University ERP System

Project Report

Course: Advanced Programming(CSE201)

Team members: Naman Galav(2024366)
Shauryah Annavel(2024526)

Date of Submission: 27/11/2025

1. Introduction

In this project, my teammate and I have recreated the desktop version of IIITD's ERP system using Java SE and Swing. In the system, we have implemented three different types of access based on the user's roles. They are Administrator, Instructor, and Student. They allow for efficient management of course registrations, grading, and scheduling. In our project, We have implemented Auth DB as an authentication database and ERP DB as an academic database. By separating authentication from ERP data, enforcing a strict role-based access control, and the addition of a "Maintenance Mode" has allowed us to ensure security and integrity during system updates.

2. Database Design & Data Separation

In our project, we have used two distinct MySQL databases to ensure security and logical separation of concerns.

1) Authentication Database (auth_db)

This database is used strictly for security purposes, like security logs and user credentials. It does not store any data related to academic data here.

- **users_auth**: Stores user_id, username, role(Admin, Instructor, Student), and the hashed password. Here, we utilize BCrypt for hashing, ensuring that plaintext passwords are never stored or accessible. It also tracks failed_attempts and lockout_end time for security.
- **password_history**: Stores the last 3 password hashes per user to prevent password reuse.

2) ERP Database (erp_db)

This database is used to store the university's operational data.

- **students**: Links to auth_db via user_id; stores profile data (Roll No, Program, Year).
- **instructors**: Links to auth_db via user_id; stores Department and Name.
- **courses**: Stores course metadata (Code, Title, Credits).
- **sections**: Specific instances of courses (Day/Time, Room, Capacity, Semester).
- **enrollments**: Links Students to Sections (Handles registration status).
- **grades**: Stores scores for specific components (Quiz, Midterm, Final) and the final letter grade after computation
- **course_prerequisites**: Defines which courses must be passed before registering for others.
- **settings**: Stores system-wide flags, specifically the maintenance_on flag.

3. Roles & Access Enforcement

The system enforces strict access rules. A SessionManager singleton tracks the currently logged-in user, and the UI adapts accordingly.

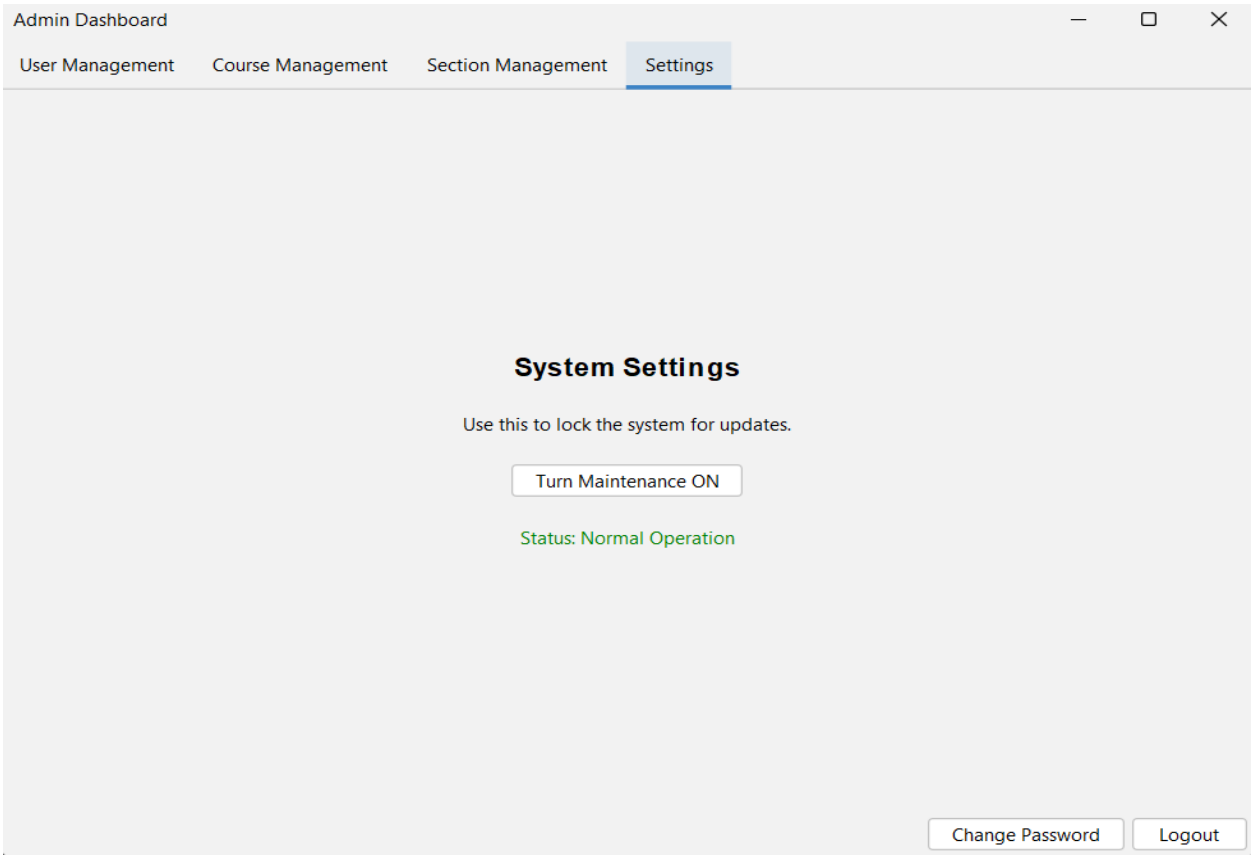
Role Enforcement

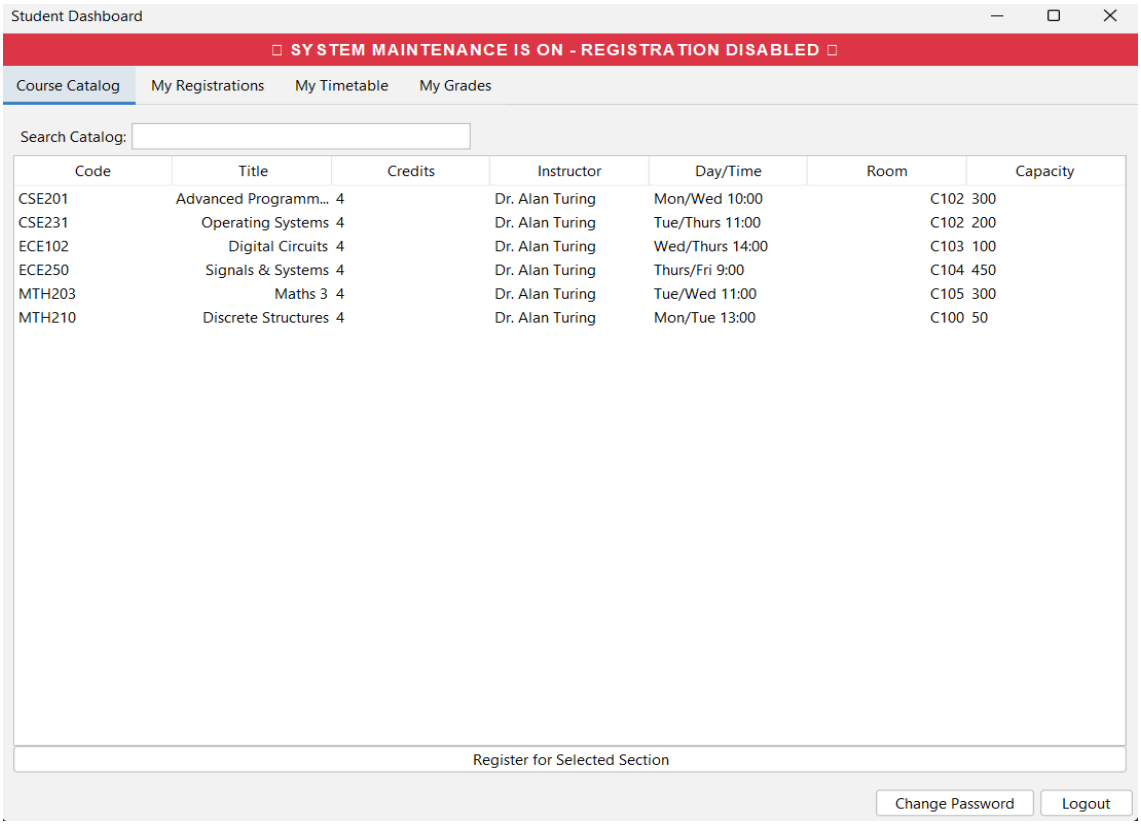
1. **Admin:** manages users, courses, sections, and global settings. Basically, has full control.
2. **Instructor:** They have restricted access. They manage assessments and grades for their own sections.
3. **Student:** They have restricted access. Student has access to viewing catalog, registers/drops sections, and views grades.

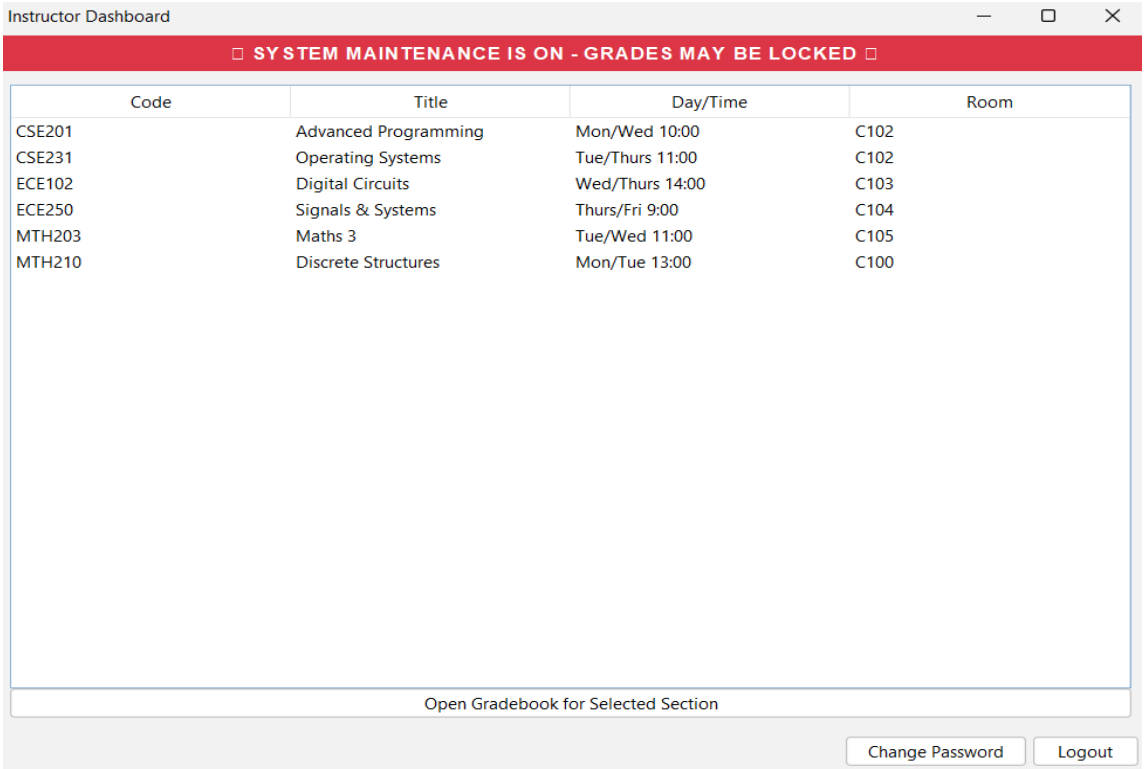
Maintenance Mode Enforcement

The system includes a global "Maintenance Mode" controlled by the Admin.

- **Mechanism:** A flag in the settings table (maintenance_on) is checked by the AccessControl class before any critical write operation.
- **Behavior:** When maintenance mode is ON, a red warning banner appears on all dashboards. Students and instructors are restricted to read-only operations, but Admin retains full privileges.





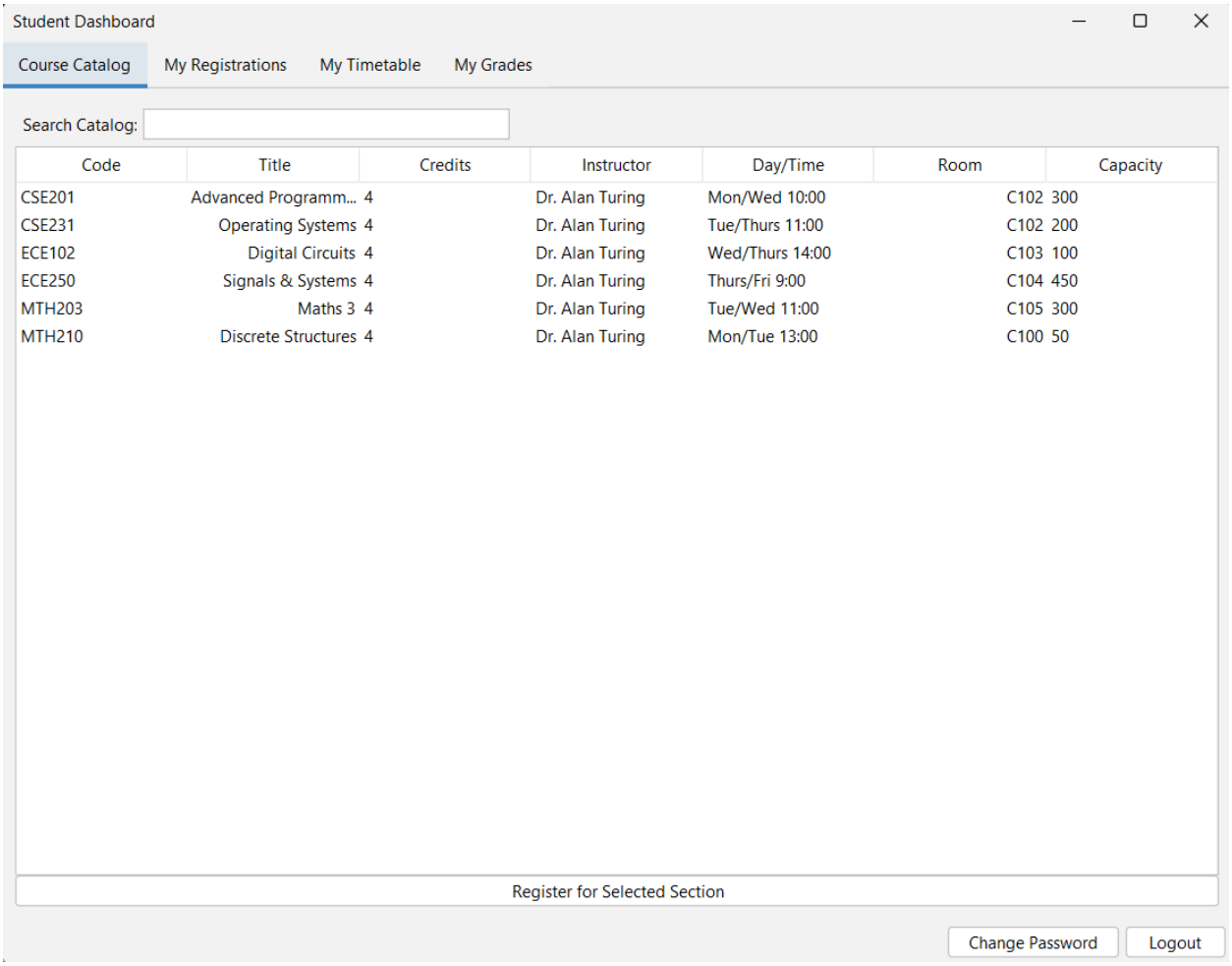


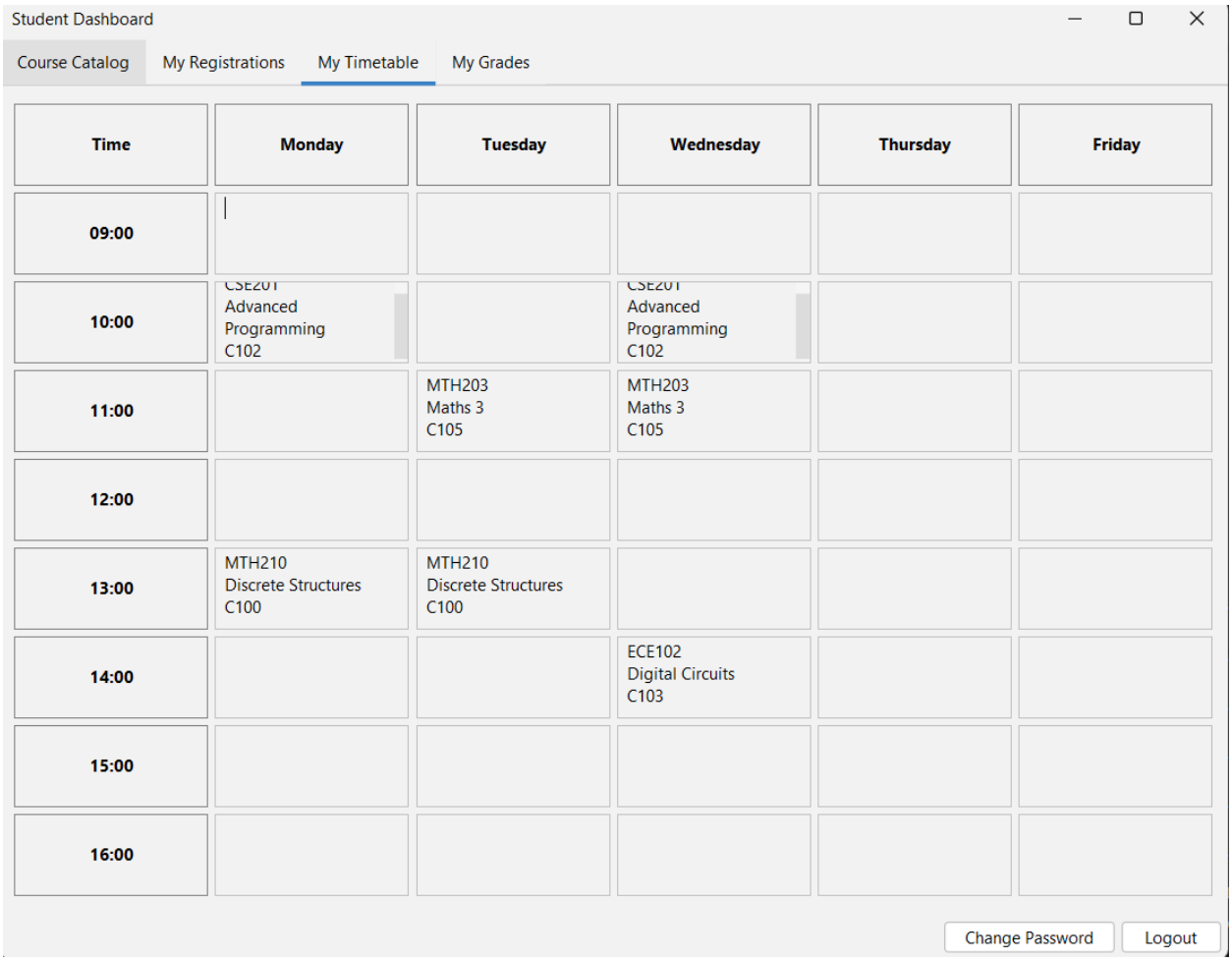
4. Features & Functionality

A. Student Module

Students have access to a tabbed dashboard providing a complete academic overview.

- **Course Catalog:** A searchable and sortable table of all available course sections for the current term. Includes logic to prevent registering for full sections or sections where prerequisites are not met.
- **My Registrations:** A list of currently enrolled courses with a "Drop" function.
- **Timetable:** A visual grid displaying the student's schedule by Day and Time.
- **Transcript:** A view of all grades and a button to download an official CSV and PDF transcript.

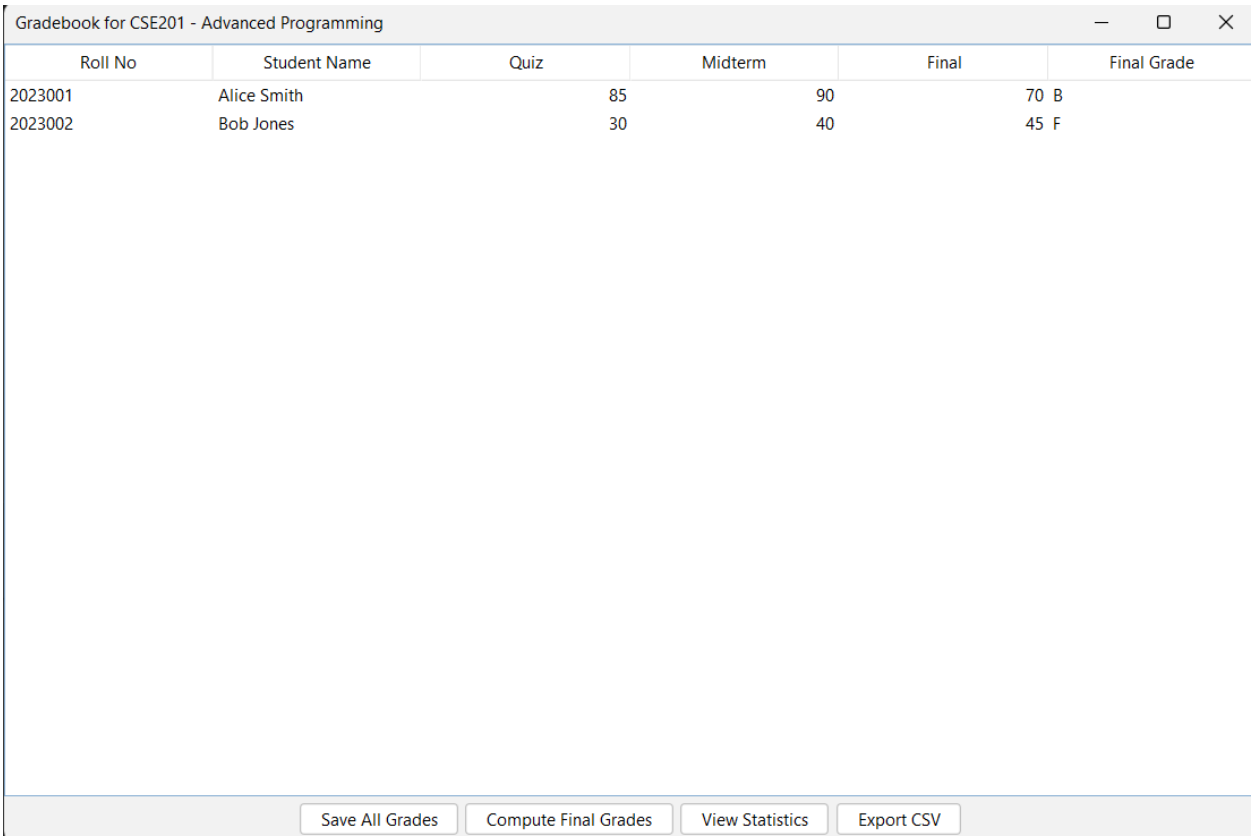




B. Instructor Module

Instructors use the system to manage their assigned classes.

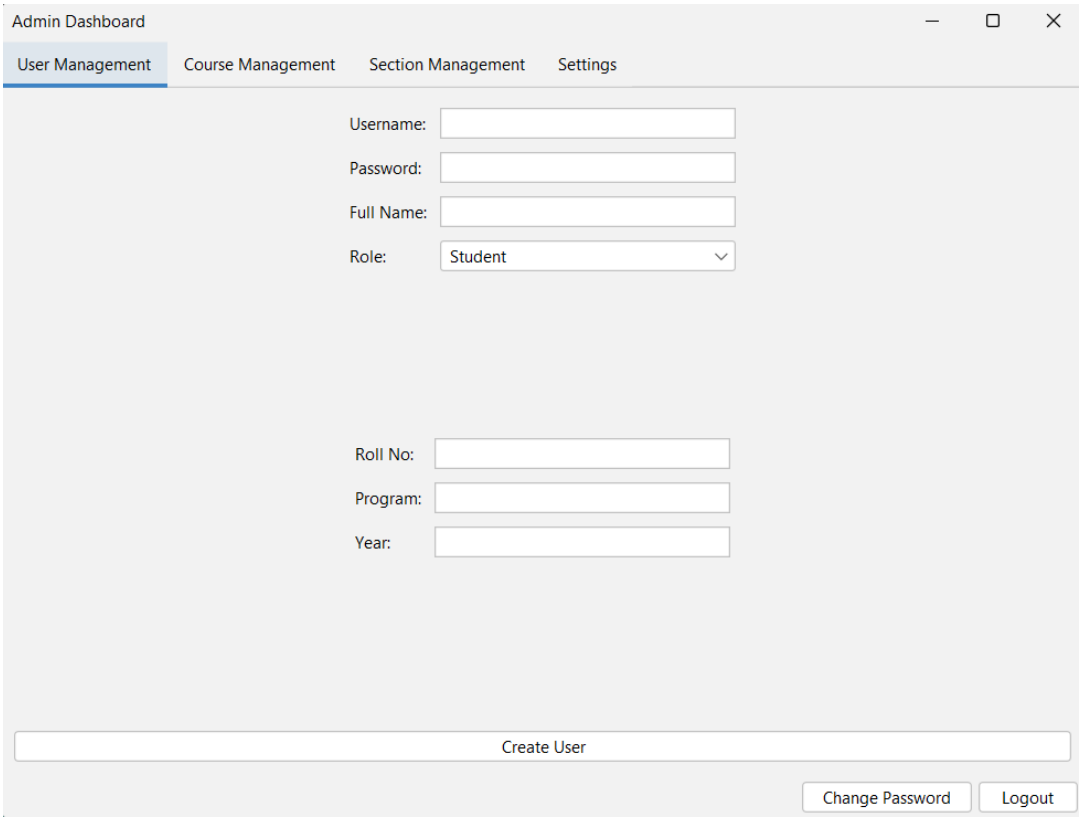
- **My Sections:** Displays only the sections assigned to the logged-in instructor.
- **Gradebook:** Displays enrolled students in the selected section and provides a detailed table where the instructor can enter marks for Quizzes, Midterms, and Finals.
- **Statistics:** A popup dialog showing the Average, Highest, and Lowest scores for the section.
- **Exporting grades:** Can export the gradebook to CSV for offline storage.

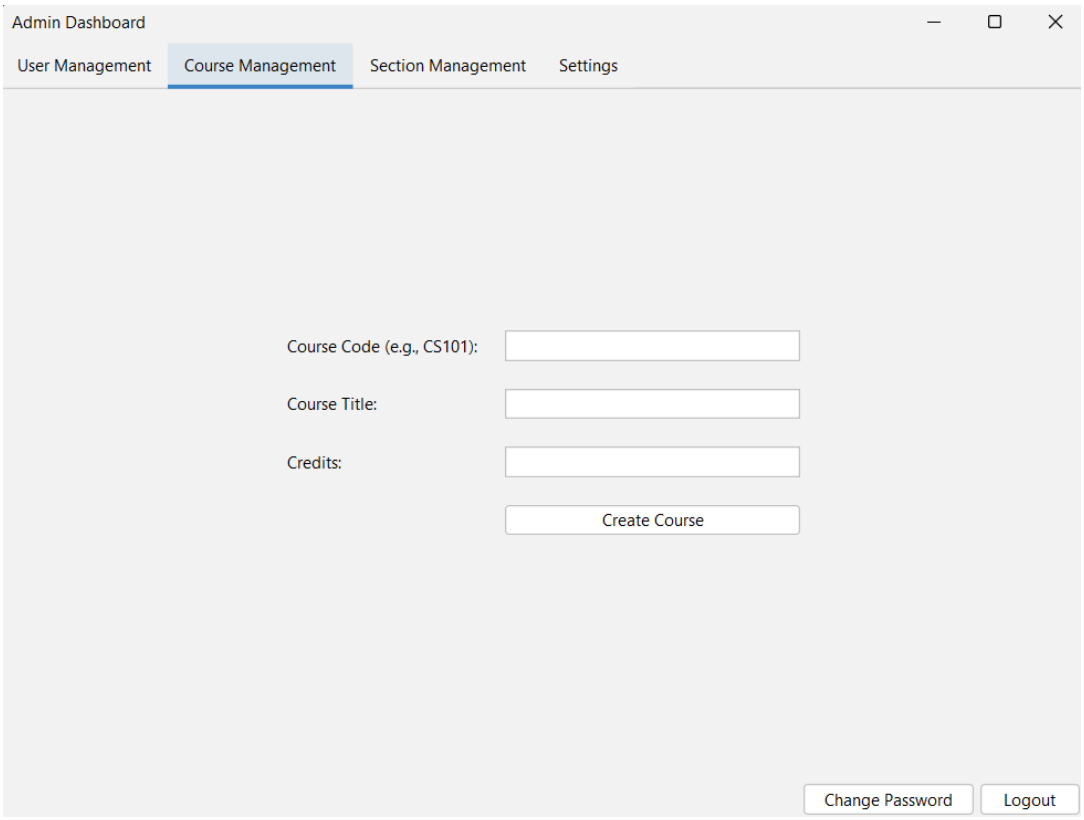


C. Admin Module

Admins serve as the backend managers, providing the following panels

- **User Management:** used to create new user accounts and choose roles(Admin, Student, Instructor) (automatically generating entries in both auth_db and erp_db).
- **Course Management:** Tools to add, delete, and edit courses.
- **Section Management:** used to create course sections with term, instructor, capacity, and schedule. Assign an instructor to each section. Can be used to update time and capacity as needed.
- **Settings:** Can toggle maintenance mode on and off.


Admin Dashboard
User Management
Course Management
Section Management
Settings
Username:
Password:
Full Name:
Role:
Student
Roll No:
Program:
Year:
Create User
Change Password
Logout


Admin Dashboard
User Management
Course Management
Section Management
Settings
Course Code (e.g., CS101):
Course Title:
Credits:
Create Course
Change Password
Logout

5. Final Grade Weighting Rule

The system automates the calculation of final letter grades to ensure consistency. The logic is encapsulated in the InstructorService class.

The Weighting Formula:
The system uses a 20/30/50 split:
$$Final Score = (Quiz \times 0.20) + (Midterm \times 0.30) + (Final \times 0.50)$$

The Grading Scale:
Once the numeric score is calculated, it is converted to a Letter Grade:

- **A:** 90 - 100
- **B:** 80 - 89
- **C:** 70 - 79
- **D:** 60 - 69
- **E:** 50 - 59
- **F:** Below 50

Instructors simply click **"Compute Final Grades"**, and the system performs this calculation for every student in the roster and updates the database.

6. Extras & Bonus Features

In addition to the core ERP requirements, several advanced functionalities have been incorporated to improve usability, security, and overall user experience.

1. **PDF Export (Student):**
    - We have used OpenPDF library to enable Students to generate a professional looking PDF transcript summurizing their personal details and completed course grades. It is located in the Student dashboard for easy access.
2. **CSV Export (Instructor):**
    - Instructors are provided with a CSV export option that outputs their gradebook data to a .csv file, allowing for offline backup or analysis in Excel.
3. **Security: Login Lockout:**
    - To protect against brute-force login attacks, the system tracks failed login attempts for every user. If a user fails to log in **5 times**, their account is locked for **15 minutes**.
4. **Security: Password History:**
    - Users are prevented from reusing any of their **last 3 passwords**. This is done by the system, which checks the password_history table when a user tries to change their password.
5. **Search & Sort:**
    - The Course Catalog offers real-time search and column sorting using TableRowSorter, enabling students to easily sort like by Credits and filter courses

dynamically.

7. Conclusion

The University ERP project we have built successfully meets all the functional and non-functional requirements outlined to us in the project instructions. It presents a clean separation of concerns through the use of its DAO and service architecture, robust security through database isolation and encryption and user friendly Swing interface enhanced with FlatLaf. The addition of maintenance control and export features makes it a viable platform for academic management.