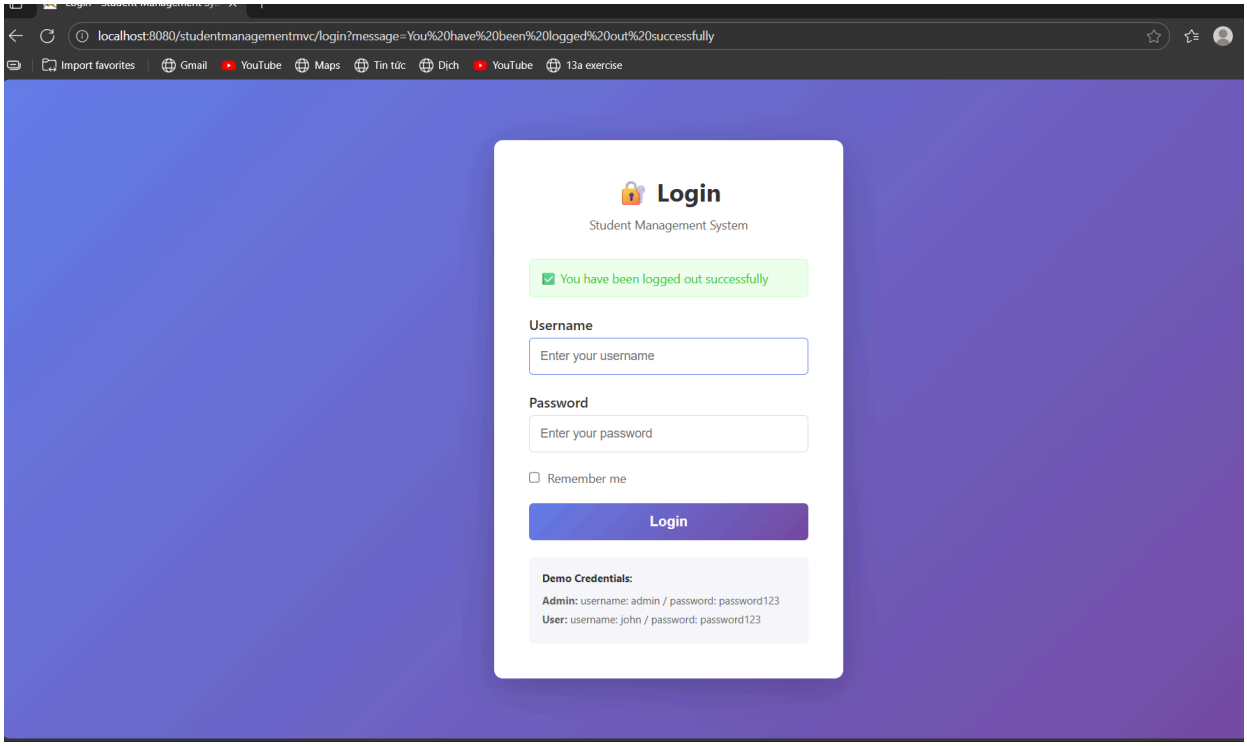
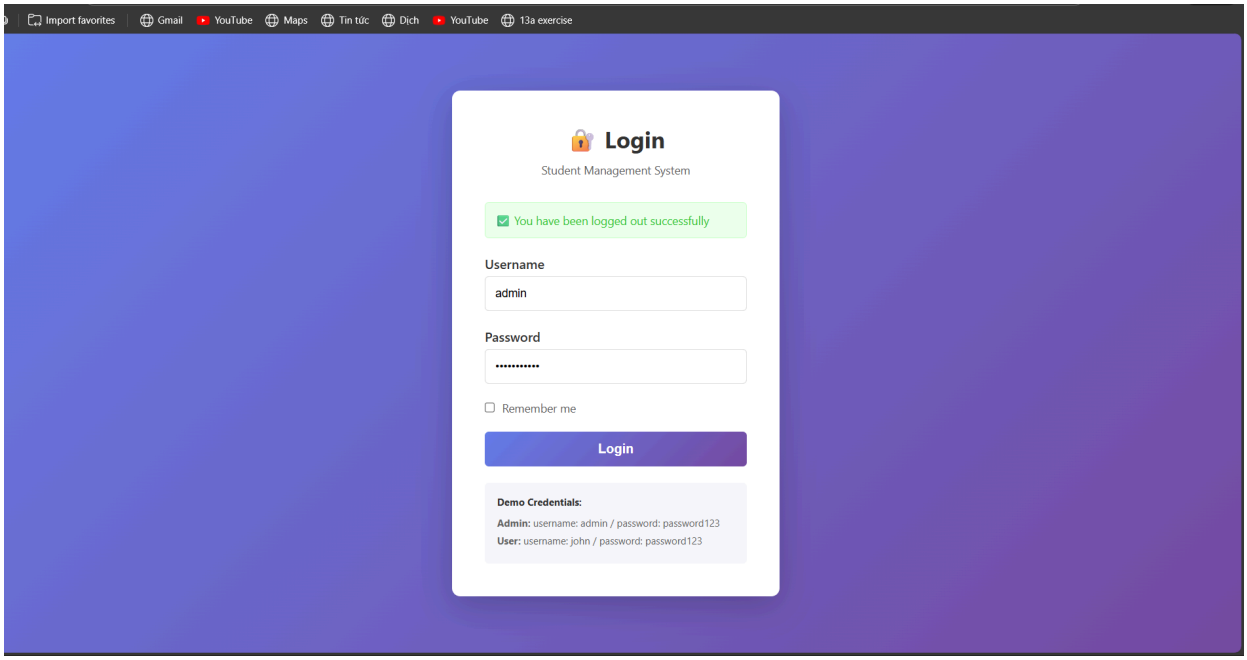
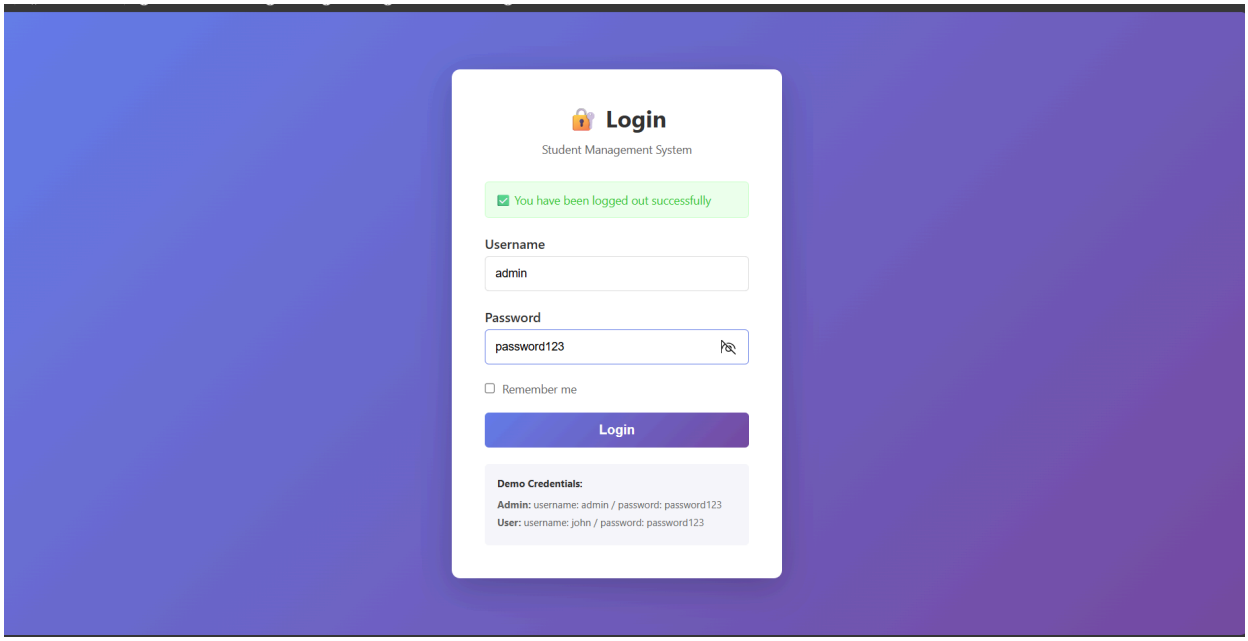
Student Name: Vũ Đình Đức
Student ID: ITCSIU23004
Course: Web-application Development
Semester: I 2025-2026
Instructor: Nguyễn Văn Sinh

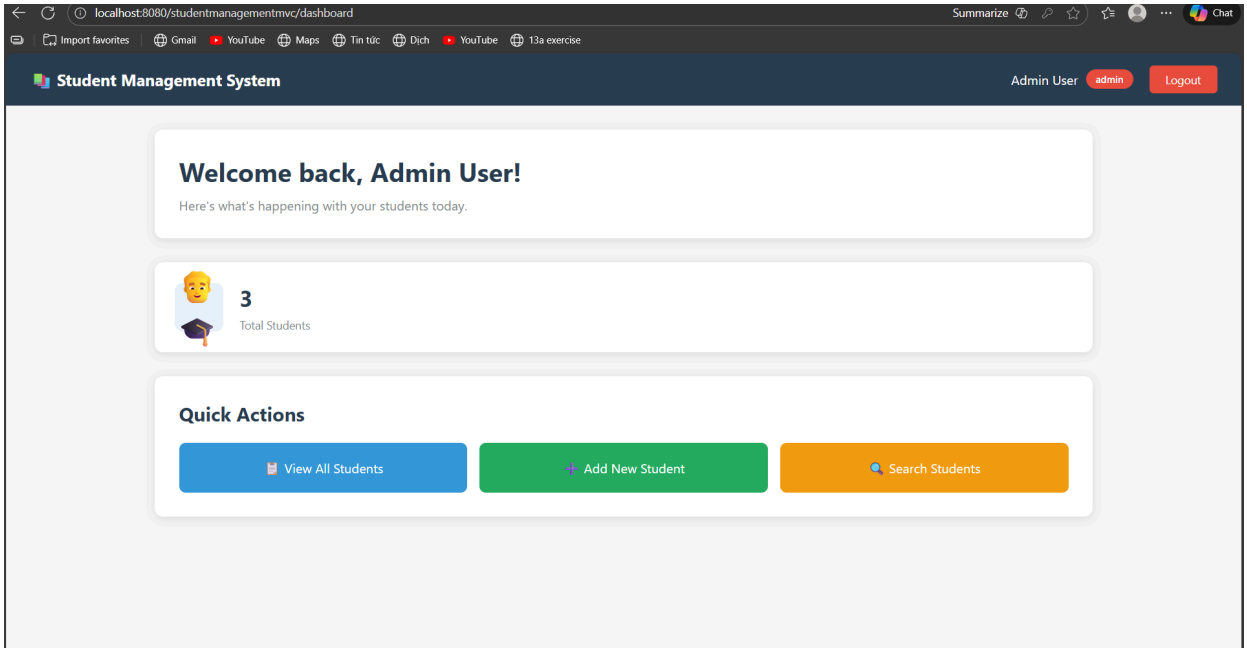
# Lab 06

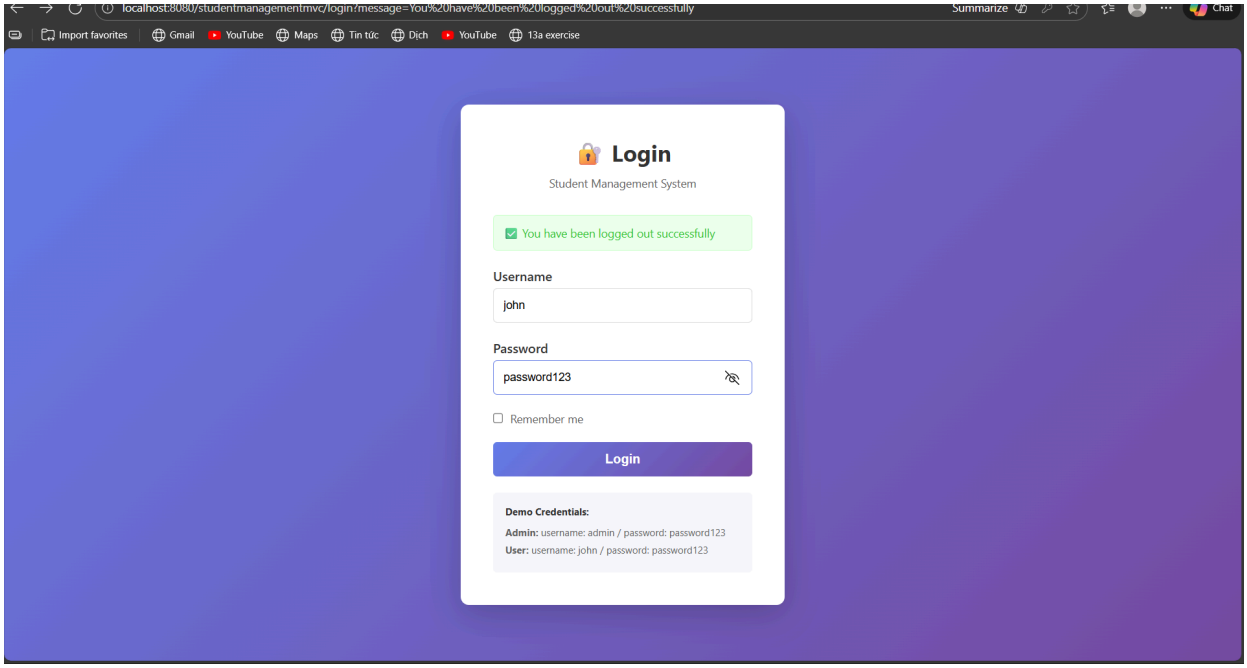
Part A: Running the demo

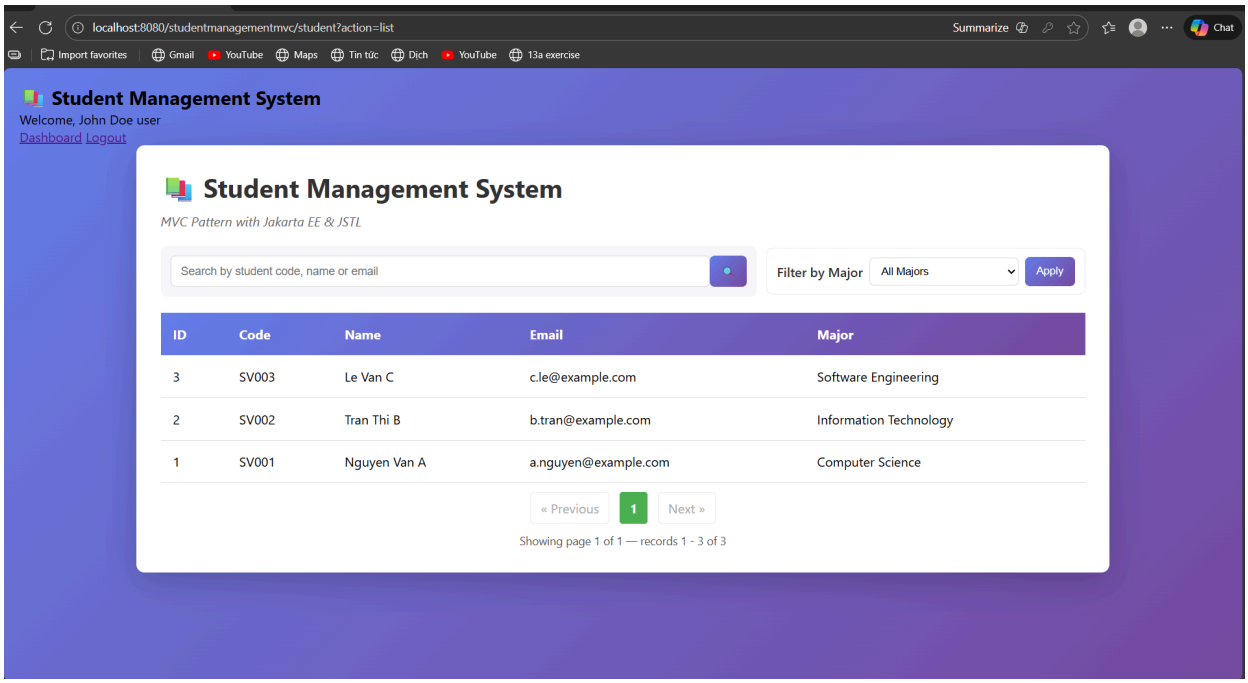

localhost:8080/studentmanagementmvc/login?message=You%20have%20been%20logged%20out%20successfully
Import favorites
Gmail
YouTube
Maps
Tin tức
Dịch
YouTube
13a exercise
Login
Student Management System
You have been logged out successfully
Username
Enter your username
Password
Enter your password
Remember me
Login
Demo Credentials:
Admin: username: admin / password: password123
User: username: john / password: password123


Import favorites
Gmail
YouTube
Maps
Tin tức
Dịch
YouTube
13a exercise
Login
Student Management System
You have been logged out successfully
Username
admin
Password
Remember me
Login
Demo Credentials:
Admin: username: admin / password: password123
User: username: john / password: password123


Login
Student Management System
You have been logged out successfully
Username
admin
Password
password123
Remember me
Login
Demo Credentials:
Admin: username: admin / password: password123
User: username: john / password: password123


localhost:8080/studentmanagementmvc/dashboard
Summarize
Chat
Import favorites
Gmail
YouTube
Maps
Tin tức
Dịch
YouTube
13a exercise
Student Management System
Admin User
admin
Logout
Welcome back, Admin User!
Here's what's happening with your students today.
3
Total Students
Quick Actions
View All Students
Add New Student
Search Students




1. GET /login -> LoginController.doGet()

   +If a session with user exists, redirect to dashboard.
   +Otherwise forward to login.jsp.

2. POST /login (form submit) -> LoginController.doPost()

+Read username, password, validate inputs.

+Call UserDAO.authenticate(username, password).

+If authentication succeeds:

->Invalidate old session (prevents session fixation).

->Create a new session, save user, role, fullName.

->Set session timeout = 30 minutes.

->(TODO) handle "remember me" via cookie.

->Redirect: admin → dashboard; else → student?action=list.

+If authentication fails: set error and forward back to login.jsp (keep username).

3. Access dashboard -> DashboardController.doGet()

   +Confirm user is logged in via session, get User from session.

   +Fetch stats via StudentDAO.getTotalStudents(), set totalStudents and welcomeMessage, forward to dashboard.jsp.

4. Logout -> LogoutController invalidates session and redirects to /login with a message.

**-AuthFilter (/*)**: allows public URLs (login, logout, static assets). Other requests require a logged-in session; otherwise redirect to /login.

**-AdminFilter (/student)**: checks action parameter. Actions in ADMIN_ACTIONS (new, insert, edit, update, delete) require user.isAdmin(); otherwise redirect to the student list with an access-denied message.

-Session handling is good: create new session after login and set a reasonable timeout. remember me is not yet implemented.

-login.jsp: login form, shows error or param.message, preserves username after failed login; uses JSTL.
-dashboard.jsp: shows ${welcomeMessage}, ${totalStudents}, and conditionally shows "Add New Student" only when sessionScope.role == 'admin'.

# UserDAO.java:

DAO class that talks to MySQL for users. I put DB URL/user/password as constants at the top.
**Connection:** getConnection() loads the MySQL driver and returns a JDBC Connection.
**SQL:** Reusable SQL strings for authenticate, update last_login, get by id/username, and insert.
**authenticate(username, password):**

- Selects the active user by username.
- If found, verifies the given password against the stored hash with BCrypt.checkpw.
- On success it maps the row to a User object and calls updateLastLogin(id). Returns User or null.

**updateLastLogin(userId):** runs UPDATE users SET last_login = NOW() for that id.
**getUserById / getUserByUsername:** simple selects that map result set to User.
**createUser(User):** hashes the plain password with BCrypt.hashpw(...) then inserts the user row.
**mapResultSetToUser:** helper to build a User from a ResultSet (id, username, password, fullName, role, isActive, timestamps).

Link Github: https://github.com/ducvu01/web_lab_06.git