A Project Report on

Student and Teacher Portal

Department of Computer Science and Engineering

Object-Oriented Programming Lab

CSE 2112

University of Dhaka

Submitted To:

Dr. Chowdhury Farhan Ahmed
Professor

Md. Redwan Ahmed Rizvee
Lecturer

Submitted By:

Group ID: 5

Name: Papry Rahman (01)
Md. Tauseef-Ur-Rahman (24)
Anirban Roy Sourav (32)
2nd Year 1st Semester
Session: 2021-2022

Project Overview

The Teacher-Student Portal is a web-based application designed to usable platform for teachers and students.

The Teacher-Student Portal is a web-based application for teachers and students, offering features like assignment submissions, grade tracking, forums, course management, to-do lists, and schedules. It helps students manage their academic tasks and stay organized while simplifying course management and resource distribution for teachers. This portal creates a cohesive and efficient academic environment with all essential tools in one user-friendly interface.

Objectives

In real life, sometimes it seems tough for us to maintain a platform where we have all the things in one platform such as routine, ranking list, resources as well as connection with peers. From the teachers' side, they need to manage the courses that they have taken, those resources, and launch assignments too. So, our application will be a central portal for students and teachers to maintain all their stuff and

Scope

Platform and Tools

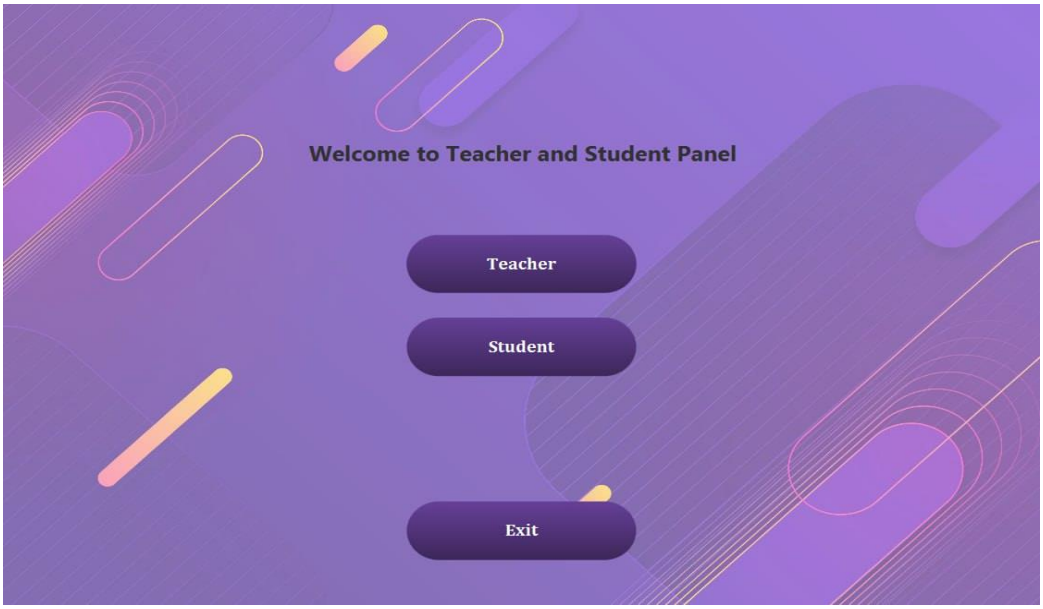
To implement object-oriented programming and design principles, we chose Java as our programming language and utilized IntelliJ as our integrated development environment (IDE). For data storage and management, we employed MySQL Workbench. We handled compiling and building the source code with Maven, which also managed our project dependencies.

The user interface elements were developed using JavaFX, with some designs created manually and others using SceneBuilder. Maven's pom.xml file was configured to include dependencies for JavaFX, logging, and other necessary components. This setup allowed us to automatically download and incorporate the correct library versions, streamlining the development process and ensuring compatibility across different environments.

Project Features

Our project is a teacher-student portal I have assigned two roles here

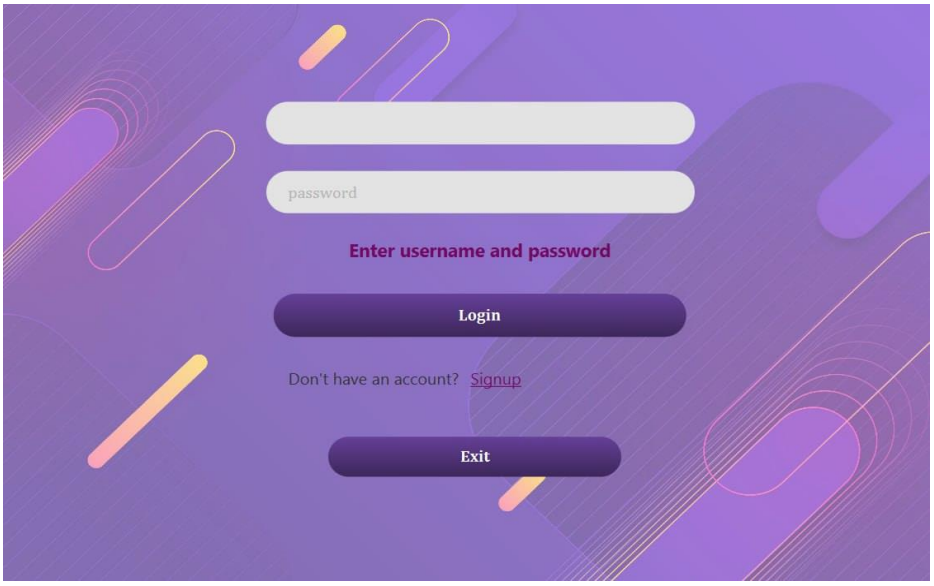
- **Teacher**
- **Student**



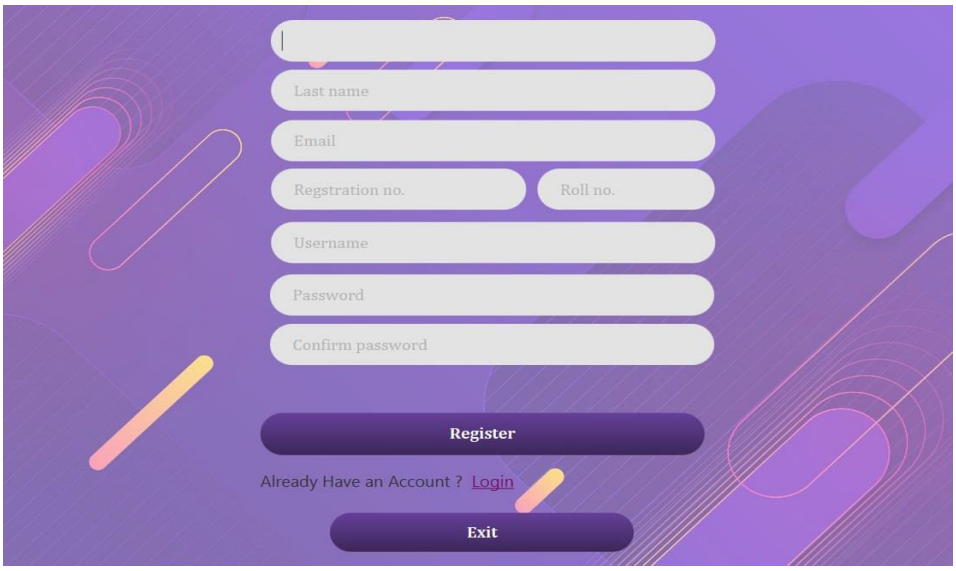
Student Portal

- **Registration, Authentication and Log-in**

  In the student portal, the first UI is the login page where we need to log in to our portal with the necessary information(username, password) and it is authenticated with our database.



For any new Student, they can register through the registration page with the necessary information,

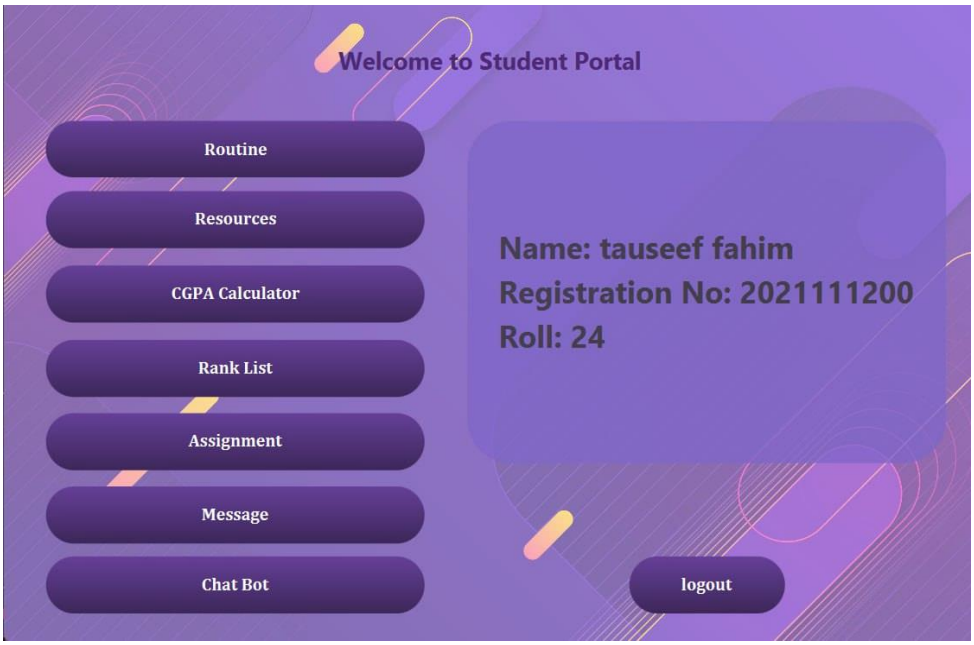


Then they will able to log in and all the information is stored in the current student.

So in this registration and login platform, we have mainly focused on the database connection, authentication, exception handling, and directing the user to our user platform.

- **Dashboard**

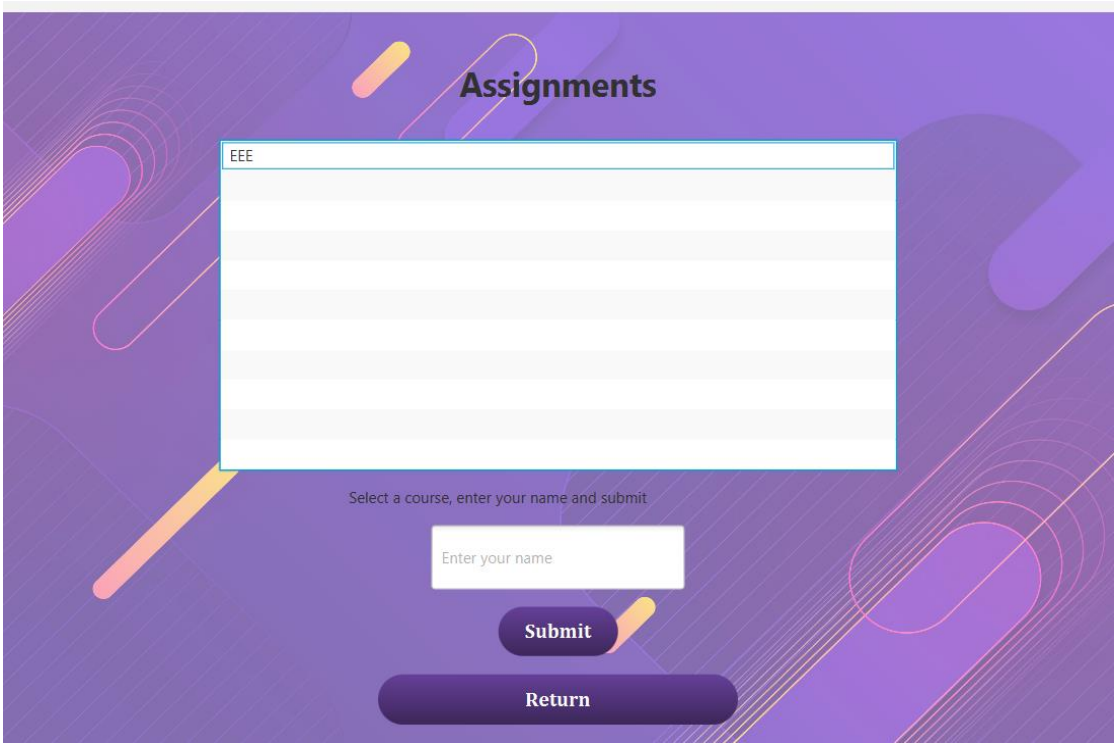
Here we have integrated the main plate form of each student and have all the necessary modules for students.



Here, after logging in all the information is shown on the right side of the dashboard. All other options are shown on the left side.

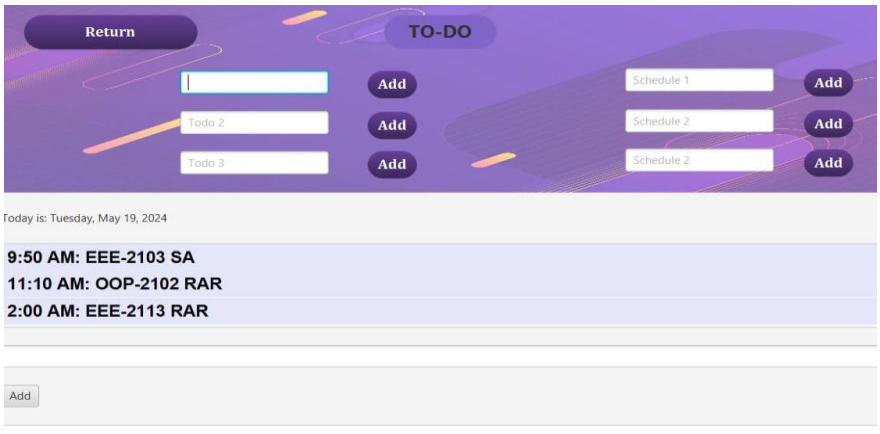
- **Assignment:**

Here students can view and submit teachers' assigned task.

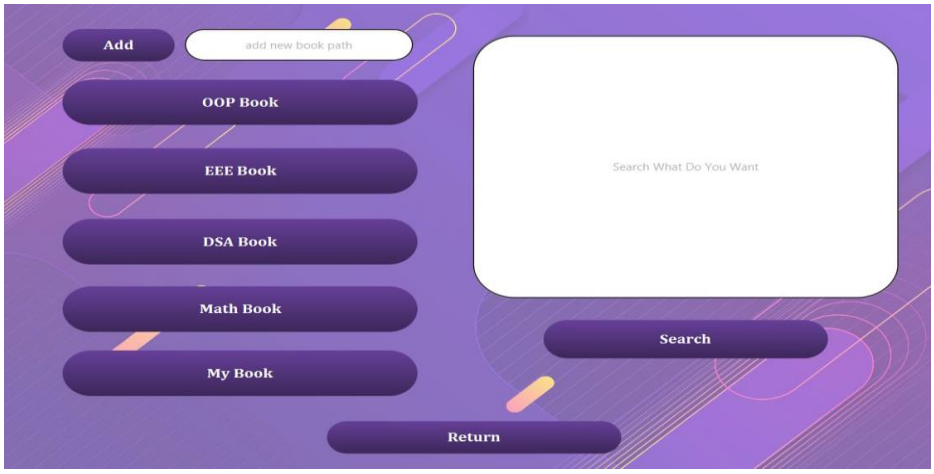


- **Routine**

  Here, we have shown our class routine, our dynamic to-do list, and a a short-time text editor. Our class routine was designed to make objects of each course's classes according to date and course description and these data are extracted from those classes. And todo lists are connected to our database so, we can dynamically add our new schedule here.



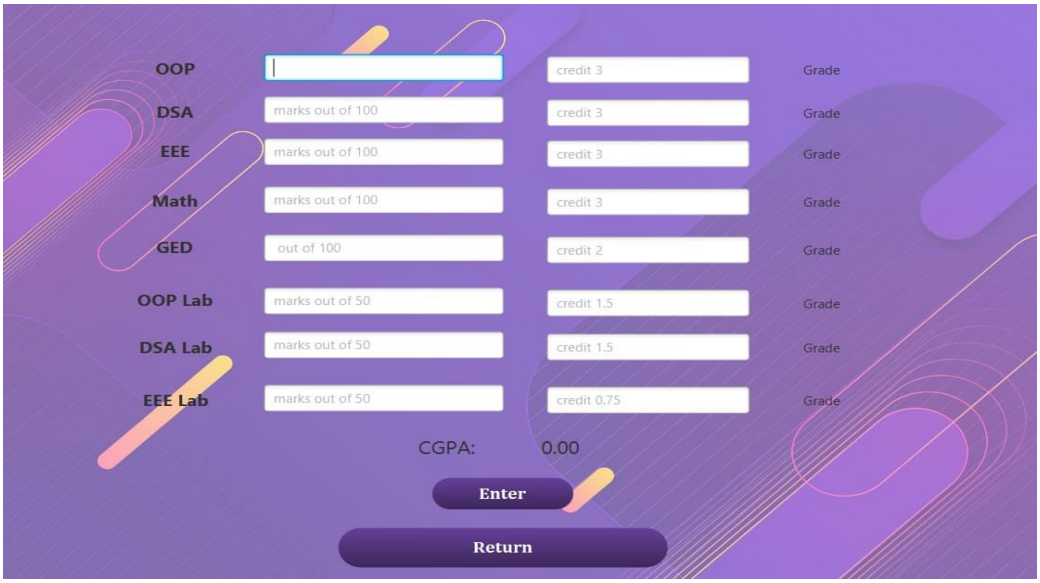
- **Resources**



Here we extract our resources file from our local memory by creating object for each course and also we can dynamically add new resources to our forum with the help of a database. Again, we have connected a search engine to take us to the web portal. Here we have run a thread to open and read multiple files and stay on the search engine.

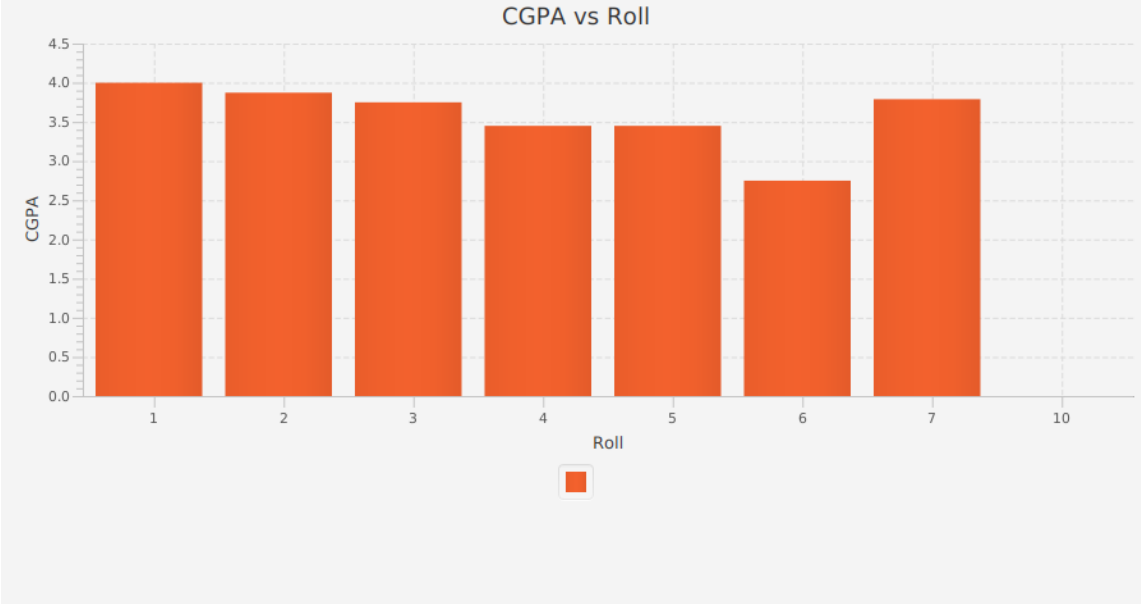
- **CGPAcalculator**

  Calculate CGPA by accepting the marks of each course. It shows the grade and final CGPA after enough calculation with considering credits.



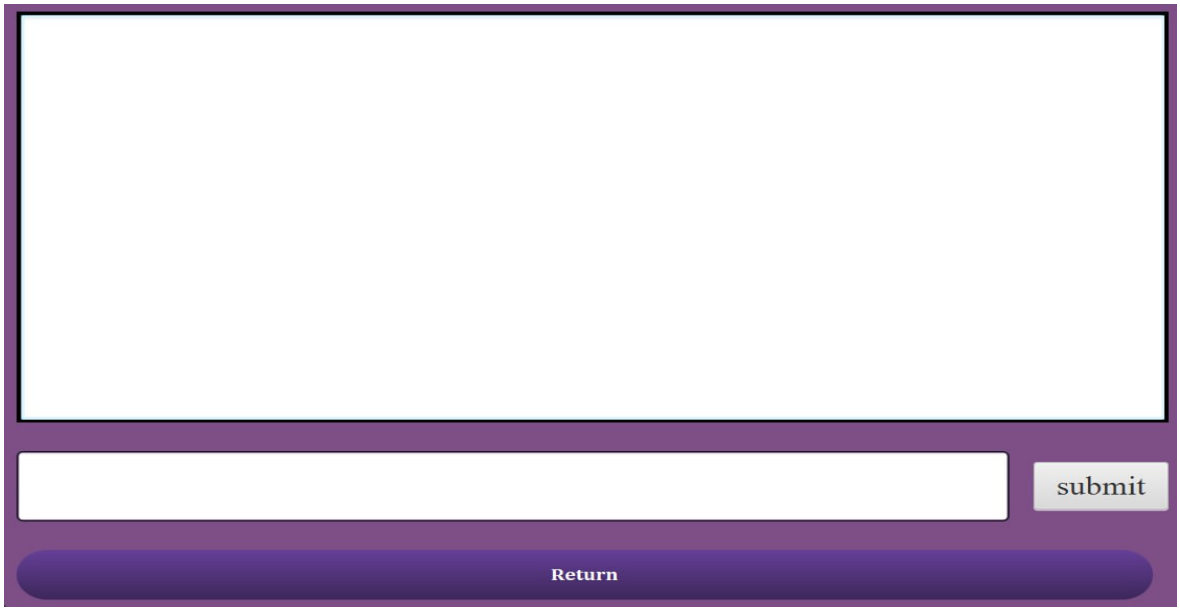
- ## Rank Graph

  Here we extract the CGPA from our database and draw a CGPA vs roll graph to see the rank and competitiveness of current students.



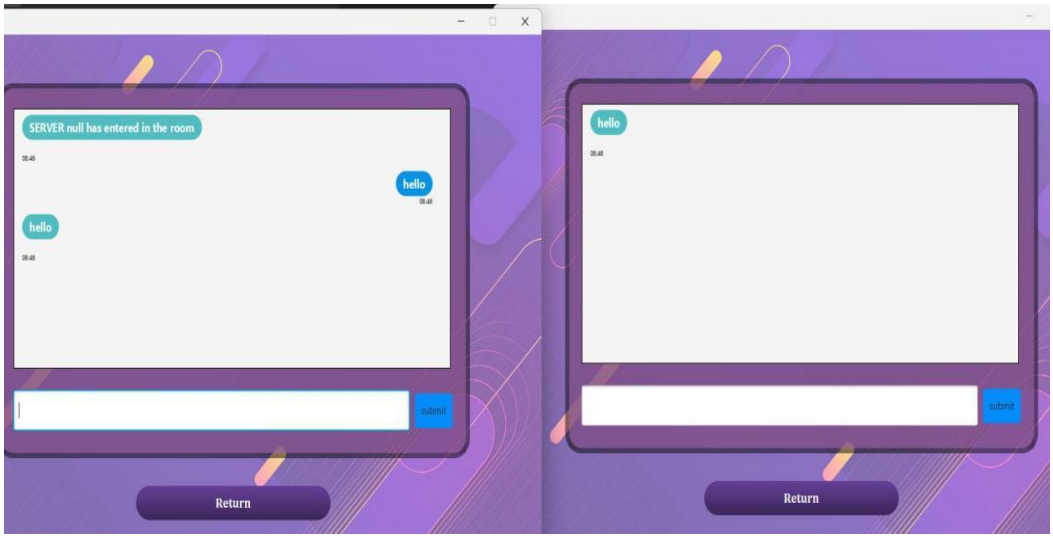
- ## ChatBot

  Here we have designed a chatbot to assist a student and carry conversation. If the question is out of the regular communication it will take us to the website where we may find those info.



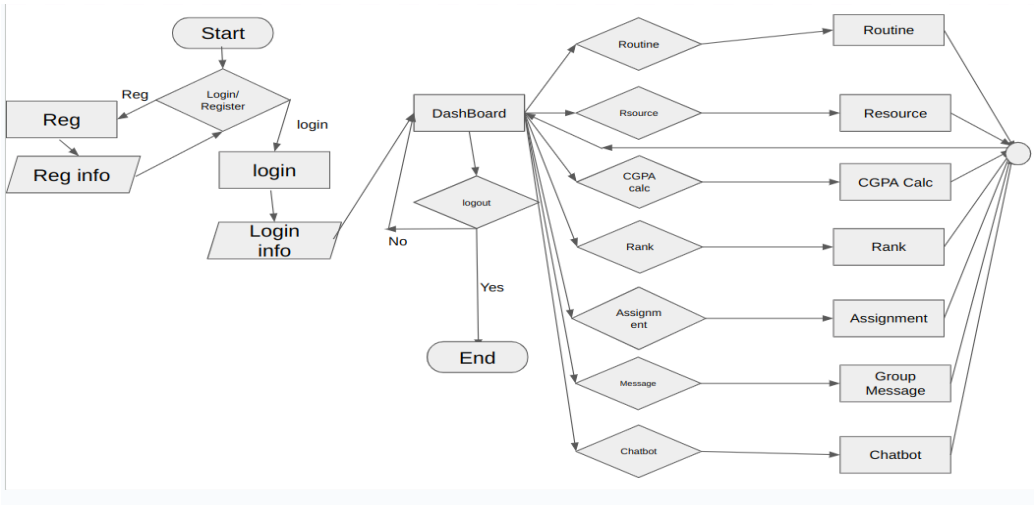
- **Message**

This is a group chat where the active students can communicate with each other and share their thoughts. Here we have used thread, and socket programming (server and client).



- **Work Flow of Students' Portal**

Here is the basic Workflow of our student portal, where all our pages are connected, they are moduled, and connected to a database.

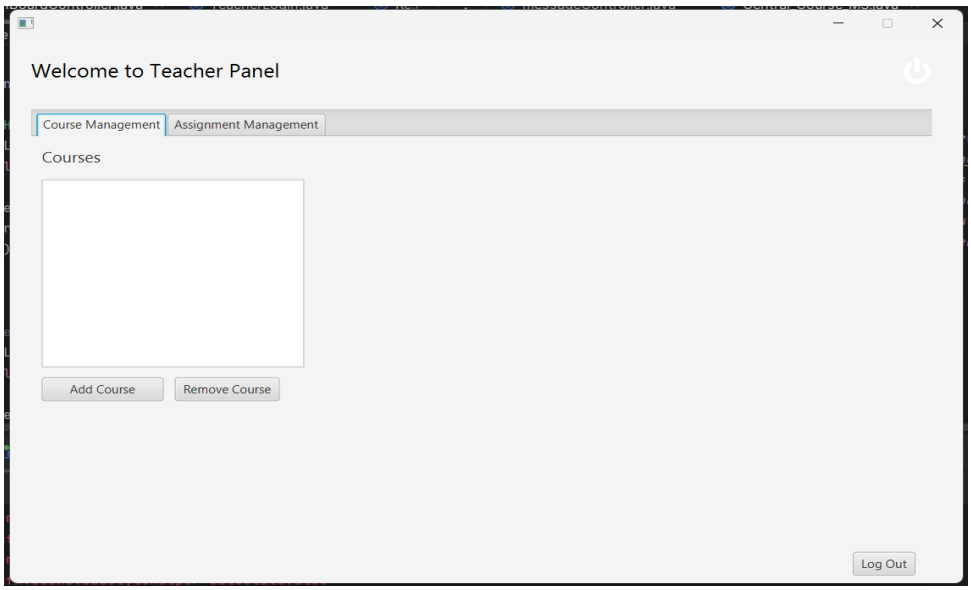


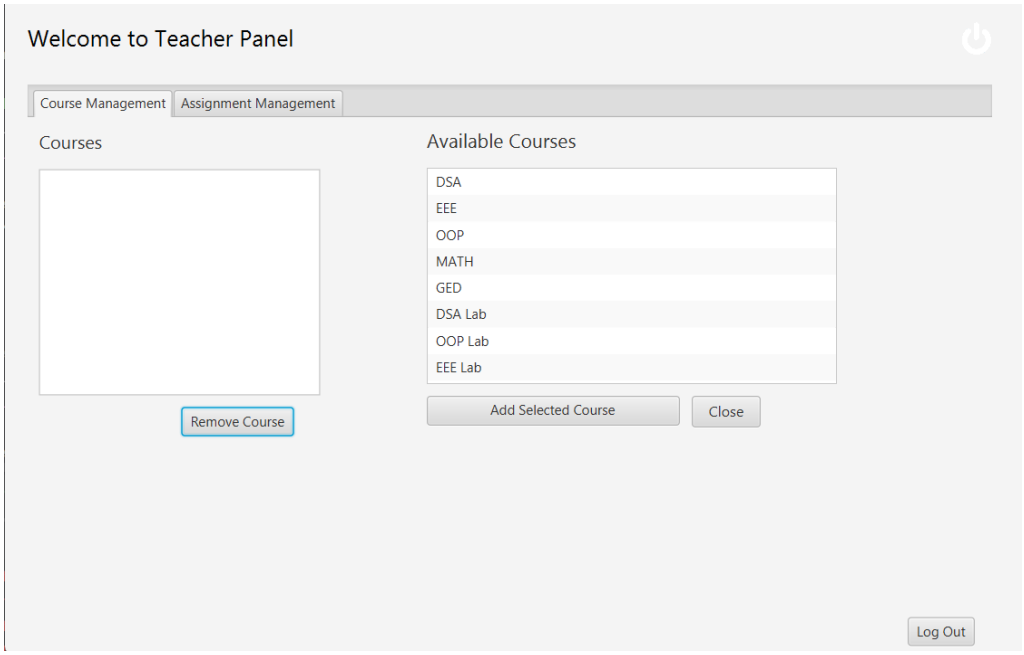
Teachers' Portal

Here teachers can register and log in like the students portal.

Dashboard

Here teacher can manage their courses, add courses to their profile, and delete courses, assign tasks for students.





Implementation of OOP and Design Principles

The main goal of the project was the implementation of object-oriented principles like polymorphism, abstraction, encapsulation, and inheritance. We have implemented all of those features in our project where necessary. Supposedly while calculating CGPA I need to apply polymorphism as different courses have different evaluation systems, we need to attract and encapsulate the current student's info, Also we need to apply inheritance where we need the features of superclass. On the other hand, we tried to follow a certain design so that our application is open for extension but closed for modification. Here as primary steps, we only add one semester but it is open to add more new semesters. We can easily add new students, resources, etc. More we tried to follow high-level modules that should not depend on low-level modules. All the files are designed to separate modules so that we can call them easily, and extend them easily. For example, in the student portal, we have connected all the modules with my student class and called them my dashboard which enables us to maintain our application. Moreover, sometimes we use subclass in our application which also can replace base class altering the correctness of the program.

Conclusion: Challenges, Discussion, and Future Plan

From the outset, our vision for the student and teacher portal was ambitious, aiming to create a comprehensive platform for integrating resources, managing courses, tracking results, and fostering connectivity among users. However, the journey encountered its share of challenges. Despite our initial aspirations, we've only managed to incorporate one semester into the platform

thus far. Nevertheless, our ambition remains undiminished, and we're determined to expand the portal to encompass all semesters of our undergraduate life in the future.

Throughout this project, we've strived to apply our understanding of object-oriented programming (OOP) and elevate our software design skills. While the progress may have been slower than anticipated, we're proud of the efforts invested and the knowledge gained along the way. This experience has been invaluable, allowing us to confront real-life problems head-on and develop solutions independently.

Looking ahead, our focus remains on continuous improvement and expansion. We're committed to leveraging our newfound expertise to enhance the portal, incorporating additional features and functionalities to better serve the needs of both students and teachers. By building upon the foundation laid thus far, we're confident in our ability to realize the full potential of the student and teacher portal, creating a robust and indispensable tool for academic success.

GitHub Link: https://github.com/TAUSEEF-01/OOP_Project_2024