# CPRG-352 - Web Application Programming

## Lab 5

## Topics: Sessions and Authentication

In this lab, we will create an application with login and logout functionality.

Surprising, even for a basic application, there are a lot of moving parts! Make sure you have all of the pieces working correctly. Stay focused!

Create a web application called **MyLogin**. Add two servlets to this application.

1. **LoginServlet**: Handles all authentication and creation/destruction of sessions.

doGet() displays a login form (login.jsp) to the user that can also show messages.

doGet() is also responsible for logging out the user. If the parameter “logout” exists, invalidate the session and display a message that the user has successfully logged out.

doPost() processes the submission of the form from login.jsp. doPost() first validates that user name and password are not empty. Then, it passes the user name and password parameters to the login() method of a service class called AccountService. If login() returns a non-null value, store the username in a session variable and **redirect** (not forward) the user to the home url. If the authentication parameters are invalid, display an appropriate error message, keeping the textboxes filled in with what the user had previously entered and forward the user to login.jsp.

The URL for LoginServlet is /login

1. **HomeServlet**: shows a welcome message (home.jsp) to the user including their username. The JSP home.jsp also shows a Logout hyperlink which makes a get request to take the user back to URL login and displays the message “You have successfully logged out.”

The url of HomeServlet is /home

The AccountService class is very simple. It has only one (non-static) method:

public User login(String username, String password)

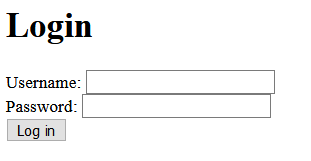
which validates the user and password. If the login is successful, return a user object, with the password set as null. If the login is not successful, return null.

There are two user names should validate successfully: **adam** and **betty**. Both users have a password of **password**. Any other login data should not be accepted. You can hard code adam and betty in the AccountService class.

You must gracefully handle all unexpected errors appropriately. E.g. 404 file not found, or thrown exceptions.

The welcome page should be set to /home. If an authenticated user returns to our website, we want to give them the /home page not /login. If the user attempts to access /login and the session object *username* exists, redirect them to /home. If the user attempts to access /home directly and the session object *username* does not exist, redirect them to /login.

Example 1: /login initial view (when application is run)



Example 2: /home when adam has logged in successfully



## System Tests

Your application must pass the following tests:

1. Login

Test: Attempt log in with incorrect credentials (i.e. user/pass)

Result: Failed authentication message

Test: Log in with correct credentials (i.e. adam/password)

Result: Home page displayed

1. Logout

Test: Log in with correct credentials, then click logout link

Result: Login page is displayed with “log out” message

1. Logged in redirection

Test: Log in with correct credentials, then change URL to /login

Result: Web page URL changed to /home and home page is displayed

1. Logged out redirection

Test: Start from login screen, then change URL to /home

Result: Web page URL changed to /login and login page is displayed

1. Session created upon login

Test: Log in with correct credentials, then open new tab and enter URL with /home

Result: Home page is displayed

1. Session destroyed upon logout

Test: Log in with correct credentials, then log out, then open a new tab and enter URL with /login

Result: Login page is displayed

1. Default page is /home

Test: Log in with correct credentials, then change URL to / (root web URL)

Result: Home page is displayed