Practice Project: Validation of the User Login

My GitHub link :

https://github.com/SK-NOORALAM-RAHAMAN/Phase-2/tree/master/Practise%20Projects/Validation%20of%20User%20login

Project Objective :

Create a servlet-based web application that shows a login page and validates it. The correct values are hard-coded. On successful login, a dashboard page is shown. The dashboard will provide a link for logging out. Incorrect logins need to be handled by showing an error message page.

Background of the problem statement:

As a part of developing an e-commerce web application, you have to prototype a login scenario for the user. There is no database involved here, so you have to use fixed values for login email id and password.

Development Environment

- Eclipse IDE for Enterprise Java Developers v2019-03 (4.11.0)
- Apache Tomcat Server v9.0
- JRE: OpenJDK Runtime Environment 11.0.2

This lab has ten subsections, namely:

1.1.1 Creating a dynamic web project

1.1.2 Creating a Login.java

1.1.3 Creating a dashboard.jsp

1.1.4 Configuring web.xml

1.1.5 Creating a signup.html

1.1.6 Checking for logout.html

1.1.7 Building the project

1.1.8 Publishing and starting the project

1.1.9 Running the project

1.1.10 Pushing the code to GitHub repositories

Step 1.1.1: Creating a dynamic web project

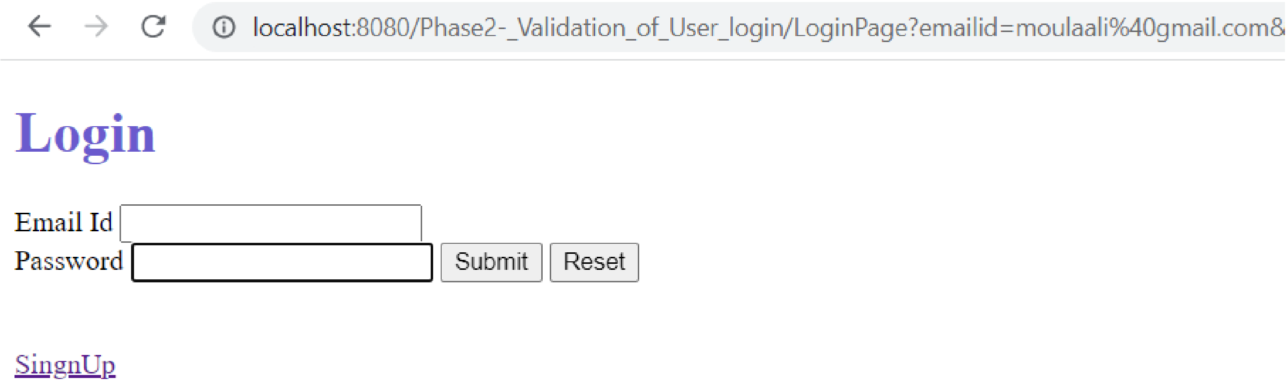
- Open Eclipse
- Go the **File** menu. Choose **New->Dynamic Web Project**
- Enter the project name as ValidationOfUserLogin Click on **Next**
- Enter nothing in the next screen and click on **Next**
- Check the checkbox **Generate web.xml deployment descriptor** and click on **Finish**
- This will create the project files in the Project Explorer

Step 1.1.2: Creating a servlet Login.java

- In the Project Explorer, expand **ServletGetPost->Java Resources**
- Right click **src** and choose **New->Servlet**
- In **Class Name,** enter **Login** and click on **Finish**
- Enter the following code:

```
package com;

import java.io.IOException;
import java.io.ObjectOutput;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 * Servlet implementation class Login
 */
public class LoginPage extends HttpServlet {
      private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public LoginPage() {
        super();
        // TODO Auto-generated constructor stub
    }

      /**
       * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse re-
sponse)
```

```
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

            String emailid = request.getParameter("emailid");
            String password=request.getParameter("password");
            PrintWriter pw=response.getWriter();
            response.setContentType("text/html");

            RequestDispatcher rd1=request.getRequestDispatcher("dashBoard.html");
            RequestDispatcher rd2=request.getRequestDispatcher("index.html");

            try {
                Class.forName("com.mysql.cj.jdbc.Driver");
                Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/db","root","root");
                PreparedStatement pstmt=con.prepareStatement("select * from login where emailid=? and password=?");
                pstmt.setString(1, emailid);
                pstmt.setString(2, password);
                ResultSet rs=pstmt.executeQuery();
                if(rs.next()) {
                    pw.println("Successfully login !");
                    rd1.forward(request, response);
                }
                else{

                    rd2.include(request, response);
                    pw.println("Ivalid login input !");
                }
            }
            catch(Exception e) {
            e.getMessage();
            }
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
                    PrintWriter pw=response.getWriter();

                    response.setContentType("text/html");
                    RequestDispatcher rd2=request.getRequestDispatcher("index.html");

                    String emailid = request.getParameter("emailid");
                    String password=request.getParameter("password");
                    try {
                        Class.forName("com.mysql.cj.jdbc.Driver");
                        Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/db","root","root");
                        PreparedStatement pstmt=con.prepareStatement("insert into login values (?,?) ");
                        pstmt.setString(1, emailid);
                        pstmt.setString(2, password);
                        int rs=pstmt.executeUpdate();
                        if(rs>0) {
                            pw.println("Account created successfully !");
                        }
                        }catch(Exception e) {
                                pw.println("Account already existed !");
                                rd2.include(request, response);
                        }
                }

        }
```



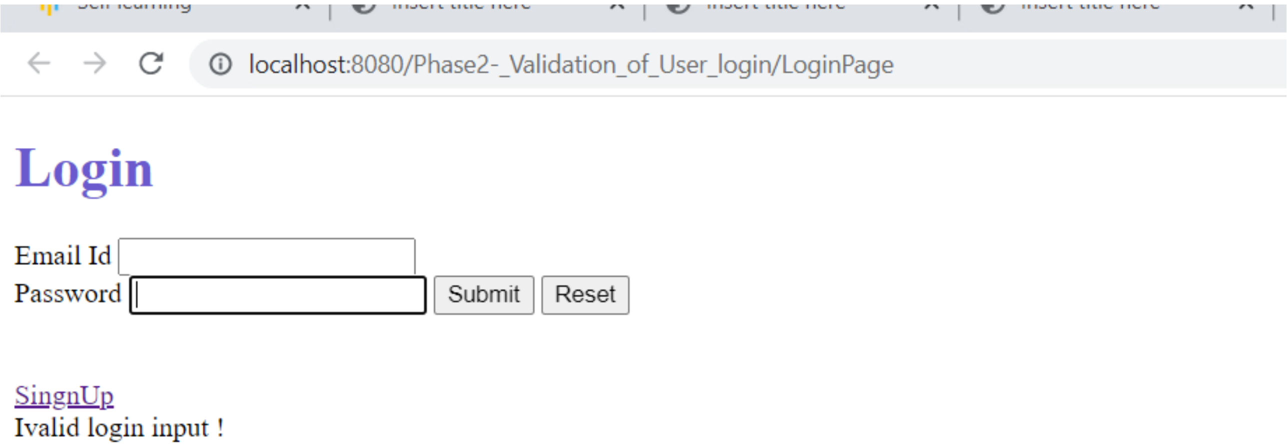
localhost:8080/Phase2-_Validation_of_User_login/LoginPage?emailid=moulaali%40gmail.com&

Login

Email Id

Password

SingnUp



localhost:8080/Phase2-_Validation_of_User_login/LoginPage

Login

Email Id

Password

SingnUp

Ivalid login input !

Step 1.1.3: Configuring web.xml

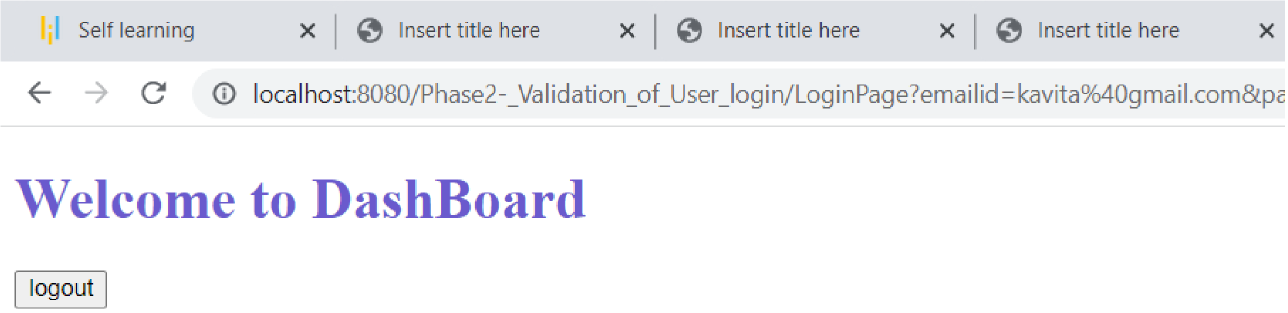
- In the Project Explorer, expand ValidationOfUserLogin**->WebContent->WEB-INF**
- Double click **web.xml** to open it in the editor
- Enter the following script:

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee" xsi:schemaLoca-
tion="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-
app_2_5.xsd" id="WebApp_ID" version="2.5">
  <display-name>Phase2- Validation of User login</display-name>
  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
    <welcome-file>index.jsp</welcome-file>
    <welcome-file>index.htm</welcome-file>
    <welcome-file>default.html</welcome-file>
    <welcome-file>default.jsp</welcome-file>
    <welcome-file>default.htm</welcome-file>
  </welcome-file-list>
  <servlet>
    <description></description>
    <display-name>LoginPage</display-name>
    <servlet-name>LoginPage</servlet-name>
    <servlet-class>com.LoginPage</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>LoginPage</servlet-name>
    <url-pattern>/LoginPage</url-pattern>
  </servlet-mapping>
   </web-app>
```

Step 1.1.4: Creating a HTML logOut.html

- Right click on Project and choose **New->HTML**
- In **Class Name,** enter **logOut.html** and click on **Finish**
- **Enter the following code:**

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
                   <h1 style="color:SlateBlue;"> Login</h1>
            <form action="LoginPage">
            <labael>Email Id</labael>
            <input type ="email" name="emailid"/><br/>
            <labael>Password</labael>

            <input type ="password" name="password"/>
            <input type ="submit" name="submit"/>
            <input type ="reset" name="reset"/>
            </form><br/><br/>
            <a href ="signUp.html">SingnUp</a><br/>

</body>
</html>
```



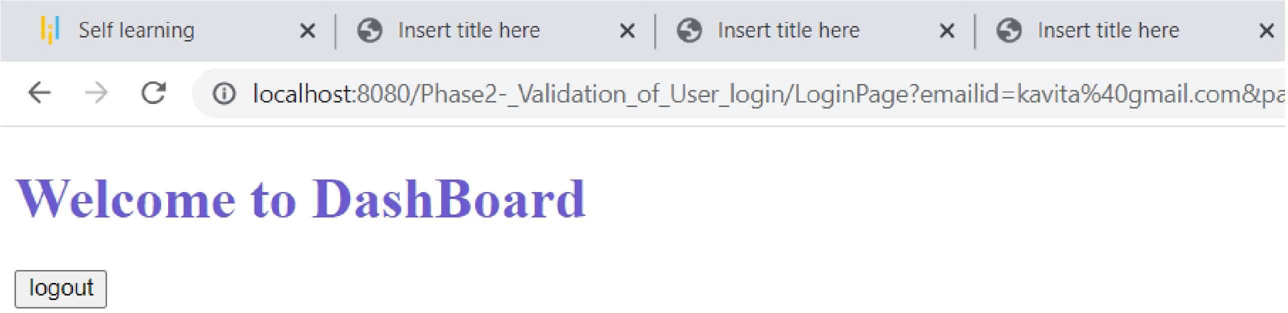
localhost:8080/Phase2-_Validation_of_User_login/LoginPage?emailid=kavita%40gmail.com&pa

Welcome to DashBoard

Step 1.1.4: Creating a JSP file

- Right click on Project and choose **New->JSP File**
- In **Class Name,** enter **dashboard** and click on **Finish**
- **Enter the following code:**

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<h1 style="color:SlateBlue;"> Welcome to DashBoard</h1>
<a href ="LoginPage"> <button>logout</button></a><br>
</body>
</html>
```



localhost:8080/Phase2-_Validation_of_User_login/LoginPage?emailid=kavita%40gmail.com&pa

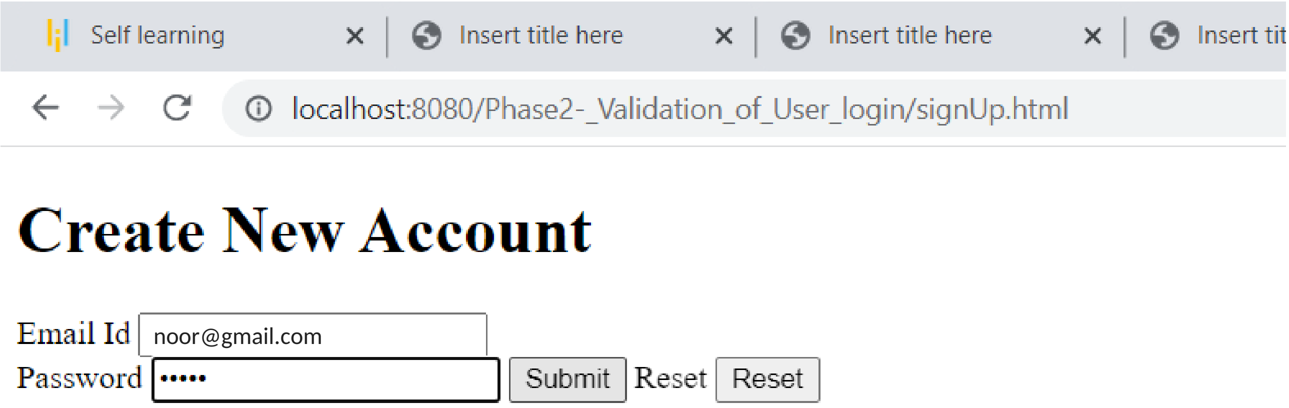
Welcome to DashBoard

Step 1.1.5: Creating a HTML signUp.html

- Right click on Project and choose **New->HTML**
- In **Class Name,** enter **signUp.html** and click on **Finish**

- **Enter the following code:**

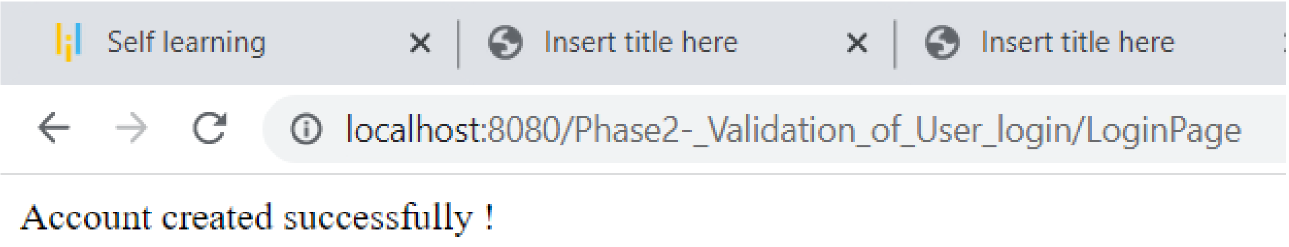
```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
                    <h1>Create New Account </h1>

                    <form action="LoginPage" method="post">

            <labael>Email Id</labael>
            <input type ="email" name="emailid"/><br/>
            <labael>Password</labael>
            <input type ="password" name="password"/>
            <input type ="submit" name="submit"/>
            <label>Reset</label>
            <input type ="reset" name="reset"/>
            </form>
</body>
   </html>
```



Create New Account

Email Id noor@gmail.com

Password •••••



Account created successfully !

Step 1.1.6: Checking for servlet-api.jar

- Before building the project, we need to add servlet-api.jar to the project
- To add it to the project, follow the below mentioned steps:
  - In the Project Explorer, right click on **ValidationOfUserLogin** and choose **Properties**
  - Select **Java Build Path** from the options on the left
  - Click on **Libraries** tab on the right
  - Under **ClassPath,** expand the node that says **Apache Tomcat**
  - If there is an existing entry for **servlet-api.jar,** then click on **Cancel** and exit the window
  - If it is not there, then click on **Classpath** entry and click on **Add External JARs** button on the right
  - From the file list, select **servlet-api.jar** file and click **Ok**
  - Click on **Apply and Close**

Step 1.1.7: Building the project

- From the **Project** menu at the top, click on **Build**
- If any compile errors are shown, fix them as required

Step 1.1.8: Publishing and starting the project

- If you do not see the **Servers** tab near the bottom of the IDE, go to the Window menu and click **Show View->Servers**
- Right click on the **Server** entry and choose **Add and Remove**
- Click the **Add** button to move **ServletGetPost** from the **Available** list to the **Configured** List
- Click **Finish**
- Right click on the **Server** entry and click on **Publish**
- Right click on the **Server** entry and click on **Start**
- This will start the server

Step 1.1.9: Running the project

- To run the project, open a web browser and type: **http://localhost:8086/ValidationOfUserLogin**

Step 1.1.10: Pushing the code to your GitHub repositories

- Open your command prompt and navigate to the folder where you have created your files.

  cd <folder path>

- Initialize your repository using the following command:

  git init

- Add all the files to your git repository using the following command:

  git add .

- Commit the changes using the following command:

  git commit . -m "Changes have been committed."

- Push the files to the folder you initially created using the following command:

```
git push -u origin master
```