**Software Testing Quality Assurance Lab**

A Project Report Submitted in Fulfillment

of the Degree of

**MASTER**

**In**

**COMPUTER APPLICATION**

**Year 2022-2023**

By

**Ms. MANJAREKAR SWARUPA SATYAVAN GEETA**

**(Seat No-806075)**

**(Application Id-170714)**

Under the Guidance of

**Prof. Richa Kulal**

****

Institute of Distance and Open Learning

Vidya Nagari, Kalina, Santacruz East – 400098.

University of Mumbai

**PCP Center**

**Satish Pradhan Dnyanasadhana College,**

**Thane.**



**Institute of Distance and Open Learning**

Vidya Nagari, Kalina, Santacruz East – 400098.

***CERTIFICATE***

This is to certify that, this project report entitled **“Software Testing Quality Assurance Lab”** is a record of work carried out by **Ms. Manjarekar Swarupa Satyavan Geeta (Seat no-806075),** student of **MCA semester-III** class and is submitted to University of Mumbai, in partial fulfilment of the requirement for the award of the degree of **Master in Computer Application**. The project report has been approved.

\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

Guide External Examiner Coordinator – M.C.A

**Declaration**

I declare that this written submission represents my ideas in my own words and where other's ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

-----------------------------------------

(Signature)

Ms. Manjarekar Swarupa Satyavan Geeta

Seat No-806075

Date:

Place:

**ACKNOWLEDGMENT**

After the completion of this work, words are not enough to express my feelings about all those who helped me to reach my goal; feeling above this is my indebtedness to the almighty for providing me this moment in my life.

It’s a great pleasure and moment of immense satisfaction for me to express my profound gratitude to my project guide, **Prof. Richa Kulal** whose constant encouragement enabled me to work enthusiastically. His perpetual motivation, patience and excellent expertise in discussion during progress of dissertation work have benefited me to an extent, which is beyond expression. His depth and breadth of knowledge of Engineering field made me realize that theoretical knowledge always help to develop efficient operational software, which is a blend of all core subjects of the field. The completion of this project would not have been possible without his encouragement, patient guidance and constant support.

I would like to thank all staff members for their valuable cooperation and permitting me to work in the computer labs.

Special thanks to my colleagues and friends for providing me useful comments, suggestions and continuous encouragement.

Finally, I thanks my family members, for their support and endurance during this work.

----------------------------------

**Ms. Manjarekar Swarupa Satyavan Geeta**

(Seat No: 806075)

**Index**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr No** | **Aim** | **Date** | **Sign** |
| 1. | To write a simple test case. |  |  |
| 2. | Implementing Web Drivers on Multiple Browser i.e chrome |  |  |
| 3. | Implementing handling multiple frames |  |  |
| 4. | Implementing Selenium WebDriver - Browser Commands |  |  |
| 5. | Implementing Selenium WebDriver - find element command ,Locator(id, css selector, Xpath), Input Box ,Buttons, Submit Buttons |  |  |
| 6. | Demonstrate different types of alerts |  |  |
| 7. | Demonstrate CheckBox and Radio Button in Selenium WebDriver |  |  |
| 8. | Demonstrate synchronization in selenium(Implicit wait) |  |  |
| 9. | Demonstrate: Select Value from DropDown using Selenium  Webdriver. |  |  |
| 10. | Demonstrate action classes using Selenium Webdriver(Mouse Events). |  |  |

**Practical -1**

**Aim-** To write a simple test case.

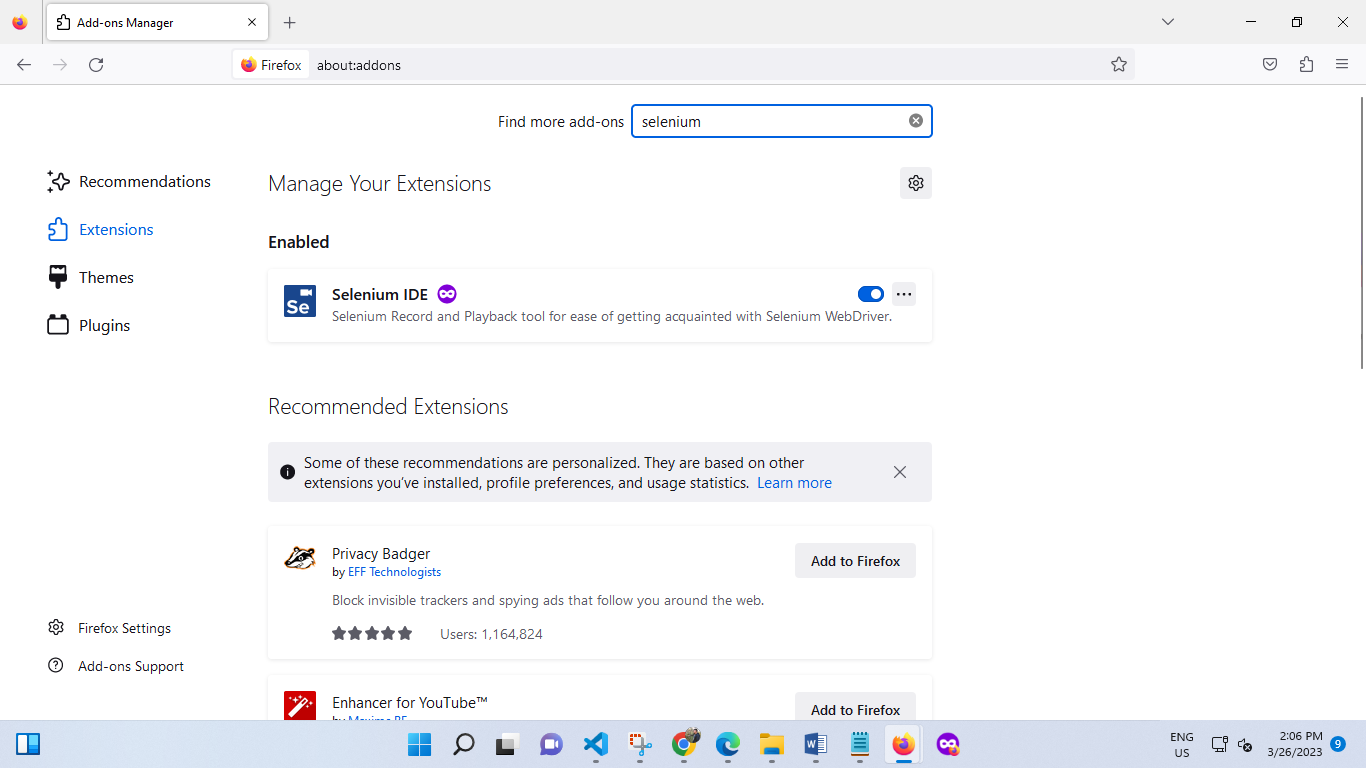
Step1: Open the Firefox browser.

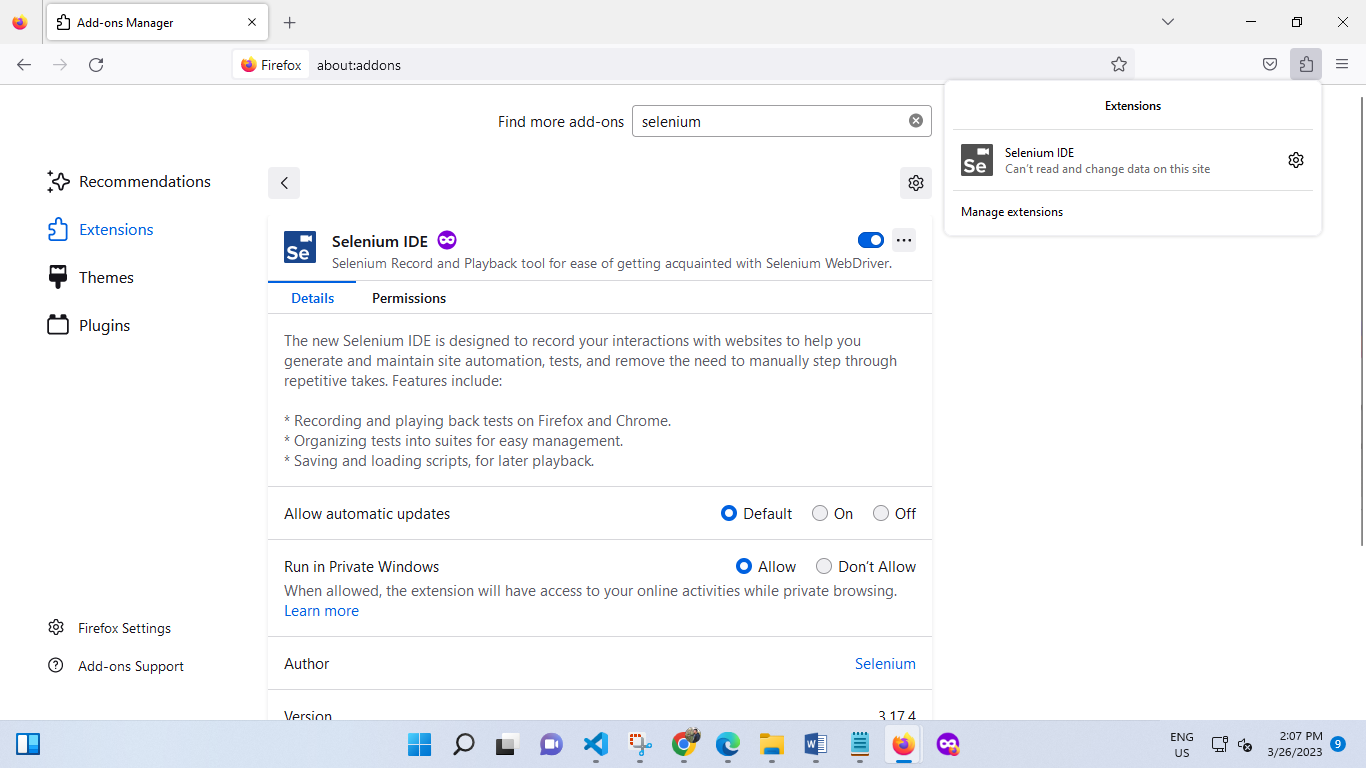
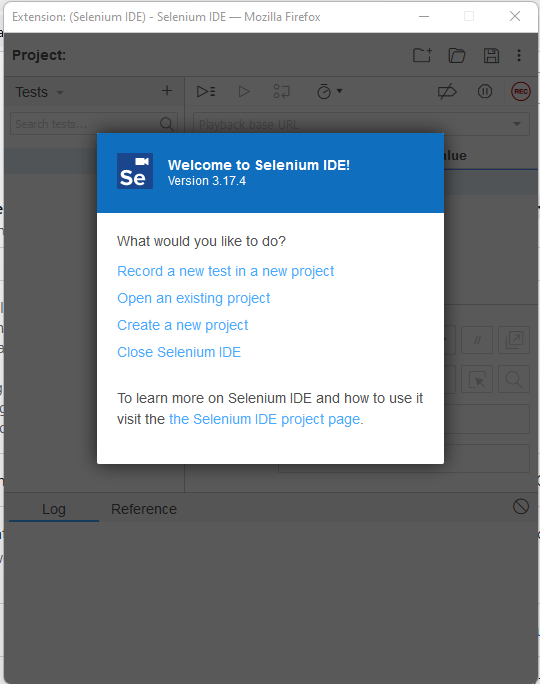
Step 2: Click on the menu in the top right corner.

Step 3: Click on Add-ons in the drop-down box.

Step 4: Click on find more add-ons and type “Selenium IDE.”

Step 5: Click on Add to Firefox.



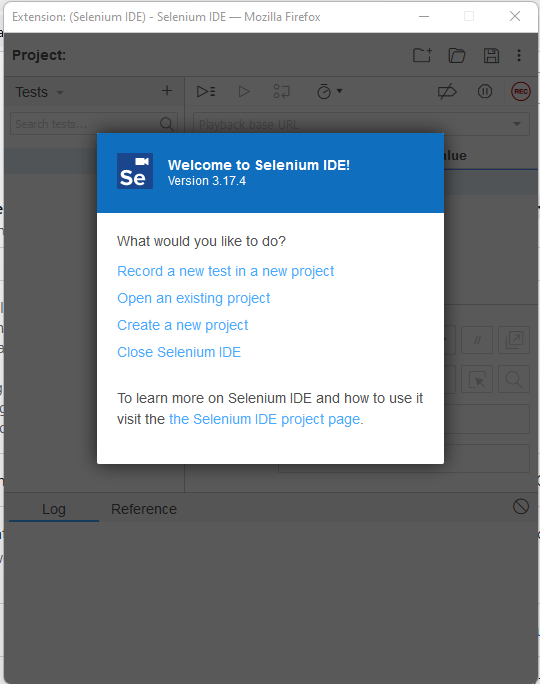
 

let’s create our first test. Consider the following use case for the tutorial:

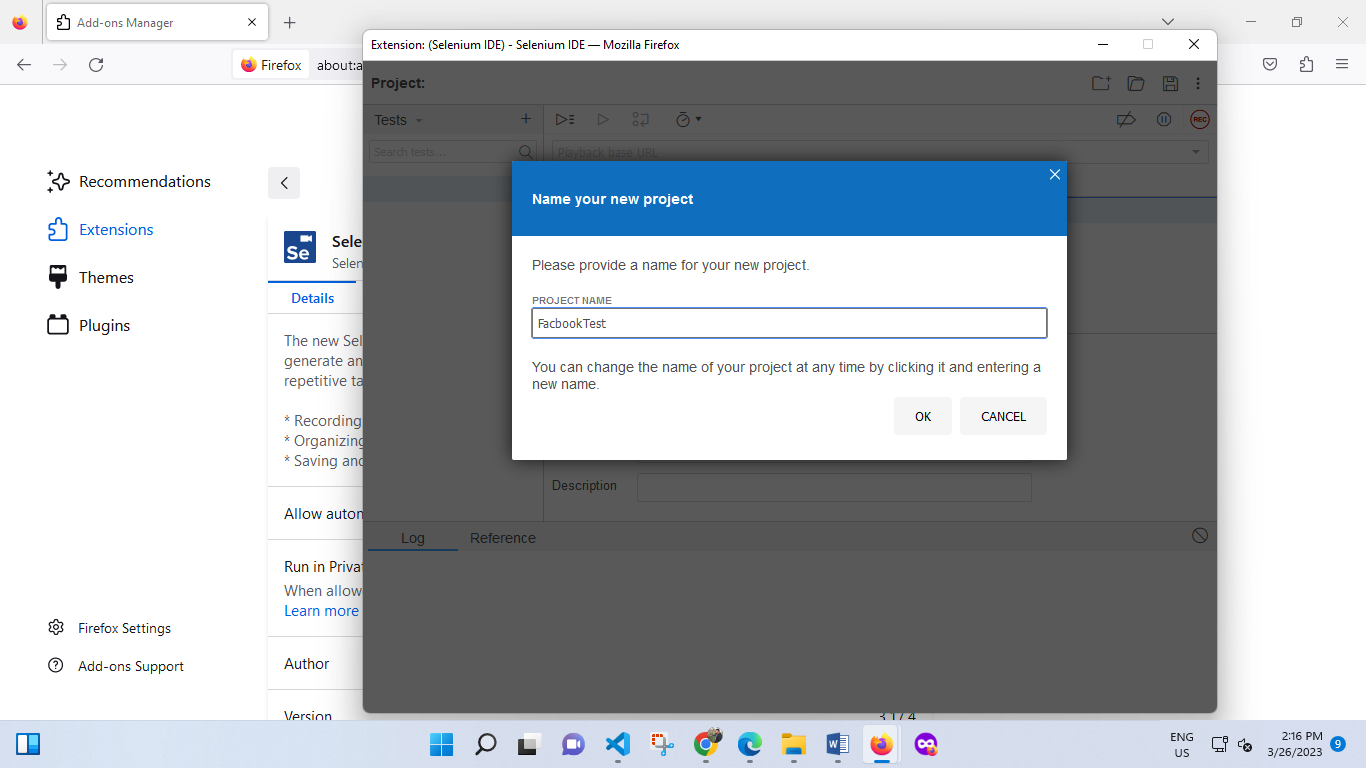
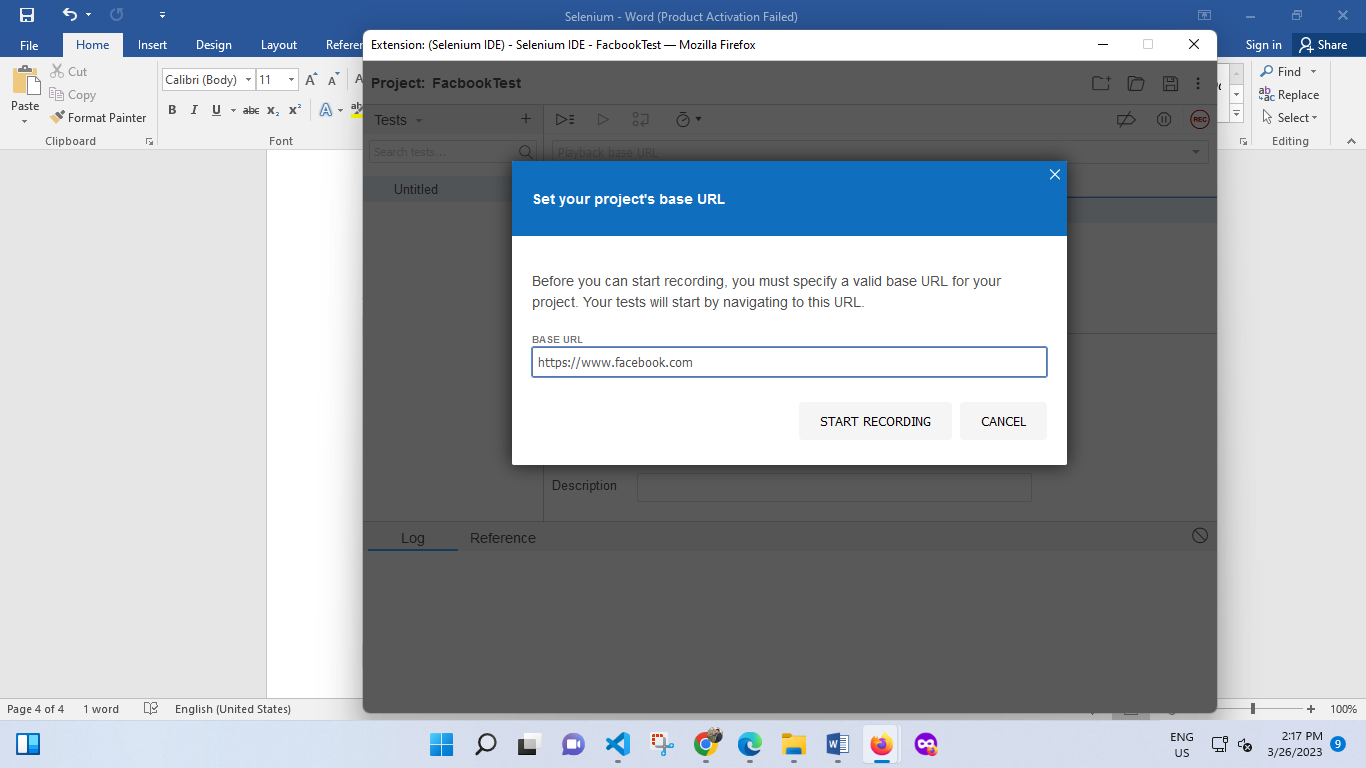
* Navigate to <https://www.facebook.com>
* Provide a dummy userID and password
* Log in with these credentials
* Assert title of the application

**Step 1:** Launch your Firefox menu and open the Selenium IDE plugin.

**Step 2:** Select “Record a test in a new project.” Provide the name for your test.



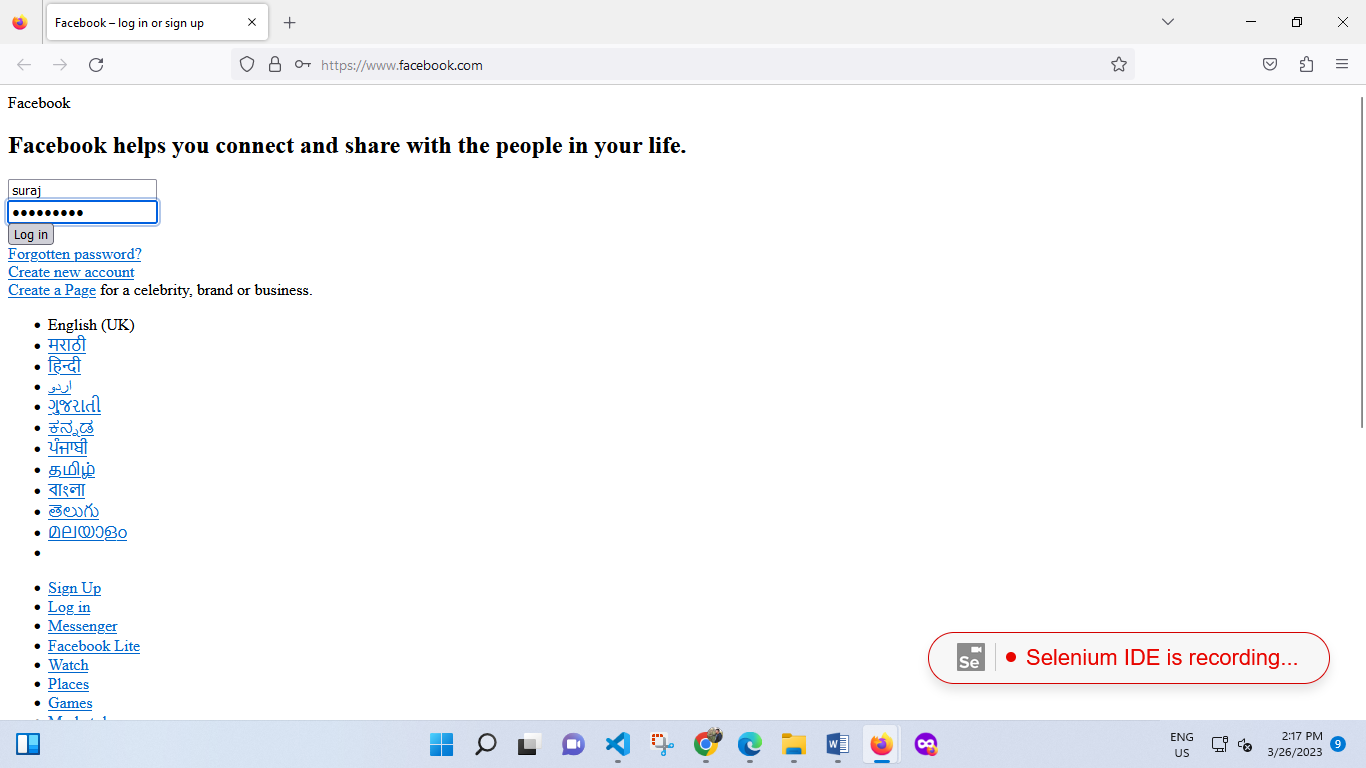
**Step 3:** Provide a link to the Facebook webpage. The IDE starts recording by navigating to this web page. To continue, click on, OK.

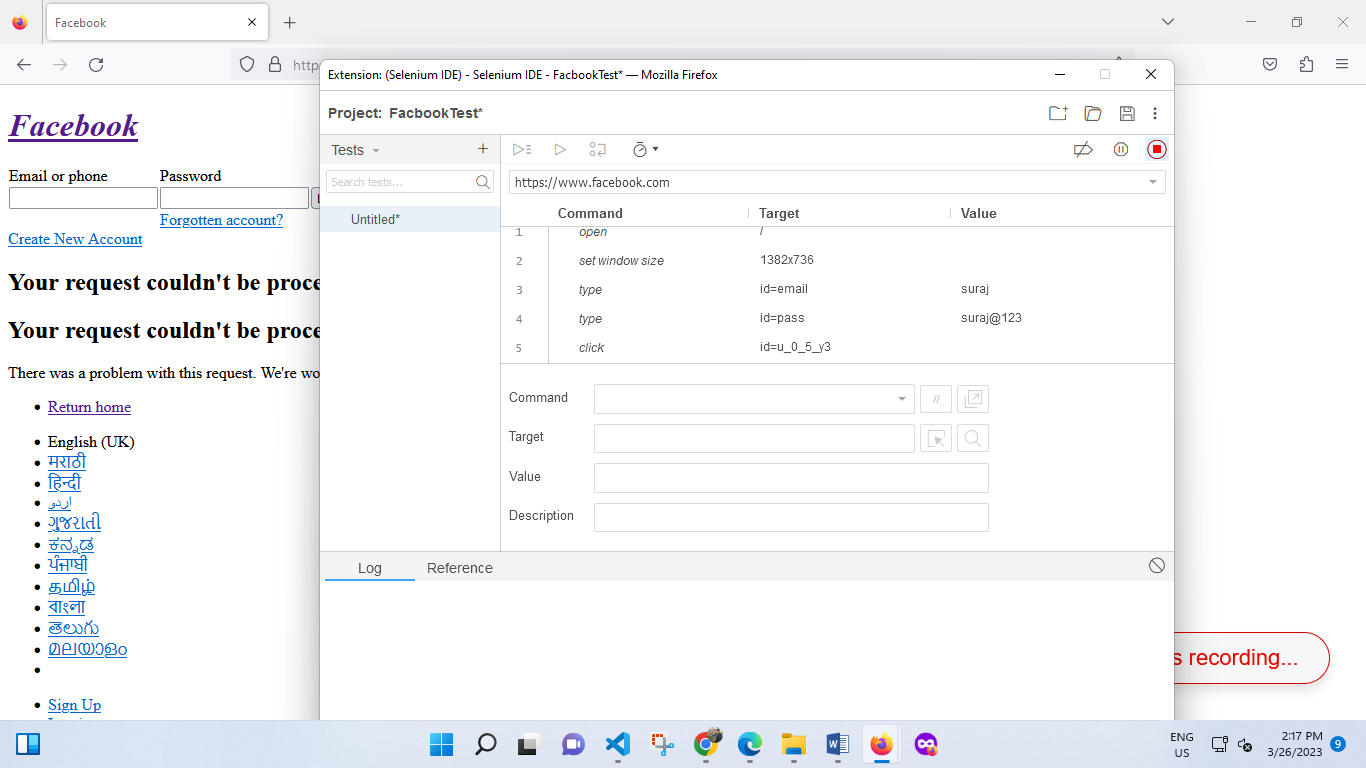
**Step 4:** Once the website opens, type “username1” in the username field and a dummy password for the password field.

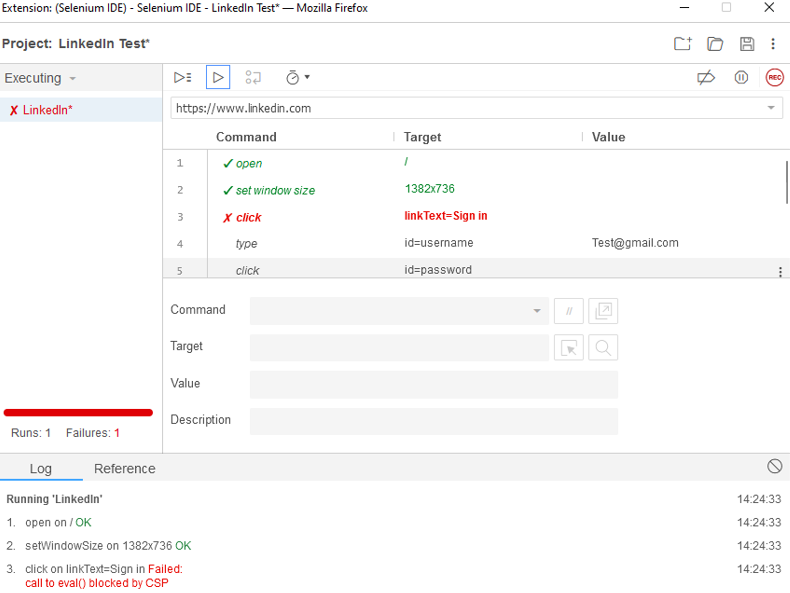
**Step 5:** Click on “Log In.”

**Step 6:** Now, we verify the title of our application. To do that, Right click>>Selenium IDE>>Assert title. As soon as this is done, a test step would be appended in the IDE editor.



Now you can go back to the IDE editor and click on the Stop icon on the top right corner. With this, we’ve successfully recorded our test case. Once the recording is stopped, the editor will look something like this:





**Practical -2**

**AIM -** Implementing Web Drivers on Multiple Browser i.e chrome

**Source code-**

package com.sdet;

import org.openqa.selenium.By;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class App {

public static void main(String[] args) {

System.setProperty("webdriver.chrome.driver","C:\\Users \\chromedriver\_win32\\chromedriver.exe");

// Instantiate a ChromeDriver class.

WebDriver driver=new ChromeDriver();

// Launch Website

driver.navigate().to("http://www.news.yahoo.com/");

//Maximize the browser

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

//Scroll down the webpage by 5000 pixels

JavascriptExecutor js = (JavascriptExecutor)driver;

js.executeScript("scrollBy(0, 5000)");

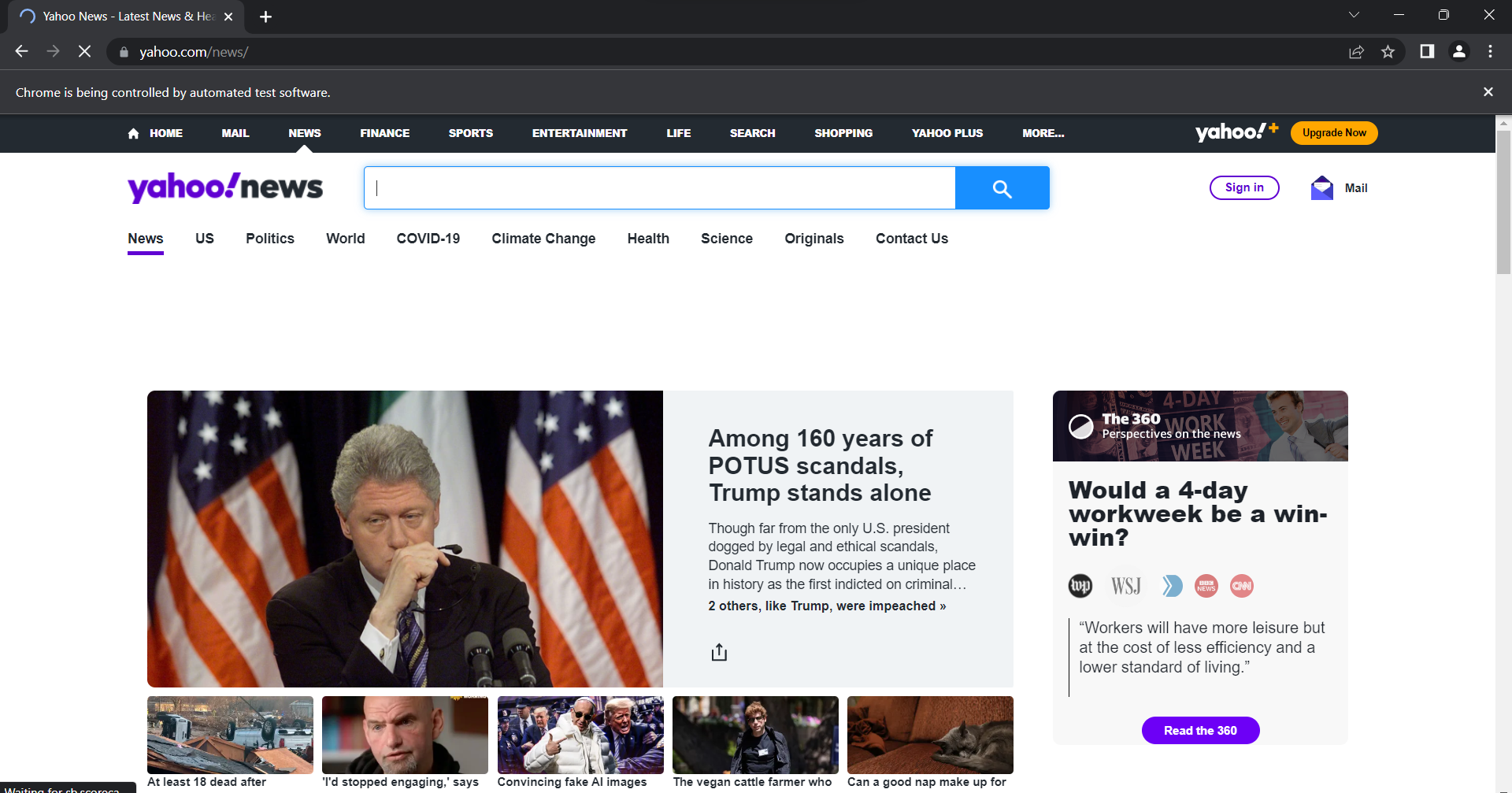
// Click on the Search button

driver.findElement(By.linkText("Core Java")).click();

}

}

**Output-**



**Practical -3**

**AIM -** Implementing handling multiple frames

**Source code-**

package multiFrames;

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;

public class multiFrames {

public static void main(String[] args) {

System.setProperty("webdriver.chrome.driver", "C:\\Users \\chromedriver\_win32\\chromedriver.exe");

WebDriver driver = new ChromeDriver();

String url = "https://the-internet.herokuapp.com/frames";

driver.get(url);

driver.manage().timeouts().implicitlyWait(5, TimeUnit.SECONDS);

// identify element

driver.findElement(By.linkText("Nested Frames")).click();

// switch to frame with frame name and identify inside element

driver.switchTo().frame("frame-bottom");

WebElement l = driver.findElement(By.cssSelector("body"));

System.out.println("Bottom frame text: " +l.getText());

// switch to main page

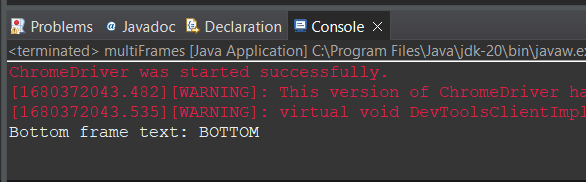
driver.switchTo().defaultContent();

driver.quit();

}

}

**Output-**

****

**Practical -4**

**AIM -** Implementing Selenium WebDriver - Browser Commands

**Source code-**

package selbrowsercmd;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class selbrowsercmdclass {

public static void main(String[] args) {

// System Property for Chrome Driver

System.setProperty("webdriver.chrome.driver",

"C:\\Users \\chromedriver\_win32\\chromedriver.exe");

// Instantiate a ChromeDriver class.

WebDriver driver = new ChromeDriver();

// Storing the Application Url in the String variable

String url = ("https://www.google.co.in/");

// Launch the ToolsQA WebSite

driver.get(url);

// Storing Title name in the String variable

String title = driver.getTitle();

// Storing Title length in the Int variable

int titleLength = driver.getTitle().length();

// Printing Title & Title length in the Console window

System.out.println("Title of the page is : " + title);

System.out.println("Length of the title is : " + titleLength)

// Storing URL in String variable

String actualUrl = driver.getCurrentUrl();

if (actualUrl.equals("https://www.google.co.in/")) {

System.out.println("Verification Successful - The correct Url is opened.");

} else {

System.out.println("Verification Failed - An incorrect Url is opened.");

}

// Storing Page Source in String variable

String pageSource = driver.getPageSource();

// Storing Page Source length in Int variable

int pageSourceLength = pageSource.length();

// Printing length of the Page Source on console

System.out.println("Total length of the Pgae Source is : " + pageSourceLength);

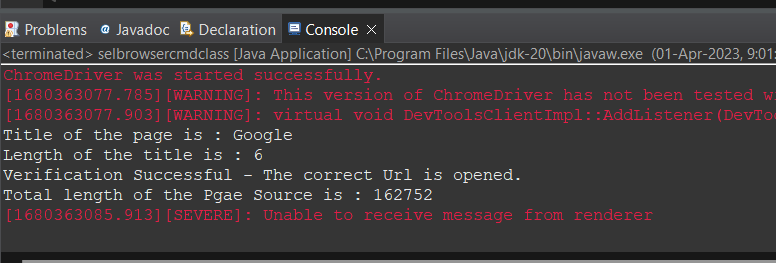
// Closing browser

driver.close();

}

}

**Output-**

****

**Practical -5**

**AIM -** Implementing Selenium WebDriver - find element command, Locator (id, css selector, Xpath), Input Box, Buttons, Submit Buttons

**Source code-**

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.\*;

public class Form {

public static void main(String[] args) {

// declaration and instantiation of objects/variables

System.setProperty("webdriver.chrome.driver","G:\\chromedriver.exe");

WebDriver driver = new ChromeDriver();

String baseUrl = "http://demo.guru99.com/test/login.html";

driver.get(baseUrl);

// Get the WebElement corresponding to the Email

Address(TextField)

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

// Get the WebElement corresponding to the Password Field

WebElement password = driver.findElement(By.name("passwd"));

email.sendKeys("abcd@gmail.com");

password.sendKeys("abcdefghlkjl");

System.out.println("Text Field Set");

// Deleting values in the text box

email.clear();

password.clear();

System.out.println("Text Field Cleared");

// Find the submit button

WebElement login = driver.findElement(By.id("SubmitLogin"));

// Using click method to submit form

email.sendKeys("abcd@gmail.com");

password.sendKeys("abcdefghlkjl");

login.click();

System.out.println("Login Done with Click");

//using submit method to submit the form. Submit used on password

field

driver.get(baseUrl);

driver.findElement(By.id("email")).sendKeys("abcd@gmail.com");

driver.findElement(By.name("passwd")).sendKeys("abcdefghlkjl");

driver.findElement(By.id("SubmitLogin")).submit();

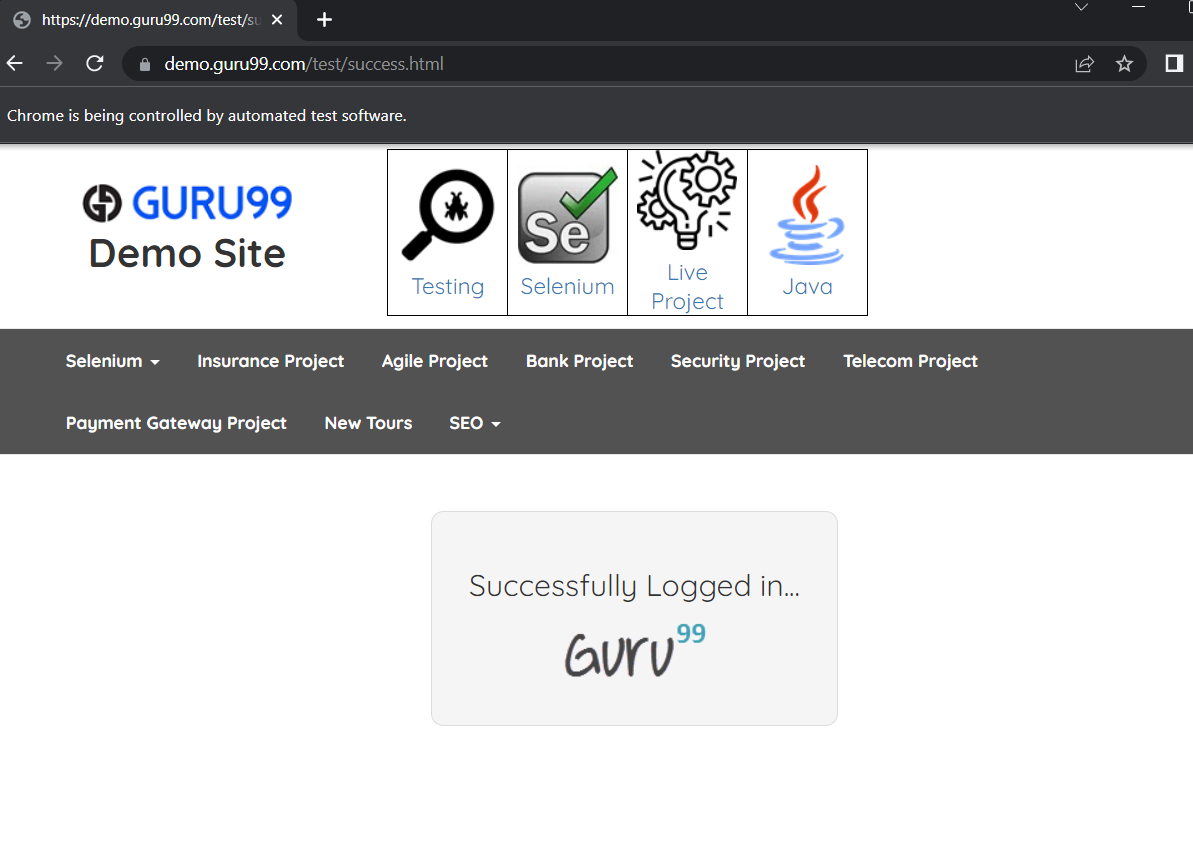
System.out.println("Login Done with Submit");

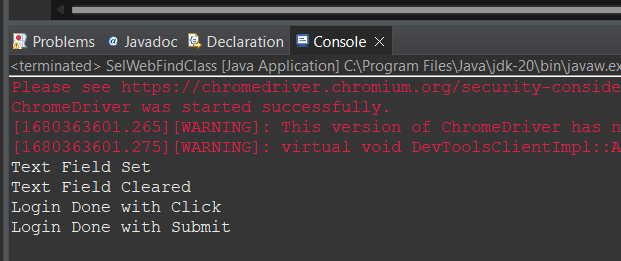
//driver.close();

}

}

**Output-**

****

****

**Practical -6**

**AIM -** Demonstrate different types of alerts.

**Source code-**

package AlertsProjectPkg;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.NoAlertPresentException;

import org.openqa.selenium.Alert;

public class AlertsProjectClass {

public static void main(String[] args) throws NoAlertPresentException, InterruptedException {

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\chromedriver\_win32\\chromedriver.exe");

WebDriver driver = new ChromeDriver();

// Alert Message handling

driver.get("http://demo.guru99.com/test/delete\_customer.php");

driver.findElement(By.name("cusid")).sendKeys("53928");

driver.findElement(By.name("submit")).submit();

// Switching to Alert

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

// Capturing alert message.

String alertMessage = driver.switchTo().alert().getText();

// Displaying alert message

System.out.println(alertMessage);

Thread.sleep(5000);

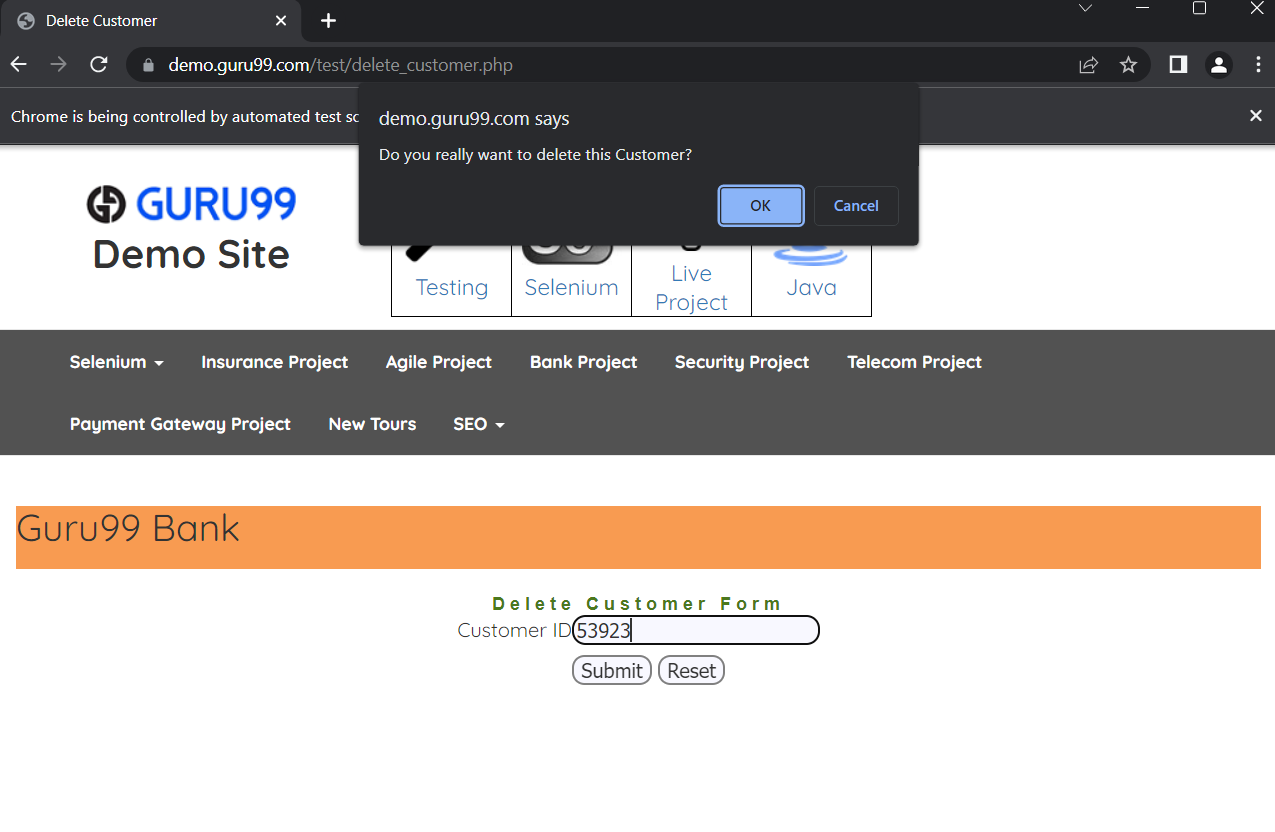
// Accepting alert

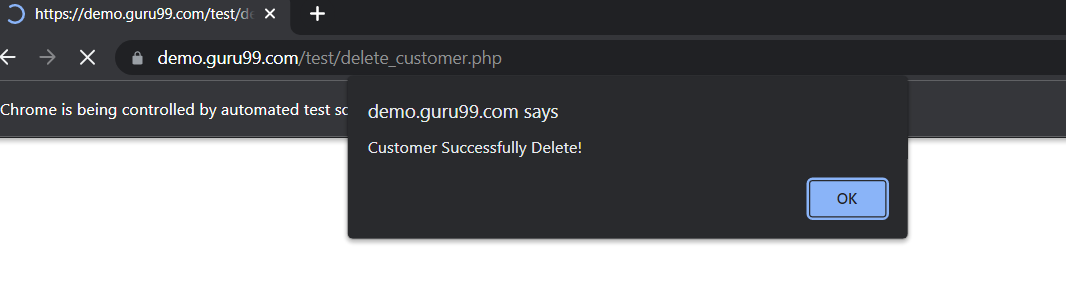
alert.accept();

}

}

**Output-**

****

****

**Practical -7**

**AIM -** Demonstrate CheckBox and Radio Button in Selenium WebDriver

**Source code-**

package CheckboxRadioBtnPkg;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.\*;

public class CheckboxRadioBtnclass {

public static void main(String[] args) {

// declaration and instantiation of objects/variables

System.setProperty("webdriver.chrome.driver",

"C:\\Users \\chromedriver\_win32\\chromedriver.exe");

WebDriver driver = new ChromeDriver();

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");

}

// Selecting Checkbox and using isSelected Method

driver.get("http://demo.guru99.com/test/facebook.html");

WebElement chkFBPersist = driver.findElement(By.id("persist\_box"));

for (int i = 0; i < 2; i++) {

chkFBPersist.click();

System.out.println("Facebook Persists Checkbox Status is -" + chkFBPersist.isSelected());

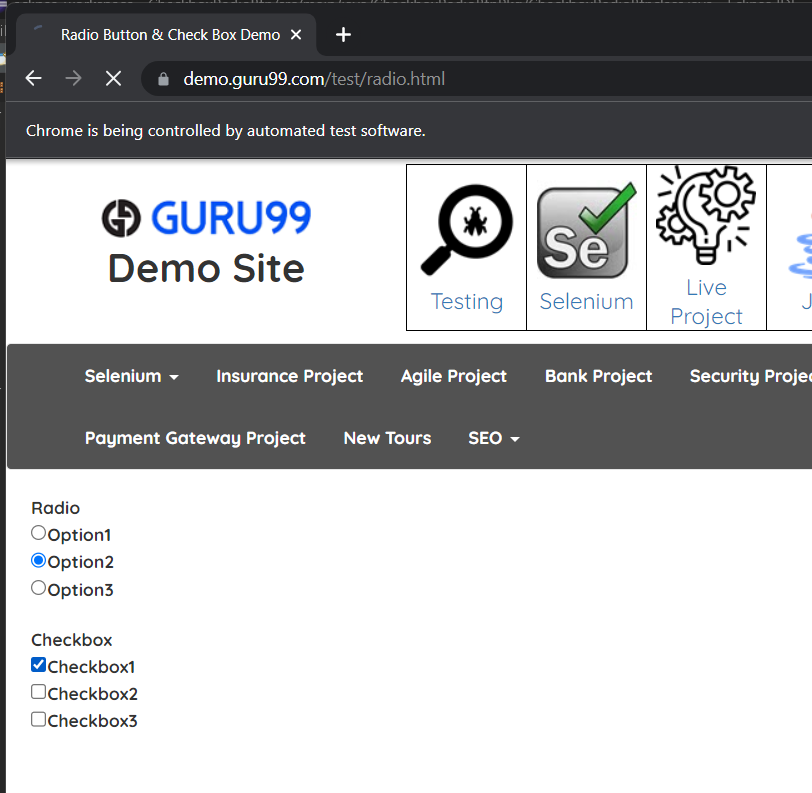
}

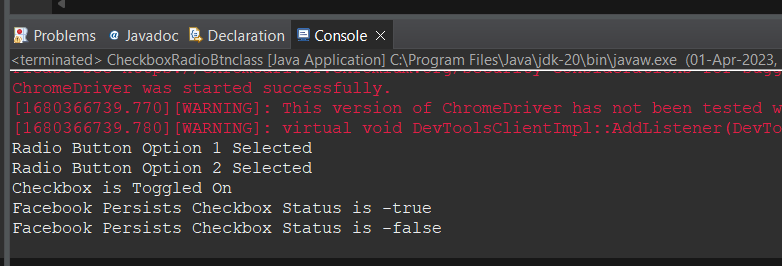
// driver.close();

}

}

**Output-**

****

****

**Practical -8**

**AIM -** Demonstrate: Select Value from DropDown using Selenium Webdriver.

**Source code-**

package syncSelPkg;

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.support.ui.ExpectedConditions;

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

public class syncSelClass {

public static void main(String[] args) throws InterruptedException

{

System.setProperty("webdriver.chrome.driver",

"C:\\Users \\chromedriver\_win32\\chromedriver.exe");

WebDriver driver = new ChromeDriver();

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

driver.manage().deleteAllCookies();

driver.manage().timeouts().pageLoadTimeout(40,TimeUnit.SECONDS);

driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS);

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

driver.findElement(By.xpath("//input[@id='login-username']")).sendKeys("JavaTpoint.com");

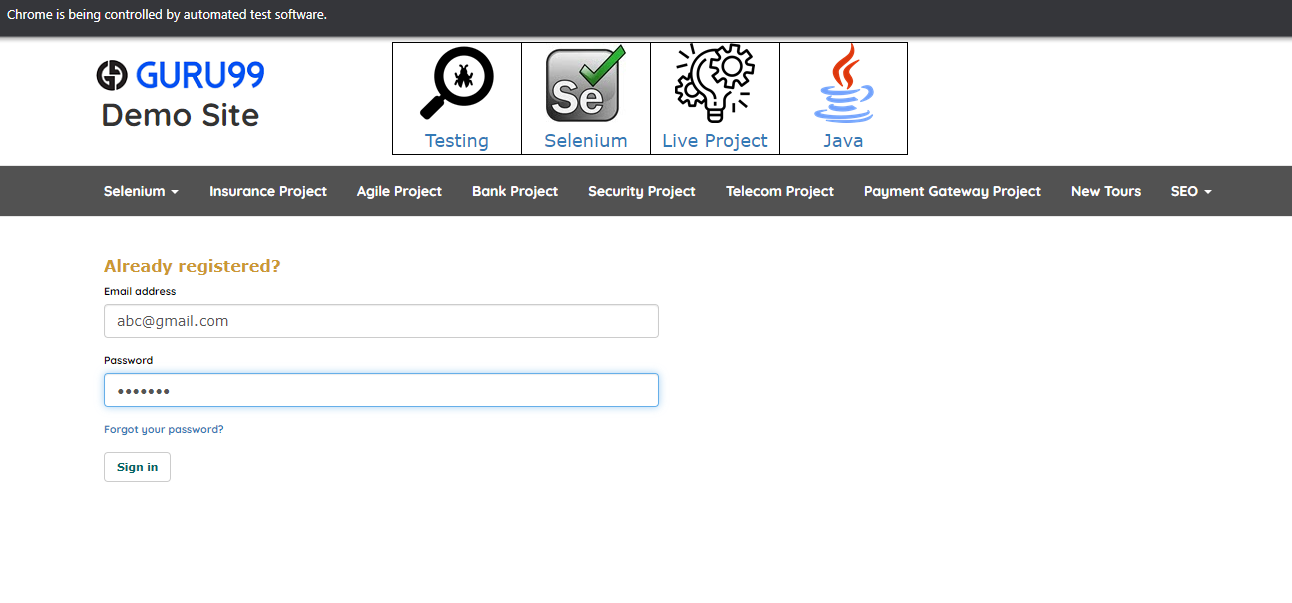
Thread.sleep(1000);

driver.findElement(By.xpath("//input[@id='login-signin']")).click();

}

}

**Output-**



**Practical -9**

**AIM -** Demonstrate: Select Value from DropDown using Selenium Webdriver.

**Source code-**

package valueDropdownPkg;

import java.util.List;

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 ValueDropdownClass {

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

// Creating instance of Chrome driver

System.setProperty("webdriver.chrome.driver",

"C:\\Users \\chromedriver\_win32\\chromedriver.exe");

WebDriver driver = new ChromeDriver();

// Step#2- Launching URL

driver.get("https://demoqa.com/select-menu");

// Maximizing window

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

// Step#3- Selecting the dropdown element by locating its id

Select select = new Select(driver.findElement(By.id("oldSelectMenu")));

// Step#4- Printing the options of the dropdown

// Get list of web elements

List<WebElement> lst = select.getOptions();

// Looping through the options and printing dropdown options

System.out.println("The dropdown options are:");

for (WebElement options : lst)

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

// Step#5- Selecting the option as 'Purple'-- selectByIndex

System.out.println("Select the Option by Index 4");

select.selectByIndex(4);

System.out.println("Select value is: " + select.getFirstSelectedOption().getText());

// Step#6- Selecting the option as 'Magenta'-- selectByVisibleText

System.out.println("Select the Option by Text Magenta");

select.selectByVisibleText("Magenta");

System.out.println("Select value is: " + select.getFirstSelectedOption().getText());

// Step#7- Selecting an option by its value

System.out.println("Select the Option by value 6");

select.selectByValue("6");

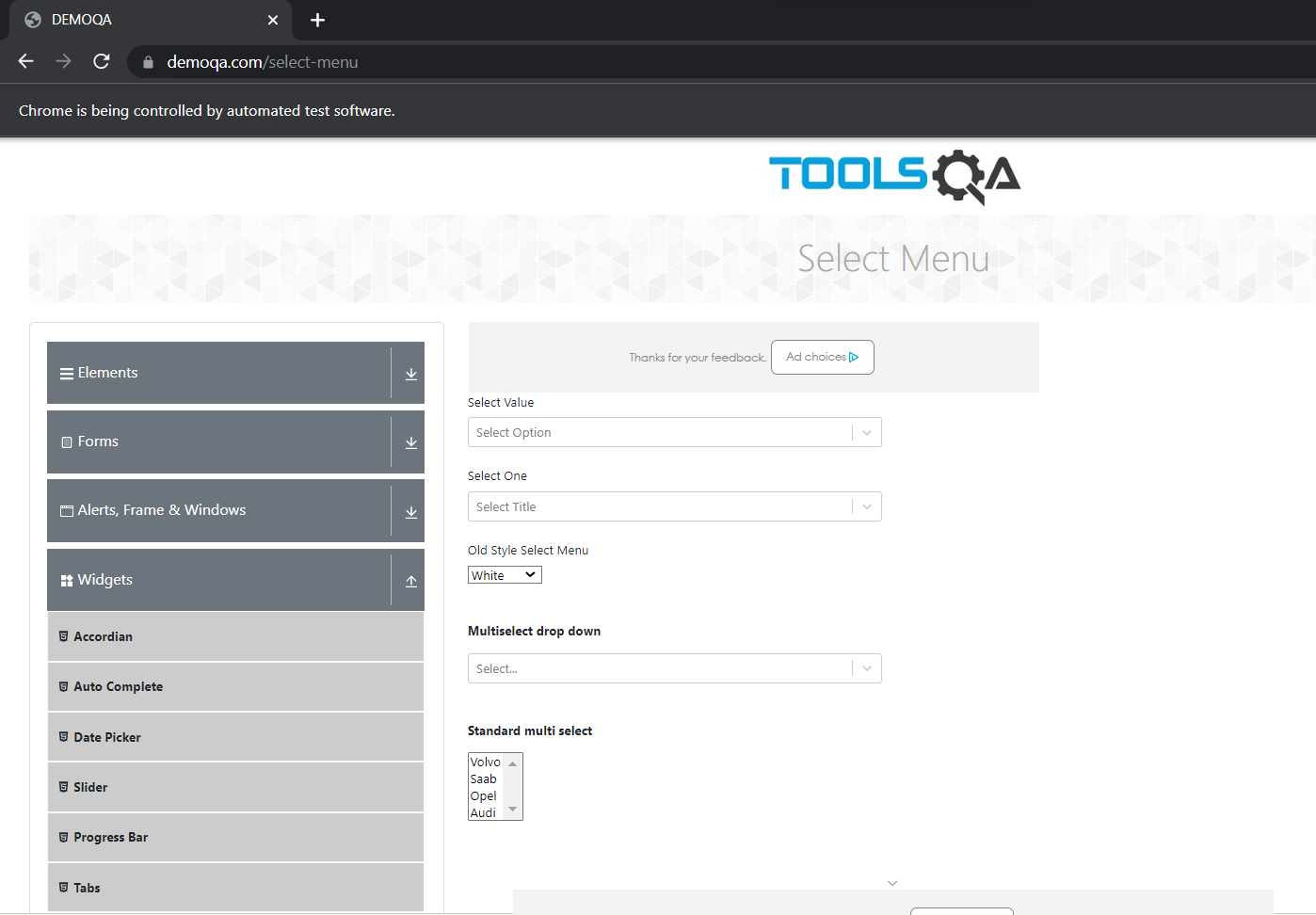
System.out.println("Select value is: " + select.getFirstSelectedOption().getText());

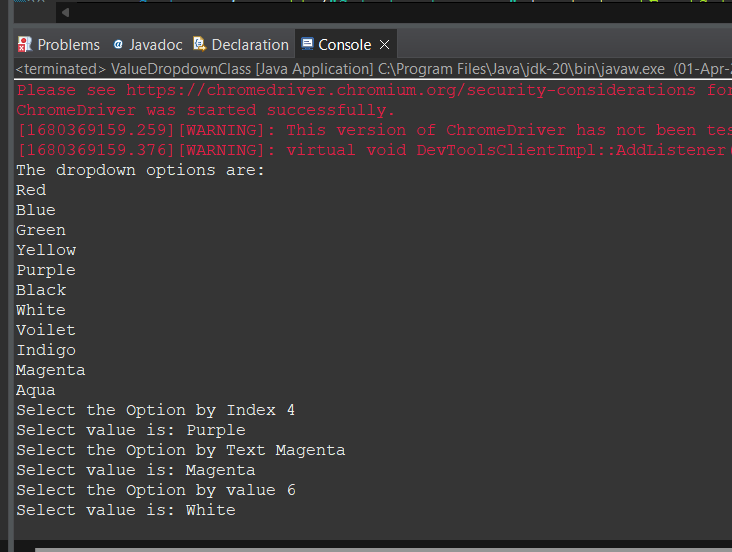
driver.quit();

}

}

**Output-**

****

****

**Practical -10**

**AIM -** Demonstrate action classes using Selenium Webdriver(Mouse Events).

**Source code-**

package mouseEventpkg;

import org.openqa.selenium.\*;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.openqa.selenium.interactions.Action;

import org.openqa.selenium.interactions.Actions;

public class MouseEventClass {

public static void main(String[] args) {

String baseUrl = "http://demo.guru99.com/test/newtours/";

System.setProperty("webdriver.chrome.driver",

"C:\\Users \\chromedriver\_win32\\chromedriver.exe");

WebDriver driver = new ChromeDriver();

driver.get(baseUrl);

WebElement link\_Home = driver.findElement(By.linkText("Home"));

WebElement td\_Home = driver

.findElement(By

.xpath("//html/body/div"

+ "/table/tbody/tr/td"

+ "/table/tbody/tr/td"

+ "/table/tbody/tr/td"

+ "/table/tbody/tr"));

Actions builder = new Actions(driver);

Action mouseOverHome = builder

.moveToElement(link\_Home)

.build();

String bgColor = td\_Home.getCssValue("background-color");

System.out.println("Before hover: " + bgColor);

mouseOverHome.perform();

bgColor = td\_Home.getCssValue("background-color");

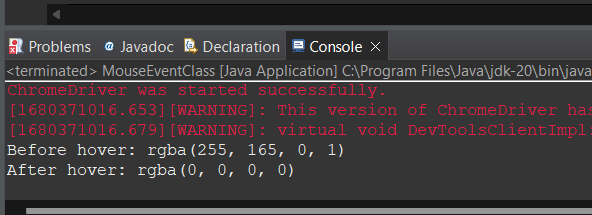
System.out.println("After hover: " + bgColor);

driver.close();

}

}

**Output-**

****