Web Application Development Lab 6 EXERCISES

AUTHENTICATION & SESSION MANAGEMENT

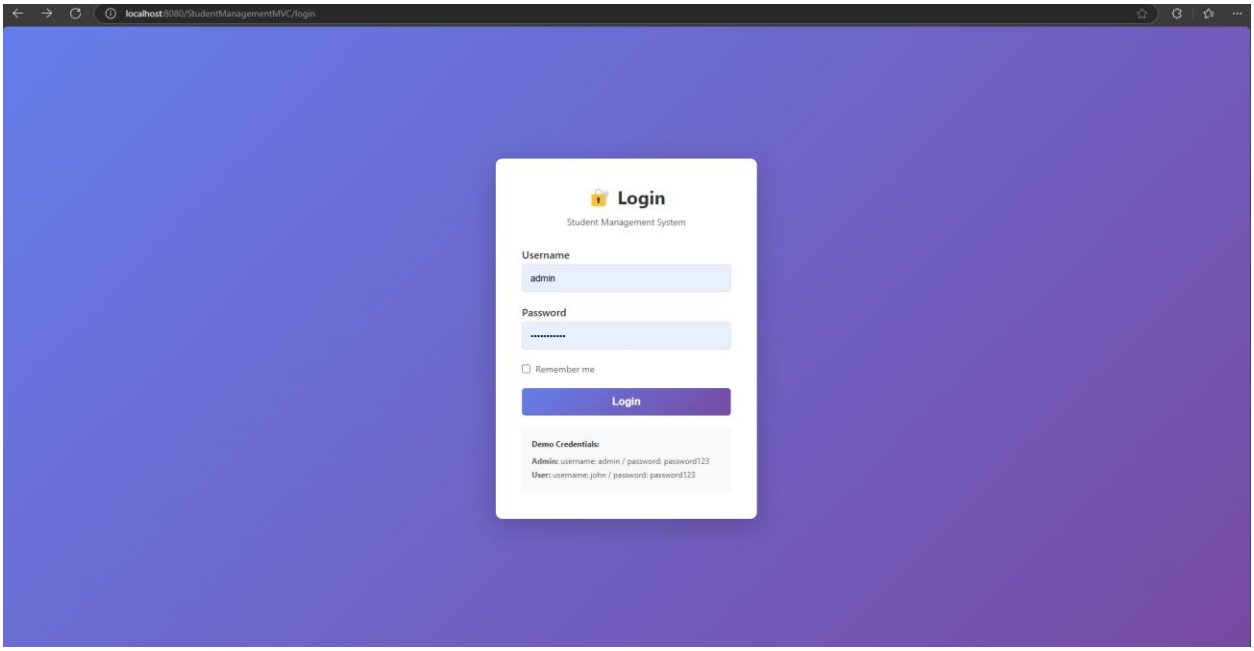
Name: Dang Thanh Lam – ITITDK23039

PART A: IN-CLASS EXERCISES

Test log-in flow

1. Access: http://localhost:8080/YourApp/

→ Redirected to login (AuthFilter)



The request is intercepted by the AuthFilter, which checks if a user session exists.

Since no session exists yet, the filter redirects the user to login.jsp.

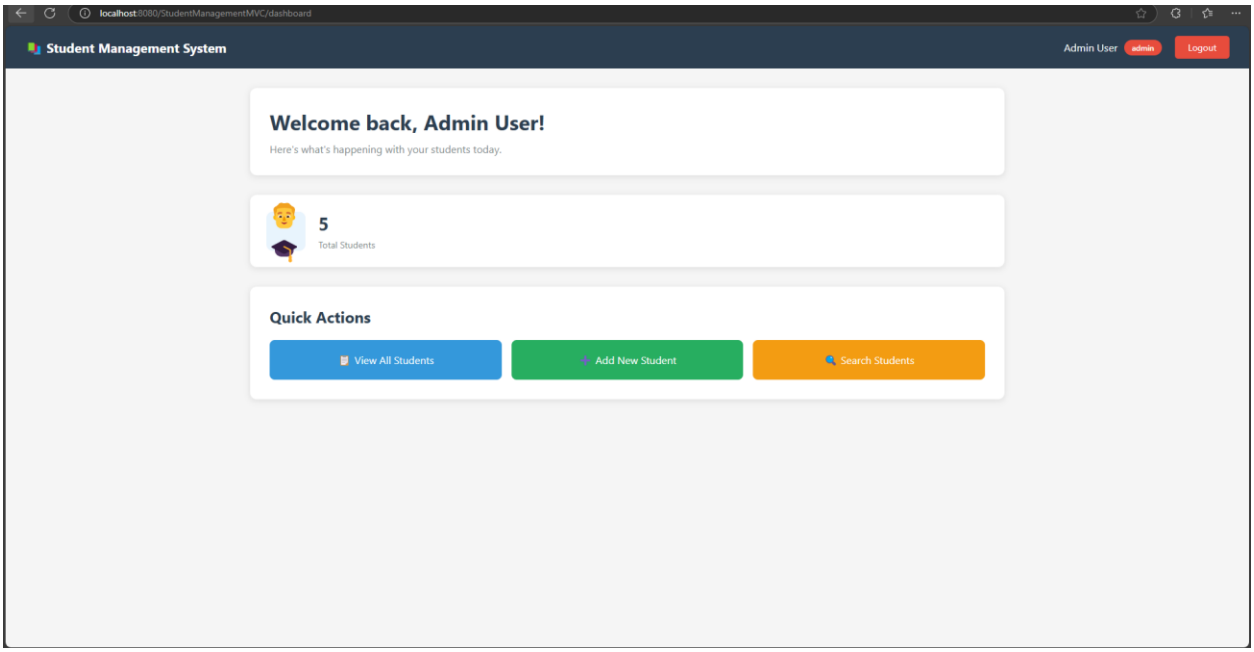
This ensures that unauthorized users cannot access any protected resources such as the dashboard or student management pages.

The redirection confirms that your authentication flow is enforced at the entry point, which is essential for application security.

2. Login with: admin / password123

→ Creates session

→ Redirected to dashboard



Result & Explanation:

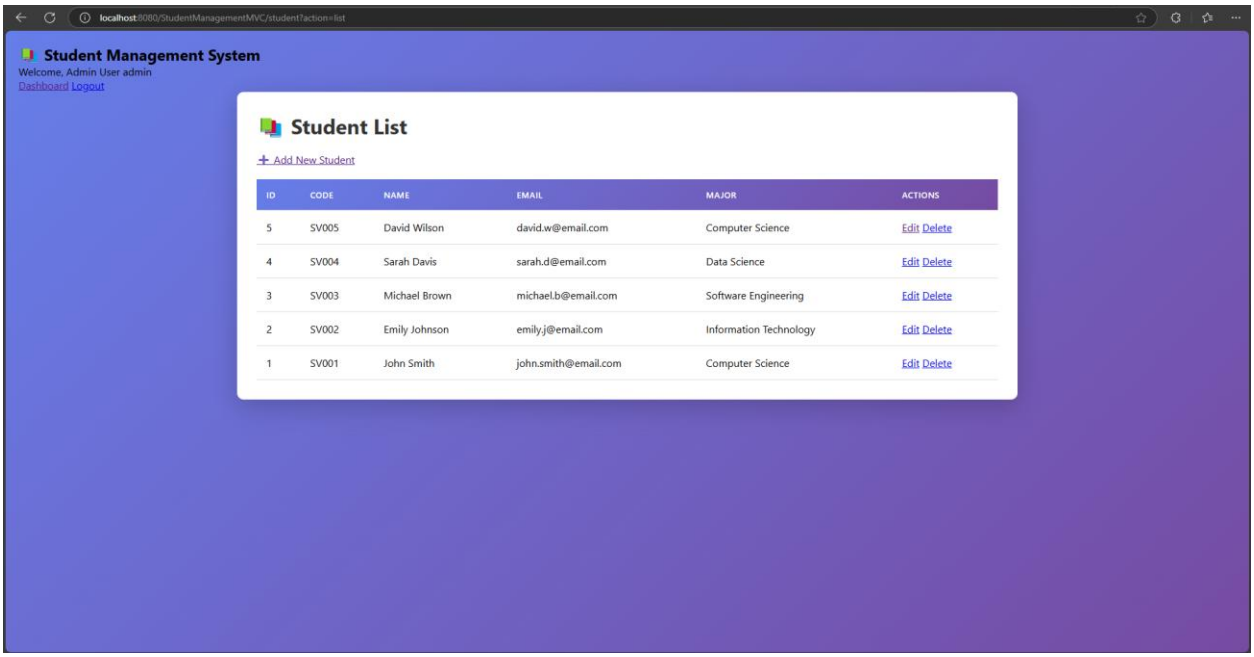
- The LoginController receives the POST request, retrieves the user record from the database, and verifies the password using BCrypt.
- Because the credentials match, a new session is created, and the user object is stored in the session.
- The user is redirected to the dashboard, and admin privileges are enabled.
- This demonstrates both correct authentication and the proper functioning of session management.

- Security best practices such as session invalidation and role-based redirection are applied, preventing session fixation attacks.

3. Click "View All Students"

→ Shows student list

→ Edit/Delete buttons visible (admin)



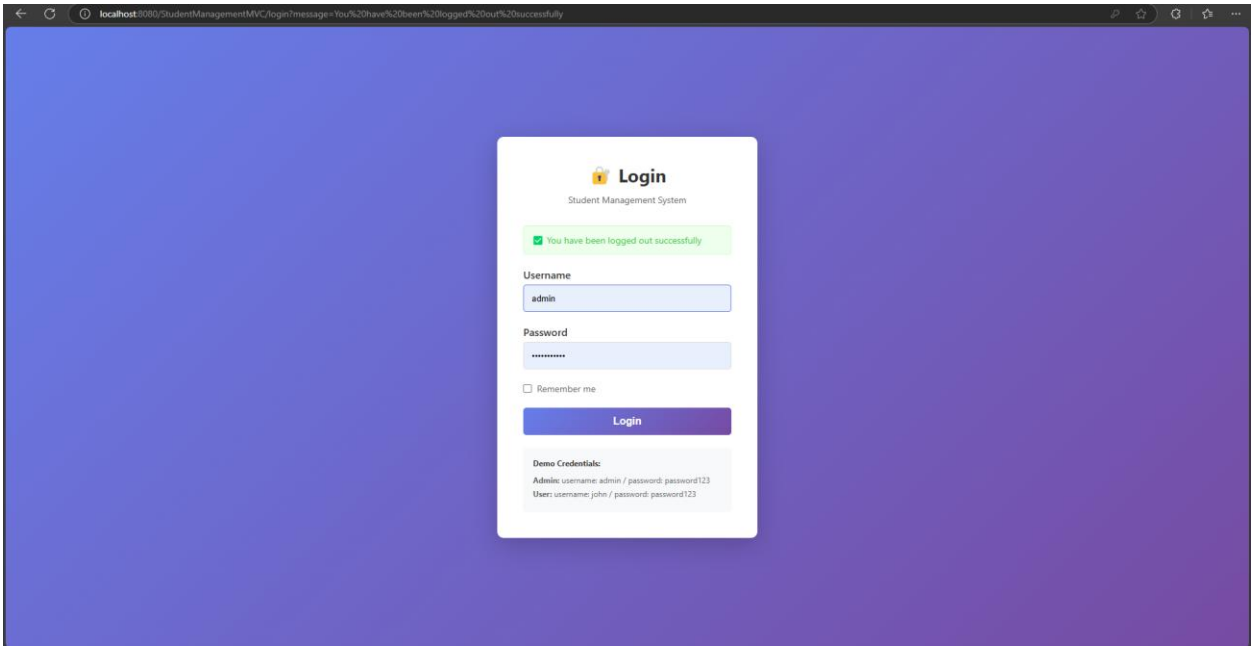
Result & Explanation:

- The request triggers StudentController with the default action list.
- All student records are retrieved using StudentDAO, and attributes are forwarded to student-list.jsp.
- The JSP renders the data and displays admin-specific controls like Edit and Delete buttons.
- This step confirms that the application correctly distinguishes between roles and applies UI-level access control, ensuring admins can perform CRUD operations while regular users cannot.

4. Logout

→ Session invalidated

→ Redirected to login



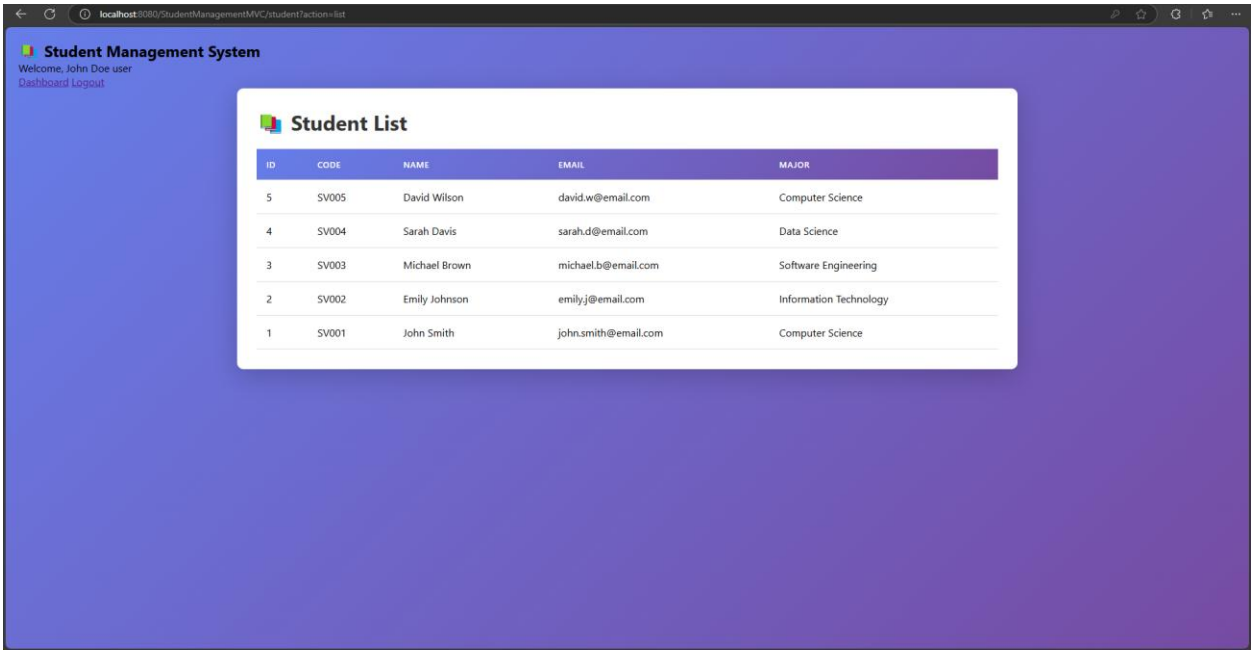
Result & Explanation:

- LogoutController invalidates the current session.
- All session attributes, including the user object, are cleared, preventing further access to protected resources.
- The user is redirected to login.jsp with a success message.
- This confirms that session invalidation works correctly, and that the application properly communicates logout feedback to the user.
- It also proves that any subsequent request without logging in is blocked by AuthFilter.

5. Login with: john / password123

→ Regular user view

→ No Edit/Delete buttons



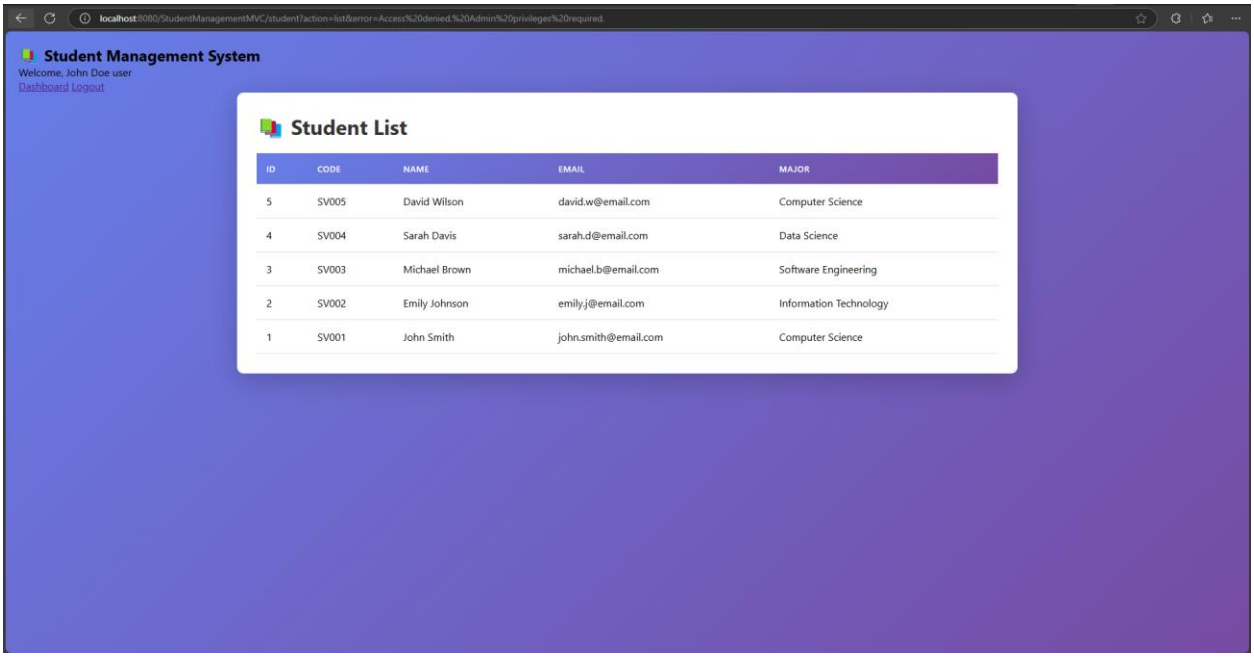
Result & Explanation:

- The login process is identical to Step 2, but the retrieved user has role user.
- Session is created with regular user privileges, which are enforced both in the dashboard and the student list.
- Edit and Delete buttons are hidden, reflecting role-based access control at the view level.
- This demonstrates that the application correctly differentiates user roles and enforces permissions, preventing unauthorized actions by non-admin users.

6. Try to access: /student?action=new

→ AdminFilter blocks access

→ Error message displayed



Result & Explanation:

- AdminFilter intercepts the request and detects that the logged-in user does not have admin privileges.
- The request is blocked, and an error message is displayed.
- This step validates server-side access control, which is critical because hiding buttons in the UI alone is insufficient for security.
- By combining session verification and role checks, the application prevents privilege escalation.