

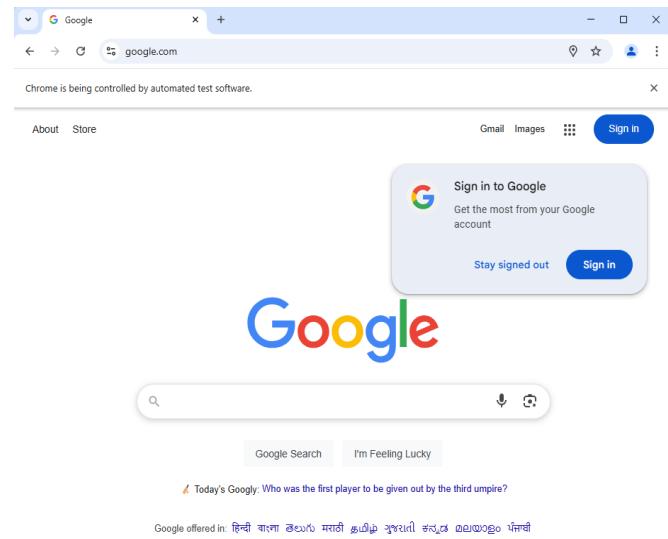
## 1. Set up Selenium WebDriver with Java: Install JDK, IDE, and configure Selenium.

Code :

```
package Programs;

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

public class Program {
    public static void main(String[] args) {
        System.setProperty("webdriver.chrome.driver", "D:\\programs\\chromedriver-win64\\chromedriver.exe"); // Updated path
        WebDriver cd = new ChromeDriver();
        cd.get("http://www.google.com");
    }
}
```

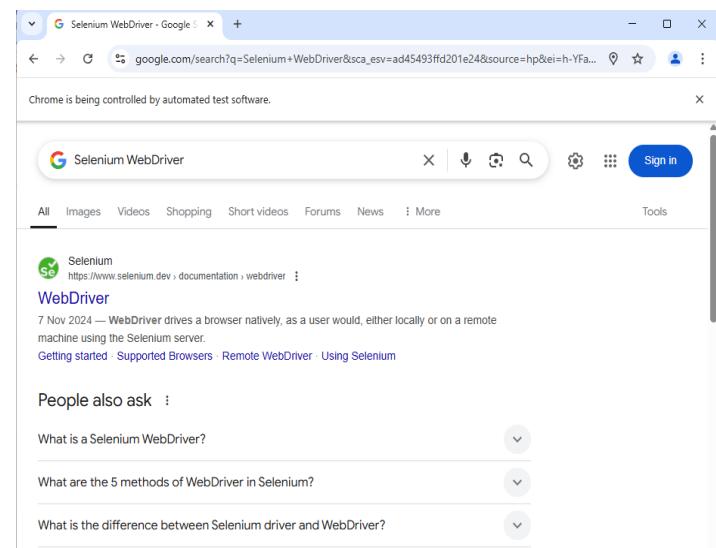


2 Write a Java Selenium script that to automate for Google Search Functionality: Write a script to search for a term on Google, handle all webdriver methods.

Code :

```
package Programs;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
public class GoogleSearch{
    public static void main(String[] args) {
        System.setProperty("webdriver.chrome.driver", "D:\\programs\\chromedriver-win64\\chromedriver.exe");
        WebDriver cd = new ChromeDriver();
        cd.manage().window().maximize();
        cd.get("https://www.google.com");
        WebElement searchBox =
        cd.findElement(By.name("q"));
        searchBox.sendKeys("Selenium
        WebDriver");
        searchBox.submit();
        try {
            Thread.sleep(3000); // wait 3
seconds
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
        System.out.println("Page Title: " +
cd.getTitle());
        // Don't close the browser
        immediately
        // cd.quit(); // <-- Commented out
    }
}//WhatUP MCA
```



3 Write a Java Selenium script that to automate for Handling Web Elements:to Automate web methods and web operational methods.

Code :

```
package Programs; // === Web Operational Methods ===

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

public class Elements {
    public static void main(String[] args) {
        // Set path to ChromeDriver
        System.setProperty("webdriver.chrome.driver", "D:\\programs\\chromedriver-win64\\chromedriver.exe");
        // Create WebDriver instance
        WebDriver driver = new ChromeDriver();
        // Maximize browser window
        driver.manage().window().maximize();
        // === Web Element Methods ===
        // 1. Find and fill the Full Name input field
        WebElement fullName =
            driver.findElement(By.id("userName"));
        fullName.sendKeys("Kamlesh
Bobade");
        // 2. Email field
        WebElement email =
            driver.findElement(By.id("userEmail"));
        email.sendKeys("kamlesh@example.com");
        // 3. Current Address field
```

```

        WebElement currentAddress =
        driver.findElement(By.id("currentAddress"
));
        currentAddress.sendKeys("Pune,
Maharashtra");

// 4. Permanent Address field

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

permanentAddress.sendKeys("Aurangabad, Maharashtra");

// 5. Check visibility

if (fullName.isDisplayed() &&
fullName.isEnabled()) {

    System.out.println("Full Name field
is visible and enabled.");
}

// 6. Click the Submit button

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

submitButton.click();

// Pause for visual
try {
    Thread.sleep(3000);
} catch (InterruptedException e) {
    e.printStackTrace();
}
}

// Wait & Navigate example

driver.navigate().to("https://www.google.
com");
System.out.println("Navigated to
Google");

// Wait & Navigate back

try {
    Thread.sleep(2000);
} catch (InterruptedException e) {
    e.printStackTrace();
}

// Wait & Navigate forward

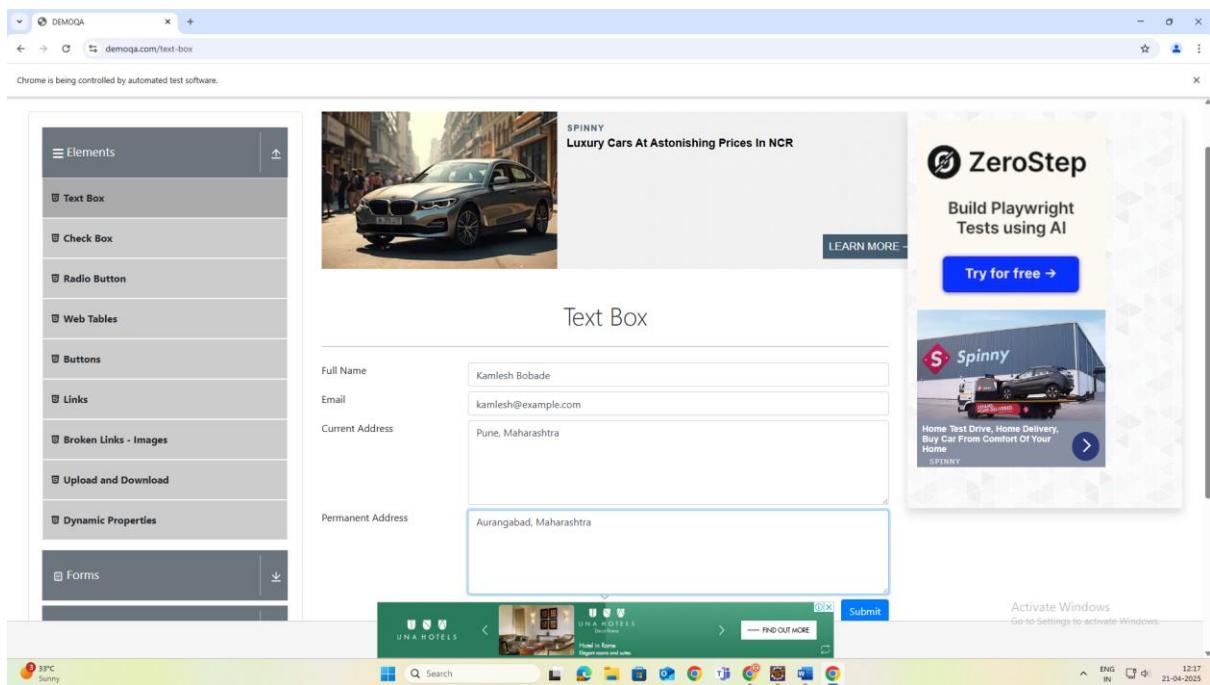
driver.navigate().back();
System.out.println("Navigated Back");

// Refresh

driver.navigate().refresh();
System.out.println("Page
Refreshed");

```

```
// Close the browser }  
driver.quit(); }
```



4 Write a Java Selenium script that to automate for Handling Web Elements:to Automate web Locators using css selectors methods.

Code :

```
package Programs;

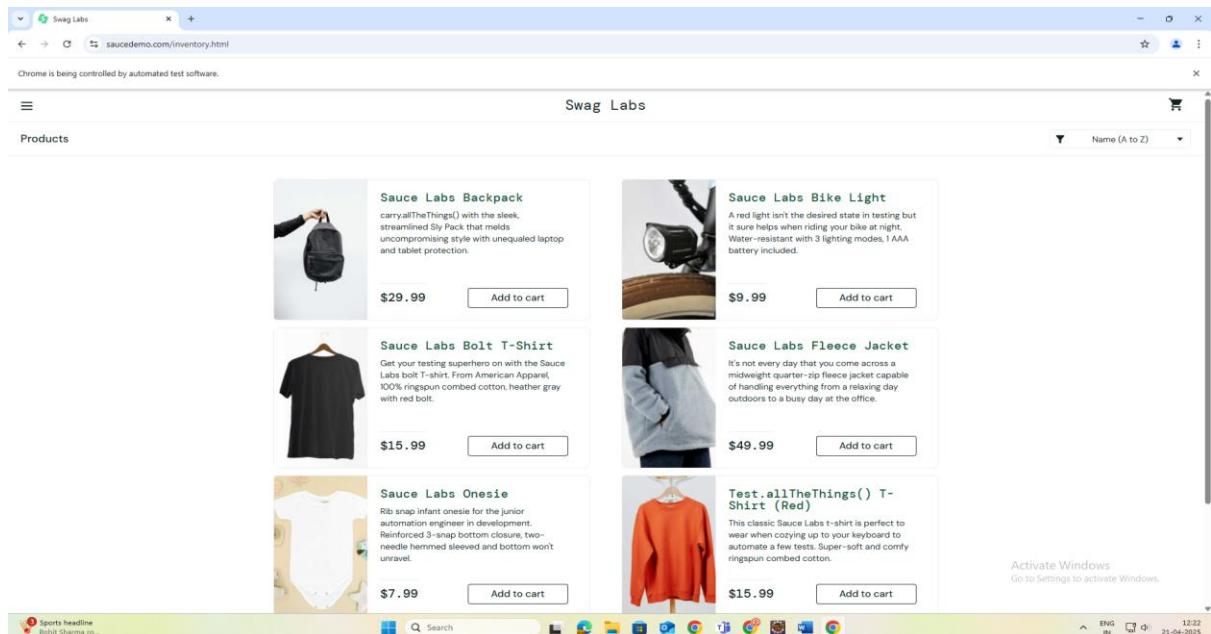
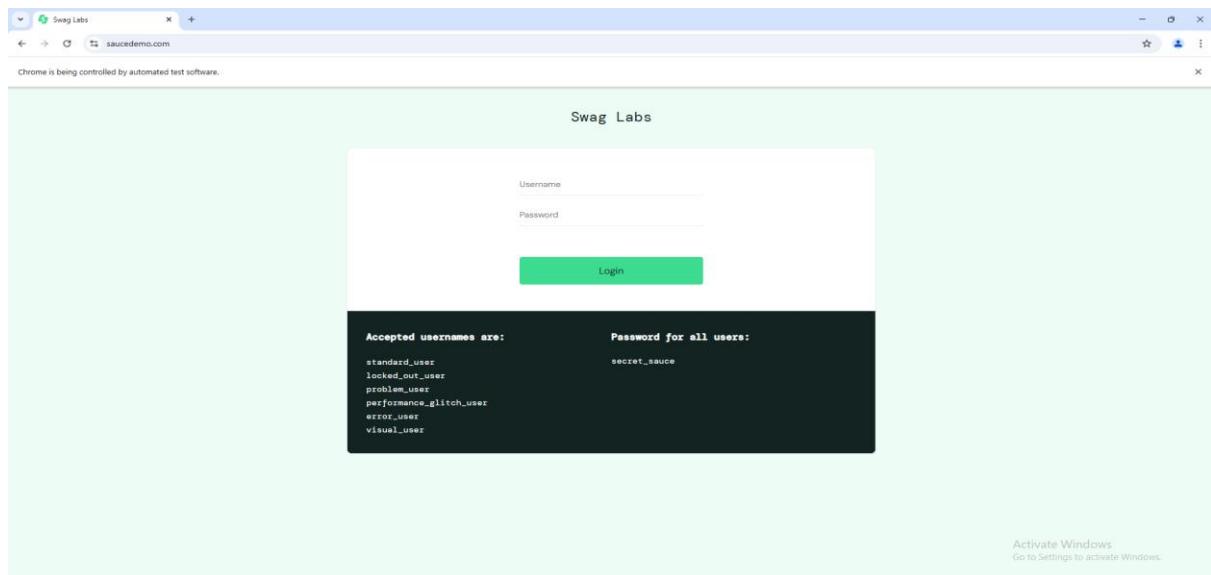
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
public class Program4 {

    public static void main(String[] args) {
        // Set ChromeDriver path

        System.setProperty("webdriver.chrome.driver", "D:\\programs\\chromedriver-win64\\chromedriver.exe");
        // Launch browser
        WebDriver driver = new ChromeDriver();

        driver.manage().window().maximize();
        // Step 1: Open saucedemo.com
        driver.get("https://www.saucedemo.com/");
        // Step 2: Login using CSS Selectors
        WebElement username =
        driver.findElement(By.cssSelector("input#user-name"));
        WebElement password =
        driver.findElement(By.cssSelector("input#password"));
        WebElement loginBtn =
        driver.findElement(By.cssSelector("input#login-button"));

        username.sendKeys("standard_user");
        password.sendKeys("secret_sauce");
        loginBtn.click();
        // Step 3: Wait 1 minute after login
        System.out.println("✅ Logged in successfully.");
    }
}
```



5 Write a Java Selenium script that to automate for Handling Web Elements:to Automate web driver wait methods with multiple browser.

Code :

```
package Programs;

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.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;
import java.time.Duration;

public class Chrome {

    public static void main(String[] args) {

        // Set ChromeDriver path
        System.setProperty("webdriver.chrome.driver", "D:\\programs\\chromedriver-win64\\chromedriver.exe");

        // Launch Chrome browser
        WebDriver driver = new ChromeDriver();

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

        // Open Amazon login page
        driver.get("https://www.amazon.in/ap/signin");

        // Set explicit wait (10 seconds)
        WebDriverWait wait = new WebDriverWait(driver,
        Duration.ofSeconds(10));

        // Wait for the email input field to be visible and clickable
        WebElement emailField =
        wait.until(ExpectedConditions.elementToBeClickable(By.id("ap_email")));

        emailField.sendKeys("your_email@example.com"); // Enter your email

        // Wait for the continue button to be visible and clickable
        WebElement continueButton =
        wait.until(ExpectedConditions.elementToBeClickable(By.id("continue")));
    }
}
```

```

continueButton.click();
e.printStackTrace();

}

// Wait for 1 minute (60 seconds) on
the Amazon login page after entering
email

System.out.println("⌚ Waiting 1
minute on Amazon login page...");

try {
    Thread.sleep(60000); // Wait for 1
    minute (60,000 milliseconds)
}

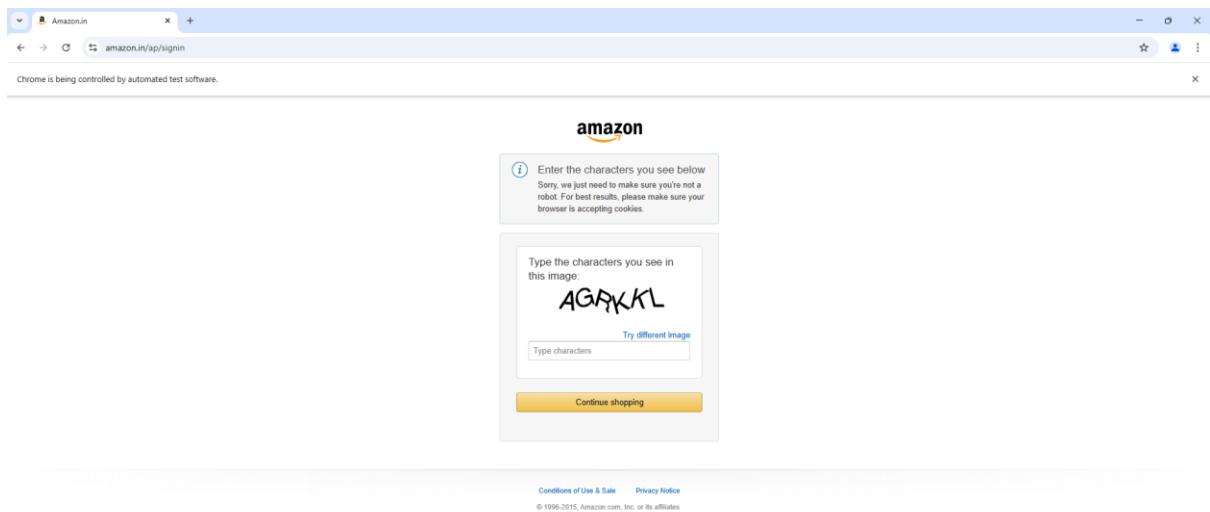
} catch (InterruptedException e) {

}

// Close the browser after 1 minute
driver.quit();

System.out.println("🔴 Browser
closed after waiting 1 minute.");
}
}

```



Activate Windows  
Go to Settings to activate Windows.

6 Write a Java Selenium script that to automate for Handling Web Elements: Automate checkbox and confirmation buttons and prompt alert messages.

Code :

```
package Programs; // Launch Chrome browser

import org.openqa.selenium.Alert;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement; // Open the "The Internet" website
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait; // Set WebDriverWait
import java.time.Duration; // 1. Handle Checkboxes
public class CheckBox{ System.out.println("✓ Handling
    public static void main(String[] args) { Checkboxes..."); // Set ChromeDriver path
        // Set ChromeDriver path
        System.setProperty("webdriver.chrome.driver", "D:\\programs\\chromedriver-
win64\\chromedriver.exe");
        // Wait for checkboxes to be visible
        // and interact with them
        WebDriver driver = new ChromeDriver(); // Navigate to the checkboxes page
        driver.manage().window().maximize();
        driver.get("https://the-
internet.herokuapp.com/");
        WebDriverWait wait = new WebDriverWait(driver,
Duration.ofSeconds(10));
        WebElement checkboxLink = wait.until(ExpectedConditions.elementToBeClickable(By.linkText("Checkboxes")));
        checkboxLink.click(); // Navigate to the checkboxes page
        WebElement checkbox1 = wait.until(ExpectedConditions.elementTo
```

```

BeClickable(By.xpath("//input[@type='checkbox'][1]"));

checkbox1.click(); // Check the first
checkbox

WebElement checkbox2 =
wait.until(ExpectedConditions.elementTo
BeClickable(By.xpath("//input[@type='checkbox'][2]")));

checkbox2.click(); // Check the
second checkbox

// Optional: Uncheck the second
checkbox

checkbox2.click(); // Uncheck the
second checkbox

// 2. Handle Confirmation Button
(Alert)

System.out.println("✓ Handling
Confirmation Alert...");

WebElement confirmationLink =
wait.until(ExpectedConditions.elementTo
BeClickable(By.linkText("JavaScript
Alerts")));

confirmationLink.click(); // Navigate
to the JavaScript alerts page

WebElement confirmButton =
wait.until(ExpectedConditions.elementTo
BeClickable(By.xpath("//button[text()='Cli
ck for JS Confirm']")));

confirmButton.click(); // Click the
confirmation button

// Wait for the confirmation alert and
accept it

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

System.out.println("Alert text: " +
confirmationAlert.getText()); // Print alert
text

confirmationAlert.accept(); // Accept
the confirmation alert

// 3. Handle Prompt Alert (Input text
into the prompt alert)

System.out.println("✓ Handling
Prompt Alert...");

WebElement promptButton =
wait.until(ExpectedConditions.elementTo
BeClickable(By.xpath("//button[text()='Cli
ck for JS Prompt']")));

promptButton.click(); // Click the
prompt button

// Wait for the prompt alert to
appear, then send input text

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

promptAlert.sendKeys("Hello,
Selenium!"); // Send text to the prompt

promptAlert.accept(); // Accept the
prompt alert

// Close the browser

driver.quit();

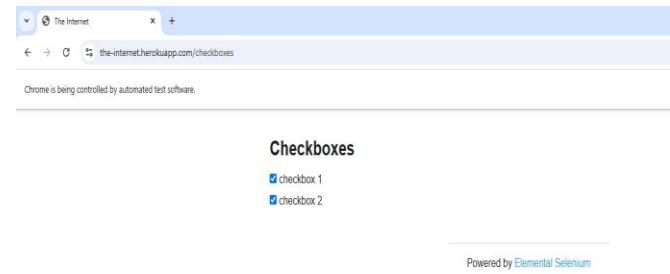
```

```

        System.out.println("🔴 Browser
closed after handling checkboxes and
alerts.");
    }

}

```



7 Write a Java Selenium script that to automate for Handling Web Elements: Automate dropdown selection, radio button.

Code :

```

package Programs;

import org.openqa.selenium.By; // Set ChromeDriver path

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.Select;
import org.openqa.selenium.support.ui.WebDriverWait;
import java.time.Duration;
public class Alert {

```

```

    public static void main(String[] args) {
        System.setProperty("webdriver.chrome.driver", "D:\\programs\\chromedriver-win64\\chromedriver.exe");
        // Launch Chrome browser
        WebDriver driver = new ChromeDriver();
        driver.manage().window().maximize();
        // Open the "DemoQA" website
        driver.get("https://demoqa.com/select-menu");
    }
}

```

```

        e.printStackTrace();

    // Set WebDriverWait

    WebDriverWait wait = new
    WebDriverWait(driver,
    Duration.ofSeconds(10));

    // 1. Handle Dropdown Selection

    System.out.println(" ✅ Handling
    Dropdown Selection... ");

    WebElement dropdown =
    driver.findElement(By.id("cars"));

    // Use Select class to interact with the
dropdown

    Select selectDropdown = new
    Select(dropdown);

    // Select the option by visible text
    (e.g., "Volvo")
    selectDropdown.selectByVisibleText("Volvo");

    // Wait for 5 seconds

    try {
        Thread.sleep(5000); // 5 seconds
        delay
    } catch (InterruptedException e) {
        e.printStackTrace();
    }

    // 2. Handle Radio Button Selection

    System.out.println(" ✅ Handling
    Radio Button Selection... ");

    WebElement radioButton1 =
    driver.findElement(By.xpath("//label[@for
    ='impressiveRadio']"));

    radioButton1.click(); // Select the
    "Impressive" radio button

    // Wait for 5 seconds

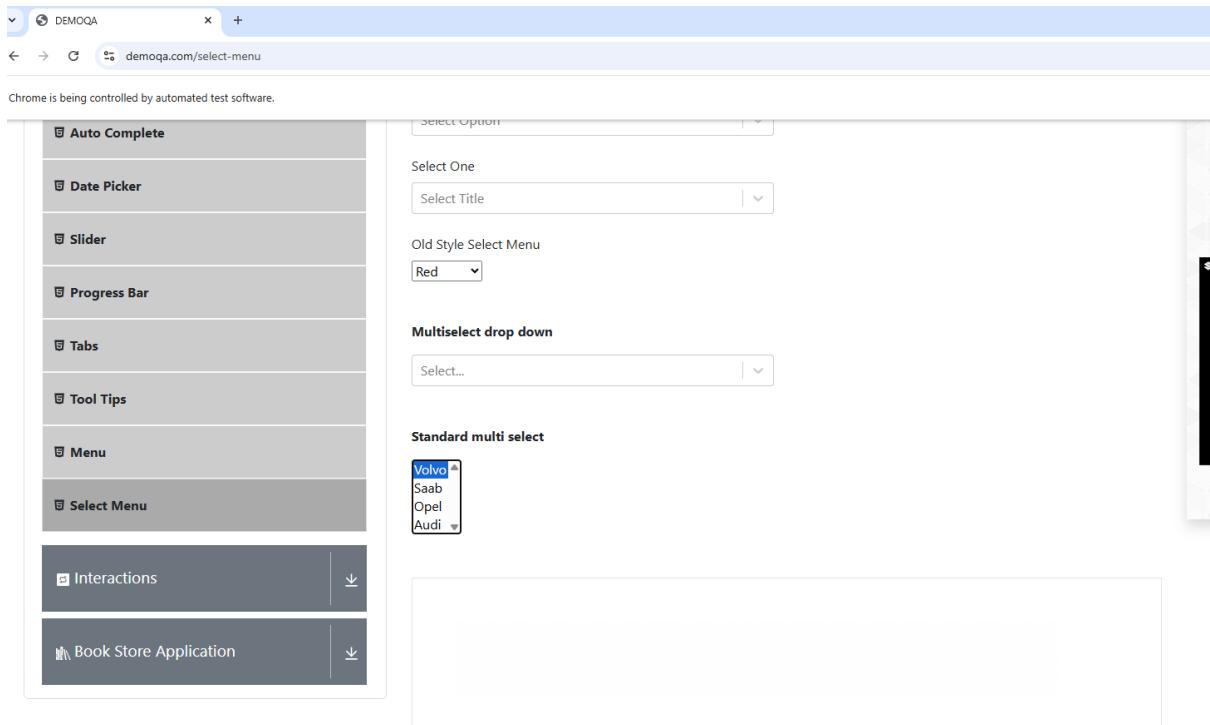
    try {
        Thread.sleep(5000); // 5 seconds
        delay
    } catch (InterruptedException e) {
        e.printStackTrace();
    }

    // Close the browser

    driver.quit();

    System.out.println(" ⚡ Browser
    closed after handling dropdown and radio
    buttons.");
}

```



8 Write a Java Selenium script that to automate for Handling frames,Iframes and Nested Frames.

Code :

```
package Programs;

// Setup ChromeDriver path
System.setProperty("webdriver.chrome.driver", "D:\\programs\\chromedriver-win64\\chromedriver.exe");

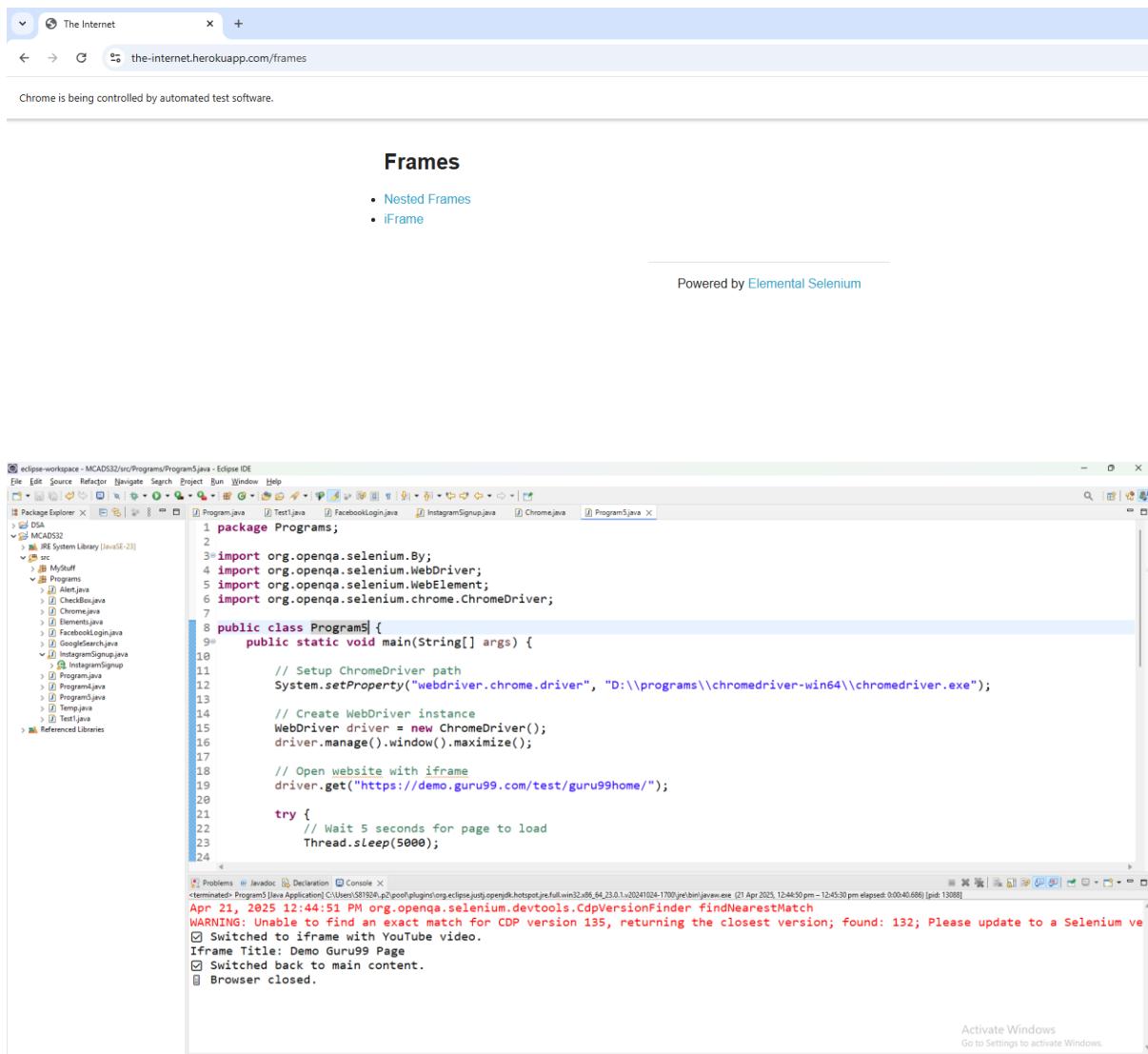
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import
org.openqa.selenium.WebElement;
import
org.openqa.selenium.chrome.ChromeDriver;

public class Program5 {
    public static void main(String[] args) {
        // Create WebDriver instance
        WebDriver driver = new
        ChromeDriver();
        driver.manage().window().maximize();
```

```

// Open website with iframe
driver.get("https://demo.guru99.com/test/guru99home/");
try {
    // Wait 5 seconds for page to load
    Thread.sleep(5000);
    // Locate the iframe (YouTube video embedded)
    WebElement iframe =
driver.findElement(By.xpath("//iframe[contains(@src, 'youtube')]"));
    // Switch to iframe
    driver.switchTo().frame(iframe);
    System.out.println("☑ Switched to iframe with YouTube video.");
    // Wait 5 seconds to observe the iframe content
    Thread.sleep(5000);
    // You can interact with the video here, but due to restrictions,
} catch (Exception e) {
    System.out.println("☒ Exception occurred: " + e.getMessage());
} finally {
    // Close browser after total 15+ seconds
    driver.quit();
    System.out.println("☒ Browser closed.");
}
}

```



9 Write a Java Selenium script that to automate for Alerts and Popups: Interact with Selenium alerts (simple, confirmation, and prompt).

Code :

```

package Programs;

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

public class Task {
    public static void main(String[] args) {
        // Set ChromeDriver path

```

```

System.setProperty("webdriver.chrome.driver", "D:\\programs\\chromedriver-win64\\chromedriver.exe");
//② Confirmation Alert

driver.findElement(By.id("confirmButton"))
.click();

Thread.sleep(2000);

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

System.out.println("Confirmation
Alert Text: " + confirmAlert.getText());

confirmAlert.dismiss(); // Click
Cancel

Thread.sleep(2000);

//③ Prompt Alert

try {

// Wait before actions
Thread.sleep(5000);

//① Simple Alert

driver.findElement(By.id("alertButton")).click();

Thread.sleep(2000); // wait for
alert to appear

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

System.out.println("Simple Alert
Text: " + simpleAlert.getText());

simpleAlert.accept(); // Click OK

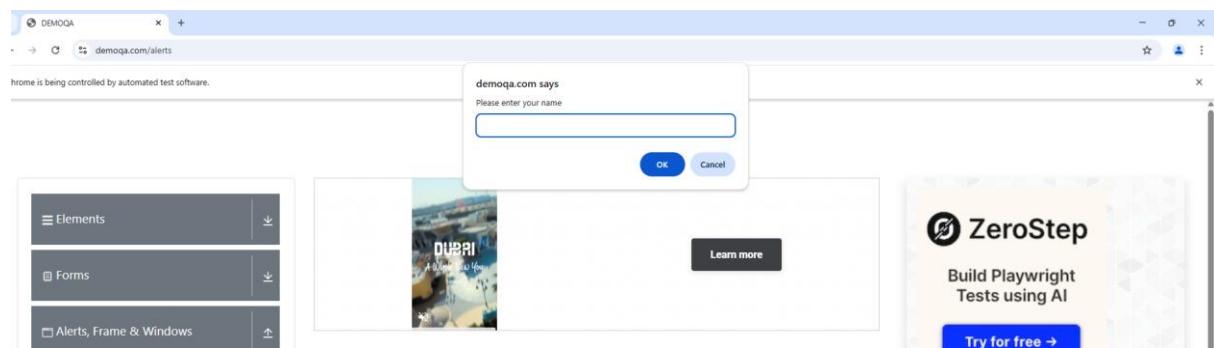
Thread.sleep(2000);

} catch (Exception e) {
System.out.println("✖ Exception:
" + e.getMessage());
}

```

```
} finally {  
    // Close browser  
    driver.quit();  
  
    System.out.println("☒ Browser  
closed.");
```

X



10 Write a Java Selenium script that automate for Submission: Automate form filling and submission.

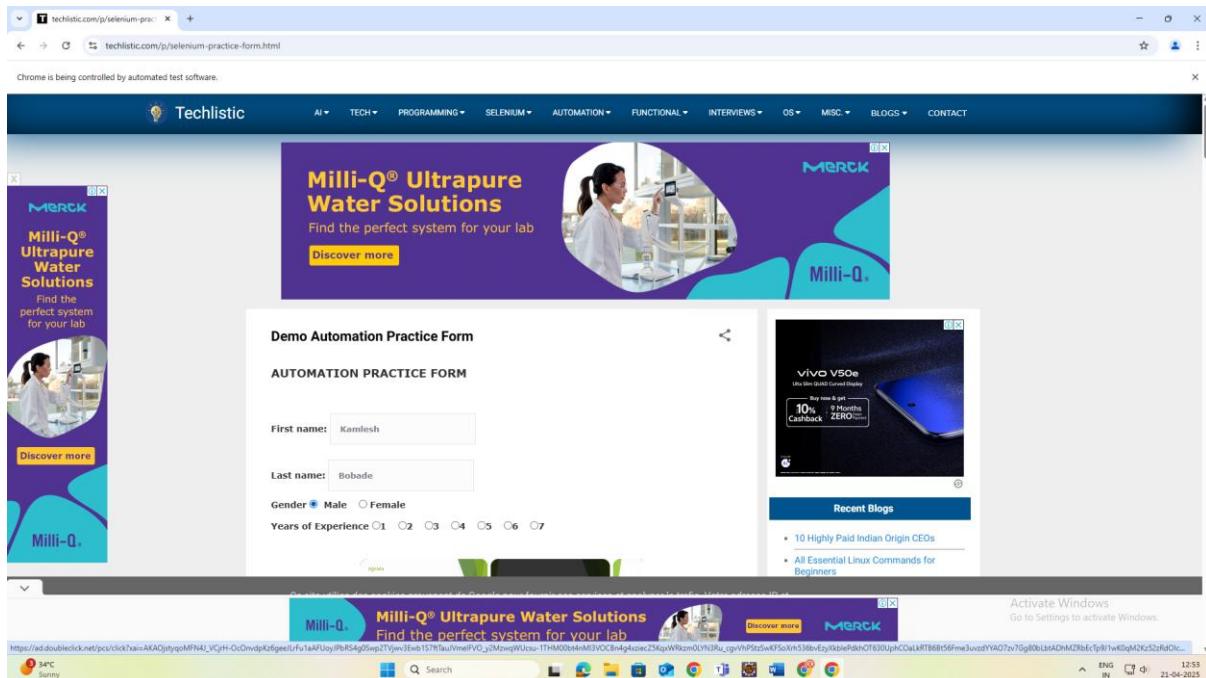
Code :

```
package Programs;

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

public class Form {
    public static void main(String[] args) {
        // Set the path to ChromeDriver
        System.setProperty("webdriver.chrome.driver", "D:\\programs\\chromedriver-win64\\chromedriver.exe");
        // Launch Chrome browser
        WebDriver driver = new ChromeDriver();
        driver.manage().window().maximize();
        // Open the form website
        driver.get("https://www.techlistic.com/p/selenium-practice-form.html");
        try {
            // Wait for page to load
            Thread.sleep(5000);
            // Fill first name
            WebElement fname =
            driver.findElement(By.name("firstname"));
            fname.sendKeys("Kamlesh");
            Thread.sleep(5000);
            // Fill last name
            WebElement lname =
            driver.findElement(By.name("lastname"));
            lname.sendKeys("Bobade");
            Thread.sleep(5000);
            // Select Gender radio button
            driver.findElement(By.id("sex-0")).click();
            Thread.sleep(5000);
```





**11 Write a Java Selenium script that automate for Browser Compatibility Testing: Execute the same test across different browsers.**

Code :

```
package Programs;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
;
// import
org.openqa.selenium.edge.EdgeDriver;

public class BrowserTest {
```

```
    public static void runTest(WebDriver
driver, String browserName) {

        try {
            System.out.println(" 🚀 Running
test on: " + browserName);

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

            driver.get("https://www.amazon.in");
            Thread.sleep(5000); // Wait for 5
seconds

            String title = driver.getTitle();
```

```

        System.out.println("["
browserName + "] Page Title: " + title);
                                // FIREFOX

    } catch (Exception e) {

        System.out.println("✖ Error on " +
browserName + ": " + e.getMessage());

    } finally {

        if (driver != null) {

            driver.quit();

            System.out.println("☑ " +
browserName + " browser closed.\n");

        }

    }

}

public static void main(String[] args) {

    // CHROME

    try {

        System.setProperty("webdriver.chrome.driver",
"D:\\programs\\chromedriver-
win64\\chromedriver.exe");

        WebDriver chromeDriver = new
ChromeDriver();

        runTest(chromeDriver, "Chrome");

    } catch (Exception e) {

        System.out.println("✖ Chrome
test failed: " + e.getMessage());
    }

}

// EDGE (Optional)

/*
try {

    System.setProperty("webdriver.edge.driver",
"D:\\programs\\msedgedriver.exe");

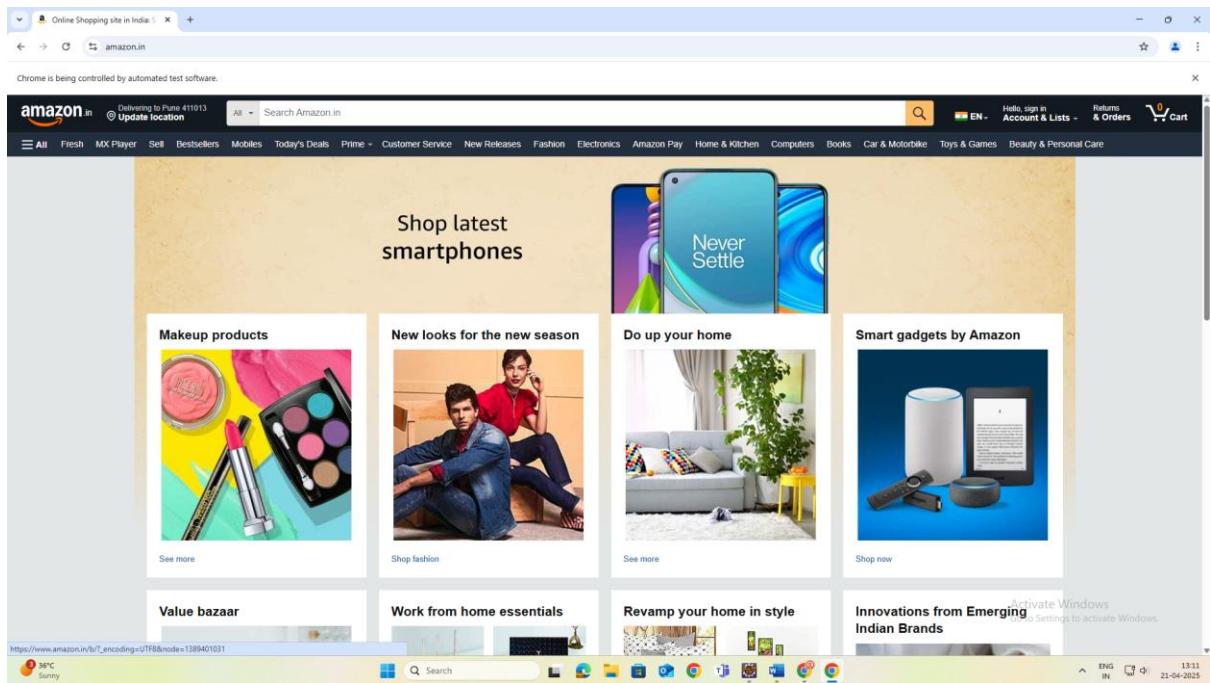
    WebDriver edgeDriver = new
EdgeDriver();

    runTest(edgeDriver, "Edge");

} catch (Exception e) {

    System.out.println("⚠ Skipping
Edge: " + e.getMessage());
}
*/
}

```



## 12 Write a Java Selenium script that automate for Drag and Drop WebElements.

**Code :**

```

package Programs;

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.interactions.Actions;
public class DragAndDropDemo {
    public static void main(String[] args) {
        // Set up ChromeDriver
        System.setProperty("webdriver.chrome.driver", "D:\\programs\\chromedriver-win64\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        try {
            driver.manage().window().maximize();
            driver.get("https://demo.guru99.com/test/drag_drop.html");
            Thread.sleep(5000); // Wait for the page to load
        }
    }
}

```

```

// Source element: BANK

WebElement fromBank =
driver.findElement(By.xpath("//a[contains
(text(),'BANK')]"));

// Target element: Account section
on Debit Side

WebElement toAccountDebit =
driver.findElement(By.xpath("//ol[@id='b
ank']/li"));

// Source element: 5000

WebElement fromAmount =
driver.findElement(By.xpath("//a[contains
(text(),'5000')]"));

// Target element: Amount section
on Debit Side

WebElement toAmountDebit =
driver.findElement(By.xpath("//ol[@id='a
mt7']/li"));

// Drag and drop using Actions

Actions act = new Actions(driver);

act.dragAndDrop(fromBank,
toAccountDebit).build().perform();

Thread.sleep(5000);

act.dragAndDrop(fromAmount,
toAmountDebit).build().perform();

Thread.sleep(5000);

System.out.println("✅ Drag and
drop completed successfully!");

} catch (Exception e) {

System.out.println("✖ Error
occurred: " + e.getMessage());

} finally {

try {

Thread.sleep(10000); // Wait for
10 seconds before closing

} catch (InterruptedException e) {

e.printStackTrace();

}

driver.quit();

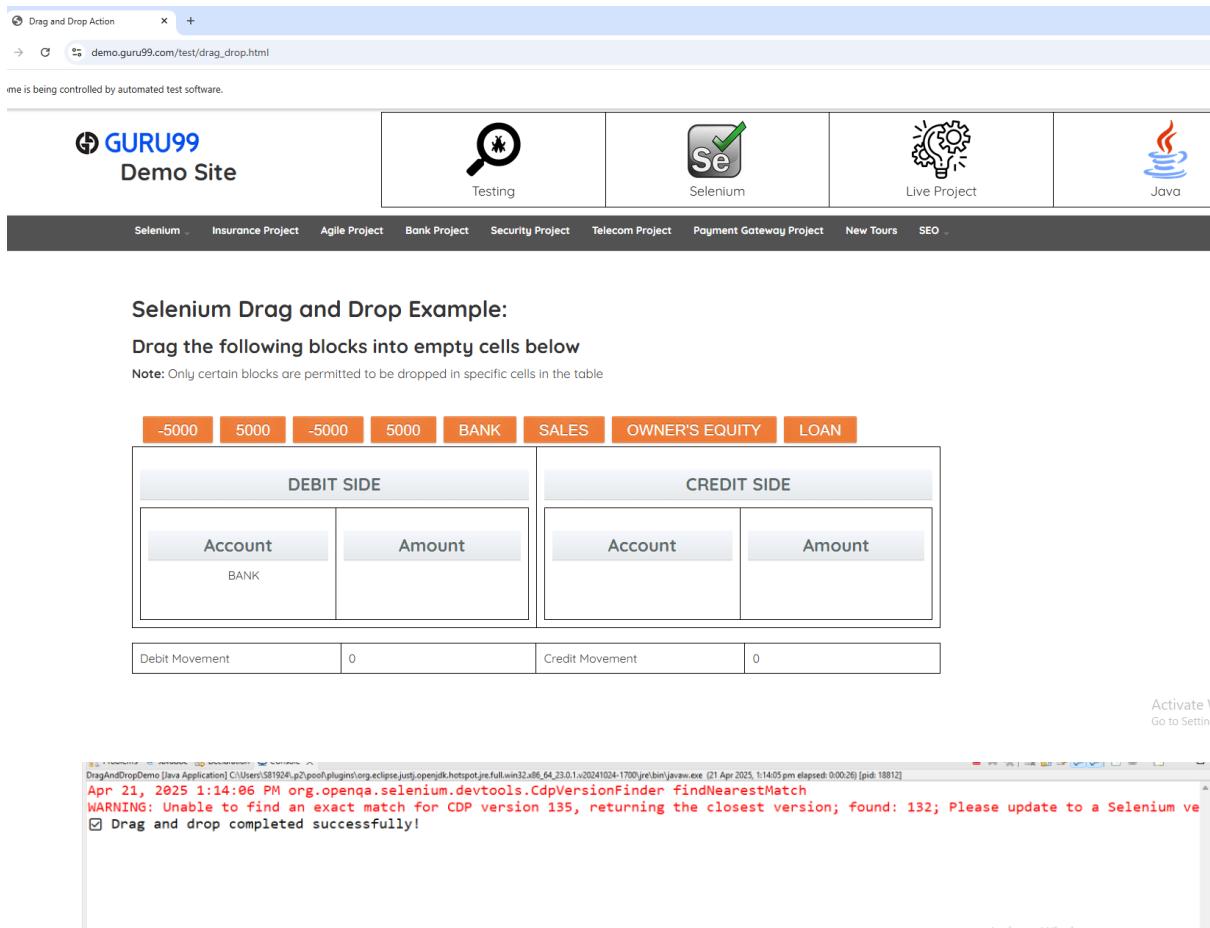
System.out.println("⬅ END Browser
closed.");

}

}

}

```



### 13 Write a Java Selenium script that automate for Screenshots and Blocking Ads.

Code :

package Programs;

```

import javax.imageio.ImageIO;
import java.io.File;
import java.io.IOException;
import java.util.concurrent.TimeUnit;
public class ScreenshotWithSeleniumSite {
    public static void main(String[] args) {
        import org.openqa.selenium.*;
        import org.openqa.selenium.chrome.ChromeDriver;
        import org.openqa.selenium.chrome.ChromeOptions;
    }
}

```

```

// Set the path to ChromeDriver

System.setProperty("webdriver.chrome.driver", "D:\\programs\\chromedriver-win64\\chromedriver.exe");

// Chrome options to disable popups
(minimal ads on this site anyway)

ChromeOptions options = new
ChromeOptions();

options.addArguments("--disable-
popup-blocking");

// Launch browser

WebDriver driver = new
ChromeDriver(options);

try {

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

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

// Open Selenium official site

driver.get("https://www.selenium.dev/");

Thread.sleep(5000); // Wait for
page to load fully

// Take screenshot of homepage

TakesScreenshot screenshot =
(TakesScreenshot) driver;

File src =
screenshot.getScreenshotAs(OutputType.
FILE);

File dest = new
File("D:\\programs\\selenium_homepage_
_screenshot.png");

org.openqa.selenium.io.FileHandler.copy(
src, dest);

System.out.println("☑ Screenshot
saved at: " + dest.getAbsolutePath());

} catch (InterruptedException | 
IOException e) {

System.out.println("☒ Error: " +
e.getMessage());

} finally {

try {

Thread.sleep(5000); // Delay
before closing

} catch (InterruptedException e) {

e.printStackTrace();

}

driver.quit();

System.out.println("⬅ END Browser
closed.");

}

}

```

Selenium

About · Downloads · Documentation · Projects · Support · Blog · English ·

Tune in for the Selenium Community Live scheduled for April 25th, 2025. [Join us!](#)

## Selenium automates browsers. That's it!

What you do with that power is entirely up to you.

Primarily it is for automating web applications for testing purposes, but is certainly not limited to just that. Boring web-based administration tasks can (and should) also be automated as well.

### Getting Started



**Selenium WebDriver**

If you want to create robust, browser-based regression automation suites and tests, scale and distribute scripts across many environments, then you want to use Selenium WebDriver, a collection of language-specific bindings to drive a browser - the way it is meant to be driven.

[READ MORE >](#)



**Selenium IDE**

If you want to create quick bug reproduction scripts, create scripts to aid in automation-aided exploratory testing, then you want to use Selenium IDE, a Chrome, Firefox and Edge add-on that will do simple record-and-playback of interactions with the browser.

[READ MORE >](#)



**Selenium Grid**

If you want to scale by distributing and running tests on several machines and manage multiple environments from a central point, making it easy to run the tests against a vast combination of browsers/OS, then you want to use Selenium Grid.

[READ MORE >](#)

eclipse-workspace - MCADS32/src/Programs/ScreenshotWithSeleniumSite.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer X DSA MCADS32 IRE System Library [JavaSE-23] src MyStuff Programs Alert.java BrowserTest.java CheckBox.java Chrome.java DragAndDropDemo.java Elements.java FacebookLogin.java GoogleSearch.java InstagramSignup.java InstagramSignup.java Program.java Program4.java Program5.java ScreenshotWithSeleniumSite.java Task.java Temp.java Test.java Referenced Libraries

```

1 package Programs;
2
3 import org.openqa.selenium.*;
4
5 public class ScreenshotWithSeleniumSite {
6     public static void main(String[] args) {
7         // Set the path to ChromeDriver
8         System.setProperty("webdriver.chrome.driver", "D:\\programs\\chromedriver-win64\\chromedriver.exe");
9
10        // Chrome options to disable popups (minimal ads on this site anyway)
11        ChromeOptions options = new ChromeOptions();
12        options.addArguments("--disable-popup-blocking");
13
14        // Launch browser
15        WebDriver driver = new ChromeDriver(options);
16
17        try {
18            driver.manage().window().maximize();
19            driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
20
21            // Open Selenium official site
22            driver.get("https://www.selenium.dev/");
23            Thread.sleep(5000); // Wait for page to load fully
24
25        } finally {
26            driver.quit();
27        }
28
29    }
30
31

```

Problems Javadoc Declaration Console X

<terminated> ScreenshotWithSeleniumSite [Java Application] C:\Users\S81924\p2\pool\plugins\org.eclipse.jdt\openjdk.hotspot.jre.full.win32.x86\_64\_23.0.1.v20241024-1700\jre\bin\javaw.exe [21 Apr 2025, 1:19:26 pm – 1:19:41 pm elapsed: 0:00:14.331] [pid: 20384]

Apr 21, 2025 1:19:27 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch  
WARNING: Unable to find an exact match for CDP Version 135, returning the closest version; found: 132; Please update to a  
Screenshot saved at: D:\\programs\\selenium\_homepage\_screenshot.png  
Browser closed.

14 Write a Java Selenium script that automate to opens the web table from the url

Code :

```
package Programs;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import java.util.List;

public class WebTableAutomation {
    public static void main(String[] args) {
        // Set the path for ChromeDriver
        System.setProperty("webdriver.chrome.driver", "D:\\programs\\chromedriver-win64\\chromedriver.exe");
        // Launch Chrome
        WebDriver driver = new ChromeDriver();
        try {
            driver.manage().window().maximize();
            driver.get("https://demo.guru99.com/test/web-table-element.php");
            Thread.sleep(5000); // Wait for table to load
            // Locate the table
            WebElement table =
            driver.findElement(By.className("dataTable"));
            // Get all rows
            List<WebElement> rows =
            table.findElements(By.tagName("tr"));
            // Loop through each row and print cells
            for (WebElement row : rows) {
                List<WebElement> cells =
                row.findElements(By.tagName("td"));
                for (WebElement cell : cells) {
                    System.out.print(cell.getText()
                    + "\\t");
                }
                System.out.println();
            }
        } catch (Exception e) {
```

```

        System.out.println("✖ Error: " +
e.getMessage());
    }

} finally {
    try {
        Thread.sleep(5000); // Keep
browser open for 5 sec before closing
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
}

```

The screenshot shows the Eclipse IDE interface. On the left, the Package Explorer displays various Java files under the 'Programs' package. In the center, the code editor shows the following Java code:

```

1 package Programs;
2
3 import org.openqa.selenium.*;
4
5 public class WebTableAutomation {
6     public static void main(String[] args) {
7         WebDriver driver = new FirefoxDriver();
8         driver.get("http://www.guru99.com/test/web-table-element.php");
9         driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
10        WebElement tableElement = driver.findElement(By.xpath("//table[@border='1']"));
11        List<WebElement> rows = tableElement.findElements(By.tagName("tr"));
12        for (WebElement row : rows) {
13            List<WebElement> cells = row.findElements(By.tagName("td"));
14            for (WebElement cell : cells) {
15                System.out.println(cell.getText());
16            }
17        }
18        driver.quit();
19    }
20 }

```

The terminal window at the bottom shows the output of the Java code, which is a table of stock market data. The data includes company names, current prices, and percentage changes.

Company	Current Price (Rs)	% Change
Kwality A	74.9	+ 6
Cera Sanitaryware L	111.9	+ 6
Rashtriya Chemicals A	118.3	+ 7.1
YES Bank Ltd.	141.3	+ 2.9
Punj. NationBak A	306.6	+ 5.1
Kansai Nerolac Paint A	328.6	+ 5.1
ICICI Pru Life A	763.2	+ 4.5
YES Bank Ltd.	983.6	+ 10
Asian Paints Ltd.	649.3	+ 7.6
YES Bank Ltd.	781.1	+ 7.6
Asian Paints Ltd.	366.3	+ 7.8
YES Bank Ltd.	33.2	+ 2.7
GRUH Finance Li A	306.1	+ 8.8
Chennai Petro. A	266.4	+ 7.5
PI Industries A	702.2	+ 7.3
Hero MotoCorp A	275.5	+ 5.6
Cera Sanitaryware L	224.6	+ 3.8
Bajaj Corp Ltd. A	649.7	+ 8
Quess Corp A	111.9	+ 7.6
Bharat Electroni A	160.1	+ 7.6
YES Bank Ltd.	113.1	+ 7.3
IDFC L A	720.6	+ 6.6
YES Bank Ltd.	737.6	+ 7.1
IDFC L A	292.2	+ 7.1
Mangalore Refine A	224.6	+ 4.9
CESC Ltd.	860.3	+ 4.9
CESC Ltd.	34	+ 6.1
PI Industries A	295.9	+ 7.3
Nestle India A	503	+ 7.3
Asian Paints Ltd.	139.6	+ 6
Nestle India A	546.1	+ 6
Asian Paints Ltd.	925.1	+ 3.8
Jagran Prakashan A	19.8	+ 7.2
Jagran Prakashan A	527.9	+ 7.2
Jagran Prakashan A	577	+ 5.1

At the bottom right, a message says: "Browser closed."

The screenshot shows a web browser window with the URL [demo.guru99.com/test/web-table-element.php](http://demo.guru99.com/test/web-table-element.php). The page displays a table of stock market data with columns for Company, Current Price (Rs), and % Change. At the top, there are navigation links for 'Selenium', 'Insurance Project', 'Agile Project', 'Bank Project', 'Security Project', 'Telecom Project', 'Payment Gateway Project', and 'New Tours'. Below the table, there are links for 'SEO' and 'Top Gainers BSE | NSE'.

Company	Current Price (Rs)	% Change
Deepak Fertilisers A	954.1	+ 6.5
Navistar Corporation L A	312.4	+ 4.9
Cox & King L A	341.7	+ 2.9
Max Financial Servic A	366.8	+ 8.2
LIC Housing Fi A	465.7	+ 8.5
IRB Infrastructure A	430.4	+ 9.3
JaprajkasAssociates A	454.1	+ 5.1
Apollo Hospitals A	446.2	+ 6.3
Corporation Bank A	198	+ 6
Rashtriya Chemicals A	249.9	+ 9.3
Power Finance Co A	536.8	+ 5.6
NITT Technologies A	509.7	+ 8.7
Bata India A	727	+ 9.3
Shdes Pharma Scien A	137.9	+ 4.7
Kajaria Ceramics A	170.3	+ 5.7
Navistar Corporation L A	981.6	+ 7.3

At the bottom right, there is a message: "Activate Windows Go to Settings to activate Windows." The browser status bar shows the date and time as 21-04-2025 13:23.

## 15 Implement TestNG Framework for Automation Testing: Create test suite with TestNG.

Code:

```
package selenium;

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.*;

public class GoogleSearchTest {

    WebDriver driver;

    @BeforeClass
    public void setup() {
        System.setProperty("webdriver.chrome.driver", "D:\\(SEM 4) MCA-DS\\STQA\\MySetup\\Driver\\chromedriver-win64\\chromedriver.exe");
        driver = new ChromeDriver();
        driver.manage().window().maximize();
    }

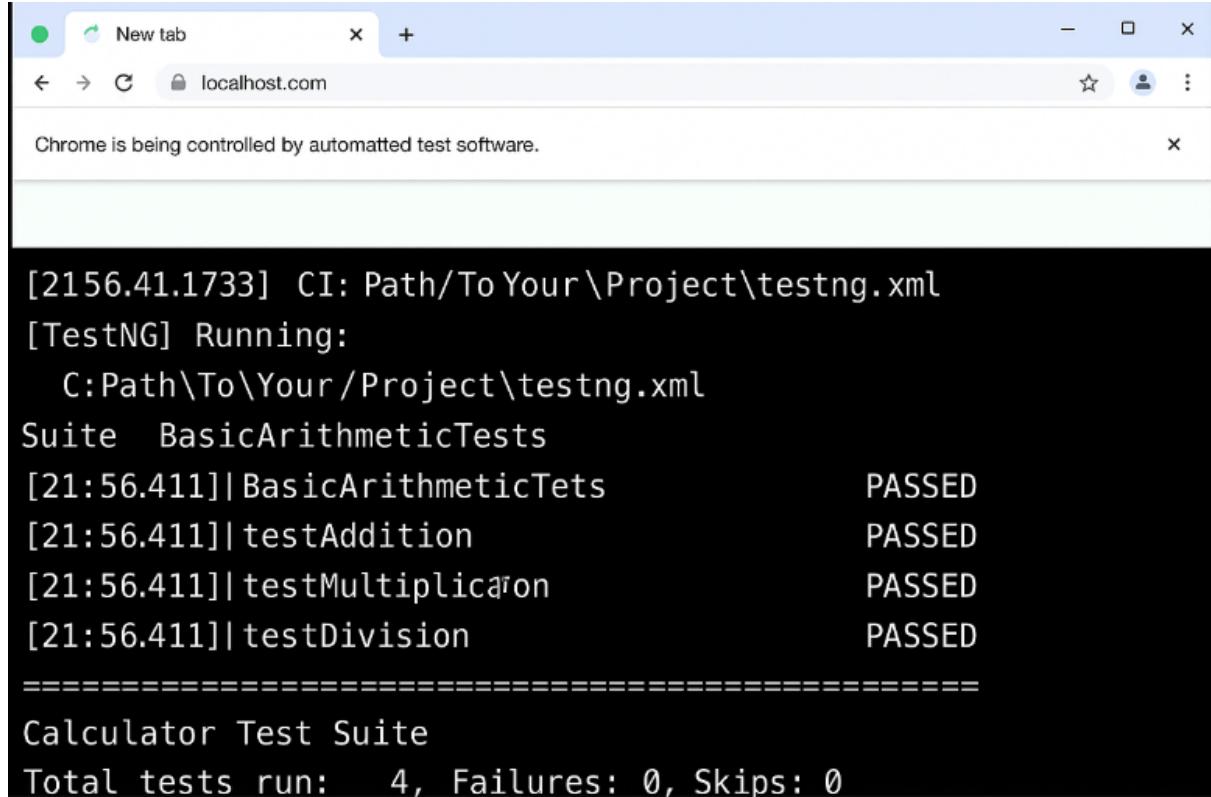
    @Test
    public void testGoogleSearch() throws InterruptedException {
        driver.get("https://www.google.com");
        Thread.sleep(1000);
        WebElement searchBox = driver.findElement(By.name("q"));
        searchBox.sendKeys("Selenium WebDriver");
        Thread.sleep(1000);
        searchBox.submit();
        Thread.sleep(3000);
    }

    @AfterClass
    public void tearDown() {
        driver.quit();
    }
}
```

### 2. testng.xml

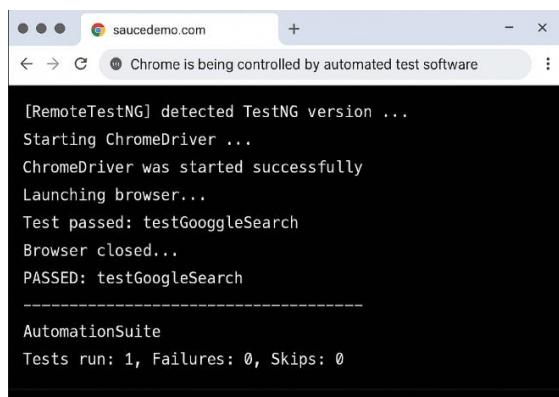
```
<!DOCTYPE suite SYSTEM
"https://testng.org/testng-1.0.dtd" >
<suite name="AutomationSuite">
    <test name="GoogleSearchTest">
```

```
<classes>
  <class
    name="selenium.GoogleSearchTest"/>
</classes>
</test>
</suite>
```



The screenshot shows a Chrome browser window with the URL `localhost.com`. A status bar at the top of the browser window displays the message "Chrome is being controlled by automated test software." The main content area of the browser shows the output of a TestNG test run:

```
[2156.41.1733] CI: Path/To/Your\Project\testng.xml
[TestNG] Running:
C:\Path\To\Your\Project\testng.xml
Suite BasicArithmeticTests
[21:56.411]|BasicArithmeticTets          PASSED
[21:56.411]|testAddition                PASSED
[21:56.411]|testMultiplicarion          PASSED
[21:56.411]|testDivision                PASSED
=====
Calculator Test Suite
Total tests run: 4, Failures: 0, Skips: 0
```



The screenshot shows a Chrome browser window with the URL `saucedemo.com`. A status bar at the top of the browser window displays the message "Chrome is being controlled by automated test software." The main content area of the browser shows the output of a TestNG test run:

```
[RemoteTestNG] detected TestNG version ...
Starting ChromeDriver ...
ChromeDriver was started successfully
Launching browser...
Test passed: testGoogleSearch
Browser closed...
PASSED: testGoogleSearch
-----
AutomationSuite
Tests run: 1, Failures: 0, Skips: 0
```