Source Code Login And Registration Page

1.Login.html

<!DOCTYPE html>

<html>

<head>

    <title>Login Page</title>

</head>

<body>

    <h1>Login Form</h1>

    <form id="loginForm" action="login-success.html" method="post">

        <label for="username">Username:</label>

        <input type="text" id="username" name="username"><br><br>

        <label for="password">Password:</label>

        <input type="password" id="password" name="password"><br><br>

        <input type="submit" value="Login">

    </form>

</body>

</html>

2.Login-Success.Html

<!DOCTYPE html>

<html>

<head>

    <title>Login Page</title>

</head>

<body>

    <h1>Login SuccessFully</h1>

</body>

</html>

3.LoginTest.Java

**package** com.example;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.WebElement;

**import** org.openqa.selenium.chrome.ChromeDriver;

**import** org.testng.annotations.AfterTest;

**import** org.testng.annotations.BeforeTest;

**import** org.testng.annotations.Test;

**public** **class** LoginTest {

WebDriver wd = **null**;

@BeforeTest

**public** **void** init() {

System.*setProperty*("webdriver.chrome.driver", "C:\\Users\\Lenovo\\Downloads\\chromedriver-win32\\chromedriver-win32\\chromedriver.exe");

wd = **new** ChromeDriver();

wd.manage().window().maximize();

}

@Test(priority = 1)

**public** **void** openLoginPage() {

wd.get("http://127.0.0.1:5500/cli/login.html");

}

@Test(priority = 2)

**public** **void** fillLoginForm() {

// Fill in the registration form fields

WebElement usernameField = wd.findElement(By.*id*("username"));

WebElement passwordField = wd.findElement(By.*id*("password"));

usernameField.sendKeys("yourUsername");

passwordField.sendKeys("yourPassword");

// Submit the registration form

WebElement submitButton = wd.findElement(By.*cssSelector*("input[type='submit']"));

submitButton.click();

// Wait for the registration to complete (you can use explicit waits here)

// Verify the registration was successful

String currentUrl = wd.getCurrentUrl();

**assert** currentUrl.contains("login-success.html"); // Replace with the success URL

}

@Test(priority = 3)

**public** **void** openLoginSuccessPage() {

wd.get("http://127.0.0.1:5500/cli/login-success.html");

}

@AfterTest

**public** **void** close() {

wd.close();

}

}

4.Registratiom.html

<!DOCTYPE html>

<html>

<head>

    <title>Registration Page</title>

</head>

<body>

    <h1>Registration Form</h1>

    <form id="registrationForm" action="registration-success.html" method="post">

        <label for="username">Username:</label>

        <input type="text" id="username" name="username"><br><br>

        <label for="email">Email:</label>

        <input type="email" id="email" name="email"><br><br>

        <label for="password">Password:</label>

        <input type="password" id="password" name="password"><br><br>

        <label for="confirmPassword">Confirm Password:</label>

        <input type="password" id="confirmPassword" name="confirmPassword"><br><br>

        <input type="submit" value="Register">

    </form>

</body>

</html>

5.Registration-Success.Html

<!DOCTYPE html>

<html>

<head>

    <title>Login Page</title>

</head>

<body>

    <h1>Registration SuccessFully</h1>

</body>

</html>

6.RegistrationTest.Java

**package** com.example;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.WebElement;

**import** org.openqa.selenium.chrome.ChromeDriver;

**import** org.testng.annotations.AfterTest;

**import** org.testng.annotations.BeforeTest;

**import** org.testng.annotations.Test;

**public** **class** RegistrationAutomation {

WebDriver wd = **null**;

@BeforeTest

**public** **void** init() {

System.*setProperty*("webdriver.chrome.driver", "C:\\Users\\Lenovo\\Downloads\\chromedriver-win32\\chromedriver-win32\\chromedriver.exe");

wd = **new** ChromeDriver();

wd.manage().window().maximize();

}

@Test(priority = 1)

**public** **void** openRegistrationPage() {

wd.get("http://127.0.0.1:5500/cli/Registration.html");

}

@Test(priority = 2)

**public** **void** fillRegistrationForm() {

// Fill in the registration form fields

WebElement usernameField = wd.findElement(By.*id*("username"));

WebElement emailField = wd.findElement(By.*id*("email"));

WebElement passwordField = wd.findElement(By.*id*("password"));

WebElement confirmPasswordField = wd.findElement(By.*id*("confirmPassword"));

usernameField.sendKeys("yourUsername");

emailField.sendKeys("youremail@example.com");

passwordField.sendKeys("yourPassword");

confirmPasswordField.sendKeys("yourPassword");

// Submit the registration form

WebElement submitButton = wd.findElement(By.*cssSelector*("input[type='submit']"));

submitButton.click();

// Wait for the registration to complete (you can use explicit waits here)

// Verify the registration was successful

String currentUrl = wd.getCurrentUrl();

**assert** currentUrl.contains("registration-success.html"); // Replace with the success URL

}

@Test(priority = 3)

**public** **void** openRegistrationSuccessPage() {

wd.get("http://127.0.0.1:5500/cli/registration-success.html");

}

@AfterTest

**public** **void** close() {

wd.close();

}

}

7.Pom.xml file

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4\_0\_0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>Mphasis</groupId>

<artifactId>AutomateTesting</artifactId>

<packaging>war</packaging>

<version>0.0.1-SNAPSHOT</version>

<name>AutomateTesting Maven Webapp</name>

<url>http://maven.apache.org</url>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>3.8.1</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.seleniumhq.selenium</groupId>

<artifactId>selenium-server</artifactId>

<version>3.141.59</version>

</dependency>

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-log4j12</artifactId>

<version>1.7.26</version>

</dependency>

</dependencies>

<build>

<finalName>AutomateTesting</finalName>

</build>

</project>

8.Testng.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">

<suite name=*"Suite"*>

<test name=*"Test1"* thread-count=*"5"* parallel=*"classes"*>

<classes>

<class name=*"com.example.RegistrationAutomation"*/>

<class name=*"com.example.LoginTest"*/>

</classes>

</test> <!-- Test -->

</suite> <!-- Suite -->