

# **AMRITSAR GROUP OF COLLEGES**

**Autonomous status conferred by UGC under UGC act-1956, (2f), NAAC-A Grade,  
(Formerly Known as Amritsar College of Engineering & Technology | Amritsar Pharmacy College)**

## **Project Report**

**On**

## **“HOSTEL MANAGEMENT SYSTEM”**

Submitted in the Partial fulfilment of the requirement for the Award of Degree of

## **Bachelor of Technology**

**In**

## **COMPUTER SCIENCE & ENGINEERING**

Batch (2021-25)



### **Submitted To:**

Head of Department  
(Computer Science &  
Engineering)

### **Submitted By:**

Shobhit Arora (2131943)  
Shrinu Shrey (2131944)  
Vicky Kumar (2131969)  
Vikrant (2131971)

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

## **ABSTRACT**

The Hostel Management System is a web-based application designed to streamline the administration of hostel-related operations, offering functionalities for both hostel administrators and students. The system simplifies manual tasks such as room allocation, leave management, and student record tracking, ensuring efficiency and reducing human error. Built using PHP for backend processing, MySQL for database management, and HTML, CSS, JavaScript, and Ajax for a dynamic and user-friendly interface, the system provides real-time access to data and services.

Admins can securely log in to manage student records, assign rooms, monitor room status, and approve or reject leave requests. Students, through their login, can view allocated rooms, request leaves, and update personal details. The integration of Ajax ensures smooth data submission and retrieval without page reloads, enhancing the overall user experience.

The system's deployment using the XAMPP platform offers a robust environment for development and testing. The Hostel Management System significantly reduces administrative workload and provides students with easy access to essential hostel services, thus improving operational efficiency.

## ACKNOWLEDGEMENT

We, the project team, would like to express our sincere gratitude to all those who provided us with the guidance and support necessary for the successful completion of this Hostel Management System project.

First and foremost, we extend our heartfelt thanks to **Er. Tejinder Sharma**, Associate Professor, Department of Computer Science and Engineering, Amritsar Group of Colleges, Amritsar, for her expert guidance, valuable insights, and unwavering support throughout the duration of this project. Her supervision and encouragement have been pivotal to the successful outcome of our work.

We are also deeply grateful to **Dr. Sandeep Kad**, Head of the Department of Computer Science and Engineering, for his support and for providing the resources and facilities necessary to carry out this project. We wish to express our appreciation to the entire faculty of the Department of Computer Science and Engineering, Amritsar Group of Colleges, for their assistance, cooperation, and constructive feedback.

In addition, we acknowledge the contributions of all team members, whose collaboration, dedication, and hard work were crucial in the execution of this project. It has been a privilege to work together and learn from one another throughout this process.

Finally, we are thankful for the opportunity to enhance our knowledge and technical skills in web development, database management, and PHP, all of which will undoubtedly contribute to our future professional growth.

## DECLARATION

We, the undersigned, hereby declare that the project report titled “**Hostel Management System**”, submitted in partial fulfilment of the requirements for the degree of **B. Tech CSE** at **Amritsar Group of Colleges, Amritsar**, is a result of our original work carried out under the supervision **Er. Tejinder Sharma**, Associate Professor, Department of Computer Science and Engineering.

We further certify that:

1. The work presented in this report is original and has been performed by us during the course of our study.
2. This report has not been submitted to any other institution for any degree, diploma, or certification in this college or elsewhere, in India or abroad.
3. We have strictly followed the guidelines provided by the college for the preparation of this report.
4. All materials and sources of data, theoretical analysis, or texts from other works have been properly acknowledged and cited in the report and listed in the bibliography.
5. We take full responsibility for the integrity and accuracy of the content presented in this report.

### Signature of Students:

Shobhit Arora (2131943)

Shrinu Shrey (2131944)

Vicky Kumar (2131969)

Vikrant (2131971)

## INDEX

S. No.	Content	Page no.
01	Abstract	I
02	Acknowledgement	II
03	Declaration	III
04	List of Figures	V
05	List of Tables	VI
06	Introduction to the Project	1
07	Objective of the Project	2
08	Features of the Project	3
09	Technology Used	5-12
10	Tools Used	13-15
11	Hardware & Software Requirement	16
12	Project Modules	17
13	Code	22-31
14	Snapshots	32-55
15	Conclusion	56
16	Future Scope	57
17	Bibliography	58

## LIST OF FIGURES

Figure. No.	Figure Name	Page No.
01	DFD Level 0: Hostel Management System	18
02	Working of Hostel Management System	18
03	DFD Level 1: Admin	19
04	DFD Level 2: Admin	19
05	DFD Level 1: User	20
06	DFD Level 2: User	20
07	UML Diagram	21
08	Admin Login	46
09	Forgot Password	46
10	Admin Dashboard	47
11	Admin Profile	47
12	Hostel Booking 1	48
13	Hostel Booking 2	48
14	Hostel Booking 3	49
15	Add Courses	49
16	Managing Courses	50
17	Add Rooms	50
18	Managing Rooms	51
19	Managing Students	51
20	Managing Leave Requests	52
21	User Login	52
22	User Registration	53
23	User Profile	53
24	Add Details	54
25	Change Password	54
26	Make Leave Request	55
27	Track Leave Requests	55

## LIST OF TABLES

Table No.	Table Name
01	Admin Table
02	Courses Table
03	Leave Requests Table
04	Registration Table
05	Rooms Table
06	States Table
07	User Registration Table

## INTRODUCTION OF THE PROJECT

Hostel management in educational institutions often involves a range of administrative tasks, from allocating rooms to students and managing leave requests to tracking fees and handling complaints. Traditionally, these processes have been managed manually, which not only consumes significant time but also introduces the potential for errors and inefficiencies. To address these challenges, a **Hostel Management System** was developed as a web-based application that automates and streamlines various hostel-related operations.

The primary goal of this project is to create an efficient, user-friendly platform that simplifies hostel management for both administrators and students. Using a combination of **PHP**, **MySQL**, **HTML**, **CSS**, **JavaScript**, and **Ajax**, the system provides a seamless interface for administrators to handle student data, assign rooms, approve or reject leave requests, and monitor room status in real time. Similarly, students can log in securely to view their room allocations, request leaves, and update personal information.

The use of **XAMPP** as the development environment has facilitated efficient system testing and management, providing a local server for handling both the backend and frontend operations. By automating these processes, the system reduces manual work, eliminates paperwork, and enhances the overall operational efficiency of hostel management.

This project not only aims to alleviate the administrative burden but also to offer students easier access to hostel services, contributing to an improved user experience for both parties. The **Hostel Management System** is scalable, allowing for potential future enhancements, including expanded features for multiple hostels or mobile platform integration.



## OBJECTIVE OF THE PROJECT

The objective of the **Hostel Management System** is to develop a comprehensive, web-based platform that automates and simplifies the management of hostel operations for both administrators and students.

The key objectives of this project are:

1. **Streamline Hostel Administration:** To provide administrators with a secure and efficient system to manage student records, assign rooms, track room availability, approve or reject leave requests, and maintain hostel operations digitally.
2. **Enhance Student Experience:** To offer students a user-friendly interface where they can view their room assignments, submit leave requests, update their personal details, and receive notifications about their leave status and room allocation.
3. **Reduce Manual Work:** To minimize the need for manual intervention in managing room allocations, leave requests, and fee tracking, thereby reducing the risk of errors and improving overall efficiency.
4. **Real-time Data Updates:** To enable real-time updates on room availability, leave request statuses, and other relevant data, providing an interactive and responsive experience for both administrators and students.
5. **Improve Data Security and Privacy:** To implement a secure login and authentication system, ensuring that student and administrative data remains protected from unauthorized access.
6. **Provide Scalable Solutions:** To design the system in a way that allows future scalability and enhancements, such as supporting multiple hostels or integrating mobile applications for a wider user base.

By achieving these objectives, the project aims to create a more efficient, user-centric, and automated approach to hostel management, ultimately improving the productivity of hostel administration and the overall experience for students.

# FEATURES OF THE PROJECT

The **Hostel Management System** is designed to automate and optimize various hostel-related operations. Below are the key features that enhance the functionality of the system:

## 1. Admin Login

- Secure login for hostel administrators using password-protected authentication.
- Allows admins to access a dashboard where they can manage all hostel operations efficiently.
- Administrators can oversee student records, room allocations, leave requests, and overall hostel status.

## 2. Student Login

- Secure login portal for students to access their hostel-related information.
- Students can view their allocated rooms, submit leave requests, and update their personal details.
- The system notifies students of any changes made by the admin, such as leave approvals or room reassignments.

## 3. Room Management

- Admins can assign rooms to students and manage room availability in real-time.
- The system maintains a detailed history of room assignments and provides information on whether rooms are occupied or vacant.
- Search functionality and filters allow admins to easily locate available rooms based on specific criteria.

## 4. Leave Management

- Students can submit leave requests, specifying the duration and reason for leave.
- Admins can view, approve, or reject pending leave requests from their dashboard.
- The system notifies students about the approval or rejection of their leave requests and maintains a history of all leave applications.

## **5. Student Information Management**

- Admins can view and manage detailed student profiles, including personal details, room assignments, and leave history.
- Students have the ability to update certain personal information, which is subject to admin approval if necessary.

## **6. Real-time Notifications**

- The system provides real-time updates and notifications to both admins and students regarding important activities, such as room assignments and leave request status.
- Notifications help keep users informed without the need for manual updates or inquiries.

## **7. Search and Filter Functionality**

- The system includes advanced search options and filtering tools to help admins locate students, rooms, or specific requests quickly and efficiently.
- Room availability, student details, and leave requests can be searched based on customizable parameters.

## **8. User-friendly Interface**

- The system's interface is designed with ease of use in mind, ensuring that both administrators and students can navigate and interact with the system effortlessly.
- HTML, CSS, and JavaScript provide a responsive design, while Ajax ensures smooth transitions and real-time updates without the need for page reloads.

## **9. Secure Authentication**

- Both admin and student login portals are secured with password authentication to protect sensitive data.
- Basic authentication protocols are in place to ensure only authorized personnel can access and manage data within the system.

## **10. Scalability and Flexibility**

- The system is built to be scalable, allowing future enhancements such as support for multiple hostels, additional user roles, or mobile integration.
- The flexible backend architecture allows for easy upgrades and additional features without major reconfiguration.

## TECHNOLOGY USED

### HTML (Hyper Text Markup Language)



**HTML** stands for Hyper Text Markup Language. It is used to design web pages using a markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between web pages. A markup language is used to define the text document within the tag which defines the structure of web pages. This language is used to annotate (make notes for the computer) text so that a machine can understand it and manipulate text accordingly. Most markup languages (e.g. HTML) are human-readable.

#### Features of HTML

The basic structure of an HTML page is laid out below. It contains the essential building-block elements (i.e. doctype declaration, HTML, head, title, and body elements) upon which all web pages are created.

#### Separation of Content and Structure:

- HTML allows web developers to structure the content of a web page, organizing it into elements like headings, paragraphs, and lists, independent of the page's visual design. This separation enhances the clarity and maintainability of the code.

#### Hyperlinks and Navigation:

- HTML enables the creation of hyperlinks (<a> tag), which allow users to navigate between web pages or different sections of a single page, providing connectivity and navigation across websites.

#### Multimedia Integration:

- HTML supports embedding multimedia content such as images, audio, and video. Tags like <img>, <audio>, and <video> allow developers to incorporate media for a richer user experience.

#### Compatibility with CSS and JavaScript:

- HTML works seamlessly with CSS for styling and JavaScript for functionality. CSS enhances the appearance, while JavaScript adds interactivity, enabling dynamic web pages.

## CSS (Cascading Style Sheets)



Cascading Style Sheets, fondly referred to as CSS, is a simply designed language intended to simplify the process of making web pages presentable. CSS allows you to apply styles to web pages. More importantly, CSS enables you to do this independent of the HTML that makes up each web page. It describes how a webpage should look: it prescribes colors, fonts, spacing, and much more.

In short, you can make your website look however you want. CSS lets developers and designers define how it behaves, including how elements are positioned in the browser.

### Features of CSS

CSS, or Cascading Style Sheets, is a language used for describing the presentation and layout of web pages written in HTML. Some of the key features of CSS include:

#### Separation of content and presentation:

- CSS allows web developers to separate the content and structure of a web page from its visual appearance and layout. This makes it easier to create consistent and flexible web pages that can be easily updated and maintained.

#### Selective styling:

- CSS allows web developers to apply styles selectively to specific HTML elements, classes, or IDs, making it easy to style individual elements or groups of elements on a web page.

#### Cascading styles:

- CSS uses a cascading style sheet model, meaning that multiple styles can be applied to the same element, with the most specific style taking precedence. This allows web developers to create complex and layered styles that can be easily customized and updated.

#### Responsive design:

- CSS includes advanced features such as media queries, which allow web developers to create responsive designs that adapt to different screen sizes and devices, making it easy to create web pages that look great on both desktop and mobile devices.

#### Animations and transitions:

- CSS includes features for creating animations and transitions, allowing web developers to create dynamic and engaging web pages that can capture the user's attention and improve the user experience.

# JAVASCRIPT

## JavaScript



JavaScript is a high-level, interpreted programming language that is widely used for building interactive web pages and web applications. It was originally created by Brendan Eich in just ten days in 1995 and was initially called Mocha, then later renamed to Live Script, before finally being called JavaScript.

JavaScript is a client-side scripting language, which means it runs in the user's web browser rather than on the server. It is often used in conjunction with HTML and CSS to add interactivity to web pages and create dynamic effects. It can be used on the server-side with Node.js to build server-side applications.

One of the key features of JavaScript is its ability to manipulate the Document Object Model (DOM) of a web page. This allows developers to create interactive user interfaces and dynamic effects such as animations and pop-up windows.

### Features of JavaScript

JavaScript has many features that make it a popular language for building web applications. Here are some of its key features:

#### Object-Oriented Programming (OOP):

- JavaScript is an object-oriented language that allows developers to define custom objects with their own properties and methods.

#### Dynamic Typing:

- JavaScript is a dynamically typed language, which means that variables can change their data type at runtime. This makes it easy to write flexible code that can adapt to different situations.

#### Interactivity:

- JavaScript is designed to create interactive user interfaces and to respond to user actions such as button clicks and keyboard input.

#### Event-Driven Programming:

- JavaScript allows developers to define functions that are triggered by events such as mouse clicks or keyboard presses.

#### Expressiveness:

- JavaScript has a concise syntax that allows developers to write powerful code in fewer lines.

## DOM (Document Object Model)



The Document Object Model (DOM) is a programming interface for HTML and XML documents. It represents the structure of a web page as a hierarchical tree of objects, where each object corresponds to a different part of the document, such as the document itself, an HTML element, or a text node.

The DOM provides a way for developers to manipulate the content and style of a web page using JavaScript. By accessing and modifying the properties of DOM elements, developers can dynamically update the content of a web page without having to reload the entire page.

The DOM is a cross-platform and language-independent interface, which means that it can be used with any programming language that supports the W3C Document Object Model standard.

### Features of DOM

The Document Object Model (DOM) is a powerful tool for web developers that provides a programming interface for HTML and XML documents. Here are some key features of the DOM:

#### Hierarchical structure:

- The DOM represents the structure of a web page as a hierarchical tree of objects. Each object in the tree represents a different part of the document, such as an HTML element, or a text node.

#### Platform independence:

- The DOM is platform-independent and can be used with any programming language that supports the W3C Document Object Model standard.

#### Dynamic content modification:

- The DOM provides a way to dynamically modify the content of a web page without having to reload the entire page. This allows developers to create dynamic and interactive web applications.

#### Event handling:

- The DOM provides a way to add event listeners to HTML elements to respond to user interactions, such as button clicks or form submissions.

#### Cross-browser compatibility:

- The DOM is designed to be compatible with all modern web browsers, allowing developers to create web applications that work consistently across different platforms.

## AJAX (Asynchronous JavaScript and XML)



Asynchronous JavaScript and XML (AJAX) is a technique used in web development to create dynamic web pages. It allows web pages to be updated asynchronously by exchanging small amounts of data with the server, rather than reloading the entire page.

With AJAX, web pages can be updated in real-time without interrupting the user's interaction with the page. This technique enables developers to create faster and more responsive web applications.

### Features of AJAX

Asynchronous JavaScript and XML (AJAX) is a web development technique that allows web pages to be updated asynchronously by exchanging small amounts of data with the server, without having to reload the entire page. Here are some key features of AJAX:

#### Asynchronous data transfer:

- AJAX allows data to be transferred between the client and server asynchronously, which means that the web page can continue to be used while data is being sent or received.

#### Partial page updates:

- With AJAX, it is possible to update only a part of the web page, rather than having to reload the entire page. This can lead to a more responsive and faster user experience.

#### Reduced server load:

- By only sending and receiving small amounts of data, AJAX can help reduce the load on the server, leading to faster response times.

#### Cross-platform compatibility:

- AJAX is compatible with all modern web browsers and can be used with any programming language that supports the W3C XMLHttpRequest (XHR) object.

#### User interactivity:

- AJAX allows developers to create web applications that are more interactive, by enabling real-time updates and responding to user input without having to reload the entire page.

#### Supports multiple data formats:

- AJAX supports multiple data formats, such as XML, JSON, and HTML, making it flexible and adaptable to a wide range of web applications.



## PHP (Hypertext Preprocessor)



PHP (Hypertext Preprocessor) is a server-side scripting language designed for web development. It is widely used for building dynamic web pages, web applications, and content management systems (CMS).

PHP was first released in 1995 by Rasmus Lerdorf, and has since become one of the most popular programming languages for web development. PHP is an open-source language and is supported by a large community of developers, which has led to the creation of many powerful frameworks and libraries.

### Features of PHP

Some key features of PHP include:

#### Server-side scripting:

- PHP is a server-side language, which means that it runs on the server and generates HTML output, which is then sent to the client's browser.

#### Database integration:

- PHP has built-in support for connecting to databases, including MySQL, PostgreSQL, and Oracle, making it easy to build web applications that store and retrieve data.

#### Object-oriented programming (OOP):

- PHP supports object-oriented programming, which allows developers to organize code into reusable and modular components.

#### Cross-platform compatibility:

- PHP can run on a variety of operating systems, making it a flexible and versatile language for web development.

#### Large community and ecosystem:

- PHP has a large and active community of developers, which has led to the creation of many powerful frameworks and libraries, such as Laravel, Symfony, and CodeIgniter.

# XAMPP



# XAMPP

XAMPP is a free and open-source cross-platform web server solution that includes Apache HTTP Server, MariaDB database server, and PHP and Perl interpreters. The name "XAMPP" stands for Cross-Platform, Apache, MariaDB, PHP and Perl.

XAMPP is designed to make it easy for developers to set up a local web server environment for testing and development purposes. It is available for Windows, Linux, and macOS, and can be downloaded and installed in just a few minutes.

Overall, XAMPP is a powerful and easy-to-use web server solution that is ideal for developers who want to test and develop web applications on a local machine before deploying to a production server.

## Features of XAMPP Server

Here are some of the key features of the XAMPP server:

### Easy to install and use:

- XAMPP is easy to install and configure, making it ideal for developers who are new to web development.

### Cross-platform support:

- XAMPP is available for Windows, Linux, and macOS, so you can use it on any platform of your choice.

### Apache web server:

- XAMPP includes the Apache HTTP server, which is a popular web server to serve web pages.

### MariaDB database server:

- XAMPP includes the MariaDB database server, which is a popular open-source relational database management system.

### PHP and Perl interpreters:

- XAMPP includes the PHP and Perl interpreters, which are used for server-side scripting.

### phpMyAdmin:

- XAMPP includes phpMyAdmin, a popular web-based tool for managing MySQL/MariaDB databases.

# MYSQL



**MySQL** is a popular open-source relational database management system (RDBMS) that is widely used for building scalable and robust web applications. It is developed, distributed, and supported by Oracle Corporation.

MySQL is a client-server database system, which means that the database runs on a server and clients can access it from remote locations over a network. It supports a variety of programming languages such as PHP, Python, Java, and C++, and provides drivers for each of these languages to connect to the database.

## Key Features of MySQL:

1. **Relational database:** MySQL is based on the relational database model, which means that it organizes data into tables with rows and columns, and uses structured query language (SQL) to manipulate and retrieve data.
2. **Scalability and performance:** MySQL is designed to handle large amounts of data and high traffic loads, making it suitable for web applications with millions of users.
3. **Security:** MySQL provides a range of security features such as user authentication, encryption, and access control to protect the database and its data from unauthorized access.
4. **Cross-platform support:** MySQL runs on multiple platforms including Windows, Linux, and macOS.
5. **Customizability:** MySQL is highly customizable and provides a range of configuration options to optimize its performance and functionality.

Overall, MySQL is a powerful and flexible RDBMS that provides a reliable and efficient way to store, manage, and retrieve data for web applications. It is widely used by businesses of all sizes, from small startups to large enterprises.

## TOOLS USED

### 1. VISUAL STUDIO CODE:

Visual Studio Code (often abbreviated as VS Code) is a free, open-source code editor developed by Microsoft. It's designed to be lightweight yet powerful, offering features and tools that support development across various programming languages and frameworks. Available on Windows, macOS, and Linux, VS Code is widely used by developers for its versatility, speed, and extensive customization options.



#### Key Features of VS Code:

##### 1. Multi-Language Support:

- VS Code supports a wide range of programming languages by default, including JavaScript, Python, C++, PHP, HTML, and CSS.

##### 2. IntelliSense:

- IntelliSense provides intelligent code completions, parameter hints, and inline suggestions based on the context of the code.

##### 3. Integrated Terminal:

- The built-in terminal allows developers to execute command-line tasks directly within the editor, supporting shells like Bash, Command Prompt, and PowerShell.

##### 4. Debugging Tools:

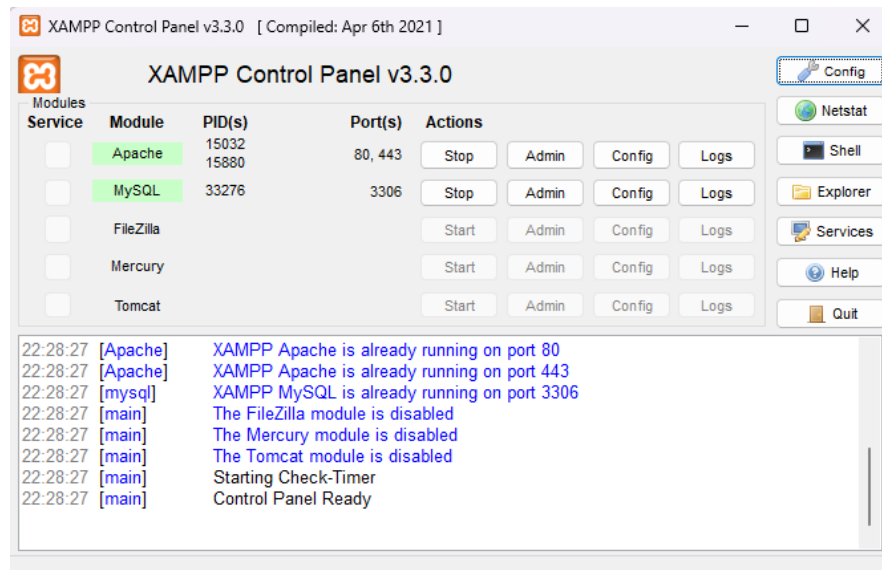
- VS Code has integrated debugging capabilities for a range of languages, allowing you to set breakpoints, step through code, and inspect variables directly in the editor.

##### 5. Extensions and Customization:

- The VS Code marketplace offers thousands of extensions, including themes, language packs, linters, and more, allowing extensive customization.

## 2. XAMPP SERVER

**XAMPP** is an open-source, cross-platform web server solution package that provides a convenient environment for developers to create and test web applications locally. It stands for **X** (cross-platform), **A**pache (web server), **M**ySQL (database), **P**HP (scripting language), and **P**erl. XAMPP is widely used for local development because it packages all the essential tools needed to run a server.



### Key Features of XAMPP:

#### 1. Cross-Platform Compatibility:

- XAMPP is available for multiple platforms, including **Windows**, **Linux**, and **macOS**, making it versatile for developers working across different operating systems.

#### 2. Apache Web Server:

- XAMPP includes the **Apache HTTP Server**, one of the most popular web servers, enabling developers to host websites and web applications locally and simulate how they would function on a real server.

#### 3. MySQL/MariaDB Database:

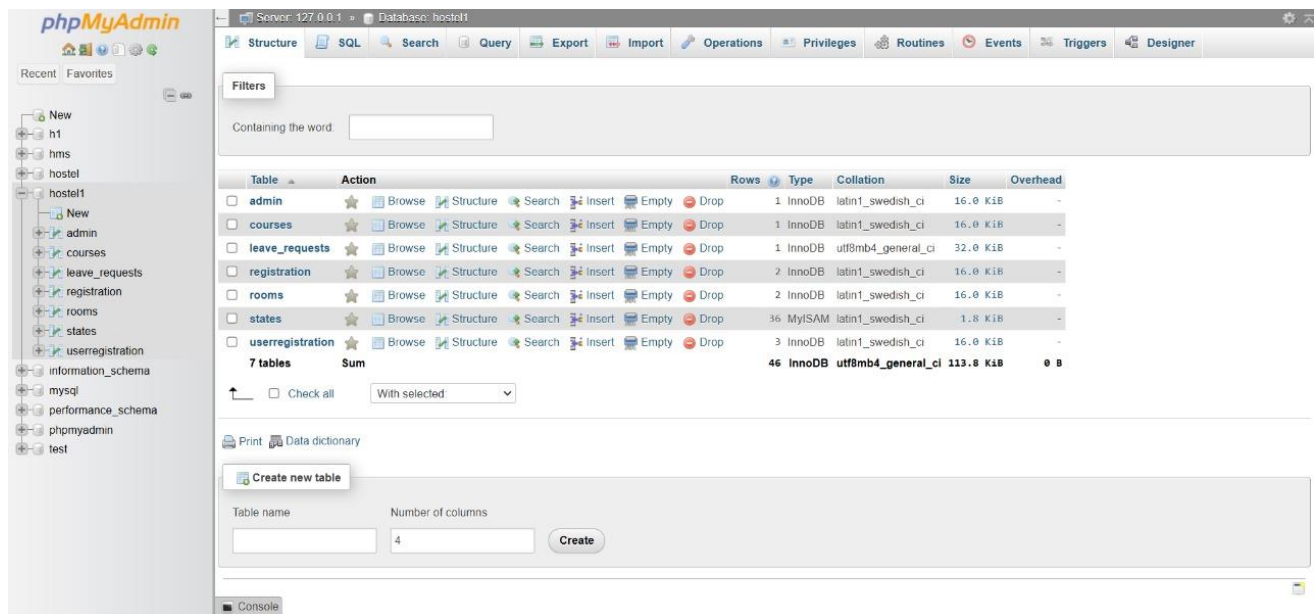
- XAMPP comes with **MySQL** (or **MariaDB**, which is the default in newer versions), a relational database management system used to store and manage data in web applications.

#### 4. phpMyAdmin:

- XAMPP integrates **phpMyAdmin**, a web-based tool to manage MySQL databases. This tool provides a graphical interface to perform tasks like creating databases, tables, running SQL queries & managing user permissions without needing to interact with MySQL command line.

### 3. PHP MY ADMIN

**phpMyAdmin** is a free, open-source web-based tool used for managing **MySQL** and **MariaDB** databases. It provides an intuitive graphical user interface (GUI) for database administration tasks such as creating databases, running queries, and managing users. phpMyAdmin simplifies database management without requiring deep knowledge of SQL or command-line operations, making it ideal for both beginners and advanced users.



#### Key Features of phpMyAdmin:

##### 1. Web-Based Interface:

- Allows users to manage MySQL databases through a simple and intuitive web browser interface.

##### 2. Database and Table Management:

- Easily create, modify, and delete databases and tables, including managing table structures (adding or removing columns).

##### 3. SQL Query Execution:

- Run custom SQL queries, and phpMyAdmin highlights SQL syntax to assist in writing and debugging queries.

##### 4. Data Import/Export:

- Supports importing and exporting data in formats such as **CSV**, **SQL**, and **XML**, which is useful for backups and database migrations.

# HARDWARE AND SOFTWARE REQUIREMENTS

## Hardware Requirements:

- **Processor:** Intel Core i3 or equivalent.
- **RAM:** 4GB or more.
- **Storage:** 50GB free disk space.
- **Network:** Stable internet connection.
- **Display:** 15-inch or larger monitor.
- **Backup:** External or cloud storage for data backup.

## Software Requirements:

- **OS:** Windows 7/8/10, Linux (Ubuntu), or macOS.
- **Web Server:** XAMPP (Apache, MySQL, PHP).
- **Database:** MySQL.
- **Backend:** PHP 7.0+.
- **Frontend:** HTML5, CSS3, JavaScript, Ajax.
- **Browser:** Google Chrome, Brave, or Microsoft Edge.
- **Development Tools:** Visual Studio Code or Sublime Text.
- **Optional:** Git, phpMyAdmin, Postman.

# PROJECT MODULES

## 1. Admin Login:

- Provides a secure login page for the hostel administrator.
- Upon login, admins can access student records, assign rooms, track room status, and approve or deny leave requests.
- Password management and basic authentication protocols are implemented to ensure data privacy.

## 2. Student Login:

- Secure login for students, allowing them to view their allocated rooms, request leaves, and check personal details.
- Students can update their profiles and monitor any updates made by the administrator related to room assignments or leave status.

## 3. Room Management:

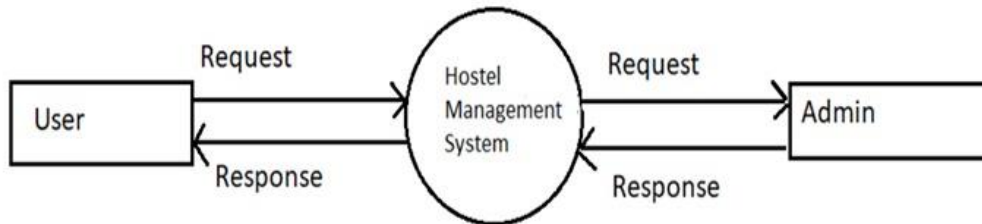
- Admins can assign, update, and view room allocations.
- Room status is shown in real-time, indicating whether a room is vacant or occupied, along with a history of room assignments.
- Additional features include room availability filters and search functionality.

## 4. Leave Module:

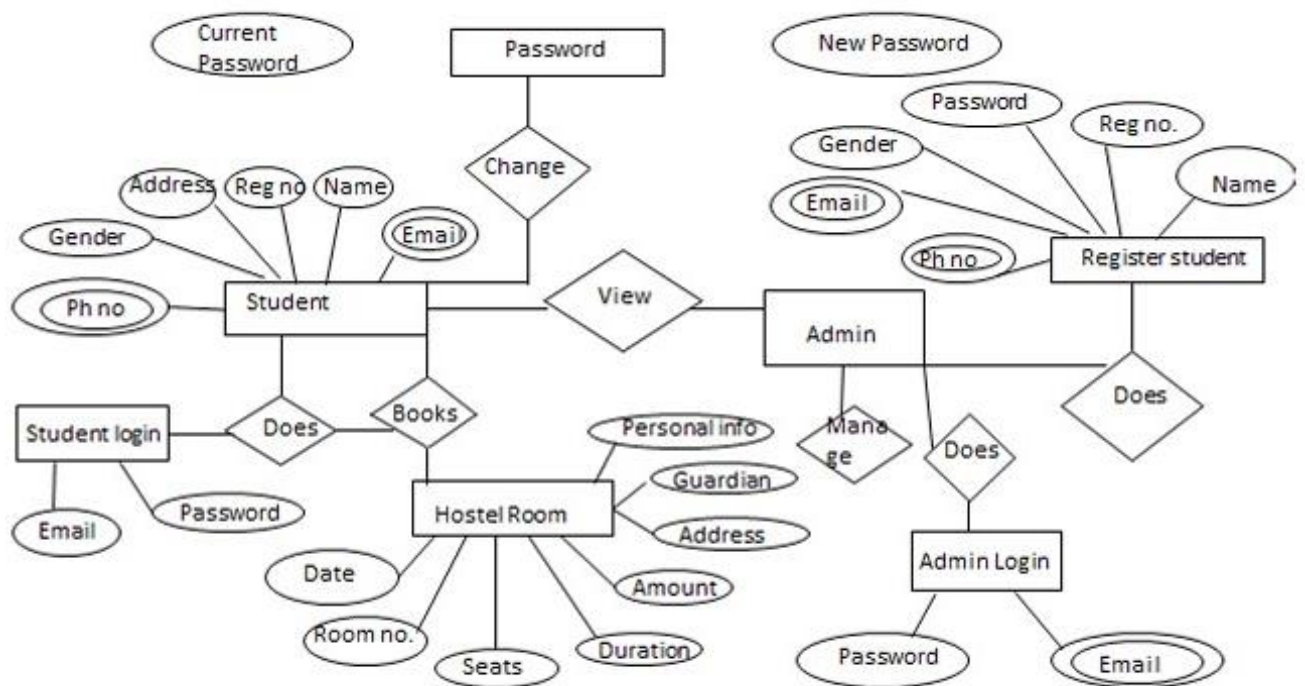
- Enables students to submit leave requests, specifying the duration and reason for leave.
- Admins can view pending leave requests and approve or reject them.
- Students receive notifications of leave approvals or denials, and the leave history is maintained.



## DFD Diagrams

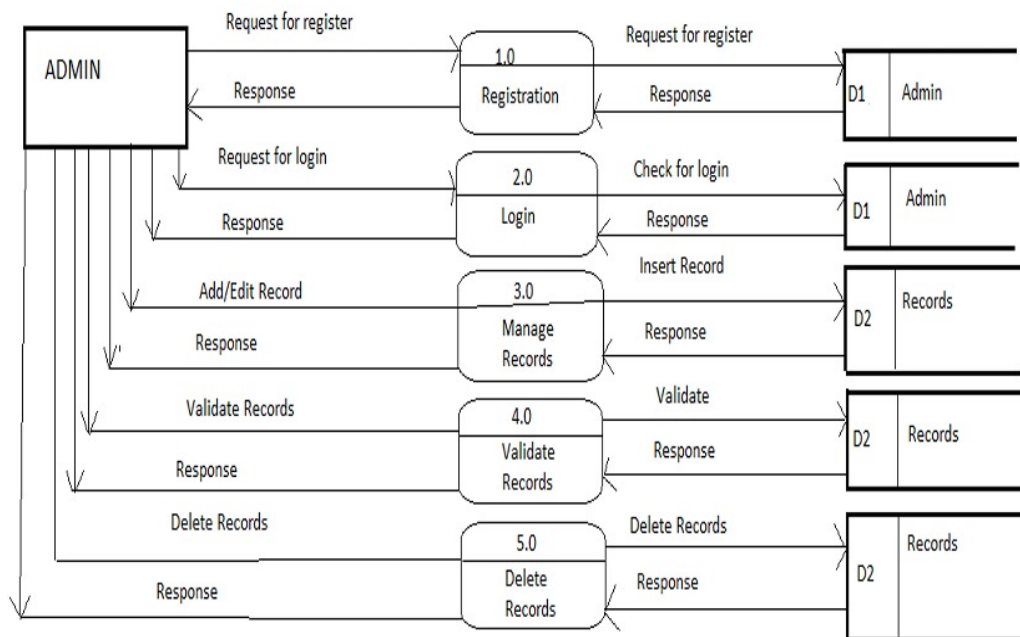


**Fig No. 1 : DFD Level 0: Hostel Management System**



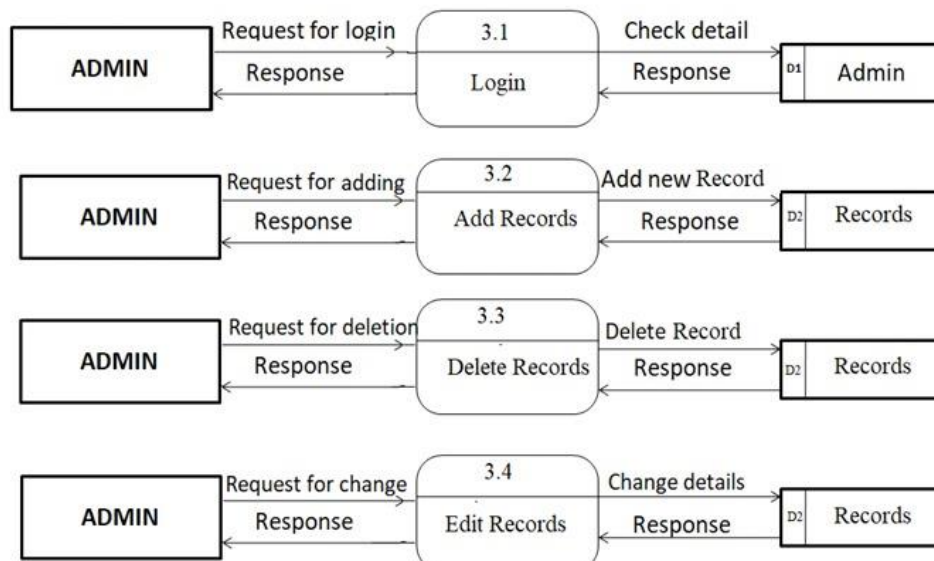
**Fig No. 2 : Working of Hostel Management System**

## 1. Admin Login:



**Fig No. 3: DFD Level 1: Admin**

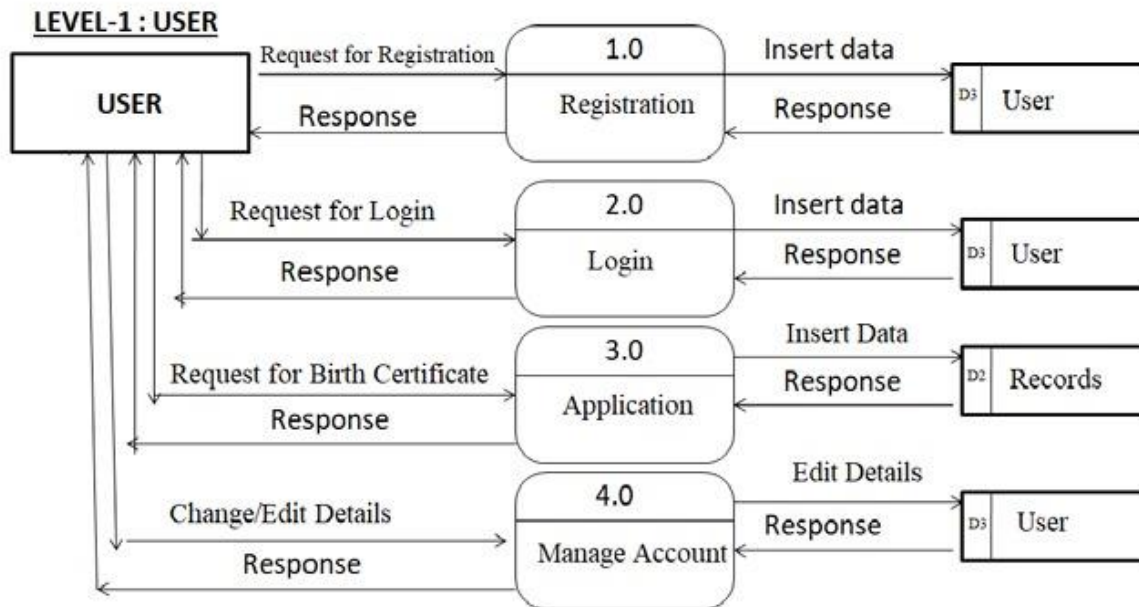
### LEVEL – 2: ADMIN



**Fig No. 4: DFD Level 2: Admin**

## 2. User Login:

Fig No. 5: DFD Level 1: User



### LEVEL – 2: USER

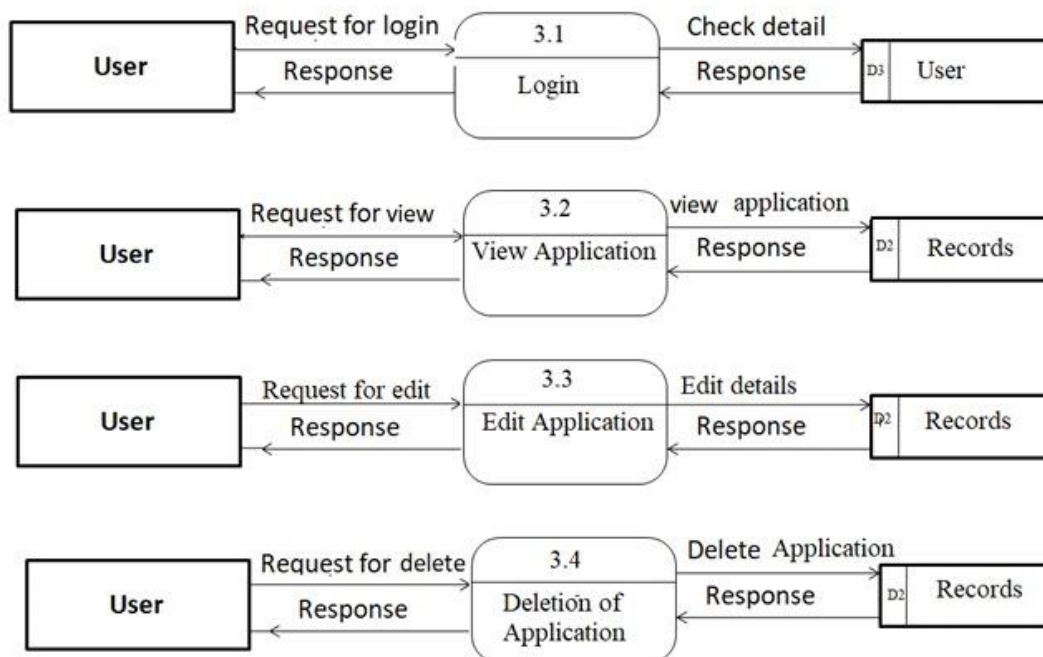


Fig No. 6: DFD Level 2: User

## UML Diagram

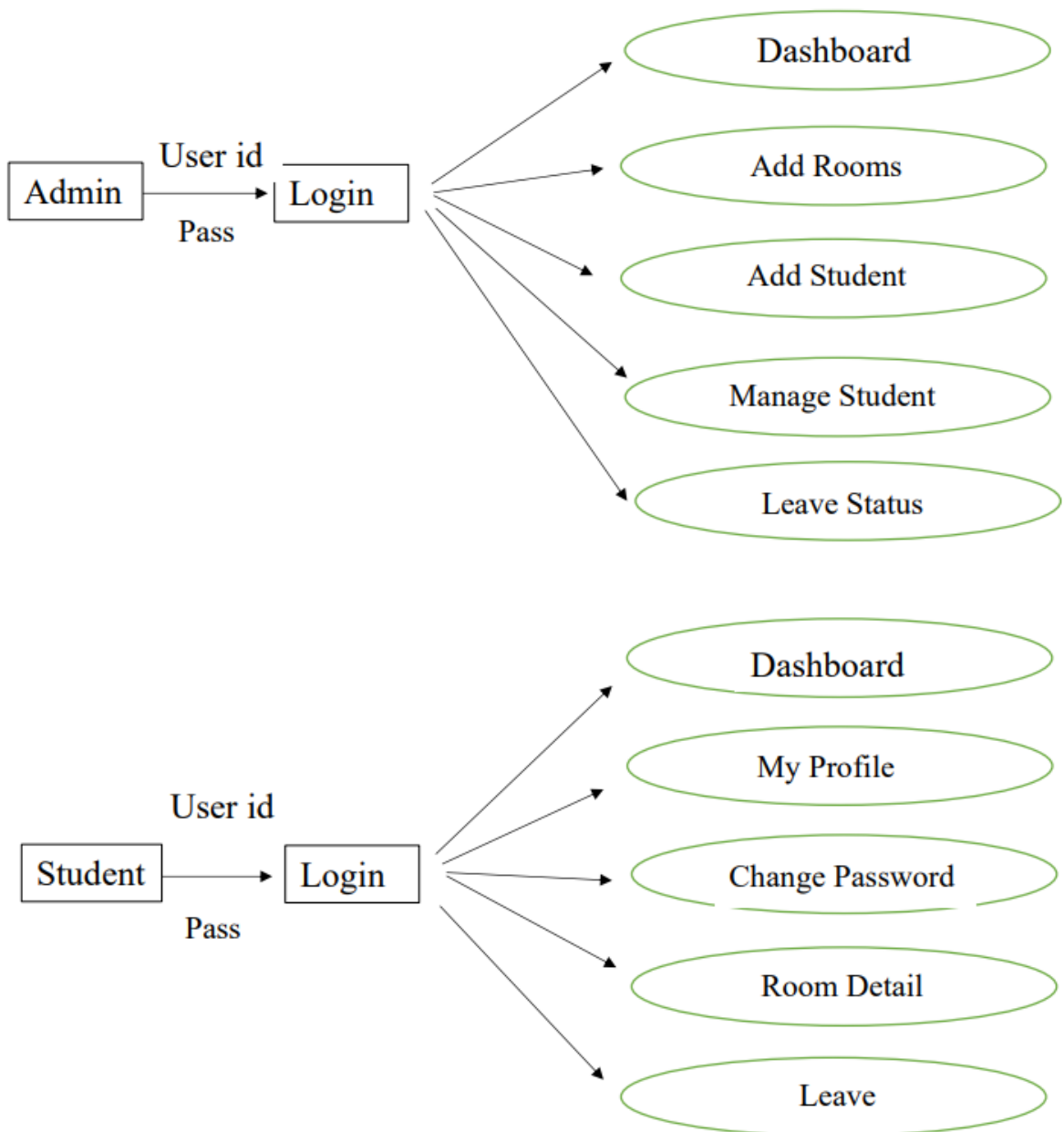


Fig No. 7: UML Diagram

## CODE

### Dashboard.php

```
<?php
session_start();
include('includes/config.php');
include('includes/checklogin.php');
check_login();
?>

<!doctype html>

<html lang="en" class="no-js">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1, minimum-scale=1,
maximum-scale=1">

    <meta name="theme-color" content="#3e454c">

    <title>Dashboard</title>

    <link rel="stylesheet" href="css/font-awesome.min.css">

    <link rel="stylesheet" href="css/bootstrap.min.css">

    <link rel="stylesheet" href="css/dataTables.bootstrap.min.css">

    <link rel="stylesheet" href="css/bootstrap-social.css">

    <link rel="stylesheet" href="css/bootstrap-select.css">

    <link rel="stylesheet" href="css/fileinput.min.css">

    <link rel="stylesheet" href="css/awesome-bootstrap-checkbox.css">

    <link rel="stylesheet" href="css/style.css">

</head>

<body>

    <?php include("includes/header.php"); ?>
```

```

<div class="ts-main-content">
  <?php include("includes/sidebar.php"); ?>
  <div class="content-wrapper">
    <div class="container-fluid">
      <div class="row">
        <div class="col-md-12">
          <h2 class="page-title" style="margin-top: 45px;">Dashboard</h2>
          <div class="row">
            <div class="col-md-12">
              <div class="row">
                <div class="col-md-3">
                  <div class="panel panel-default">
                    <div class="panel-body bk-primary text-light">
                      <div class="stat-panel text-center">
                        <div class="stat-panel-number h1">My Profile</div>
                      </div>
                    </div>
                    <a href="my-profile.php" class="block-anchor panel-footer">Full Detail
<i
          All &nbsp;   <i class="fa fa-arrow-right"></i></a>
        </div>
      </div>
    </div>

    var doctx = document.getElementById("chart-area3").getContext("2d");
    window.myDoughnut = new Chart(doctx).Pie(doughnutData, { responsive: true });

    // Doughnut Chart from doughnutData
    var doctx = document.getElementById("chart-area4").getContext("2d");
    window.myDoughnut = new Chart(doctx).Doughnut(doughnutData, { responsive: true });
  }
</script>

</body>
</html>

```

### **Login.php :**

```
<?php
session_start();
include('includes/config.php');
if (isset($_POST['login'])) {
    $email = $_POST['email'];
    $password = $_POST['password'];

    $stmt = $mysqli->prepare("SELECT email,password,id FROM userregistration WHERE email=?
and password=? ");

    $stmt->bind_param('ss', $email, $password);
    $stmt->execute();
    $stmt->bind_result($email, $password, $id);
    $rs = $stmt->fetch();
    $stmt->close();
    $_SESSION['id'] = $id;
    $_SESSION['login'] = $email;
    $uip = $_SERVER['REMOTE_ADDR'];
    $ldate = date('d/m/Y h:i:s', time());
    if ($rs) {
        $uid = $_SESSION['id'];
        $uemail = $_SESSION['login'];
        $ip = $_SERVER['REMOTE_ADDR'];
        $geopluginURL = 'http://www.geoplugin.net/php.gp?ip=' . $ip;
        $addrDetailsArr = unserialize(file_get_contents($geopluginURL));
        $city = $addrDetailsArr['geoplugin_city'];
        $country = $addrDetailsArr['geoplugin_countryName'];
        $log = "insert into userLog(userId,userEmail,userIp,city,country)
values('$uid','$uemail','$ip','$city','$country')";
        $mysqli->query($log);
        if ($log) {
            header("location:dashboard.php");
        }
    } else {
```

```

        echo "<script>alert('Invalid Username/Email or password');</script>";
    }
}
?>

<!doctype html>
<html lang="en" class="no-js">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1, minimum-scale=1, maximum-scale=1">
    <meta name="description" content="">
    <meta name="author" content="">
    <meta name="theme-color" content="#3e454c">
    <title>Student Hostel Registration</title>
    <link rel="stylesheet" href="css/font-awesome.min.css">
    <link rel="stylesheet" href="css/bootstrap.min.css">
    <link rel="stylesheet" href="css/dataTables.bootstrap.min.css">>
    <link rel="stylesheet" href="css/bootstrap-social.css">
    <link rel="stylesheet" href="css/bootstrap-select.css">
    <link rel="stylesheet" href="css/fileinput.min.css">
    <link rel="stylesheet" href="css/awesome-bootstrap-checkbox.css">
    <link rel="stylesheet" href="css/style.css">
    <script type="text/javascript" src="js/jquery-1.11.3-jquery.min.js"></script>
    <script type="text/javascript" src="js/validation.min.js"></script>
    <script type="text/javascript" src="http://code.jquery.com/jquery.min.js"></script>
    <script type="text/javascript">
        function valid() {
            if (document.registration.password.value != document.registration.cpassword.value) {
                alert("Password and Re-Type Password Field do not match !!");
                document.registration.cpassword.focus();
            }
        }
    }
}
?>

```



[illegible]

## **Index.php**

```
<?php
session_start();
include('includes/config.php');

if (isset($_POST['login'])) {
    $email = $_POST['email'];
    $password = $_POST['password'];

    // Prepare and execute the statement
    $stmt = $mysqli->prepare("SELECT email, password, user_id FROM userregistration WHERE
email = ? AND password = ?");

    $stmt->bind_param('ss', $email, $password);
    $stmt->execute();

    // Bind the result variables
    $stmt->bind_result($db_email, $db_password, $id);
    $rs = $stmt->fetch();
    $stmt->close();

    if ($rs) {
        // Set session variables after successful login
        $_SESSION['id'] = $id; // Store user ID in session
        $_SESSION['login'] = $db_email; // Store user email in session

        // Debug output
        echo "Logged in successfully. User ID: " . $_SESSION['id'] . ", Email: " . $_SESSION['login'];

        $uip = $_SERVER['REMOTE_ADDR'];
        $ldate = date('d/m/Y h:i:s', time());

        // Log user login details
```

</head>

<body>

<?php include('includes/header.php'); ?>

<div class="ts-main-content">

<?php include('includes/sidebar.php'); ?>

<div class="content-wrapper">

<div class="container-fluid">

<div class="row">

<div class="col-md-12">

<h2 class="page-title" style="color:white;">User Login</h2>

<div class="row">

<div class="col-md-6 col-md-offset-3">

<div class="login-form">

<div class="col-md-8 col-md-offset-2">

<form action="" class="mt" method="post">

<label for="" class="text-uppercase text-sm"  
style="color:white;">Email</label>

<input type="text" placeholder="Email" name="email" class="form-control mb">

<label for="" class="text-uppercase text-sm"  
style="color:white;">Password</label>

<input type="password" placeholder="Password" name="password"  
class="form-control mb">

<button class="btn btn-primary btn-block" type="submit" name="login"  
value="Login">Login</button>

</form>

<div class="text-center text-light" style="color:white; margin-top: 17px;">

<a href="forgot-password.php" style="color:white;">Forgot  
password?</a>

</div>

</div>

</div>

```
        </div>
    </div>
</div>
</div>
</div>
</div>
</div>
</div>
<script src="js/jquery.min.js"></script>
<script src="js/bootstrap-select.min.js"></script>
<script src="js/bootstrap.min.js"></script>
<script src="js/jquery.dataTables.min.js"></script>
<script src="js/dataTables.bootstrap.min.js"></script>
<script src="js/Chart.min.js"></script>
<script src="js/fileinput.js"></script>
<script src="js/chartData.js"></script>
<script src="js/main.js"></script>
</body>

</html>
```

### **Change\_password.php :**

```
<?php
session_start();
include('includes/config.php');
date_default_timezone_set('Asia/Kolkata');
include('includes/checklogin.php');
check_login();

$ai = $_SESSION['id'];

if (isset($_POST['changepwd'])) {
    $op = $_POST['oldpassword'];
    $np = $_POST['newpassword'];
    $udate = date('d-m-Y h:i:s', time());

    $sql = "SELECT password FROM userregistration WHERE password = ?";
    $chngpwd = $mysqli->prepare($sql);
    $chngpwd->bind_param('s', $op);
    $chngpwd->execute();
    $chngpwd->store_result();

    if ($chngpwd->num_rows > 0) {
        $con = "UPDATE userregistration SET password = ?, passUpdateDate = ? WHERE user_id = ?";
        $chngpwd1 = $mysqli->prepare($con);
        $chngpwd1->bind_param('ssi', $np, $udate, $ai);
        $chngpwd1->execute();
        $_SESSION['msg'] = "Password Changed Successfully!";
    } else {
        $_SESSION['msg'] = "Old Password does not match!";
    }
}
?>
```

```

<body>
    <?php include('includes/header.php'); ?>
    <script src="js/bootstrap-select.min.js"></script>
    <script src="js/bootstrap.min.js"></script>
    <script src="js/jquery.dataTables.min.js"></script>
    <script src="js/dataTables.bootstrap.min.js"></script>
    <script src="js/Chart.min.js"></script>
    <script src="js/fileinput.js"></script>
    <script src="js/chartData.js"></script>
    <script src="js/main.js"></script>
    <script>
        function checkpass() {
            $("#loaderIcon").show();
            jQuery.ajax({
                url: "check_availability.php",
                data: 'oldpassword=' + $("#oldpassword").val(),
                type: "POST",
                success: function (data) {
                    $("#password-availability-status").html(data);
                    $("#loaderIcon").hide();
                },
                error: function () { }
            });
        }
    </script>
</body>
</html>

```

### **Full profile.php**

```
<?php
session_start();

$mysql_hostname = "localhost";
$mysql_user = "root";
$mysql_password = "";
$mysql_database = "hostel1";

// Establish connection using mysqli
$bd = new mysqli($mysql_hostname, $mysql_user, $mysql_password, $mysql_database);

// Check for connection error
if ($bd->connect_error) {
    die("Could not connect database: " . $bd->connect_error);
}
?>

<script language="javascript" type="text/javascript">
    function f2() {
        window.close();
    }
    function f3() {
        window.print();
    }
</script>

<!DOCTYPE html

    PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

<head>

    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />

    <title>Student Information</title>
```

```
<link href="style.css" rel="stylesheet" type="text/css" />

<link href="hostel.css" rel="stylesheet" type="text/css">

</head>


<body>

<table width="100%" border="0">

    <?php

        // Fetch data using prepared statements to avoid SQL injection

        $stmt = $bd->prepare("SELECT * FROM registration WHERE emailid = ?");

        $stmt->bind_param("s", $_GET['id']);

        $stmt->execute();

        $result = $stmt->get_result();

        <tr>

            <td colspan="2" class="font"><?php echo ucfirst($row['firstName']); ?> <?php echo
ucfirst($row['lastName']); ?>'S

                <span class="font1"> information &raquo;</span> </td>

            </tr>

            <tr>

                <td colspan="2"
class="font">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~

                    <div align="right">Reg Date : <span class="comb-value"><?php echo $row['postingDate'];
?></span></div>

                </td>

            </tr>

            <tr>

                <td colspan="2" class="heading" style="color: red;">Room Related Info &raquo;</td>

            </tr>

            <tr>

                <td colspan="2" class="font1">

                    <table width="100%" border="0">

                        <tr>

                            <td width="32%" valign="top" class="heading">Room no : </td>
```



```

        <td class="comb-value1"><span class="comb-value"><?php echo $row['seater'];
?></span></td>

    </tr>

<tr>

    <td colspan="2" align="right">

        <form id="form1" name="form1" method="post" action="">

            <table width="100%" border="0" cellpadding="0" cellspacing="0">

                <tr>

                    <td width="14%">&nbsp;</td>

                    <td width="35%" class="comb-value"><label>

                        <input name="Submit" type="submit" class="txtbox4" value="Prints this Document "

                            onClick="return f3();" />

                        </label></td>

                    <td width="3%">&nbsp;</td>

                    <td width="26%"><label>

                        <input name="Submit2" type="submit" class="txtbox4" value="Close this document "

                            onClick="return f2();" />

                        </label></td>

                    <td width="8%">&nbsp;</td>

                    <td width="14%">&nbsp;</td>

                </tr>

            </table>

        </form>

    </td>

</tr>

<tr>

    <td colspan="2">&nbsp;</td>

</tr>

</table>

</body>

</html>

```

### Guardian\_approval.php

```
<?php
include('includes/config.php');

if (isset($_GET['id'], $_GET['action'])) {
    $id = $_GET['id'];
    $action = $_GET['action'];

    if (!in_array($action, ['approve', 'reject'])) {
        die("Invalid action. Action must be 'approve' or 'reject'.");
    }

    $guardian_status = ($action == 'approve') ? 'Approved' : 'Rejected';

    if (!filter_var($id, FILTER_VALIDATE_INT)) {
        die("Invalid ID format.");
    }

    $stmt = $mysqli->prepare("SELECT guardian_status FROM leave_requests WHERE id = ?");
    $stmt->bind_param("i", $id);
    $stmt->execute();
    $result = $stmt->get_result();

    if ($row = $result->fetch_assoc()) {
        if ($row['guardian_status'] == 'Pending') {
            $update_stmt = $mysqli->prepare("UPDATE leave_requests SET guardian_status = ?
WHERE id = ?");
            $update_stmt->bind_param("si", $guardian_status, $id);

            if ($update_stmt->execute()) {
                echo "<h2>Thank you for your response!</h2>
<p>Your decision has been recorded as: <b>$guardian_status</b>.</p>";
            }
        }
    }
}
```

```

        echo "<script>
            setTimeout(function() {
                window.close();
            }, 5000);
        </script>";
    } else {
        echo "<h2>Error</h2><p>There was an error updating the guardian's decision: " .
$update_stmt->error . "</p>";
    }
} else {
    $existing_status = $row['guardian_status'];
    echo "<h2>Decision Already Recorded</h2>
        <p>Your previous decision for this leave request was: <b>$existing_status</b>.</p>";
    echo "<script>
        setTimeout(function() {
            window.close();
        }, 5000);
    </script>";
}
} else {
    echo "<h2>Invalid Request</h2>
        <p>No leave request found with the provided ID.</p>";
    echo "<script>
        setTimeout(function() {
            window.close();
        }, 5000);
    </script>";
}
} else {
    die("Invalid request. Missing parameters.");
}
?>

```

### **MyProfile.php**

```
<?php
session_start();
include('includes/config.php');
date_default_timezone_set('Asia/Kolkata');
include('includes/checklogin.php');
check_login();
$aid = $_SESSION['id'];

if (isset($_POST['update'])) {
    $fname = $_POST['fname'];
    $mname = $_POST['mname'];
    $lname = $_POST['lname'];
    $gender = $_POST['gender'];
    $contactno = $_POST['contact'];
    $email = $_POST['email'];

    $query = "UPDATE userRegistration SET firstName=?, middleName=?, lastName=?, gender=?,
contactNo=?, email=?, updationDate=CURRENT_TIMESTAMP WHERE user_id=?";

    $stmt = $mysqli->prepare($query);

    $stmt->bind_param('ssssisi', $fname, $mname, $lname, $gender, $contactno, $email, $aid);
    $stmt->execute();

    echo "<script>alert('Profile updated Successfully');</script>";
}
?>

<!doctype html>
<html lang="en" class="no-js">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">
```

```

    <meta name="viewport" content="width=device-width, initial-scale=1, minimum-scale=1,
maximum-scale=1">

    <meta name="description" content="">

    <meta name="author" content="">

    <meta name="theme-color" content="#3e454c">

    <title>Profile Updation</title>
</head>
<body>

    <?php include('includes/header.php'); ?>

    <div class="ts-main-content">

        <?php include('includes/sidebar.php'); ?>

        <div class="content-wrapper">

            <div class="container-fluid">

                <?php

                    $ret = "SELECT *, DATE_FORMAT(updationDate, '%d-%m-%Y %h:%i:%s %p') as
formattedDate FROM userRegistration WHERE user_id=?";

                    $stmt = $mysqli->prepare($ret);

                    $stmt->bind_param('i', $aid);

                    $stmt->execute();

                    $res = $stmt->get_result();

                    while ($row = $res->fetch_object()) {

                        ?>

                        <div class="row">

                            <div class="col-md-12">

                                <div class="col-sm-6 col-sm-offset-4">

                                    <button class="btn btn-primary" type="submit" name="update"
value="Update Profile">Update Profile</button>

                                </div>

                            </div>

                        </div>

                    }

                </div>

            </div>

        </div>

    </body>

</html>

```

### **Registration.php**

```
<?php
session_start();
include('includes/config.php');

// Fetch the latest registration number and increment it
$query = "SELECT regNo FROM userRegistration ORDER BY user_id DESC LIMIT 1";
$result = $mysqli->query($query);
if ($result->num_rows > 0) {
    $row = $result->fetch_assoc();
    $lastRegNo = $row['regNo'];

    // Extract the numeric part and increment it
    $regNum = intval(substr($lastRegNo, 4)) + 1;
} else {
    $regNum = 1; // Start with 1 if no previous registration exists
}

// Format as REG-01, REG-02, etc.
$newRegNo = "REG-" . str_pad($regNum, 2, '0', STR_PAD_LEFT);

$query = "INSERT INTO userRegistration(regNo, firstName, middleName, lastName, gender,
contactNo, email, password)
VALUES(?, ?, ?, ?, ?, ?, ?, ?)";

$stmt = $mysqli->prepare($query);

$stmt->bind_param('sssssss', $regno, $fname, $mname, $lname, $gender, $contactno, $emailid,
$password);

if ($stmt->execute()) {
    echo "<script>alert('Student successfully registered');</script>";

    // Redirect to the same page to avoid duplicating registration number
    echo "<script>window.location.href='" . $_SERVER['PHP_SELF'] . "';</script>";
} else {
    echo "<script>alert('Registration failed');</script>";
}
```

```

    }
}
?>

<script src="js/jquery.min.js"></script>
<script src="js/bootstrap-select.min.js"></script>
<script src="js/bootstrap.min.js"></script>
<script src="js/jquery.dataTables.min.js"></script>
<script src="js/dataTables.bootstrap.min.js"></script>
<script src="js/Chart.min.js"></script>
<script src="js/fileinput.js"></script>
<script src="js/chartData.js"></script>
<script src="js/main.js"></script>
</body>
<script>
function checkAvailability() {
    $("#loaderIcon").show();
    jQuery.ajax({
        url: "check_availability.php",
        data: 'emailid=' + $("#email").val(),
        type: "POST",
        success: function (data) {
            $("#user-availability-status").html(data);
            $("#loaderIcon").hide();
        },
        error: function () {
            event.preventDefault();
            alert('error');
        }
    });
}
</script>
</html>

```

## **Room\_details.php**

```
<?php
session_start();
include('includes/config.php');
include('includes/checklogin.php');
check_login();
?>

<!doctype html>

<html lang="en" class="no-js">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1, minimum-scale=1,
maximum-scale=1">

    <meta name="description" content="">

    <meta name="author" content="">

    <meta name="theme-color" content="#3e454c">

    <title>Room Details</title>

    <link rel="stylesheet" href="css/font-awesome.min.css">

    <link rel="stylesheet" href="css/bootstrap.min.css">

    <link rel="stylesheet" href="css/dataTables.bootstrap.min.css">

    <link rel="stylesheet" href="css/bootstrap-social.css">

    <link rel="stylesheet" href="css/bootstrap-select.css">

    <link rel="stylesheet" href="css/fileinput.min.css">

    <link rel="stylesheet" href="css/awesome-bootstrap-checkbox.css">

    <link rel="stylesheet" href="css/style.css">

    <script language="javascript" type="text/javascript">

        var popUpWin = 0;

        function popUpWindow(URLStr, left, top, width, height) {

            if (popUpWin) {

                if (!popUpWin.closed) popUpWin.close();

            }

            popUpWin = window.open(URLStr, left, top, width, height);

        }

    </script>

</head>

<body>
```



```

        popUpWin = open(URLStr, 'popUpWin',
'toolbar=no,location=no,directories=no,status=no,menubar=no,scrollbars=yes,resizable=no,copyhistor
y=yes,width=' + 510 + ',height=' + 430 + ',left=' + left + ', top=' + top + ',screenX=' + left + ',screenY='
+ top + ');

    }

```

```

</script>

```

```

</head>

```

```

<body>

```

```

    <?php include('includes/header.php'); ?>

```

```

        <td><b>Correspondense Address</b></td>

```

```

        <td colspan="2">

```

```

            <?php echo $row->corresAddress; ?><br />

```

```

            <?php echo $row->corresCity; ?>, <?php echo $row->corresPincode; ?><br />

```

```

            <?php echo $row->corresState; ?>

```

```

        </td>

```

```

        <td><b>Permanent Address</b></td>

```

```

        <td colspan="2">

```

```

            <?php echo $row->pmntAddress; ?><br />

```

```

            <?php echo $row->pmntCity; ?>, <?php echo $row->pmntPincode; ?><br />

```

```

            <?php echo $row->pmnatetState; ?>

```

```

        </td>

```

```

    </tr>

```

```

    <?php

```

```

        $cnt = $cnt + 1;

```

```

    } ?>

```

```

</tbody>

```

```
        </table>
    </div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>

<!-- Loading Scripts -->
<script src="js/jquery.min.js"></script>
<script src="js/bootstrap-select.min.js"></script>
<script src="js/bootstrap.min.js"></script>
<script src="js/jquery.dataTables.min.js"></script>
<script src="js/dataTables.bootstrap.min.js"></script>
<script src="js/Chart.min.js"></script>
<script src="js/fileinput.js"></script>
<script src="js/chartData.js"></script>
<script src="js/main.js"></script>

</body>

</html>
```

## Track requests.php

```
<?php
session_start();
include('includes/config.php');
include('includes/checklogin.php');
check_login();
?>

<!doctype html>
<html lang="en" class="no-js">

<body>
    <?php include('includes/header.php'); ?>

    <div class="ts-main-content">
        <?php include('includes/sidebar.php'); ?>
        <div class="content-wrapper">
            <div class="container-fluid">
                <div class="row">
                    <div class="col-md-12">
                        <h2 class="page-title" style="margin-top: 45px;"> Track Leave Requests</h2>
                        <div class="panel panel-default">
                            <div class="panel-heading">Leave Requests</div>
                            <div class="panel-body">
                                <table id="zctb" class="table table-bordered " cellspacing="0" width="100%">

                                    <tbody>

                                        <?php
                                            $user_id = $_SESSION['id']; // Assuming `id` stores the user ID
                                            $query = "SELECT leave_start_date, leave_end_date, reason, status,
guardian_status FROM leave_requests WHERE student_id = ?";
                                            $stmt = $mysqli->prepare($query);
                                            $stmt->bind_param("i", $user_id);
```

```

$stmt->execute();

$result = $stmt->get_result();

?>

<?php while ($row = $result->fetch_assoc()) { ?>

    <tr>

        <td><?php echo $row['leave_start_date']; ?></td>

        <td><?php echo $row['leave_end_date']; ?></td>

        <td><?php echo $row['reason']; ?></td>

        <td class="text-center"><?php echo $row['guardian_status']; ?></td>

        <td class="text-center"><?php echo $row['status']; ?></td>

    </tr>

<?php } ?>

</table>

<!-- Loading Scripts -->

<script src="js/jquery.min.js"></script>

<script src="js/bootstrap-select.min.js"></script>

<script src="js/bootstrap.min.js"></script>

<script src="js/jquery.dataTables.min.js"></script>

<script src="js/dataTables.bootstrap.min.js"></script>

<script src="js/Chart.min.js"></script>

<script src="js/fileinput.js"></script>

<script src="js/chartData.js"></script>

<script src="js/main.js"></script>

</body>

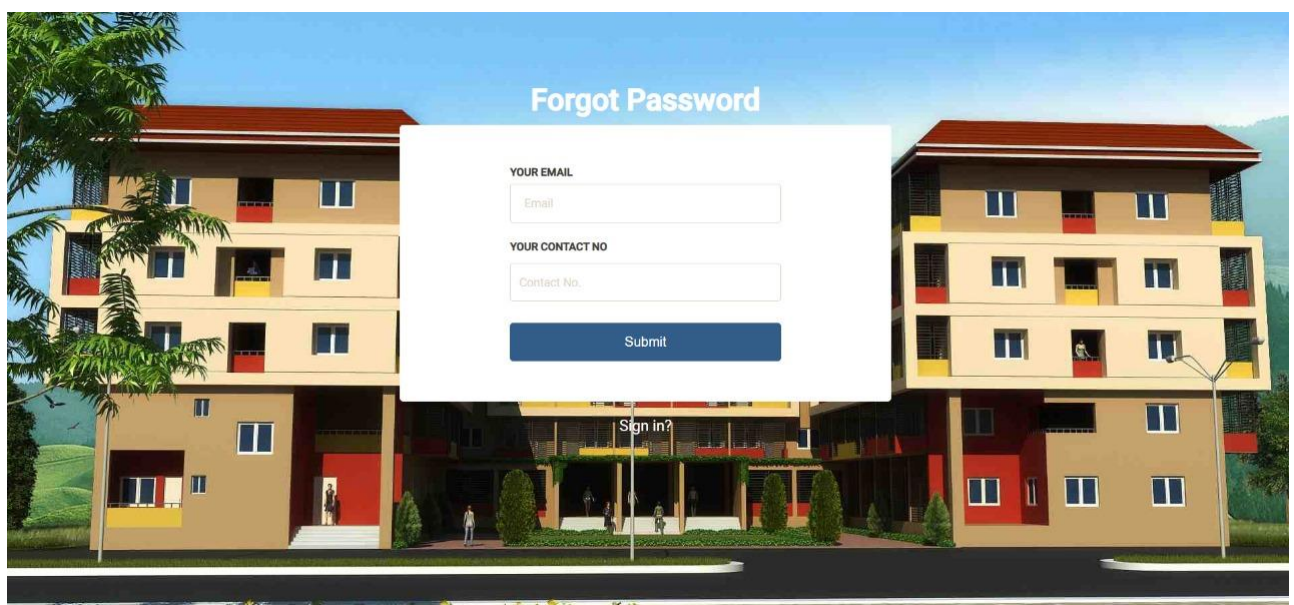
</html>

```

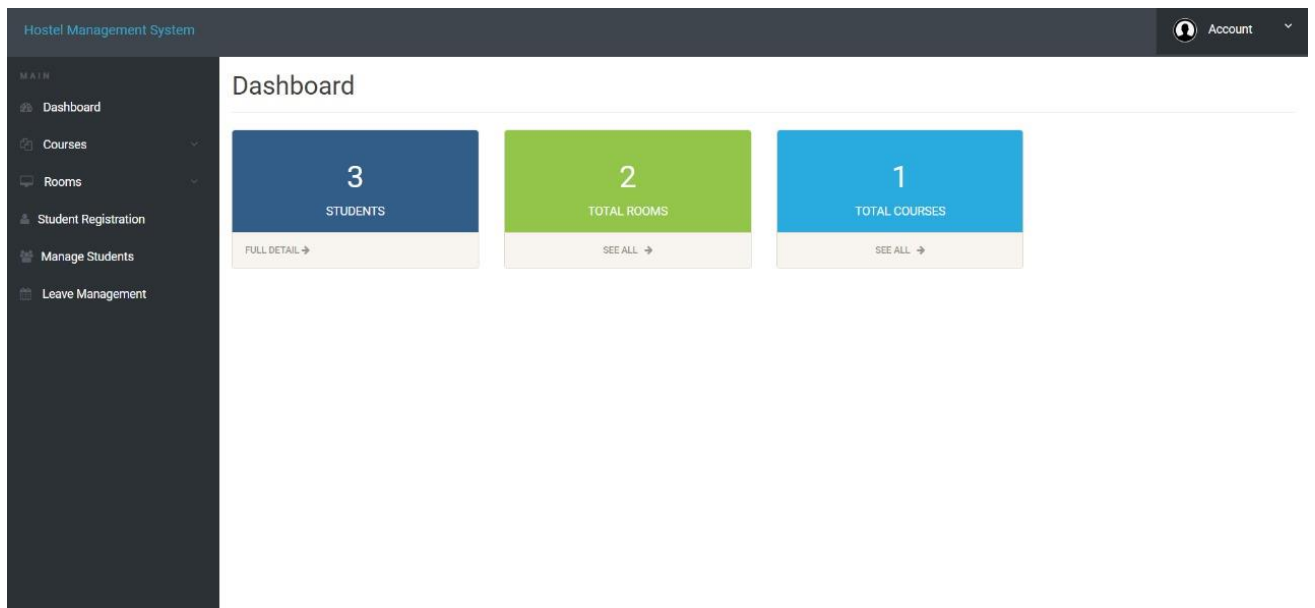
## SNAPSHOTS



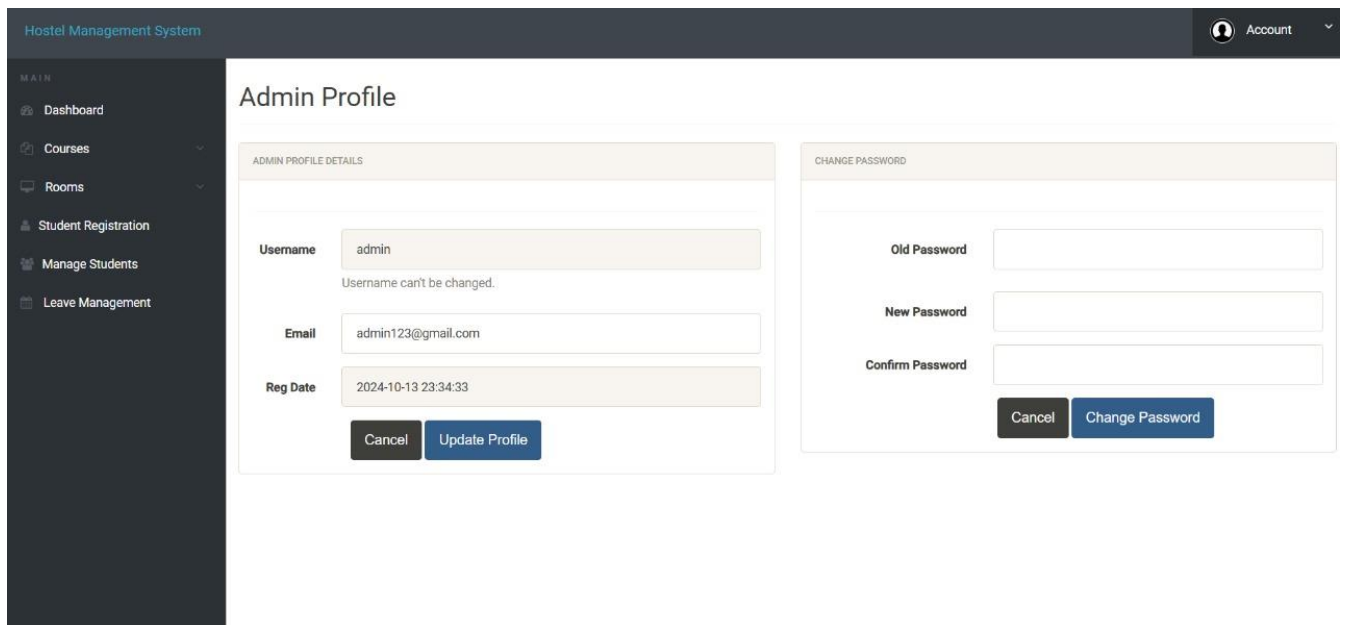
**Fig No. 8 : Admin Login**



**Fig No. 9: Forgot Password**



**Fig No. 10: Admin Dashboard**



**Fig No. 11: Admin Profile**

Hostel Management System
Account

MAIN
Dashboard
Courses
Rooms
Student Registration
Manage Students
Leave Management

Registration

FILL ALL INFO

Room Related info

Room no.
Select Room

Seater

Fees Per Month

Food Status
☒ Without Food ☐ With Food(Rs 2000.00 Per Month Extra)

Stay From
dd-mm-yyyy

Duration
Select Duration in Month

Total Amount

Personal info

**Fig No. 12 : Hostel Booking 1 (Admin )**

Hostel Management System
Account

MAIN
Dashboard
Courses
Rooms
Student Registration
Manage Students
Leave Management

Personal info

Course
Select Course

Registration No :
Select Registration No

First Name :

Middle Name :

Last Name :

Gender :
Select Gender

Contact No :
Enter contact number

Email id :

Emergency Contact:

Guardian Name :

**Fig No. 13 : Hostel Booking 2 (Admin )**

The screenshot displays the 'Hostel Management System' Admin interface. On the left is a dark sidebar with a 'MAIN' menu containing: Dashboard, Courses, Rooms, Student Registration, Manage Students, and Leave Management. The top header shows 'Hostel Management System' and an 'Account' dropdown. The main content area is titled 'Correspondence Address' in green. It contains a form with the following fields: 'Address' (a large text input), 'Country' (a text input), 'State' (a dropdown menu currently showing 'Select State'), 'City' (a text input), and 'Pincode' (a text input). Below this is a section titled 'Permanent Address' in green, which includes a checkbox labeled 'Permanent Address same as Correspondence Address' and a single 'Address' text input field.

**Fig No. 14 : Hostel Booking 3 (Admin)**

The screenshot displays the 'Hostel Management System' Admin interface for adding courses. The sidebar and header are identical to the previous figure. The main content area is titled 'Add Courses'. Below the title is a light orange header bar with the text 'ADD COURSES'. The form contains three text input fields: 'Course Code', 'Course Name (Short)', and 'Course Name(Full)'. At the bottom of the form is a blue button labeled 'Add Course'.

**Fig No. 15: Add Courses (Admin)**



Hostel Management System

Account

MAIN

Dashboard

Courses

Rooms

Student Registration

Manage Students

Leave Management

## Manage Courses

ALL COURSES DETAILS

Show 10 entries

Search:

Sno.	Course Code	Course Name(Short)	Course Name(Full)	Reg Date	Action
1	AGC-01	B.Tech CSE	Bachelor of Technology in Computer Science and Engineering	2024-10-23 16:05:01	<a href="#">✎</a> <a href="#">✕</a>

Showing 1 to 1 of 1 entries

PREVIOUS1NEXT

**Fig No. 16 : Managing Courses (Admin)**

Hostel Management System

Account

MAIN

Dashboard

Courses

Rooms

Student Registration

Manage Students

Leave Management

## Add a Room

ADD A ROOM

Select Seater

Select Seater

Room No.

Fee(Per Student)

Add Room

**Fig No. 17 : Add Rooms (Admin )**

Hostel Management System
Account

MAIN
Dashboard
Courses
Rooms
Student Registration
Manage Students
Leave Management

## Manage Rooms

ALL ROOM DETAILS

Show 10 entries
Search:

Sno.	Seater	Room No.	Fees (PM)	Posting Date	Action
1	2	101	8000	2024-10-28 22:45:57	<a href="#">Edit</a> <a href="#">Delete</a>
2	2	102	8000	2024-11-18 20:21:52	<a href="#">Edit</a> <a href="#">Delete</a>

Showing 1 to 2 of 2 entries
PREVIOUS 1 NEXT

**Fig No. 18 : Managing Rooms**

Hostel Management System
Account

MAIN
Dashboard
Courses
Rooms
Student Registration
Manage Students
Leave Management

## Manage Students

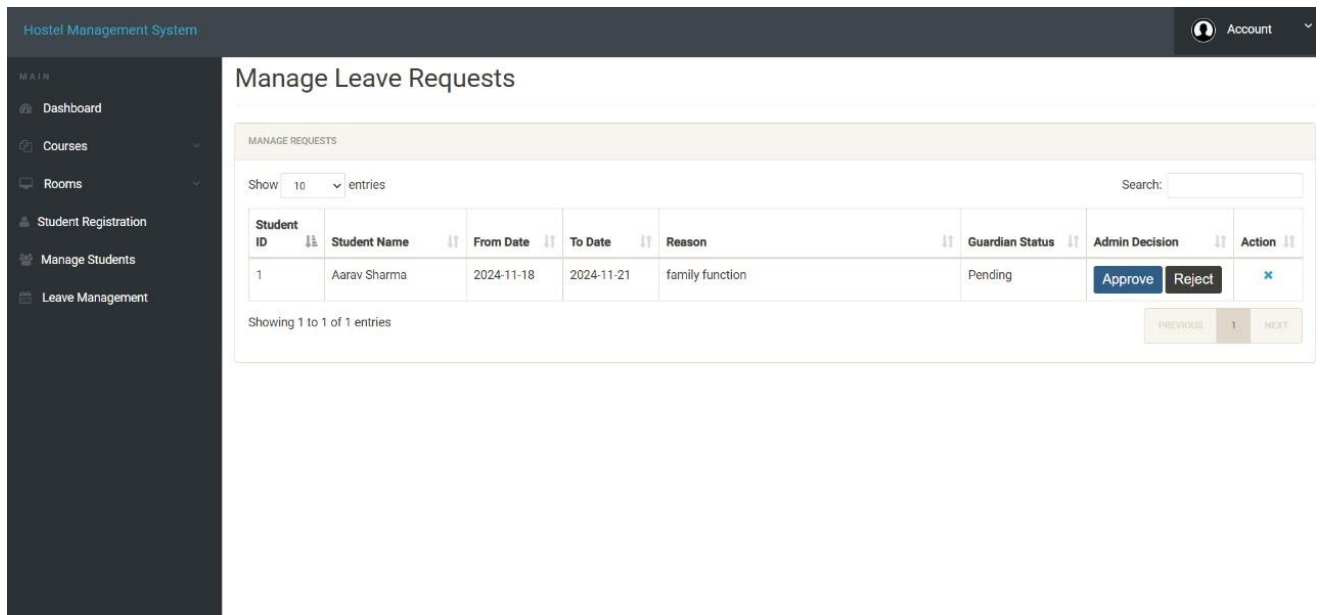
ALL STUDENT DETAILS

Show 10 entries
Search:

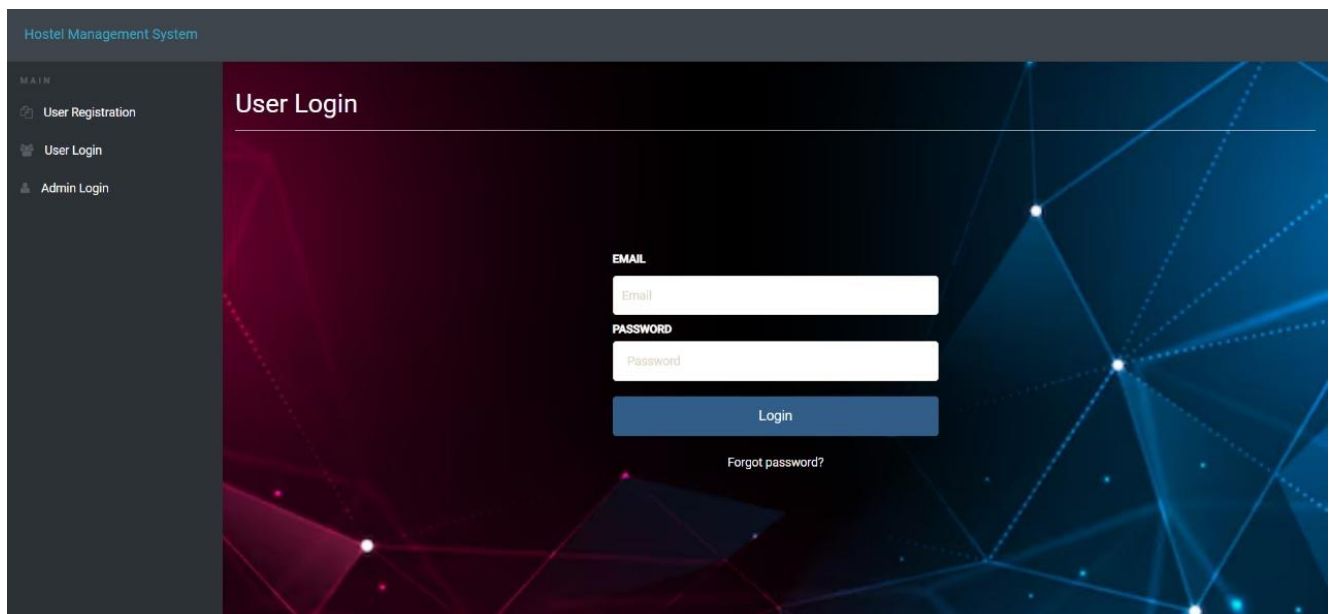
Sno.	Student Name	Reg no	Contact no	Room no	Seater	Staying From	Action
1	AaravSharma	REG-01	6785231490	101	2	2024-11-01	<a href="#">Edit</a> <a href="#">Delete</a>
2	KartikMahajan	REG-02	5469871230	101	2	2024-11-01	<a href="#">Edit</a> <a href="#">Delete</a>

Showing 1 to 2 of 2 entries
PREVIOUS 1 NEXT

**Fig No. 19 : Managing Students (Admin)**



**Fig No. 20 : Managing Leave Requests (Admin)**



**Fig No. 21 : User Login**

Hostel Management System

MAIN

User Registration

User Login

Admin Login

Student Registration

FILL ALL INFO

Registration No :

REG-04

First Name :

Middle Name :

Last Name :

Gender :

Select Gender

Contact No :

Enter contact number

Email id:

**Fig No. 22 : User Registration**

Hostel Management System

Logout

MAIN

Dashboard

My Profile

Change Password

Room Details

Leave Requests

Aarav's Profile

LAST UPDATION DATE:

Registration No:

REG-01

First Name:

Aarav

Middle Name:

Last Name:

Sharma

Gender:

Male

Contact No :

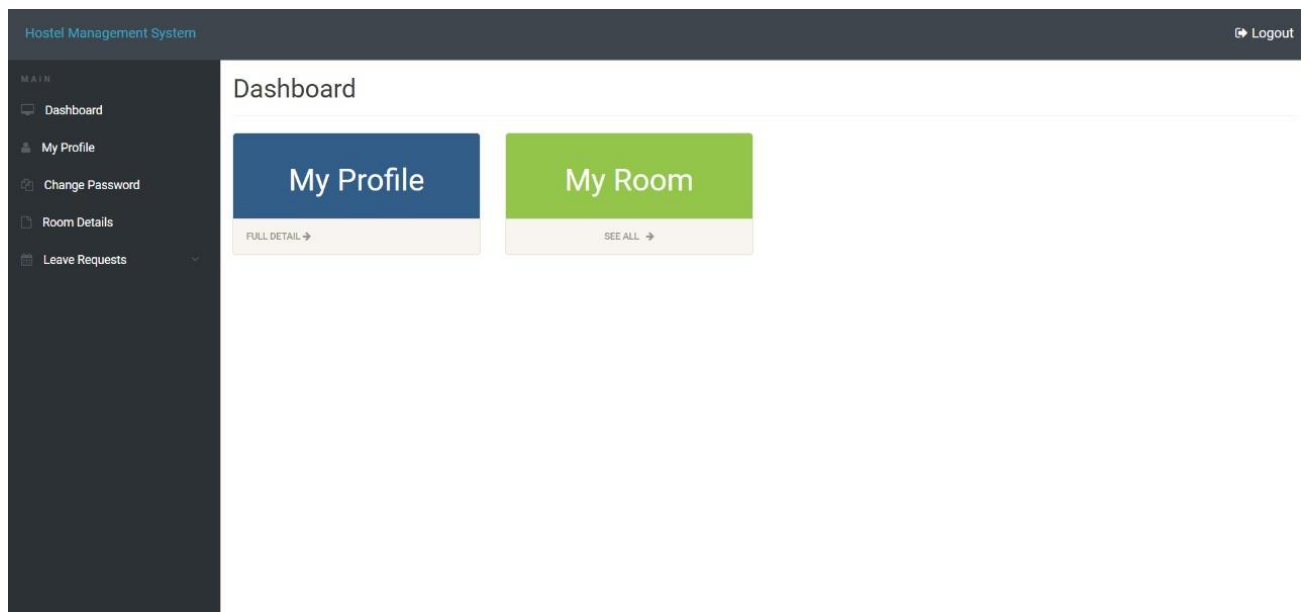
+91 9876543210

Email id:

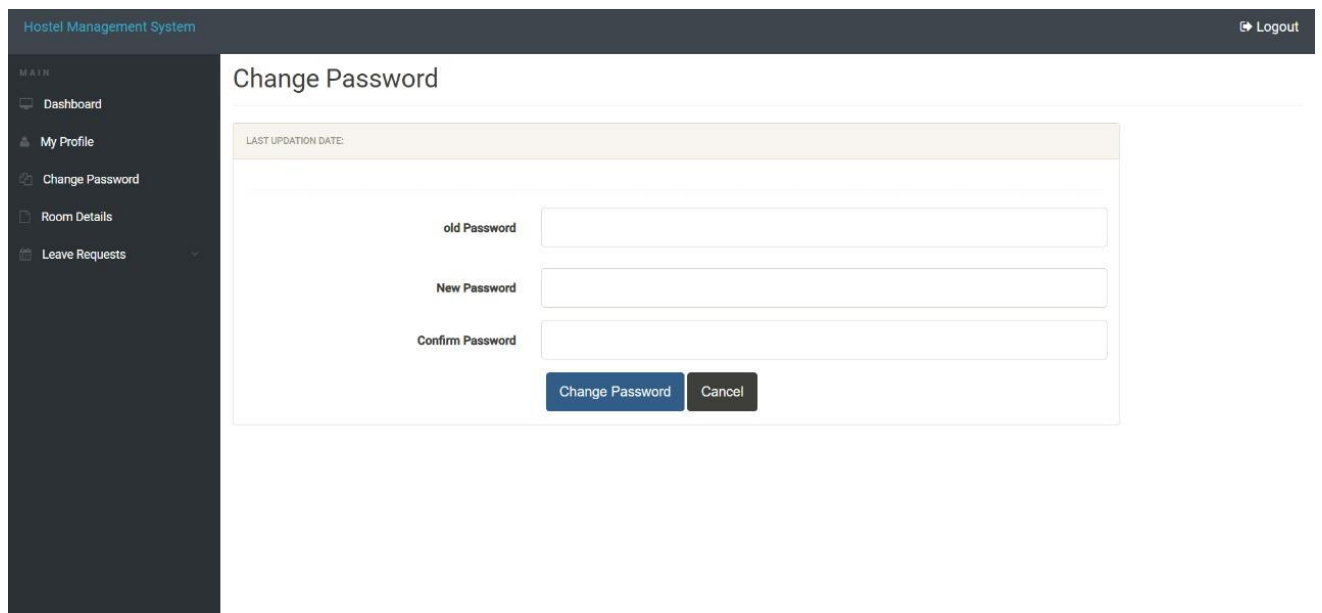
aarav123@gmail.com

Update Profile

**Fig No. 23 : User Profile**



**Fig No. 24 : Add Details (User)**



**Fig No. 25 : Change Password (User)**

Hostel Management System

Logout

MAIN

Dashboard

My Profile

Change Password

Room Details

Leave Requests

## Make Leave Request

FILL ALL INFO

From Date

dd-mm-yyyy

To Date

dd-mm-yyyy

Reason

Enter reason here

Submit Leave Request

**Fig No. 26 : Make Leave Request (User)**

Hostel Management System

Logout

MAIN

Dashboard

My Profile

Change Password

Room Details

Leave Requests

## Track Leave Requests

LEAVE REQUESTS

From Date	To Date	Reason	Guardian Status	Admin Status
2024-11-18	2024-11-21	family function	Pending	Pending

**Fig No. 27 : Track Leave Requests (User)**

## CONCLUSION

The Hostel Management System has successfully fulfilled its objective of providing a web-based solution to streamline the complex and time-consuming administrative tasks involved in hostel management. With its focus on automating processes such as room allocation, leave management, and student record tracking, the system reduces the reliance on manual work, ensuring greater accuracy and efficiency. The implementation of PHP for backend operations, MySQL for database management, and a combination of HTML, CSS, JavaScript, and Ajax for the frontend ensures the system is both powerful and user-friendly, offering a seamless experience for both administrators and students.

The key strengths of this system lie in its simplicity, real-time data access, and smooth user interaction. By using Ajax for asynchronous data handling, the system eliminates the need for page reloads, enhancing the overall usability and responsiveness. Administrators can easily manage student data, assign rooms, approve leave requests, and monitor room status in real-time. On the other hand, students have easy access to essential services such as viewing room allocations, requesting leaves, and updating personal information, all through a secure login system.

The deployment of the system using XAMPP ensures a stable and efficient development and testing environment, making the system ready for real-world implementation. Through the automation of various hostel-related processes, this project contributes to reducing administrative burden, saving valuable time, and eliminating human errors. Additionally, the system improves the user experience for students by providing them with easy and immediate access to their hostel-related information.

This project has not only enhanced the technical skills of the development team in web development, PHP, and database management but has also provided valuable insights into the practical challenges of building real-world applications. Moving forward, the Hostel Management System can be extended to handle more complex operations such as fee management, complaint tracking, or multi-hostel management. It can also be adapted for mobile platforms to further enhance accessibility and user engagement.

In conclusion, the Hostel Management System offers a robust, efficient, and scalable solution to modernize hostel operations in educational institutions. It improves administrative efficiency, provides students with enhanced access to services, and serves as a foundation for future upgrades and extensions, making it a comprehensive solution for hostel management.

## FUTURE SCOPE OF THE PROJECT

The Hostel Management System, while already functional and efficient, offers significant potential for future enhancements and scalability. As the needs of educational institutions evolve, the system can be expanded and upgraded to provide more comprehensive features, better integration, and improved user experience. Some potential areas for future development include:

1. **Mobile Application Integration:** To make the system more accessible, a mobile application could be developed for both Android and iOS platforms. This would allow students and administrators to access hostel services on-the-go, further enhancing convenience and real-time interaction.
2. **Fee Management System:** One of the key features that can be added is an integrated fee management system. This would allow students to view their fee structure, make payments, track dues, and generate receipts, thereby reducing manual tracking and enhancing financial management for hostel administrators.
3. **Multi-Hostel Support:** The current system is designed for a single hostel, but it can be scaled to handle multiple hostels within an institution. This would involve adding the ability to manage different hostels under the same platform, enabling administrators to view and control operations across all hostels from a centralized interface.
4. **Complaint Management and Feedback System:** Adding a module for managing student complaints and feedback would help improve hostel facilities and services. Students could submit complaints or suggestions, and administrators could track and resolve issues in a timely manner, improving the overall hostel environment.
5. **Security Enhancements:** As the system expands, it may require enhanced security measures. Features like two-factor authentication (2FA), encrypted data storage, and secure payment gateways could be implemented to further safeguard sensitive data and ensure the privacy and security of both students and administrators.
6. **User Interface Improvements:** While the current system offers a user-friendly interface, continuous improvements based on user feedback and emerging design trends could enhance the overall aesthetic and functionality. A more intuitive and responsive UI would provide a better experience across various devices and screen sizes.
7. **Cloud Deployment:** Hosting the system on the cloud could improve scalability, performance, and accessibility. Cloud deployment would also ensure better data backup and disaster recovery solutions, making the system more reliable and robust for long-term use.



## BIBLIOGRAPHY

### Websites:

- <https://www.google.com/>
- <https://chatgpt.com/>
- <https://www.php.net/>
- <https://www.javascript.com/>
- <http://www.jdbc-tutorial.com/>
- <https://www.tutorialspoint.com/java/>
- <http://www.javatpoint.com/java-tutorial>
- <https://docs.oracle.com/javase/tutorial/>
- <http://www.tutorialspoint.com/mysql/>