

Assignment-10

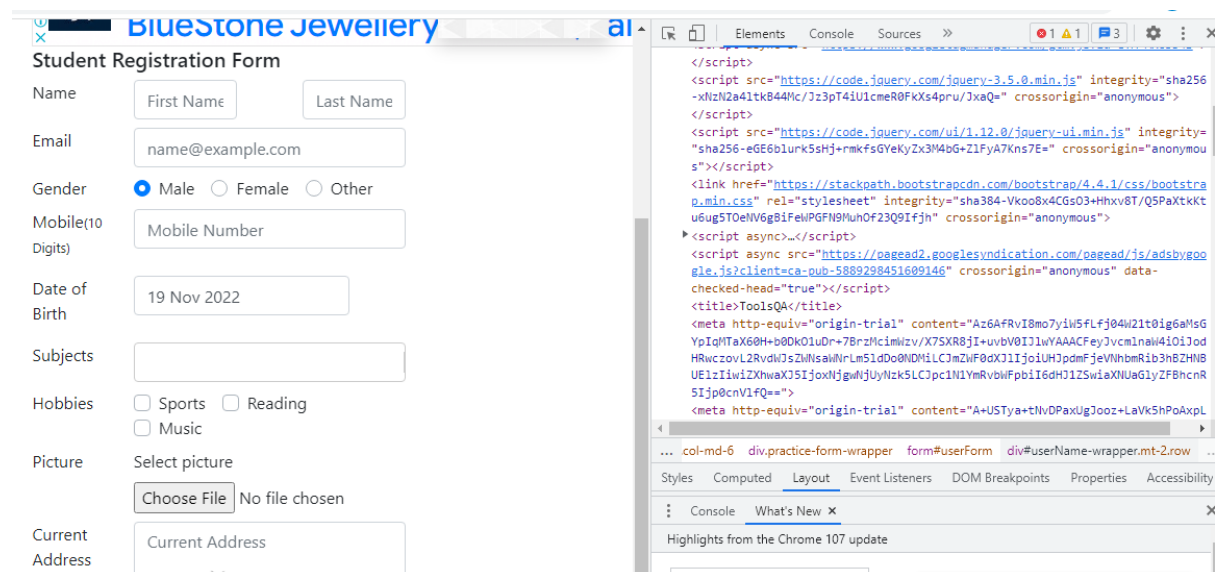
Test any website and write the script for finding elements by including all locators.

```
package Assignment10;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class ScriptusingLocators {

    public static void main(String[] args) {

        System.setProperty("webdriver.chrome.driver", "C:\\Users\\HARSHAL\\Desktop\\selenium\\chrome
        driver_win32 (1)\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("https://demoqa.com/automation-practice-form");
        driver.findElement(By.id("firstName"));

        driver.findElement(By.name("gender"));
        driver.findElement(By.className("practice-form-wrapper"));
        driver.get("https://demoqa.com/links");
        driver.findElement(By.linkText("Home"));
        driver.get("https://demoqa.com/text-box");
        driver.findElement(By.xpath("//input[@id='userName']"));
        driver.close();
    }
}
```



Test any website and write the script for testing textbox, dropdown,checkbox and radio button for the registration form.

```
package Assignment10;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.Select;
import org.openqa.selenium.*;
public class TestingtextboxDropdown {

    public static void main(String[] args) {

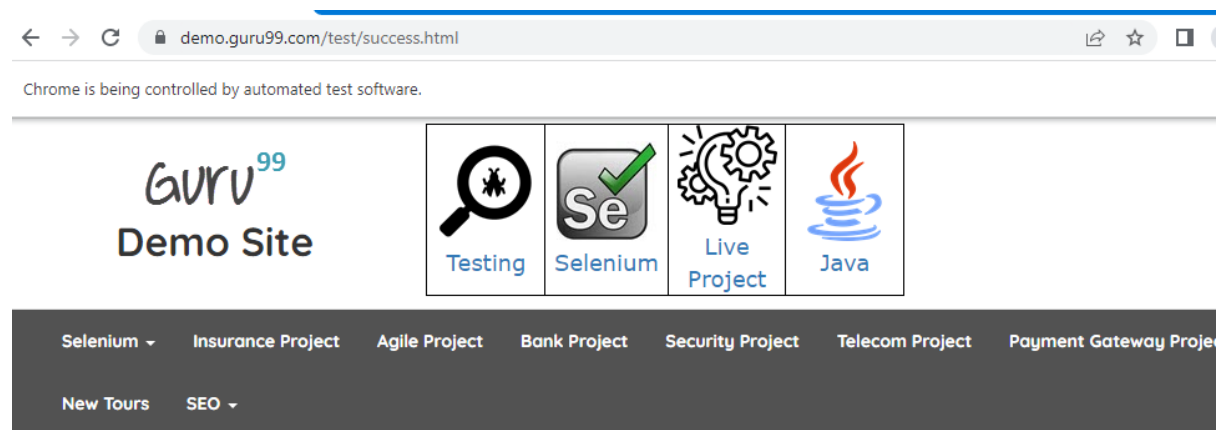
        System.setProperty("webdriver.chrome.driver","C:\\Users\\HARSHAL\\Desktop\\selenium\\
chromedriver_win32 (1)\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        String baseUrl = "http://demo.guru99.com/test/login.html";
        driver.get(baseUrl);
        WebElement email = driver.findElement(By.id("email"));
        WebElement password = driver.findElement(By.id("passwd"));
        email.sendKeys("abcd@gmail.com");
        password.sendKeys("abcdefghlkjl");
        System.out.println("Text Field Set");
        email.clear();
        password.clear();
        System.out.println("Text Field Cleared");
        WebElement login = driver.findElement(By.id("SubmitLogin"));
        email.sendKeys("abcd@gmail.com");
        password.sendKeys("abcdefghlkjl");
        login.click();
        System.out.println("Login Done with Click");
        driver.get("http://demo.guru99.com/test/radio.html"); WebElement radio1 =
driver.findElement(By.id("vfb-7-1"));
        WebElement radio2 = driver.findElement(By.id("vfb-7-2"));
        //Radio Button1 is selected
        radio1.click();
        System.out.println("Radio Button Option 1 Selected");

        //Radio Button1 is de-selected and Radio Button2 is selected
        radio2.click();
        System.out.println("Radio Button Option 2 Selected");
        // Selecting CheckBox
        WebElement option1 = driver.findElement(By.id("vfb-6-0"));
        // This will Toggle the Check box
        option1.click();
        // Check whether the Check box is toggled on
        if (option1.isSelected()) {
            System.out.println("Checkbox is Toggled On");
        } else {
            System.out.println("Checkbox is Toggled Off");
        }
    }
}
```

```

        System.setProperty("webdriver.chrome.driver", "C:\\Users\\HARSHAL\\Desktop\\selenium\\
chromedriver_win32 (1)\\chromedriver.exe");
        String baseURL = "http://demo.guru99.com/test/newtours/register.php";
        driver.get(baseURL);
        Select drpCountry = new Select(driver.findElement(By.name("country")));
        drpCountry.selectByVisibleText("ANTARCTICA");
        drpCountry.selectByIndex(1);
        drpCountry.selectByValue("ANTARCTICA");
        driver.close();
    }
}

```



```

r- Login Done with Click
Radio Button Option 1 Selected
Radio Button Option 2 Selected
Checkbox is Toggled On

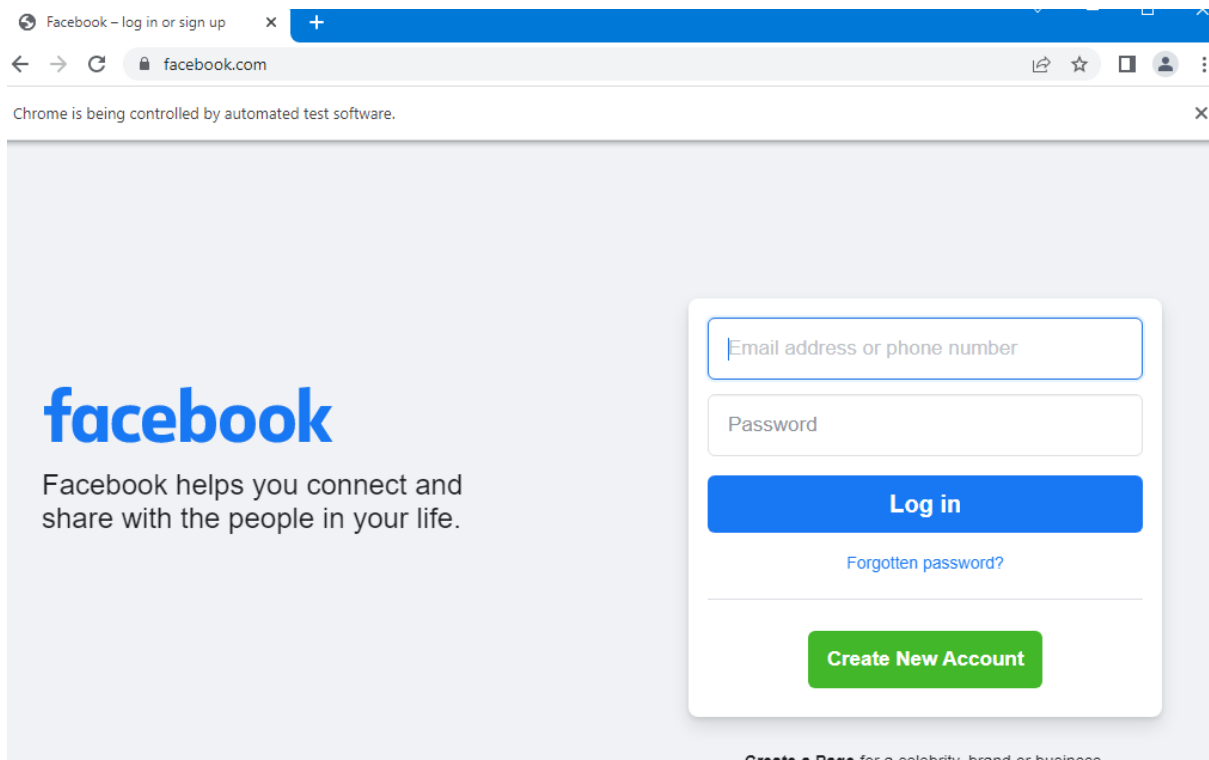
```

Test any website and write the script for that involves keyboard actions dragAndDrop(source,target), keyDown(modifier_key),keyUp(modifier_key),moveToElement(toElement),release(),sendKeys(onElement, charsequence).

```
package Assignment10;
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 KeyboardActions {

    public static void main(String[] args) {
        String baseUrl = "http://www.facebook.com/";

        System.setProperty("webdriver.chrome.driver", "C:\\Users\\HARSHAL\\Desktop\\selenium\\chromedriver_win32 (1)\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get(baseUrl);
        WebElement txtUsername = driver.findElement(By.id("email"));
        Actions builder = new Actions(driver);
        Action seriesOfActions = builder
            .moveToElement(txtUsername)
            .click()
            .keyDown(txtUsername, Keys.SHIFT)
            .sendKeys(txtUsername, "hello")
            .keyUp(txtUsername, Keys.SHIFT)
            .doubleClick(txtUsername)
            .contextClick()
            .build();
        seriesOfActions.perform();
        driver.close();
    }
}
```



What is meant by XPath in Selenium. Explain XPath Absolute and XPath Relative.

XPath in Selenium is a technique that allows you to navigate the structure of a webpage's HTML. Using XPath in Selenium helps find elements that are not found by locators such as ID, class, or name. XPath is a syntax for finding elements on web pages, and XPath in Selenium can be used on both HTML and XML documents. There are other, more straightforward Selenium locators available that search for elements using tags or CSS class names. Yet they may not be sufficient to select all DOM elements of an HTML document. By using XPath, Selenium users can search for a page element in a more dynamic way. This capability gives testers the flexibility to work with locators so they are more advantageous.

Absolute XPath

Absolute XPath is the direct way to find the element. But the disadvantage of the absolute XPath is that if there are any changes made in the path of the element then that XPath fails.

The key characteristic of XPath is that it begins with the single forward slash(/), which means you can select the element from the root node.

Below is an example of an absolute XPath expression of the element shown in the below screen.

Absolute XPath: `/html/body/div[1]/div/div[2]/header/div/div[2]/a/img`

Relative XPath For Relative XPath, the path starts from the middle of the HTML DOM structure.

Unlike Absolute XPath, Relative XPath starts with the double forward slash (//), which means it can search the element anywhere on the webpage.

You can start from the middle of the HTML DOM structure with no need to write a long XPath.

Relative XPath: `//*[@id="block-perfecto-main-menu"]/ul/li[6]/a`

Demonstrate explicit and Implicit waits via a script.

```
package Assignment10;
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.Alert;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class Wait_Demonstration {

    public void setup() throws InterruptedException {

        System.setProperty("webdriver.chrome.driver", "C:\\Users\\hp\\Downloads\\chromedriver_
win32\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();

        driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS);
        driver.get("https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref_confirm");

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

        driver.switchTo().frame("iframeResult");

        driver.findElement(By.xpath("//button[text()='Try it']")).click();

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

        String alertText = AlertBox.getText();

        AlertBox.dismiss();

        System.out.println("Alert Text: "+alertText);

        driver.quit();

    }
}
```

```

package Assignment10;
import java.util.concurrent.TimeUnit;
import java.util.Date;
import org.openqa.selenium.By;
import org.openqa.selenium.Keys;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;
public class Wait_Demonstration {

    public void setup() throws InterruptedException {

        System.setProperty("webdriver.chrome.driver","C:\\Users\\hp\\Downloads\\chromedriver_
win32\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("https://google.com/");

        // Defining Explicit Wait
        WebDriverWait wait = new WebDriverWait(driver,20);

        WebElement searchTextBox;

        //Waiting until the search text box becomes visible
        searchTextBox=
wait.until(ExpectedConditions.visibilityOfElementLocated(By.name( "q")));

        driver.findElement(By.name("q")).sendKeys("cherry" + Keys.ENTER);

        WebElement firstSearchResult = driver.findElement(By.tagName("h3"));
        System.out.println("First Search Result with Cherry is :
"+firstSearchResult.getAttribute("textContent"));
        driver.quit();

    }
}

```

With the help of code snippets, explain how we can create right-click and mouse hover actions in Selenium.

```

package Assignment10;
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.firefox.FirefoxDriver;
import org.openqa.selenium.interactions.Actions;
public class RightClickMouseHover {

    public static void main(String[] args) {

```

```

System.setProperty("webdriver.chrome.driver", "C:\\Users\\HARSHAL\\Desktop\\selenium\\
chromedriver_win32 (1)\\chromedriver.exe");
WebDriver driver = new ChromeDriver();
driver.get("https://demoqa.com/buttons");
    System.out.println("demoqa webpage displayed");
driver.manage().window().maximize();
Actions actions = new Actions(driver);

WebElement btnElement = driver.findElement(By.id("rightClickBtn"));

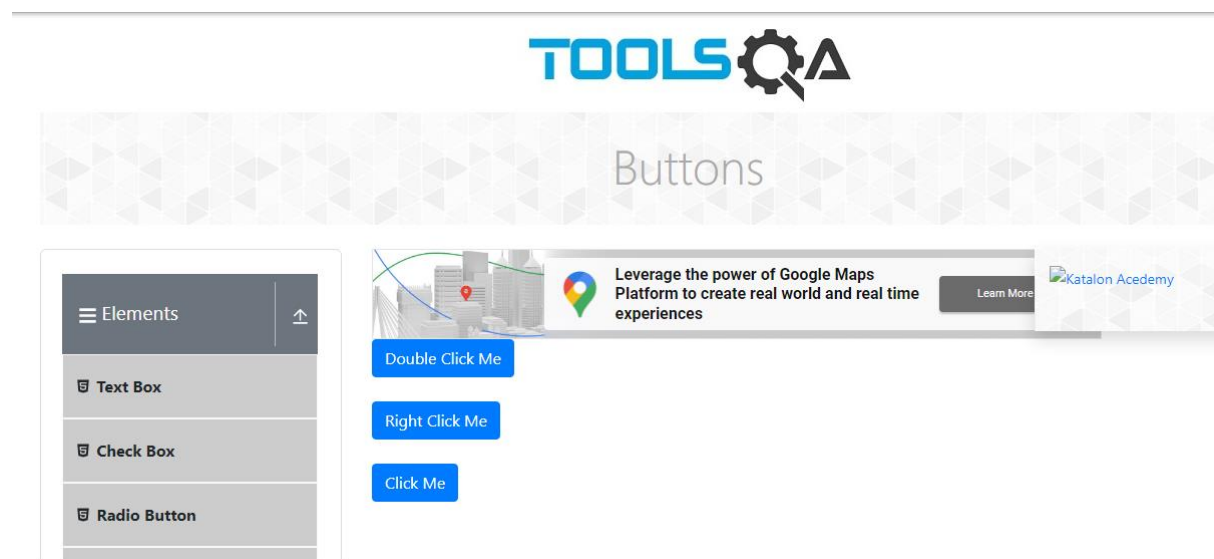
actions.contextClick(btnElement).perform();
System.out.println("Right click Context Menu displayed");

WebElement elementOpen =
driver.findElement(By.xpath("//*[id='rightclickItem']/div[1]"));
elementOpen.click();

driver.switchTo().alert().accept();
System.out.println("Right click Alert Accepted");

driver.close();

```




```

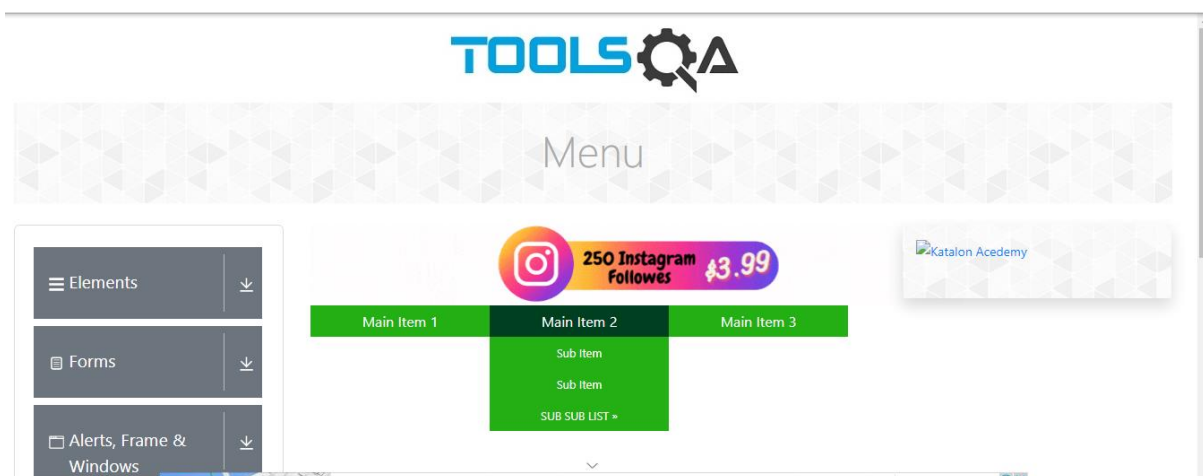
package Assignment10;
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.interactions.Actions;

public class MouseHover {

    public static void main(String[] args) {

        System.setProperty("webdriver.chrome.driver", "C:\\Users\\HARSHAL\\Desktop\\selenium\\chromedriver_win32 (1)\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("https://demoqa.com/menu/");
        System.out.println("demoqa webpage Displayed");
        driver.manage().window().maximize();
        driver.manage().timeouts().implicitlyWait(10000, TimeUnit.MILLISECONDS);
        Actions actions = new Actions(driver);
        WebElement menuOption = driver.findElement(By.xpath("//div[contains(text(),'Music')]"));
        actions.moveToElement(menuOption).perform();
        System.out.println("Done Mouse hover on 'Music' from Menu");
        WebElement subMenuOption = driver.findElement(By.xpath("//div[contains(text(),'Rock')]"));
        actions.moveToElement(subMenuOption).perform();
        System.out.println("Done Mouse hover on 'Rock' from Menu");
        WebElement selectMenuOption =
driver.findElement(By.xpath("//div[contains(text(),'Alternative')]"));
        selectMenuOption.click();
        System.out.println("Selected 'Alternative' from Menu");
        driver.close();
    }
}

```



Can we handle a windows-based Alerts in selenium.. Demonstrate.

Yes, it is possible to handle Windows based pop-ups in Selenium webdriver. Sometimes on clicking a link or a button, another window gets opened. It can be a pop up with information or an advertisement. The methods getWindowHandles and getWindowHandle are used to handle child windows. The getWindowHandles method stores all the handle ids of the opened windows in the form of Set data structure. The getWindowHandle method stores the handle id of the window in focus. Since the getWindowHandles method holds all the opened window handle ids, we can iterate through these handle ids with the iterator and next methods.

To switch to a specific window, switchTo().window() method can be used. The handle id of the window where we want to switch is passed as a parameter to this method.

```
package Assignment10;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import java.util.concurrent.TimeUnit;
import java.util.List;
import java.util.Set;
import java.util.Iterator;
public class WindowsBasedAlert {

    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();
        driver.manage().timeouts().implicitlyWait(15, TimeUnit.SECONDS);
        driver.get("https://secure.indeed.com/account/login");
        driver.findElement(By.id("login-google-button")).click();
        Set<String> s = driver.getWindowHandles();
        Iterator<String> i = s.iterator();
        String c = i.next();
        String p = i.next();
        driver.switchTo().window(c);
        System.out.println("Page title of child window: " + driver.getTitle());
        driver.switchTo().window(p);
        driver.quit();
    }
}
```

ChromeDriver was started successfully.

Page title of child window: Sign In | Indeed Accounts

Page title of parent window: Sign in – Google accounts