| **Name of Student:** Ajay Karthikesan | | | |
| --- | --- | --- | --- |
| **Roll Number:** 57 | | **Practical Number:** 3 | |
| **Aim of Practical:**  Implement Browser command and navigation Commands.  Implement the find element command  Demonstrate the Locator(id,css selector, path)  Demonstrate synchronization in selenium | | | |
| **DOP:** 13.10.23 | | **DOS:** 16.10.23 | |
| **CO Mapped:** - | **PO Mapped:** - | **Faculty Signature:** | **Marks:** |

## 

## Practical No. 3

**Aim:**

Implement Browser command and navigation Commands.

Implement the find element command

Demonstrate the Locator(id,css selector, path)

Demonstrate synchronization in selenium

**Theory:**

Selenium WebDriver - Browser Commands:

The very basic browser operations of WebDriver include opening a browser; perform few tasks and

then closing the browser.

Given are some of the most commonly used Browser commands for Selenium WebDriver.

1. Get Command:

Method: get(String arg0) : void

In WebDriver, this method loads a new web page in the existing browser window. It accepts

String as parameter and returns void.

2. Get Title Command:

Method: getTitle(): String

In WebDriver, this method fetches the title of the current web page. It accepts no parameter

and returns a String.

3. Get Current URL Command:

Method: getCurrentUrl(): String

In WebDriver, this method fetches the string representing the Current URL of the current web

page. It accepts nothing as parameter and returns a String value.

4. Close Command:

Method: close(): void

This method terminates the current browser window operating by WebDriver at the current

time. If the current window is the only window operating by WebDriver, it terminates the

browser as well. This method accepts nothing as parameter and returns void.

FindElement:

Selenium Find Element command takes in the By object as the parameter and returns an object

of type list WebElement in Selenium. By object in turn can be used with various locator

strategies such as find element by ID Selenium, Name, Class Name, XPATH etc.

What are Locators?

Locator is a command that tells Selenium IDE which GUI elements ( say Text Box, Buttons,

Check Boxes etc.) it needs to operate on. Identification of correct GUI elements is a prerequisite

to creating an automation script. But accurate identification of GUI elements is more difficult

than it sounds. Sometimes, you end up working with incorrect GUI elements or no elements at

all! Hence, Selenium provides a number of Locators to precisely locate a GUI element

Locator Strategy can be any of the following values.

● ID

● Selenium find element by Name

● Class Name

● Tag Name

● Link Text

● Partial Link Text

● XPATH

SOURCE CODE:

To synchronize between script execution and application, we need to wait after

performing appropriate actions. Let us look at the ways to achieve the same.

Thread.Sleep:

Thread.Sleep is a static wait and it is not a good way to use in scripts as it is sleep

without condition.

Explicit Waits:

An 'explicit wait,' waits for a certain condition to occur before proceeding further. It

is mainly used when we want to click or act on an object once it is visible.

Implicit wait:

Implicit wait is used in cases where the WebDriver cannot locate an object

immediately because of its unavailability. The WebDriver will wait for a specified

implicit wait time and it will not try to find the element again during the specified

time period.

Once the specified time limit is crossed, the webDriver will try to search the element

once again for one last time. Upon success, it proceeds with the execution; upon

failure, it throws exceptions.

It is a kind of global wait which means the wait is applicable for the entire driver.

Hence, hardcoding this wait for longer time periods will hamper the execution time.

**Code:**

File: Demo.java

package vesitajayk57.stqa\_practical3;

import java.time.Duration;

import java.util.Scanner;

import org.openqa.selenium.By;

import org.openqa.selenium.ElementNotInteractableException;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

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

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

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

public class Demo {

private final WebDriver webDriver;

private final Scanner scanner = new Scanner(System.in);

Demo(WebDriver webDriver) {

this.webDriver = webDriver;

}

public void runNavCommands() {

System.out.println("Running Navigation Commands demo...");

webDriver.get("https://www.linuxmint.com");

webDriver.get("https://www.archlinux.org");

webDriver.navigate().forward();

webDriver.navigate().back();

webDriver.navigate().refresh();

waitForEnterKey("Navigation Commands");

}

public void runSomeBrowserCommands() {

System.out.println("Running Browser Commands demo...");

webDriver.get("https://www.selenium.dev/selenium/web/web-form.html");

webDriver.manage().timeouts().implicitlyWait(Duration.ofMillis(500));

System.out.println("Title of Webpage = " + webDriver.getTitle());

System.out.println("Current URL = " + webDriver.getCurrentUrl());

WebElement textBox = webDriver.findElement(By.name("my-text"));

WebElement submitButton = webDriver.findElement(By.cssSelector("button"));

textBox.sendKeys("Selenium");

submitButton.click();

WebElement message = webDriver.findElement(By.id("message"));

System.out.println("Text of element with id 'message' = " + message.getText());

waitForEnterKey("Browser Commands");

}

public void runLocatorDemo() {

System.out.println("Running Locator demo...");

final String htmlDocumentUrl = "file:///home/ajayk/Documents/MCA/3SEM/Practicals/STQA/3Practical/stqa-practical3/stqa-practical3/index.html";

webDriver.get(htmlDocumentUrl);

webDriver.manage().timeouts().implicitlyWait(Duration.ofMillis(500));

WebElement lname = webDriver.findElement(By.id("lname"));

WebElement fnameId = webDriver.findElement(By.cssSelector("#newsletter"));

WebElement inputF = webDriver.findElement(By.xpath("//input[@value='f']"));

System.out.println("Tag name of element with id 'lname' = " + lname.getTagName());

System.out.println("Newsletter is checked(css selector) = " + fnameId.isSelected());

System.out.println("Value of property 'value' (xpath) = " + inputF.getDomProperty("value"));

waitForEnterKey("Locator");

}

public void runSynchronizationDemo() {

System.out.println("Running Synchronization demo...");

runThreadWaitDemo();

runImplicitWaitDemo();

runExplicitSyncDemo();

runExplicitSyncWithOptionsDemo();

waitForEnterKey("Synchronization");

}

private void runThreadWaitDemo() {

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

System.out.println("Waiting for 2 seconds(via Thread.sleep())");

try {

Thread.sleep(2000);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

private void runImplicitWaitDemo() {

final String htmlDocumentUrl = "file:///home/ajayk/Documents/MCA/3SEM/Practicals/STQA/3Practical/stqa-practical3/stqa-practical3/index.html";

System.out.println("Waiting for 3 seconds(implicit wait)");

webDriver.get(htmlDocumentUrl);

webDriver.manage().timeouts().implicitlyWait(Duration.ofSeconds(3));

System.out.println("Current URL = " + webDriver.getCurrentUrl());

}

private void runExplicitSyncDemo() {

webDriver.get("https://www.selenium.dev/selenium/web/dynamic.html");

WebElement revealed = webDriver.findElement(By.id("revealed"));

Wait<WebDriver> wait = new WebDriverWait(webDriver, Duration.ofSeconds(2));

webDriver.findElement(By.id("reveal")).click();

wait.until(d -> revealed.isDisplayed());

revealed.sendKeys("Displayed");

System.out.println("[ExplicitSync]Value of property 'value' = " + revealed.getDomProperty("value"));

}

private void runExplicitSyncWithOptionsDemo() {

webDriver.get("https://www.selenium.dev/selenium/web/dynamic.html");

WebElement revealed = webDriver.findElement(By.id("revealed"));

Wait<WebDriver> wait = new FluentWait<>(webDriver).withTimeout(Duration.ofSeconds(2))

.pollingEvery(Duration.ofMillis(300)).ignoring(ElementNotInteractableException.class);

webDriver.findElement(By.id("reveal")).click();

wait.until(d -> {

revealed.sendKeys("Displayed");

return true;

});

System.out.println("[ExplicitSyncWithOptions]Value of property 'value' = " + revealed.getDomProperty("value"));

}

private void waitForEnterKey(String demoName) {

while (true) {

System.out.println("Press enter after taking screenshots of " + demoName + " demo.");

String entered = scanner.nextLine();

if (entered.equals("")) {

break;

}

}

}

public void end() {

webDriver.quit();

scanner.close();

}

}

File: HelloSelinium.java

package vesitajayk57.stqa\_practical3;

import org.openqa.selenium.chrome.ChromeDriver;

public class HelloSelenium {

public static void main(String[] args) {

Demo demo = new Demo(new ChromeDriver());

demo.runSomeBrowserCommands();

demo.runNavCommands();

demo.runLocatorDemo();

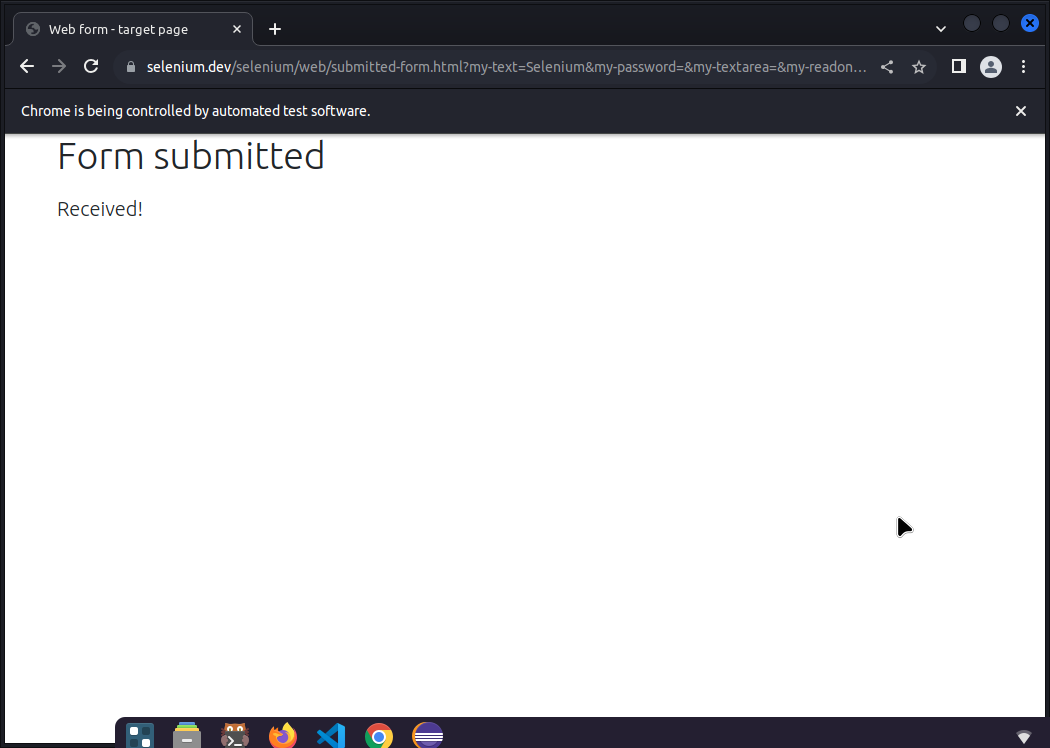
demo.runSynchronizationDemo();

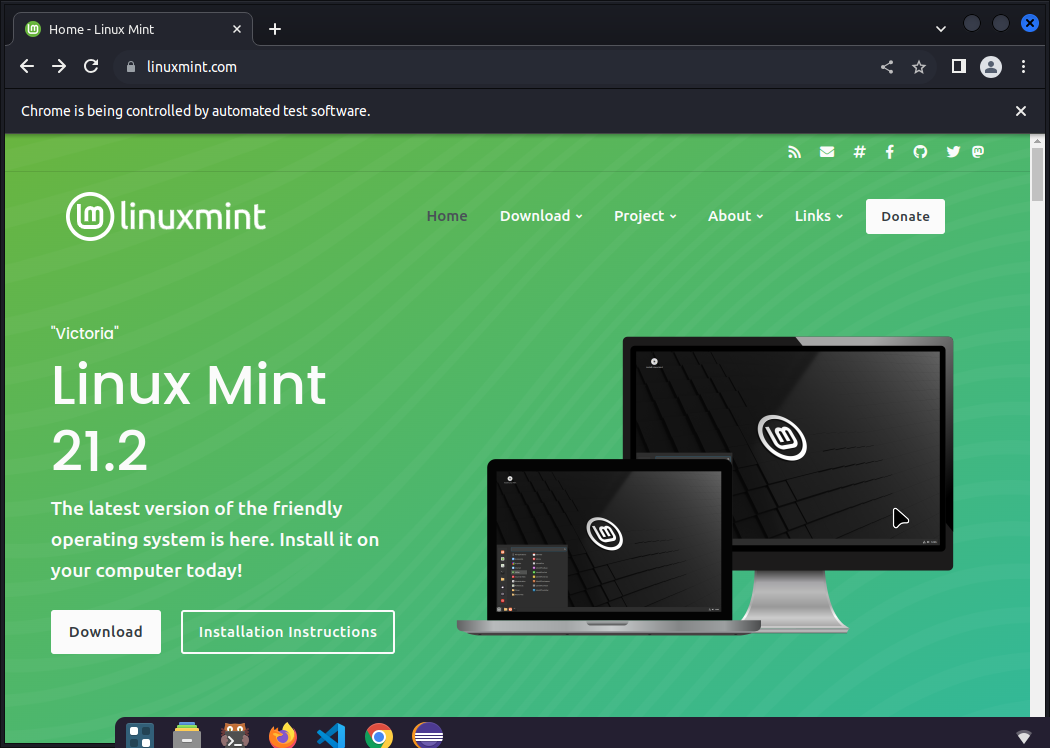
demo.end();

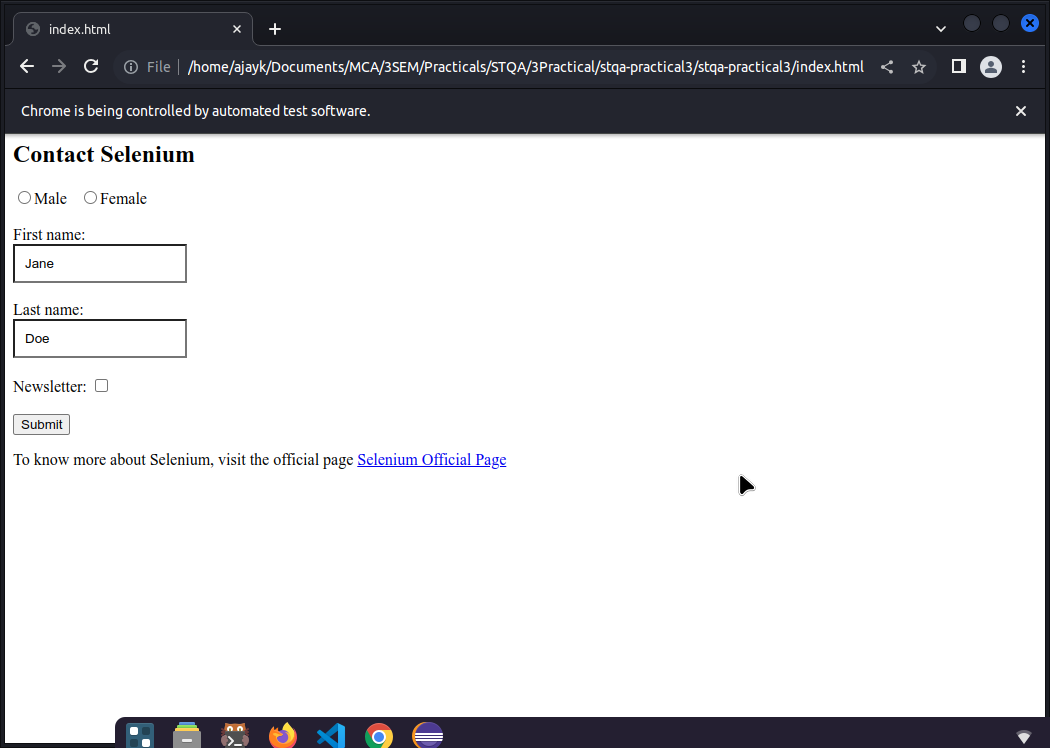
}

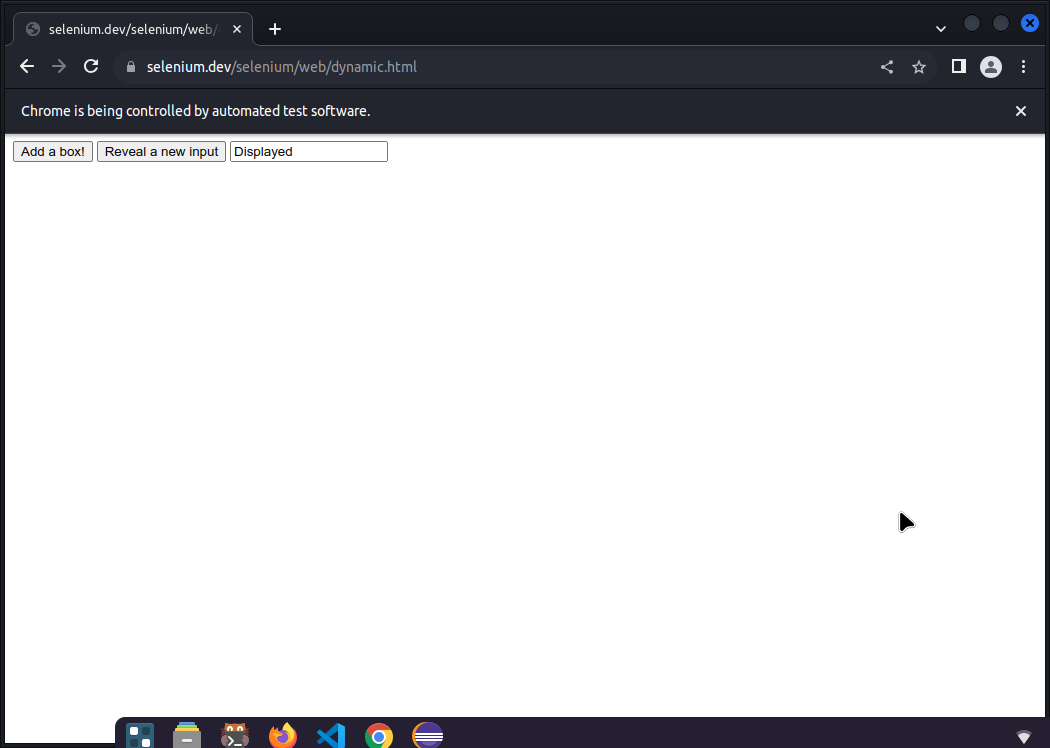
}

**Output:**











**Conclusion:**

I completed the following tasks:

Implement Browser command and navigation Commands.

Implement the find element command

Demonstrate the Locator(id,css selector, path)

Demonstrate synchronization in selenium