# Chapter 1

## INTRODUCTION

### Aim of the project

The aim of our Student Attendance System project in college is to modernize and simplify the process of monitoring and managing student attendance. We aim to move away from the traditional pen-and-paper method, which can be time-consuming and prone to errors, towards an automated system that enhances efficiency and accuracy. Our goal is to provide college administrators and faculty members with a user-friendly platform that allows for easy recording and tracking of student attendance, thereby reducing administrative burden and improving overall productivity. By leveraging technology such as biometric scanner or mobile applications we aim to offer students convenient and accessible options for marking their attendance.

Moreover, our project aims to promote transparency and accountability within the college community by providing real-time access to attendance data. Through seamless integration with existing college databases and student information systems, our system will ensure data consistency and facilitate.

effective communication between faculty, students, and administration. Ultimately, our objective is to create a Student Attendance System that not only streamlines administrative processes but also fosters a culture of punctuality and responsibility among students, leading to improved academic performance and success.

### Overview of the project

The Student Attendance Management System, a PHP project, redefines attendance tracking in educational institutions. Automating attendance recording, it liberates educators from manual tasks and integrates seamlessly with school databases, providing instant access to records for efficient monitoring and decision-making.

By fostering a culture of punctuality and accountability among students, it cultivates a conducive learning environment. With its user-friendly interface, educators can navigate attendance data effortlessly, identifying trends and addressing an issues.

promptly. Additionally, the system generates comprehensive reports, offering insights into attendance patterns and facilitating data-driven interventions. Beyond simplifying administrative tasks, this project empowers educators to focus on delivering quality education and supporting student growth. By promoting student ownership of their attendance, it installs a sense of responsibility that extends beyond the classroom. Ultimately, the Student Attendance Management System represents a significant leap forward in attendance management technology, enhancing efficiency, accountability, and student success in educational institutions.

* 1. **Outcome of the project**

The outcome of the Student Attendance Management System project is a profound revolution in attendance tracking for educational institutions. This intricate digital solution simplifies complex processes, offering real-time access to attendance data. It fosters accountability among students and empowers educators to make data-driven decisions swiftly.

Ultimately, it establishes a new standard for efficiency and involvement in attendance management practices.

### BENEFITS

1. Timesaving: Automates attendance tracking, reducing manual effort.
2. Accuracy: Provides real-time, precise attendance records.
3. Efficiency: Streamlines administrative tasks, enhancing productivity.
4. Accountability: Encourages student responsibility and punctuality.
5. Transparency: Facilitates easy access to attendance data for educators and administrators.

# Chapter 2

## SOFTWARE REQUIREMENT

### Software Used:

Operating system : Windows 98,7,8 or 10 or Linux

Languages(Front end) : HTML,PHP and Bootstrap

(Back end) : SQL

IDE : Xampp and MySQL

### Software Description:

* + 1. **XAMPP(PhpMyAdmin)**

PhpMyAdmin can manage a whole MySQL server as well as a single database. To accomplish the latter, you'll need a properly set up MySQL user who can read/write only the desired database. It's up to you to look up the appropriate part in the MySQL manual.

* Xampp browses and drop databases, tables, views, columns and indexes and create, copy, drop, rename and alter databases, tables, columns and indexes.
* Its maintenance server, databases and tables, with proposals on server configuration and execute, edit and bookmark any SQL-statement, even batch-queries.
* It loads text files into tables, create and read dumps of tables and export data to various format of some where : CSV, XML, PDF, 150/IEC 26300 –
* Open Document Text and Spreadsheet, Word, and LTX formats and import data and MySQL structures from Open Document spreadsheets, as well as XML, CSV and SQL files administer multiple servers manage MySQL users
* Privileges and check referential integrity in MyISAM tables and using Query-byexample (QBE), create complex queries automatically connecting required tables and create PDF graphics of your Database layout.
* Create, edit, export, and drop events and triggers communicate in synchronize two databases residing on the same as well as remote servers.
  + 1. **PHP**:
* You need PHP 5.2.0 or newer, with session support, the Standard PHP Library (SPL) extension and JSON support.
* To support uploading of ZIP files, you need the PHP zip extension.
* For proper support of multibyte strings (e.g. UTF 8, which is currently the default), you should install the mbstring and ctype extensions.
* You need GD2 support of multi-byte string(eg. UTF-8, which is currently the default), you should install the Mb string and c Type extensions.

### The SQL Language:

SQL is a language for relational database. SQL is a non-procedural i.e., when we use SQL, we specify what we want to be done not how to do it.

Features of SQL

* SQL is an interactive query language.
* SQL is a database administration language.
* SQL is a database programming language.
* SQL is a client/server language
* SQL is a distributed database language.
* SQL is a database gateway language.

### Basic SQL Commands

* Data Definition Language commands (DDL)
* Data Manipulation Language commands (DML)
* Transaction Control Language commands (TCL)
* Data control Language commands (DCL)

### HTML:

To publish information for global distribution, one needs a university-understood language, a kind of publishing mother tongue that all computers may potentially understand. The publishing language used by the World Wide Web is HTML (Hyper Text Markup Language)

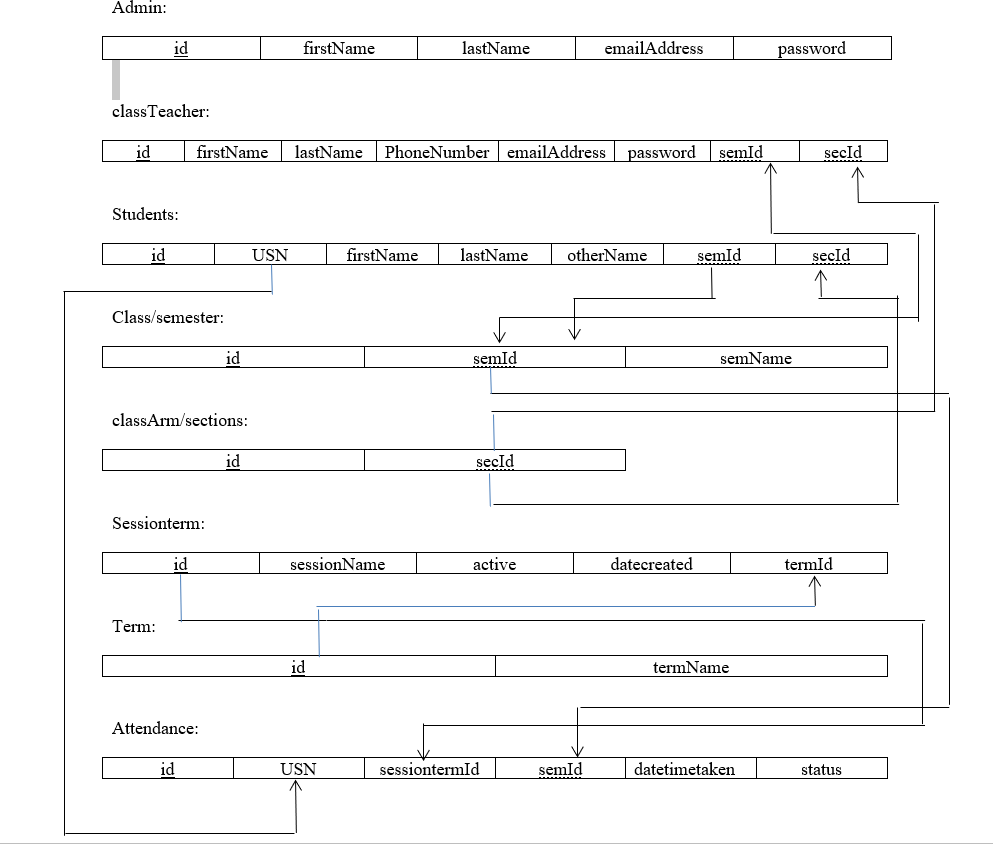
* Publish online documents with headings, text, tables, list, photos etc.
* Retrieve online information via hypertext links, at the click of a button.
* Design forms for conducting transactions with remote services, for use in searching information, making reservation, ordering products etc.
* Includes spreadsheets, video clips, sound clips, and other applications directly in the documents.

### Bootstrap:

Bootstrap is a front-end framework for building responsive websites. It offers pre-designed components and layouts, simplifying web development and ensuring consistency across different devices.

* Responsive Grid: Bootstrap's grid system helps create adaptable layouts for different screen sizes.
* Pre-styled Components: It offers ready-made components like buttons and forms for quick development.
* Customization: Bootstrap can be easily customized to suit specific project needs.
* Browser Compatibility: Ensures websites look and work consistently across different browsers.

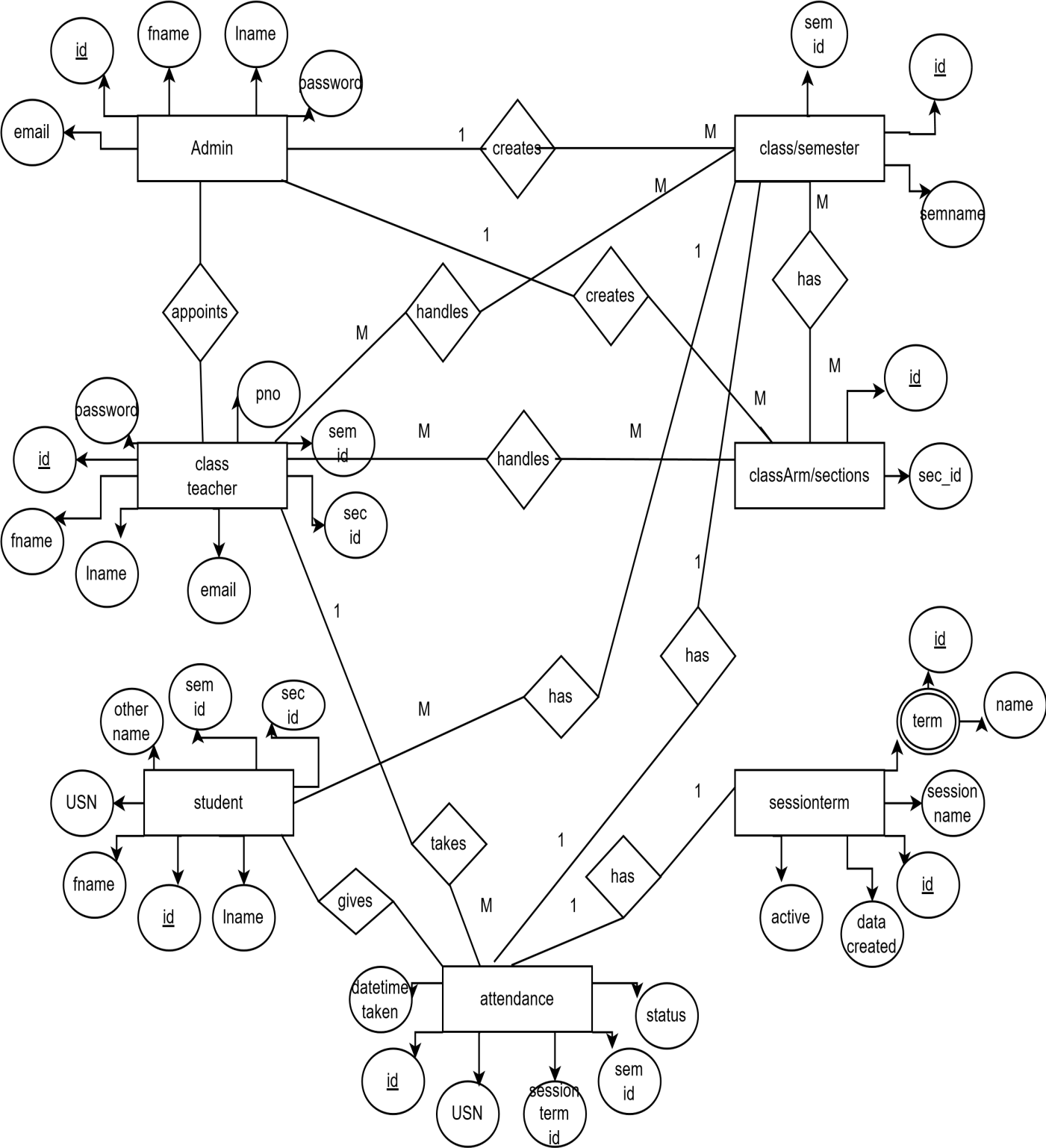
# Chapter 3



## DESIGN

### Schema diagram

* 1. **ER diagram**



### Use case diagram

Login

ADD Class

ADD Faculties

HOD

Modify Branch

ADD Student

Modify Attendance

Take Attendance

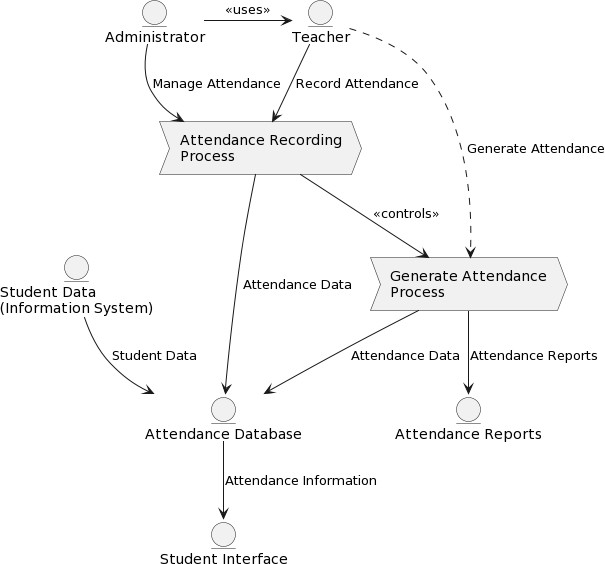
CLASS TEACHER

View Attendance

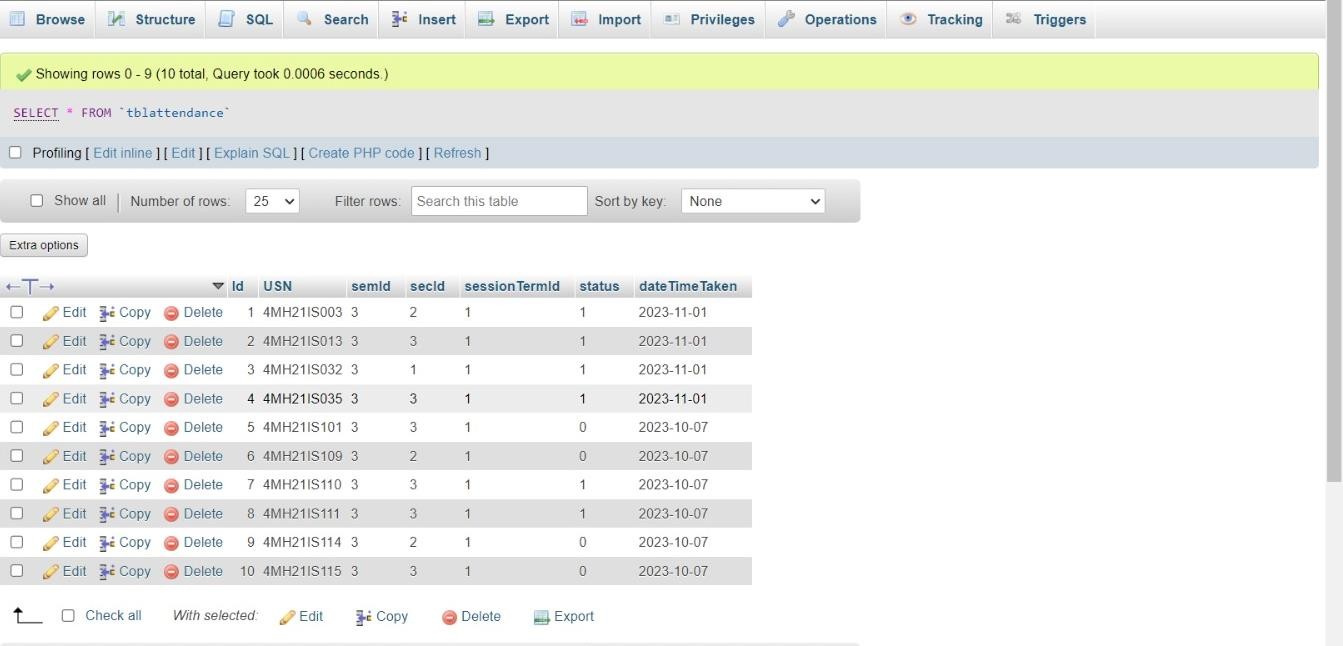
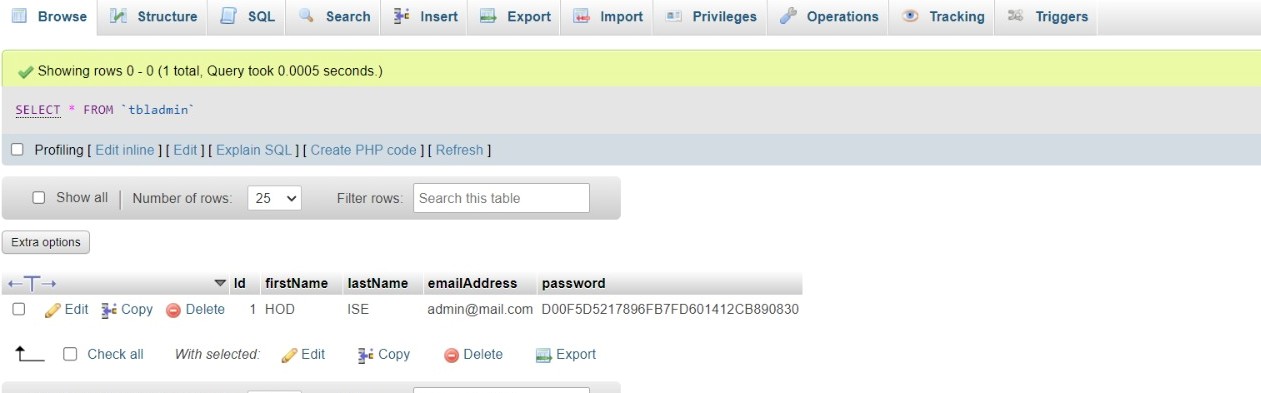
Attendance Report

Log Out

* 1. **Data flow diagram**



# Chapter 4



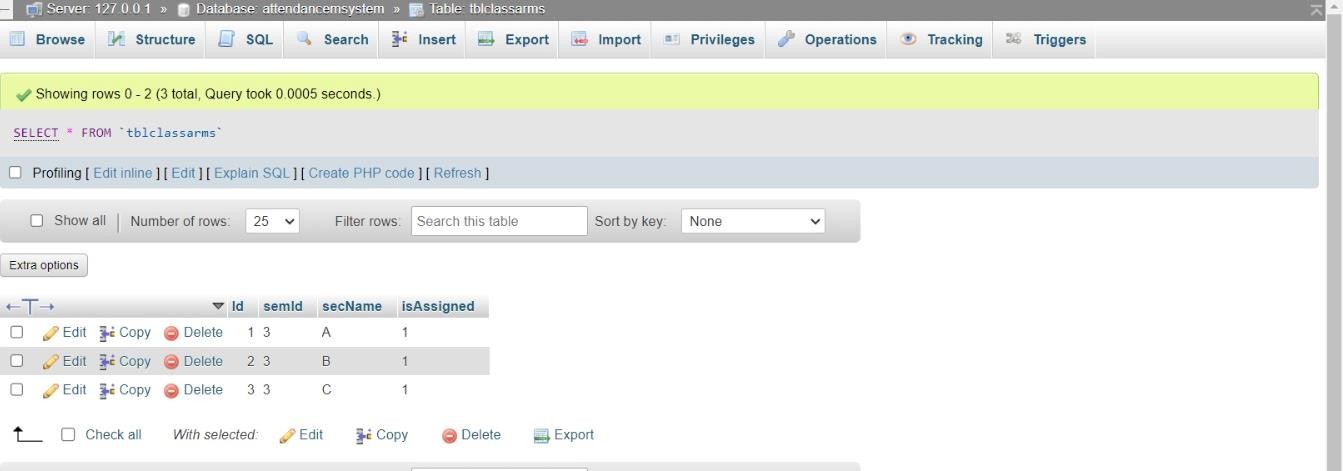
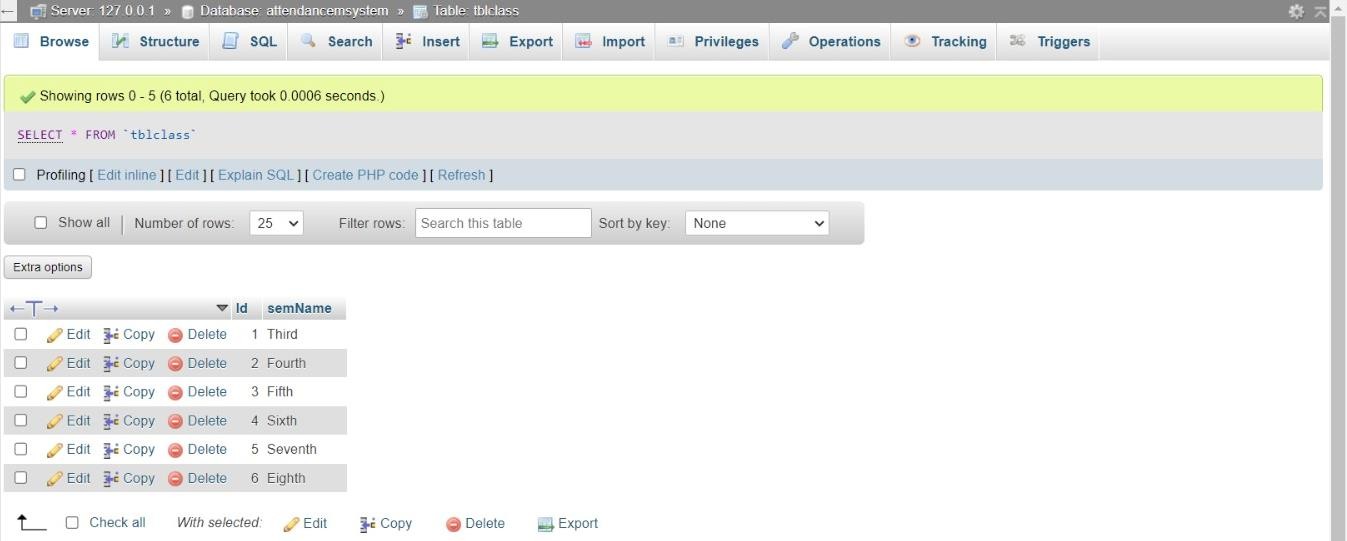
## DESCRIPTION OF TABLES

#### desc tbladmin:

Fig 4.1.1 tbladmin

#### desc tblattendance:

Fig 4.1.2 tblattendance



#### desc tblclass:

Fig 4.1.3 tblclass

#### desc tblclassarm:

Fig 4.1.4 tblclassarms

#### .

**desc tblclassteacher:**

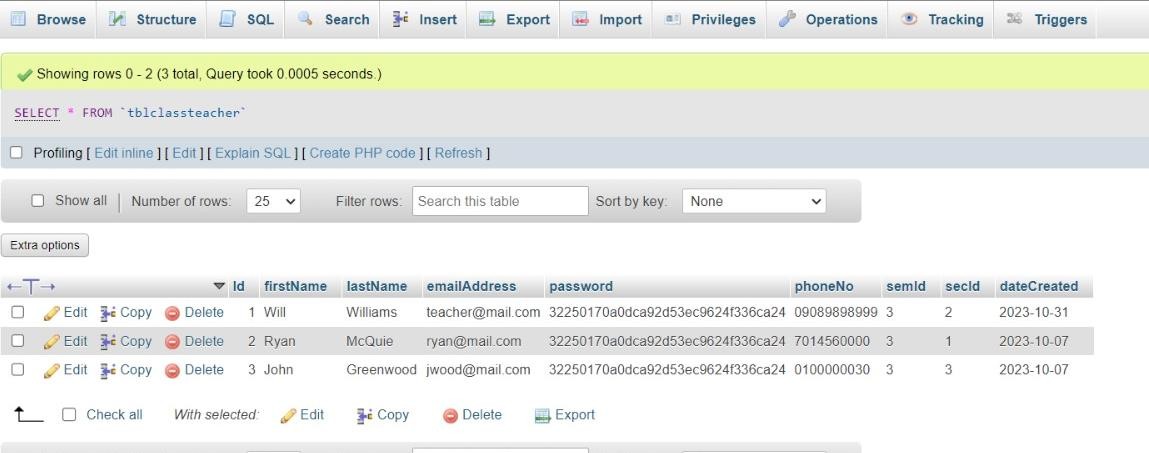


Fig 4.1.5 tblclassteacher

#### desc tblsessionterm:

Fig 4.1.6 tblsessionterm

#### desc tblstudents:

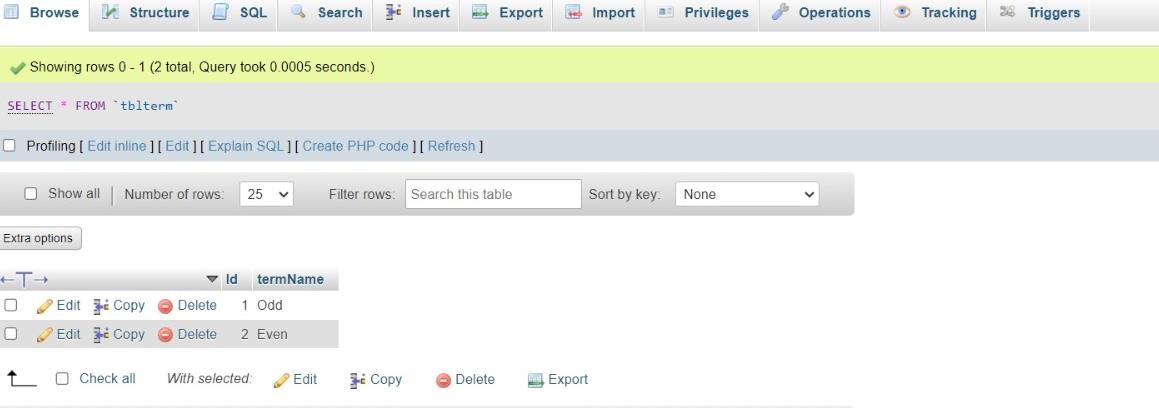
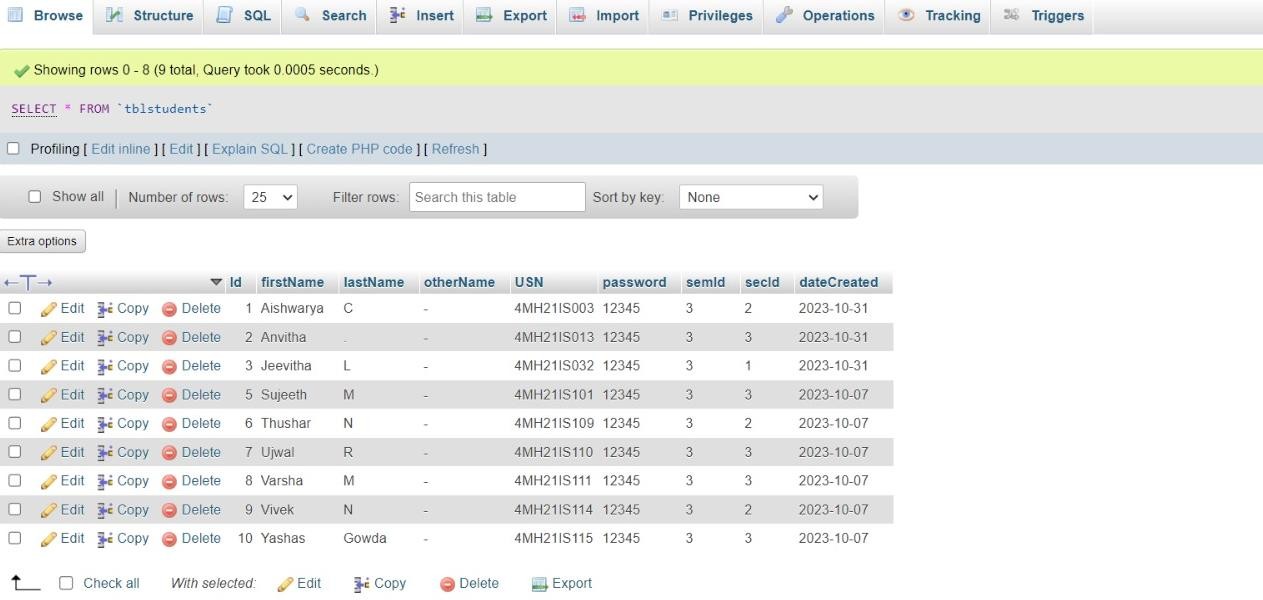
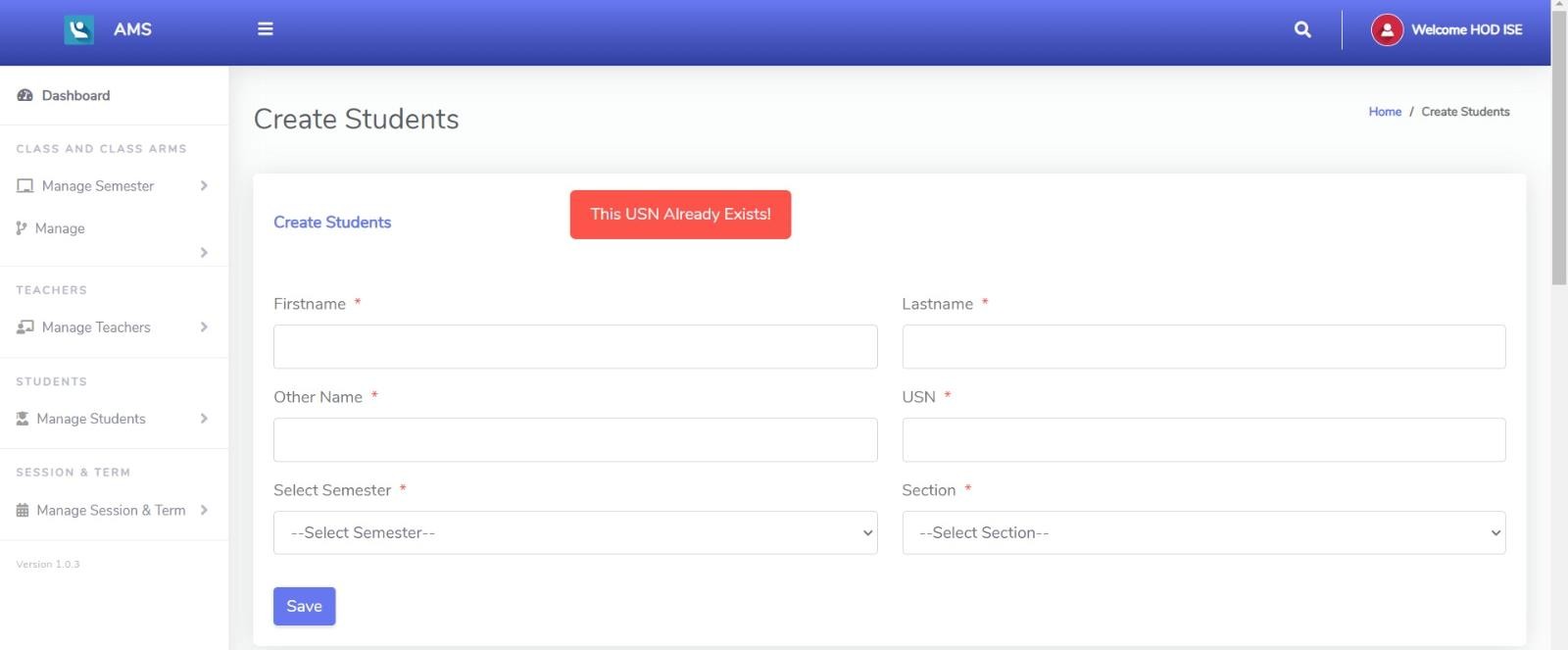
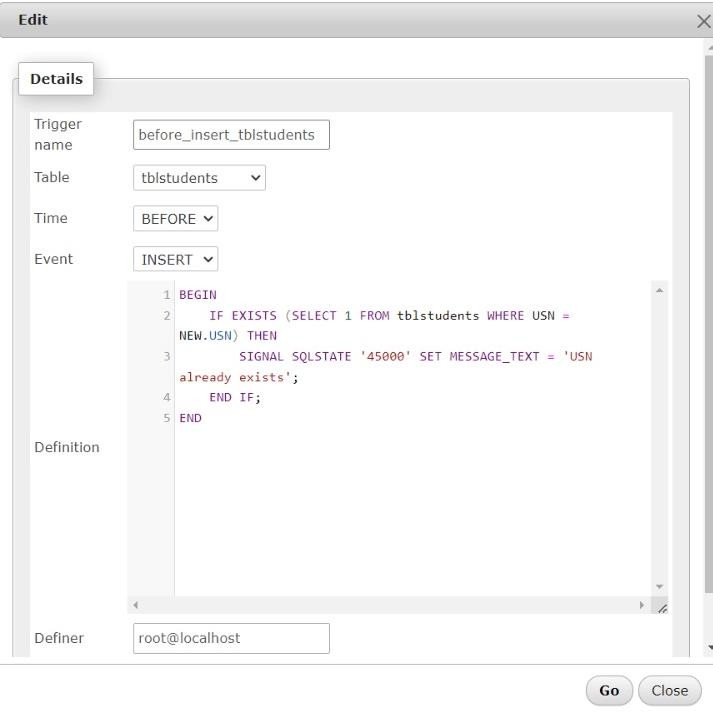


Fig 4.1.7 tblstudents

#### desc tblterm:

Fig 4.1.8 tblterm

# Chapter 5



## RESULT ANALYSIS

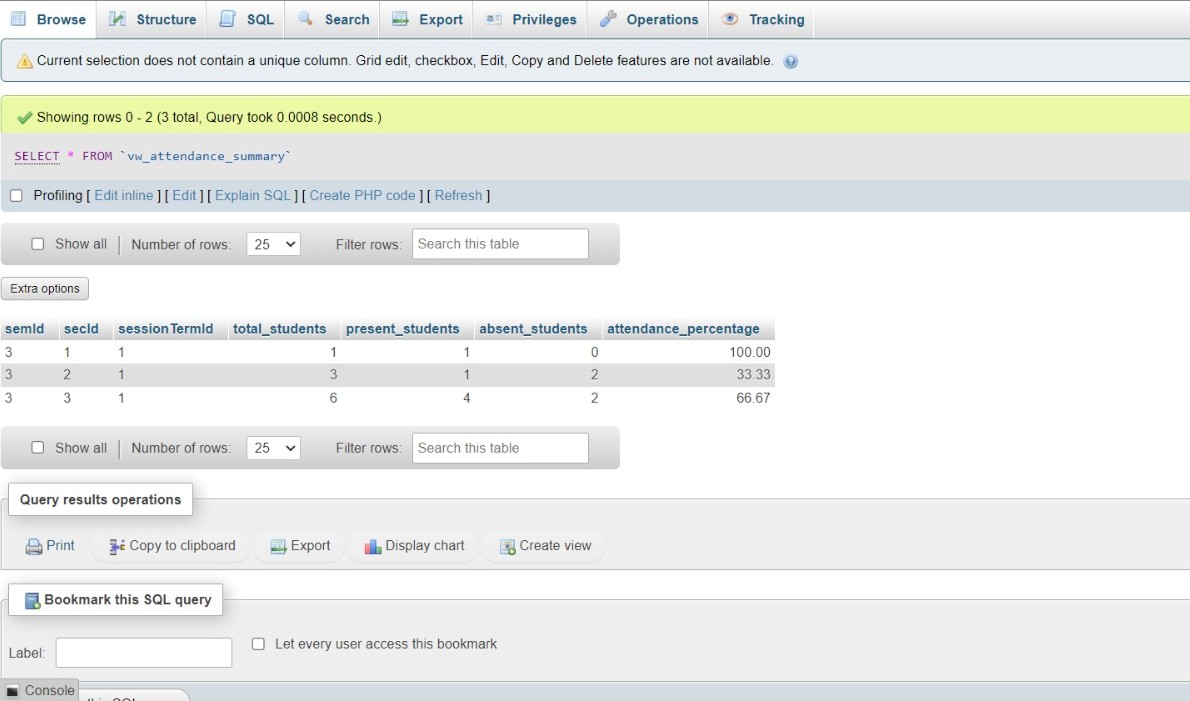
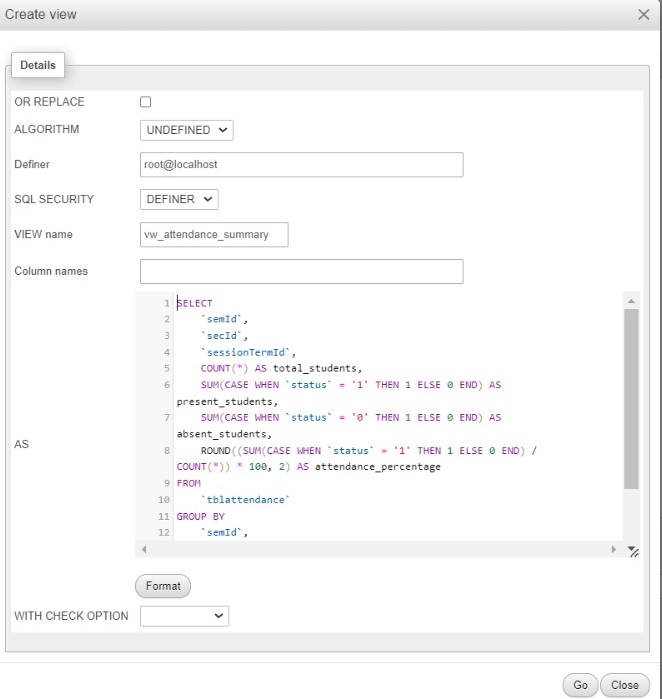
### TRIGGER:

* + - There is a basic trigger in this project.
    - If we make an inserted, then the TRIGGER updates the status value insertion after the event has occurred.

#### Code for Trigger

**Output for applied Trigger**

### VIEW:

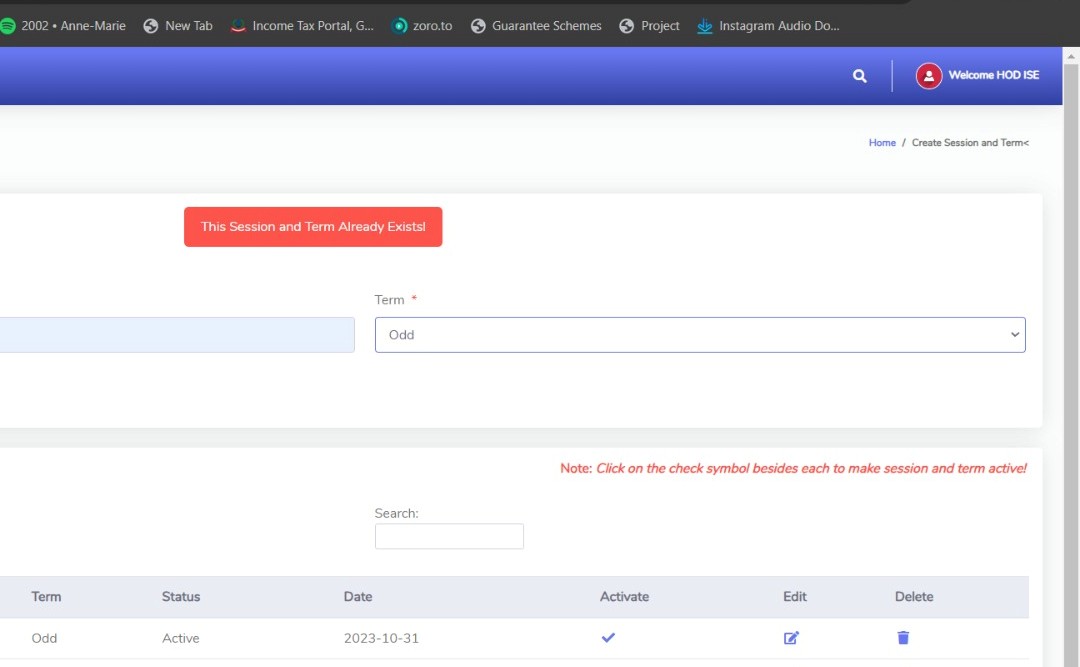
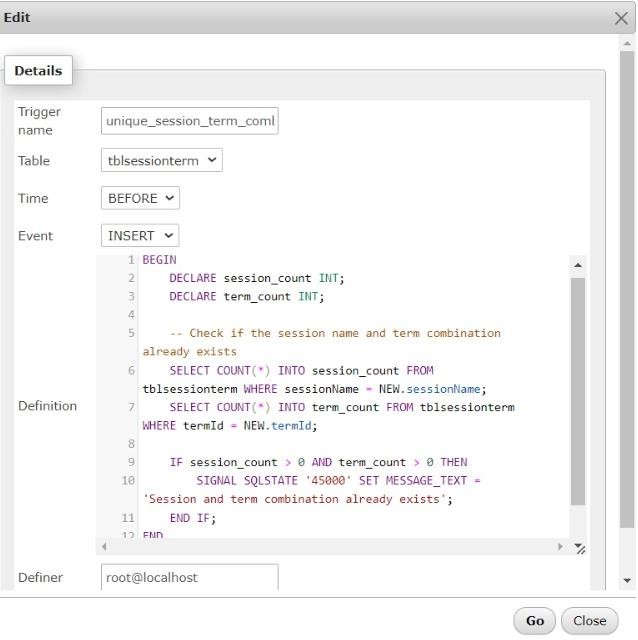


* + - A virtual table derived from one or more tables, presenting selected data for querying.

### Code for creating a view for table attendance

**Output for view creation**

### ASSERTION:



* + - An assertion is a piece of SQL which makes sure a condition is satisfied, else or it stops the action being taken on a database.
    - An assertion is a constraint that might be dependent upon multiple rows of multiple tables.
    - Domain constraints, functional dependency and referential integrity are special forms of assertion are dependent (involve) on single row of a table at a time.
    - Any modification to a database is allowed only if it would not cause any assertion are checked only when UPDATE or INSERT actions are performed against the table.

#### Code for Assertion

**Output for Assertion**

### TESTING:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TEST CASE ID | TEST CASES | CASE TYPE | EXCEPTED OUTPUT | ACTUAL OUTPUT | SATUS |
| 1 | Login with wrong username and wrong password | 1. Invalid username and password | The system will not accept the invalid username  or password | As expected | Fail |
| 2 | Login with correct username and wrong password | 2. Invalid Password | The system will not accept the  invalid password | The system will not allow to login into  the system | Pass |
| 3 | Login with wrong username and correct password | 3.Invalid username | The system will not accept the invalid username | The system will not allow to login into  the system | Pass |
| 4 | Login with correct username and correct password | 1.Require field  validation | Field should not be empty | Users have to enter the value | Pass |

* 1. **NORMALIZATION:**

The complete tables of the database in the project is normalized, obeying all the rules of normalization

### 1NF:

1NF disallows relations within relations or relations as attribute values within tuples. The only attribute values permitted by INF are single atomic (or indivisible) values.

#### Example: tblsessiontrem

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| id | sessionname | termId | is active | datecreated |

Each cell contains a single value. There are no repeating groups or nested structures. The table structure ensures that each attribute contains atomic values, satisfying the requirements of 1NF.

### 2NF:

A functional dependency X Y is a **full functional dependency** if removal of any attribute A from X means that the dependency does not hold any more; that is, for any attribute A epsilon X, (X-{A}) does not functionally determine Y.

#### Example: tblclassarms

|  |  |  |  |
| --- | --- | --- | --- |
| id | semId | secName | is Assigned |

Each attribute (secName and isAssigned) is functionally dependent on the primary key (Id).There are no repeating groups or nested structures, ensuring that the table satisfies the requirements of 1NF. There are no partial dependencies, meaning each non-key attribute depends on the entire primary key. This ensures compliance with the requirements of 2NF

### 3NF:

**Transitive functional dependency**

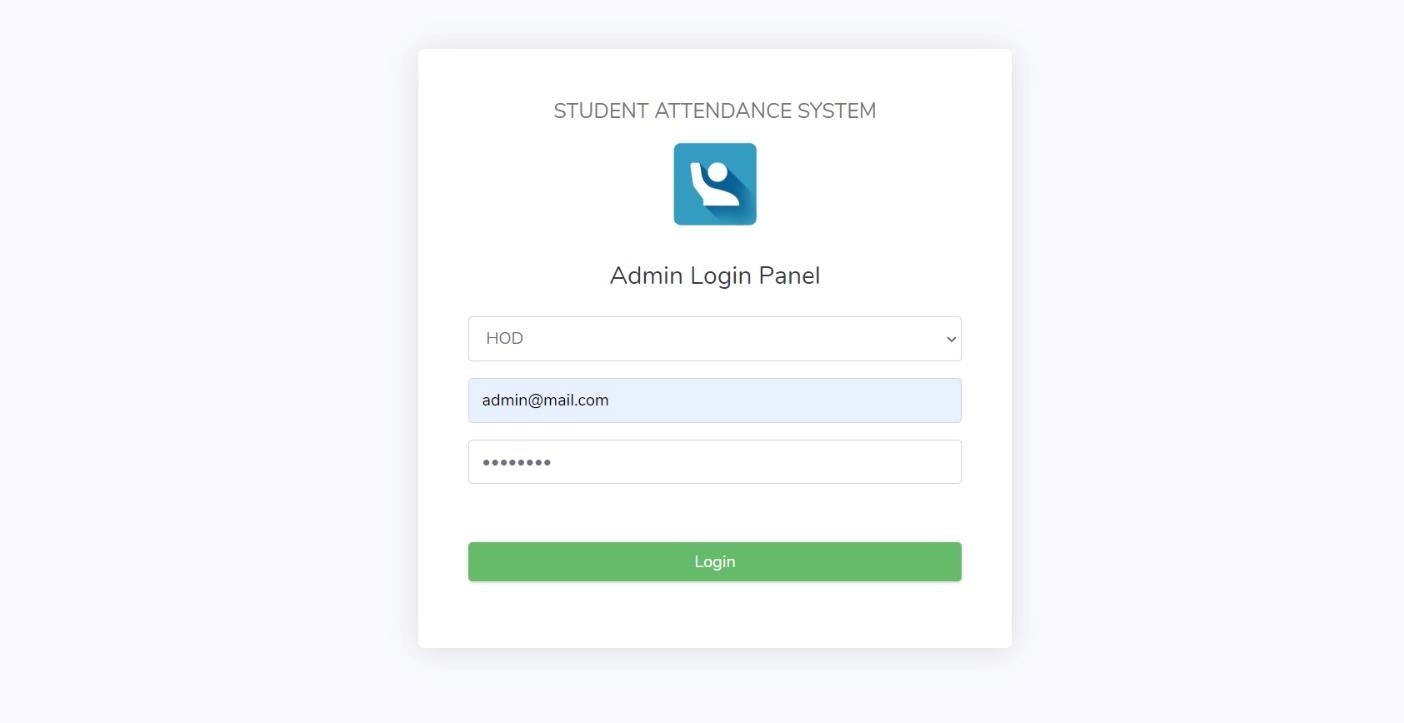
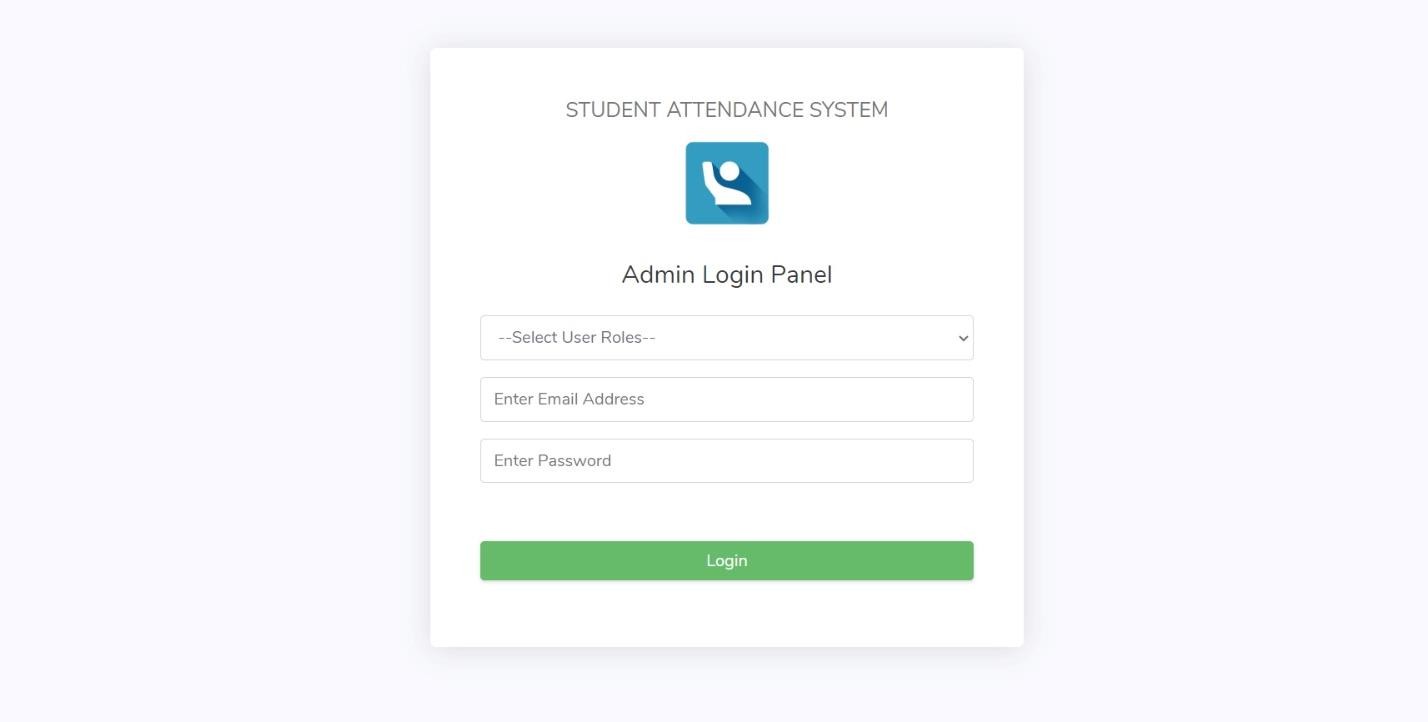
A functional dependency X Y in a relation schema R is a **transitive dependency** if there exists a set of attribute Z. that are neither a primary nor a subset of any key of R(candidate key) and both X Z and Y Z holds.

Definition: A relation schema R is in third normal form (3NF) if it is in 2NF and no non prime attribute A in R is transitively dependent on the primary key.

#### Example: tblstudents

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| id | firstName | lastname | othername | usn | password | semId | secId | datecreated |

tblstudents has a primary key Id and stores information about students. tblclassarms has a primary key Id and stores information about class sections. tblstudents has a foreign key semId referencing the primary key of tblclass table. There's no direct relationship between tblstudents and tblclassarms. Thus ,the 3NF is maintained as there are no transitive dependencies and each table serves a specific purpose.

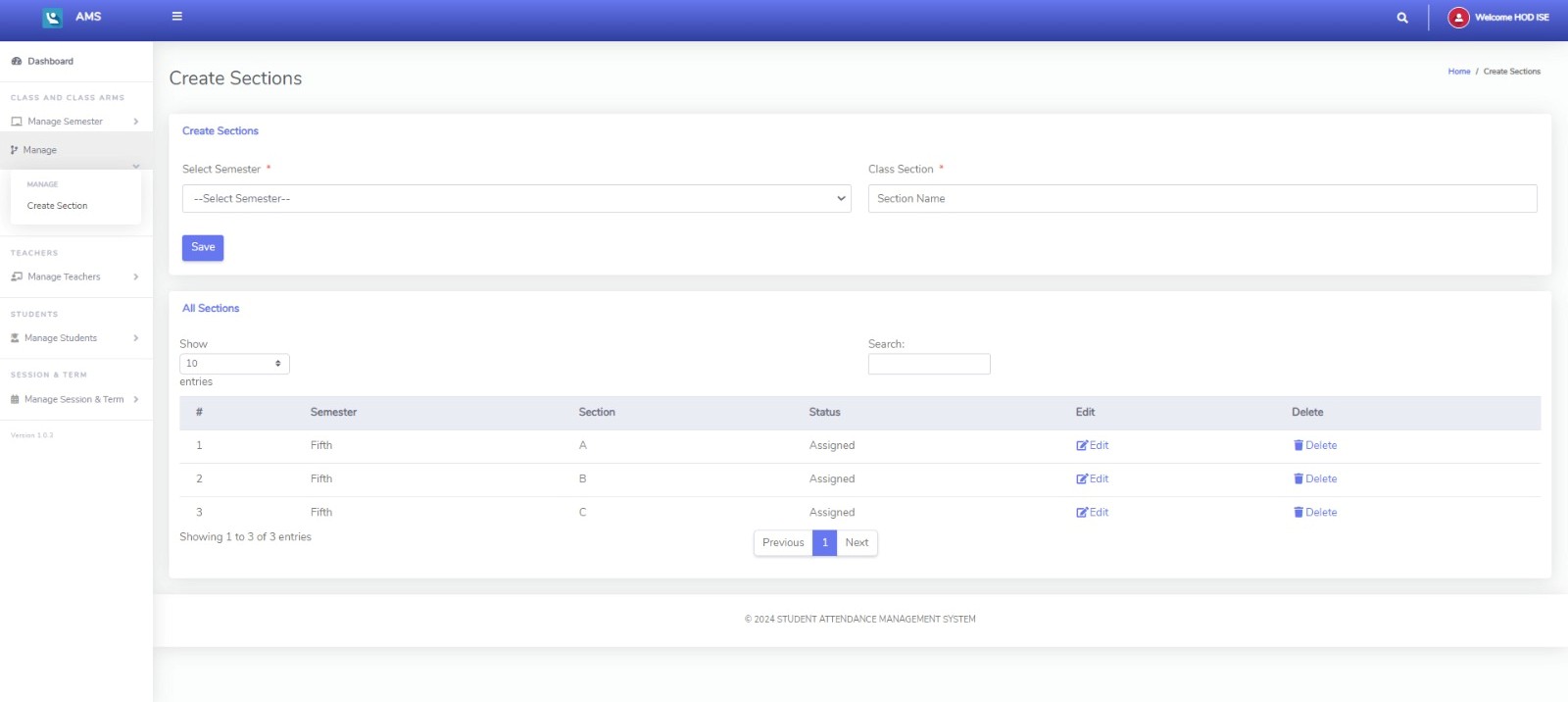
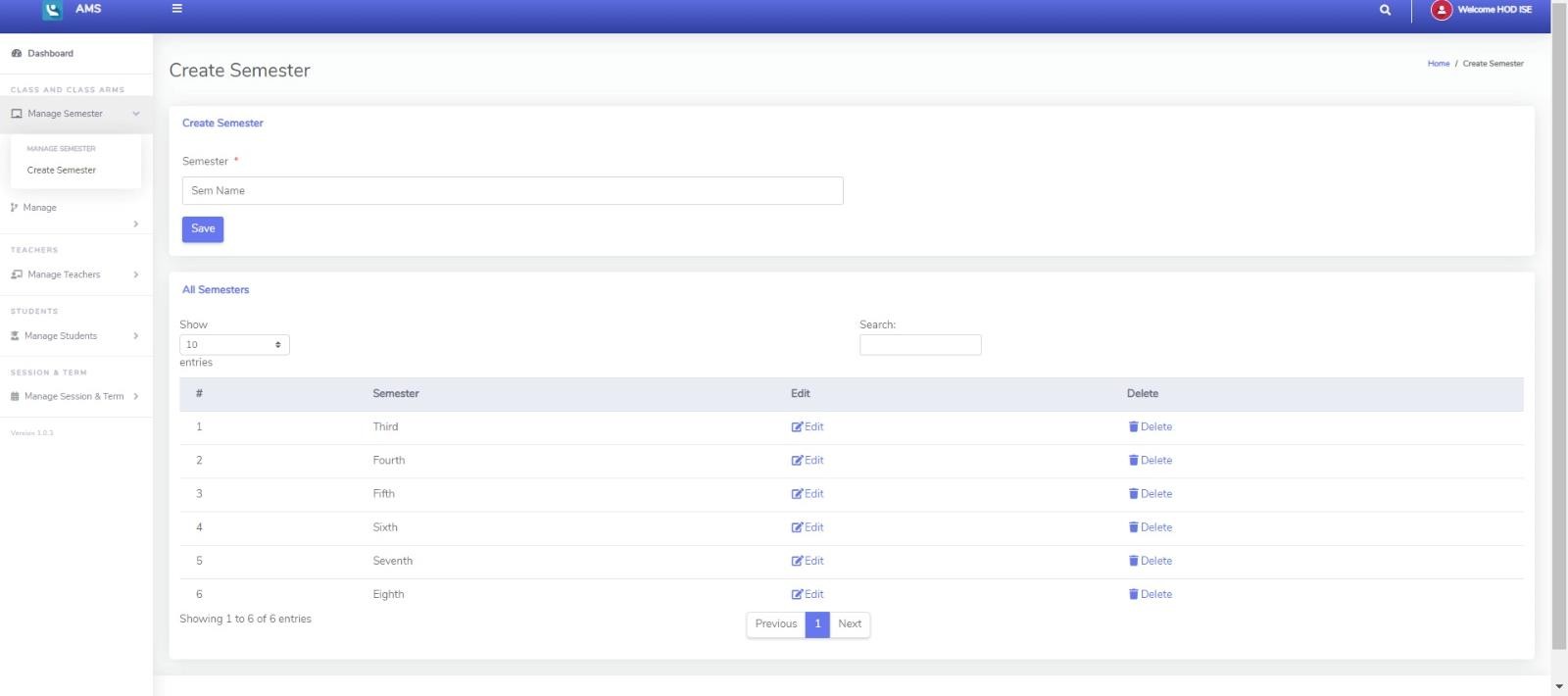


# Chapter 6

## SNAP SHOTS

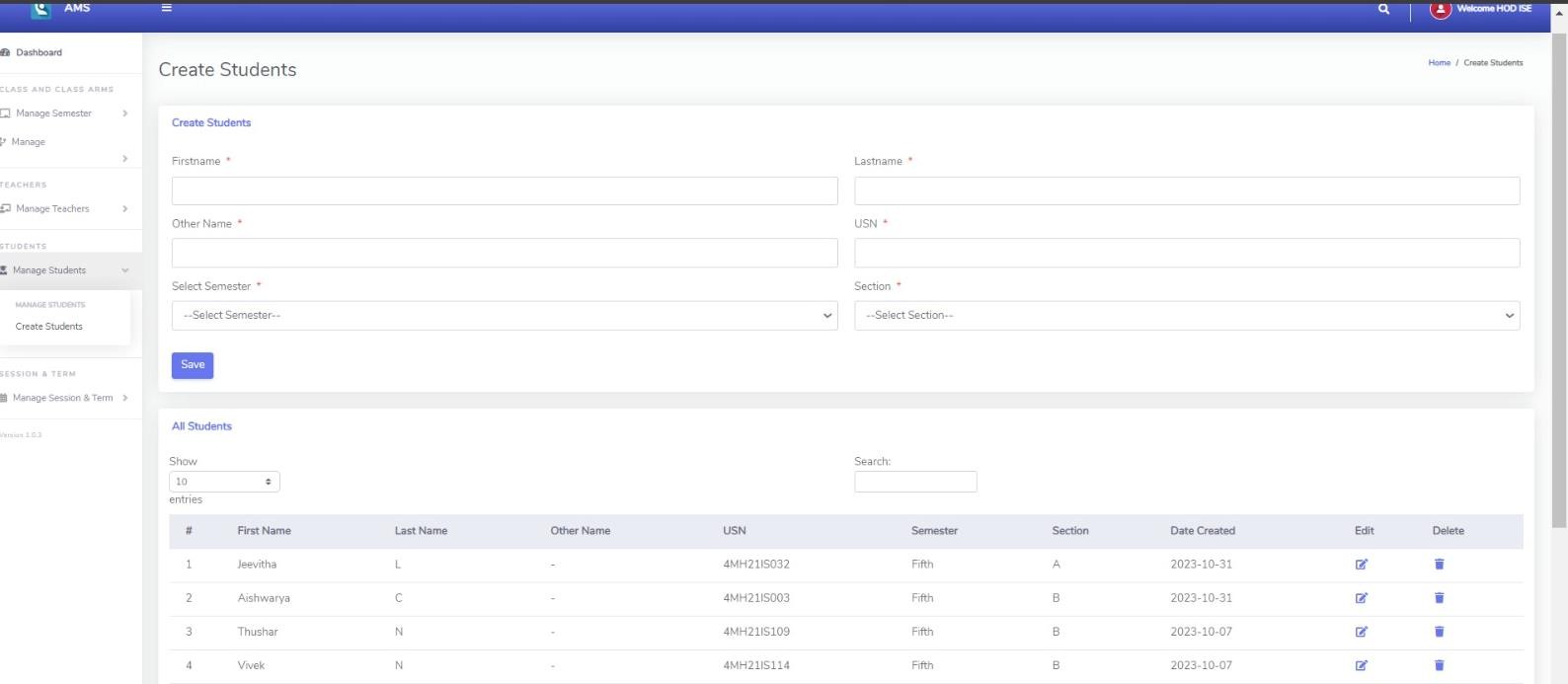
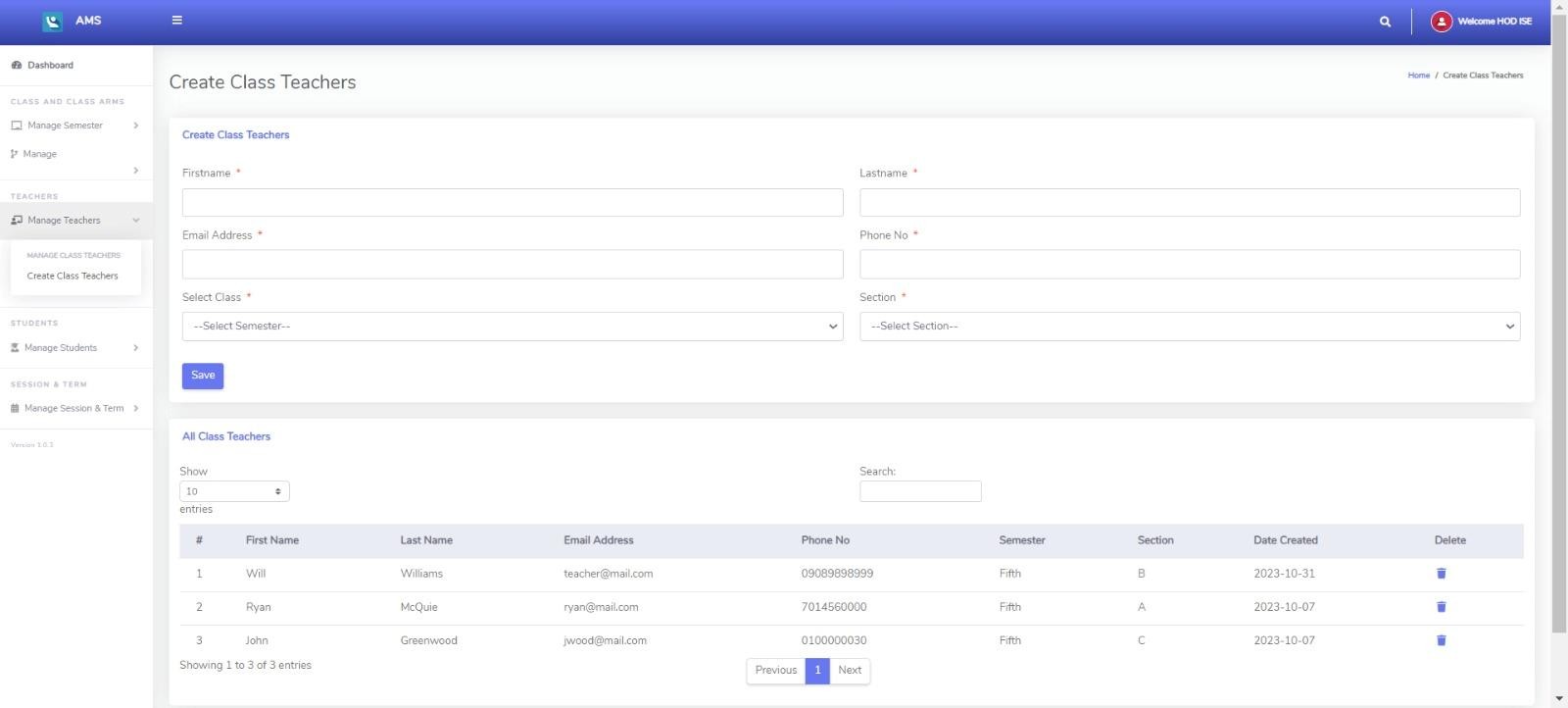
### Fig 6.1.1 Home page

**Fig 6.1.2 HOD login page**



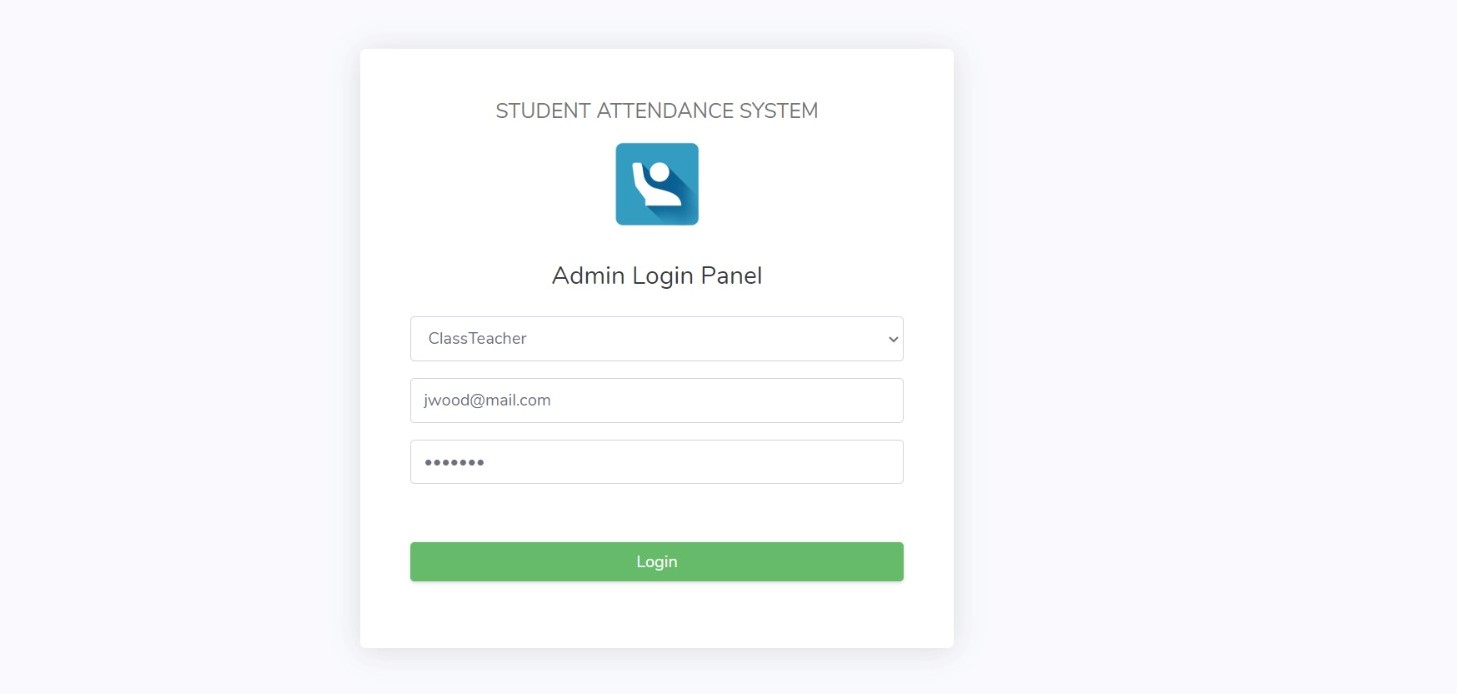
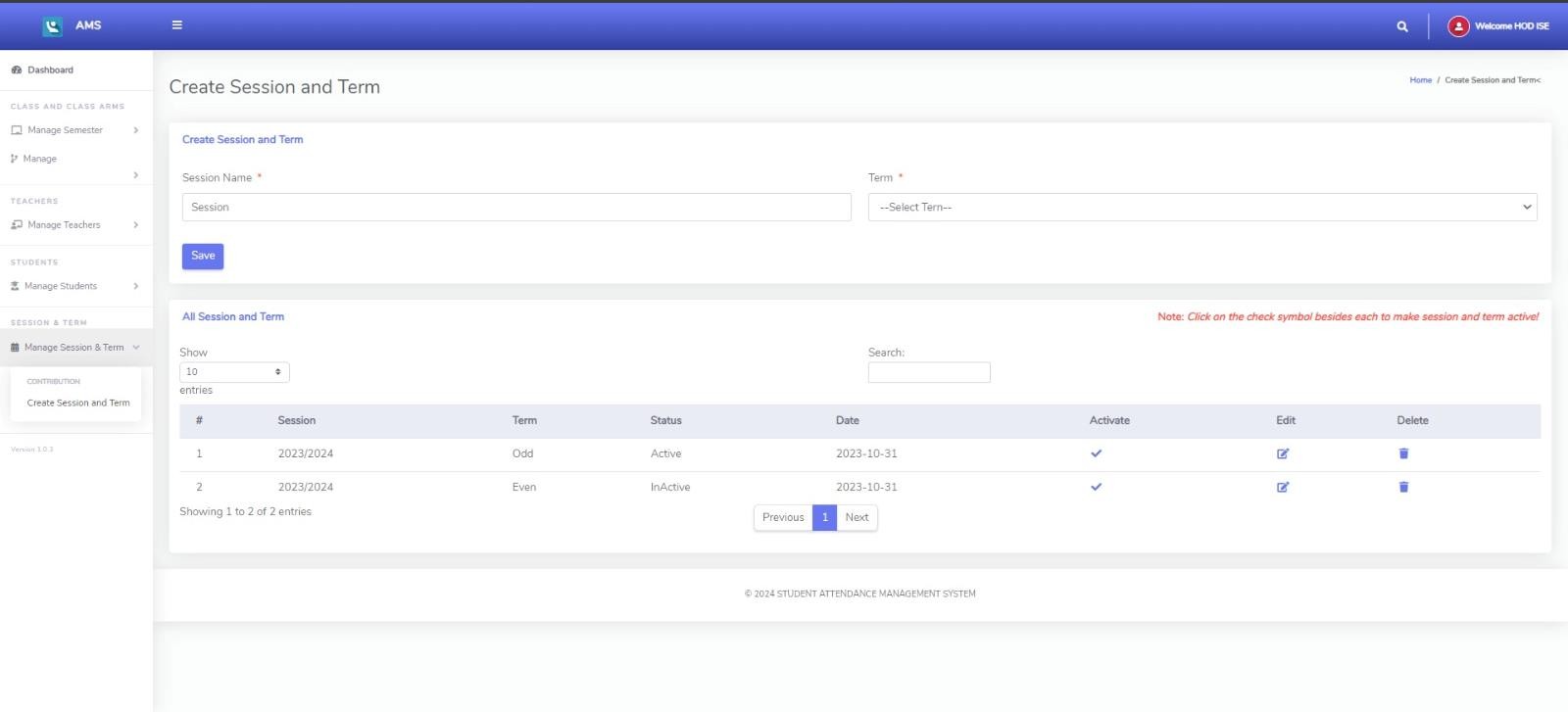
### Fig 6.1.3 Add semester page

**Fig 6.1.4 Add section page**



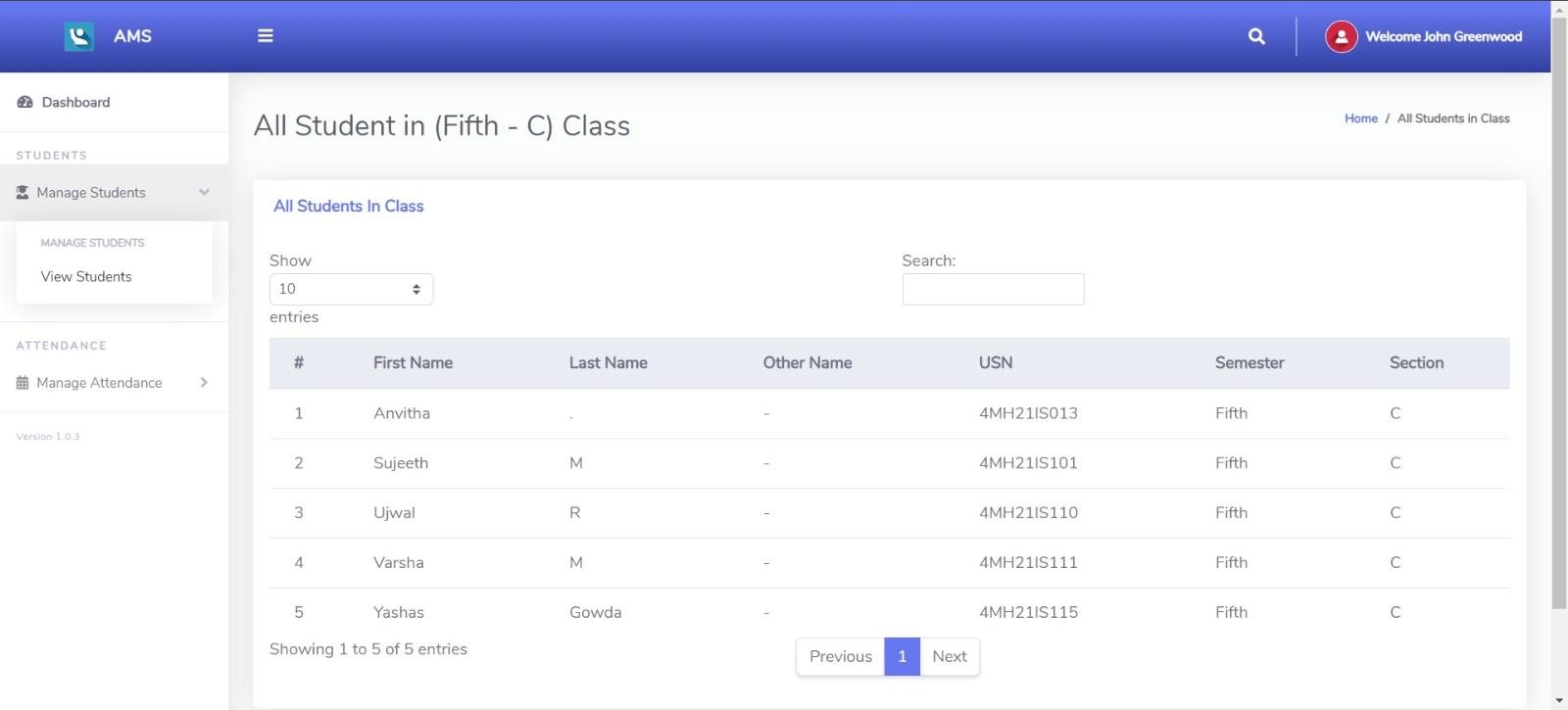
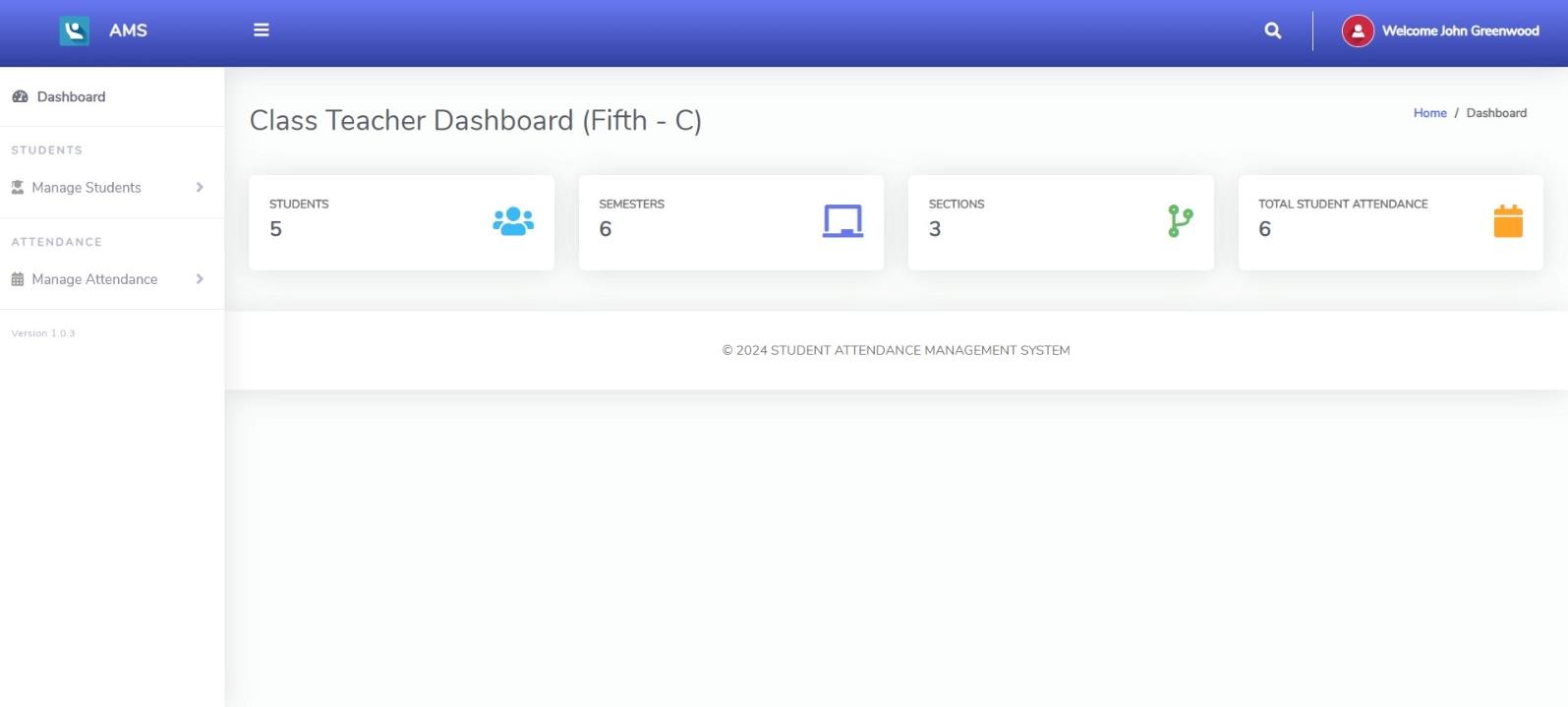
### Fig 6.1.5 Add class teacher

**Fig 6.1.6 Add students page**



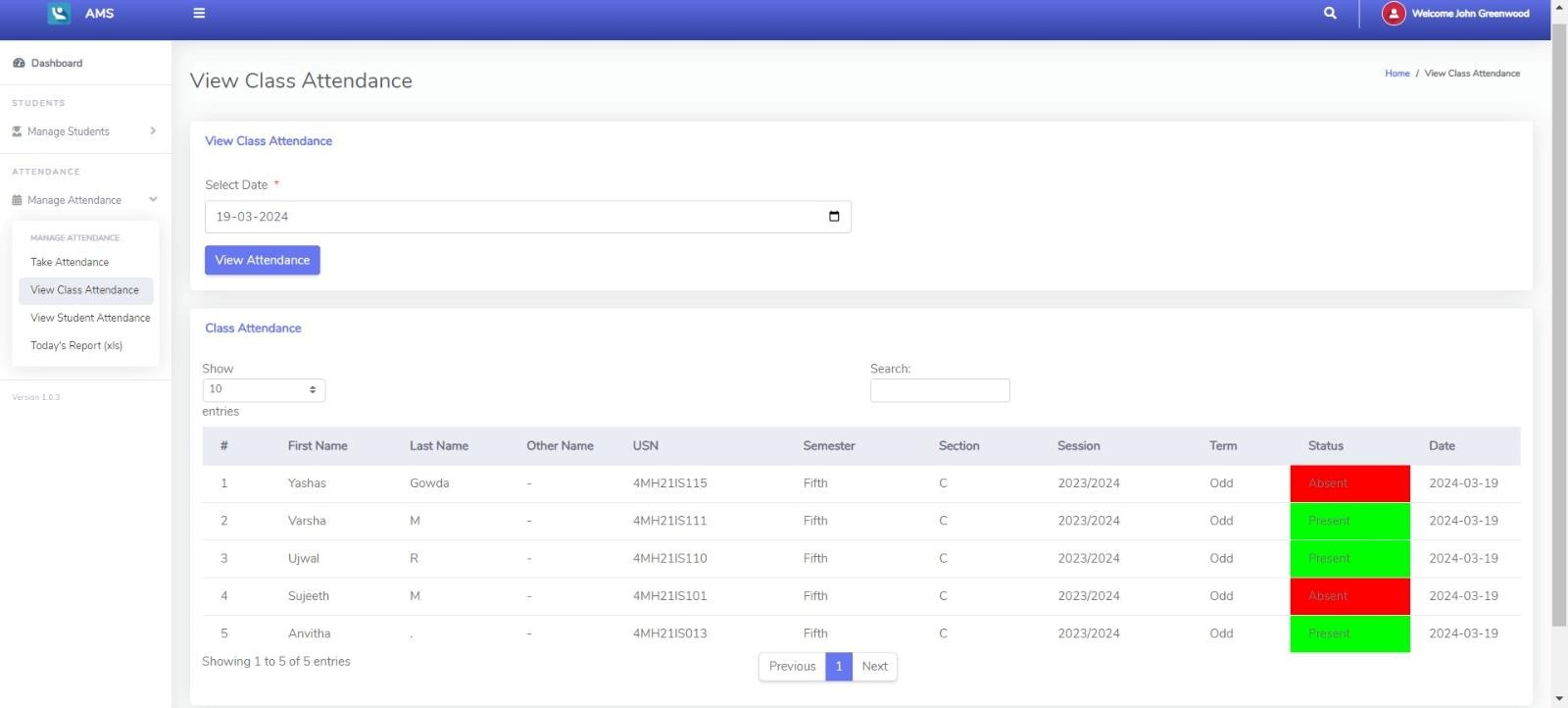
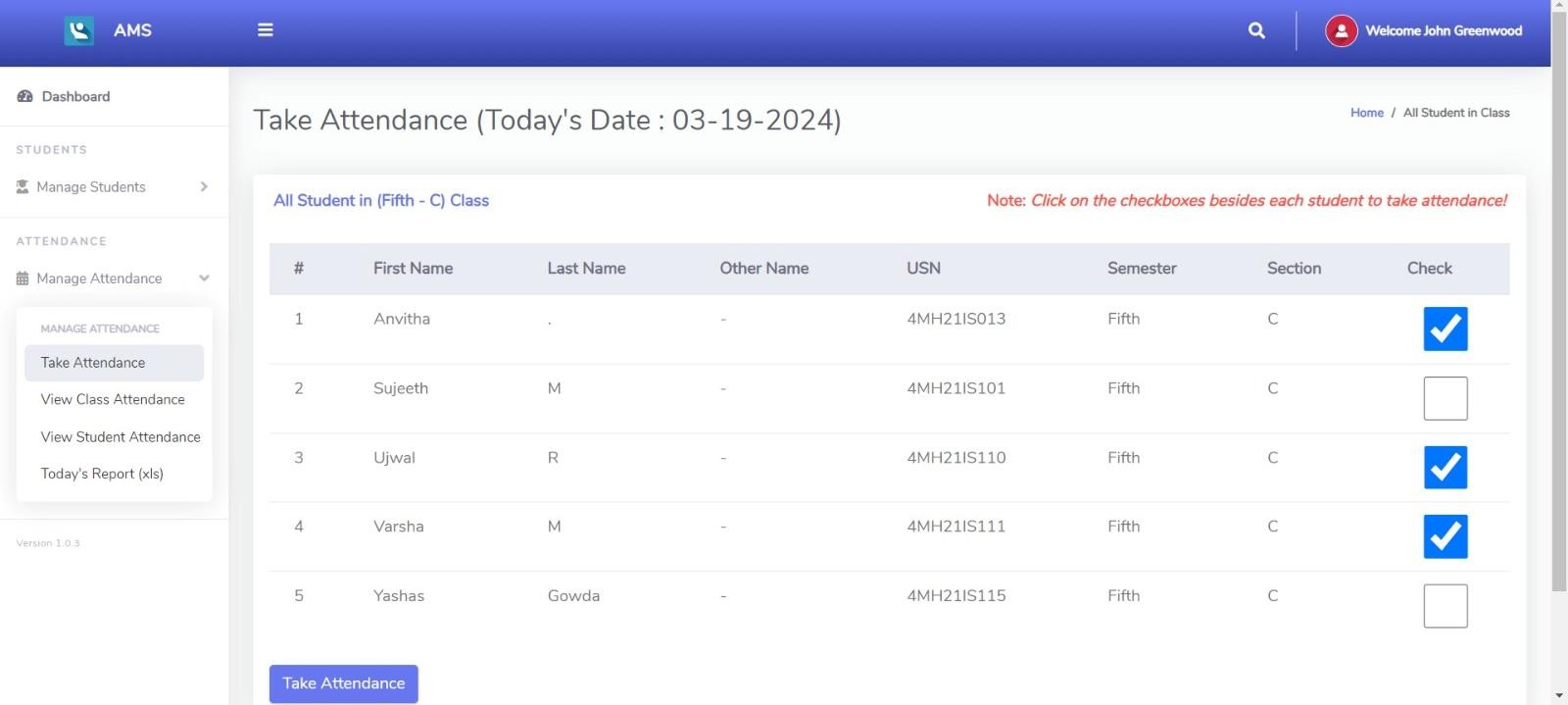
### Fig 6.1.7 Add session and term page

**Fig 6.1.8 Faculty login page**



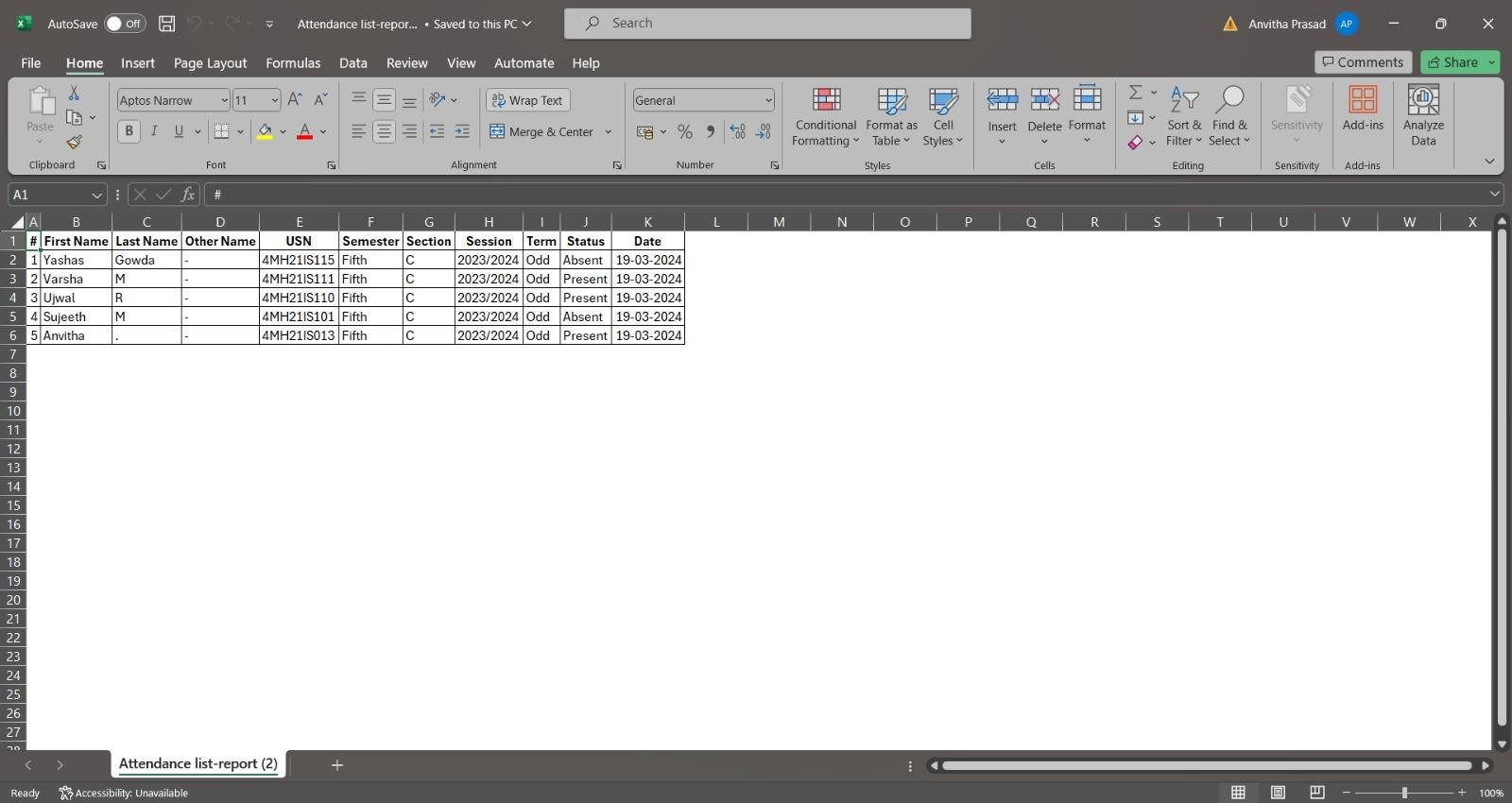
### Fig 6.1.9 Faculty dashboard

**Fig 6.1.10 View students page**



### Fig 6.1.11 Take attendance page

**Fig 6.1.12 View class attendance**



**Fig 6.1.13 Report in excel**

## Conclusion

In conclusion, the student attendance management system project aims to streamline attendance tracking processes for educational institutions. Through our efforts, we've developed a user-friendly system that automates attendance recording, simplifies data management, and facilitates insightful reporting. By implementing this system, educational institutions can save time, improve accuracy, and enhance overall efficiency in attendance management. Moving forward, ongoing maintenance and user feedback will be crucial to ensuring the system's effectiveness and usability.

## Future Work and enhancement

Enhancing the student attendance management system is vital for modern educational institutions to adapt to evolving needs effectively. Future enhancements aim to improve accuracy, efficiency, and user experience. Biometric integration, such as fingerprint or facial recognition, ensures secure and accurate attendance tracking, while geolocation tracking provides real-time verification of attendance, fostering accountability and remote tracking capabilities. Automated attendance reports streamline communication and reduce manual effort, enhancing transparency and providing timely feedback to stakeholders.

Continuous improvement of the student attendance management system is imperative for educational institutions to meet the demands of modern education effectively. Future enhancements such as biometric integration, geolocation tracking, and automated attendance reports offer opportunities to enhance accuracy, efficiency, and user engagement. By implementing these enhancements, institutions can streamline attendance tracking processes, foster accountability, and ultimately contribute to better student outcomes and academic success.

## References

1. **HTML Tutorial - http**[**s://www.w3sc**](http://www.w3schools.com/html)**h**[**ools.com**](http://www.w3schools.com/html)**/h**[**tm**](http://www.w3schools.com/html)**l**
2. **CSS Tutorial** [**- https://www.w3schools.com/Css**](https://www.w3schools.com/Css)
3. **Wikipedia -** [**http://en.wikipedia.org/wiki/PHP for php .**](http://en.wikipedia.org/wiki/PHPforphp)
4. **PHP Tutorial** [**- https://www.w3schools.com/php**](https://www.w3schools.com/php)

#### SQL Tutorial - https://[www.w3schools.com/sq](http://www.w3schools.com/sq)