**Validation of the User Login**

**Github link:**

[**https://github.com/saranya611/testapp/validation\_user**](https://github.com/saranya611/testapp/validation_user)

**Write up:**

**Step1:** Creating a dynamic web project

* Open Eclipse
* Go to the **File** menu. Choose **New->Dynamic Web Project**
* Enter the project name as **SimpleServletApp**. 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 2:** Creating an HTML page

* In the Project Explorer, expand the project **SimpleServletApp.**
* Expand **WebContent**. Right click on **WebContent** . Choose **New->HTML File**
* Enter the filename as **index.html** and click on **Finish**
* Create another **HTML File** as **loginpage.html** and click on **Finish**
  + - * Created form which collect the username and password from user along with submit and reset buttons.
      * Form action is on **LoginApp**

**Step 3:** Creating a servlet

* In the Project Explorer, expand **SimpleServletApp->Java Resources**
* Right click on **src** and choose **New->Servlet**
* In **Class Name,** enter **LoginApp** and click on **Finish**

**Step 4:** Checking for servlet-api.jar

* Before building the project, we need to add servlet-api.jar to the project
* Servlet-api.jar file is already present in your practice lab.
* To add it to the project, follow the below mentioned steps:
  + - * In the Project Explorer, right click on **LoginApp** 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 5:** Building the project

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

**Step 6:** Publishing and starting the project

* If you do not see the **Servers** tab near the bottom of the IDE, go to **Window** menu and click on **Show View->Servers**
* Right click on the **Server** entry and choose **Add and Remove**
* Click the **Add** button to move **LoginApp** from the **Available** list to the **Configured** list
* Click on **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 7:** Running the project

* To run the project, open a web browser and type:

[**http://** [**localhost:8282/SimpleServletApp/LoginApp**](http://localhost:8282/SimpleServletApp/LoginApp)](http://localhost:8080/ServletConcept)

* On welcome page, click on **login**
* The login form appearance and asks for **username** and **password**
  + - * If the password and username matches , it displays **“successfully logged in..!”**
      * Otherwise, it displays **“invalid username or password, try again later..!”**