**Validation of the User Login**

**DESCRIPTION**

**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

**Creating a dynamic web project**

* Open Eclipse
* Go the **File** menu. Choose **New->Dynamic Web Project**
* Enter the project name as **UserLogin**. 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

**Creating an HTML page**

* In the Project Explorer, expand the project **LoginLogout**
* Expand **WebContent**. Right click on **WebContent**. Choose **New->HTML File**
* Enter the filename as **index.html** and click on **Finish**
* Enter the following code:

<!DOCTYPE html>

<html>

<head>

<meta name=*"viewport"* content=*"width=device-width, initial-scale=1"*>

<style>

/\* Style all input fields \*/

**input** {

width: *100%*;

padding: *12px*;

border: *1px solid #ccc*;

border-radius: *4px*;

box-sizing: *border-box*;

margin-top: *6px*;

margin-bottom: *16px*;

}

/\* Style the submit button \*/

**input**[type=submit] {

background-color: *#04AA6D*;

color: *white*;

}

/\* Style the container for inputs \*/

*.container* {

background-color: *#f1f1f1*;

padding: *20px*;

}

/\* The message box is shown when the user clicks on the password field \*/

*#message* {

display:*none*;

background: *#f1f1f1*;

color: *#000*;

position: *relative*;

padding: *20px*;

margin-top: *10px*;

}

*#message* **p** {

padding: *10px 35px*;

font-size: *18px*;

}

/\* Add a green text color and a checkmark when the requirements are right \*/

*.valid* {

color: *green*;

}

*.valid:before* {

position: *relative*;

left: *-35px*;

content: *"✔"*;

}

/\* Add a red text color and an "x" when the requirements are wrong \*/

*.invalid* {

color: *red*;

}

*.invalid:before* {

position: *relative*;

left: *-35px*;

content: *"✖"*;

}

</style>

</head>

<body>

<h3>Password Validation</h3>

<p>Try to submit the form.</p>

<div class=*"container"*>

<form action=*"/action\_page.php"*>

<label for=*"usrname"*>Username</label>

<input type=*"text"* id=*"usrname"* name=*"usrname"* required>

<label for=*"psw"*>Password</label>

<input type=*"password"* id=*"psw"* name=*"psw"* pattern=*"(?=.\*\d)(?=.\*[a-z])(?=.\*[A-Z]).{8,}"* title=*"Must contain at least one number and one uppercase and lowercase letter, and at least 8 or more characters"* required>

<input type=*"submit"* value=*"Submit"*>

</form>

</div>

<div id=*"message"*>

<h3>Password must contain the following:</h3>

<p id=*"letter"* class=*"invalid"*>A <b>lowercase</b> letter</p>

<p id=*"capital"* class=*"invalid"*>A <b>capital (uppercase)</b> letter</p>

<p id=*"number"* class=*"invalid"*>A <b>number</b></p>

<p id=*"length"* class=*"invalid"*>Minimum <b>8 characters</b></p>

</div>

<script>

**var** myInput = document.getElementById("psw");

**var** letter = document.getElementById("letter");

**var** capital = document.getElementById("capital");

**var** number = document.getElementById("number");

**var** length = document.getElementById("length");

// When the user clicks on the password field, show the message box

myInput.onfocus = **function**() {

document.getElementById("message").style.display = "block";

}

// When the user clicks outside of the password field, hide the message box

myInput.onblur = **function**() {

document.getElementById("message").style.display = "none";

}

// When the user starts to type something inside the password field

myInput.onkeyup = **function**() {

// Validate lowercase letters

**var** lowerCaseLetters = /[a-z]/g;

**if**(myInput.value.match(lowerCaseLetters)) {

letter.classList.remove("invalid");

letter.classList.add("valid");

} **else** {

letter.classList.remove("valid");

letter.classList.add("invalid");

}

// Validate capital letters

**var** upperCaseLetters = /[A-Z]/g;

**if**(myInput.value.match(upperCaseLetters)) {

capital.classList.remove("invalid");

capital.classList.add("valid");

} **else** {

capital.classList.remove("valid");

capital.classList.add("invalid");

}

// Validate numbers

**var** numbers = /[0-9]/g;

**if**(myInput.value.match(numbers)) {

number.classList.remove("invalid");

number.classList.add("valid");

} **else** {

number.classList.remove("valid");

number.classList.add("invalid");

}

// Validate length

**if**(myInput.value.length >= 8) {

length.classList.remove("invalid");

length.classList.add("valid");

} **else** {

length.classList.remove("valid");

length.classList.add("invalid");

}

}

</script>

</body>

</html>

Click on the **Save** icon

**Building the project**

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

**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 **User\_Password\_Validation** 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

**Running the project**

* To run the project, open a web browser and type: [localhost:8088/User\_Password\_Validation/](http://localhost:8088/User_Password_Validation/)

**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**