Short Report: University ERP System

Shreyas Modi (2024554) and Jai Ahuja (2024275)

1 Introduction

This project is a University ERP System built using Java Swing and MySQL. The aim was to create a small but functional ERP where students, instructors, and admins each have separate roles and permissions. We also used two databases: one only for authentication (storing usernames, roles, and password hashes) and another for all ERP records like courses, enrollments, grades, etc. The separation helped keep login related data secure and independent from academic data.

2 System Overview

The application is divided into layers:

- **UI Layer:** All student, instructor, and admin screens (JFrames and panels).
- **Service Layer:** Handles the main logic such as registration checks, grade calculation, and maintenance mode rules.
- **Data Layer:** Interacts with MySQL using JDBC and all queries for ERP and Auth DBs are placed here.
- **Utility Layer:** Helper classes for exporting transcripts, password hashing (bcrypt), dialogs, and validation.

3 Database Design

We used two databases: `auth` (for login) and `erp` (for everything else). Below is a short description of the tables we used.

Auth Database Tables

- **username_password** – stores usernames, bcrypt hashed passwords, user roles and lockout info.

ERP Database Tables

- **students** – student profiles (name, roll number, program, year).
- **instructor_name_username** – maps instructor usernames to full names.
- **admins** – list of admin accounts.
- **courses** – course details, assessment weights, seats, and section identifiers.
- **sections** – specific offerings of a course (day/time, room, instructor, capacity).
- **enrollments** – links students to their registered sections. Duplicate enrollments are prevented.
- **grades** – stores marks for quiz, midsem, endsem, assignments, etc.
- **maintainence** – stores a simple ON/OFF flag for maintenance mode.
- **notifications** – used for marking system notifications for each user.
- **registration_dates** – defines the registration window.
- **relation1** – maps usernames to designation to support role checks.
- **subjectsxname_instructor** – lists which courses each instructor teaches.
- **subjectsxname_students** – lists which courses each student is taking.

Key relationships

- Every username in ERP also exists in the Auth DB.
- A course can have multiple sections.
- A student can enroll in many sections, but only once per section.
- Grades link a student to a specific subject and section.

4 Features

Below is a summary of the main features we implemented.

Common

- Secure login (with bcrypt).
- Role-based dashboards.
- Meaningful error/success messages.
- Tables for course and section listings.

Student

1. Browse Course Catalog

Students can view all available courses along with sections, instructor, credits, and seat availability.

2. Register for a Section

The system checks:

- Seat availability
- Whether the student already enrolled before
- If registration dates are still valid
- Maintenance mode status

3. Drop a Section

Students may drop registered courses within the allowed time window.

4. View Timetable

A timetable is generated from the student's enrolled sections.

5. View Grades

Students can see all marks and final grade (if computed).

6. Download Transcript

Transcript export supports CSV and PDF formats.

Screenshots

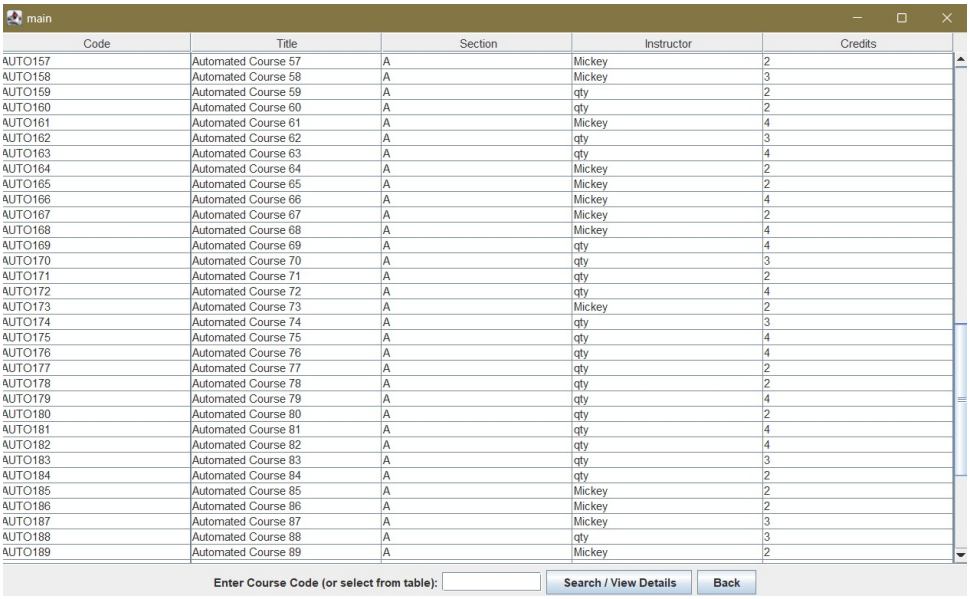

Figure 1: Course Catalog Screen

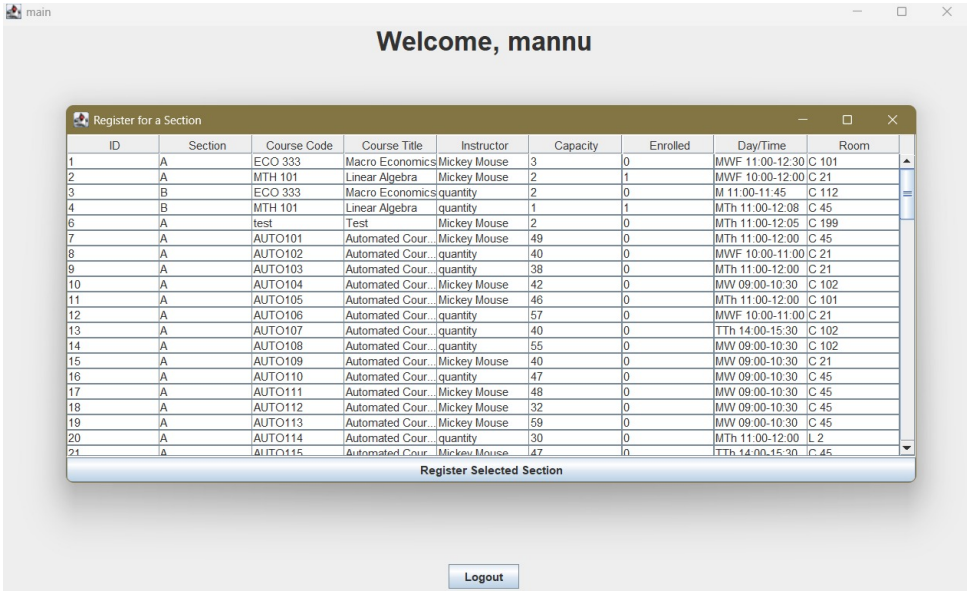

Figure 2: Registration Screen

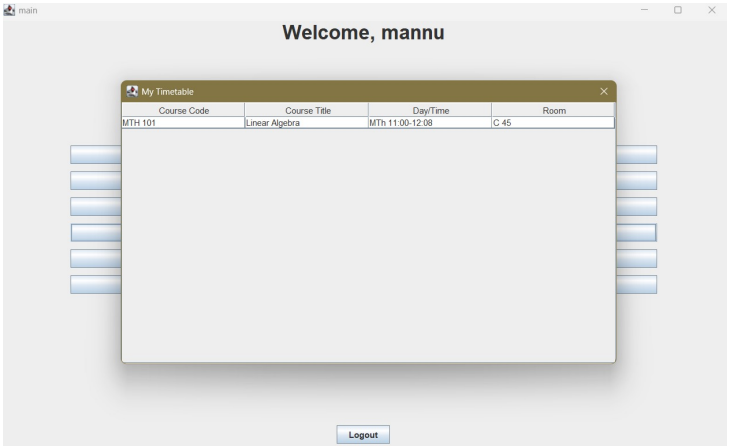

Figure 3: Timetable Screen

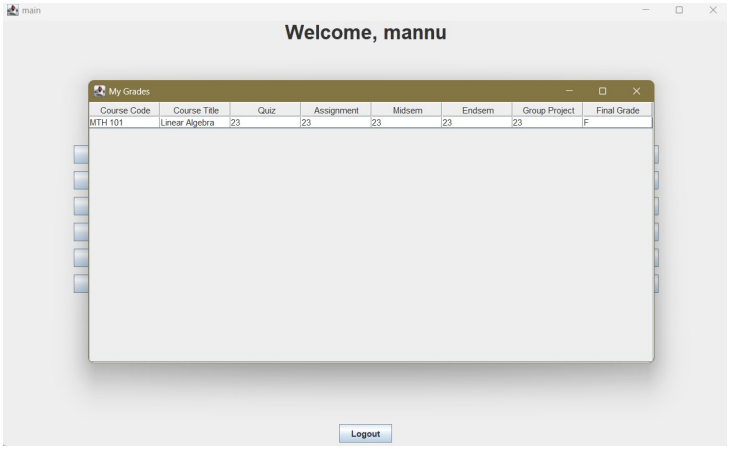

Figure 4: View grades

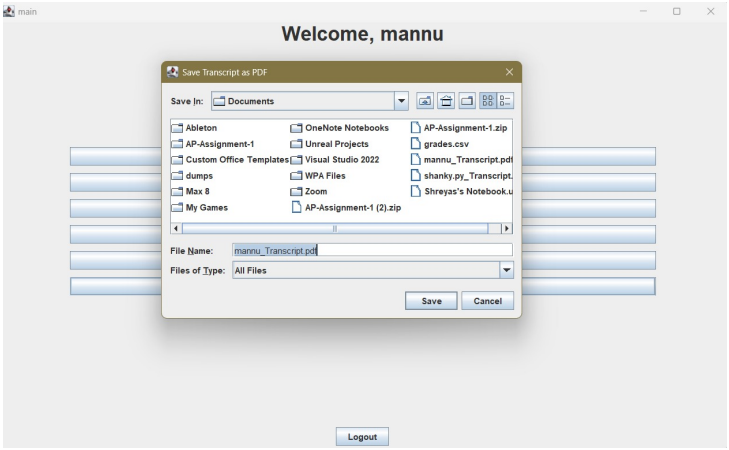

Figure 5: Transcript Export

Instructor Features

1. My Sections

Instructors only see sections assigned to them.

2. Grade Entry

Instructors can enter marks for quizzes, assignments, midsem, endsem, and project.

3. Final Grade Calculation

We used the following rule in our project:

$$FinalGrade = Quiz\% \times Quiz + Assn\% \times Assn + Midsem\% \times Midsem + Endsem\% \times Endsem$$

- These percentages vary from course to course and can also be modified with the restriction of all summing up to 100%.
- Grading Policy:-
  - Final Grade $\geq$ 95 - A+
  - Final Grade $\geq$ 90 - A
  - Final Grade $\geq$ 80 - A-
  - Final Grade $\geq$ 70 - B
  - Final Grade $\geq$ 60 - B-
  - Final Grade $\geq$ 50 - C
  - Rest - F

4. Restrictions

- Instructors cannot modify other instructors' sections.
- No grade editing is allowed during maintenance mode.

Screenshots

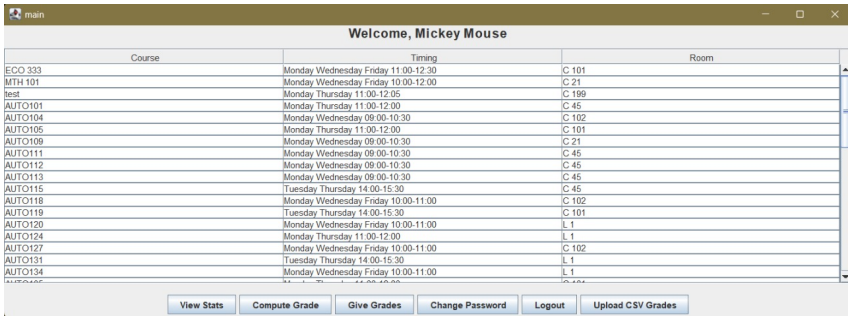

Figure 6: My Sections page

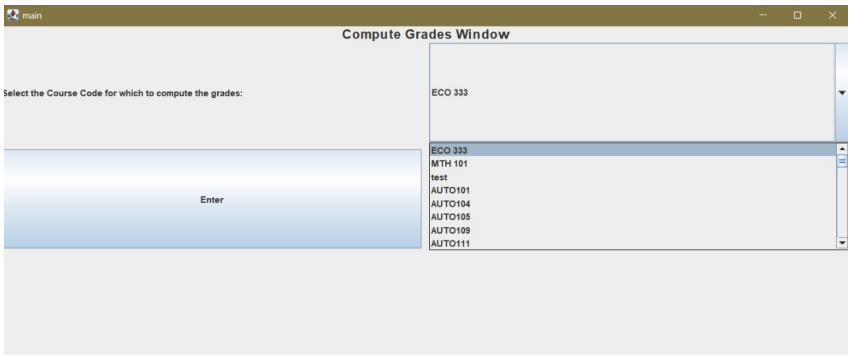

Figure 7: Compute Grades page

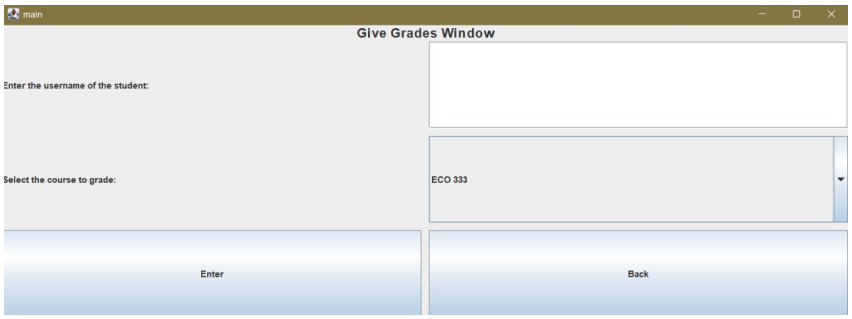

Figure 8: Give Grades page

Admin Features

1. Add Users

Admins can add students and instructors. This updates both Auth DB and ERP DB.

2. Course and Section Creation

Admins can create new courses, set capacities, assign instructors, and configure timings.

3. Maintenance Mode

Admin can turn maintenance mode ON or OFF. If ON:

- students cannot register/drop,
- instructors cannot enter grades,
- admins still have full access.

A visible banner is shown to all users during maintenance.

Screenshots

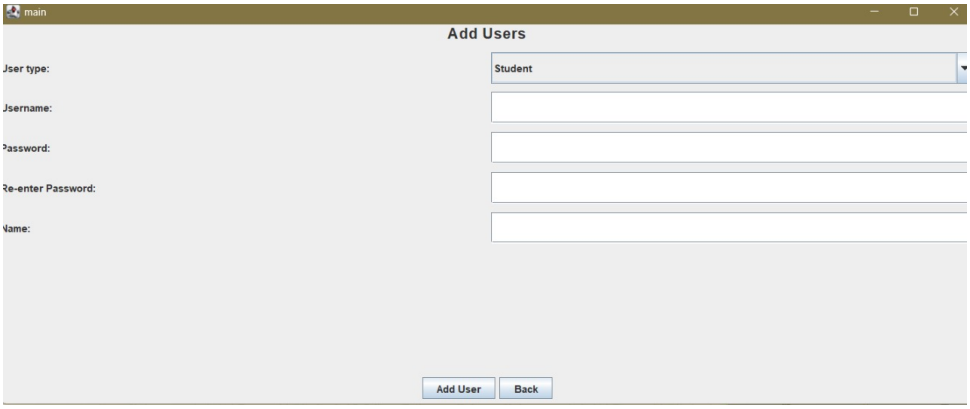

Figure 9: Add Users page

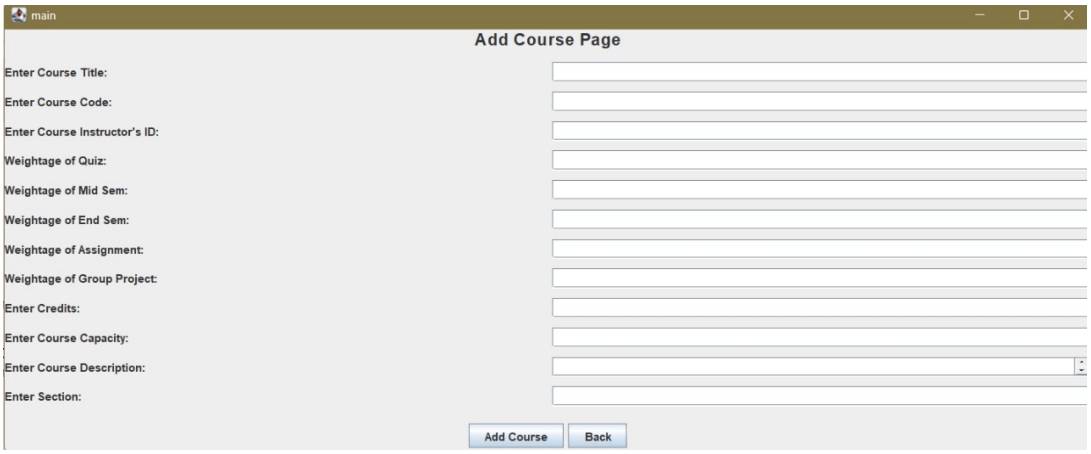

Figure 10: Add Course page

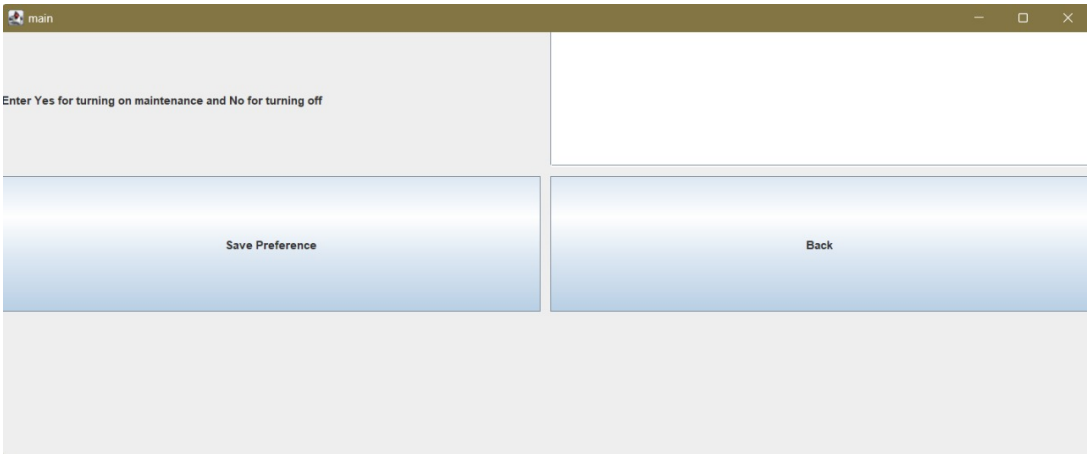

Figure 11: Set Maintenance page

5 Role Enforcement and Maintenance Mode

Role enforcement is done at the service layer. Before performing any operation, the system checks:

- **User Role** (Student/Instructor/Admin)
- **If the user owns the resource** (e.g., instructor editing only their sections)
- **Maintenance Mode Flag** (from the maintainence table)

If the user is not allowed, a pop-up message is shown and the action is blocked.

6 Extras Implemented

Apart from the required features, we added:

- CSV and PDF transcript export
- login attempt tracking (failed attempts)
- sorted tables in many screens
- clean UI with formatted dialogs
- small notification subsystem

7 How to Run the Application

- Install Java 17+ and MySQL.
- Import the project and load both SQL files (Auth DB and ERP DB).
- Update DB credentials inside the config file/classes.
- Run `Main.java`.
- In case you want to run using terminal, type these commands:-
  - `javac -cp ".;../lib/*" Main/Main.java`
  - `java -cp ".;../lib/*" Main.Main`

Sample Credentials

Password is same as their usernames.

- Admin: `sam`
- Instructor: `qty`
- Student: `shanky`

8 Conclusion

Overall, this project helped us understand how a real university ERP works internally. We implemented role-based access, separate databases for authentication and academic data, maintenance mode control, and a full registration + grading workflow. The project has scope for many more features, but even in its current form it is stable and covers all important requirements of the assignment.