

VIETNAM NATIONAL UNIVERSITY OF HO CHI MINH CITY
THE INTERNATIONAL UNIVERSITY
SCHOOL OF COMPUTER SCIENCE AND ENGINEERING



Web Application Development - IT093IU

Homework Lab 6: AUTHENTICATION & SESSION MANAGEMENT

Nguyễn Vũ Thành Tính - ITCSIU23039

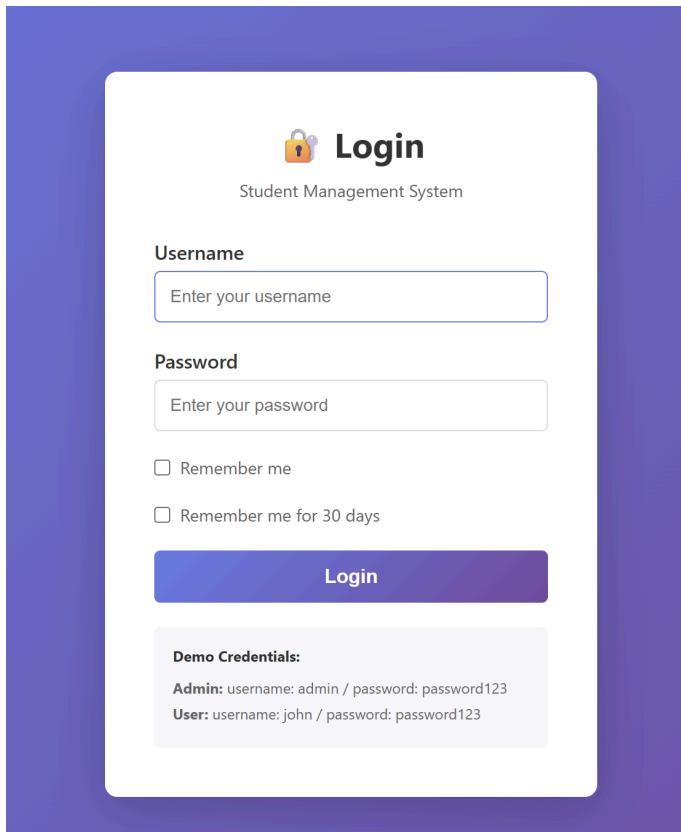
Lab Instructor: N.T.Nghia

Github Links:	3
1. Login page.....	3
2. Login with wrong credentials (error message).....	4
3. Dashboard after login.....	5
4. Student list as admin (with Edit/Delete buttons).....	6
5. Student list as regular user (without Edit/Delete).....	7
6. Accessing protected page without login (redirect).....	8
7. Non-admin trying admin action (blocked).....	9
8. Logout success message.....	10

Github Links:

https://github.com/ngvuthinh/Web_Lab/tree/main/Lab_6

1. Login page



Explanation

- User accesses: `http://localhost:8080/StudentManagementMVC/`.
- The request passes through `AuthFilter` (file: `filter/AuthFilter.java`).
- `AuthFilter` checks whether a session exists: If session is null, user is not logged in.
- `AuthFilter` redirects the user to "login".
- The redirected request is handled by `LoginController.doGet()` (file: `controller/LoginController.java`).
- `LoginController` forwards the request to: `views/login.jsp`

2. Login with wrong credentials (error message)

The screenshot shows a login form titled "Login" for a "Student Management System". The form includes fields for "Username" (containing "admin") and "Password" (containing "Enter your password"). A red error message box at the top states "Invalid username or password". Below the form are two "Remember me" checkboxes and a "Login" button. At the bottom, there is a "Demo Credentials:" section listing "Admin: username: admin / password: password123" and "User: username: john / password: password123".

Explanation:

- User submits login form in login.jsp with action="login" and method="post".
- Request is processed by LoginController.doPost().
- Controller calls UserDAO.authenticate(username, password).
- UserDAO validates the hashed password using BCrypt.checkpw().
- Validation fails (returns null), so LoginController sets an error message attribute.
- Controller forwards back to views/login.jsp to display the error.

3. Dashboard after login

The screenshot shows the Student Management System dashboard. At the top, there's a header bar with the system name, user information (Admin User, admin), password input, and a logout button. Below the header is a main content area. It starts with a welcome message: "Welcome back, Admin User! Here's what's happening with your students today." There are two summary cards: one for "Total Students" (20) and one for "My Visits" (1). Below these are four "Quick Actions" buttons: "View All Students" (blue), "Add New Student" (green), "Search Students" (orange), and "Change Password" (purple).

Explanation:

- UserDAO returns a valid User object upon successful authentication.
- LoginController creates a new session: session.setAttribute("user", user).
- Controller redirects to /dashboard.
- Request reaches DashboardController.doGet() (file: controller/DashboardController.java).
- DashboardController forwards to views/dashboard.jsp to render statistics and user info.

4. Student list as admin (with Edit/Delete buttons)

The screenshot shows a web application titled "Student Management System". At the top, it says "Welcome, Admin User admin" and provides links for "Dashboard" and "Logout". Below this is a section titled "Student List" with a "Add New Student" button. A table displays five student records:

ID	CODE	NAME	EMAIL	MAJOR	ACTIONS
1	SV010	Hoang Thuy Linh	linh.hoang@example.com	Computer Science	Edit Delete
2	SV011	Dang Van Lam	lam.dang@example.com	Information Technology	Edit Delete
3	SV012	Doan Van Hau	hau.doan@example.com	Software Engineering	Edit Delete
4	SV013	Nguyen Quang Hai	hai.nguyen@example.com	Business Administration	Edit Delete
5	SV014	Phan Van Duc	duc.phan@example.com	Computer Science	Edit Delete

Explanation:

- Admin clicks "View All Students", sending a request to: /student?action=list.
- Request passes through AuthFilter (allowed due to active session).
- Request is handled by StudentController.doGet() (file: controller/StudentController.java).
- Controller calls studentDAO.getAllStudents() and attaches the list to the request.
- Controller forwards to views/student-list.jsp.
- JSP checks sessionScope.role: Since role is "admin", Edit/Delete buttons are displayed.

5. Student list as regular user (without Edit/Delete)

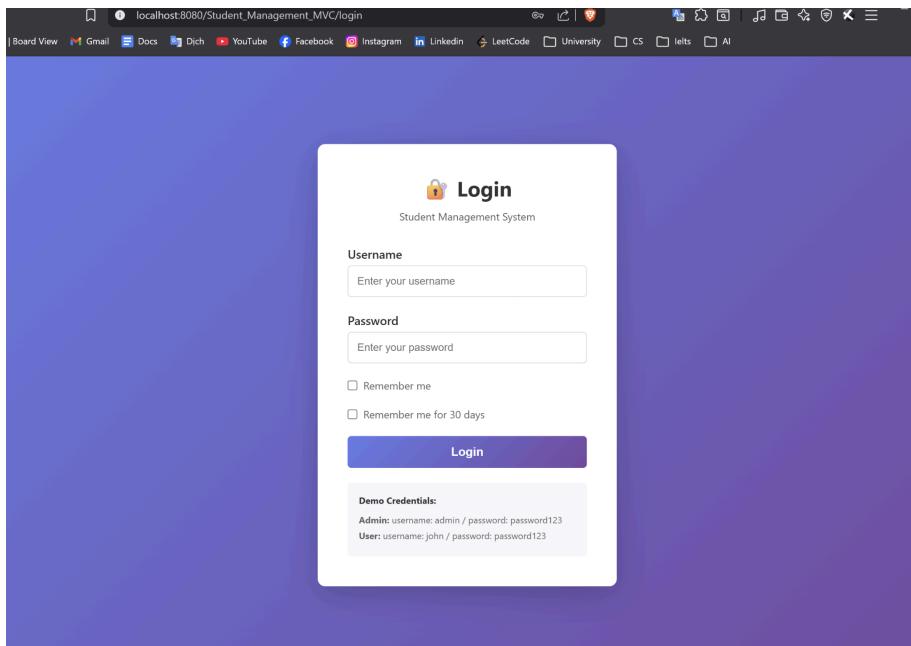
The screenshot shows a web application titled "Student Management System". At the top, it says "Welcome, John Doe user" and provides links for "Dashboard" and "Logout". Below this, the title "Student List" is displayed above a table. The table has columns for ID, CODE, NAME, EMAIL, and MAJOR. It contains five rows of student data:

ID	CODE	NAME	EMAIL	MAJOR
1	SV010	Hoang Thuy Linh	linh.hoang@example.com	Computer Science
2	SV011	Dang Van Lam	lam.dang@example.com	Information Technology
3	SV012	Doan Van Hau	hau.doan@example.com	Software Engineering
4	SV013	Nguyen Quang Hai	hai.nguyen@example.com	Business Administration
5	SV014	Phan Van Duc	duc.phan@example.com	Computer Science

Explanation:

- Regular user (role "user") accesses /student?action=list.
- StudentController retrieves student data and forwards to views/student-list.jsp.
- JSP checks sessionScope.role: Since role is "user", the check fails.
- The "Add New Student" button and "Actions" column (Edit/Delete) are hidden from the view.

6. Accessing protected page without login (redirect)



Explanation:

- User attempts to access protected URL: /dashboard.
- Request passes through AuthFilter.
- AuthFilter checks for session.getAttribute("user").
- Since no user is in session, AuthFilter blocks the request.
- AuthFilter redirects the user to /login.

7. Non-admin trying admin action (blocked)

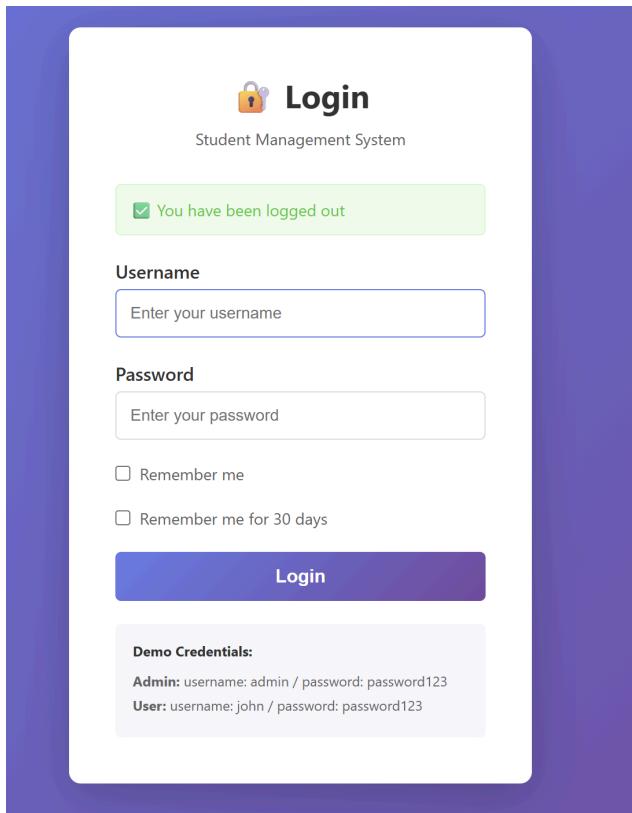
The screenshot shows a web application titled "Student Management System". The URL in the address bar is "localhost:8080/Student_Management_MVC/student?action=list&error=Access%20denied". The page header includes a welcome message "Welcome, John Doe user" and links to "Dashboard" and "Logout". A prominent red error message box displays the text "Access denied. Admin privileges required.". Below this, a table titled "Student List" lists five student entries:

ID	CODE	NAME	EMAIL	MAJOR
1	SV010	Hoang Thuy Linh	linh.hoang@example.com	Computer Science
2	SV011	Dang Van Lam	lam.dang@example.com	Information Technology
3	SV012	Doan Van Hau	hau.doan@example.com	Software Engineering
4	SV013	Nguyen Quang Hai	hai.nguyen@example.com	Business Administration
5	SV014	Phan Van Duc	duc.phan@example.com	Computer Science

Explanation:

- Regular user manually enters URL: /student?action=new.
- Request passes AuthFilter (session exists) but reaches AdminFilter (file: filter/AdminFilter.java).
- AdminFilter checks the user role: if (!user.isAdmin()).
- Since the role is "user", AdminFilter blocks the request.
- The filter redirects the user to the list page with an error parameter: error=Access denied

8. Logout success message



Explanation:

- User clicks logout, sending request to: /logout.
- Request hits LogoutController.doGet() (file: controller/LogoutController.java).
- Controller invalidates the session using session.invalidate().
- User is redirected back to /login with a success message.