

**SIMATS SCHOOL OF ENGINEERING**

**SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES**

**CHENNAI-602105**

**Student Management System**

**A CAPSTONE PROJECT REPORT**

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

**BACHELOR OF ENGINEERING**

**IN**

**INFORMATION TECHNOLOGY**

**Submitted by**

**M.Monish (192210315)**

**C.Harshavardhan Reddy (192210375)**

**Under the Supervision of**

**Dr. M.kathiravan**

**July 2024**

**DECLARATION**

We, **M. Monish, C.Harshavardhan Reddy**, students of **Bachelor of Engineering in Information Technology**, Department of Computer Science and Engineering, Saveetha Institute of Medical and Technical Sciences, Saveetha University, Chennai, hereby declare that the work presented in this Capstone Project Work entitled **Student Management system** is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics.

(M. Monish 192210315)

(C. Harshavardhan Reddy 92210375)

Date:

Place:

**CERTIFICATE**

This is to certify that the project entitled **“Student Management system”** submitted by **M.Monish, C.Harsha vardhan Reddy** has been carried out under my supervision. The project has been submitted as per the requirements in the current semester of B. Tech Information Technology.

Teacher-in-charge

Dr. M. Kathiravan

**Table of Contents**

|  |  |
| --- | --- |
| **S.NO** | **TOPICS** |
| 1 | **Abstract** |
| 2 | **Introduction** |
| 3 | **Project Description**  About your project |
| 4 | **Problem Description**  Program to build a simple Software for < > |
| 5 | **Tool Description**  User interface  Features |
| 6 | **Operations**  Store the First name of the student.  Store the Last name of the student.  Store the unique Roll number for every student.  Store the CGPA of every student.  Store the courses registered by the student. |
| 7 | **Approach / Module Description / Functionalities**  The idea is to form an individual functions for every operation. All the functions are unified together to form software. |
| 8 | **Implementation**  Coding |
| 9 | **Output**  Output with Screenshots |
| 10 | **Conclusion**  Future Enhancement  **References** |

**Abstract:**

A Student Management System (SMS) utilizing PHP and MySQL is an efficient, web based solution designed to streamline administrative tasks in educational institutions. This system automates the management of student data, including registration, attendance, grades, and personal information. PHP, a server-side scripting language, ensures dynamic and interactive functionalities, while MySQL, a robust relational database management system, securely stores and retrieves data. The SMS provides user-friendly interfaces for administrators, teachers, and students, enhancing communication and operational efficiency. It supports role-based access control, ensuring data privacy and integrity. Features such as real-time updates, report generation, and easy data accessibility contribute to informed decision-making and improved educational outcomes. The implementation of this system reduces manual workload, minimizes errors, and

facilitates seamless information flow within the institution, making it an indispensable tool for modern educational management**.**

**Introduction:**

A Student Management System (SMS) is a comprehensive platform designed to facilitate

the efficient administration of educational institutions by automating various academic and

administrative tasks. Utilizing PHP and MySQL, a web-based SMS leverages the robust

capabilities of these technologies to provide a dynamic, scalable, and user-friendly interface for managing student information, streamlining administrative processes, and enhancing

communication between students, teachers, and administrators. PHP, a popular server-side

scripting language, offers the versatility and functionality required to develop interactive and

dynamic web applications. Its ease of integration with various databases, including MySQL, makes it an ideal choice for developing a robust student management system. MySQL, an open-source relational database management system, provides a reliable and efficient way to store, retrieve,

and manage vast amounts of data, ensuring data integrity and performance even with high volumes of transactions. The core features of an SMS typically include student enrollment and registration, attendance tracking, grade management, timetable scheduling, and communication modules. These functionalities aim to minimize manual paperwork, reduce redundancy, and provide realtime access to essential data. Additionally, an SMS can offer role-based access, ensuring that students, teachers, and administrators have appropriate permissions tailored to their specific needs and responsibilities. Implementing a student management system using PHP and MySQL not only enhances the operational efficiency of educational institutions but also improves the overall

learning experience for students. By centralizing and automating critical processes, it allows educators to focus more on teaching and less on administrative burdens, ultimately fostering a

more organized and effective educational environment.

**Project Description:**

The Student Management System is a web-based application developed using PHP and MySQL that aims to simplify the management of student information. This system will handle various student-related data such as personal details, academic records, attendance, and more, providing a centralized platform for administrators, teachers, and students to interact efficiently.

**User Authentication and Authorization:**

Secure login for administrators, teachers, and students.

Role-based access control to ensure data security and privacy.

**Student Information Management:**

Add, edit, delete, and view student personal details.

Manage student contact information and emergency contacts.

**Academic Records:**

Record and manage student grades and performance.

Generate transcripts and report cards.

**Attendance Tracking:**

Mark and track student attendance.

Generate attendance reports and alerts for absenteeism.

Class and Course Management:

Manage class schedules and course information.

Assign teachers to courses and classes.

**Fees Management:**

Record and track student fee payments.

Generate fee receipts and outstanding fee reports.

**Communication:**

Send notifications and announcements to students and parents.

Integrate email and SMS functionalities for communication.

**Reports and Analytics:**

Generate various reports (attendance, grades, fees, etc.)

Provide insights and analytics for administrative purposes

**Implementation Plan:**

**Phase 1: Requirement Analysis and Planning**

Gather requirements and define project scope.

Design database schema and system architecture.

**Phase 2: UI/UX Design**

Create wireframes and mockups.

Design user interface for different user roles.

**Phase 3: Development**

Set up development environment (PHP, MySQL, Apache).

Implement user authentication and authorization.

Develop modules for student information management, attendance tracking, grade management, and fees management.

Integrate communication module and reporting functionalities.

**Phase 4: Testing**

Perform unit testing, integration testing, and system testing.

Fix bugs and optimize performance.

**Phase 5: Deployment and Maintenance**

Deploy the system on a web server.

Provide training and documentation for users.

Perform regular maintenance and updates.

**Problem Description:**

Background:

Educational institutions, such as schools, colleges, and universities, manage a significant amount of student information. This includes personal details, academic records, attendance, and more. Traditionally, this data has been managed manually or through disparate digital systems, leading to inefficiencies, data inaccuracies, and difficulties in accessing and sharing information.

**Problem Statement:**

The manual and fragmented approach to student information management presents several challenges:

**Inefficiency and Time-Consuming Processes:**

Administrative tasks, such as student registration, attendance tracking, and grade management, are time-consuming and prone to errors when performed manually.

The need to handle physical records and spreadsheets leads to inefficient use of administrative resources.

**Data Inaccuracies and Redundancy:**

Manual data entry and maintenance can result in errors, inconsistencies, and duplication of information.

Lack of a centralized database makes it difficult to ensure data integrity and accuracy.

**Difficulty in Data Access and Sharing:**

Teachers, administrators, and students may find it challenging to access necessary information quickly.

Disparate systems and lack of integration impede the smooth flow of information among stakeholders.

**Limited Reporting and Analytics:**

Generating reports and analyzing student data for insights is cumbersome without an automated system.

Lack of timely and accurate reports affects decision-making processes.

Security and Privacy Concerns:

Manual handling of sensitive student information poses risks related to data security and privacy.

Ensuring secure access to student records for authorized users is difficult without a robust system.

**Proposed Solution:**

Develop a web-based Student Management System (SMS) using PHP and MySQL to address the identified problems. The proposed system will:

**Streamline Administrative Tasks:**

Automate processes such as student registration, attendance tracking, grade management, and fee collection.

Reduce the time and effort required for administrative tasks, increasing overall efficiency.

**Centralize Student Data:**

Maintain a centralized database for all student-related information, ensuring data accuracy and reducing redundancy.

Simplify data entry and maintenance with user-friendly interfaces.

**Enhance Data Accessibility and Sharing:**

Provide secure, role-based access to student information for administrators, teachers, and students.

Facilitate quick and easy access to necessary data, improving communication and collaboration among stakeholders.

**Improve Reporting and Analytics:**

Enable the generation of various reports, such as attendance records, grade summaries, and fee statements.

Provide analytics and insights to support data-driven decision-making.

**Ensure Data Security and Privacy:**

Implement secure authentication and authorization mechanisms to protect sensitive student information.

Ensure compliance with data privacy regulations and best practices for information security.

**Tools Description:**

**HTML (HyperText Markup Language)**

Description: HTML is the standard markup language for creating web pages. It structures the content on the web by using elements represented by tags.

Usage in Project:

Creating the layout of your reservation system.

Structuring content such as forms, tables, and buttons.

**PHP:**

Database Interaction: PHP connects to databases (e.g., MySQL) to retrieve, insert, and manipulate data, facilitating the creation of data-driven websites.

Session Management: It manages user sessions, essential for maintaining user states across multiple pages in applications like shopping carts and login systems.

Server-Side Scripting: PHP is a server-side scripting language used to create dynamic web pages by embedding it into HTML.

**JavaScript**

Description: JavaScript is a programming language that allows you to implement complex features on web pages. It can update and change both HTML and CSS.

Usage in Project:

Adding interactivity to the reservation system (e.g., form validation, dynamic content updates).

Managing data (e.g., user inputs, seat availability).

**VS Code (Visual Studio Code)**

Description: VS Code is a source-code editor developed by Microsoft. It includes support for debugging, syntax highlighting, intelligent code completion, snippets, and code refactoring.

Usage in Project:

Writing and editing HTML, CSS, and JavaScript code.

Using extensions to enhance functionality (e.g., live server, linters).

**Additional Tools and Technologies**

Description: Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development.

Usage in Project:

Quickly designing and customizing responsive mobile-first sites.

**Operations:**

**1. User Authentication and Authorization:**

**Operations:**

**User Registration:**

Users can create an account by providing personal details and selecting a role (administrator, teacher, student).

The system validates the provided information and stores it securely in the database.

**User Login:**

Users can log in by entering their username and password.

The system verifies credentials and grants access based on the user's role.

**Password Recovery:**

Users can recover their password by providing their registered email.

The system sends a password reset link to the user's email.

Session Management:

The system manages user sessions to maintain login state and secure access.

**2. Student Information Management:**

**Operations:**

**Add Student:**

Administrators and teachers can add new student profiles by entering personal, contact, and academic details.

The system validates and stores the information in the database.

**Edit Student:**

Authorized users can update student information.

The system retrieves the existing data, allows edits, and saves changes.

**View Student:**

Users can view detailed student profiles.

The system fetches and displays student data from the database.

**Delete Student:**

Administrators can delete student records.

The system removes the selected student data from the database.

**Search and Filter Students:**

Users can search and filter students based on criteria like name, class, or grade.

The system retrieves and displays matching results.

**3. Academic Records:**

**Operations:**

**Enter Grades:**

Teachers can enter and update grades for students.

The system validates and stores the grades in the database.

**View Academic Performance:**

Users can view individual student grades and academic performance.

The system fetches and displays grade data.

**Generate Transcripts:**

The system generates academic transcripts for students.

Users can view, print, or export transcripts.

**Report Cards:**

The system generates report cards summarizing student grades.

Users can print or export report cards.

**4. Attendance Tracking:**

**Operations:**

**Mark Attendance:**

Teachers can mark daily attendance for their classes.

The system records attendance status (present, absent, late) in the database.

**View Attendance:**

Users can view attendance records for individual students or classes.

The system retrieves and displays attendance data.

**Generate Attendance Reports:**

The system generates attendance reports.

Users can print or export attendance reports.

**Absentee Alerts:**

The system sends alerts and notifications for absenteeism.

Notifications can be sent via email or SMS.

**5. Class and Course Management:**

**Operations:**

**Manage Class Schedules:**

Administrators can create and update class schedules and timetables.

The system stores schedule information and ensures it is accessible to users.

**Assign Teachers:**

Administrators can assign teachers to specific classes and courses.

The system updates teacher assignments in the database.

**Update Course Details:**

Users can view and edit course information, including syllabus and schedule.

The system saves updated course details.

**Enroll Students:**

Administrators and teachers can enroll students in courses.

The system records and manages student enrollments.

**6. Fees Management:**

**Operations:**

**Record Fee Payments:**

Administrators can record and track student fee payments.

The system updates payment records and generates receipts.

**View Fee Status:**

Users can view fee payment status and outstanding balances.

The system retrieves and displays fee data.

**Generate Fee Receipts:**

The system generates fee receipts for payments made.

Users can print or export fee receipts.

**Outstanding Fee Reports:**

The system generates reports on outstanding fees.

Users can print or export these reports.

**Approach / Module Description / Functionalities**

**Approach:**

The development of the Student Management System (SMS) will follow a modular approach, breaking down the system into distinct functional components. Each module will address specific requirements and functionalities, ensuring a structured and scalable design. The project will be developed using agile methodologies, allowing iterative development, testing, and refinement.

**Module Descriptions and Functionalities:**

**1. User Authentication and Authorization:**

**Description:**

This module handles the secure login and registration processes for different types of users (administrators, teachers, and students). It ensures that only authorized users can access specific features and data.

**Functionalities:**

User registration with role assignment (admin, teacher, student).

Secure login with password hashing.

Password recovery via email.

Role-based access control to restrict unauthorized access.

Session management for secure and efficient user interaction.

2. **Student Information Management:**

**Description:**

This module manages all aspects of student personal data. It allows administrators and teachers to add, edit, view, and delete student records.

**Functionalities:**

Add new student profiles with personal, contact, and academic details.

Edit existing student information.

View detailed student profiles.

Delete student records.

Search and filter student lists.

**3. Academic Records:**

**Description:**

This module manages student grades and academic performance. It allows teachers to enter and update grades and generate academic reports.

**Functionalities:**

Enter and update student grades for various subjects.

View individual student academic performance.

Generate transcripts and report cards.

Export academic reports to PDF or print.

**4. Attendance Tracking:**

**Description:**

This module tracks student attendance. It provides an interface for teachers to mark attendance and generate attendance reports.

**Functionalities:**

Mark daily attendance for classes.

Track attendance status (present, absent, late).

Generate attendance reports for individual students or classes.

Send alerts and notifications for absenteeism.

**5. Class and Course Management:**

**Description:**

This module manages class schedules, course information, and teacher assignments. It ensures efficient organization and management of classes and courses.

**Functionalities:**

Create and manage class schedules and timetables.

Assign teachers to specific classes and courses.

View and update course details, including syllabus and schedule.

Enroll students in courses.

**6. Fees Management:**

**Description:**

This module handles the recording and tracking of student fee payments. It provides functionality for generating fee receipts and managing outstanding balances.

**Functionalities:**

Record and track fee payments for students.

Generate and print fee receipts.

Manage outstanding balances and send payment reminders.

Generate fee reports and summaries.

**7. Communication:**

**Description:**

This module facilitates communication between administrators, teachers, students, and parents. It supports sending notifications and announcements via email and SMS.

**Functionalities:**

Send notifications and announcements to users.

Compose and send messages via email and SMS.

Manage communication history.

Send bulk messages or individual notifications.

**8. Reports and Analytics:**

**Description:**

This module provides tools for generating various reports and visualizing data through charts and graphs. It supports data-driven decision-making processes.

**Functionalities:**

Generate reports on attendance, grades, fees, and more.

Customizable report formats.

Export reports to PDF or print.

Visualize data through charts and graphs for insights.

**Implementation:**

**Index.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Courses - Student Management System</title>

<link rel="stylesheet" href="st.css">

</head>

<body>

<header>

<div class="logo">Student Management System</div>

<nav>

<ul>

<li><a href="home.html">Home</a></li>

<li><a href="students.html">Students</a></li>

<li><a href="courses.html">Courses</a></li>

<li><a href="profile.html">Profile</a></li>

<li><a href="file:///C:/Users/jayap/.vscode/inde.html">Logout</a></li>

</ul>

</nav>

</header>

<main>

<section class="courses-section">

<h1>Courses</h1>

<table class="courses-table">

<thead>

<tr>

<th>ID</th>

<th>Course Name</th>

<th>Instructor</th>

<th>Credits</th>

<th>Actions</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Introduction to Programming</td>

<td>John Doe</td>

<td>4</td>

<td>

<button class="edit-button">Edit</button>

<button class="delete-button">Delete</button>

</td>

</tr>

<tr>

<td>2</td>

<td>Calculus I</td>

<td>Jane Smith</td>

<td>3</td>

<td>

<button class="edit-button">Edit</button>

<button class="delete-button">Delete</button>

</td>

</tr>

<!-- Add more course records as needed -->

</tbody>

</table>

</section>

</main>

<footer>

<p>&copy; 2024 Student Management System. All rights reserved.</p>

</footer>

</body>

</html>

**Styles.css**

body {

margin: 0;

font-family: Arial, sans-serif;

background-color: #f2f2f2;

}

header {

display: flex;

justify-content: space-between;

align-items: center;

padding: 20px;

background-color: #5cb85c;

color: #fff;

}

header .logo {

font-size: 1.5em;

font-weight: bold;

}

header nav ul {

list-style: none;

display: flex;

gap: 20px;

margin: 0;

padding: 0;

}

header nav ul li {

display: inline;

}

header nav ul li a {

color: #fff;

text-decoration: none;

}

main {

padding: 20px;

}

.profile-section {

margin: 0 auto;

max-width: 600px;

background-color: #fff;

padding: 20px;

border-radius: 5px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

}

.profile-section h1 {

text-align: center;

margin-bottom: 20px;

}

.profile-form label {

display: block;

margin-bottom: 5px;

font-weight: bold;

}

.profile-form input,

.profile-form textarea {

width: 100%;

padding: 10px;

margin-bottom: 20px;

border: 1px solid #ccc;

border-radius: 3px;

}

.profile-form textarea {

resize: vertical;

}

.save-button {

width: 100%;

padding: 10px;

background-color: #5cb85c;

color: #fff;

border: none;

border-radius: 3px;

cursor: pointer;

}

.save-button:hover {

background-color: #4cae4c;

}

footer {

text-align: center;

padding: 10px;

background-color: #5cb85c;

color: #fff;

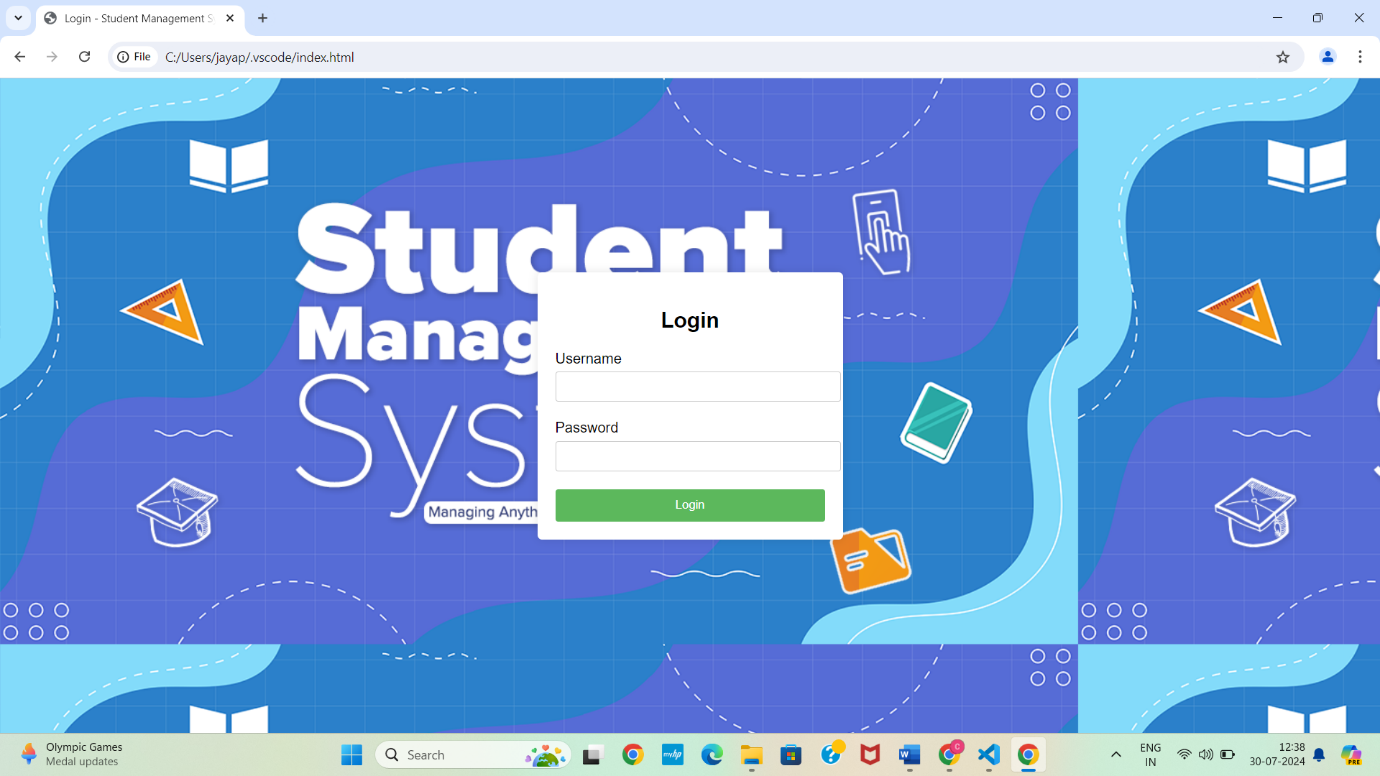
position: fixed;

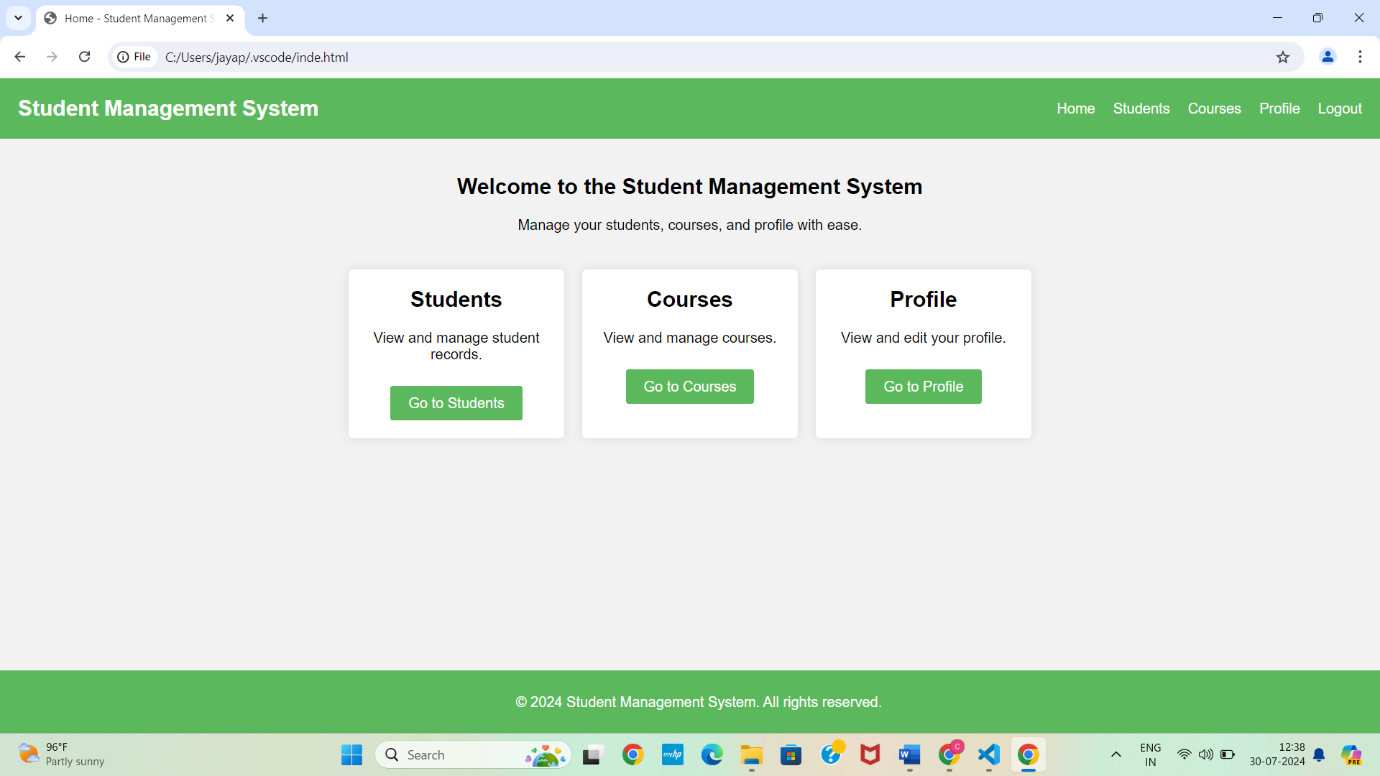
width: 100%;

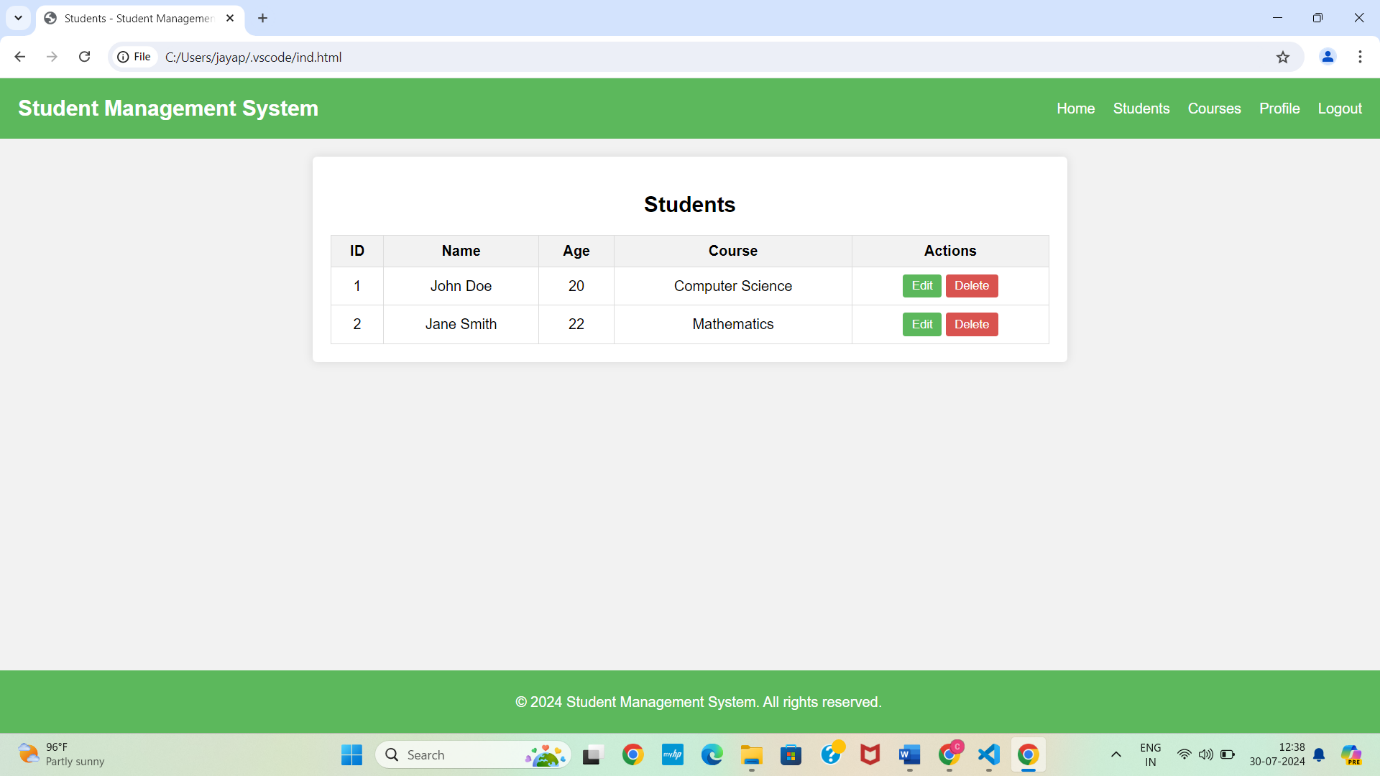
bottom: 0;

}

**OUTPUT:**







**Conclusion:**

The Student Management System (SMS) using PHP and MySQL provides a comprehensive and efficient solution for managing student information within educational institutions. Its modular design ensures streamlined administrative tasks, accurate data management, and improved communication among administrators, teachers, and students. By integrating essential functionalities such as user authentication, academic records management, attendance tracking, and reporting, the system enhances operational efficiency and supports informed decision-making. Overall, the SMS offers a reliable platform for managing educational processes effectively, contributing to better educational outcomes and improved institutional administration.

**Future Enhancements:**

1. Mobile Application: Develop dedicated mobile apps for Android and iOS to enhance accessibility and user experience for students, teachers, and parents.

2. Integration with Learning Management Systems (LMS): Integrate with LMS platforms like Moodle or Blackboard to offer a unified solution for both administrative and academic needs.

3. Advanced Analytics and Reporting: Implement advanced analytics and machine learning for predictive insights and comprehensive reporting on student performance and trends.

4. Online Payment Integration: Incorporate online payment gateways to facilitate secure and convenient fee payments directly through the system.

5. Automated Attendance via RFID/Biometric Systems: Introduce RFID or biometric systems for automated and accurate attendance tracking.

6. Parent Portal: Create a dedicated portal for parents to monitor their child's academic progress, attendance, and other relevant information.

**References:**

* Walia, E. S., & Gill, E. S. K. (2014). A framework for web based student record management system using PHP. International Journal of Computer Science and Mobile Computing, 3(8), 24-33.
* Bharamagoudar, S. R., Geeta, R. B., & Totad, S. G. (2013). Web based student information management system. International Journal of Advanced Research in Computer and Communication Engineering, 2(6), 2342-2348.
* Pratama, A. F., & Manurung, O. K. (2023). Web-Based Student Lecture Data Information System Using PHP and MySQL. Jurnal Pendidikan LLDIKTI Wilayah 1 (JUDIK), 3(01), 23-28.
* Madora, H., Wahyunigrum, I., & Noval, M. (2021, February). PHP Programming for Achieving Students in Informatics Management Department State of Polytechnic Sriwijaya. In 4th Forum in Research, Science, and Technology (FIRST-T1-T2-2020) (pp. 442-449). Atlantis Press.
* Wang, H., Xu, H., Li, Q., & Fu, Y. (2018). PHP‐based collaborative education and management system for water hydraulic laboratory. Computer Applications in Engineering Education, 26(2), 259-271