Names: Ishimwe Thierry

Student I.D: 23756

WEB-TECHNOLOGY FINAL PROJECT

**[ PART 6]**

Technical Documentation: Student Conduct System

Introduction:

The Student Conduct System is a software application designed to manage and track students' behavior in school. It provides a centralized platform for recording and monitoring disciplinary actions, generating reports, and communicating with parents regarding their children's conduct. This technical documentation aims to provide an overview of the architecture, implementation details, libraries/frameworks used, and other relevant technical information about the application.

Architecture:

The Student Conduct System follows a client-server architecture, consisting of the following components:

Client Interface: This component allows school administrators and staff to access the system through a user-friendly web-based interface. It provides features for managing student profiles, recording disciplinary actions, generating reports, and sending messages to parents.

Server: The server-side of the application handles the processing and storage of data. It manages the application logic, database interactions, and communication with external services (e.g., email or SMS gateway for sending messages to parents).

Database: The system utilizes a relational database management system (e.g., MySQL, PostgreSQL) to store student information, disciplinary records, user accounts, and other relevant data. The database is accessed and manipulated through the server-side application.

Implementation:

The Student Conduct System is implemented using the following technologies and programming languages:

Backend: The server-side logic is primarily developed using a backend programming language such as Python, Java, or PHP. It handles user authentication, data processing, database interactions, and communication with external services.

Frontend: The client-side interface is built using web technologies like HTML, CSS, and JavaScript. It provides an intuitive and responsive user interface for interacting with the system.

Frameworks and Libraries: The application may employ various frameworks and libraries to streamline development and enhance functionality. Examples include:

Backend Framework: Flask (Python), Spring (Java), or Laravel (PHP)

Frontend Framework: React, Angular, or Vue.js

Database ORM: SQLAlchemy (Python), Hibernate (Java), or Eloquent (PHP)

Message Sending to Parents:

To send messages to parents regarding their children's behavior, the system integrates with external communication services, such as email or SMS gateways. The specific implementation details may vary based on the chosen service provider. Here is a high-level overview of the process:

User Trigger: A school administrator or staff member initiates a message sending process from the system's user interface. They select the student(s) and compose the message.

Server-side Processing: The server receives the message request and verifies the user's authorization to send messages. It retrieves the parent's contact information from the database associated with the selected student(s).

Communication Service Integration: The server interacts with the chosen communication service (e.g., API integration) to deliver the messages. This could involve sending emails or SMS messages, based on the contact information available.

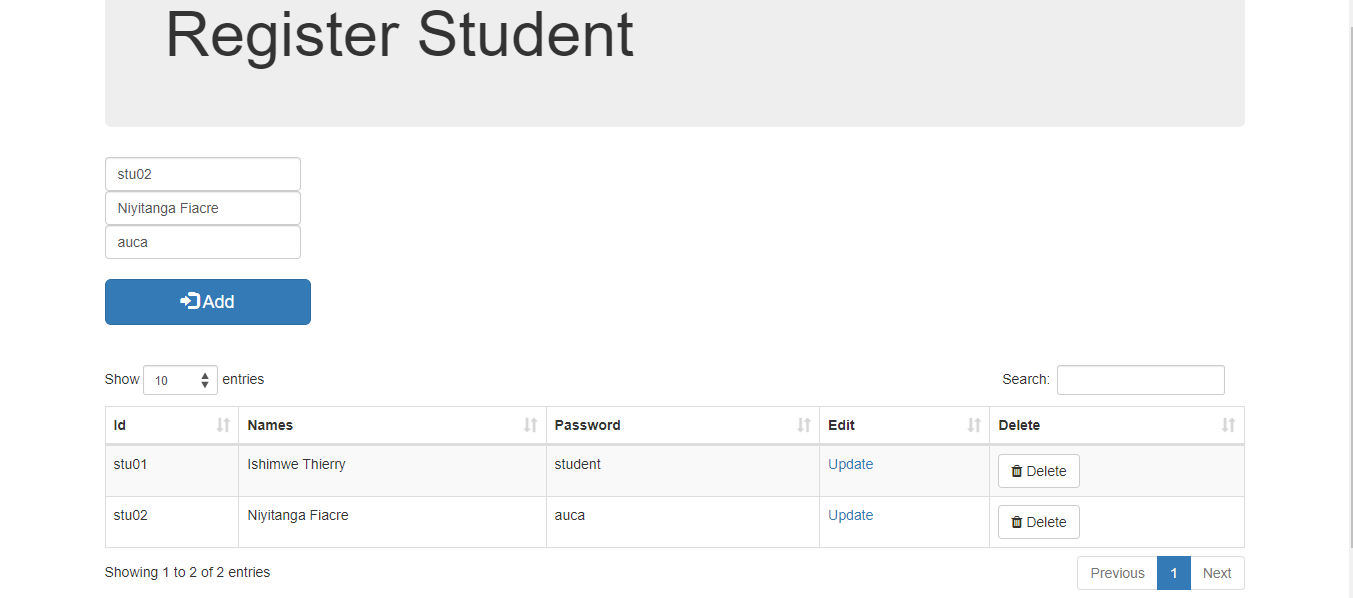
Status Tracking: The system keeps track of the message status (e.g., sent, delivered, failed) for auditing and monitoring purposes. This information is stored in the database for future reference.

Conclusion:

This technical documentation provides an overview of the architecture, implementation details, libraries/frameworks used, and the process of sending messages to parents in the Student Conduct System. It serves as a reference for developers, administrators, and other stakeholders involved in the application's development, deployment, and maintenance.

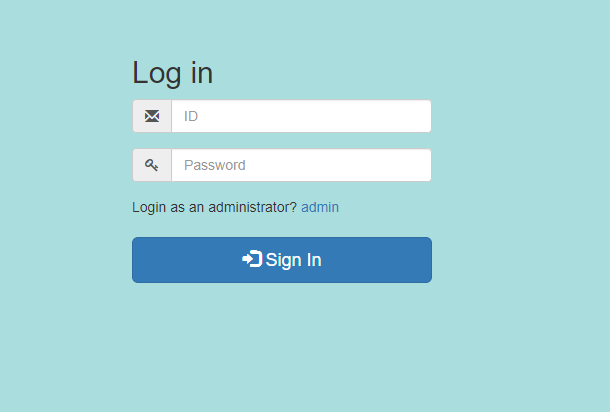
**Screenshots/ workflow**

* 1. **The system shall provide signup system for new students to register new accounts.**

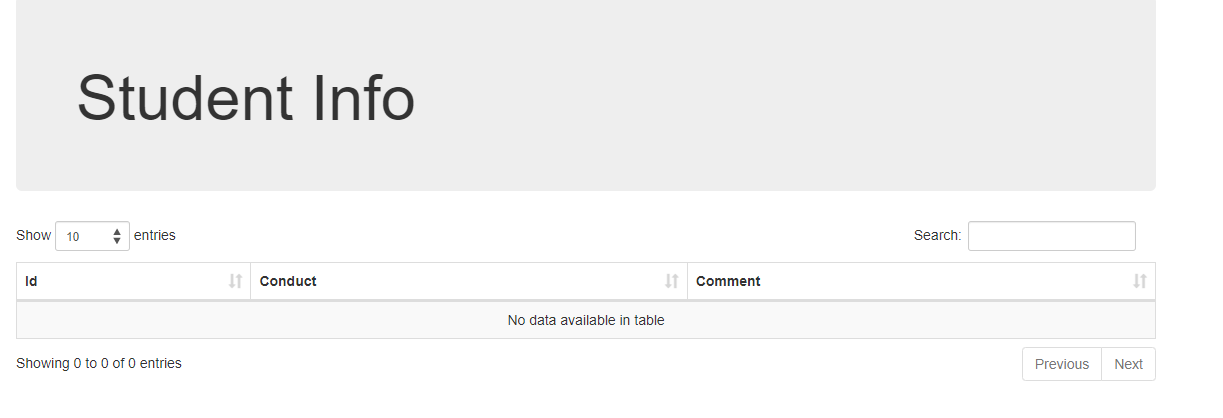
****

* 1. **The system must provide login system for returning university students with accounts.**

**Login form**

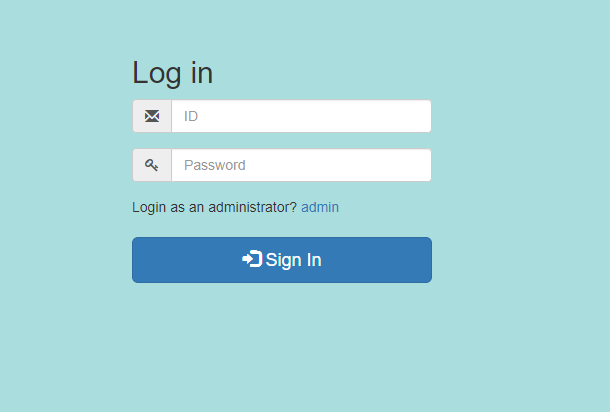


**After Logging into Student Account**



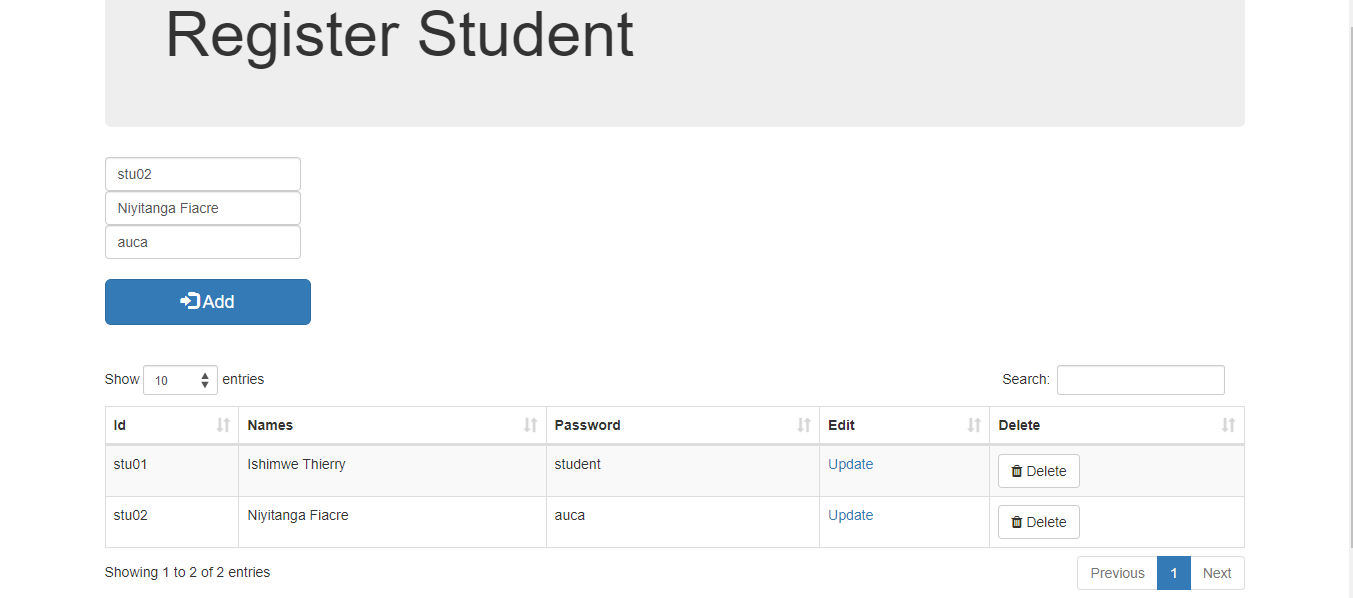
* 1. **The system shall provide a login system for the admin where they shall add,**

**delete, view students to the database.**

****

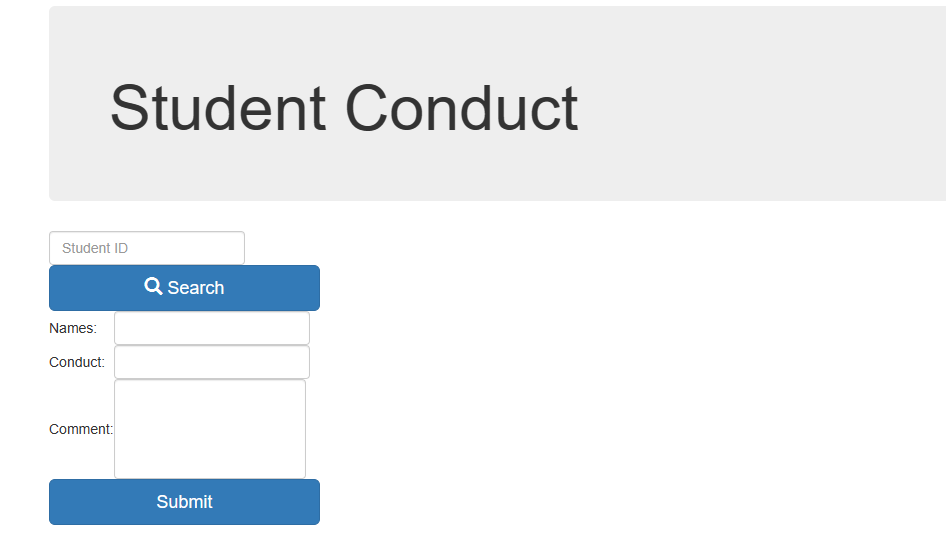
After inserting I clicked Display to see if it was inserted

1. I can also delete using the Delete Button (I deleted the first record for demonstration)

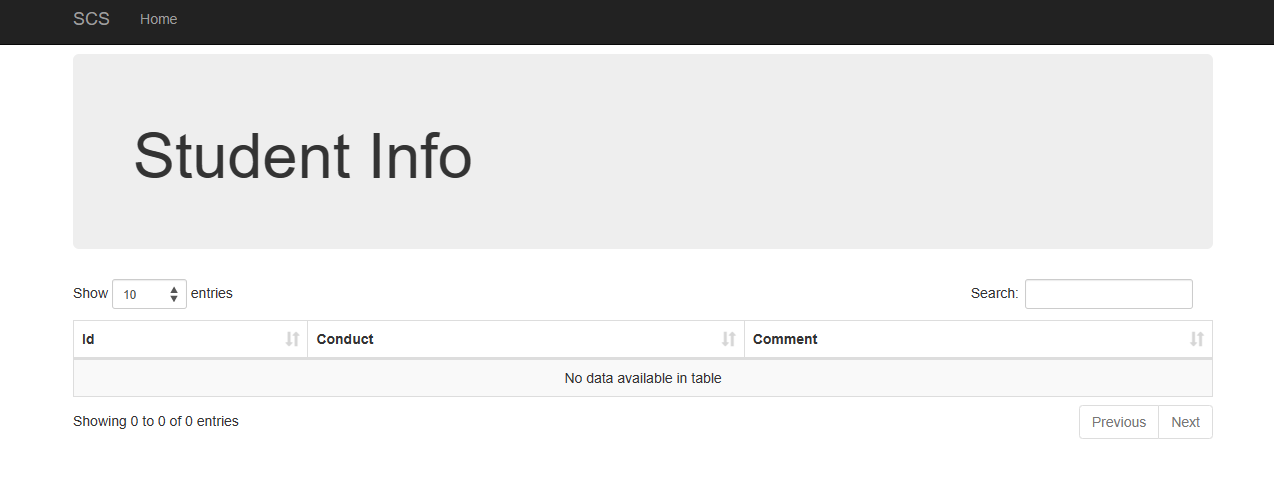


* 1. **The system shall provide the admin the capacity to view all students and to search for any student and admin can add conduct of students and comments about student’s behaviors.**

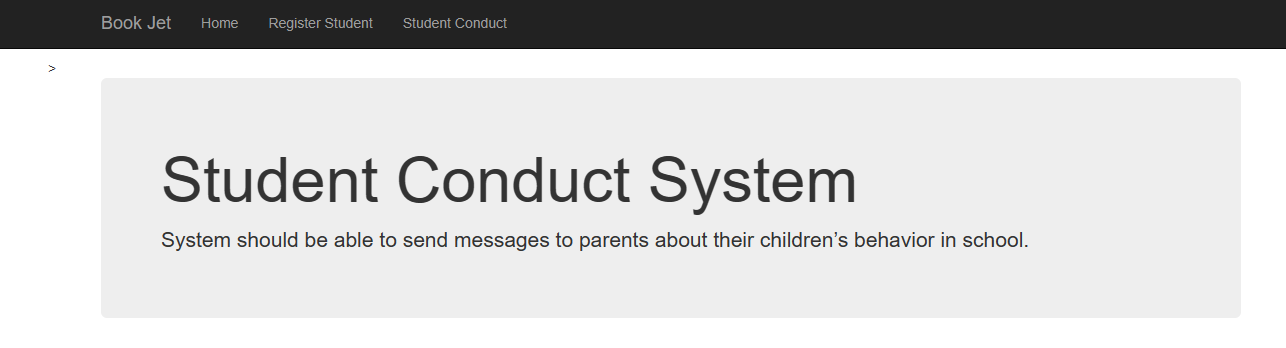
Once logged in as admin you click on View submission

****

Then parents or students can view all applications about students submitted by the admin or school:



**6. The system shall provide a home option to display available fields to website visitors.**



**[ PART 2]**

Using MVC architecture implement each feature, and for each feature provide a screenshots demonstrating the implementation of Model, View and Controller components. Use Ajax at least in sending one request.

Side Note: Before running the project please add the following jars: PostgreSQL and boots faces into the library in NetBeans (To avoid exceptions because we need to import them)

I added them inside the folder so that you can find them

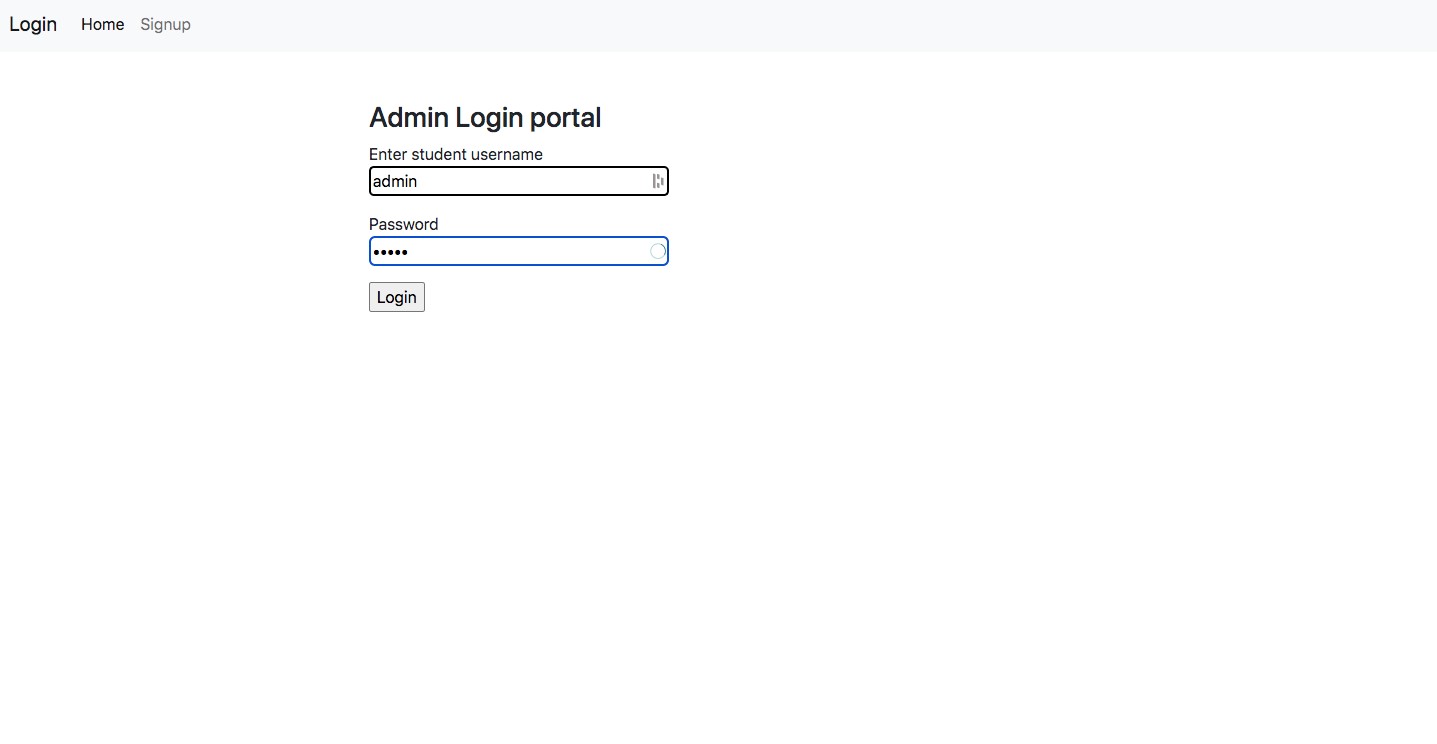
side by side with my project **23756\_Ishimwe\_Thierry\_WebTechProject**

**[ PART 3 ]**

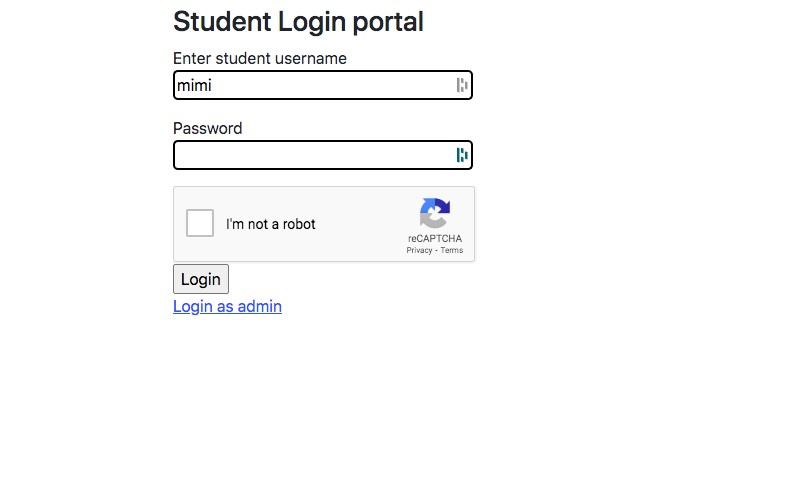
**Implement at least 2 non-functional (non-domain specific) features of your webapplication.**

1. Security: Students must have valid credentials to be authenticated.

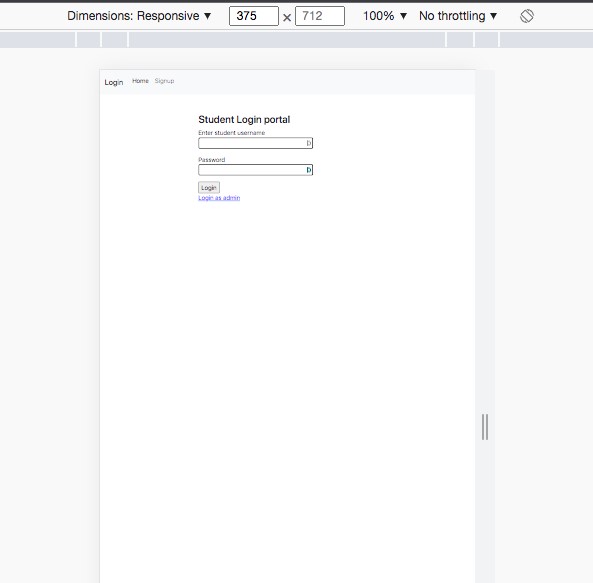
Here if I type a wrong password it will not allow to get into the account no matter how many tries

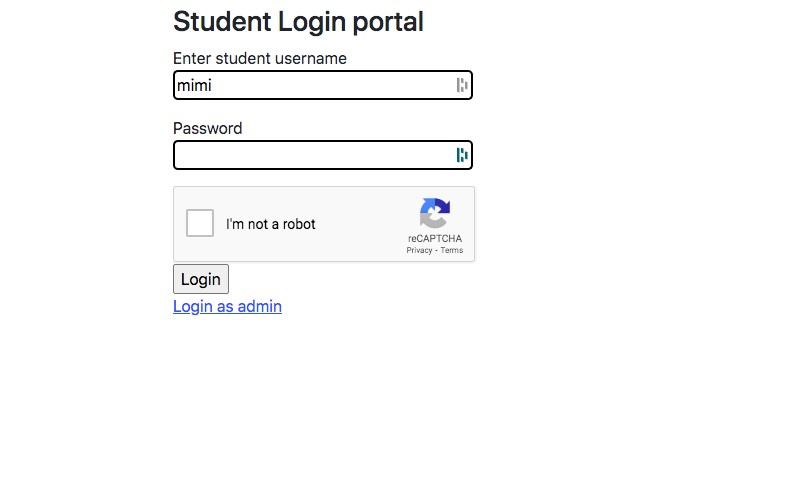


There is also captcha verification for added security



1. Responsive UI: The web application is responsive to smaller screens
   1. The navigation bar collapse with hamburger menu





On the login side of both returning students and admin.

Firstly I registered

Implementation in the project:

Inside my Login Bean (Login.java) class is where the validation is so that you must select the reCaptcha before logging in