Test Plan — IIIT DELHI ERP

This document outlines the testing strategy for the IIIT DELHI ERP desktop application, focusing on functional verification, role-based access control, and core data integrity.

1. Purpose and Scope

1.1. Purpose

The primary goal of this test plan is to define the necessary functional tests required to validate the IIIT DELHI ERP desktop application, implemented in **Java + Swing**, against its core specifications. Verification focuses on authentication, role-based access control, student enrollment, grade recording, and initial resource loading behaviours.

1.2. Scope

The testing scope covers the following critical areas:

- **Application Startup:** Verify the application correctly initializes and loads expected resources (e.g., configuration files, JDBC drivers).
- **Authentication & Roles:** Validate successful and unsuccessful login attempts for **Admin** , **Instructor** , and **Student** accounts, and confirm role-matched dashboard access.
- **Core Functionality:** Verify basic create/update actions, including **Admin** user creation , **Change Password** , student section **Enrollment** , and instructor **Grade Entry**.
- **Access Control:** Ensure strict RBAC enforcement—for instance, students cannot execute admin functions, and instructors are restricted to managing their own assigned sections.
- **Data Integrity:** Verify mechanisms that prevent critical errors, such as **Duplicate Enrollments**.

2. Test Environment and Prerequisites

2.1. Test Environment Specification

Component	Specification	Notes
Operating System	Windows 10/11 (Developer Environment)	Standard desktop OS for running the Java Swing application.
Java Version	JDK 17+	Minimum required version; JDK 24 validated during development.
Database System	PostgreSQL	Running locally; connections are split between two databases.
Configuration File	`config/app.properties`	Defines connection settings for `univ_auth` and `univ_erp` databases.
JDBC Driver	`lib/postgresql-42.7.4.jar`	Must be explicitly included on the runtime classpath.
Codebase	`AP-Project`	Project workspace opened in VS Code.

2.2. Prerequisites

The following steps must be completed before test execution:

1. **Database Server:** Ensure the PostgreSQL server is running and reachable via the network.
2. **Schema Application:** Run `sql/auth_schema.sql` and `sql/erp_schema.sql` to create the required tables in the two separate databases.
3. **Data Seeding:** Load the initial test dataset by executing `testing/test_data.sql`. This script populates the databases with the necessary test users, courses, and sections.
4. **Project Build:** Compile the Java project by running `compile.bat` from the project root directory.
5. **Application Launch:** Launch the application using the recommended configuration (refer to `.vscode/launch.json`), or execute manually from the project root:

```
java -cp "bin;lib\postgresql-42.7.4.jar" edu.univ.erp.Main
```

3. Test Cases (Functional & Access Control)

ID	Test Case Title	Steps	Test Data	Expected Result
T1	**Application Startup and Resource Load**	**Start application via launch configuration.**	**N/A**	**Application starts without critical errors (e.g., `ClassNotFoundException` for JDBC); `config/app.properties` loaded; UI resources (e.g., background image) display correctly.**

T2	**Admin Login (Happy Path)**	**Login with provided Admin credentials.**	**admin1 / admin123**	**Login succeeds; the Admin dashboard opens with full management controls.**
T3	**Instructor Login**	**Login with provided Instructor credentials.**	**inst1 / inst123**	**Login succeeds; the Instructor dashboard opens, limited to their assigned sections.**
T4	**Student Login**	**Login with provided Student credentials.**	**stu1 / stu123**	**Login succeeds; the Student dashboard opens, limited to their own records.**
T5	**Invalid Login**	**Attempt login with a correct username but incorrect password.**	**admin1 / wrongpass**	**Login fails with a clear error message (e.g., "incorrect username or password") ; application remains stable.**
T6	**Access Control Enforcement**	**Log in as Student (stu1) and attempt to access an Admin-only feature (e.g.,**	**stu1**	**Action is blocked by the system; the user sees a clear permission error message (e.g., "Action not allowed").**

		Create User screen).		
T7	**Change Password**	**As Student (`stu1`), navigate to Change Password dialog, update password, then log out and re-login using the new password.**	**Old: `stu123`, New: `newpass`**	**Password change succeeds; new password authenticates; old password is rejected upon re-login.**
T8	**Enroll Student in Section**	**As Student (`stu1`) via the UI flow, enroll in section S1.**	**`stu1`, Section S1**	**Enrollment record created in `enrollments` table; UI reflects the new enrollment in the student's timetable/registrations.**

T9	**Prevent Duplicate Enrollment**	**Attempt to enroll Student (stu1) in section S1 again immediately after successful T08 enrollment.**	**stu1, Section S1**	**The UI/Service layer prevents the action; a clear error message is shown (e.g., "Section already registered").**
T10	**Instructor Grade Entry**	**As Instructor (inst1) for section S1, enter a score for Student (stu1).**	**inst1, Section S1, Score: 85**	**The grade record is successfully created/updated in the grades table ; score is visible under the student's grade view.**

4. Pass/Fail Criteria

- **A test case passes when the actual system behaviour matches the expected result without unexpected application termination or unhandled exceptions.**
- **Overall Acceptance: The project is accepted for grading if all critical tests (T1–T6, T8–T9) pass successfully.**

5. Notes and Guidance

- **VS Code Execution: If running the app via VS Code, ensure the `Launch Main (AP-Project)` configuration is selected and that the `cwd` (Current Working Directory) correctly points to the project root. This guarantees that `config/app.properties` and any required image assets are correctly located.**
- **Connectivity Debugging: In case of JDBC connectivity failures, always verify the connection settings in `config/app.properties` and confirm that the PostgreSQL server is actively running and the provided user credentials are valid.**