University ERP System

Short Report

1. Introduction

This report presents the design, implementation, and functioning of the University ERP System developed using Java Swing and MySQL. The application supports three roles—Admin, Instructor, and Student—each with specific permissions and workflows. The system uses two databases: Auth DB (for login + roles) and ERP DB (for academic data).

2. System Architecture Overview

The project follows a clean layered structure:

• UI Layer (Swing Panels in edu.univ.erp.ui)

• API Layer (edu.univ.erp.api.*) separating UI and services

• Service Layer (edu.univ.erp.service)

• Data Access Layer (edu.univ.erp.data)

• Auth Layer (edu.univ.erp.auth) for login + password hashing

• Domain Models (edu.univ.erp.domain)

• Utility Layer (edu.univ.erp.util) for CSV exports

. Database Design (Both DBs)

The system uses two separate databases to enforce security:

AUTH_DB (users_auth):

• user_id (Primary Key)

• username (unique)

• role (Student/Instructor/Admin)

• password_hash

• status

• last_login

ERP_DB Tables:

• students(user_id, roll_no, program, year)

• instructors(user_id, department)

• courses(course_id, code, title, credits)

• sections(section_id, course_id, instructor_id, day_time, room, capacity, semester, year)

• enrollments(enrollment_id, student_id, section_id, status)

• grades(grade_id, enrollment_id, component, score, final_grade)

• settings(setting_key, setting_value) (for Maintenance Mode)

4. Role-Based Access Enforcement

Every action checks user role and maintenance mode:

• Admin: Full CRUD on users, courses, sections;

• Instructor: Only their assigned sections; cannot edit others.

• Student: Only their own data.

• Maintenance Mode ON: Students & instructors become read-only.

These rules are enforced inside the Service Layer using guard checks.

5. Feature Summary

STUDENT:

• Browse course catalog

• Register/drop sections (with seat + deadline checks)

• View timetable

• Check grades

• Download transcript (CSV)

INSTRUCTOR:

• View assigned sections

• Enter assessment scores

• Compute final grade (20% Quiz, 30% Midterm, 50% End-sem)

• Export grade CSV

ADMIN:

• Create users (Auth DB + ERP DB profiles)

• Create/edit/delete courses

• Create/edit/delete sections

• Assign instructor

• Toggle Maintenance Mode

6. Final Grade Weighting Rule

The application supports customizable final-grade computation, configurable by the instructor for each section. When the instructor selects "Compute Final", a weight-setter dialog appears with three editable fields:

- Quiz %
- Midterm %
- End-sem %

(Default values pre-filled: 20%, 30%, 50%, but the instructor may change them to any valid percentages that sum to 100.)

Once weights are set, the application computes:

$$\text{Final Score} = \left(\frac{QuizWeight}{100}\right) \cdot QuizScore + \left(\frac{MidWeight}{100}\right) \cdot MidtermScore + \left(\frac{EndWeight}{100}\right) \cdot EndSemScore$$

7. Bonus Features Implemented

• CSV export for grades (Instructor)

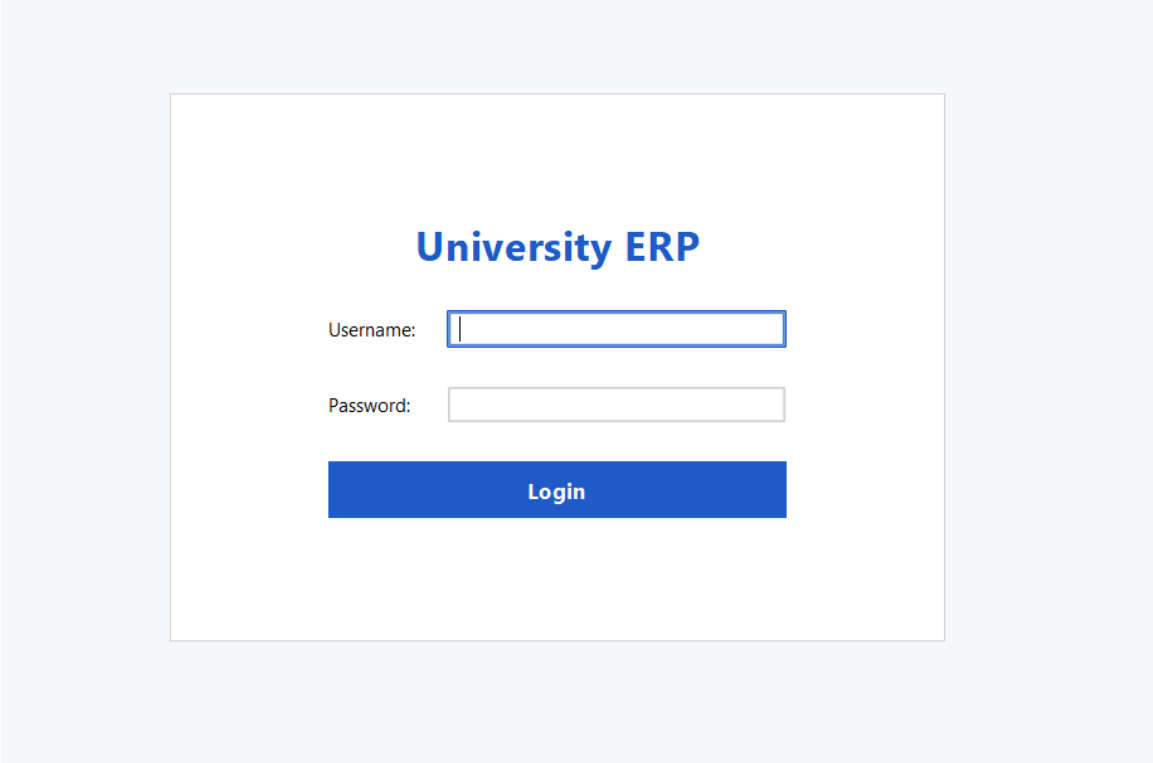
• CSV export for transcript (Student)

• Change Password dialog

• Login lockout after 5 incorrect attempts

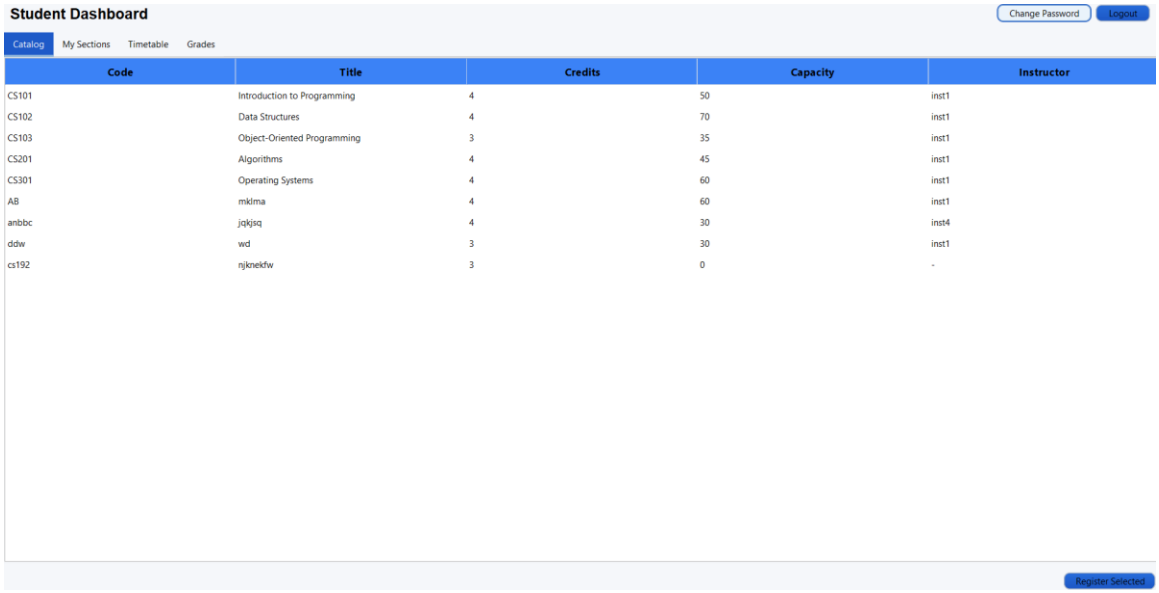
8. Screenshots

• Login Screen

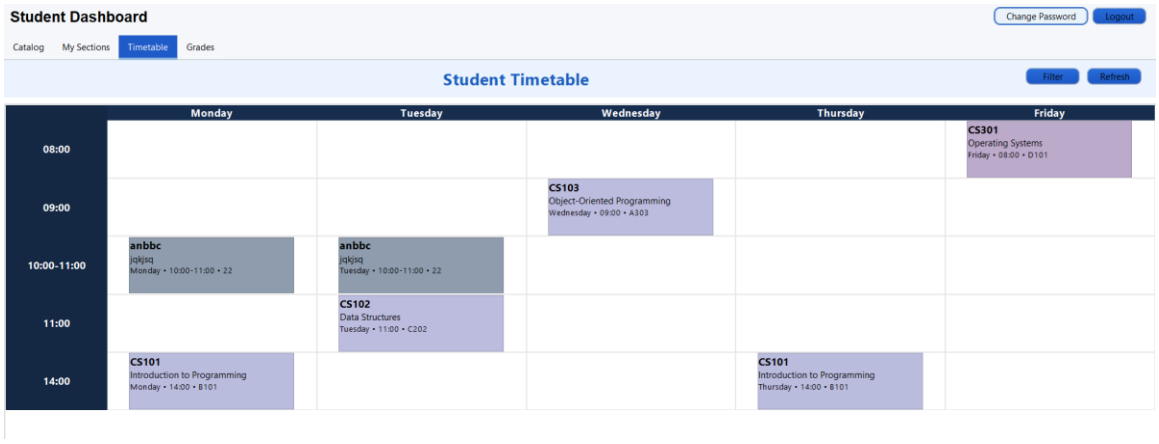


Student

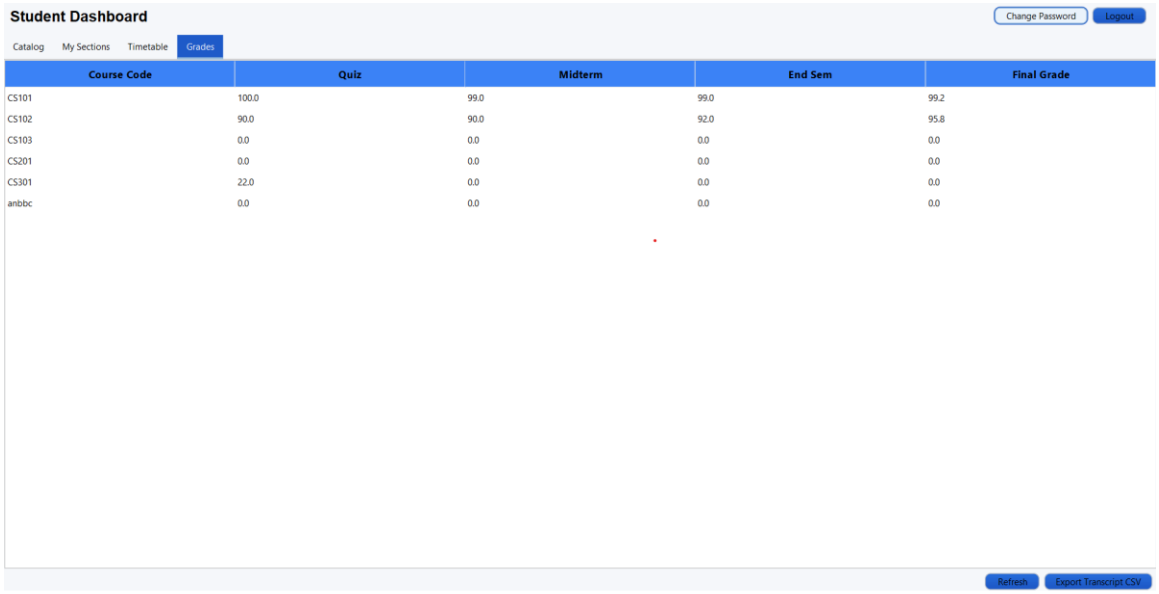
• Catalog Page



• Timetable Page

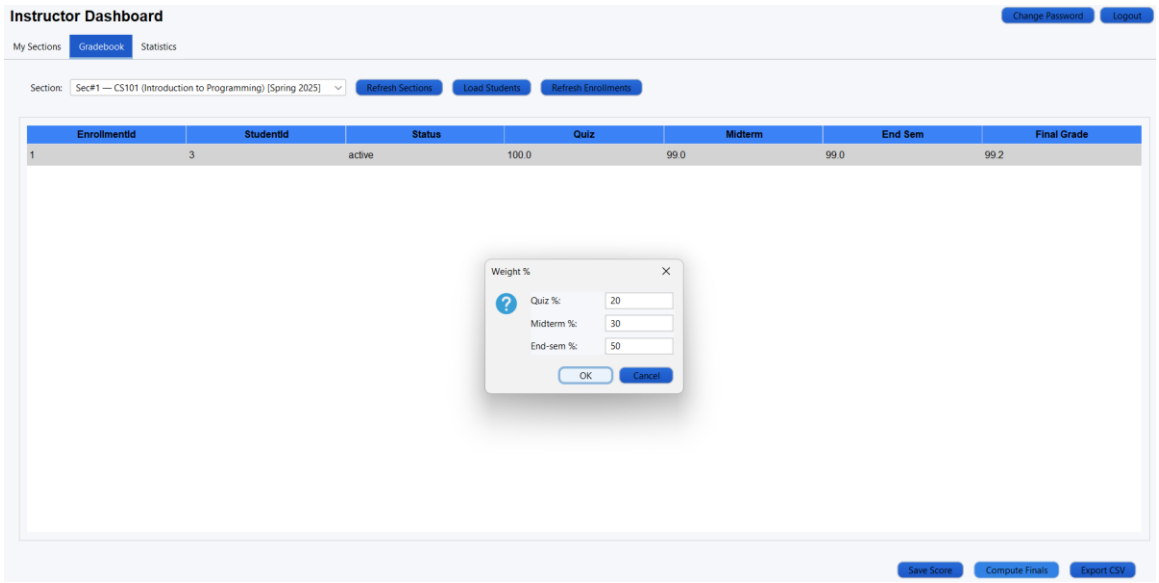


• Grades Page

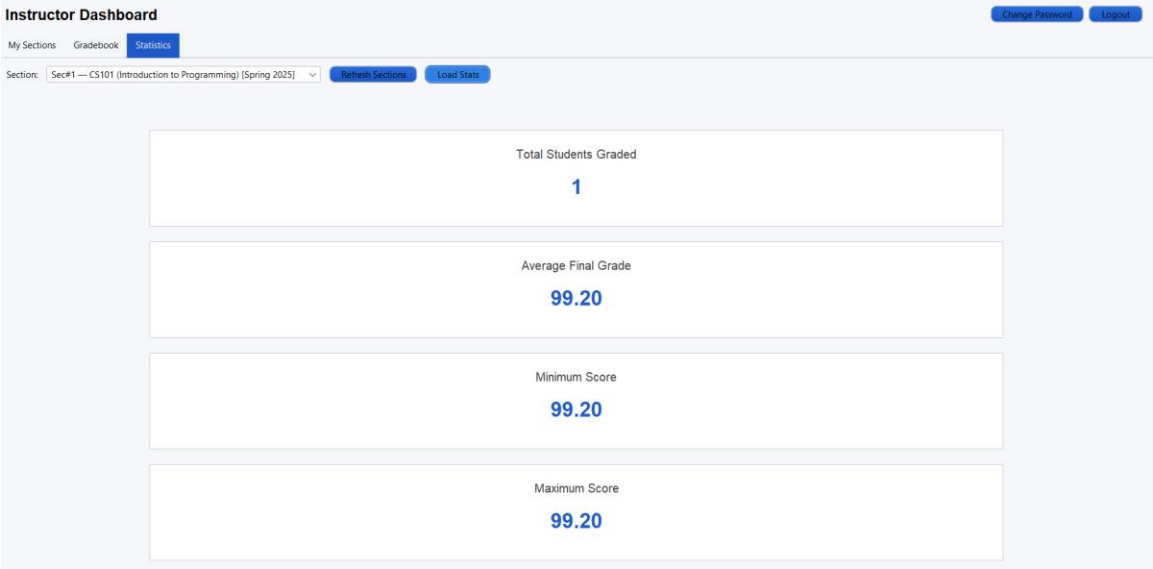


Instructor

• Instructor Grade Entry Screen

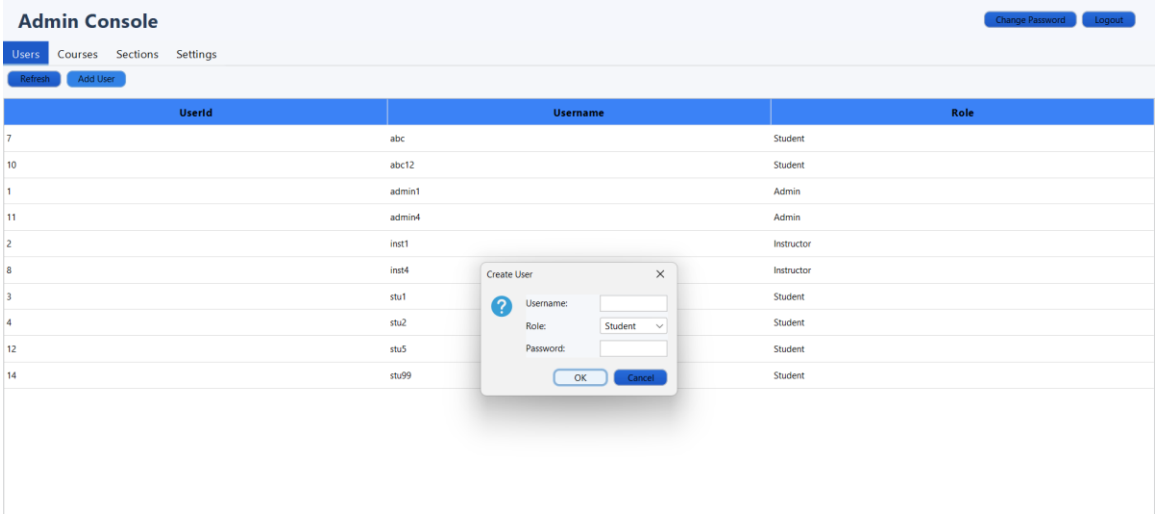


• Statistics

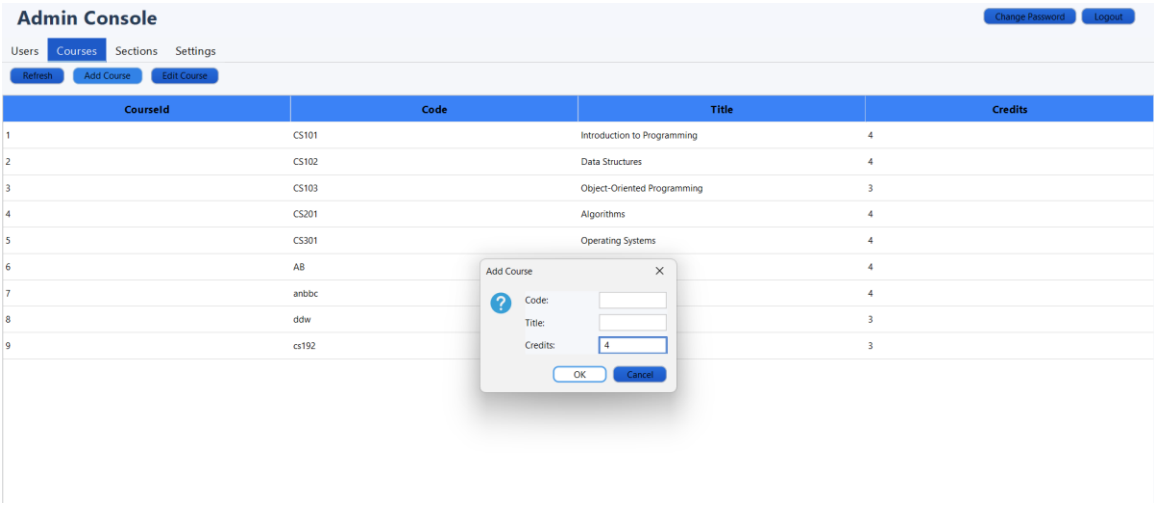


Admin

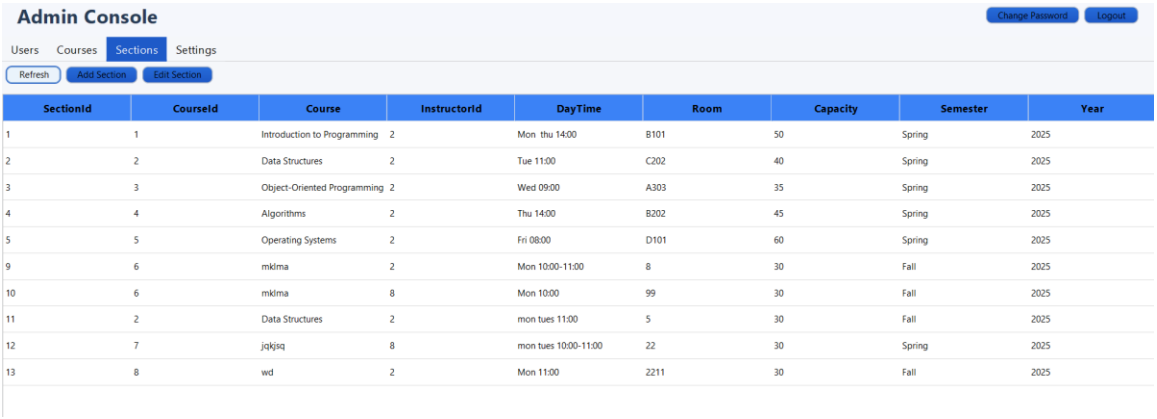
• Admin User Management



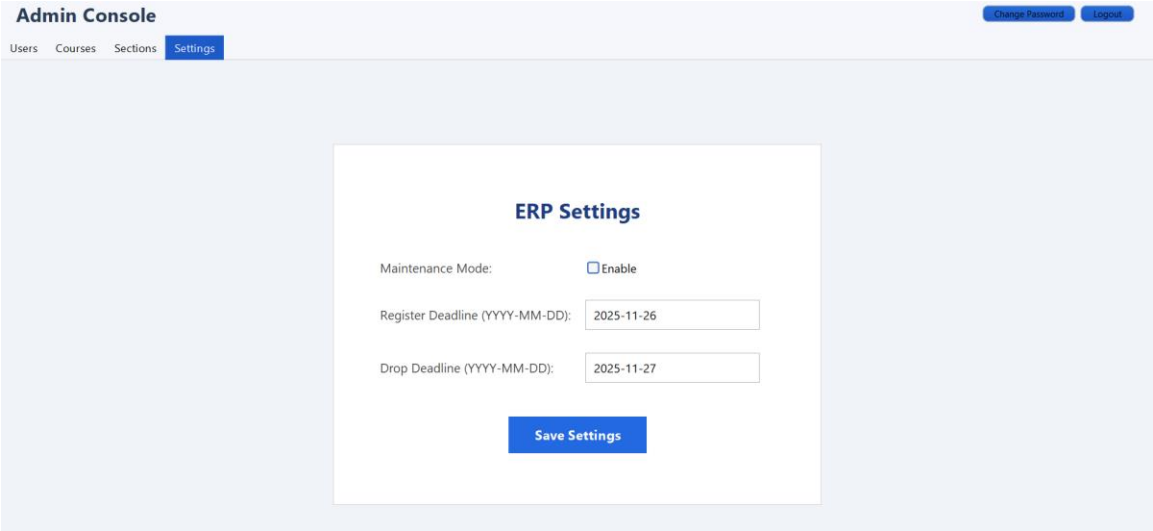
• Admin Course Panel



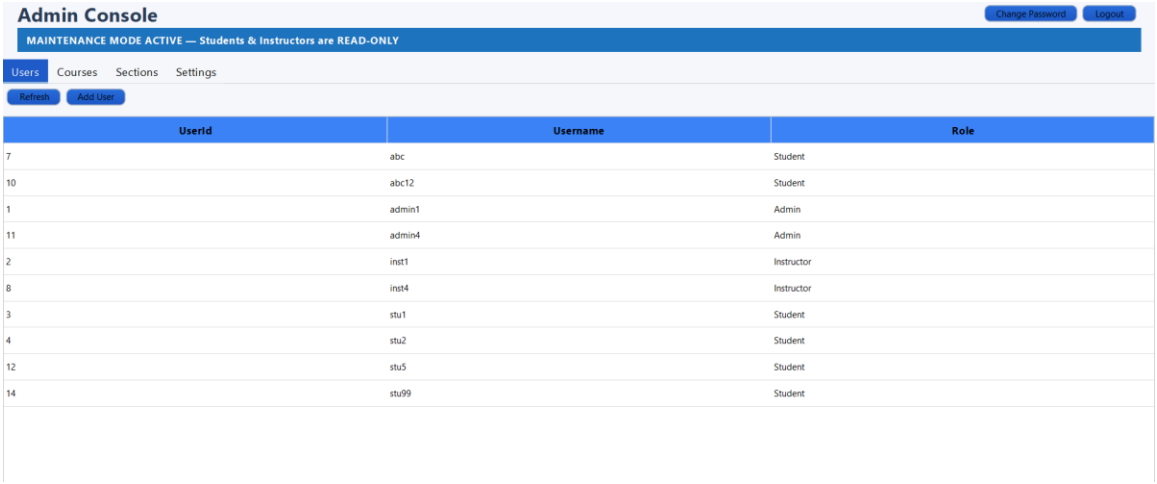
• Admin Section Panel



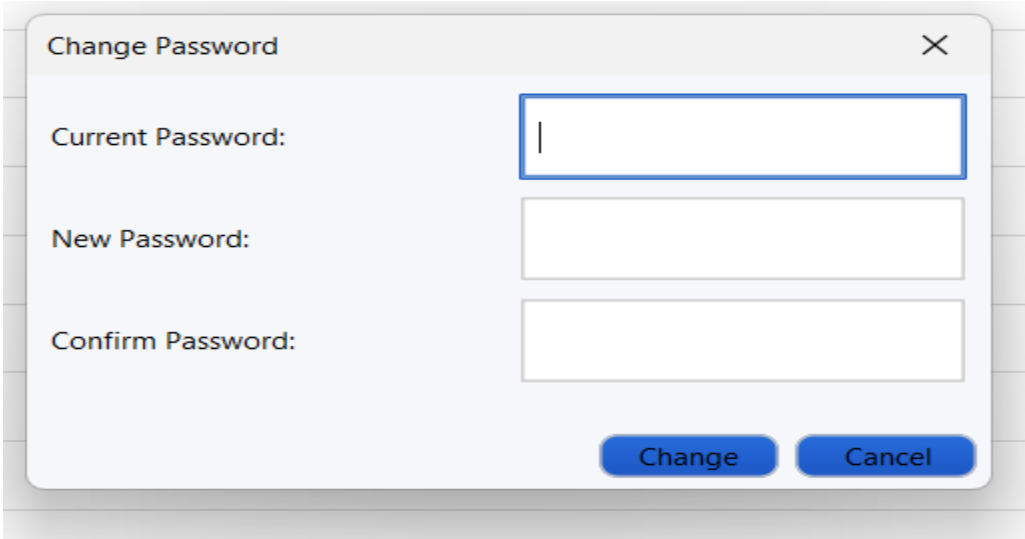
• Admin Section Panel



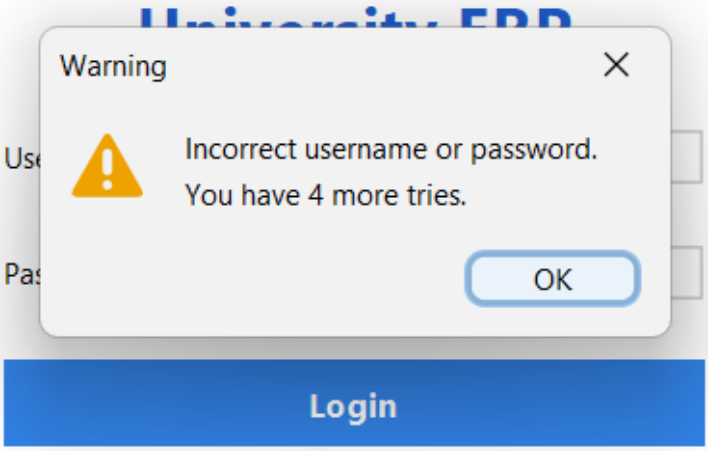
• Maintenance Mode Banner



• Change Password dialog



• Login lockout after 5 incorrect attempts



9. Conclusion

The University ERP System provides a secure, role-based academic management workflow with clear separation of authentication and academic data. The system meets the functional requirements and supports extensibility for future features.