

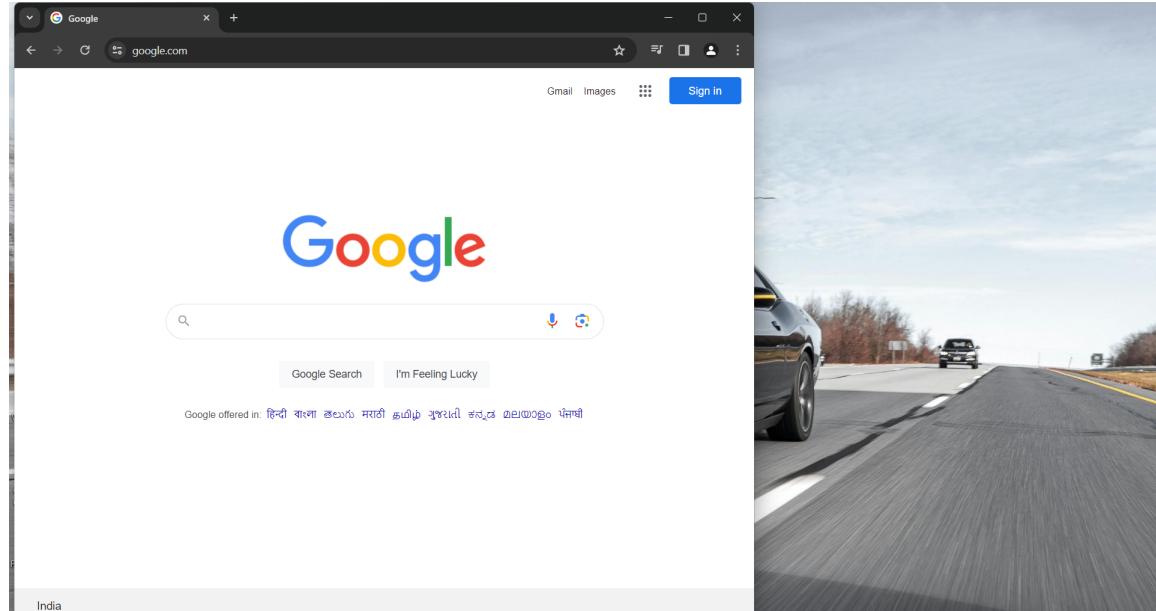
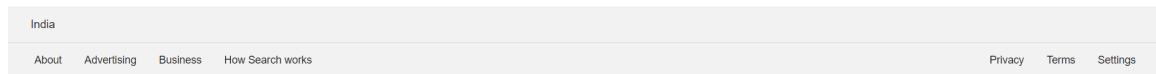
# **CSSL4 Automation Testing**

**Output:**

Main Window Handle: 32EA995F5BDA21B77CEBBA2D076542B6

Position of the Window: (0, 0)

Size of the Window: (1536, 864)



# 1 Write a Selenium Java Program to demonstrate window operations.

```
package Record;

import org.openqa.selenium.Point;
import org.openqa.selenium.Dimension;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class Lab1
{
    public static void main(String[] args)
    {
        WebDriver driver = new ChromeDriver(); // to initialize a new ChromeDriver instance

        driver.get("https://www.google.com"); // to open a website

        driver.manage().window().minimize(); // to minimize the window

        driver.manage().window().maximize(); // to maximize the window

        driver.manage().window().fullscreen(); // to make the window fullscreen

        System.out.println("Main Window Handle: " + driver.getWindowHandle());

        System.out.println("Position of the Window: " +
                           → driver.manage().window().getPosition());

        System.out.println("Size of the Window: " + driver.manage().window().getSize());

        Point p = new Point(0,0); // to create an object of Point class
        driver.manage().window().setPosition(p);

        Dimension d = new Dimension(1000,1000); // to create an object of Dimension class
        driver.manage().window().setSize(d);

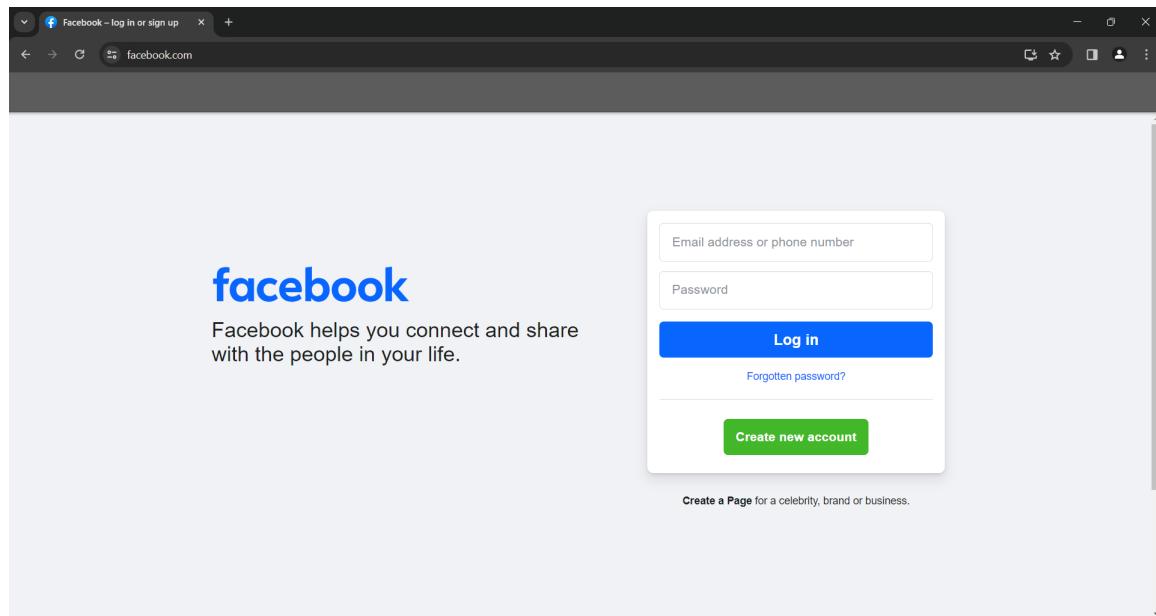
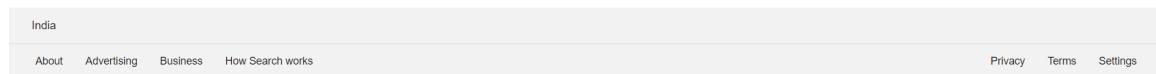
        driver.quit(); // to close the WebDriver session
    }
}
```

**Output:**

Title: Facebook – log in or sign up  
Current URL: <https://www.facebook.com/>



Google offered in: हिन्दी বাংলা താംസ് മരാറ്റി തമിൽ മുഖ്യാലീ കന്നಡ മലയാളം ഘോർജ്ജ്



## 2 Write a Selenium Java Program to demonstrate navigation.

```
package Record;

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

public class Lab2
{
    public static void main(String[] args)
    {
        WebDriver driver = new ChromeDriver();

        driver.manage().window().maximize(); // maximizes the window

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

        driver.navigate().to("https://www.facebook.com"); // navigating to a new website

        driver.navigate().back(); // navigating back to the previous website

        driver.navigate().forward(); // navigating forward to a website

        driver.navigate().refresh(); // refreshing the website

        System.out.println("Title: " + driver.getTitle());

        System.out.println("Current URL: " + driver.getCurrentUrl());

        driver.quit();
    }
}
```

**Output:**

Title: Selenium Grid Online — Run Selenium Test On Cloud

Current URL: <https://www.lambdatest.com/selenium-playground/input-form-demo>

tayyab@gmail.com

Tayyab

The screenshot shows a web browser window titled "Selenium Grid Online | Run Selenium Test On Cloud". The URL in the address bar is "lambdatest.com/selenium-playground/input-form-demo". The page has a header with the Lambdatest logo and navigation links for Platform, Enterprise, Resources, Developers, and Pricing. There are also "Login", "Book a Demo", and "Get Started Free" buttons. The main content area is titled "Form Demo" and contains a form titled "Input form validations". The form includes fields for Name\*, Email\*, Password\*, Company\*, Website\*, City\*, Country\*, Address\*, and Address 2\*. All fields have placeholder text and are marked with asterisks indicating they are required. A "Feedback" button is located in the bottom right corner of the form area.

The screenshot shows a web browser window titled "Native App Testing on Real iPhone 15". The URL in the address bar is "lambdatest.com/test-on-iphone-15". The page has a header with the Lambdatest logo and navigation links for Platform, Enterprise, Resources, Developers, and Pricing. There are also "Login", "Book a Demo", and "Get Started Free" buttons. The main content area features a large heading "Start Testing Your Apps On iPhone 15 Series". Below it, a sub-headline says "Test your native mobile applications, web apps and website on the Real iPhone 15 online; iPhone 15, Plus, Pro and Pro Max.". There is a "Real Device" section with a "Start" button highlighted with a teal circle. To the right, there is a list of iPhone models with their screen sizes and pixel densities. At the bottom, there are "Start free with Google" and "Start free with Email" buttons, along with a note about being trusted by 2M+ users and logos for Microsoft, Vimeo, NVIDIA, Telstra, and rubrik.

### 3 Write a Selenium Java Program to demonstrate driver operations.

```
package Record;

import java.time.Duration;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;

public class Lab3
{
    public static void main(String[] args)
    {
        WebDriver driver = new ChromeDriver();

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

        driver.get("https://www.lambdatest.com/selenium-playground/input-form-demo");

        driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(1));

        System.out.println("Title: " + driver.getTitle());

        System.out.println("Current URL: " + driver.getCurrentUrl());

        WebElement elementById = driver.findElement(By.id("inputEmail4"));
        elementById.sendKeys("tayyab@gmail.com");
        System.out.println(elementById.getAttribute("value"));

        WebElement elementByName = driver.findElement(By.name("name"));
        elementByName.sendKeys("Tayyab");
        System.out.println(elementByName.getAttribute("value"));

        driver.navigate().to("https://www.lambdatest.com/test-on-iphone-15");

        driver.navigate().back();

        driver.quit();
    }
}
```

**Output:**

Title:Selenium Grid Online — Run Selenium Test On Cloud

Selected option in dropdown1: India

Selected option in dropdown2: AUSTRALIA

The screenshot shows a web browser window titled "Selenium Grid Online | Run Selenium Test On Cloud". The URL is "lambdatest.com/selenium-playground/input-form-demo". The page has a header with the LAMBDATEST logo and navigation links for Platform, Enterprise, Resources, Developers, Pricing, Login, Book a Demo, and Get Started Free. The main content is titled "Form Demo" and "Input form validations". It contains several input fields: Name\*, Email\*, Password\*, Company\*, Website\*, Country\* (dropdown menu showing "India"), City\*, Address\*, Address 2, State\*, Zip Code\*, and a file input field. A "CLICK HERE" button is also present.

The screenshot shows a web browser window titled "Register: Mercury Tours" with the URL "demo.guru99.com/test/newtours/register.php". The page features a sidebar with a "SAVINGS! Rent A Car" banner and a "CLICK HERE" button. The main form is divided into sections: "Mailing Information" (Address, City, State/Province, Postal Code, Country dropdown set to "ANTARCTICA") and "User Information" (User Name, Password, Confirm Password). A "Submit" button is at the bottom. The footer copyright notice is "© 2005, Mercury Interactive (v. 011003-1.01-058)".

This screenshot is identical to the previous one, showing the "Register: Mercury Tours" page with the URL "demo.guru99.com/test/newtours/register.php". The "Country" field in the "Mailing Information" section is now set to "AUSTRALIA". The rest of the form and the footer remain the same.

#### 4 Write a Selenium Java Program to demonstrate dropdown or list.

```
package Record;

import java.time.Duration;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.ui.Select;
import org.openqa.selenium.chrome.ChromeDriver;

public class Lab4
{
    public static void main(String[] args) throws InterruptedException
    {
        WebDriver driver = new ChromeDriver();

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

        driver.get("https://www.lambdatest.com/selenium-playground/input-form-demo");

        System.out.println("Title: " + driver.getTitle());

        WebElement dropdownList1 = driver.findElement(By.name("country"));

        Select dropdown1 = new Select(dropdownList1);
        dropdown1.selectByVisibleText("India"); // selects the value in the dropdown menu
        System.out.println("Selected option in dropdown1: " +
                           dropdown1.getFirstSelectedOption().getText());

        driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(5));

        driver.navigate().to("https://demo.guru99.com/test/newtours/register.php");

        WebElement dropdownList2 = driver.findElement(By.name("country"));

        Select dropdown2 = new Select(dropdownList2);
        dropdown2.selectByVisibleText("ANTARCTICA");
        Thread.sleep(5000);
        dropdown2.selectByIndex(12); // selects the value in the dropdown menu using the index
        System.out.println("Selected option in dropdown2: " +
                           dropdown2.getFirstSelectedOption().getText());

        driver.quit();
    }
}
```

**Output:**

This page is used for practicing iframes in web automation. We have three types of frames in web applications, those are:

1. frame - The **<frame>** tag was used in HTML 4 to define one particular window (frame) within a **<frameset>**.
2. frameset - The **<frameset>** tag was used in HTML 4 to define a frameset.
3. iframe - The **<iframe>** tag specifies an inline frame. An inline frame is used to embed another document within the current HTML document.

Out of these, frame and frameset were no longer being supported in HTML 5. Below we have three frames to practice.

This textbox is used for entering the text after switching back from frames.

main frame

Frame1:

Menu f t i n b

In HTML we have dropdown controls we use very often. There are two types of dropdown available.

1. Single-Selection Dropdown
2. Multi-Selection Dropdown

This page can be used for practicing the above controls using selenium.

Course Name: Java

Frame2:

Menu f t i n b

**HYR Tutorials**

This page is used for practicing iframes in web automation. We have three types of frames in web applications, those are:

1. frame - The **<frame>** tag was used in HTML 4 to define one particular window (frame) within a **<frameset>**.
2. frameset - The **<frameset>** tag was used in HTML 4 to define a frameset.
3. iframe - The **<iframe>** tag specifies an inline frame. An inline frame is used to embed another document within the current HTML document.

Out of these, frame and frameset were no longer being supported in HTML 5. Below we have three frames to practice.

This textbox is used for entering the text after switching back from frames.

returned to main webpage

Frame1:

Menu f t i n b

**HYR Tutorials**

## 5 Write a Selenium Java Program to demonstrate frame and iframe.

```
package Record;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.ui.Select;
import org.openqa.selenium.chrome.ChromeDriver;

public class Lab5
{
    public static void main(String[] args) throws InterruptedException
    {
        WebDriver driver = new ChromeDriver();

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

        driver.get("https://www.hyrtutorials.com/p/frames-practice.html");

        WebElement mainIframe = driver.findElement(By.id("name"));
        mainIframe.sendKeys("main frame"); // Entering text in the main frame

        Thread.sleep(2000);

        driver.switchTo().frame("frm1"); // switching to frame1

        WebElement course = driver.findElement(By.id("course")); // Selecting the dropdown menu
        Select courseName = new Select(course);
        courseName.selectByVisibleText("Java");

        Thread.sleep(2000);

        driver.switchTo().defaultContent(); // redirecting back to the main frame

        driver.findElement(By.id("name")).clear(); // cleared the main frame text
        driver.findElement(By.id("name")).sendKeys("returned to main webpage");

        driver.quit();
    }
}
```

## Output:

Alert Text: I am an alert box!

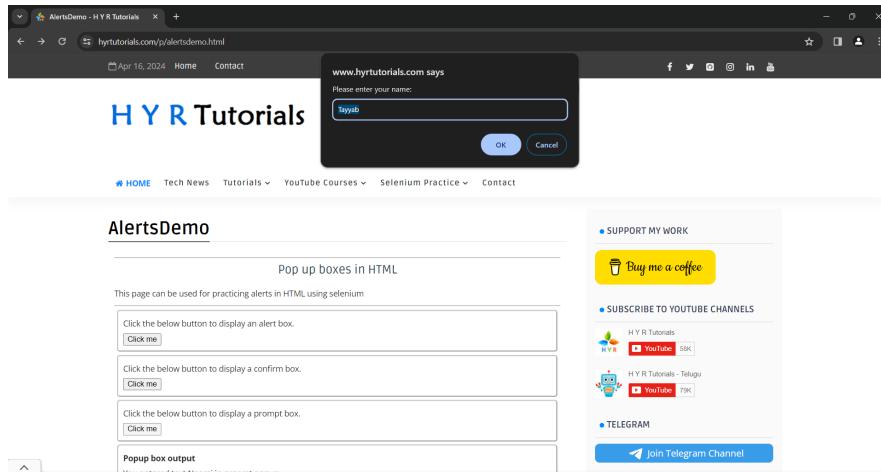
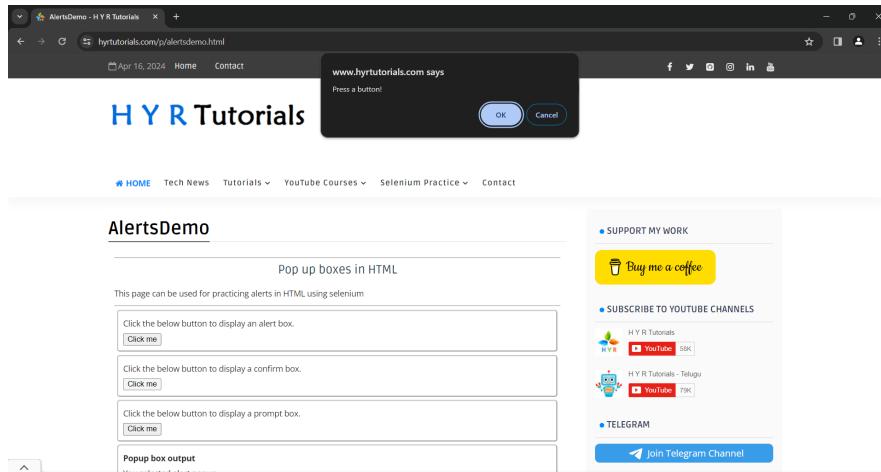
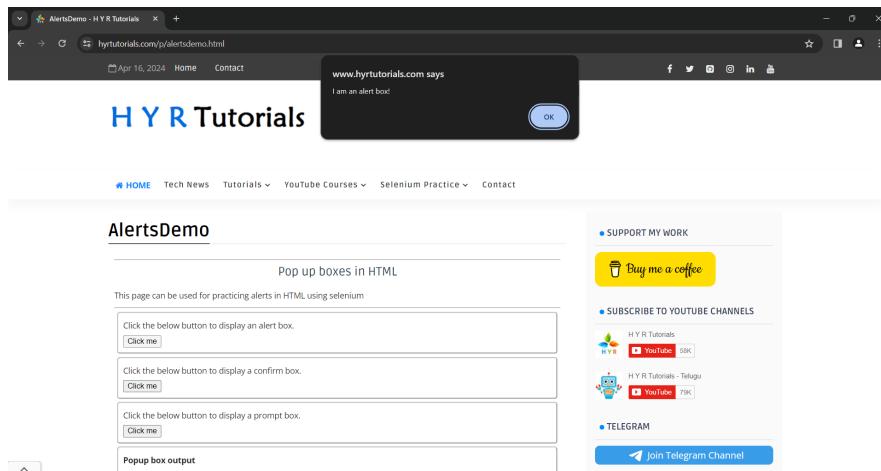
Alert Text output: You selected alert popup

Confirm Text: Press a button!

Confirm Text output: You pressed OK in confirmation popup

Prompt Request Text: Please enter your name:

Prompt Text output: You entered text Tayyab in propmt popup



## 6 Write a Selenium Java Program to demonstrate alert.

```

package Record;

import org.openqa.selenium.By;
import org.openqa.selenium.Alert;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class Lab6
{
    public static void main(String[] args) throws InterruptedException
    {
        WebDriver driver = new ChromeDriver();

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

        driver.get("https://www.hyrtutorials.com/p/alertsdemo.html");

        // Alert Box:
        driver.findElement(By.id("alertBox")).click(); // Click on the button that triggers the
        // alert dialog
        Alert alert = driver.switchTo().alert(); // Switch to the alert dialog
        String alertText = alert.getText(); // Get the text from the alert dialog
        System.out.println("Alert Text: " + alertText);
        Thread.sleep(3000);
        alert.accept(); // Accept the alert (Clicking OK)
        System.out.println("Alert Text output: " +
        // driver.findElement(By.id("output")).getText() + "\n");
        Thread.sleep(3000);

        // Confirmation Box:
        driver.findElement(By.id("confirmBox")).click(); // Click on the button that triggers
        // the confirm dialog
        Alert confirm = driver.switchTo().alert(); // Switch to the conform dialog
        String confirmText = confirm.getText(); // Get the text from the conform dialog
        System.out.println("Confirm Text: " + confirmText);
        Thread.sleep(3000);
        confirm.accept(); // Accept the alert (Clicking OK)
        System.out.println("Confirm Text output: " +
        // driver.findElement(By.id("output")).getText() + "\n");
        Thread.sleep(3000);

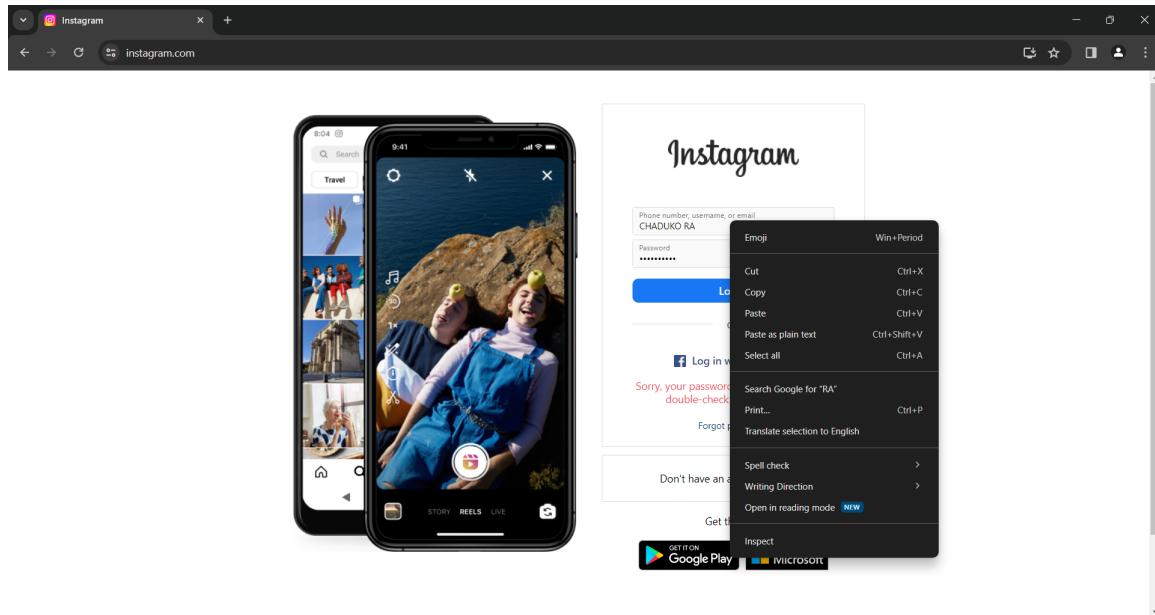
        // Prompt Box:
        driver.findElement(By.id("promptBox")).click(); // Click on the button that triggers
        // the prompt dialog
        Alert prompt = driver.switchTo().alert(); // Switch to the prompt dialog
        String promptText = prompt.getText(); // Get the text from the prompt dialog
        System.out.println("Prompt Request Text: " + promptText);
        prompt.sendKeys("Tayyab");
        Thread.sleep(3000);
        prompt.accept(); // Accept the alert (Clicking OK)
        System.out.println("Prompt Text output: " +
        // driver.findElement(By.id("output")).getText() + "\n");
        Thread.sleep(3000);

        driver.quit();
    }
}

```

**Output:**

Title: Instagram



## 7 Write a Selenium Java Program to demonstrate actions(Mouse and Keyboard).

```
package Record;

import java.time.Duration;
import org.openqa.selenium.By;
import org.openqa.selenium.Keys;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.interactions.Action;
import org.openqa.selenium.interactions.Actions;

public class Lab7
{
    public static void main(String[] args)
    {
        WebDriver driver = new ChromeDriver();

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

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

        System.out.println("Title: " + driver.getTitle());

        driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(5));

        WebElement username = driver.findElement(By.name("username"));
        WebElement password = driver.findElement(By.name("password"));
        WebElement submit = driver.findElement(By.xpath("//button[@type='submit']"));

        Actions builder = new Actions(driver);
        Action action = builder.moveToElement(username) // mouse move over to the username
        ↪ field
            .click()
            .keyDown(username, Keys.SHIFT)
            .sendKeys(username, "chaduko ra")
            .keyUp(username, Keys.SHIFT)
            .moveToElement(password) // mouse move over to the password field
            .click()
            .keyDown(password, Keys.SHIFT)
            .sendKeys(password, "chaduko ra")
            .keyUp(Keys.SHIFT)
            .doubleClick(username)
            .contextClick()
            .build();

        action.perform();

        submit.click();

        driver.quit();
    }
}
```

**Output:**

LinkedIn

Articles People Learning Jobs Get the app Join now Sign in

# Welcome to your professional community

Email or phone

Password

 Show  
Forgot password?

Sign in

or

By clicking Continue, you agree to LinkedIn's [User Agreement](#), [Privacy Policy](#), and [Cookie Policy](#).



## 8 Write a Selenium Java Program to demonstrate screenshot.

```
package Record;

import java.io.File;
import java.io.IOException;
import org.apache.commons.io.FileUtils;
import org.openqa.selenium.TakesScreenshot;
import org.openqa.selenium.OutputType;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class Lab8
{
    public static void main(String[] args) throws InterruptedException
    {
        WebDriver driver = new ChromeDriver();

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

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

        Thread.sleep(5000);

        TakesScreenshot tk = (TakesScreenshot) driver;

        File srcfle = tk.getScreenshotAs(OutputType.FILE);
        File Destfle = new File("E:/screenshot.png"); // Stores the screenshot in E drive as
        ↵   screenshot.png
        // You can also change the path to either C drive, D drive or to downloads

        try
        {
            FileUtils.copyFile(srcfle,Destfle);
        }
        catch(IOException e)
        {
            System.out.println(e.getMessage());
        }
    }
}
```

**Output:**

Starting the Browers to Launch

The Actual Title is: W3Schools Online Web Tutorials

The Launch Completed

PASSED: testAssert

=====

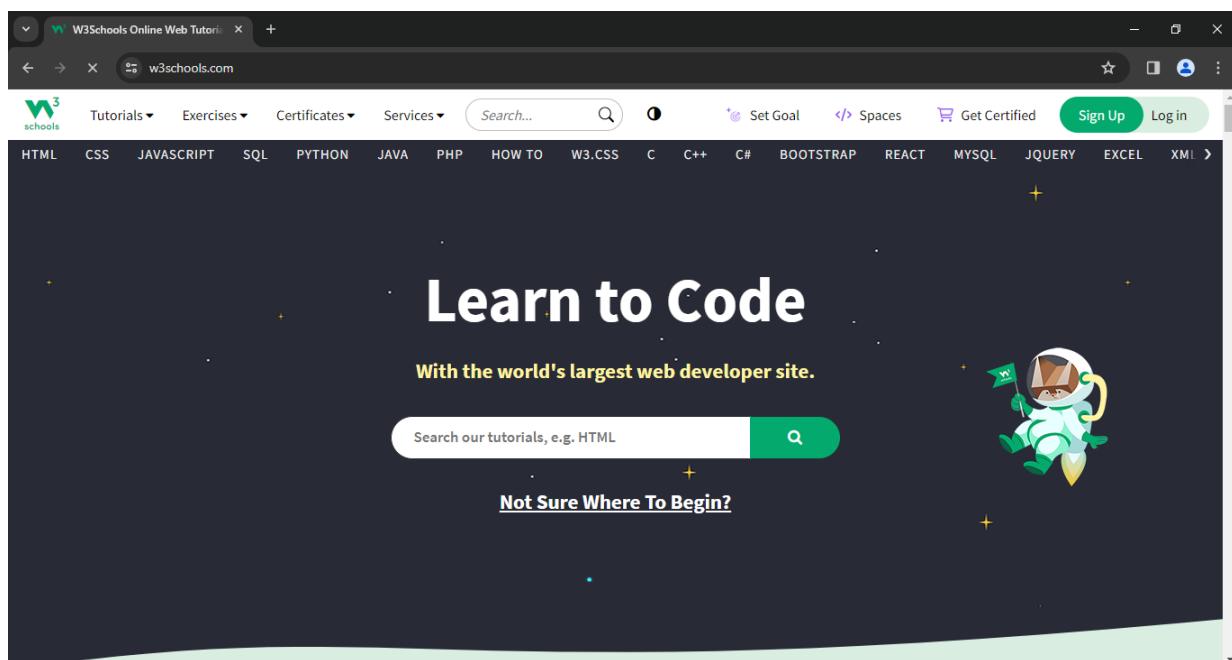
Default test

Tests run: 1, Failures: 0, Skips: 0

=====

Default suite

Total tests run: 1, Passes: 1, Failures: 0, Skips: 0



## 9 Write a Selenium Java Program to demonstrate testing, assertions and annotations.

### Program 9a: Assertions

```
package Record;

import org.testng.Assert;
import org.testng.annotations.Test;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.AfterMethod;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class Lab9a
{
    WebDriver driver = new ChromeDriver();

    @BeforeMethod
    public void Before()
    {
        System.out.println("Starting the Browers to Launch");
    }

    @Test
    public void testAssert()
    {
        driver.get("https://www.w3schools.com/");

        String ActualTitle = driver.getTitle();
        String ExpectedTitle = "Most Reliable App & Cross Browser Testing Platform / 
        ↳ browserstack";

        Assert.assertNotEquals(ActualTitle, ExpectedTitle);
        System.out.println("The Actual Title is: " + ActualTitle);
    }

    @AfterMethod
    public void After()
    {
        driver.quit();
        System.out.println("The Launch Completed");
    }
}
```

Output:

Test Case Before Suite  
Test Case Before Test  
Test Case Before Class  
Test Case Before Method  
Tayyab Executing Test Case 1  
Test Case After Method  
Test Case Before Method  
Tayyab Executing Test Case 2  
Test Case After Method  
Test Case After Class  
Test Case After Test

PASSED: test2  
PASSED: test1

=====

Default test  
Tests run: 2, Failures: 0, Skips: 0

=====

Test Case After Suite

=====

Default suite  
Total tests run: 2, Passes: 2, Failures: 0, Skips: 0

=====

**Program 9b: Annotations**

```
package Record;

import org.testng.annotations.Test;
import org.testng.annotations.BeforeSuite;
import org.testng.annotations.AfterSuite;
import org.testng.annotations.BeforeTest;
import org.testng.annotations.AfterTest;
import org.testng.annotations.BeforeClass;
import org.testng.annotations.AfterClass;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.AfterMethod;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class Lab9b
{
    WebDriver driver = new ChromeDriver();

    @BeforeSuite
    public void beforeSuite(){
        System.out.println("Test Case Before Suite");
    }

    @BeforeTest
    public void beforeTest(){
        System.out.println("Test Case Before Test");
    }

    @BeforeClass
    public void beforeClass(){
        System.out.println("Test Case Before Class");
    }

    @BeforeMethod
    public void beforeMethod(){
        System.out.println("Test Case Before Method");
    }

    @Test
    public void test1(){
        System.out.println("Tayyab Executing Test Case 1");
    }

    @AfterMethod
    public void afterMethod(){
        System.out.println("Test Case After Method");
    }

    @Test
    public void test2(){
        System.out.println("Tayyab Executing Test Case 2");
    }

    @AfterClass
    public void afterClass(){
        System.out.println("Test Case After Class");
    }

    @AfterTest
    public void afterTest(){
        System.out.println("Test Case After Test");
    }

    @AfterSuite
    public void afterSuite(){
        System.out.println("Test Case After Suite");
    }
}
```

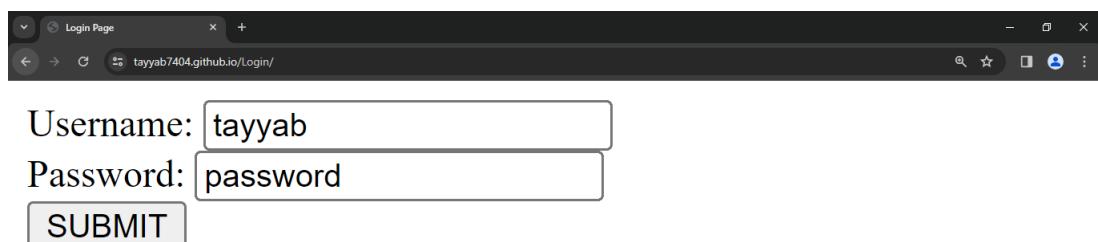
**Output:**

```
[tayyab, password]
[neeraj, chaduko bro]
```

```
PASSED: testLogin("tayyab", "password")
PASSED: testLogin("neeraj", "chaduko bro")
```

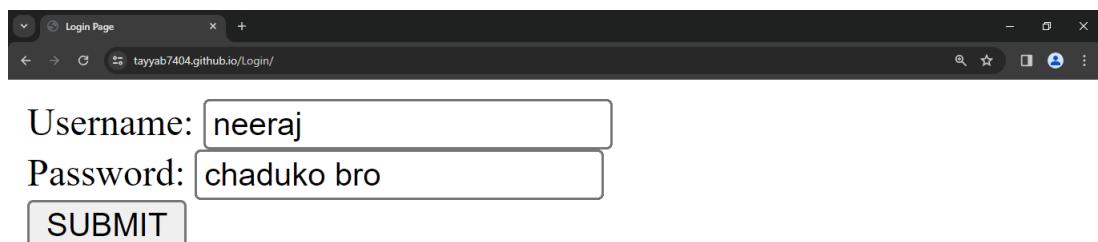
```
=====
Default test
Tests run: 1, Failures: 0, Skips: 0
=====
```

```
=====
Default suite
Total tests run: 2, Passes: 2, Failures: 0, Skips: 0
=====
```



Username: tayyab  
Password: password  
SUBMIT

## Login Successful!



Username: neeraj  
Password: chaduko bro  
SUBMIT

## Login Successful!

## 10 Write a Selenium Java Program to demonstrate data driven test.

```
package Record;

import java.io.File;
import java.io.FileInputStream;
import java.util.Arrays;
import org.apache.poi.ss.usermodel.DataFormatter;
import org.apache.poi.xssf.usermodel.XSSFSheet;
import org.apache.poi.xssf.usermodel.XSSFWorkbook;
import org.testng.annotations.DataProvider;

public class ExcelDataProvider
{
    @DataProvider(name="loginData")
    public String[][] getData() throws Exception
    {
        File excelFile = new File("./src/test/resources/testData.xlsx");

        FileInputStream fis = new FileInputStream(excelFile);
        XSSFWorkbook workbook = new XSSFWorkbook(fis);

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

        int rows = sheet.getPhysicalNumberOfRows();
        int columns = sheet.getRow(0).getLastCellNum();

        String[][] data = new String[rows-1] [columns];

        for (int i = 0; i < rows-1; i++)
        {
            for (int j = 0; j < columns; j++)
            {
                DataFormatter df = new DataFormatter();
                data[i][j] = df.formatCellValue(sheet.getRow(i+1).getCell(j));
            }
        }

        workbook.close();
        fis.close();

        for (String[] dataArr : data)
        {
            System.out.println(Arrays.toString(dataArr));
        }

        return data;
    }
}
```



```
package Record;

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.Test;

public class Lab10
{
    @Test(dataProvider = "loginData", dataProviderClass = ExcelDataProvider.class)

    public void testLogin(String username, String password) throws Exception
    {
        WebDriver driver = new ChromeDriver();

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

        driver.get("https://tayyab7404.github.io/Login/");

        WebElement usernameField = driver.findElement(By.id("username"));
        WebElement passwordField = driver.findElement(By.id("password"));
        WebElement submitButton = driver.findElement(By.id("submit"));

        usernameField.sendKeys(username);
        passwordField.sendKeys(password);
        submitButton.click();

        Thread.sleep(5000);

        driver.quit();
    }
}
```