1.Project Title: Student Record System with Login and Registration
2.Objective:
The goal of the **Student Record System** project is to develop a web-based application that can manage student records securely. The system is designed to perform the following objectives:

- To create a platform where students' data can be stored, viewed, updated, and deleted.
- To provide a secure login and registration process for users to authenticate and access the system.
- To ensure a smooth and responsive user experience with a front-end design that is aesthetic and functional using **Bootstrap**.
- To allow CRUD (Create, Read, Update, Delete) operations on student records with appropriate user authorization.

3.Required Tool:
1. Xampp
2. Laptop
3. internet
4. VS code IDE

4. Theory:

4.1 Authentication System : The application features a secure login system where users need to authenticate using a username and a password. The password hashing mechanism ensures that passwords are not stored in plain text in the database. Instead, hashed passwords are compared with the user input during login.

4.2 CRUD Operations

The system allows users to interact with student records through CRUD operations. Each student has the following attributes:

- name
- course
- age

The operations are as follows:

- **Create**: Add new student records.
- **Read**: View the list of students in the system.
- **Update**: Modify existing student details.
- **Delete**: Remove student records from the system.

4.3 . Database Structure

Two main tables are used in the MySQL database:

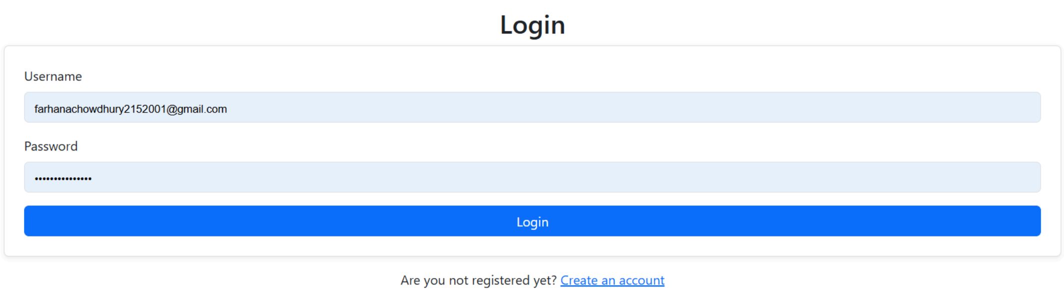
1. **users**: Stores user credentials (username, password).
2. **students**: Stores student information (id, name, course, age).

5. System Components

The system is composed of the following components:

4.1. Login Page (login.php)

A page where users enter their credentials. The entered password is verified against the hashed password stored in the database. If valid, the user is redirected to the home page (index.php).

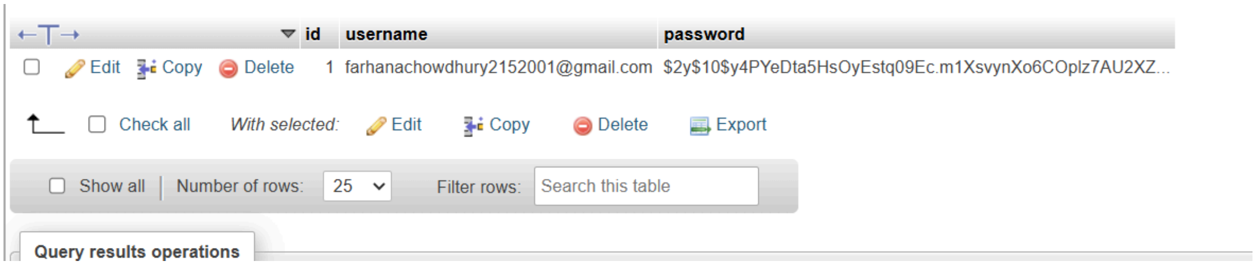


Login

Username	farhanachowdhury2152001@gmail.com
Password	•••••••••••••••

Are you not registered yet? Create an account

Fig-1: Login Page

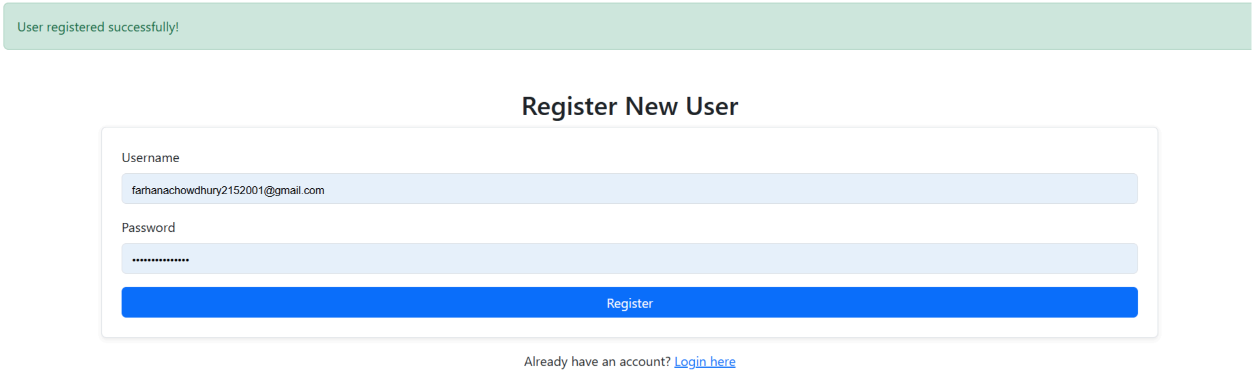


	id	username	password
□ Edit Copy Delete	1	farhanachowdhury2152001@gmail.com	$2y$10$y4PYeDta5HsOyEstq09Ec.m1XsvynXo6COplz7AU2XZ...

Fig-2:User Database

4.2. Registration Page (register.php)

New users can register by entering a username and password. The password is hashed before being stored in the database.



User registered successfully!

Register New User

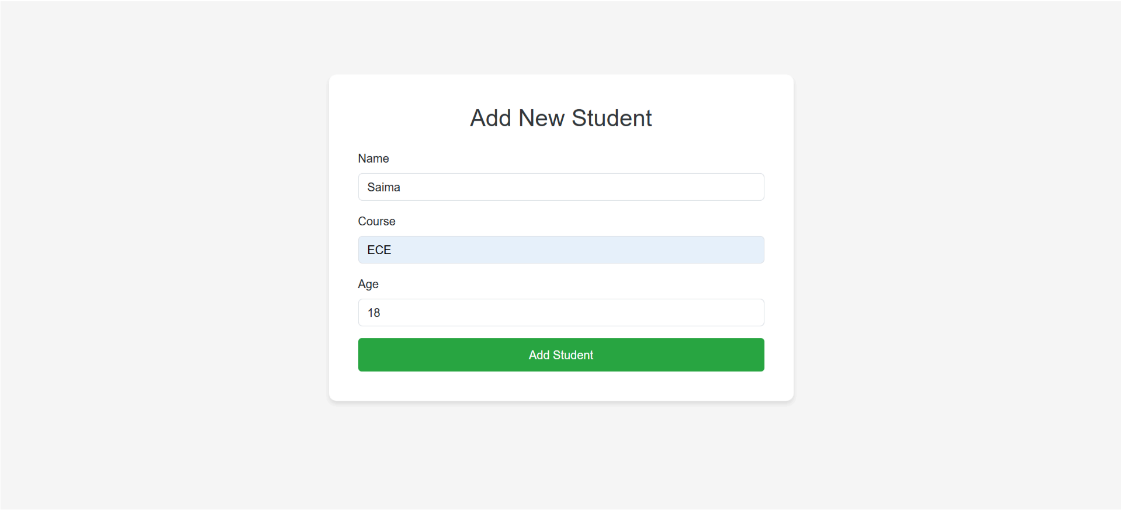
Username: farhanachowdhury2152001@gmail.com

Password: •••••••••••••••

Fig-3: Register Page

4.3. Student Management Pages

4.3.1. Add Student (add_student.php): A form to add new student records.



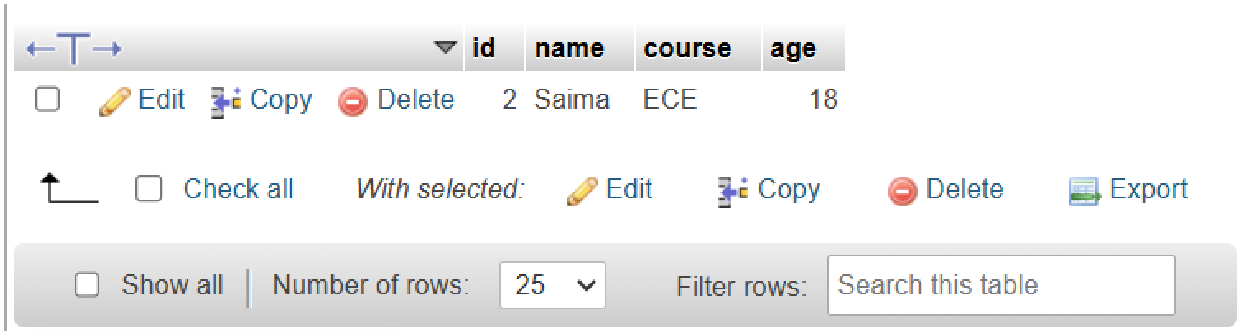
Add New Student

Name: Saima

Course: ECE

Age: 18

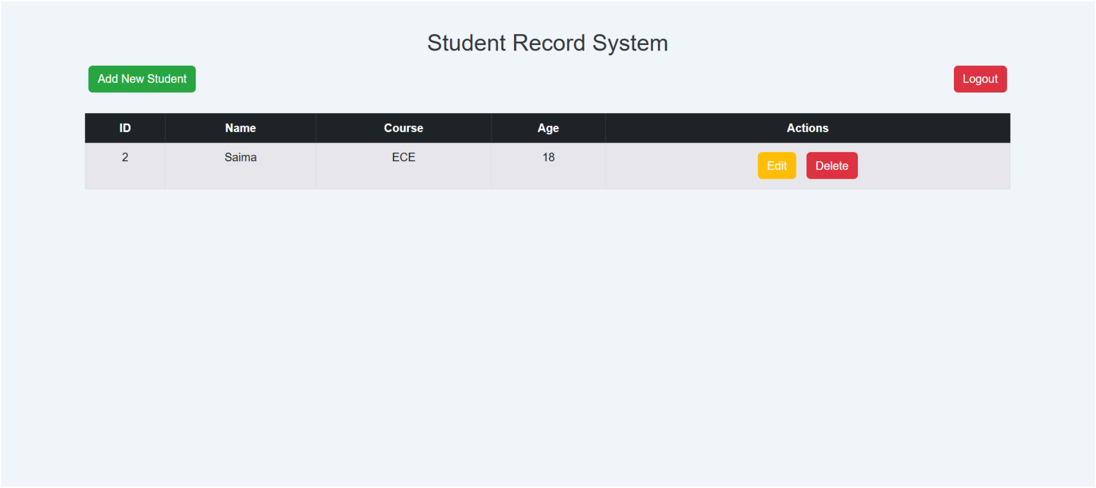
Fig-4: Add Student Page



id	name	course	age
2	Saima	ECE	18

Fig-5: Student Database

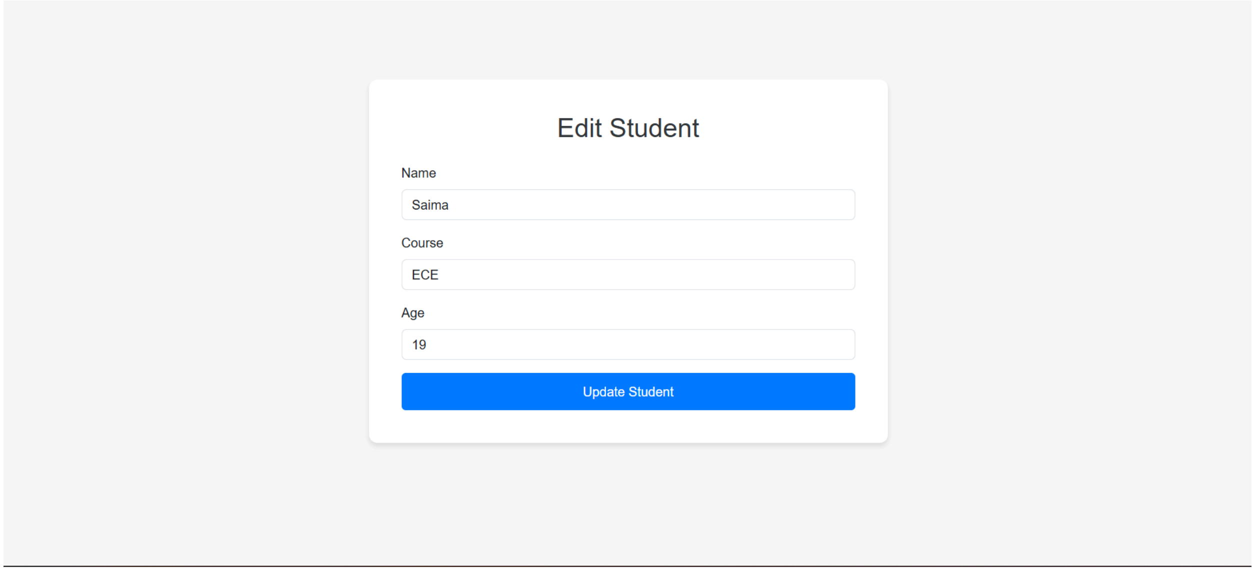
4.3.2 View Students (index.php): Displays a list of students from the database.


Student Record System

ID	Name	Course	Age	Actions
2	Saima	ECE	18	Edit Delete

Fig-6 : View Student page

4.3.3 Edit Student (edit_student.php): A form to edit existing student records.



Edit Student

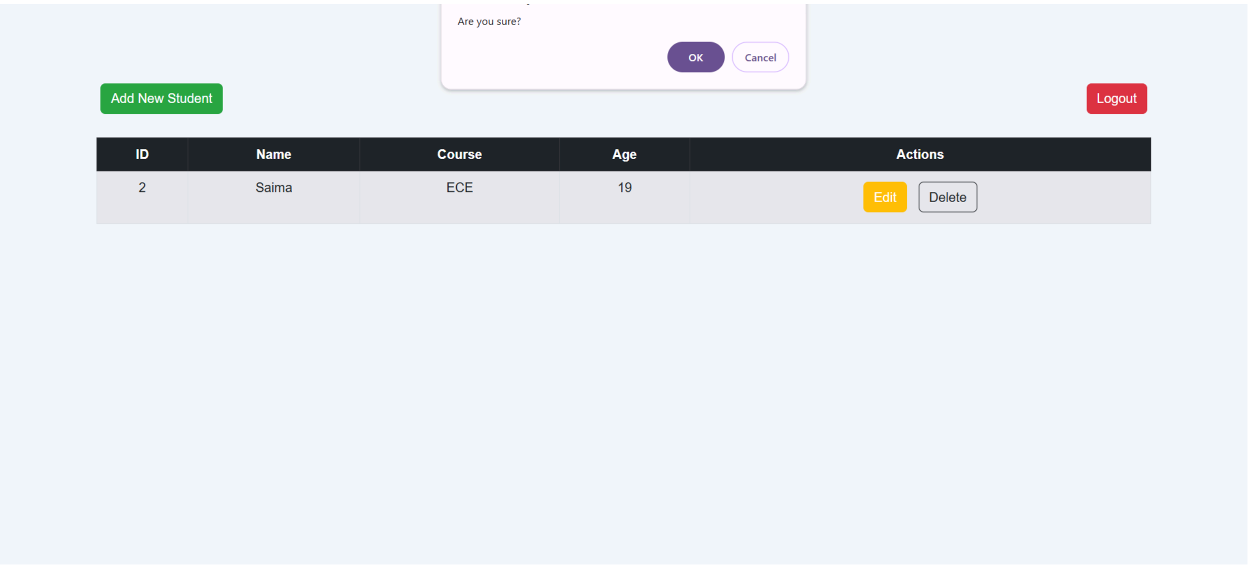
Name: Saima

Course: ECE

Age: 19

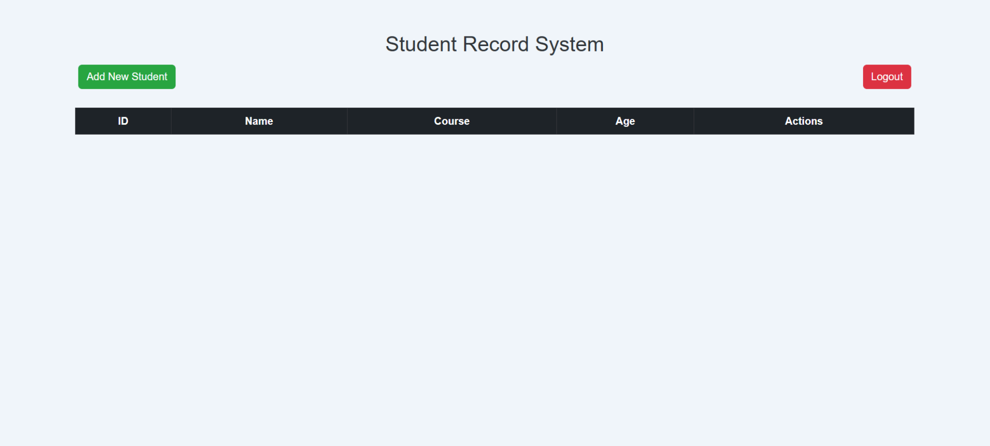
Fig-8 : Edit Student Page

4.3.4 Delete Student (delete_student.php): Deletes a student record based on the student ID.



Are you sure?

ID	Name	Course	Age	Actions
2	Saima	ECE	19	Edit Delete



Student Record System

ID	Name	Course	Age	Actions

Fig-9 : Delete Student Page

4.4. Logout (logout.php)

Logs out the user by destroying the session and redirecting to the login page.

---

5. Workflow:

The following is the step-by-step workflow of the system:

1. **User Registration**: The user accesses the registration page (register.php), submits the form with their username and password. The password is hashed, and the user is created in the users table.
2. **User Login**: The user accesses the login page (login.php), submits their username and password. The password is checked against the hash stored in the database. If valid, a session is created, and the user is redirected to index.php.
3. **Student Management**:
    - **View Students**: The user can view all student records in index.php.
    - **Add New Student**: The user can add new student records via the form on add_student.php.
    - **Edit Student**: The user can edit existing student records using the form on edit_student.php.
    - **Delete Student**: The user can delete a student record via the delete_student.php script.

4. **User Logout**: The user can log out using logout.php, which destroys the session and redirects them to the login page.

6. Technology Used:

The following tools and technologies were used in this project:

- **PHP**: For backend logic and database interactions.
- **MySQL**: For database management to store user and student data.
- **Bootstrap**: To design a responsive and aesthetically pleasing user interface.
- **HTML/CSS**: For page structure and styling.
- **Session Management**: To handle user login and session authentication using PHP $_SESSION.
- **Password Hashing**: For secure storage of user passwords using PHP's password_hash() and password_verify() functions.

7. Results

- **User Authentication**: The login and registration pages work effectively, ensuring only authorized users can access the system.
- **CRUD Operations**: The application correctly handles the creation, retrieval, updating, and deletion of student records, ensuring data integrity.
- **UI Design**: The user interface is responsive, modern, and visually appealing, thanks to the use of **Bootstrap**.
- **Password Security**: User passwords are securely hashed and stored, following industry standards for password security.
- **Functionality**: The system operates smoothly, with minimal errors or issues.

8. Conclusion

This **Student Record System** project meets the key objectives of managing student records securely and efficiently. By implementing basic CRUD operations and incorporating a secure login and registration system, the application is a robust and user-friendly solution for managing student data.

The use of **PHP** and **MySQL** enables seamless backend operations, while **Bootstrap** ensures the system is accessible and aesthetically pleasing. The application can be extended with additional features such as search and filtering, adding more user roles, and including advanced features like email notifications.

This project demonstrates how to create a secure, fully functional web application for managing student records, which can serve as a foundation for more complex systems in educational institutions.

9. Future Work

- **Search and Filter**: Adding functionality to search and filter student records based on various criteria.
- **User Roles**: Implementing different user roles, such as Admin and Student, to provide varied levels of access.
- **Profile Management**: Adding more detailed profiles for students, including contact information and academic performance.
- **Email Notifications**: Sending email notifications on actions like student registration or record updates.

This project can be scaled and enhanced in future iterations to provide additional functionality and greater security features.

10.References:

[1] W3C, "HTML5 and CSS3: Web Standards and Tools," *W3C Recommendations*, W3C, 2020. [Online]. Available: https://www.w3.org/TR/html5/

[2] A. S. Tanenbaum, "Database Systems: Design, Implementation, and Management," 7th ed., Boston, MA, USA: Pearson, 2017.

[3] P. J. Deitel and H. M. Deitel, "PHP and MySQL for Dynamic Web Sites: Visual QuickPro Guide," 4th ed., Berkeley, CA, USA: Peachpit Press, 2017.