

PROJECT REPORT

PROJECT MEMBERS:

1.DIVYARAJ CHAUHAN (2024204)

2.DAKSH KUMAR (2024169)

1. Introduction

1.1 Project Overview

The University ERP System is a comprehensive desktop application designed to manage the core academic and administrative operations of a university. The system streamlines course management, student registration, grading, and transcript generation while ensuring data integrity through a dual-database architecture. It facilitates interaction between three key stakeholders: **Administrators**, **Instructors**, and **Students**.

1.2 Key Features

- **Role-Based Access Control (RBAC):** Provides distinct dashboards and permissions for Admin, Instructor, and Student roles.
- **Dual Database Architecture:** Ensures separation of concerns using `university_auth_db` for credentials and `university_erp_db` for academic data.
- **Secure Authentication:** Passwords are hashed using the **BCrypt** algorithm via the `jbcrypt` library to ensure no plaintext passwords are ever stored.
- **Maintenance Mode:** A global system setting that restricts critical write operations (e.g., registration, dropping, grading) for non-admin users.
- **Academic Deadlines:** Enforces configured **Registration** and **Drop** deadlines, blocking student actions outside the approved window.

1.3 Technology Stack

Component	Technology / Library	
Language	Java (JDK 17+)	
UI Framework	Java Swing with FlatDarkLaf Theme	
Database	MySQL (via JDBC)	

Security	jBCrypt for password hashing	
Utilities	PdfExporter (mock TXT/PDF), CsvExporter (for roster/transcript)	

2. System Architecture

2.1 Database Schema

The system employs a dual-database architecture to strictly separate authentication credentials from operational data.

- **Auth Database (university_auth_db):** Stores user login credentials (username, role) and the password_hash. This ensures no academic record (ERP DB) is exposed even if the authentication database is compromised.
- **ERP Database (university_erp_db):** Stores all academic profiles and transactional data, linked only by user_id. Tables include: students, instructors, courses, sections, enrollments, grades, and settings.

2.2 Weighted Score Computation & Grading Logic

The grading system calculates a weighted average, with instructors setting the assessment components, max marks, and weight percentages for their section.

- Weighted Grade Formula: The core calculation uses the individual component score's percentage multiplied by its weightage:

$$\text{Total Weighted Score} = \sum \left(\frac{\text{Component Score}}{\text{Max Marks}} \times \text{Weightage \%} \right)$$
- **Letter Grade Assignment:** When the instructor **Publishes Grades**, the system automatically calculates the final score and assigns a letter grade based on configurable numeric cutoffs (e.g., A, B, C, D, F).

3. Module Details

3.1 Admin Module

The Admin dashboard focuses on system-wide control and data setup.

- **User Management:** Create new Student (requires Roll No, Program) or Instructor (requires Department) accounts. Includes a feature to **Reset Password** for any existing user.
- **Course & Section Management:** Full CRUD operations for **Courses** and **Sections**. **Deletion is restricted** for courses or sections if dependent enrollments exist.
- **System Controls:** Toggle **Maintenance Mode** (Read-Only Status) and configure **Registration/Drop Deadlines**.

- **Database Management:** Dedicated buttons for **Backup** and **Restore** of the ERP database.

3.2 Instructor Module

Instructors manage students in their assigned sections.

- **Gradebook:** Allows entry, updating, and deletion of multiple grade components per student for their sections.
- **Grade Security:** Grade submission is blocked if the system is in Maintenance Mode.
- **Class Stats:** Provides summary statistics including **Average** and **Median** scores per grade component, displayed alongside a dynamically generated bar chart for visual analysis.

3.3 Student Module

Students manage their schedule and academic records.

- **Course Catalog:** View all available sections. **Registration** is restricted by capacity, course conflicts, and the Admin-set registration deadline.
- **My Registrations:** View currently enrolled sections. **Drop Section** is restricted by the Admin-set drop deadline.
- **My Timetable:** Displays a weekly schedule of all enrolled classes.
- **Transcript:** Students can view their academic records (including final letter grades) and **Download** the official transcript as a **CSV** file or a mock **PDF** (exported as a formatted .txt file).

4. Compilation and Execution

4.1 Prerequisites

- Java JDK 17 or higher.
- MySQL Server running (default port 3306).
- Apache Maven for dependency management.

4.2 Database Configuration

The database credentials must be configured in
ERP/src/edu/univ/erp/data/DatabaseConnector.java.

- **Configured Database Password:** "Admin@12"
- **Database Setup:** The two seed scripts (auth_db_seed.sql, erp_db_seed.sql) must be run prior to execution.

4.3 Build and Run

1. **Build:** Run mvn clean install from the project root.
2. **Run Application:** Execute the main class: edu.univ.erp.Main.

4.4 Default Credentials

Role	Username	Password
Admin	admin1	adminpass
Instructor	inst1	instpass
Student	stu1	stupass1
Student	stu2	stupass2

5. Authentication and Security

5.1 Password Security

The system ensures passwords are never stored in plaintext by utilizing the **BCrypt hashing algorithm**.

- **Hashing Implementation:** Uses BCrypt.hashpw(password, BCrypt.gensalt()) to generate a unique, randomly salted hash for every new password.
- **Verification:** Login is validated using the secure comparison function BCrypt.checkpw(password, storedHash).
- **Secure Password Change:** All user dashboards integrate a **Change Password Dialog**, which enforces re-hashing of the new password before updating the users_auth table.

5.2 Database Separation

The architecture strictly separates user credentials from academic data via **two databases**:

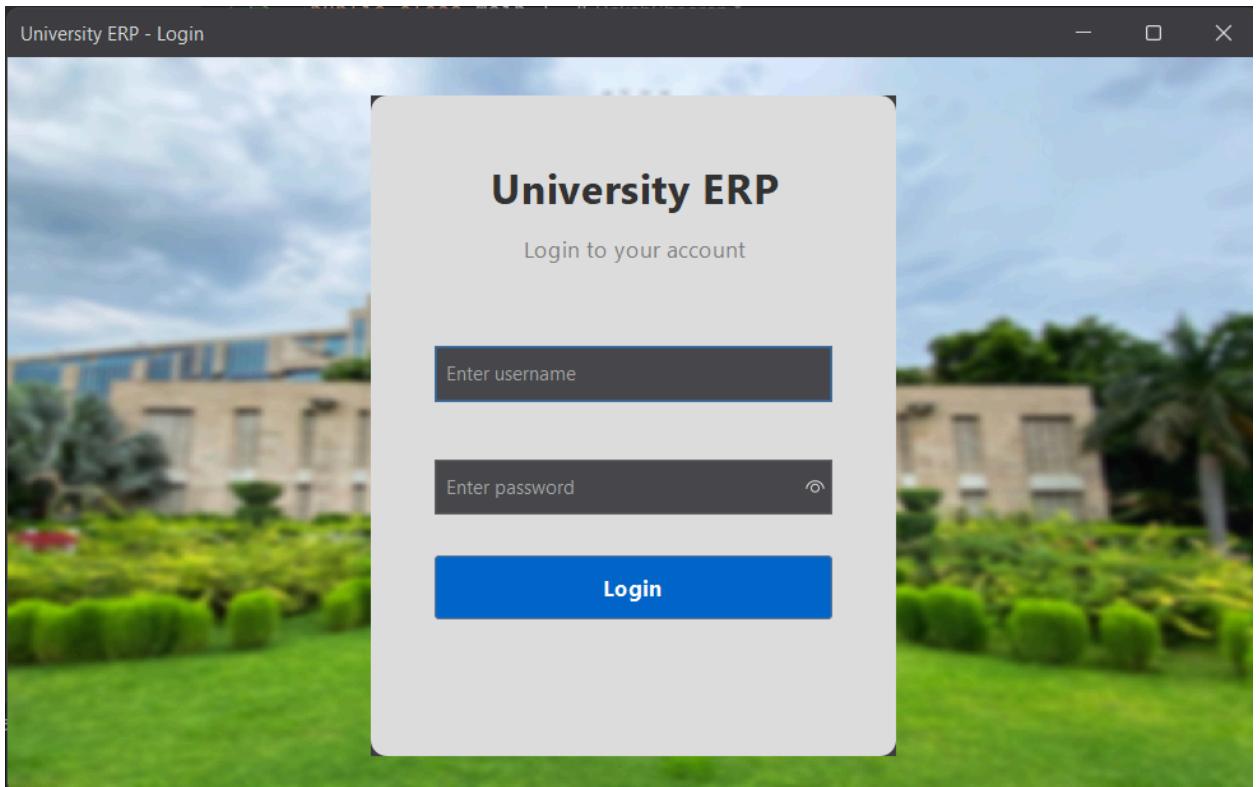
1. **university_auth_db**: Stores sensitive login data.
2. **university_erp_db**: Stores transactional data (e.g., grades, enrollment, sections).

Records are linked solely via the user_id foreign key, ensuring no credential leakage into the operational database.

6. Required Project Screenshots

1. Authentication

- **Login Window:**



2. Student Interface

- **Student Dashboard - Timetable:**

Student ERP Dashboard - stu1

University ERP

S

Course Catalog

My Registrations

My Timetable

My Grades

Transcript

My Weekly Timetable

Course Code	Title	Schedule	Room	Instructor
CS101	Intro to Computer Science	Mon: 10:00-10:50 Wed: 10:00-10:50 Fri: 10:00-10:50	A-101	inst1
DSN202	Design Thinking	Thu: 14:00-15:20 Tue: 14:00-15:20	C-205	inst1
spp	social	Sat: 10:00-10:50	c21	inst1
SOS	202	Mon 10:00-10:50 Wed 10:00-10:50	C301	inst1
21	CS	Mon 10:00-10:50 Fri 10:00-10:50	2	inst2
CSE231	AP	Mon 10:00-10:50 Thu 09:00-10:50 Fri 10:00-10:50	25	inst4

- Student Dashboard - Registered Sections:

Student ERP Dashboard - stu1

University ERP

Course Catalog

My Registrations

My Timetable

My Grades

Transcript

My Registered Sections

Course Code	Section ID	Status
CS101	1	Enrolled
DSN202	2	Enrolled
spp	3	Enrolled
SOS	4	Enrolled
21	10	Enrolled
CSE231	12	Enrolled

Last date to drop: 2025-12-01

Drop Selected Section

The screenshot shows a student dashboard titled 'Student ERP Dashboard - stu1'. On the left, there's a sidebar with links: Course Catalog, My Registrations, My Timetable, My Grades, and Transcript. The main area is titled 'University ERP' and shows a table of 'My Registered Sections'. The table has columns for Course Code, Section ID, and Status. It lists six courses: CS101, DSN202, spp, SOS, 21, and CSE231, all marked as 'Enrolled'. Below the table, a message says 'Last date to drop: 2025-12-01' and there's a button labeled 'Drop Selected Section'.

- **Student Dashboard - Academic Records:**

The screenshot shows the Student ERP Dashboard titled "Student ERP Dashboard - stu1". At the top right is a blue square icon with a white letter "S". The main header reads "University ERP". On the left, there's a vertical sidebar with five buttons: "Course Catalog", "My Registrations", "My Timetable", "My Grades", and "Transcript". The "Transcript" button is highlighted with a blue border. In the center, a dropdown menu says "Select Course to View: DSN202 - Design Thinking". Below it is a table showing academic components and their marks:

Component	Marks Obtained / Max	Weightage
Exam	77.0 / 100.0	80.0%
Quiz	88.0 / 100.0	20.0%

At the bottom, it says "Weighted Grade Summary" and displays the cumulative grade: "Current Cumulative Grade: 79.20 / 100.00". It also shows the "Official Final Grade: B-".

3. Instructor Interface

- **Instructor Dashboard - Gradebook:**
-
-

Instructor Dashboard - inst1

Instructor Panel

Select Section: 2: DSN202

Roll No	Name
2024001	stu1
2024002	stu2

Grades for Selected Student

Component	Score	Max Marks	Weightage %
Exam	77.0	100.0	80.0
Quiz	88.0	100.0	20.0

Manage Grades

Component Name:

Score Obtained:

Max Marks (e.g. 50):

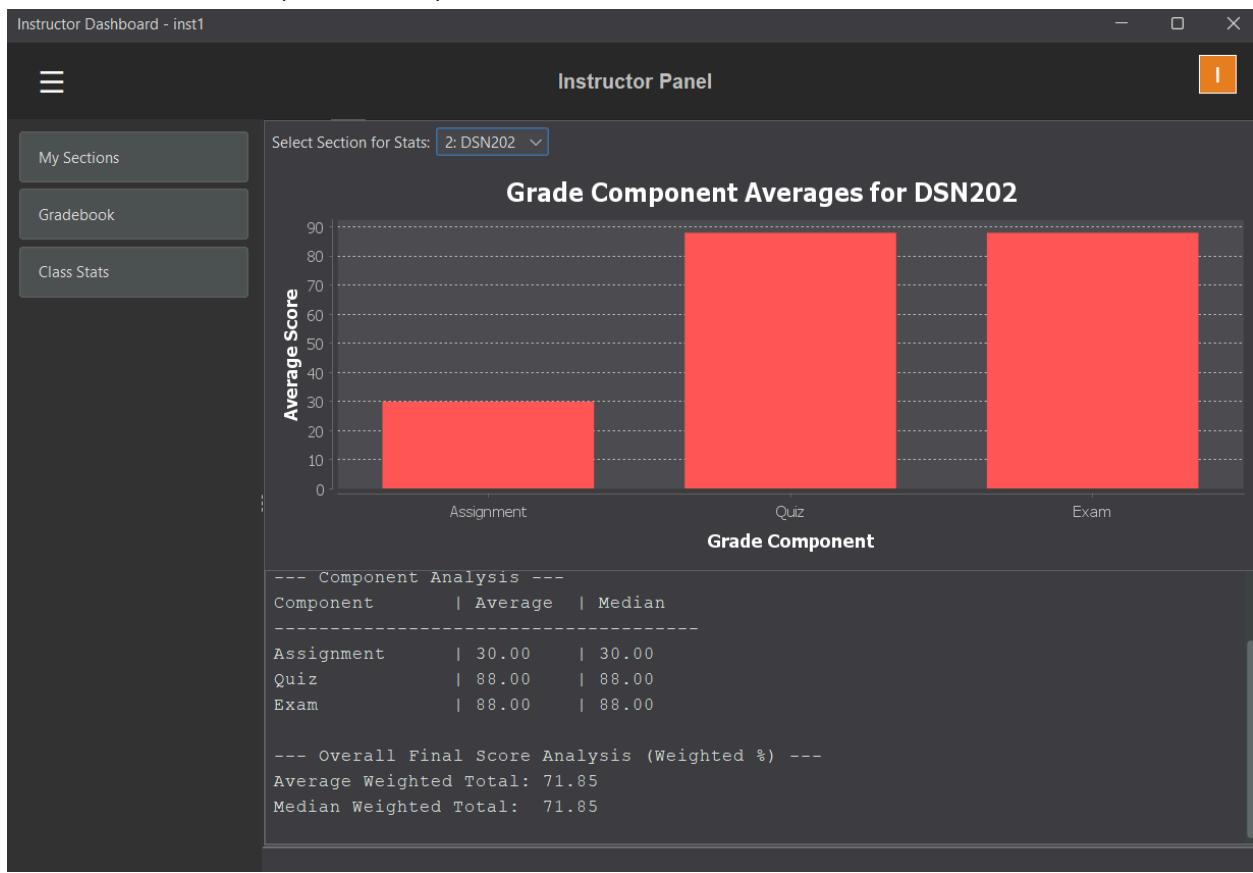
Weightage % (e.g. 20):

Save Grade

Publish Grades for Class **Export Roster (CSV)**

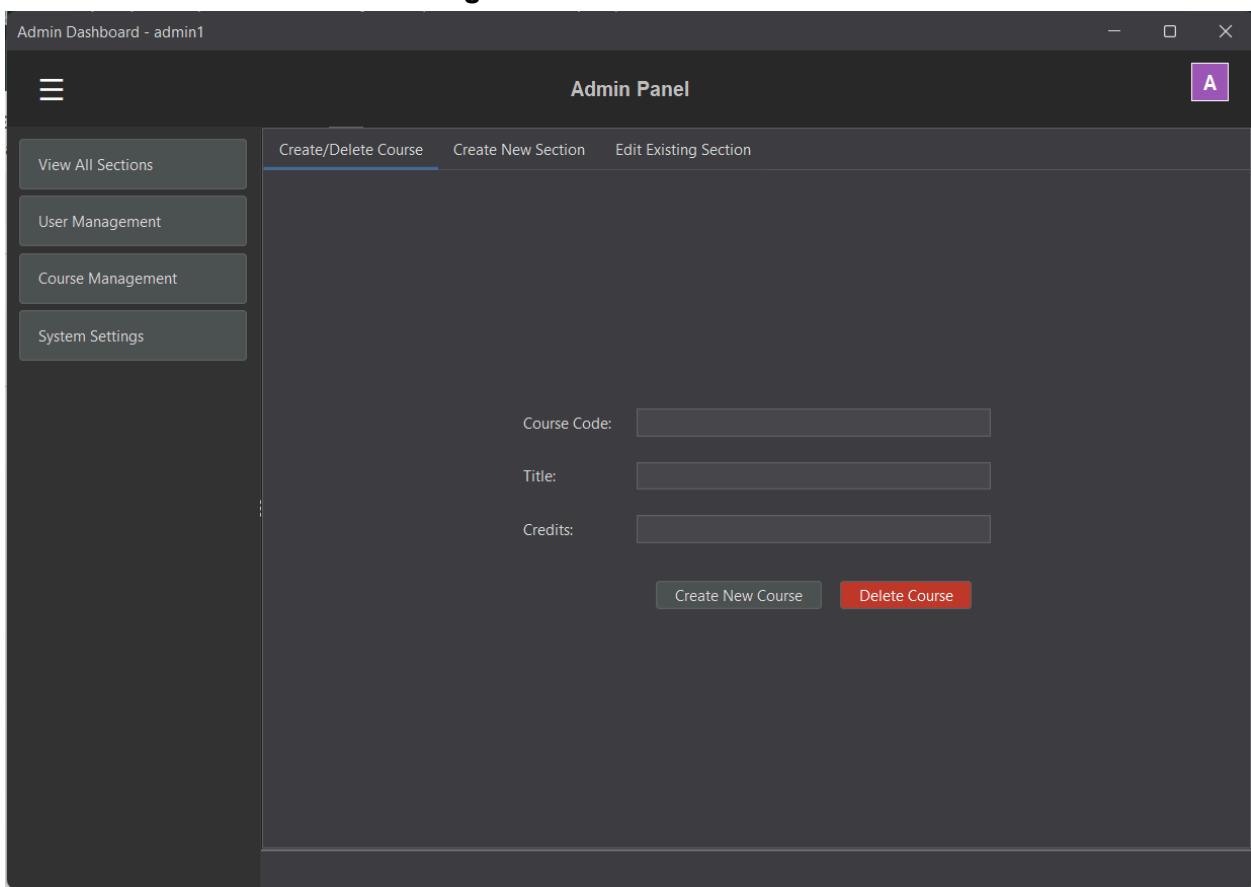
The screenshot shows the 'Instructor Panel' of a dashboard titled 'Instructor Dashboard - inst1'. On the left, there's a sidebar with three buttons: 'My Sections', 'Gradebook', and 'Class Stats'. The main area has a dropdown for 'Select Section' set to '2: DSN202'. Below it is a table with two rows: one for 'stu1' (Roll No: 2024001) and one for 'stu2' (Roll No: 2024002). To the right, a table titled 'Grades for Selected Student' shows two components: 'Exam' and 'Quiz'. The 'Exam' component has a score of 77.0, max marks of 100.0, and a weightage of 80.0. The 'Quiz' component has a score of 88.0, max marks of 100.0, and a weightage of 20.0. Below these tables is a section for 'Manage Grades' with fields for Component Name, Score Obtained, Max Marks, and Weightage %. At the bottom are buttons for 'Publish Grades for Class' and 'Export Roster (CSV)', and a large green button labeled 'Save Grade'.

- Instructor Dashboard(Class Stats)-



4. Admin Interface

- **Admin Dashboard - Course Management:**



- **Admin Dashboard - User Management:**

The screenshot shows the Admin Panel interface for managing users. On the left, a sidebar menu includes 'View All Sections', 'User Management' (which is selected), 'Course Management', and 'System Settings'. The main area is titled 'Admin Panel' and contains two sections: 'Create New User' and 'Existing Users'.

Create New User: This section contains fields for 'Username', 'Password', 'Role' (set to 'Student'), 'Roll Number', and 'Program'. A 'Create User' button is at the bottom.

Existing Users: This section includes a dropdown 'View: Students' and buttons for 'Refresh List' and 'Reset Password'. A table lists existing users with columns for 'Username', 'Roll Number', and 'Program'.

Username	Roll Number	Program
stu1	2024001	B.Tech CS
stu2	2024002	B.Tech CSD
stu4	2024003	B.Tech CSAM
stu5	2024005	B.Tech
stu6	2024006	Btech
stu7	2024007	BTech
stu3	2024201	btech

- **Admin Dashboard - System Settings:**

The screenshot shows the Admin Panel interface with the title "Admin Dashboard - admin1" at the top left. On the right side, there is a purple square icon containing a white letter "A". The main area is titled "Admin Panel" and contains several sections:

- Maintenance Mode Toggle:** Shows the status as "OFF (Active)" in green text. A red button labeled "Turn ON" is visible.
- Academic Calendar Deadlines:** Includes fields for "Registration Deadline (YYYY-MM-DD):" set to "2025-12-31" and "Drop Deadline (YYYY-MM-DD):" set to "2025-12-31". A blue button labeled "Update Dates" is below these fields. Below the buttons, it says "Current Settings: Register by: 2025-12-31 | Drop by: 2025-12-31".
- Database Management (ERP DB):** Contains two buttons: "Backup ERP DB" (green) and "Restore ERP DB" (red).

A sidebar on the left lists navigation options: "View All Sections", "User Management" (which is currently selected and highlighted in blue), "Course Management", and "System Settings".

-

- **Change Password Dialog:**

