1.Write code for handling multiple windows

import java.util.Set;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class WindowHandling {

public static void main(String[] args) {

WebDriverManager.chromedriver().setup();

driver = new ChromeDriver();

driver.get("https://www.example.com");

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

String currentWindowHandle = driver.getWindowHandle();

WebElement newWindowLink = driver.findElement(By.id("new-window-link"));

newWindowLink.click();

Set<String> windowHandles = driver.getWindowHandles();

for (String windowHandle : windowHandles) {

if (!windowHandle.equals(currentWindowHandle)) {

driver.switchTo().window(windowHandle);

break;

}

}

// Do some action on the new window

WebElement newWindowHeading = driver.findElement(By.tagName("h1"));

System.out.println("The heading on the new window is: " + newWindowHeading.getText());

// Switch back to the original window

driver.switchTo().window(currentWindowHandle);

// Do some action on the original window

WebElement originalWindowHeading = driver.findElement(By.tagName("h1"));

System.out.println("The heading on the original window is: " + originalWindowHeading.getText());

driver.quit();

}

}

2. Write code for positive and negative scenarios for Alerts

import org.openqa.selenium.Alert;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class AlertHandling {

public static void main(String[] args) {

WebDriverManager.chromedriver().setup();

driver = new ChromeDriver();

driver.get("https://www.example.com");

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

WebElement alertButton = driver.findElement(By.id("alert-button"));

alertButton.click();

// Switch to the alert and accept it (positive scenario)

Alert alert = driver.switchTo().alert();

String alertText = alert.getText();

System.out.println("The alert message is: " + alertText);

alert.accept();

WebElement confirmButton = driver.findElement(By.id("confirm-button"));

confirmButton.click();

// Switch to the confirm alert and dismiss it (negative scenario)

Alert confirmAlert = driver.switchTo().alert();

String confirmAlertText = confirmAlert.getText();

System.out.println("The confirm alert message is: " + confirmAlertText);

confirmAlert.dismiss();

driver.quit();

}

}

3. Write code for Cross Browser Testing

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.openqa.selenium.remote.DesiredCapabilities;

import org.testng.annotations.AfterTest;

import org.testng.annotations.BeforeTest;

import org.testng.annotations.Parameters;

import org.testng.annotations.Test;

public class CrossBrowserTesting {

WebDriver driver;

String appURL = "https://www.example.com";

@BeforeTest

@Parameters("browser")

public void setup(String browserName) {

if (browserName.equalsIgnoreCase("chrome")) {

System.setProperty("webdriver.chrome.driver", "path/to/chromedriver");

driver = new ChromeDriver();

} else if (browserName.equalsIgnoreCase("firefox")) {

System.setProperty("webdriver.gecko.driver", "path/to/geckodriver");

DesiredCapabilities capabilities = DesiredCapabilities.firefox();

capabilities.setCapability("marionette", true);

driver = new FirefoxDriver(capabilities);

}

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

}

@Test

public void test() {

driver.get(appURL);

// Perform test steps here

}

@AfterTest

public void teardown() {

driver.quit();

}

}

XML file with parameters:

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

<suite name="CrossBrowserTesting Suite">

<test name="Chrome Test">

<parameter name="browser" value="chrome"/>

<classes>

<class name="CrossBrowserTesting"/>

</classes>

</test>

<test name="Firefox Test">

<parameter name="browser" value="firefox"/>

<classes>

<class name="CrossBrowserTesting"/>

</classes>

</test>

</suite>

4. Write code for handling Frames

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class FrameHandling {

public static void main(String[] args) {

WebDriverManager.chromedriver().setup();

driver = new ChromeDriver();

driver.get("https://www.example.com");

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

// Switch to frame by index

driver.switchTo().frame(0);

// Perform actions in the frame

WebElement element = driver.findElement(By.id("frame\_element"));

element.click();

// Switch back to default content

driver.switchTo().defaultContent();

// Switch to frame by name or ID

driver.switchTo().frame("frame\_name\_or\_id");

// Perform actions in the frame

WebElement element2 = driver.findElement(By.id("frame\_element2"));

element2.click();

// Switch back to default content

driver.switchTo().defaultContent();

driver.quit();

}

}

5. Automate Menu and Sub Menu and click on a link in Sub Menu and navigate to the page and click on an element

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class MenuAutomation {

public static void main(String[] args) {

WebDriverManager.chromedriver().setup();

driver = new ChromeDriver();

driver.get("https://www.example.com");

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

// Find the main menu item and click it

WebElement mainMenuItem = driver.findElement(By.id("main\_menu\_item"));

mainMenuItem.click();

// Find the sub-menu item and click it

WebElement subMenuItem = driver.findElement(By.id("sub\_menu\_item"));

subMenuItem.click();

// Wait for the page to load

try {

Thread.sleep(3000);

} catch (InterruptedException e) {

e.printStackTrace();

}

// Find an element on the page and click it

WebElement pageElement = driver.findElement(By.id("page\_element"));

pageElement.click();

driver.quit();

}

}

6. Select multiple options from the dropdown

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.support.ui.Select;

public class MultiSelectDropdown {

public static void main(String[] args) {

WebDriverManager.chromedriver().setup();

driver = new ChromeDriver();

driver.get("https://www.example.com");

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

WebElement dropdown = driver.findElement(By.id("multi\_select\_dropdown"));

Select select = new Select(dropdown);

select.selectByVisibleText("Option 1");

select.selectByVisibleText("Option 3");

select.selectByVisibleText("Option 5");

for (WebElement option : select.getAllSelectedOptions()) {

System.out.println(option.getText());

}

driver.quit();

}

}

7. How to do file upload in Selenium?

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class FileUpload {

public static void main(String[] args) {

System.setProperty("webdriver.chrome.driver", "path/to/chromedriver");

WebDriverManager.chromedriver().setup();

driver = new ChromeDriver();

driver.get("https://www.example.com");

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

WebElement fileInput = driver.findElement(By.id("file\_input"));

fileInput.sendKeys("path/to/file");

WebElement submitButton = driver.findElement(By.id("submit\_button"));

submitButton.click();

driver.quit();

}

}

8. Write code for taking Screenshot

import java.io.File;

import org.openqa.selenium.OutputType;

import org.openqa.selenium.TakesScreenshot;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class ScreenshotExample {

public static void main(String[] args) {

WebDriverManager.chromedriver().setup();

driver = new ChromeDriver();

driver.get("http://leaftaps.com/opentaps/control/main");

driver.manage().window().maximize(); File screenshot = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);

String screenshotPath = "path/to/screenshot.png";

try {

FileUtils.copyFile(screenshot, new File(screenshotPath));

System.out.println("Screenshot saved: " + screenshotPath);

} catch (IOException e) {

System.out.println("Failed to save screenshot: " + e.getMessage());

}

driver.quit();

}

}

9. Read data from Excel and given that data as input for login and password and click on submit and validate the popup which says Login is successful

import java.io.FileInputStream;

import java.io.IOException;

import java.util.concurrent.TimeUnit;

import org.apache.poi.ss.usermodel.CellType;

import org.apache.poi.xssf.usermodel.XSSFCell;

import org.apache.poi.xssf.usermodel.XSSFRow;

import org.apache.poi.xssf.usermodel.XSSFSheet;

import org.apache.poi.xssf.usermodel.XSSFWorkbook;

import org.openqa.selenium.Alert;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class ExcelLogin {

public static void main(String[] args) throws IOException {

System.setProperty("webdriver.chrome.driver", "path/to/chromedriver");

WebDriver driver = new ChromeDriver();

driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

driver.get("https://www.example.com/login");

// Read data from Excel file

String filePath = "path/to/data.xlsx";

FileInputStream fis = new FileInputStream(filePath);

XSSFWorkbook workbook = new XSSFWorkbook(fis);

XSSFSheet sheet = workbook.getSheet("Sheet1");

// Get login and password data from Excel file

XSSFRow row = sheet.getRow(1);

XSSFCell cell = row.getCell(0);

cell.setCellType(CellType.STRING);

String login = cell.getStringCellValue();

cell = row.getCell(1);

cell.setCellType(CellType.STRING);

String password = cell.getStringCellValue();

// Enter login and password data and click submit button

driver.findElement(By.id("login\_input")).sendKeys(login);

driver.findElement(By.id("password\_input")).sendKeys(password);

driver.findElement(By.id("submit\_button")).click();

// Validate popup message that says login is successful

Alert alert = driver.switchTo().alert();

String alertText = alert.getText();

if (alertText.equals("Login successful!")) {

System.out.println("Login successful!");

} else {

System.out.println("Login failed: " + alertText);

}

alert.accept();

driver.quit();

}

}

10. Write test scenario for Credit Cards

Test scenarios for Credit Cards:

1.Verify that the user can add a new credit card:

Open the Credit Card page.

Click on the "Add New Card" button.

Enter valid credit card details in the form.

Click on the "Save" button.

Verify that the credit card is successfully added to the user's account.

2.Verify that the user can edit an existing credit card:

Open the Credit Card page.

Click on the "Edit" button next to an existing credit card.

Modify the credit card details.

Click on the "Save" button.

Verify that the credit card is successfully updated in the user's account.

3.Verify that the user can delete an existing credit card:

Open the Credit Card page.

Click on the "Delete" button next to an existing credit card.

Confirm the deletion.

Verify that the credit card is successfully deleted from the user's account.

4.Verify that the user cannot add a credit card with invalid details:

Open the Credit Card page.

Click on the "Add New Card" button.

Enter invalid credit card details in the form.

Click on the "Save" button.

Verify that the system displays an error message indicating the invalid details.

5.Verify that the user cannot add more than the allowed number of credit cards:

Open the Credit Card page.

Verify the number of credit cards currently added to the user's account.

If the maximum number of credit cards is already added, try to add another one.

Verify that the system displays an error message indicating that the user has reached the maximum number of credit cards.

6.Verify that the user can view the details of an existing credit card:

Open the Credit Card page.

Click on the "View" button next to an existing credit card.

Verify that the details of the credit card are displayed correctly.

7.Verify that the user can select a default credit card:

Open the Credit Card page.

Click on the "Set as Default" button next to an existing credit card.

Verify that the selected credit card is now set as the default one.

8.Verify that the user can make a payment using a saved credit card:

Navigate to the Payment page.

Select a product or service to purchase.

Select a saved credit card as the payment method.

Enter the necessary payment details.

Click on the "Submit Payment" button.

Verify that the payment is successful and the user receives a confirmation message.

11. How to find more than one web element in the list?

We can find more than one web element by using findElements method in selenium webdriver.

12. How to perform upload files using selenium?

To perform upload files we can use the sendKeys method.

WebElement fileInput = driver.findElement(By.id("fileInput"));

fileInput.sendKeys("C:\\path\\to\\file.txt");

13. How to shift between tabs of the same browser using selenium?

To switch between tabs in the same browser we can use switchTo method in selenium.

// switch to the first tab (index 0)

driver.switchTo().window(driver.getWindowHandles().toArray()[0]);

// switch to the second tab (index 1)

driver.switchTo().window(driver.getWindowHandles().toArray()[1]);

After using switchTo method for switching tabs use getTitle method to find the title of the page so that you can know the current tab.