How to run:

Open terminal

Enter the commands as follows (one after the other):

```
javac -cp ".;mysql-connector-j-9.4.0.jar;jbcrypt-0.4.jar" database\*.java
auth\*.java ui\login\*.java ui\studentdash\*.java Main.java

java -cp ".;mysql-connector-j-9.4.0.jar;jbcrypt-0.4.jar" Main
```

- First command: **compiles** all your source files with external libraries available.
- Second command: **runs** the program, starting from Main.
- Or using the bat file ./erpbuild.bat and then press any key and then ./erprun.bat and hit enter

Then , the login screen appears:

1. **Java Version : Java 24.0.2**

   java version "24.0.2" 2025-07-15 Java(TM) SE Runtime Environment (build 24.0.2+12-54) Java HotSpot(TM) 64-Bit Server VM (build 24.0.2+12-54, mixed mode, sharing)

2. **Database Setup**

   ERP System Database

   │

   ├── Auth DB

   │ ├── Users (user_id, username, email,password_hash,role,created_at)

   │

   └── ERP DB
   ├── students (student_id, user_id, roll_no, name, program_type, semester, hostel)

   ├── instructors (instructor_id, user_id, name, department)

   ├── ta (ta_id, instructor_id, student_id)

   ├── courses (course_id, code, title, credits, department,allowed_programs,min_semester,min_cgpa)

   ├── sections (section_id, course_id, instructor_id, day_time, capacity,semester,year,grade_locked,temp_locked,final_locked)

   ├── enrollments (enrollment_id, student_id, section_id, status,)

├── student_component_marks (enrollment_id, component_id, marks_obtained, max_marks, graded_by, graded_time)

├── student_component_marks_backup (backup_id, enrollment_id, component_id, marks_obtained, max_marks, graded_by, graded_time)

├── course_components (component_id, course_id,componetnt_ name, weightage,created_by, max_marks)

├── fees (fee_id,student_id,total_amount,paid_amount,due_date)

├── semester_result (result_id,student_id,semester,sgpa)

├── course_offerings (offering_id,course_id,semester,year,offered_to_program,created_by)

├── course_prerequisites (course_id,prereq_course_id)

├── admin_settings (id,registeration_lock,maintainance_mode,announcement,instructor_announcement)

└── grades (grade_id, enrollment_id, grade, grade_point)

3. **Database Connection Settings**

   Settings are done in code as follows:

   - private static final String DB_URL = "jdbc:mysql://mysql-2afd9e51-vikast42.c.aivencloud.com:13280/auth_db";
   - private static final String DB_USER = "avnadmin";
   - private static final String DB_PASS = "AVNS_Bhw7Z8oFBh2yS-oKN9b"

Default Accounts:

- Seed data should include at least:
  - **Admin** →'admin'
  - **Instructor** → 'drmehta' , 'drsharma'
  - **Students** →'nirvan' , 'priya' , 'rahul' , 'anita'

    For all users password remains the same as 'user@123'

- Passwords are stored as **hashes** in the Auth DB (using jBCrypt).

user_id	username	email	password_hash	role	created_at
1	nirvan	nirvan@iiitd.com	$2a$12$yHlV8uRh96qq43MvA72CmeNw0F3Bs....	student	2025-10-28 16:54:31
2	priya	priya@iiitd.com	$2a$12$yHlV8uRh96qq43MvA72CmeNw0F3Bs....	student	2025-10-28 16:54:31
4	drmehta	drmehta@iiitd.com	$2a$12$yHlV8uRh96qq43MvA72CmeNw0F3Bs....	instructor	2025-10-28 16:54:31
5	drsharma	drsharma@iiitd.com	$2a$12$yHlV8uRh96qq43MvA72CmeNw0F3Bs....	instructor	2025-10-28 16:54:31
6	admin	admin@iiitd.com	$2a$12$yHlV8uRh96qq43MvA72CmeNw0F3Bs....	admin	2025-10-28 16:54:31
7	rahul	rahul@iiitd.com	$2a$12$yHlV8uRh96qq43MvA72CmeNw0F3Bs....	student	2025-10-28 16:56:44
8	anita	anita@iiitd.com	$2a$12$yHlV8uRh96qq43MvA72CmeNw0F3Bs....	student	2025-10-28 16:56:44

‘auth_db’ seed script:

```
-- Create Auth DB
CREATE DATABASE IF NOT EXISTS auth_db;
USE auth_db;

-- Users table
CREATE TABLE IF NOT EXISTS Users (
    user_id INT AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(50) UNIQUE NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL,
    password_hash VARCHAR(255) NOT NULL,
    role VARCHAR(20) NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

-- Insert default accounts (passwords must be hashed with jBCrypt in code)
INSERT INTO Users (username, email, password_hash, role) VALUES
('admin', 'admin@example.com', '<bcrypt_hash>', 'Admin'),
('drmehta', 'mehta@example.com', '<bcrypt_hash>', 'Instructor'),
('drsharma', 'sharma@example.com', '<bcrypt_hash>', 'Instructor'),
('nirvan', 'nirvan@example.com', '<bcrypt_hash>', 'Student'),
('priya', 'priya@example.com', '<bcrypt_hash>', 'Student'),
('rahul', 'rahul@example.com', '<bcrypt_hash>', 'Student'),
('anita', 'anita@example.com', '<bcrypt_hash>', 'Student');
```

‘erp_db’ seed script:

```
-- Create ERP DB
CREATE DATABASE IF NOT EXISTS erp_db;
USE erp_db;

-- Students table
CREATE TABLE IF NOT EXISTS students (
    student_id INT AUTO_INCREMENT PRIMARY KEY,
```

```
    user_id INT NOT NULL,
    roll_no VARCHAR(20) UNIQUE NOT NULL,
    name VARCHAR(100) NOT NULL,
    program_type VARCHAR(50),
    semester INT,
    hostel VARCHAR(50)
);

-- Instructors table
CREATE TABLE IF NOT EXISTS instructors (
    instructor_id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT NOT NULL,
    name VARCHAR(100),
    department VARCHAR(50)
);

-- TA table
CREATE TABLE IF NOT EXISTS ta (
    ta_id INT AUTO_INCREMENT PRIMARY KEY,
    instructor_id INT,
    student_id INT
);

-- Courses table
CREATE TABLE IF NOT EXISTS courses (
    course_id INT AUTO_INCREMENT PRIMARY KEY,
    code VARCHAR(20),
    title VARCHAR(100),
    credits INT,
    department VARCHAR(50),
    allowed_programs VARCHAR(100),
    min_semester INT,
    min_cgpa DECIMAL(3,2)
);

-- Sections table
CREATE TABLE IF NOT EXISTS sections (
    section_id INT AUTO_INCREMENT PRIMARY KEY,
    course_id INT,
    instructor_id INT,
    day_time VARCHAR(50),
    capacity INT,
    semester INT,
    year INT,
```

```
    grade_locked BOOLEAN DEFAULT FALSE,
    temp_locked BOOLEAN DEFAULT FALSE,
    final_locked BOOLEAN DEFAULT FALSE
);

-- Enrollments table
CREATE TABLE IF NOT EXISTS enrollments (
    enrollment_id INT AUTO_INCREMENT PRIMARY KEY,
    student_id INT,
    section_id INT,
    status VARCHAR(20)
);

-- Student Component Marks
CREATE TABLE IF NOT EXISTS student_component_marks (
    enrollment_id INT,
    component_id INT,
    marks_obtained DECIMAL(5,2),
    max_marks DECIMAL(5,2),
    graded_by VARCHAR(50),
    graded_time TIMESTAMP
);

-- Student Component Marks Backup
CREATE TABLE IF NOT EXISTS student_component_marks_backup (
    backup_id INT AUTO_INCREMENT PRIMARY KEY,
    enrollment_id INT,
    component_id INT,
    marks_obtained DECIMAL(5,2),
    max_marks DECIMAL(5,2),
    graded_by VARCHAR(50),
    graded_time TIMESTAMP
);

-- Course Components
CREATE TABLE IF NOT EXISTS course_components (
    component_id INT AUTO_INCREMENT PRIMARY KEY,
    course_id INT,
    component_name VARCHAR(50),
    weightage DECIMAL(5,2),
    created_by VARCHAR(50),
    max_marks DECIMAL(5,2)
);
```

```
-- Fees table
CREATE TABLE IF NOT EXISTS fees (
    fee_id INT AUTO_INCREMENT PRIMARY KEY,
    student_id INT,
    total_amount DECIMAL(10,2),
    paid_amount DECIMAL(10,2),
    due_date DATE
);

-- Semester Result
CREATE TABLE IF NOT EXISTS semester_result (
    result_id INT AUTO_INCREMENT PRIMARY KEY,
    student_id INT,
    semester INT,
    sgpa DECIMAL(4,2)
);

-- Course Offerings
CREATE TABLE IF NOT EXISTS course_offerings (
    offering_id INT AUTO_INCREMENT PRIMARY KEY,
    course_id INT,
    semester INT,
    year INT,
    offered_to_program VARCHAR(50),
    created_by VARCHAR(50)
);

-- Course Prerequisites
CREATE TABLE IF NOT EXISTS course_prerequisites (
    course_id INT,
    prereq_course_id INT
);

-- Admin Settings
CREATE TABLE IF NOT EXISTS admin_settings (
    id INT AUTO_INCREMENT PRIMARY KEY,
    registration_lock BOOLEAN DEFAULT FALSE,
    maintenance_mode BOOLEAN DEFAULT FALSE,
    announcement TEXT,
    instructor_announcement TEXT
);

-- Grades table
CREATE TABLE IF NOT EXISTS grades (
```

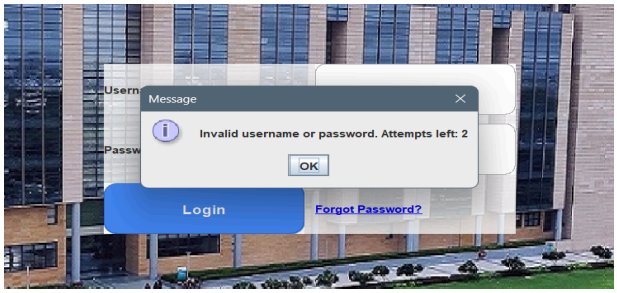
```
    grade_id INT AUTO_INCREMENT PRIMARY KEY,
    enrollment_id INT,
    grade VARCHAR(5),
    grade_point DECIMAL(3,2)
);

-- Insert sample data
INSERT INTO courses (code, title, credits, department, allowed_programs, min_semester,
min_cgpa) VALUES
('CS101', 'Intro to Programming', 4, 'CSE', 'B.Tech', 1, 0.0),
('MA201', 'Discrete Mathematics', 3, 'Math', 'B.Tech', 2, 6.0);

INSERT INTO students (user_id, roll_no, name, program_type, semester, hostel) VALUES
(4, '2025CS01', 'Nirvan', 'B.Tech CSE', 1, 'Hostel A'),
(5, '2025CS02', 'Priya', 'B.Tech CSE', 1, 'Hostel B'),
(6, '2025CS03', 'Rahul', 'B.Tech CSE', 1, 'Hostel C'),
(7, '2025CS04', 'Anita', 'B.Tech CSE', 1, 'Hostel D');

INSERT INTO instructors (user_id, name, department) VALUES
(2, 'Dr. Mehta', 'CSE'),
(3, 'Dr. Sharma', 'Math');
```

Acceptance Tests (from requirements):

2.1 Authentication

- **Test:** Login with default accounts (‘admin’,’drmehta’,’nirvan’).
- **Expected Result:** System authenticates using hashed passwords in ‘auth_db’ and opens the correct dashboard (Admin, Instructor, Student).

	user_id	username	email	password_hash	role	created_at
	1	nirvan	nirvan@iiitd.com	$2a$12$yHlV8uRh96qq43MvA72CmeNw0F3Bs....	student	2025-10-28 16:54:31
	2	priya	priya@iiitd.com	$2a$12$yHlV8uRh96qq43MvA72CmeNw0F3Bs....	student	2025-10-28 16:54:31
	4	drmehta	drmehta@iiitd.com	$2a$12$yHlV8uRh96qq43MvA72CmeNw0F3Bs....	instructor	2025-10-28 16:54:31
	5	drsharma	drsharma@iiitd.com	$2a$12$yHlV8uRh96qq43MvA72CmeNw0F3Bs....	instructor	2025-10-28 16:54:31
	6	admin	admin@iiitd.com	$2a$12$yHlV8uRh96qq43MvA72CmeNw0F3Bs....	admin	2025-10-28 16:54:31
	7	rahul	rahul@iiitd.com	$2a$12$yHlV8uRh96qq43MvA72CmeNw0F3Bs....	student	2025-10-28 16:56:44
	8	anita	anita@iiitd.com	$2a$12$yHlV8uRh96qq43MvA72CmeNw0F3Bs....	student	2025-10-28 16:56:44

- Also it gives error on wrong authentication:
- 

2.2 Role-Based Access

- **Test:** Admin can access *Admin Settings*, Instructor can access *Course Offerings*, Student can access *Enrollments*.
- **Expected Result:** Each role sees only its permitted modules.(to see images kindly open the short report ,it has all screenshots related to the different dashboards)

2.3 Course Management

- **Test:** Admin adds a new course .(even can edit,delete the existing course)
- **Expected Result:** Course appears in manage courses simultaneously

2.4 Enrollment

- **Test:** Student enrolls in a section.
- **Expected Result:** Record is created in 'enrollments' table with status = "active."

2.5 Grading

- **Test:** Instructor enters marks for a component.
- **Expected Result:** Entry saved in student_component_marks with correct marks_obtained and graded_by.

3. Extra Tests Added

3.1 Data Integrity

- **Test:** Attempt duplicate enrollment for same student and section.
- **Expected Result:** System prevents duplicate entry.

3.2 Locking Mechanisms

- **Test:** Admin enables registeration_lock..
- **Expected Result:** Students cannot enroll until lock is disabled.

3.3 Backup Consistency

- **Test:** Marks entry triggers backup.
- **Expected Result:** Record also appears in student_component_marks_backup.

3.4 Forgot password:

- **Test:** user enter correct registration email
- **Expected Result:** Allowed to set a new password .

4. Accounts and Data Used

- **Admin Accounts:**
  - `admin` (password: `user@123`)
- **Instructor Accounts:**
  - **`drmehta` (password: `user@123`)**
  - **`drsharma` (password: `user@123`)**
- **Student Accounts:**
  - **`nirvan`, `priya`, `rahul`, `anita` (password: `user@123`)**

Test Summary:

Authentication worked correctly, with valid logins opening the appropriate dashboards and invalid credentials being rejected with clear error messages. Role-based access was enforced, ensuring that admins, instructors, and students could only access their respective modules. Core workflows such as course creation, enrollment, grading, and transcript export were validated successfully, with records appearing in the correct tables (`courses`, `enrollments`, `student_component_marks`) and reflected in the UI.

Extra integrity checks confirmed that duplicate enrollments were blocked, registration locks prevented new enrollments when enabled, and marks entries were consistently backed up. The forgot password flow allowed users to reset credentials securely. Data integrity and security requirements were met, with passwords stored only as hashes in the Auth DB and no sensitive information present in the ERP DB. Overall, the ERP system passed all acceptance and additional tests, demonstrating reliable functionality, secure authentication, and proper role-based workflows.

(known issue: the erp runs slow on other's laptop other than the host , however it runs without any delay in the localhost laptop)