Java Servlet Project

To : Make a LOGIN form using the Servlet interface.

Download the code from the QR to computer

Steps :

1)open eclipse ide EE (enterprises edition)

2) keep ready : Tomcat Server + servlet .jar
***will be included in the Github Repositry**

3) Create new Dynamic Web project by Rt. clicking on the project window

*** Before step 2 Rt.click the project and add the jar file:**
Rtclick -> buildpath->configure build path -> libraries Tab-> classpath-> then Add Ext.JarFiles -> at the left side , go to the location and include it -> apply and Apply and close

1. **Now follow it :**
   Create a Dynamic Web Project:
   - Go to **`File`** -> **`New`** -> **`Dynamic Web Project`**.
   - Enter "LoginServletProject" as the project name.
   - Choose the Target Runtime as Apache Tomcat 9.0.
   - Click **`Next`** and then **`Finish`**.
2. **Create a Servlet**:
   - Right-click on the **`src`** folder in your project.
   - Go to **`New`** -> **`Servlet`**.
   - Enter the package name and the servlet class name (e.g., **`com.yourcompany.servlets.LoginServlet`**).
   - Click **`Next`**.
   - Specify the URL mapping (e.g., **`/login`**).
   - Click **`Next`**.
   - Select the methods you want to implement (**`doGet`**, **`doPost`**, etc.).
   - Click **`Finish`**.

3. **Implement the Servlet**:
   - In the newly created servlet class, you can implement the `doGet` and/or `doPost` methods to handle HTTP requests.
   - Write your login logic within these methods.
4. **Configure Deployment Descriptor (web.xml)** (Optional for Servlet 3.0+):
   - If you're not using Servlet 3.0 annotation-based configuration, you might need to configure the servlet in the `web.xml` file.
   - Right-click on the `WebContent/WEB-INF` folder.
   - Go to `New` -> `XML File`.
   - Enter `web.xml` as the file name and click `Finish`.
   - Configure your servlet mapping and other servlet-related configurations in this file.
5. **Configure Deployment Assembly**:
   - Right-click on the project.
   - Go to `Properties`.
   - Choose `Deployment Assembly`.
   - Ensure that the necessary resources (e.g., `src`, `WebContent`, `WEB-INF`) are being deployed properly.

7 in the Web App folder there : make 2 html files :

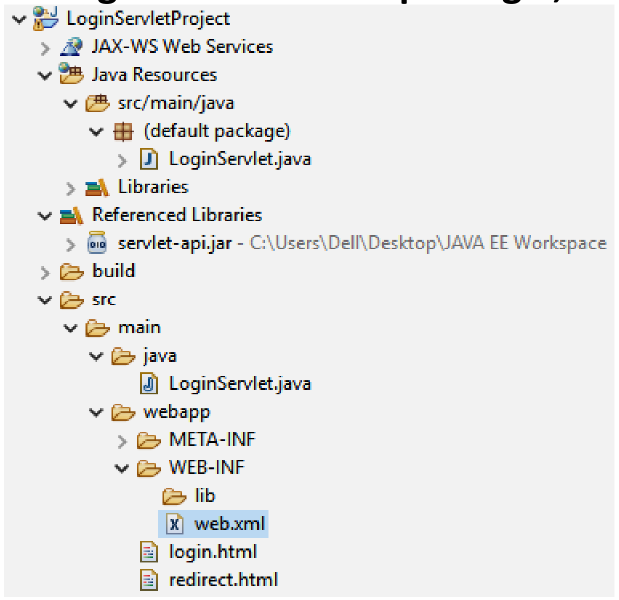
login.html , and the redirect.html

in the web app inside the lib folder -> rt click -> then new -> other -> search xml file -> rename the web.xml ,

8. In the src -> rt click add the servlet , Name : LoginServlet
No package was made for it , it will go in the default package , so leave it empty.

9. See the Hirearchy:

make the files according to it



```
v LoginServletProject
  > JAX-WS Web Services
  v Java Resources
    v src/main/java
      v (default package)
        > LoginServlet.java
    > Libraries
  v Referenced Libraries
    > servlet-api.jar - C:\Users\Dell\Desktop\JAVA EE Workspace
  > build
  v src
    v main
      v java
          LoginServlet.java
      v webapp
        > META-INF
        v WEB-INF
            lib
            web.xml
          login.html
          redirect.html
```

10. Add the coding to the respective files.. {End of the Descriptive Guide}

Coding of the : LoginServlet.java keep in the Src:

```
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/LoginServlet")
public class LoginServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // Hardcoded username and password for validation
        String validUsername = "aayush";
        String validPassword = "pwd2326";

        // Get the username and password entered by the user
        String username = request.getParameter("username");
        String password = request.getParameter("password");

        // Check if the entered username and password match the valid credentials
        if (username != null && password != null && username.equals(validUsername) && password.equals(validPassword)) {
            // Redirect to redirect.html with username as parameter
            response.sendRedirect("redirect.html?username=" + username);
        } else {
            // Credentials are incorrect, redirect back to login.html
            // Notify user to give correct username and password
            response.sendRedirect("login.html?error=incorrect");
        }
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // Redirect to login.html for GET requests
        response.sendRedirect("login.html");
    }
}
```

Coding of the : login.html keep in the webapp:

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login Page</title>
</head>
<body>
    <h1>This is the login page</h1>
    <form action="LoginServlet" method="post">
        <label for="username">Username:</label>
        <input type="text" id="username" name="username" required><br><br>
        <label for="password">Password:</label>
        <input type="password" id="password" name="password" required><br><br>
        <input type="submit" value="Login">
    </form>
</body>
</html>
```

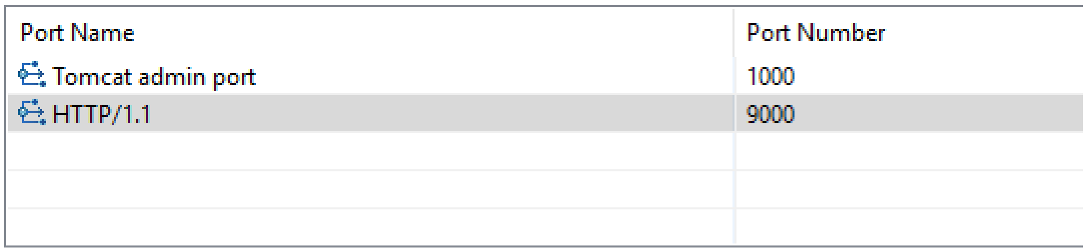
Coding of the : redirect.html keep in the webapp:

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Successful Login</title>
</head>
<body>
    <h1>Successful Login</h1>
    <p>Hello <span id="username"></span>, welcome!</p>

    <script>
        // Function to retrieve URL parameter by name
        function getUrlParameter(name) {
            name = name.replace(/[\[]/, '\\[').replace(/[\]]/, '\\]');
            var regex = new RegExp('[\\?&]' + name + '=([^&#]*)');
            var results = regex.exec(location.search);
            return results === null ? '' : decodeURIComponent(results[1].replace(/\+/g, ' '));
        }

        // Get the username parameter from the URL
        var username = getUrlParameter('username');

        // Display the username on the page
        document.getElementById('username').textContent = username;
    </script>
  </body>
  </html>
```

Coding of the : web.xml keep in the webapp->web-inf->lib:

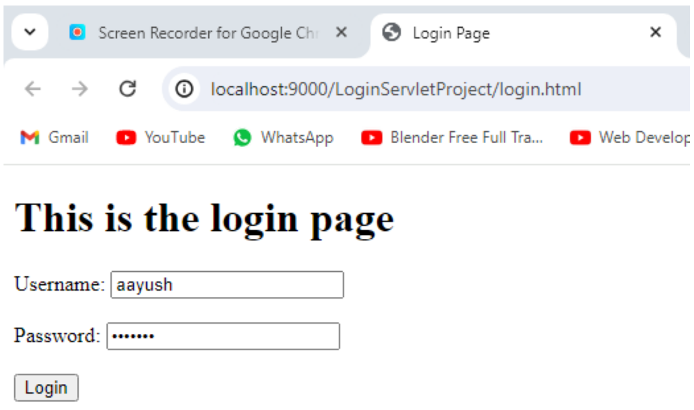
```
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd"
    version="4.0">

    <servlet>
        <servlet-name>LoginServlet</servlet-name>
        <servlet-class>LoginServlet</servlet-class>
    </servlet>

    <servlet-mapping>
        <servlet-name>LoginServlet</servlet-name>
        <url-pattern>/LoginServlet</url-pattern>
    </servlet-mapping>

</web-app>
```

The port no. used for the Apache Tomcat 9 was



Port Name	Port Number
Tomcat admin port	1000
HTTP/1.1	9000

Run->run on server -> next->finish



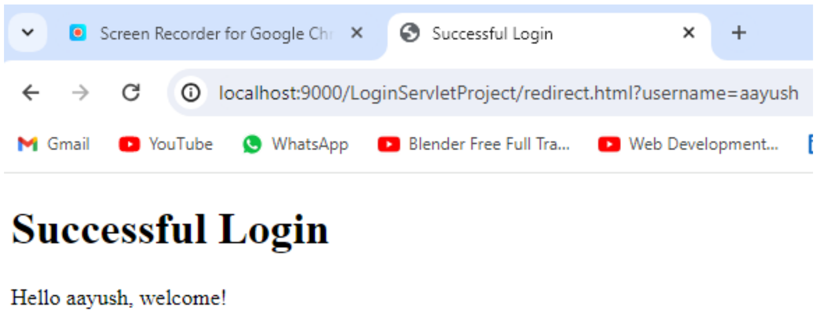
localhost:9000/LoginServletProject/login.html

This is the login page

Username: aayush

Password: •••••••

Login



localhost:9000/LoginServletProject/redirect.html?username=aayush

Successful Login

Hello aayush, welcome!