Name:

Index Number:

NIC:

Photo

**Table of Contents**

**University Management System**

1. Introduction
2. Objective of the Project
3. Tools and Technologies Used
4. Project UI and Explanation  
    • Homepage  
    • Login Page  
    • Register Page  
    • Dashboard  
     ◦ Add Student  
     ◦ View All Students  
     ◦ Update Student  
     ◦ Delete Student  
     ◦ Search Student
5. Database Design  
    • Users Table  
    • Students Table
6. Challenges Faced and Solution
7. Submit Web Application Development Project
8. Future Improvements (Optional Section)
9. Conclusion

**University Management System**

**Web Application Development Project Report**

**1. Introduction**

This project, titled **“University Management System”**, is a comprehensive web-based application designed to streamline and automate the administrative tasks related to managing student data within a private educational institution. The core aim is to minimize manual processes, reduce paper-based records, and improve data accessibility and integrity.

The system provides a centralized platform for authorized users (such as campus administrators) to **register students, view and search student details, update existing records, and delete outdated or incorrect entries**. Additionally, the application supports **user authentication**, ensuring that only authorized users can access sensitive data.

The frontend is developed using **HTML and CSS** for structural and stylistic design, while **JavaScript** is used to enhance user interaction and validate forms. The backend is powered by **PHP**, a widely-used server-side scripting language, and data is stored in a **MySQL** relational database. All development and testing have been done using **WAMP server**, a local development environment that integrates Apache, MySQL, and PHP.

In essence, this project reflects real-world campus needs and offers a functional prototype that can be extended into a full-fledged student information system.

**2. Objective of the Project**

The **main objectives** of this project are outlined below:

* **To create an efficient and reliable student management system** that allows campus administrators to manage student records digitally.
* **To implement secure user authentication** using a login and registration system, ensuring that only authorized personnel can access and modify student data.
* **To design a user interface that is clean, responsive, and easy to navigate**, even for non-technical users.
* **To reduce the reliance on manual record-keeping** and prevent data redundancy or loss through digital data management.
* **To implement full CRUD functionality** (Create, Read, Update, Delete) for student records and store all data securely in a structured MySQL database.
* **To demonstrate the use of full-stack web development principles** with front-end and back-end technologies.

**3. Tools and Technologies Used**

The following tools and technologies were used in the design and implementation of this system:

| **Technology** | **Description** |
| --- | --- |
| **HTML** | Used for structuring the web pages, forms, and content |
| **CSS** | Used for styling the pages and making the layout visually appealing |
| **JavaScript** | Used for client-side validation, form interactivity, and dynamic UI elements |
| **PHP** | Handles server-side logic, form submissions, and database interactions |
| **MySQL** | Stores all user and student records in structured tables |
| **XAMP** | Local server environment (Windows, Apache, MySQL, PHP) for testing the project |

**Other optional tools used:**

* **VS Code**: As the primary code editor.
* **phpMyAdmin**: For managing MySQL databases visually.

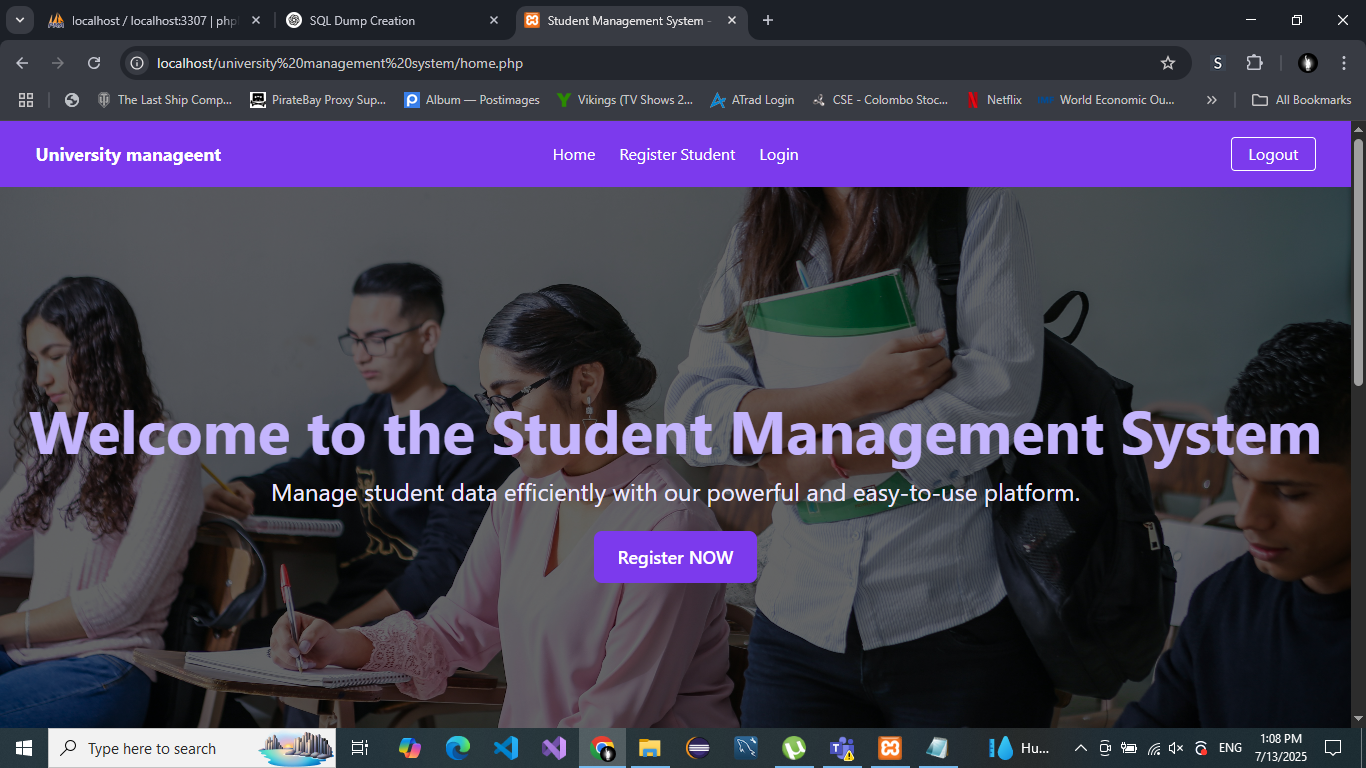
**4. Project UI and Explanation**

The User Interface (UI) of the Student Management System has been designed with **simplicity, clarity, and ease of use** in mind. The design follows a standard layout pattern that users are familiar with, such as navigation menus, form inputs, and content containers.

**Homepage:**

The **Homepage** is the landing page for the system. It includes a **navigation bar** with the following links:

* Home
* Login
* Register Student
* About
* Logout (visible only after login)

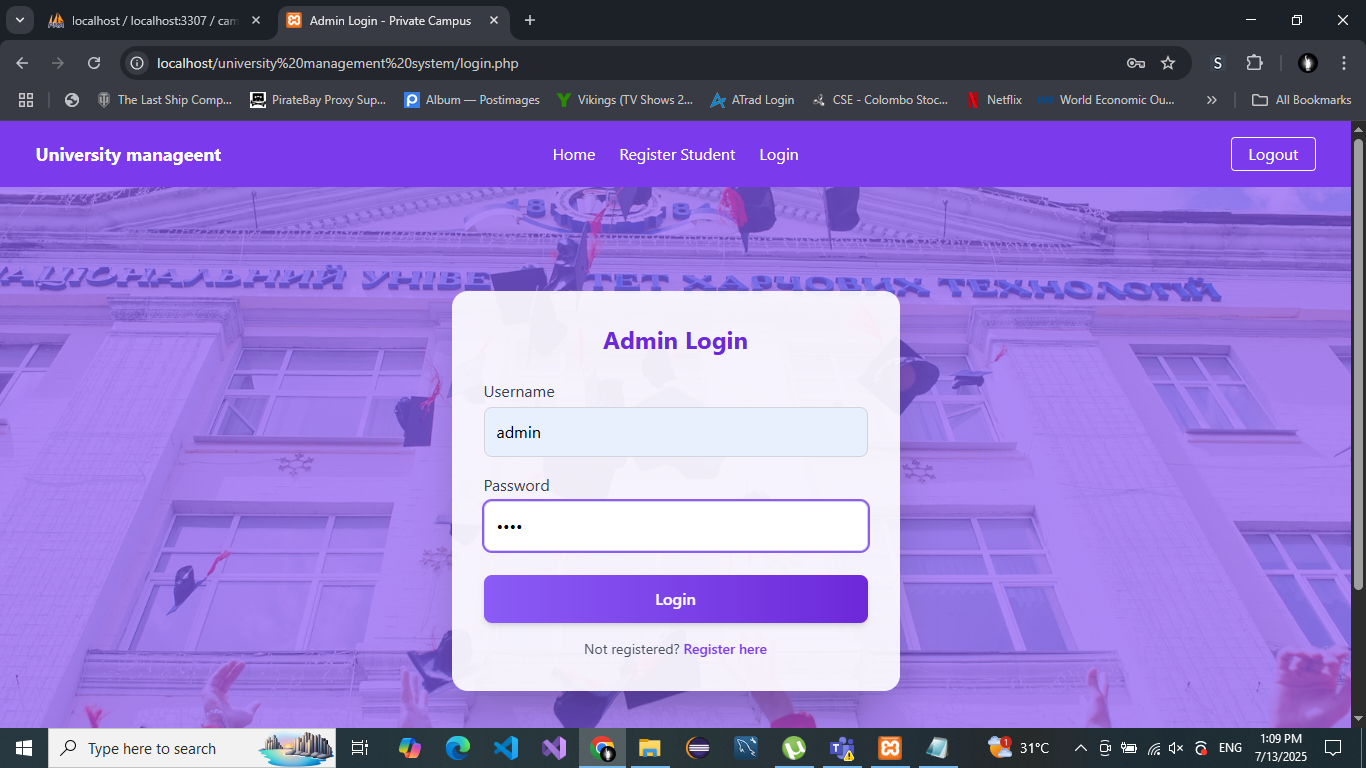


The hero section displays a **brief introduction** about the platform and includes a **Register Student** button that redirects users to the registration page.

**Login Page:**

The login page is a secure entry point for the admin:

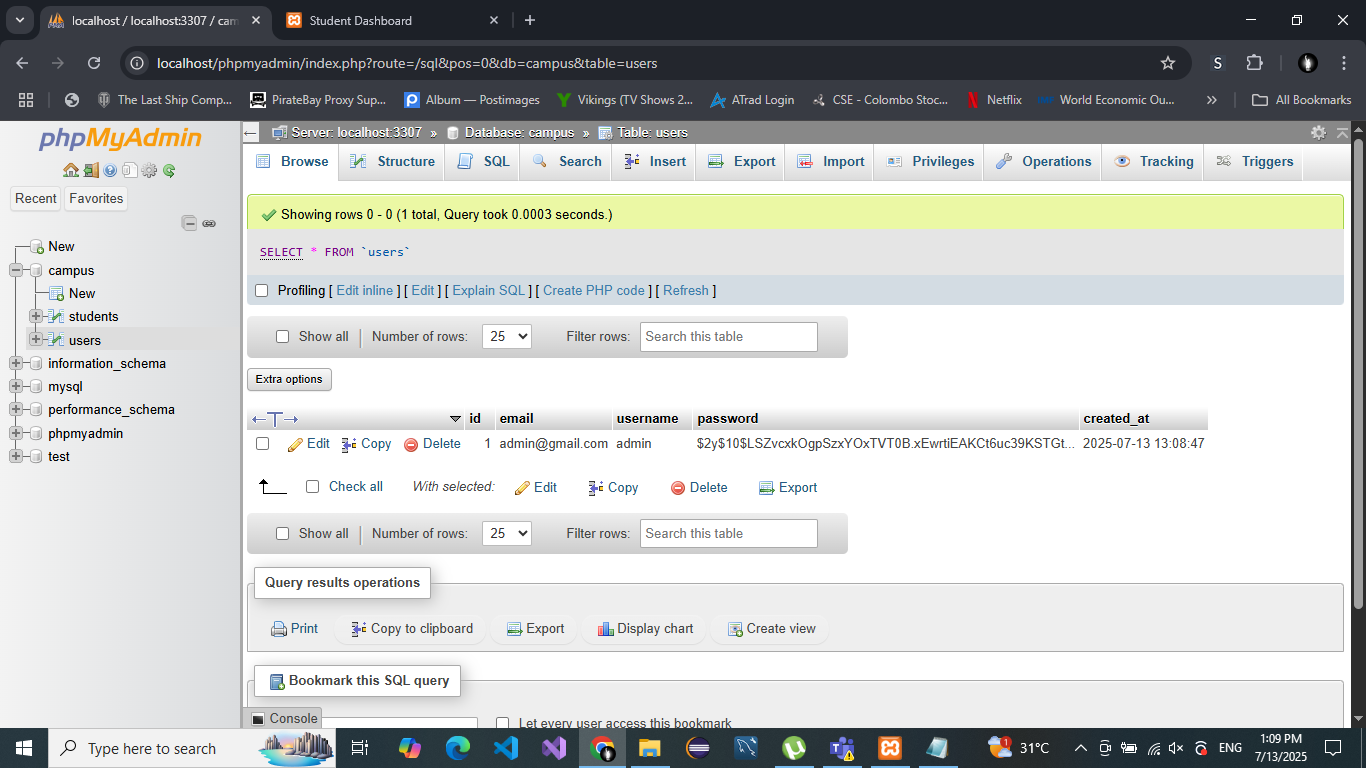
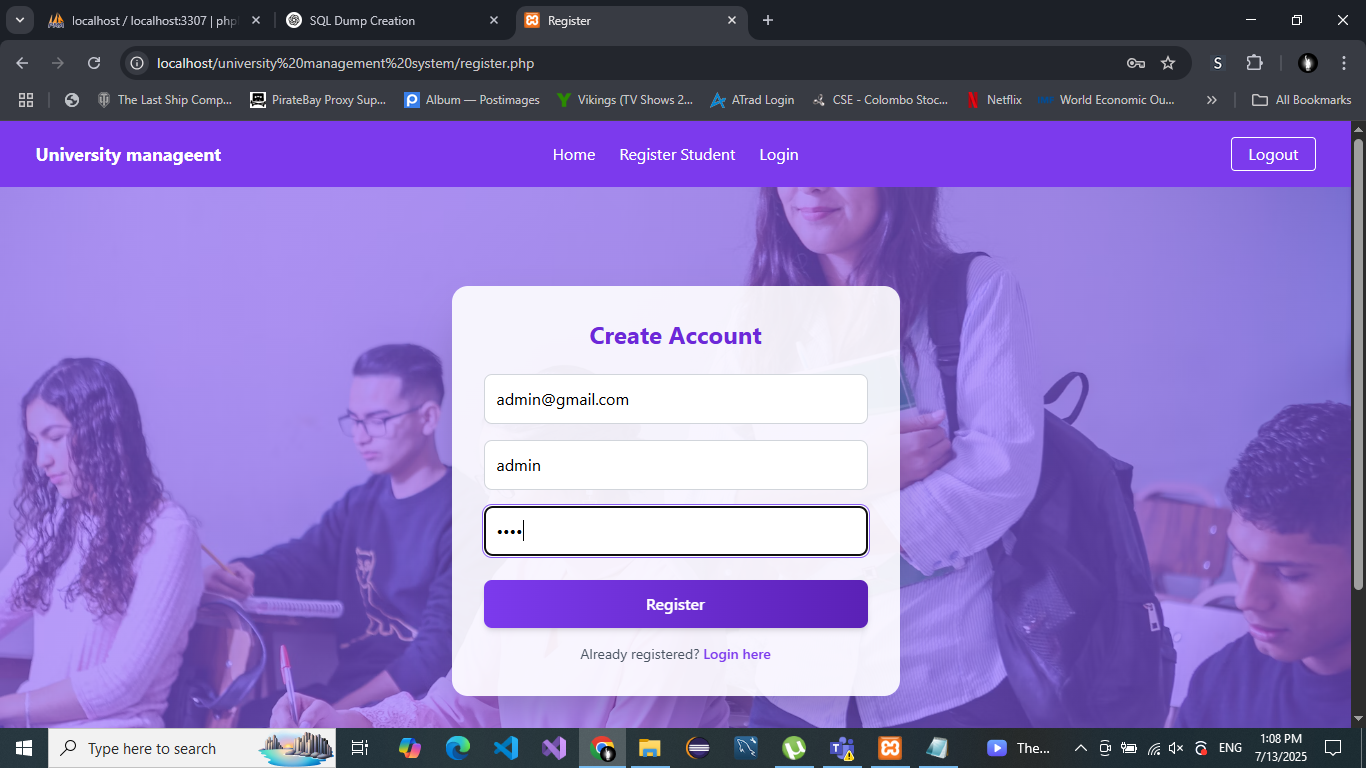
* Fields: Email, Password
* Upon clicking the **Login** button, a PHP script is triggered which checks the credentials.
* If correct, the user is redirected to the dashboard.
* If incorrect, an error message is displayed ("Invalid username or password").



**Register Page:**

Allows new admins to create an account:

* Fields: Email, Username, Password
* PHP handles form validation and inserts the data into the users table after sanitizing the inputs.

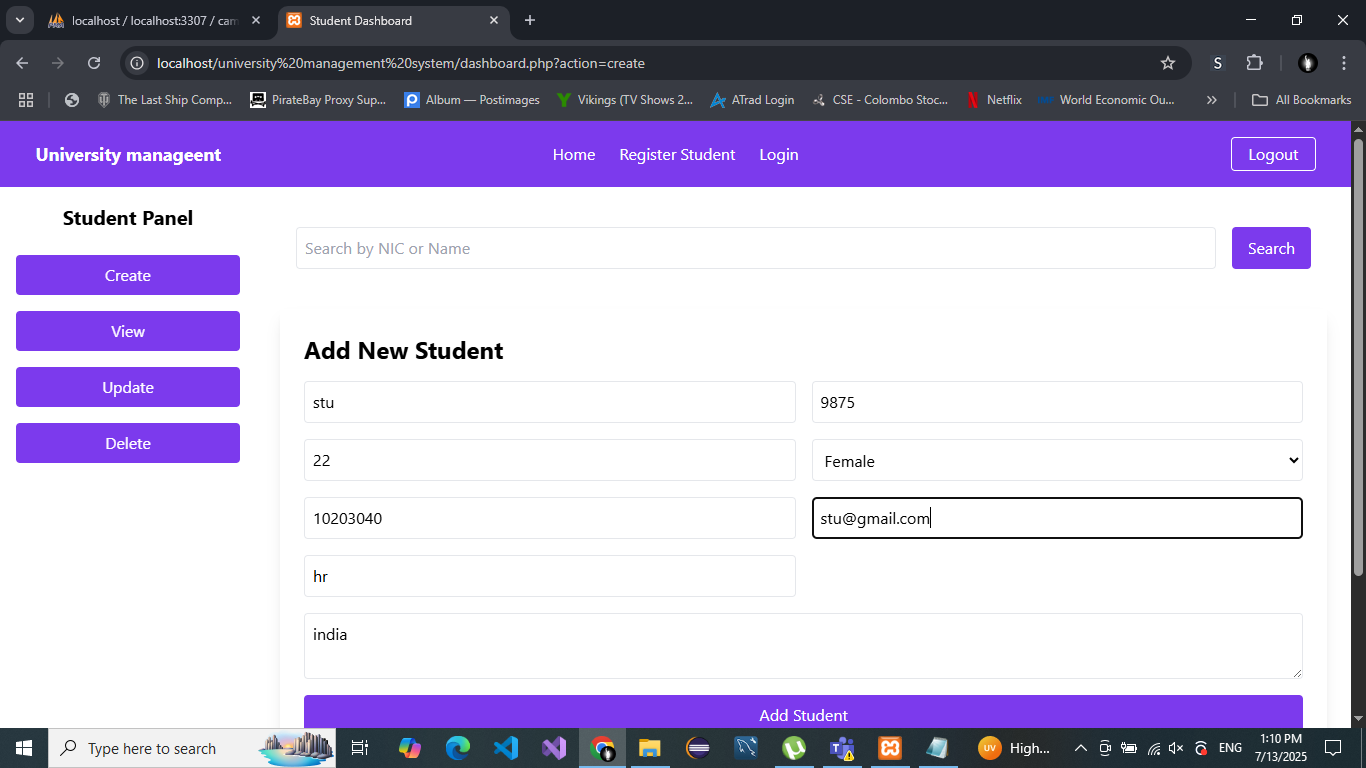


**Dashboard:**

The dashboard becomes accessible **after a successful login**. It includes a **sidebar navigation** with the following options:

* Add Student
* View All Students
* Update Student
* Delete Student
* Search Student

Each function is explained below:

****

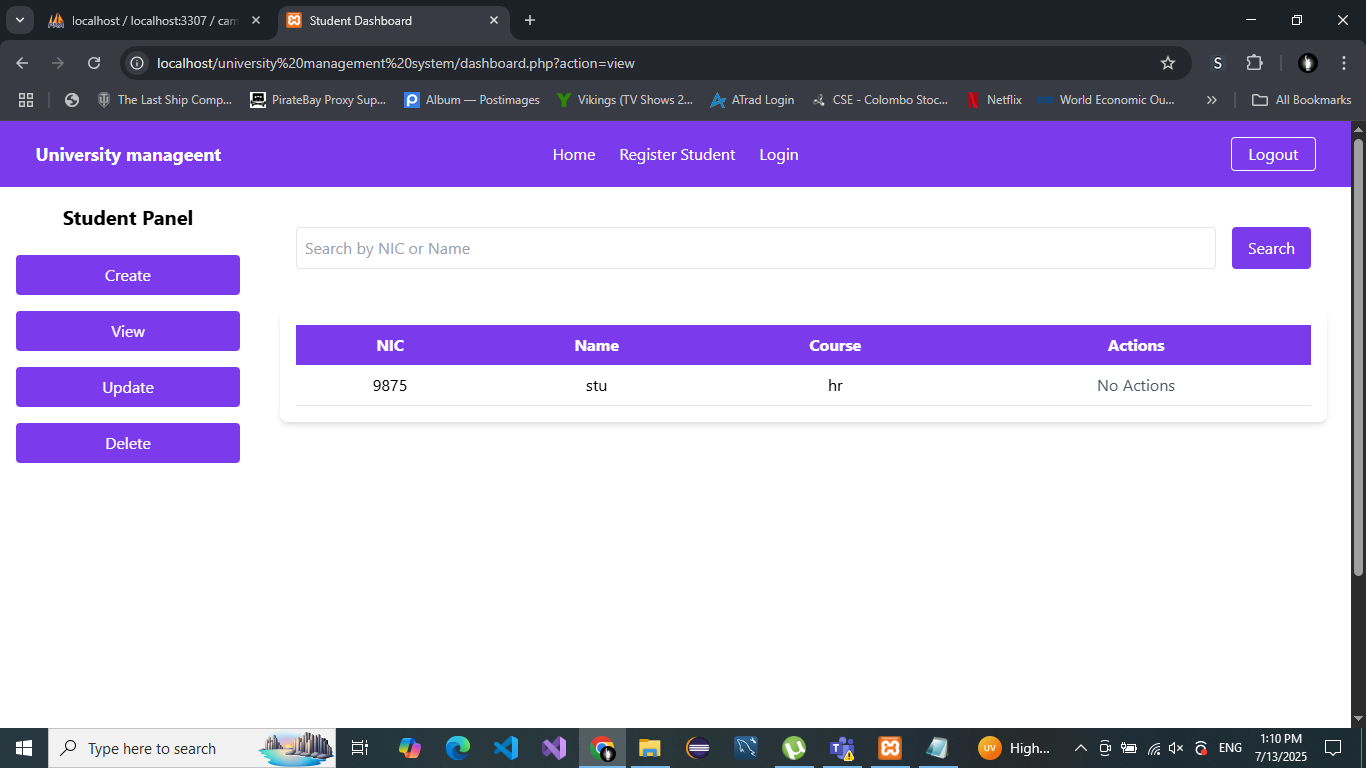
**Add Student:**

Admins can fill a form with fields such as:

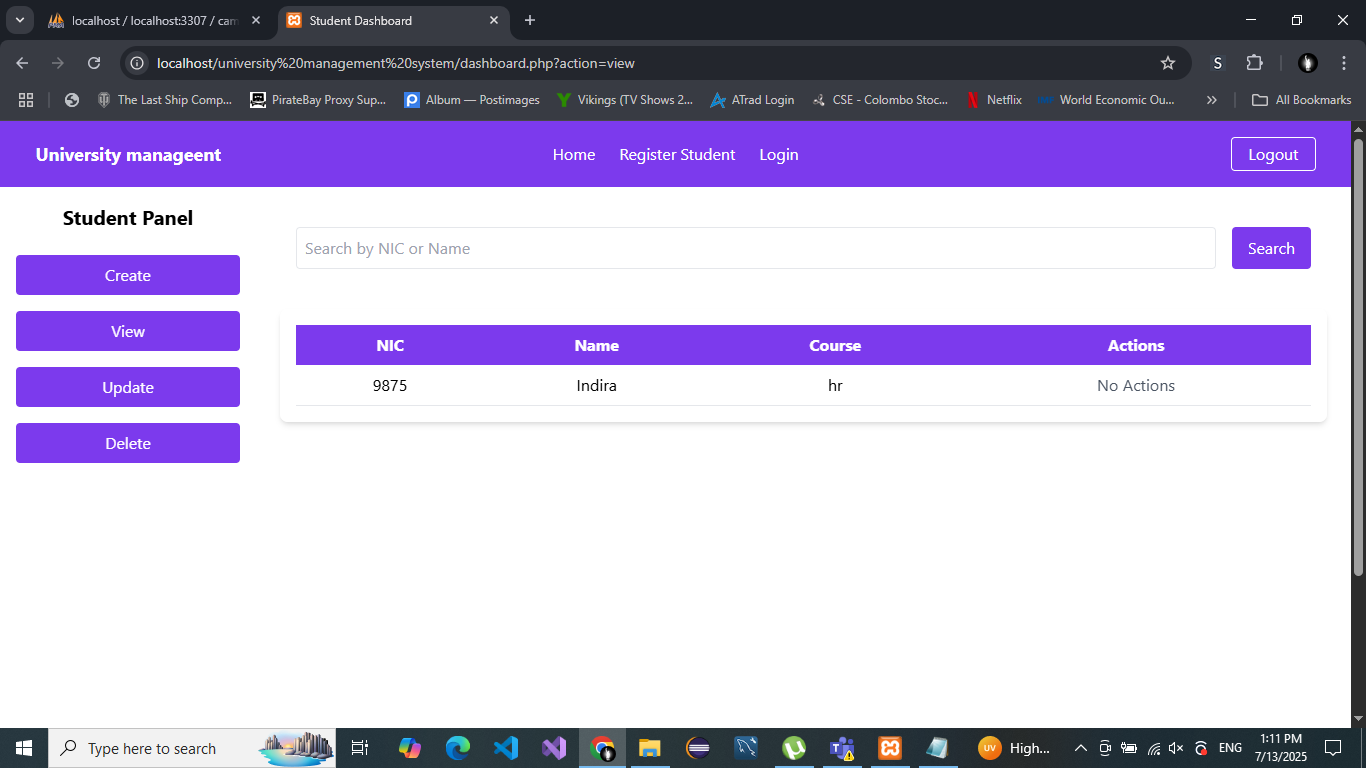
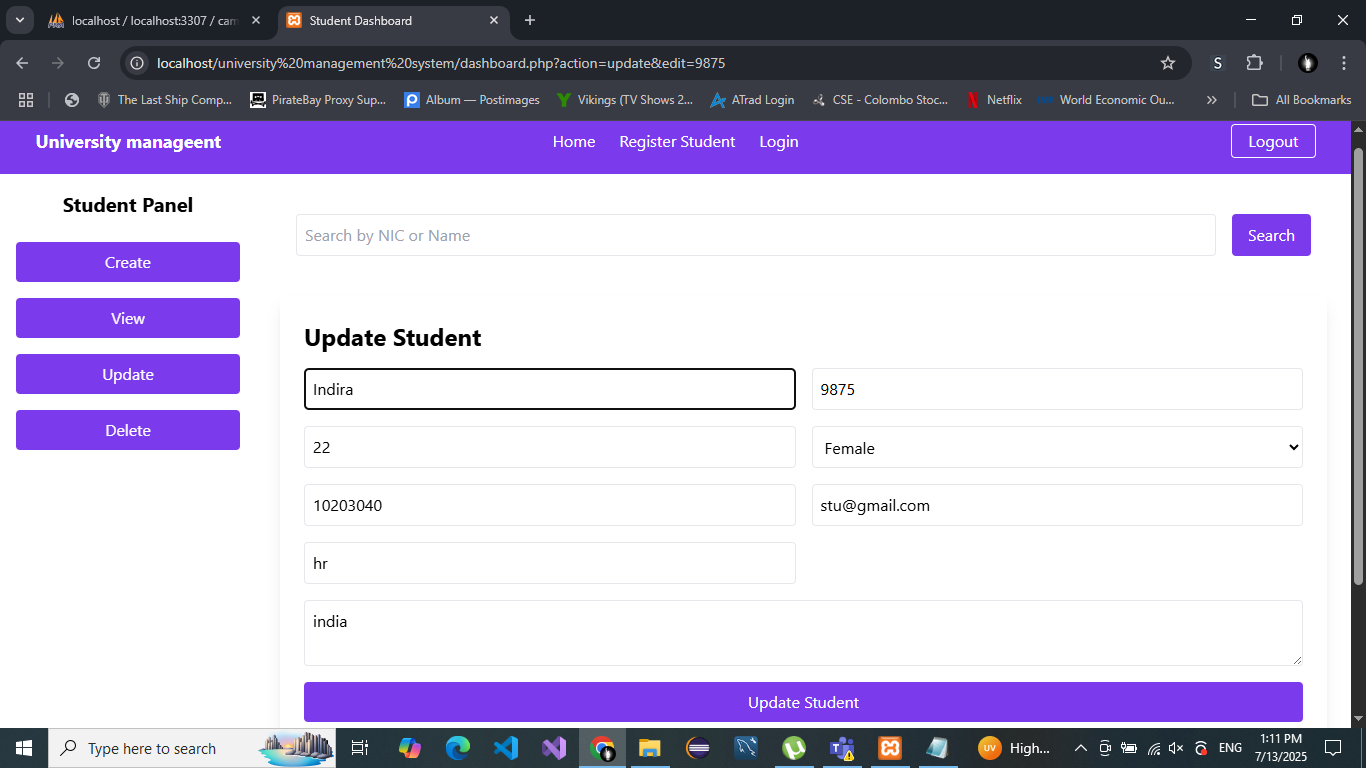
* NIC
* Name
* Gender
* Address
* Contact Number
* Email
* Course  
  On submission, the form data is validated and added to the database.

**View All Students:**

Displays all student records in a structured table with options to:

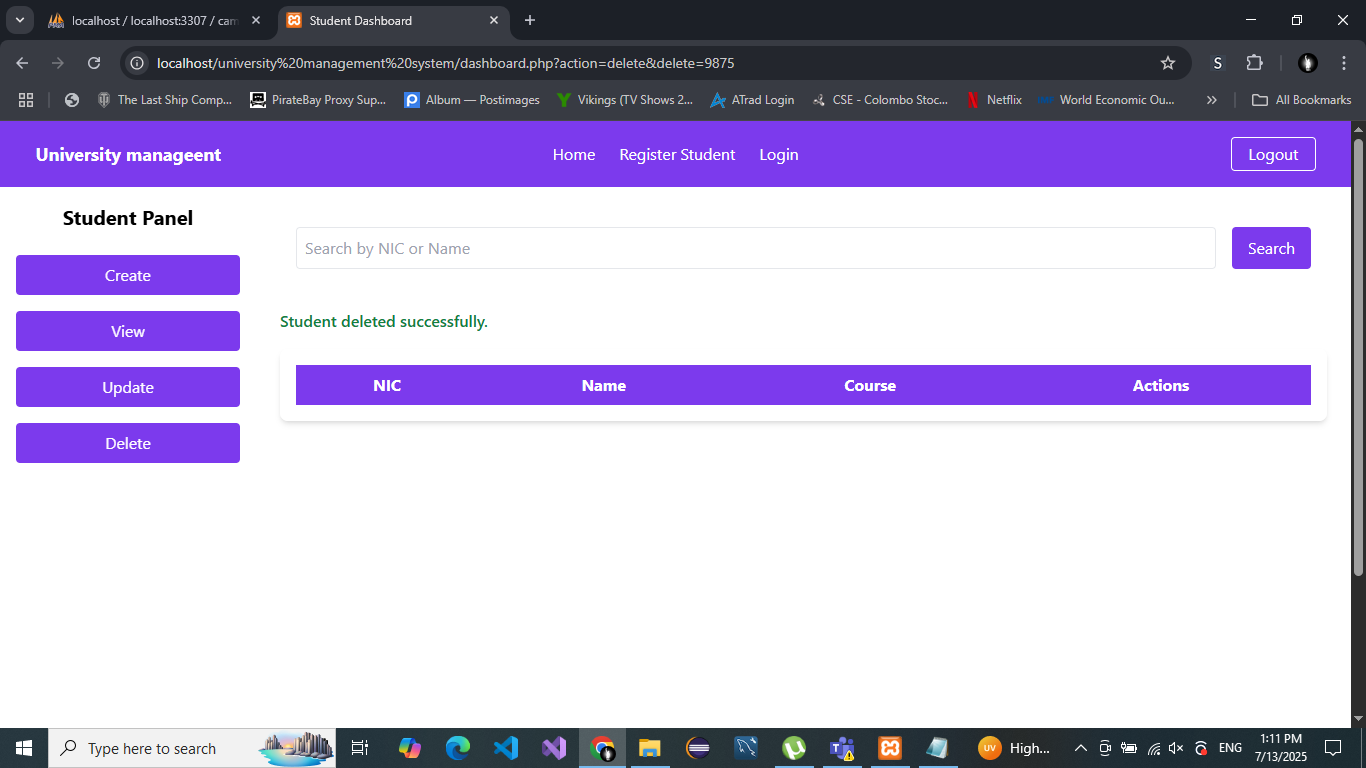
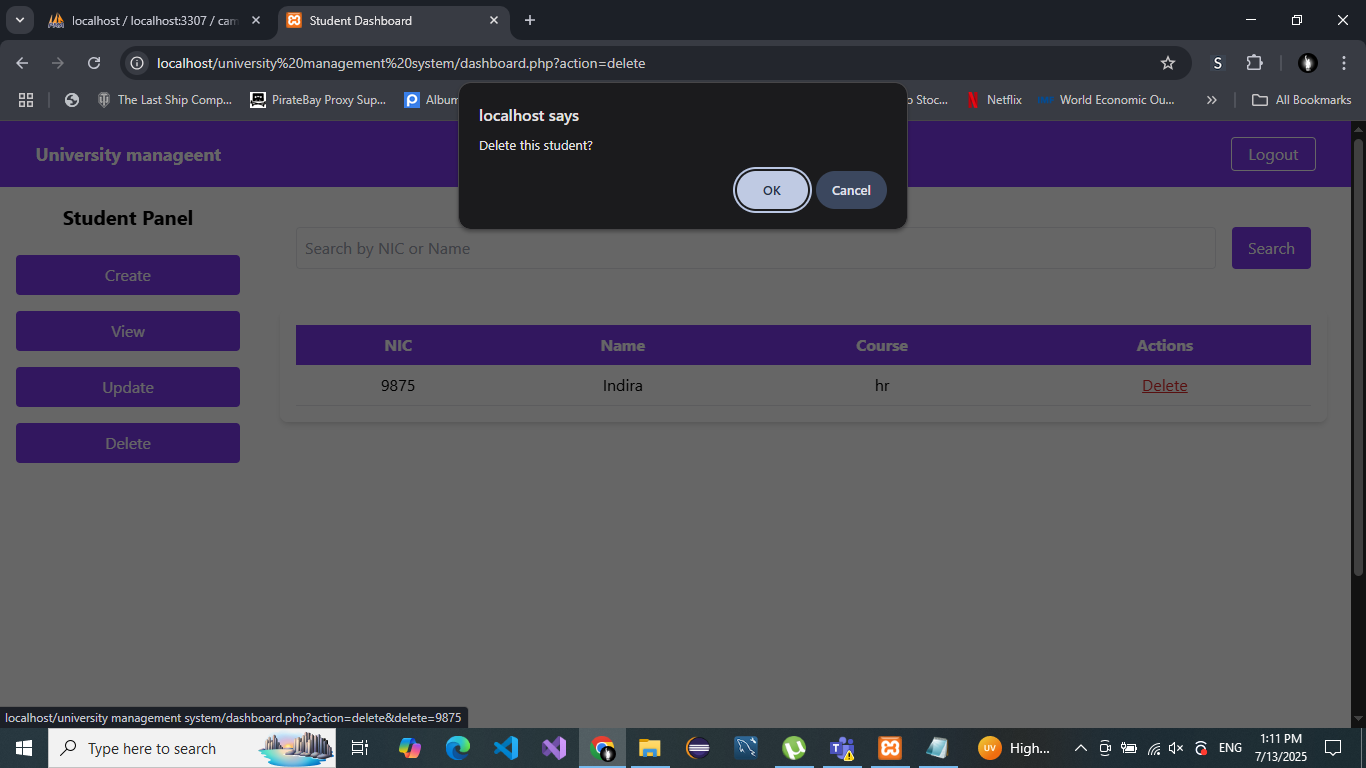


**Update Student:**

Allows admins to select a student and update their existing records. A form is pre-filled with current values. 

**Delete Student:**

Admins can remove student records permanently from the system.



**Search Student:**

Search by NIC or Name to quickly find student information using a search bar.

All pages are fully responsive and styled using CSS for a clean and professional look.

**5. Database Design**

The system uses two main tables:

**Users Table (for authentication)**

| **Field** | **Type** |
| --- | --- |
| id | INT (Primary Key, Auto-increment) |
| email | VARCHAR(100) |
| username | VARCHAR(100) |
| password | VARCHAR(255) (hashed) |

**Students Table**

| **Field** | **Type** |
| --- | --- |
| id | INT (Primary Key, Auto-increment) |
| nic | VARCHAR(20) |
| name | VARCHAR(100) |
| gender | VARCHAR(10) |
| address | TEXT |
| contact | VARCHAR(15) |
| email | VARCHAR(100) |
| course | VARCHAR(100) |

This simple relational schema ensures all data is properly indexed and normalized.

**6. Challenges Faced and Solution**

| **Challenge** | **Solution** |
| --- | --- |
| PHP-MySQL connection failed | Verified dbconnect.php, confirmed XAMP server was running, fixed credentials |
| Password not storing securely | Used password\_hash() and password\_verify() for hashing and checking |
| Form data vulnerable to SQL Injection | Used mysqli\_prepare() with bound parameters to prevent injection attacks |
| Responsive design for mobile devices | Applied media queries and flexbox layout techniques in CSS |
| Dashboard UI becoming cluttered | Restructured the layout using sidebars and collapsible menus |
| Search function returning wrong results | Updated search query logic to use LIKE and % wildcard symbols |
| Error messages not displaying | Added session-based error handling and alerts using PHP sessions |

These challenges helped us better understand real-world web development debugging and error handling practices.

**7. Submit Web Application Development Project**

**Files Included in Submission:**

* home.html – Homepage
* login.php – Handles login functionality
* register.php – Handles new user registration
* dashboard.php – Main admin dashboard
* dbconnect.php – MySQL database connection
* campus.sql – MySQL dump file for table creation
* Student\_Management\_WebApp\_Report.docx – This report

**8. Future Improvements (Optional Section)**

If this project were to be extended further, the following features could be added:

* Multi-user roles (e.g., admin, teacher, student)
* Password reset with email verification
* Pagination on view pages for better UX
* Export student data to CSV or PDF
* Notification system for important updates
* Mobile app integration using React Native or Flutter

**Conclusion**

This **University Student Management System** web application is a practical and scalable solution for managing student information in a digital format. It improves efficiency, reduces human error, and enhances data security. The system not only fulfills current administrative needs but also lays the groundwork for more advanced features in the future.

This project allowed us to apply full-stack development skills, from designing frontend pages to securing backend interactions and handling real-world development challenges.