**Automate a webapp**

**Code:**

**Index.html**

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>LogIn Form</title>

<style type=*"text/css"*>

/\* Bordered form \*/

**form** {

border: *3px solid #f1f1f1*;

/\*Make the form in center place, using max-width:500px and margin:auto; \*/

max-width: *600px*;

margin: *auto*;

}

/\* Full-width inputs \*/

**input**[type=text]**,** **input**[type=password] {

width: *100%*;

padding: *12px 20px*;

margin: *8px 0*;

display: *inline-block*;

border: *1px solid #ccc*;

box-sizing: *border-box*;

}

/\* Set a style for all buttons \*/

**button** {

background-color: *#04AA6D*;

color: *white*;

padding: *14px 20px*;

margin: *8px 0*;

border: *none*;

cursor: *pointer*;

width: *100%*;

}

/\* Add a hover effect for buttons \*/

**button***:hover* {

opacity: *0.8*;

}

/\* Extra style for the cancel button (red) \*/

*.cancelbtn* {

width: *auto*;

padding: *10px 18px*;

background-color: *#f44336*;

}

/\* Add padding to containers \*/

*.container* {

padding: *16px*;

}

/\* The "Forgot password" text \*/

**span***.psw* {

float: *right*;

padding-top: *16px*;

}

/\* Change styles for span and cancel button on extra small screens \*/

@media screen and (max-width: 300px) {

**span***.psw* {

display: *block*;

float: *none*;

}

*.cancelbtn* {

width: *100%*;

}

}

</style>

</head>

<body>

<form action=*"Login.html"* method=*"post"*>

<fieldset>

<div class=*"container"*>

<label for=*"uname"*><b>Username</b></label>

<input type=*"text"* placeholder=*"Enter Username"* name=*"uname"* required> <br><br>

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

<input type=*"password"* placeholder=*"Enter Password"* name=*"psw"* required><br><br>

<button type=*"submit"*>Login</button> <br>

<label>

<input type=*"checkbox"* checked=*"checked"* name=*"remember"*> Remember me

</label>

</div>

<hr>

<div class=*"container"* style="background-color:*#f1f1f1*">

<span class=*"psw"*>Forgot <a href=*"#"*>password?</a></span>

</div>

</fieldset>

</form>

</body>

</html>

**Login.html**

Successs

**Registration.html**

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Register here</title>

<style type=*"text/css"*>

**form** {

border: *3px solid #f1f1f1*;

/\*Make the form in center place, using max-width:500px and margin:auto; \*/

max-width: *700px*;

margin: *auto*;

}

*.container* {

padding: *1px*;

background-color: *#f1f1f1*;

}

/\* Full-width input fields \*/

**input**[type=text]**,** **input**[type=password] {

width: *95%*;

padding: *15px*;

margin: *5px 0 22px 0*;

display: *inline-block*;

border: *none*;

background: *light*;

}

**input**[type=text]*:focus***,** **input**[type=password]*:focus* {

background-color: *#ddd*;

outline: *none*;

}

*.registerbtn*{

padding: *16px 20px*;

margin: *8px 0*;

border: *none*;

cursor: *pointer*;

width: *100%*;

opacity: *0.9*;

background-color: *#04AA6D*;

color: *white*;

}

</style>

</head>

<body>

<form action = *"Login.html"*>

<div class = *"container"*>

<fieldset>

<h1>Registration Form </h1>

<h3> Please fill in this form to create an account: </h3><br>

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

<input type =*"text"* placeholder = *"Enter Email"* name=*"email"* id=*"email"* required><br><br>

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

<input type =*"password"* placeholder = *"Enter Password"* name=*"psw"* id=*"psw"* required><br><br>

<label for =*"psw-repeat"*><b>Repeat Password:</b></label>

<input type =*"password"* placeholder = *"Repeat Password"* name=*"psw-repeat"* id=*"psw-repeat"* required><br><br>

<label for=*"role"*><b>Select Role</b></label>

<input type=*"radio"* name=*"radio"* id=*"admin"* required> Admin

<input type=*"radio"* name=*"radio"* id=*"user"* required> User

<input type=*"radio"* name=*"radio"* id=*"vendor"* required> Vendor <br> <br>

<label for=*"skill"*><b>Select Skill:</b></label>

<input type=*"checkbox"* name =*"skill"* id=*"skill1"* required> Java

<input type=*"checkbox"* name =*"skill"* id=*"skill2"* required> MySql

<input type=*"checkbox"* name =*"skill"* id=*"skill3"* required> Selenium Webdriver<br><br>

<button type=*"submit"* class=*"registerbtn"*>Register</button>

</fieldset>

</div>

</form>

</body>

</html>

**LoginFormTest.java**

package LoginForm;

import static org.testng.Assert.assertEquals;

import static org.testng.Assert.assertTrue;

import java.io.File;

import java.io.IOException;

import org.openqa.selenium.By;

import org.openqa.selenium.Dimension;

import org.openqa.selenium.OutputType;

import org.openqa.selenium.TakesScreenshot;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.edge.EdgeDriver;

import org.openqa.selenium.io.FileHandler;

import org.testng.annotations.AfterTest;

import org.testng.annotations.BeforeMethod;

import org.testng.annotations.BeforeTest;

import org.testng.annotations.Test;

public class LoginFormTest {

final String siteURL = "http://localhost:9898/AutomateAWebapp/";

final String driverPath ="C:\\Users\\SARANYA V\\eclipse\\chromedriver\_win32\\chromedriver.exe";

WebDriver driver;

@BeforeTest

void setUp() throws Exception {

//Set Up

System.setProperty("webdriver.chrome.driver",driverPath);

driver=new ChromeDriver();

driver.get(siteURL);

//Add maximize size for snapshot

driver.manage().window().maximize();

}

@Test

void ftestLogInButton() throws InterruptedException, IOException {

//Find Input Field

WebElement button = driver.findElement(By.cssSelector("body > form > fieldset > div:nth-child(1) > button"));

assertTrue(button.isDisplayed());

assertTrue(button.isEnabled());

button.submit();

Thread.sleep(2000);

//Step-1 :: type cast driver object into takeScreenshot instance

TakesScreenshot tsc = (TakesScreenshot) driver;

//Step-2 :: Generate a screenshot as file

File rsc = tsc.getScreenshotAs(OutputType.FILE);

FileHandler.copy(rsc,new File("login.png"));

}

@Test

void detestLink() throws InterruptedException, IOException {

//Find Input Field

WebElement link = driver.findElement(By.cssSelector("body > form > fieldset > div:nth-child(3) > span > a"));

assertTrue(link.isDisplayed());

assertTrue(link.isEnabled());

link.click();

Thread.sleep(2000);

//Step-1 :: type cast driver object into takeScreenshot instance

TakesScreenshot tsc = (TakesScreenshot) driver;

//Step-2 :: Generate a screenshot as file

File rsc = tsc.getScreenshotAs(OutputType.FILE);

FileHandler.copy(rsc,new File("link-verification.png"));

}

@Test

void btestUserInput() throws InterruptedException, IOException {

//Find Input Field

WebElement input = driver.findElement(By.cssSelector("body > form > fieldset > div:nth-child(1) > input[type=text]:nth-child(2)"));

assertTrue(input.isDisplayed());

assertTrue(input.isEnabled());

//Enter yourname

input.sendKeys("saranya");

//Step-1 :: type cast driver object into takeScreenshot instance

TakesScreenshot tsc = (TakesScreenshot) driver;

//Step-2 :: Generate a screenshot as file

File rsc = tsc.getScreenshotAs(OutputType.FILE);

FileHandler.copy(rsc,new File("user.png"));

Thread.sleep(2000);

}

@Test

void ctestUserPassword() throws InterruptedException, IOException {

//Find Input Field

WebElement input = driver.findElement(By.cssSelector("body > form > fieldset > div:nth-child(1) > input[type=password]:nth-child(6)"));

assertTrue(input.isDisplayed());

assertTrue(input.isEnabled());

//Enter yourpassword

input.sendKeys("saranya611");

Thread.sleep(2000);

//Step-1 :: type cast driver object into takeScreenshot instance

TakesScreenshot tsc = (TakesScreenshot) driver;

//Step-2 :: Generate a screenshot as file

File rsc = tsc.getScreenshotAs(OutputType.FILE);

FileHandler.copy(rsc,new File("password.png"));

}

@Test

void dtestCheckBox() throws InterruptedException, IOException {

//Find Input Field

WebElement checkbox = driver.findElement(By.cssSelector("body > form > fieldset > div:nth-child(1) > label:nth-child(11) > input[type=checkbox]"));

assertEquals(true, checkbox.isDisplayed());

assertEquals(true, checkbox.isEnabled());

checkbox.click();

assertEquals(false, checkbox.isSelected());

Thread.sleep(2000);

//Step-1 :: type cast driver object into takeScreenshot instance

TakesScreenshot tsc = (TakesScreenshot) driver;

//Step-2 :: Generate a screenshot as file

File rsc = tsc.getScreenshotAs(OutputType.FILE);

FileHandler.copy(rsc,new File("checkBox.png"));

}

}

**RegistrationFormTest.java**

package LoginForm;

import static org.testng.Assert.assertEquals;

import static org.testng.Assert.assertTrue;

import java.io.File;

import java.io.IOException;

import org.openqa.selenium.By;

import org.openqa.selenium.Dimension;

import org.openqa.selenium.OutputType;

import org.openqa.selenium.TakesScreenshot;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.edge.EdgeDriver;

import org.openqa.selenium.io.FileHandler;

import org.testng.annotations.AfterTest;

import org.testng.annotations.BeforeMethod;

import org.testng.annotations.BeforeTest;

import org.testng.annotations.Test;

public class LoginFormTest {

final String siteURL = "http://localhost:9898/AutomateAWebapp/";

final String driverPath ="C:\\Users\\SARANYA V\\eclipse\\chromedriver\_win32\\chromedriver.exe";

WebDriver driver;

@BeforeTest

void setUp() throws Exception {

//Set Up

System.setProperty("webdriver.chrome.driver",driverPath);

driver=new ChromeDriver();

driver.get(siteURL);

//Add maximize size for snapshot

driver.manage().window().maximize();

}

@Test

void ftestLogInButton() throws InterruptedException, IOException {

//Find Input Field

WebElement button = driver.findElement(By.cssSelector("body > form > fieldset > div:nth-child(1) > button"));

assertTrue(button.isDisplayed());

assertTrue(button.isEnabled());

button.submit();

Thread.sleep(2000);

//Step-1 :: type cast driver object into takeScreenshot instance

TakesScreenshot tsc = (TakesScreenshot) driver;

//Step-2 :: Generate a screenshot as file

File rsc = tsc.getScreenshotAs(OutputType.FILE);

FileHandler.copy(rsc,new File("login.png"));

}

@Test

void detestLink() throws InterruptedException, IOException {

//Find Input Field

WebElement link = driver.findElement(By.cssSelector("body > form > fieldset > div:nth-child(3) > span > a"));

assertTrue(link.isDisplayed());

assertTrue(link.isEnabled());

link.click();

Thread.sleep(2000);

//Step-1 :: type cast driver object into takeScreenshot instance

TakesScreenshot tsc = (TakesScreenshot) driver;

//Step-2 :: Generate a screenshot as file

File rsc = tsc.getScreenshotAs(OutputType.FILE);

FileHandler.copy(rsc,new File("link-verification.png"));

}

@Test

void btestUserInput() throws InterruptedException, IOException {

//Find Input Field

WebElement input = driver.findElement(By.cssSelector("body > form > fieldset > div:nth-child(1) > input[type=text]:nth-child(2)"));

assertTrue(input.isDisplayed());

assertTrue(input.isEnabled());

//Enter yourname

input.sendKeys("saranya");

//Step-1 :: type cast driver object into takeScreenshot instance

TakesScreenshot tsc = (TakesScreenshot) driver;

//Step-2 :: Generate a screenshot as file

File rsc = tsc.getScreenshotAs(OutputType.FILE);

FileHandler.copy(rsc,new File("user.png"));

Thread.sleep(2000);

}

@Test

void ctestUserPassword() throws InterruptedException, IOException {

//Find Input Field

WebElement input = driver.findElement(By.cssSelector("body > form > fieldset > div:nth-child(1) > input[type=password]:nth-child(6)"));

assertTrue(input.isDisplayed());

assertTrue(input.isEnabled());

//Enter yourpassword

input.sendKeys("saranya611");

Thread.sleep(2000);

//Step-1 :: type cast driver object into takeScreenshot instance

TakesScreenshot tsc = (TakesScreenshot) driver;

//Step-2 :: Generate a screenshot as file

File rsc = tsc.getScreenshotAs(OutputType.FILE);

FileHandler.copy(rsc,new File("password.png"));

}

@Test

void dtestCheckBox() throws InterruptedException, IOException {

//Find Input Field

WebElement checkbox = driver.findElement(By.cssSelector("body > form > fieldset > div:nth-child(1) > label:nth-child(11) > input[type=checkbox]"));

assertEquals(true, checkbox.isDisplayed());

assertEquals(true, checkbox.isEnabled());

checkbox.click();

assertEquals(false, checkbox.isSelected());

Thread.sleep(2000);

//Step-1 :: type cast driver object into takeScreenshot instance

TakesScreenshot tsc = (TakesScreenshot) driver;

//Step-2 :: Generate a screenshot as file

File rsc = tsc.getScreenshotAs(OutputType.FILE);

FileHandler.copy(rsc,new File("checkBox.png"));

}

}