULE:

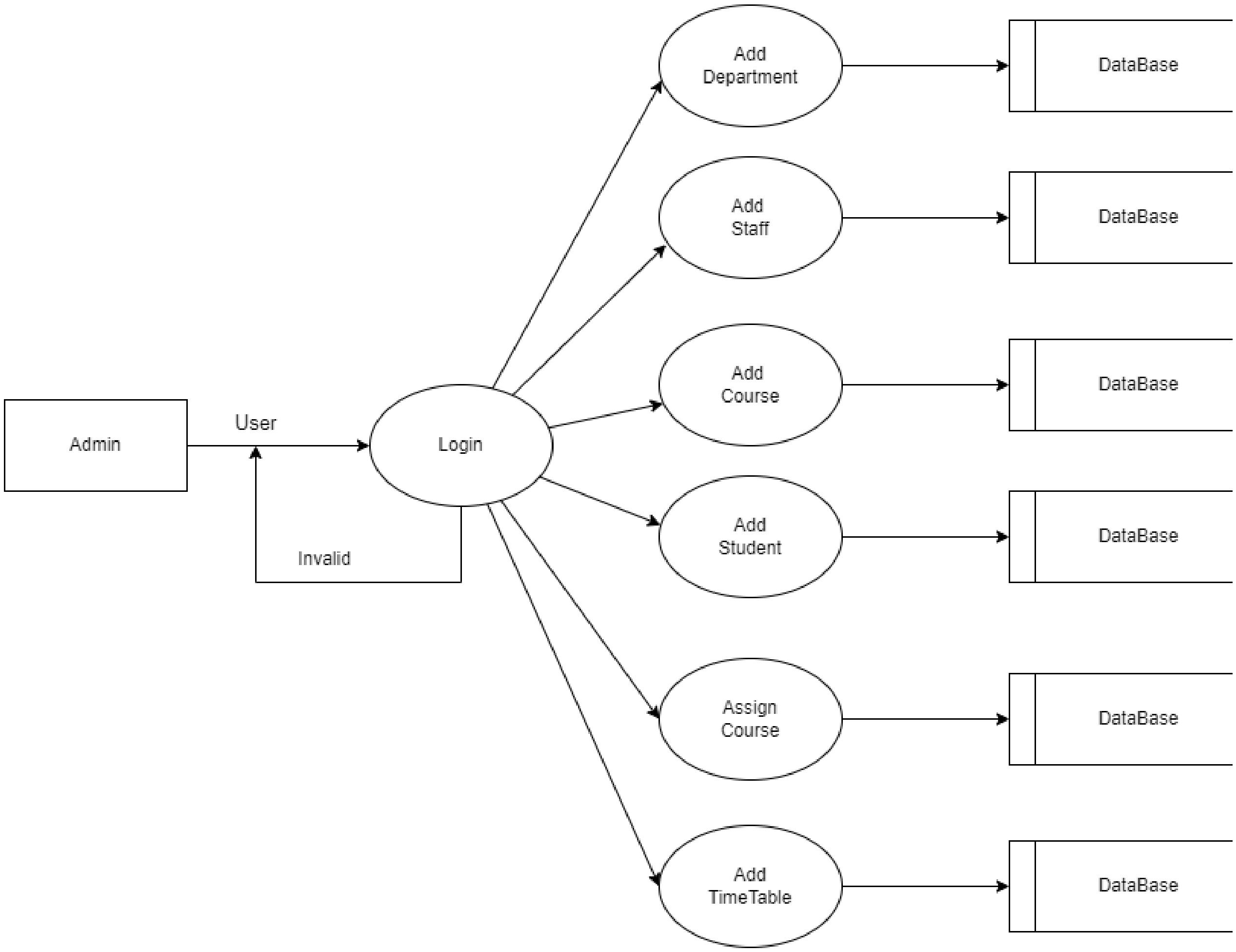


Fig 4.5 Admin Module Diagram

Fig 4.5 illustrates the structure and components of the Admin Module in the Student Attendance Management System, providing a visual overview of its administrative functionalities and interactions.

STAFF MODULE:

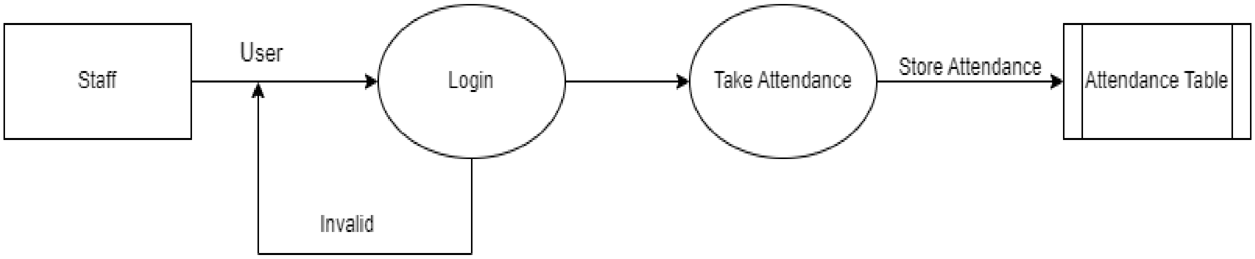


Fig 4.6 Staff Module Diagram

Figure 4.6 the Staff Module Diagram visually represents the features and interactions within the staff component of the Student Attendance Management System, offering a clear overview of staff-related functionalities and processes.

STUDENT MODULE:

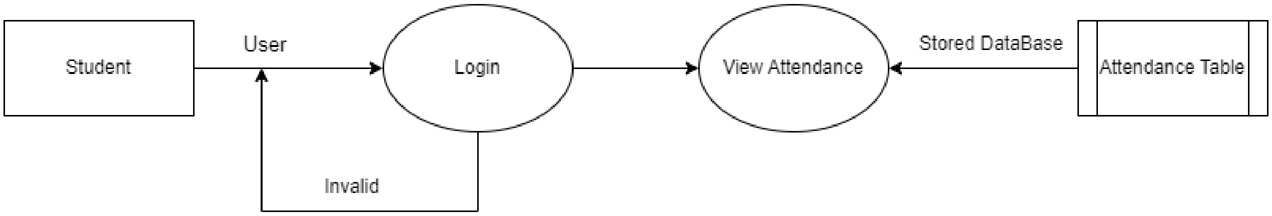


Fig 4.7 Student Module Diagram

Fig 4.7 depicts the diagram of the Student Module within the Student Attendance Management System, illustrating its features and interactions for efficient attendance tracking and engagement.

4.4 Class Diagram:

A Class Diagram is a type of Unified Modeling Language (UML) diagram used in software engineering to visualize the structure of a system in terms of classes, attributes, methods, and their relationships. It provides a high-level overview of the classes in a system, their properties, and the interactions between them.

In a Class Diagram:

- Class: Represents a blueprint for creating objects. It includes attributes (data) and methods (functions) that define the behavior of the objects.
- Attributes: Describe the properties or characteristics of a class, such as variables that store data.
- Methods: Represent the functions or operations that can be performed by the class.

  Relationships: Depict how classes are related to each other, including associations, aggregations, and inheritances

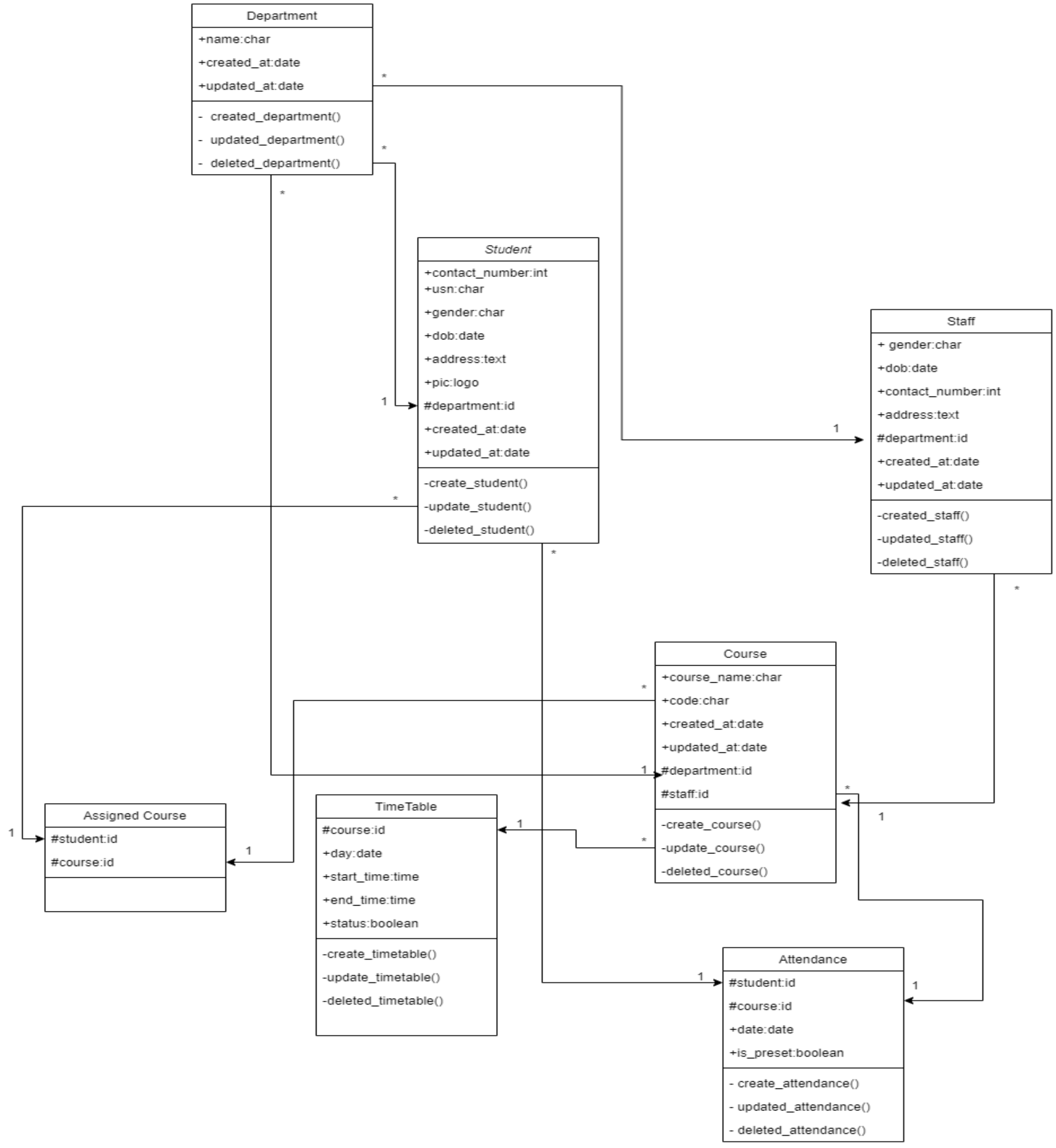


Fig 4.8 Class Diagram

Figure 4.8 illustrates the Class Diagram, visually depicting the relationships and interactions between different classes and objects within the Student Attendance Management System, enhancing our understanding of its structural design.

4.5 Sequence diagram:

A Sequence Diagram is a type of Unified Modeling Language (UML) diagram used in software engineering to visualize the interactions and order of messages exchanged between various objectsor components within a system over a specific period of time. It illustrates the dynamic behavior of the system by showing the sequence of actions and their temporal relationships.

In a Sequence Diagram:

- Objects: Represent the instances of classes or components involved in the interaction.
- Lifelines: Represent the duration of existence of an object during the interaction.
- Messages: Depict the communication or interaction between objects.

Activation Bars: Show the duration of time during which an object is actively processing a message

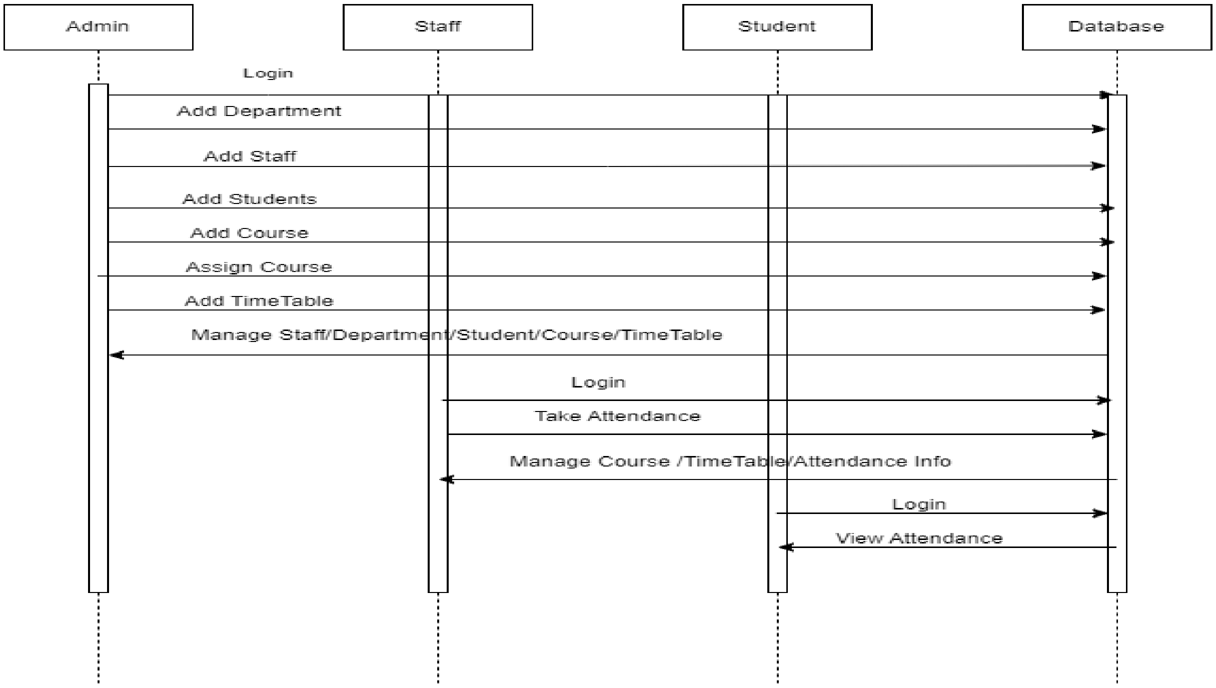


Fig 4.9 Sequence diagram

In Fig 4.9 sequence diagram of the Student Attendance Management System, the interactions and order of actions between various system components are visually represented, aiding in understanding the flow of attendance-related processes.

4.6 ER diagram and schema:

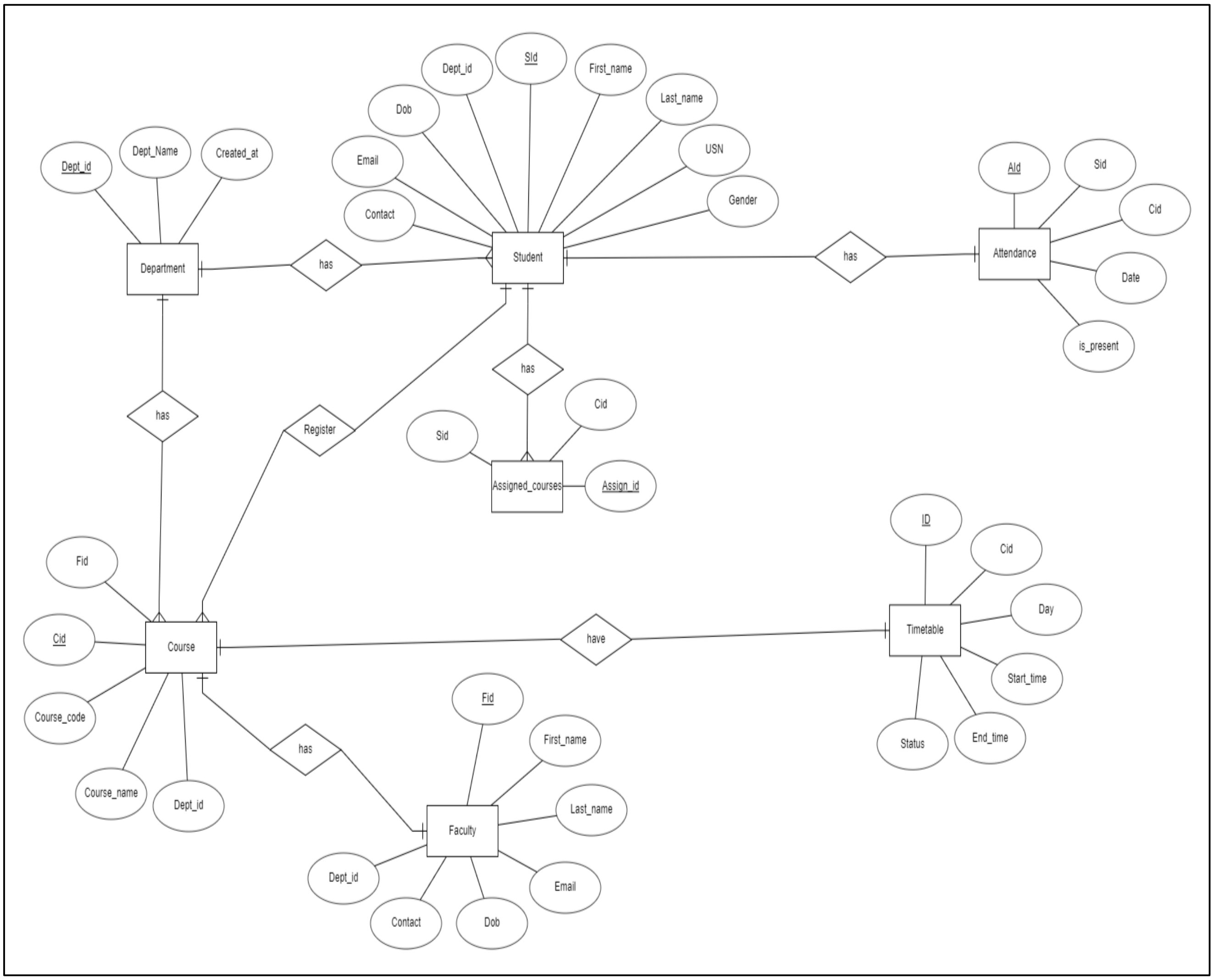


Fig 4.10 ER Diagram

Fig 4.10 presents an Entity-Relationship (ER) Diagram illustrating the data structure and relationships within the Student Attendance Management System, providing a visual representation of its database design and functionality.

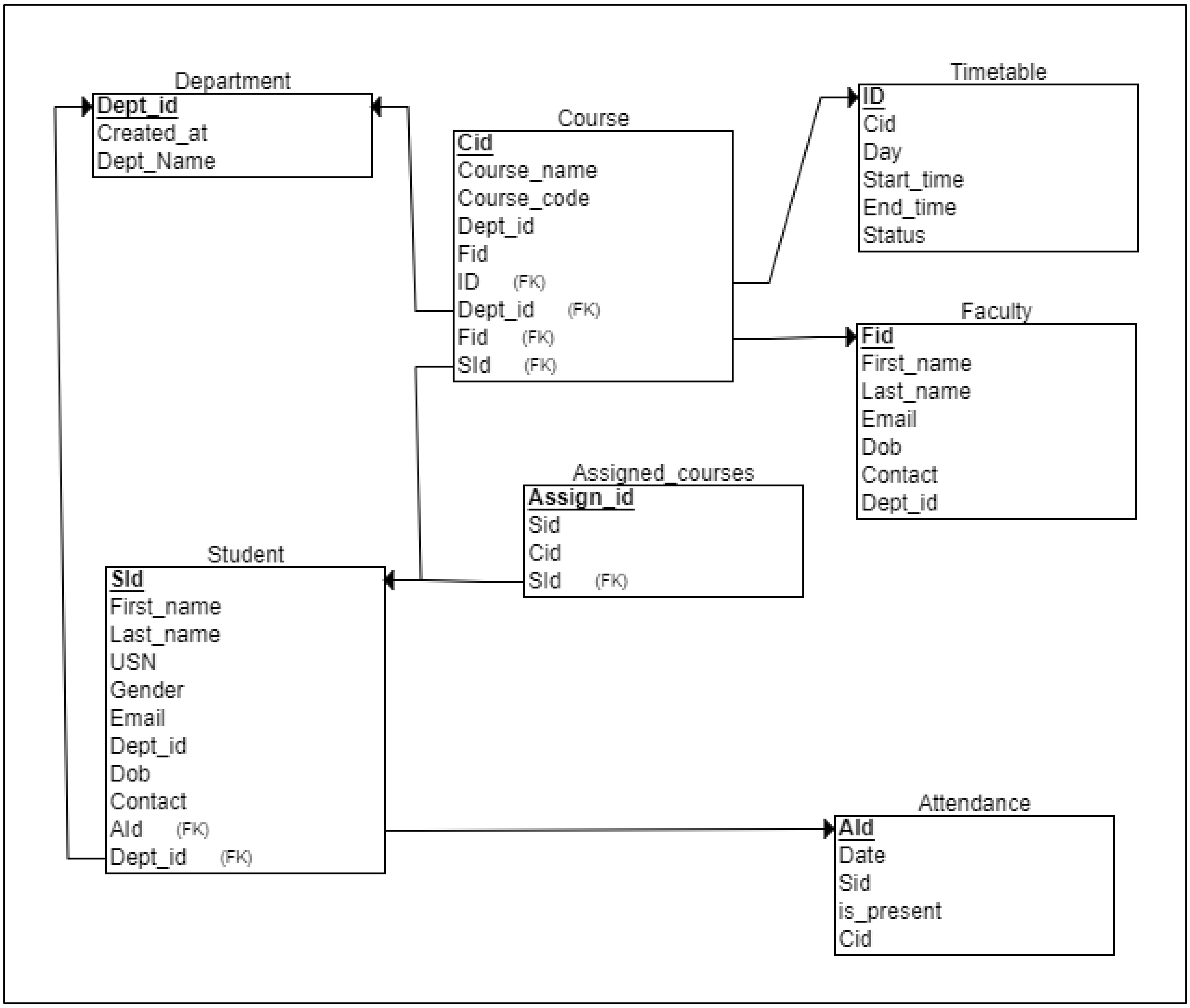
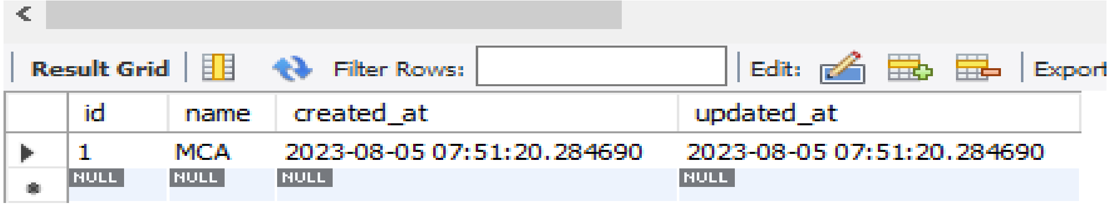


Fig 4.11 Schema diagram

Fig 4.11 presents a visual schema diagram illustrating the structural organization and relationships within the Student Attendance Management System, offering insights into its database design and component interactions.

4.7 DATA STRUCTURE USED

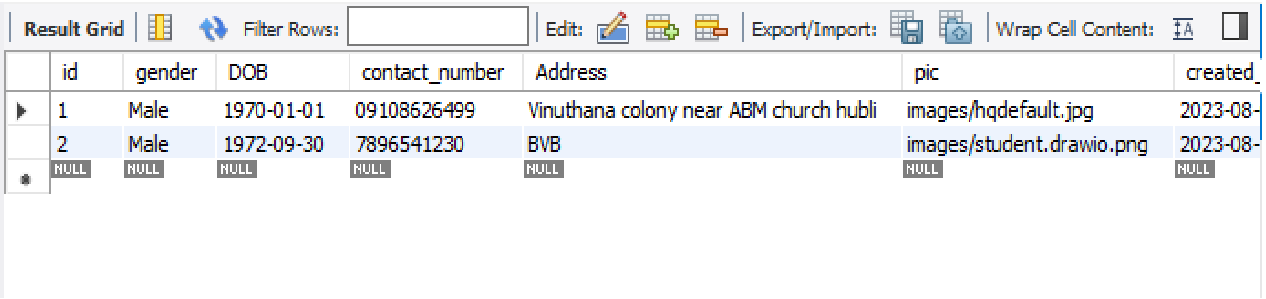
4.7.1 select * from ams_dept;



id	name	created_at	updated_at
1	MCA	2023-08-05 07:51:20.284690	2023-08-05 07:51:20.284690
NULL	NULL	NULL	NULL

Fig 4.7.1 shows that In the database, the query "select * from ams_dept" retrieves all records from the "ams_dept" table within the Attendance Management System, facilitating efficient data retrieval and management of department-related information.

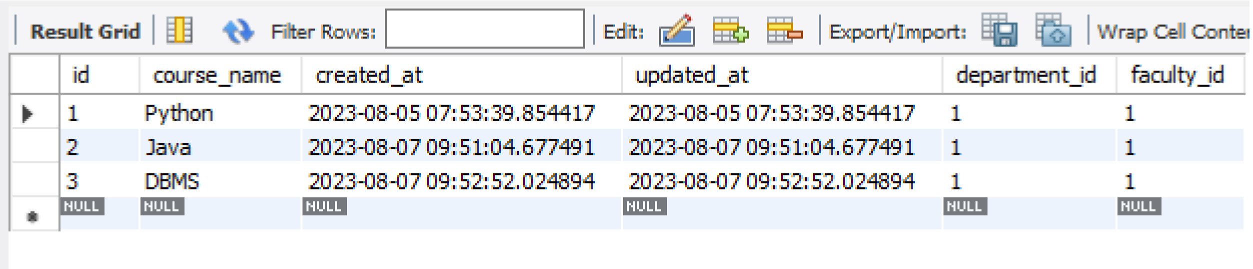
4.7.2 select * from ams_faculty;



id	gender	DOB	contact_number	Address	pic	created_
1	Male	1970-01-01	09108626499	Vinuthana colony near ABM church hubli	images/hqdefault.jpg	2023-08-
2	Male	1972-09-30	7896541230	BVB	images/student.drawio.png	2023-08-
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Fig 4.7.2 shows that In the database, the query "select * from ams_faculty" retrieves all records from the " ams_faculty " table within the Attendance Management System, facilitating efficient data retrieval and management of faculty-related information.

4.7.3 select * from ams_course;

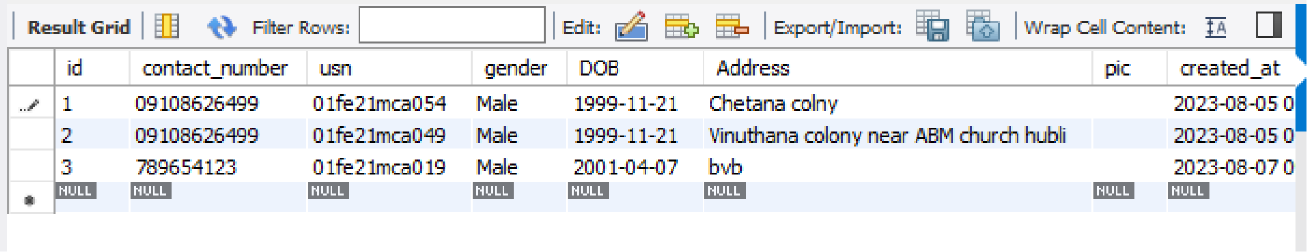


id	course_name	created_at	updated_at	department_id	faculty_id
1	Python	2023-08-05 07:53:39.854417	2023-08-05 07:53:39.854417	1	1
2	Java	2023-08-07 09:51:04.677491	2023-08-07 09:51:04.677491	1	1
3	DBMS	2023-08-07 09:52:52.024894	2023-08-07 09:52:52.024894	1	1
NULL	NULL	NULL	NULL	NULL	NULL

Fig 4.7.3 shows that In the database, the query "select * from ams_course" retrieves all records from the " ams_course " table within the Attendance Management System, facilitating efficient data

retrieval and management of course-related information.

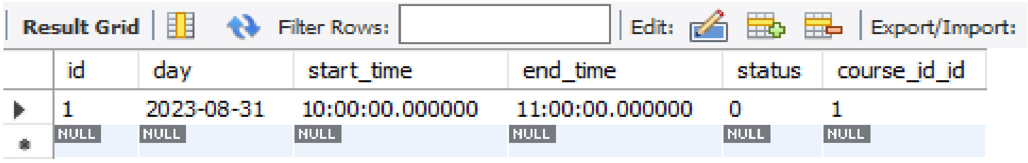
4.7.4 select * from ams_student;



id	contact_number	usn	gender	DOB	Address	pic	created_at
1	09108626499	01fe21mca054	Male	1999-11-21	Chetana colny		2023-08-05 0
2	09108626499	01fe21mca049	Male	1999-11-21	Vinuthana colony near ABM church hubli		2023-08-05 0
3	789654123	01fe21mca019	Male	2001-04-07	bvb		2023-08-07 0
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Fig 4.7.4 shows that In the database, the query "select * from ams_student" retrieves all records from the " ams_student " table within the Attendance Management System, facilitating efficient data retrieval and management of student-related information

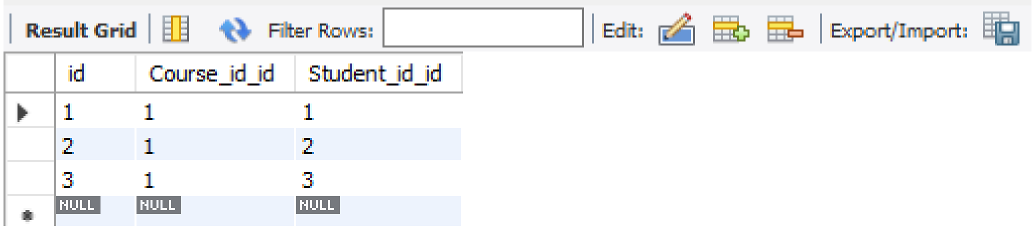
4.7.5 select * from ams_timetable;



id	day	start_time	end_time	status	course_id_id
1	2023-08-31	10:00:00.000000	11:00:00.000000	0	1
NULL	NULL	NULL	NULL	NULL	NULL

Fig 4.7.4 shows that In the database, the query "select * from ams_timetable" retrieves all records from the " ams_timetable " table within the Attendance Management System, facilitating efficient data retrieval and management of timetable-related information

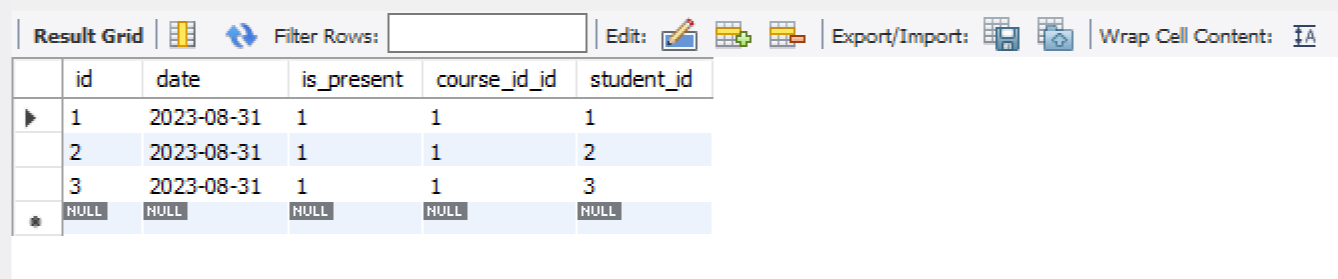
4.7.6 select * from ams_assigned_courses;



id	Course_id_id	Student_id_id
1	1	1
2	1	2
3	1	3
NULL	NULL	NULL

Fig 4.7.4 shows that In the database, the query "select * from ams_assigned_courses" retrieves all records from the " ams_assigned_courses " table within the Attendance Management System, facilitating efficient data retrieval and management of assignedcourse-related information

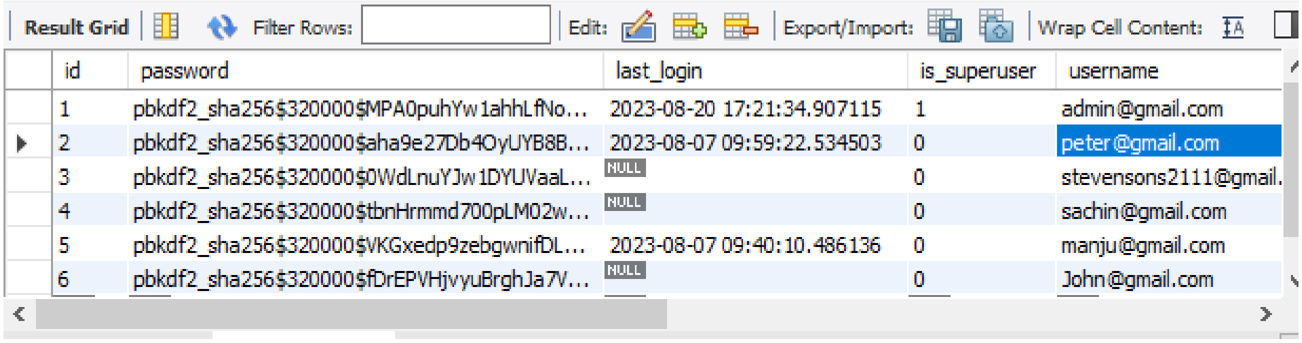
4.7.7 select * from ams_attendance;;



	id	date	is_present	course_id_id	student_id
▶	1	2023-08-31	1	1	1
	2	2023-08-31	1	1	2
	3	2023-08-31	1	1	3
*	NULL	NULL	NULL	NULL	NULL

Fig 4.7.4 shows that In the database, the query "select * from ams_attendance" retrieves all records from the " ams_attendance " table within the Attendance Management System, facilitating efficient data retrieval and management of attendancw-related information

4.7.8 select * from auth_user;



	id	password	last_login	is_superuser	username
	1	pbkdf2_sha256$320000$MPA0puhYw1ahhLfNo...	2023-08-20 17:21:34.907115	1	admin@gmail.com
▶	2	pbkdf2_sha256$320000$aha9e27Db4OyUYB8B...	2023-08-07 09:59:22.534503	0	peter@gmail.com
	3	pbkdf2_sha256$320000$0WdLnuYJw1DYUVaaL...	NULL	0	stevensons2111@gmail.
	4	pbkdf2_sha256$320000$tbnHrmmd700pLM02w...	NULL	0	sachin@gmail.com
	5	pbkdf2_sha256$320000$VKGxedp9zebgwnifDL...	2023-08-07 09:40:10.486136	0	manju@gmail.com
	6	pbkdf2_sha256$320000$fDrEPVHjvyuBrghJa7V...	NULL	0	John@gmail.com

Fig 4.7.4 shows that In the database, the query "select * from auth_user" retrieves all records from the " auth_user " table within the Attendance Management System, facilitating efficient data retrieval and management of auth_user-related information

CHAPTER 5

IMPLEMENTATION

The implemented Student Attendance Management System revolutionizes attendance tracking in education. It automates attendance recording and reporting, user-friendly for all stakeholders. It combines manual and automated methods, providing insightful reports for decision-making. The system enhances communication through timely notifications. As it evolves, predictive analytics may bring proactive interventions. Overall, it streamlines operations, boosts engagement, and enriches education.

5.1 Proposed Methodology

To overcome the drawbacks of the existing system, the proposed system has been evolved. This project aims to reduce the paper work and saving time to generate accurate results from the student's attendance. The system provides with the best user interface. The efficient reports can be generated by using this proposed system.

Advantages of Proposed Methodology:

- It is trouble-free to use.
- It is a relatively fast approach to enter attendance
- Is highly reliable, approximate result from user
- Best user Interface
- Efficient reports

Feasibility Study:

Feasibility analysis begins once the goals are defined. It starts by generating broad possible solutions, which are possible to give an indication of what the new system should look lime. This is where creativity and imagination are used. Analysts must think up new ways of doing things- generate new ideas. There is no need to go into the detailed system operation yet. The solution should provide enough information to make reasonable estimates about project cost and give users an indication of how the new system will fit into the organization. It is important not to exert considerable effort at this stage only to find out that the project is not worthwhile or that there is a need significantly change the original goal.

Feasibility of a new system means ensuring that the new system, which we are going to implement, is efficient and affordable. There are various types of feasibility to be determined. They are,

- **Economically Feasible:**

Development of this application is highly economically feasible. The only thing to be done is making an environment with an effective supervision. It is cost effective in the sense that has eliminated the paper work completely. The system is also time effective because the calculations are automated which are made at the end of the month or as per the user requirement.

- **Technical Feasibility:**

The technical requirement for the system is economic and it does not use any other additional Hardware and software. Technical evaluation must also assess whether the existing systems can be upgraded to use the new technology and whether the organization has the expertise to use it. Install all upgrades framework into the .Net package supported widows based application. this application depends on Microsoft office and intranet service ,database. Enter their attendance and generate report to excel sheet

- **Operational Feasibility:**

The system working is quite easy to use and learn due to its simple but attractive interface. User requires no special training for operating the system. Technical performance include issues such as determining whether the system can provide the right information for the Department personnel student details, and whether the system can be organized so that it always delivers this information at the right place and on time using intranet services. Acceptance revolves around the current system and its personnel.

5.2 Description of Modules:

The system should be designed in such a way that only authorized people should be allowed to access some particular modules. The records should be modified by only administrators and no one else. The user should always be in control of the application and not the vice versa. The user interface should be consistent so that the user can handle the application with ease and speed. The application should be visually, conceptually clear.

1. Admin
2. Staff
3. Student

5.2.1 Admin :Roles and Activities.

- He/she is a super user who enjoys all the privileges .
- Add Update Delete Subject to be studied.
- Approve/Disapprove teacher/student account.
- Add Staff/student account .
- View attendance of each student.
- View attendance average Percentage
- Can send instruction/notification to every teacher.
- Update Records

5.2.2 Staff :Roles and Activities.

- He/she is required to create account and then gets approved by the admin.
- Staff can Manage Multiple lectures multiple students Attendance.
- View the list of students assigned to them.
- Mark attendance after each lecture
- View attendance of each student assigned to them.
- Generate/Download Report of attendance in PDF Format
- Check Average Attendance
- Update Profile
- Upload Profile Pic
- Update Password(encrypted format)

5.2.3 Student : Roles and Activities.

- He/she needs to create an account.
- Students can register himself/herself for Multiple lectures.
- Students can Check Attendance for Multiple lectures.
- Can view their attendance from anywhere around the globe.
- Check Average Attendance
- Update Profile
- Upload Profile Pic
- Update Password(encrypted format)

CODE

Code Snippet for Attendance

```
<div class="container">
    {%  for tt in tt_record %}
    {{tt_record }}
    {% endfor %}
    <h1>Mark Attendance</h1>
    <form method="POST" action="">
        {% csrf_t
        <table class="table table-striped table-toggle">
            <thead>
                <tr>
                    <th>Sl No</th>
                    <th>Photo</th>
                    <th>USN</th>
                    <th>Student Name</th>
                    <th>Attendance</th>
                </tr>
            </thead>
            <div class="form-group">
        <label for="toggle-all" class="toggle-label">All Absent:</label>
        <label class="toggle-switch">
            <input type="checkbox" id="toggle-all">
            <span class="slider"></span>
        </label>
    </div>
            <tbody>
            {% regroup assigned_students by Student_id.usn as student_list %}
            {% for students in student_list|dictsort:"grouper" %}
            {% for s in students.list %}
                <tr>
                    <td>{{ forloop.counter }}</td>
                       <td>
                        {% if s.Student_id.pic %}
                           <img src="{{s.Student_id.pic.url}}" alt="image"  width="60px" height="60px">
                            {% else %}
                             No Image Available
                             {% endif %}
                          </td>
                    <td>{{ s.Student_id.usn }}</td>
                    <td>{{ s.Student_id.admin.first_name }} {{ s.Student_id.admin.last_name }}</td>
                    <td>
                        <label class="toggle-switch">
                            <input type="checkbox" name="{{ s.id }}" checked data-off="Present" data-
on="Absent">
                            <span class="slider"></span>
                        </label>
                    </td>
                </tr>
                {% endfor %}
                {% endfor %}
```

Views.py

```
class Mark_Attendance_View(View):
    def get(self, request, id,day):
        assigned_students = Assigned_Courses.objects.filter(Course_id_id=id)
        parsed_date = datetime.strptime(day, '%Y-%m-%d')
        context = {
            'assigned_students': assigned_students,
        }
        return render(request, 'facultypages/Take_Attendance.html', context)

    def post(self, request, id,day):
        assigned_students = Assigned_Courses.objects.filter(Course_id_id=id)
        parsed_date = datetime.strptime(day, '%Y-%m-%d')

        for assigned_student in assigned_students:
            attendance_status = request.POST.get(str(assigned_student.Student_id.id))
            is_present = attendance_status == 'on'
            Attendance.objects.create(student=assigned_student.Student_id, is_present=is_present,
course_id=assigned_student.Course_id,date=parsed_date)

        Timetable.objects.filter(course_id=id).update(status=True)
        return redirect('/staffhome')
```

Code Snippet for Attendance History

- HTML Page

```
<table class="table table-hover text-nowrap">
    <thead>
        <tr>
            <th>Sl NO</th>
            <th>Student</th>
            <th>Attendance Count</th>
            <th>Present Count</th>
            <th>Absent Count</th>
            <th>Average Progress</th>
            <th>Details</th>
        </tr>
    </thead>
    <tbody>
        {% for data in attendance_data|dictsort:"student.usn" %}
        <tr>
```

```
            <td>{{ forloop.counter }}</td>
            <td>
              {% if s.student.pic %}
              <img src="{{data.student.pic.url}}" alt="image" width="60px" height="60px">
              {% else %}
              {% endif %}
              {{data.student.usn}}
            </td>
            <td>{{ data.attendance_count }}</td>
            <td>{{ data.present_count }}</td>
            <td>{{ data.absent_count }}</td>
            <td>
              <div class="progress">
                <div class="progress-bar
                  {% if data.average_progress >= 75 %}progress-success
                  {% elif data.average_progress >= 50 %}progress-warning
                  {% else %}progress-danger
                  {% endif %}" role="progressbar" style="width: {{ data.average_progress }}%;"
    aria-valuenow="{{ data.average_progress }}" aria-valuemin="0" aria-valuemax="100">{{
    data.average_progress }}%
                </div>
              </div>
            </td>
            <td><a href="{% url 'Dates_View' sid=data.student.id  %}"><button type="button"
    class="btn btn-info">Details</button></a></td>
          </tr>
          {% endfor %}
```

- **Views.py**

```
class AttendanceHistoryView(View):
  def get(self, request, course_id):
    course = get_object_or_404(Course, id=course_id)
    students = Student.objects.filter(assigned_courses__Course_id=course_id)
    attendance_data = []
    for student in students:
      attendance_count = student.attendance_set.filter(course_id=course_id).count()
      present_count = student.attendance_set.filter(course_id=course_id, is_present=True).count()
```

```
        absent_count = attendance_count - present_count
        average_progress = int((present_count / attendance_count) * 100) if attendance_count > 0 else
        attendance_data.append({
            'student': student,
            'attendance_count': attendance_count,
            'present_count': present_count,
            'absent_count': absent_count,
            'average_progress': average_progress
        })
    context = {
        'attendance_data': attendance_data,
        'course': course,
        'students':students
    }
    return render(request, 'facultypages/Attendance_History.html', context)
```

Chapter 6

TESTING

The Student Attendance Management System underwent thorough assessments to verify its accuracy, usability, and functionality. This included testing individual parts, how they work together, and ensuring it meets user needs. Different scenarios were checked, like attendance marking and report generation. This testing phase confirmed the system's reliability, performance, and overall user-friendliness.

6.1 Test Plan and Test Cases:

Introduction:

Software testing is the practice of examining a software item to find inconsistencies between provided input and expected output. Also, to assess the functionality of software. Testing determines the quality of a products; software testing is an important phase in the development software. The software testing is a verification and validation procedure. Testing phase supports 64 testing techniques from which in this work I have tested basic 3 techniques these are described below. All implemented test cases are tested as shown in the table given below,

Unit testing: Unit testing is the process of testing a single unit or a group of linked units. It falls under the heading of white box testing. It is commonly used by programmers to guarantee that the unit they have constructed produces the intended result when given certain input.

Integration testing: When a group of components is merged to produce a result, it is called integration testing. Integration testing is used to assess the interaction between software and physical components when they have a relationship. Both white box and black box testing could be applied.

System testing: The process of ensuring that software works in a variety of situations is known as system testing. It still works in various scenarios (for example, operating system). The entire system is put through its paces. It falls under the heading of black box testing because of its environment and implementation.

Validation Testing: The system has been tested and implemented successfully and thus ensured that all requirements as listed in the software requirements specification are completely fulfilled. In case of erroneous input corresponding error messages are displayed. Testing is the process of executing a program with the intent of finding an error. A good test case is one that high probability of finding analysis yet discovered error. A successful test uncovers an yet undiscovered error. The above object implies a dramatic change in view point. Testing cannot show the absence of a defect, it can show that software errors are present.

Test plan and Test Cases:

Test-ID	Module	Test Case Name	Test Data	Expected Result	Actual Result	Pass/ Fail
TC-001	Login Module	Valid Login	Username: adm Password: admi	Admin successfully logged in	Admin logged in successfully	Pass
TC-002	Login Module	Invalid Login	Username: admin Passwor invalid	Error message: Invali credentials	Error message: Invalid credenti	Pass
TC-003	Department Module	Add department	Enter valid department details	New department added to the system	New department added to the system	Pass
TC-004	Department Management Module	Edit department Details	Update department details	department details updated successfully	department updated successfully	Pass
TC-005	Staff Module	Add Staff	Enter valid Staf details	New Staff added to the system	New Staff added to the sys	Pass
TC-006	Staff Module	Edit Staff Details	Update Staff details	Staff details updated successfully	Staff details updated successfully	Pass
TC-007	Student Module	Student Department	Enter valid Student details	New Student added to the system	New Student added to the system	Pass
TC-008	Student Module	Edit Student Details	Update Student details	Student details updated successfully	Student details updated successfully	Pass
TC-009	Course Module	Add Course	Enter valid Cou details	New Course added to the system	New Course added to the system	Pass
TC-010	Course Module	Edit Course Details	Update Course details	Course details updated successfully	Course details updated successfully	Pass

TC-011	Assign Module	Add Assign	Enter valid Assign details	New Assign added to the system	New Assign added to the system	Pass
TC-012	Assign Module	Edit Assign Details	Update Assign details	Assign details updated successfully	Assign details updated successfully	Pass
TC-013	Timetable Module	Add Timetable	Enter valid Timetable details	New Timetable added to the system	New Timetable added to the system	Pass
TC-014	Timetable Module	Edit Timetable Details	Update Timetable details	Timetable details updated successfully	Timetable details updated successfully	pass
TC-015	Attendance Module	Add Attendance	Enter valid Attendance details	New Attendance added to the system	New Attendance added to the system	Pass
TC-016	Attendance Module	Edit Attendance Details	Update Attendance details	Attendance details updated successfully	Attendance details updated successfully	Pass

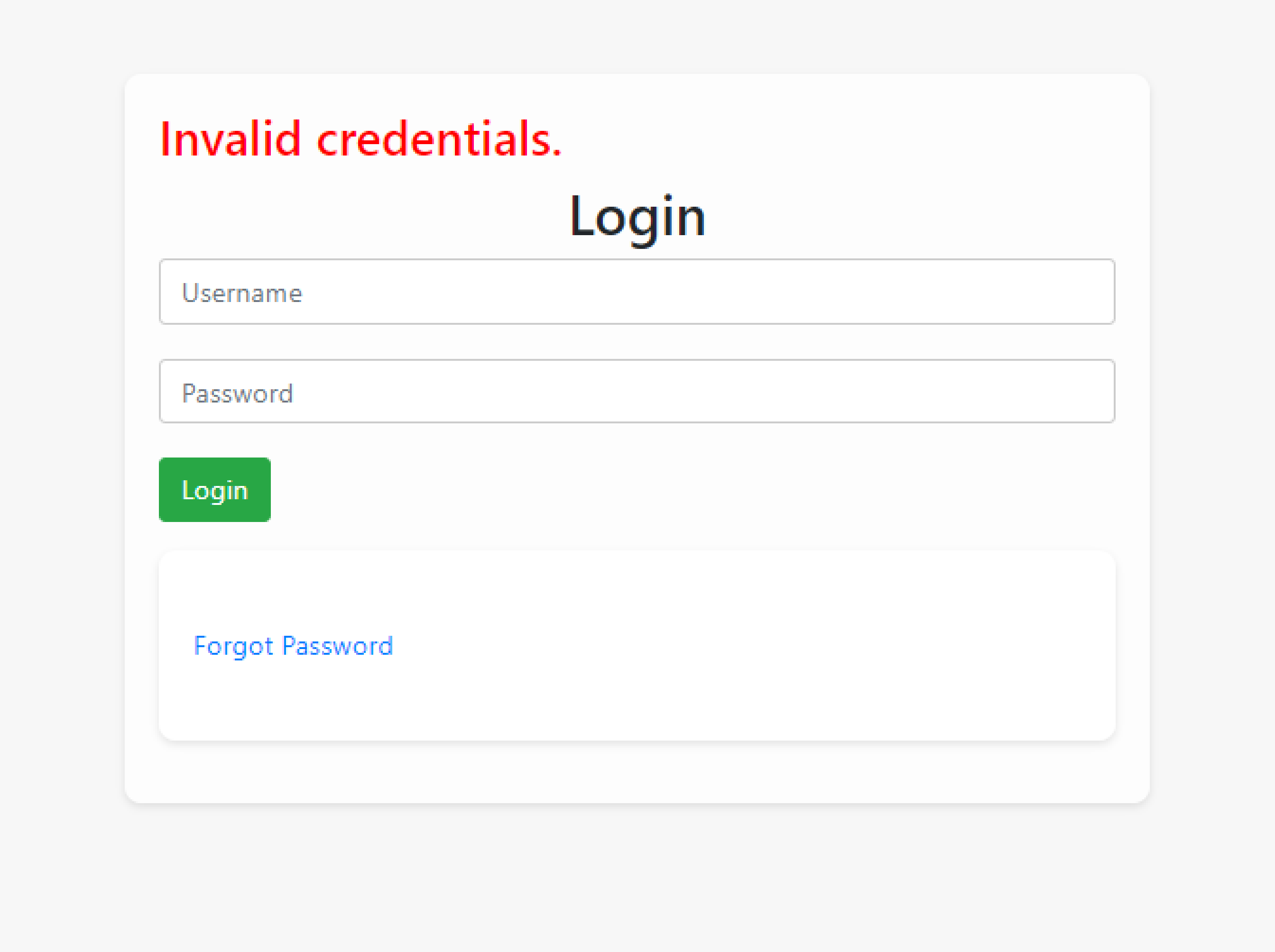
Tab. 6.1 Test plan and Test Cases

CHAPTER 7

RESULTS & DISCUSSIONS

The results and discussions of implementing the Student Attendance Management System showcase its effectiveness in simplifying attendance tracking. Automation improves accuracy, while the user-friendly interface promotes engagement. Generated reports offer insights for better decision-making. Overall, the system streamlines operations and enhances student involvement.

7.1 Login Authentication fail:

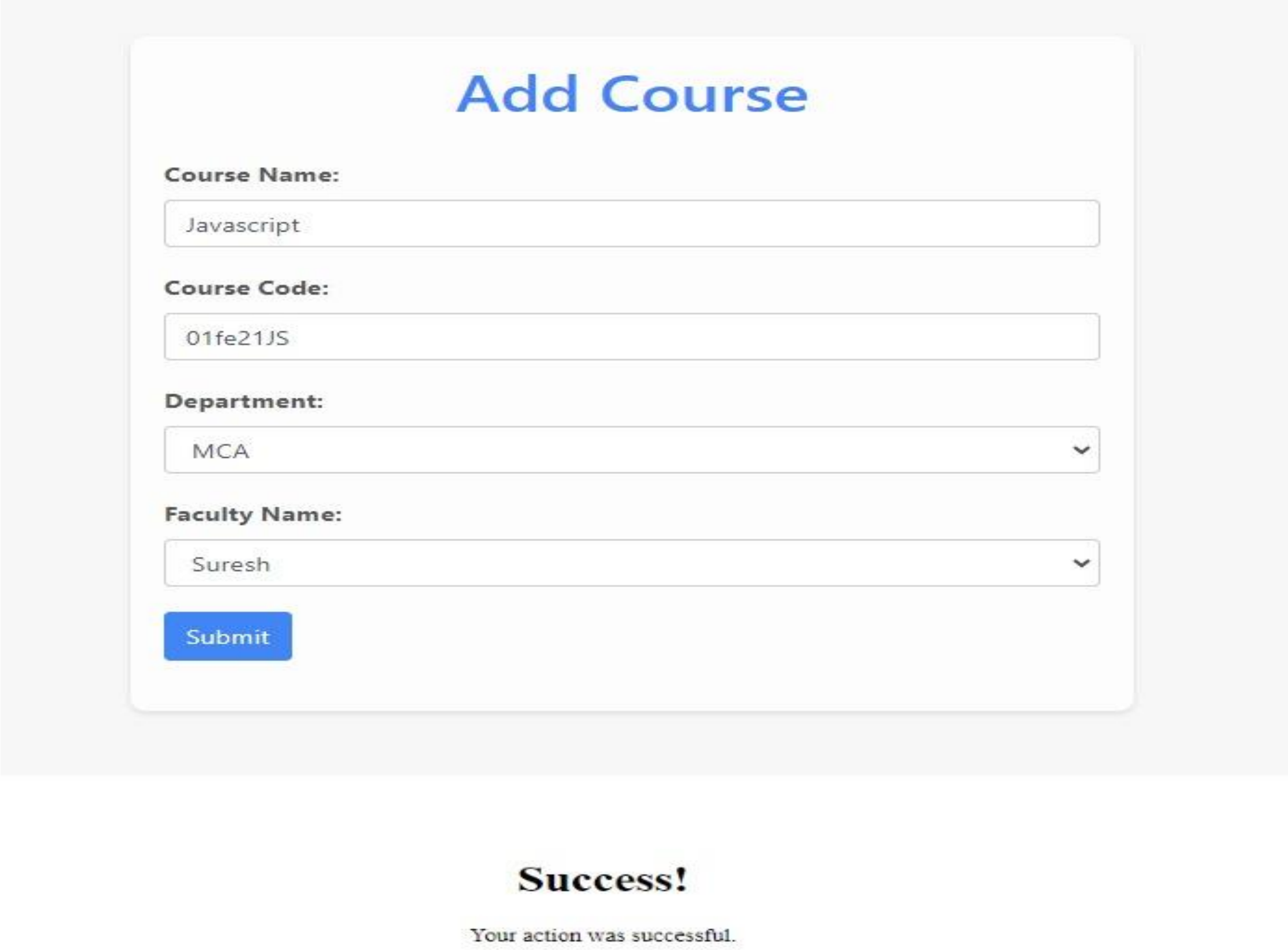


Invalid credentials.

Fig 7.1 Login Authentication Fail

Fig 7.1 Login Authentication Fail shows that when login credentials are incorrect, the system notifies users about authentication failure, preventing unauthorized access and ensuring security.

7.2 Course Information Created Successfully



Add Course

Course Name: Javascript

Course Code: 01fe21JS

Department: MCA

Faculty Name: Suresh

Success!

Your action was successful.

Fig 7.2 Add Course Successfully

Fig 7.2 Add Course Successfully shows that the addition of courses is executed seamlessly, ensuring successful integration into the system's curriculum management with accuracy and efficiency.

CONCLUSION

In conclusion, the Student Attendance Management System serves as an invaluable tool for educational institutions, revolutionizing the way attendance is tracked and managed. By automating processes, improving accuracy, and enhancing communication between stakeholders, the system contributes to streamlined operations and increased student engagement. Its user-friendly interface and insightful reporting capabilities make it an essential asset for educators, administrators, students, and parents. As the system continues to evolve and adapt to changing educational landscapes, it holds the potential to further enhance efficiency and student success in the long run.

FUTURE ENHANCEMENT

Looking ahead, the Student Attendance Management System has exciting potential for growth. By incorporating enhancements like biometric authentication, predictive analytics, mobile apps, and automated reminders, it will become even more efficient and user-friendly. The system's adaptability to multiple languages, integration with other tools and machine learning will make it a vital part of modern education. As organizations seek efficiency and transparency, these advancements will continue to improve attendance tracking, ensuring students' success and making education more accessible and effective for all.

BIBLIOGRAPHY

[1] Ian Somerville, Software Engineering, 9th, Pearson Ed, 2015

[2] Ramez Elmasri, Shamkant B. Navathe, Database Systems, Sixth Edition, PEARSON,2013.

[3] Robert W Sebesta, Programming the World Wide Web, 8th Edition, Pearson education, 2015.

[4] https://getbootstrap.com/docs/5.0

[5] Bootstrap Icons · Official open source SVG icon library for Bootstrap (getbootstrap.com)

[6] https://stackoverflow.com/

[7] https://dev.mysql.com/doc/workbench/en/

[8] https://masteringdjango.com/django

[9] tutorials/mastering-Django-models/

[10] Introducing Microsoft .NET, Second Edition author David S. Platt.

[11] Joe Mayo, “Microsoft Visual Studio 2010: A Beginner's Guide”, Tata McGraw Hill, 2010.

[12] Alex Mackey, “Introducing .NET 4.0: With Visual Studio 2010”, Press, USA, 2010.

APPENDIX

DESCRIPTION ON TECHNOLOGY USED

HTML: Web pages and net programs are created the use of Hypertext Markup Language, that's a popular markup language. It paperwork a triad with Cascading Style Sheets (CSS) and JavaScript as cornerstone technology of the World Wide Web. Images and different objects, which include interactive paperwork, can be embedded into the rendered web page the use of HTML constructs.

CSS: Cascading Style Sheets portrays how HTML additives are to be proven on screen, paper or in different media. CSS spares a parcel of work. It can manipulate the layout of various net pages all at once. CSS can be a basis generation of the World Wide Web, close by HTML and JavaScript

JAVASCRIPT: JavaScript, regularly alluded to as JS. is a deciphered, significant level programming language. The World Wide Web was based on JavaScript as one of its three center innovations It engages naturally Web pages and in this manner is a fundamental piece of web applications. The unending bigger piece of sites uses it and all the significant internet browsers have a serious JavaScript motor to execute it

Bootstrap: An open-source CSS system that objectives responsive, versatile first web advancement, Bootstrap is free and open-source. This incorporates CSS-and (alternatively) JavaScript-based plan layouts for typography, structures, buttons, route, and other connection point components. Bootstrap is a system that permits you to plan sites quicker and more straightforward.

Django: Django is a free, open-source Python based High-Level Web Framework. It basically follows the Model View Template (MVT) architectural pattern. It was originally created by Adrian Holovaty and Simon Willison It was created on 2003 at Lawrence Journal World Newspaper.

PYTHON: The Python language is an interpreted, objective, high-level language with dynamic semantics. The easy-to-learn syntax of Python emphasizes readability and therefore reduces the cost of program maintenance. Modularity and code reuse are encouraged with Python's modules and packages.

SQLite3: SQLite is a relational database management system (RDBMS) compiled into a C library, Contrary to most other database management systems, SQLite is not a client-server engine. SQLite is embedded inside the application. For local/client storage in application programs such as web browsers, SQLite is a popular choice.

EXPLANATION ON TOOLS

PYCHARM: PyCharm is a computer programming integrated development environment (IDE) that focuses on the Python programming language. PyCharm is available in Windows, Mac OS X. and Linux versions. There is a Community Edition that is provided under the Apache License, a Professional Edition with more features that is released under a subscription-funded proprietary 2022 license, and an educational edition.

CHROME: Google Chrome is a cross-platform web browser that was created by the company Google. It was first published for Microsoft Windows in 2008, and it was made using free software components from Apple Web Kit and Mozilla Firefox. It was eventually ported to Linux, macOS. iOS and Android, where it is now the operating system's default browser.

DIAGRAMS.NET: diagrams.net (formerly draw.io) is a cross-platform graph sketching application written in HTML5 and JavaScript that is free and open source. Flowcharts, wireframes. UML diagrams, organizational charts, and network diagrams can all be created using its interface. diagrams net is a cross-browser web app as well as an offline desktop application for Linux, macOS, and Windows. The Electron framework is used to create its offline application.

ERDPLUS: ERDPlus is a web-based database modeling tool that is quick and easy to create. Entity-relationship diagram (ERD) Relational schema (relational diagram) Star schema (dimension model)