2. Login App

Introduction

You have to build an "Web Application" which contains two pages. One is the "Login" Page and Other is "Landing" Page.

Requirements/Features

a. Login Page
   - Email Input (with format validation)
   - Password Input
   - Submit Button

b. Landing Page
   - Success/Error based on authentication

Dependency/Tools used

a. Language: Java
b. Framework/Interface: Servlet, JSP
c. Frontend: HTML
d. Pattern: MVC
e. IDE: Eclipse

Implementation

In this section, I will describe the whole implementation of this login app.

I used two servlets: one for login (`LoginServlet`) and one for logout (`LogoutServlet`).

The application starts with the login page (`login.html`) that takes input of the email and passport from the user with a form and submits the information with a button. I have used the HTML "email" type of the input so that it automatically checks the pattern of the given email address and alerts the user incase of wrong format.

After clicking the submit button, the application will go to the `LoginServlet` where the input email and passwords are checked for authentication. In the ideal case, it should be checked with the user-data fetched from the database. But to complete the assignment in a timely fashion and since it was not mandatory for this assignment, I did not use the database for now, and validated the email and password by comparing them with hardcoded data. In my code, I have hardcoded the email as "`tanvir@gmail.com`" and "`1234`".

If the input email and password match with the correct (hardcoded) ones, the user is validated. In that case, I created a session (`HttpSession`) for the user and redirects him/her to the success page (`success.jsp`). In the success page, there is a logout button which takes the user to the `LogoutServlet` where the session is destroyed and the user is redirected to the main login page (`login.html`).

Otherwise, if the input email and password do not match, the user is not validated. So we send him/her to the error page (`failure.jsp`).

To prevent logging in just by hitting the success link, we used the concept of session and checked the session attributes in the success page (`success.jsp`). If the session attribute is null there, it means it has not passed the authentication in the `LoginServlet,` so we do not allow him in the success page.

Database Design

Though I have not incorporated database stuff for now, I would like to discuss the possible database design for this assignment. Since it is a login app just to validate the user, a simple "USER" table would be enough. The "USER" table might contain the following attributes: "user_id", "user_name", "emai_id", "password".

Coding Practices Followed

a. MVC pattern is followed.
b. Codes are well-documented.