

# **UNIVERSITY ERP SYSTEM**

## **Final Report (Java + MySQL Desktop Application)**

### **1. Overview**

The University ERP System is a Java Swing–based desktop application built to simplify and centralize the management of university academic processes. It provides three user roles, Admin, Instructor, and Student, each with secure and restricted access to relevant features.

Data is stored and managed using two databases:

- a. Auth DB (auth\_db) - Handles user authentication, password security, and account status.
- b. ERP DB (erp\_db) - Manages students, instructors, courses, sections, enrollments, grades, and system settings.

### **2. System Architecture**

a. Dual Database Design

b. Auth DB

Description: Authentication and user management

Key Tables: users\_auth

c. ERP DB

Description: Academic records, enrollments, grades, settings

Key Tables: students, instructors, courses, sections, enrollments, grades, notifications, system-settings.

d. The two databases are linked through the user\_id field, ensuring every user in auth\_db has a corresponding academic profile in erp\_db.

e. Directory Overview

- edu.univ.erp.auth — Authentication, session control, lockout system
- edu.univ.erp.data — JDBC database connectivity
- edu.univ.erp.service — Core logic for each user role
- edu.univ.erp.ui — Java Swing dashboards & dialogs
- edu.univ.erp.util — Utilities (CSV/PDF export, bcrypt hashing, etc.)

### **3. Roles and Permissions**

a. Admin, key roles

- Manage users, courses, sections
- Toggle Maintenance Mode
- Assign instructors to sections
- Delete courses/sections
- Full write-access anytime

b. Instructor, key roles

- Manage only their assigned sections
- Record assessment scores
- Compute final grades
- View class statistics
- Import/export grades via CSV

c. Student, key roles

- Browse available courses
- Register for sections if capacity allows
- Drop sections before deadline
- View timetable
- View component-wise and final grades
- Export transcript (CSV/PDF)

## 4. Final-Grade Weighting Rule

Final grades use this weighted average:

Quiz — 20%

Midterm — 30%

Endsem — 50%

Formula: Final Score =  $(0.2 \times \text{Quiz}) + (0.3 \times \text{Midterm}) + (0.5 \times \text{Endsem})$

Grade Boundaries:

A: 85–100

B: 70–84

C: 55–69

D: 40–54

F: <40

The system auto-inserts FINAL results into the grades table.

## 5. Maintenance Mode & Enforcement

Maintenance Mode is stored in `system_settings`.

When ON:

- Students and Instructors can log in but cannot modify data
- All write actions trigger friendly blocking messages
- Admin retains full privileges
- UI shows warnings and read-only lockouts

Enforced using `SettingsService.checkMaintenanceBlock` and `Session.isReadOnly` across all services.

## 6. Database Design (Table Lists)

Auth DB (`auth_db`)

`users_auth`: `user_id`, `username`, `role`, `password_hash`, `status`, `last_login`, `failed_attempts`, `last_failed`

ERP DB (erp\_db)

students: student\_id, user\_id, roll\_no, program, year

instructors: instructor\_id, 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\_score, final\_grade

notifications: id, message, created\_at

system\_settings: id, maintenance

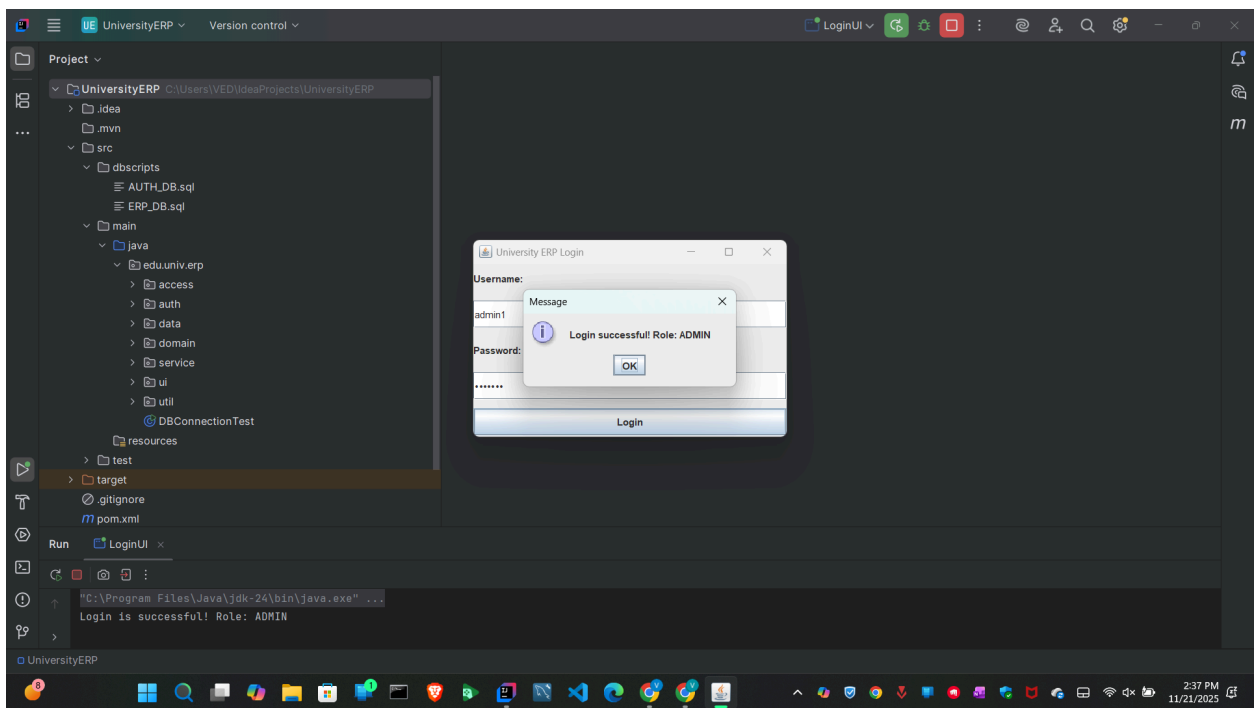
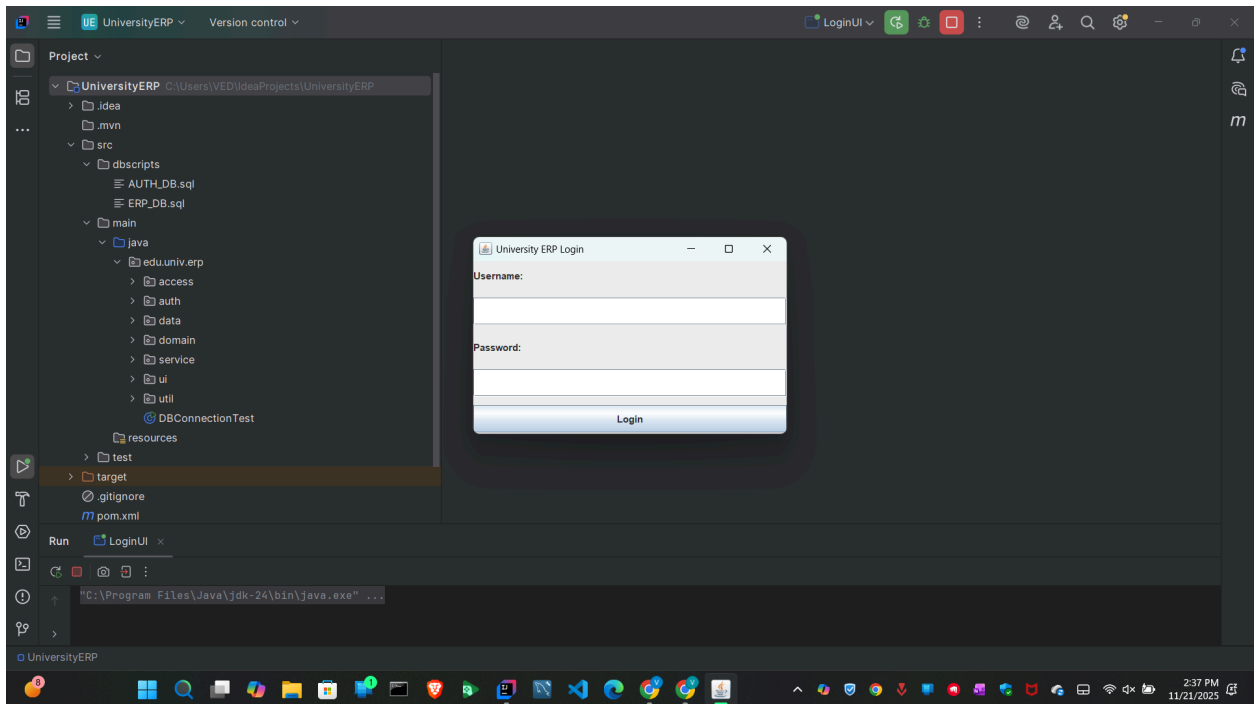
## **7. Extras & Enhancements**

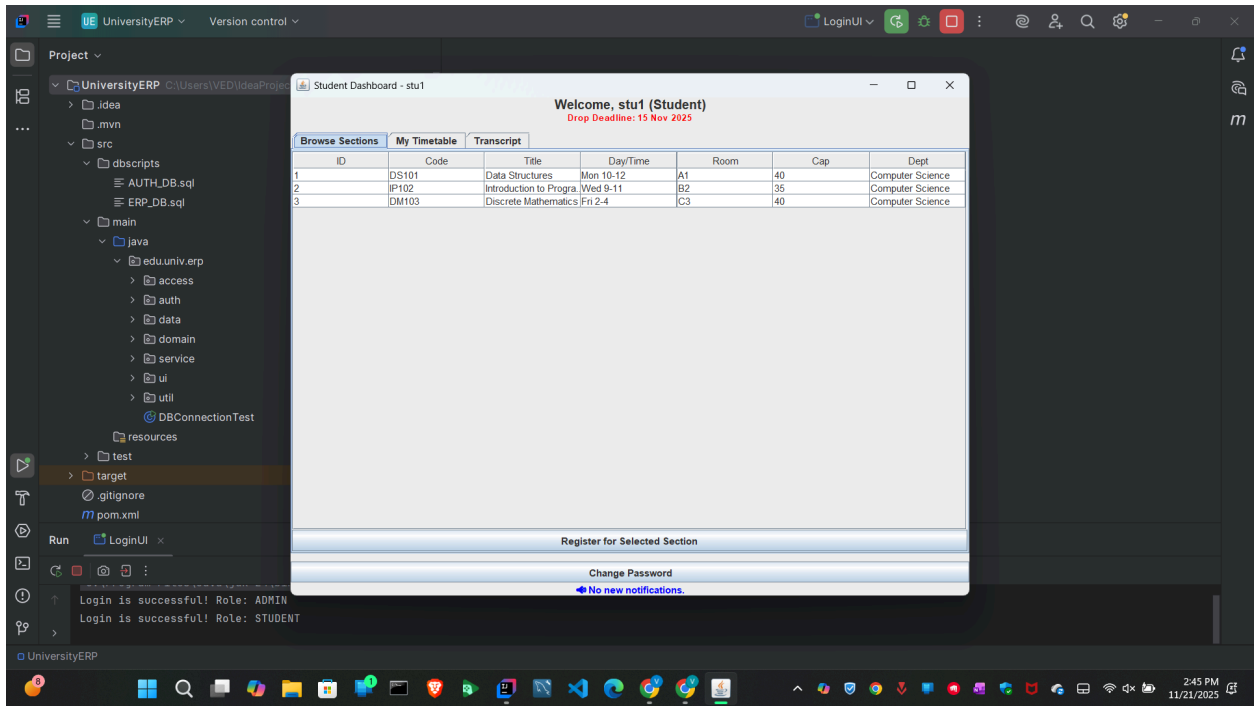
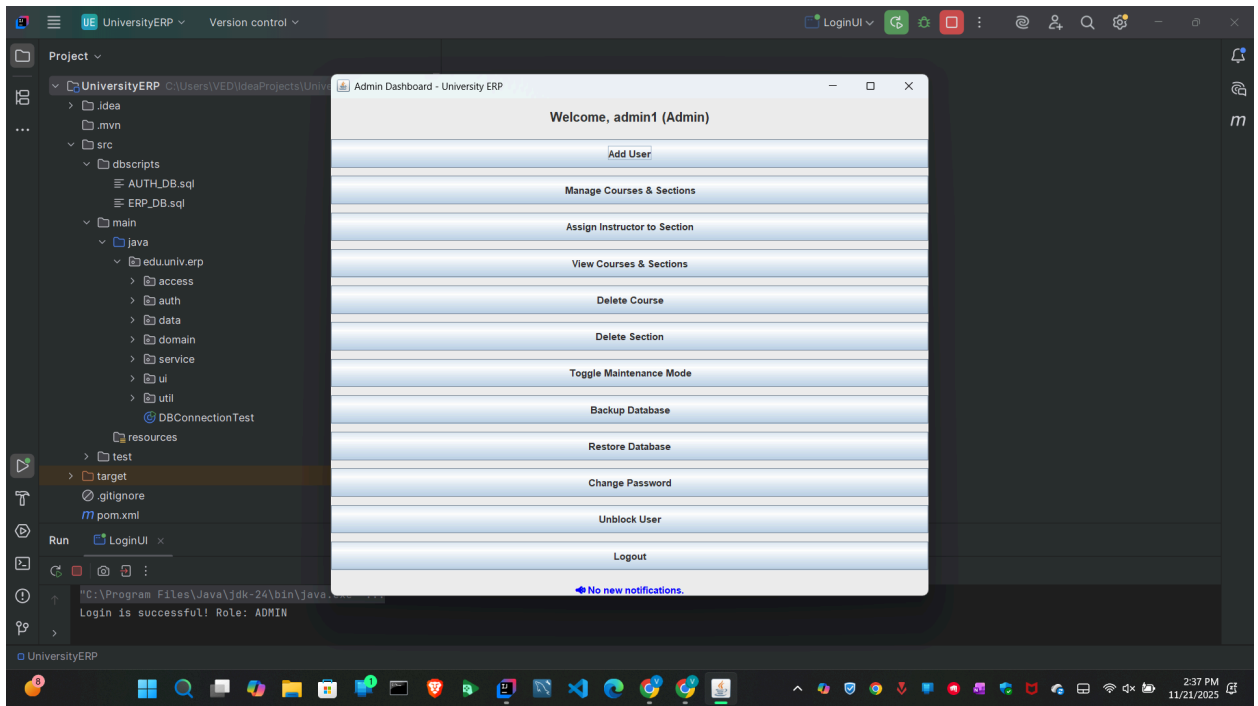
- a. Account Lockout after 5 failures
- b. Password Change for all roles
- c. Maintenance Mode read-only enforcement
- d. "Please Wait" dialogs for long operations
- e. CSV/PDF Export (transcripts & grades)
- f. CSV Import for grades
- g. Sorting in all tables (setAutoCreateRowSorter(true))
- h. Clean UI with FlatLaf
- i. User messages
- j. Deadline warning shown in Student Dashboard

## **8. Observations and Results**

- All major acceptance tests passed
- UI is responsive and clean
- No crashes or data corruption
- Data is consistent across dual-DB architecture
- Security features (bcrypt, lockout, maintenance mode) function correctly

## 9. Screenshots





UniversityERP Version control LoginUI

Project

- UniversityERP C:\Users\VED\IdeaProject
  - idea
  - mvn
  - src
    - dbscripts
      - AUTH\_DB.sql
      - ERP\_DB.sql
    - main
      - java
        - edu.univ.erp
          - access
          - auth
          - data
          - domain
          - service
          - ui
          - util
        - DBConnectionTest
      - resources
      - test
      - target
      - gitignore
      - pom.xml

Run LoginUI

Login is successful! Role: ADMIN  
Login is successful! Role: STUDENT

Student Dashboard - stu1

Welcome, stu1 (Student)  
Drop Deadline: 15 Nov 2025

Browse Sections My Timetable Transcript

EnrollID	Code	Title	Day/Time	Room
1	DS101	Data Structures	Mon 10-12	A1

Drop Selected Section

Change Password  
No new notifications.

UniversityERP Version control LoginUI

Project

- UniversityERP C:\Users\VED\IdeaProject
  - idea
  - mvn
  - src
    - dbscripts
      - AUTH\_DB.sql
      - ERP\_DB.sql
    - main
      - java
        - edu.univ.erp
          - access
          - auth
          - data
          - domain
          - service
          - ui
          - util
        - DBConnectionTest
      - resources
      - test
      - target
      - gitignore
      - pom.xml

Run LoginUI

Login is successful! Role: ADMIN  
Login is successful! Role: STUDENT

Student Dashboard - stu1

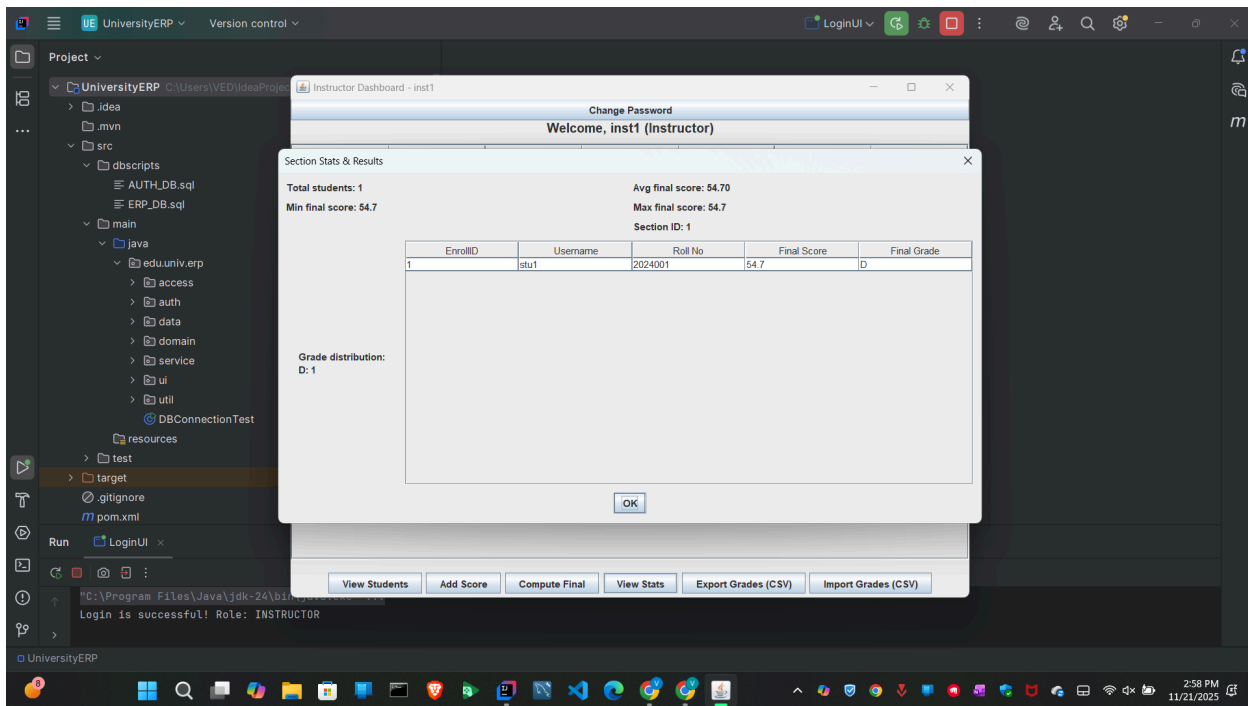
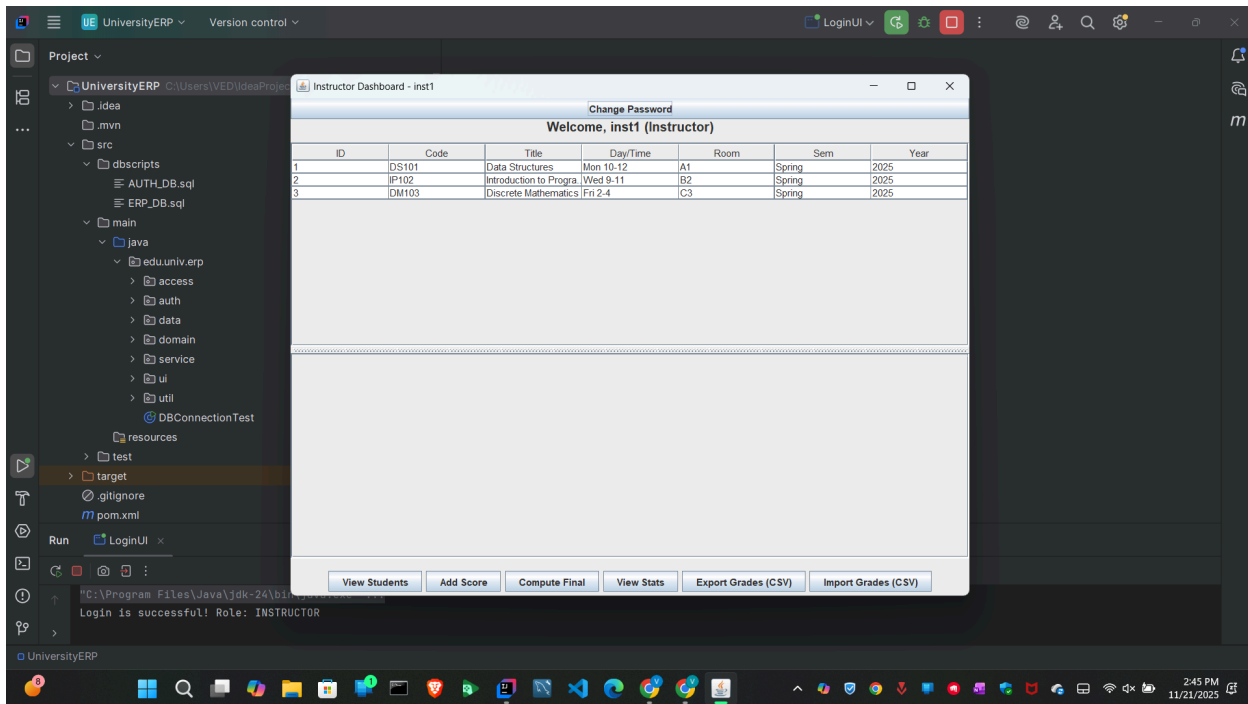
Welcome, stu1 (Student)  
Drop Deadline: 15 Nov 2025

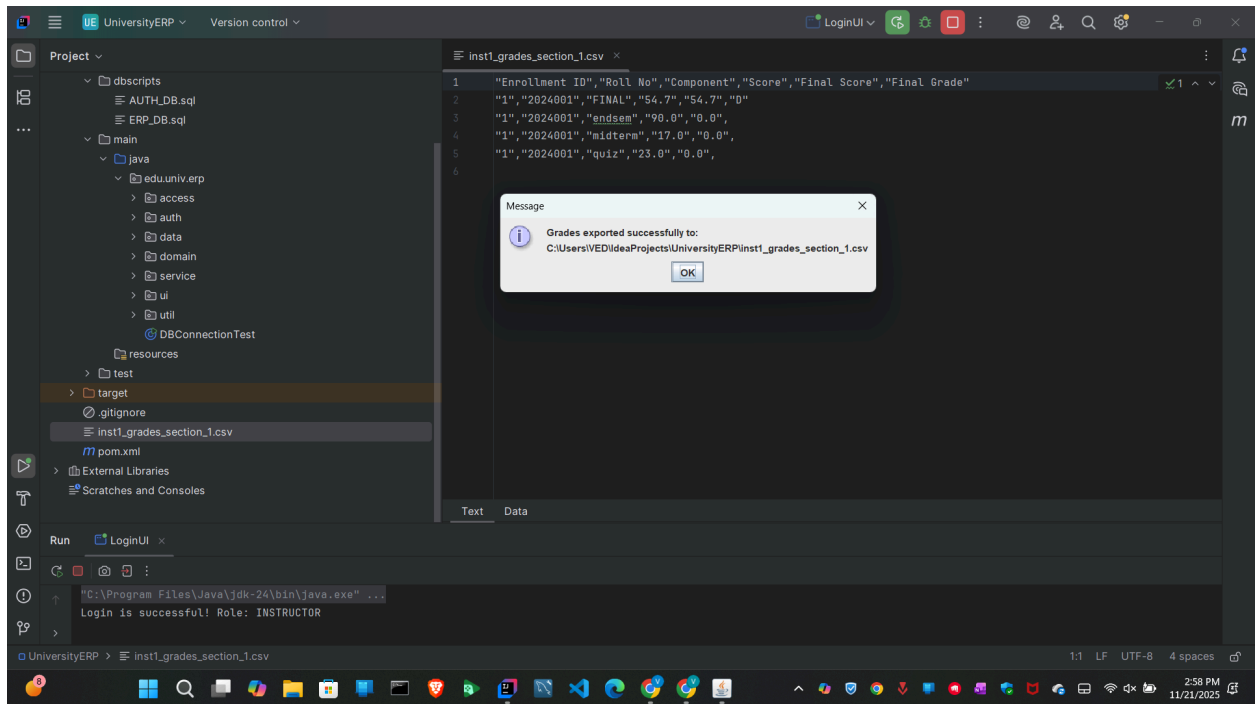
Browse Sections My Timetable Transcript

Course Code	Course Title	Final Score	Final Grade
-------------	--------------	-------------	-------------

Export as CSV Export as PDF

Change Password  
No new notifications.





## 10. Conclusion

The University ERP System fully satisfies all functional and technical requirements of the project. It demonstrates reliable authentication mechanisms, logical role separation, secure data handling, complete academic workflow support, and polished UI/UX.

---