**SOFTWARE TESTING & QUALITY ASSURANCE LAB**

A Lab Manual Submitted in Fulfilment

of the Degree of

**MASTER**

**In**

**COMPUTER APPLICATION**

**Year 2022-2023**

By

**Mr. Gupta Vaibhav Ravindra Kalyani**

**(Seat No.:-806062)**

**(Application Id:- 169591)**

Under the Guidance of

**Asst. Prof. Ms. 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 Lab Manual entitled **“Software Testing & Quality Assurance Lab”** is a record of work carried out by **Mr. Gupta Vaibhav Ravindra Kalyani (Seat no:- 806062),** 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 Lab Manual has been approved.

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

Guide External Examiner Coordinator – M.C.A

**Approval of Lab Manual**

This is to certify that the Lab Manual entitled **“Software Testing & Quality Assurance Lab”,** for **Master in Computer Application** submitted to University of Mumbai by **Mr. Gupta Vaibhav Ravindra Kalyani (Seat no:-806062)** abonafide student of Institute of Distance and Open Learning, Vidyanagari,Kalina, Santracruz East has been approvedfor the award of **Master in Computer Application**.

**Examiner**

**1.**

**2.**

Date:

Place:

**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)

**Mr. Gupta Vaibhav Ravindra Kalyani**

**Seat No:- 806062**

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 Practical guide, **Asst. Prof. Ms. Richa Kulal.** whose constant encouragement enabled me to work enthusiastically. Her perpetual motivation, patience and excellent expertise in discussion during progress of dissertation work have benefited me to an extent, which is beyond expression. Her 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 her 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.

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

**Mr. Gupta Vaibhav Ravindra Kalyani**

(Seat No:- 806062)

**INDEX**

|  |  |  |
| --- | --- | --- |
| Index No. | Topic Name | Page No. |
| 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) |  |

**Experiment 1**

**Aim : To write Simple test case.**

**What is a Test Case?**

A **Test Case** is a set of actions executed to verify a particular feature or functionality of your software application. A Test Case contains test steps, test data, precondition, postcondition developed for specific test scenario to verify any requirement. The test case includes specific variables or conditions, using which a testing engineer can compare expected and actual results to determine whether a software product is functioning as per the requirements of the customer.

**Test Scenario Vs Test Case**

Test scenarios are rather vague and cover a wide range of possibilities.

Testing is all about being very specific.

For a [Test Scenario](https://www.guru99.com/test-scenario.html): Check Login Functionality there many possible test cases are:

* Test Case 1: Check results on entering valid User Id & Password
* Test Case 2: Check results on entering Invalid User ID & Password
* Test Case 3: Check response when a User ID is Empty & Login Button is pressed, and many more

This is nothing but a Test Case.

**The format of Standard Test Cases**

Below is a format of a standard login Test cases example.

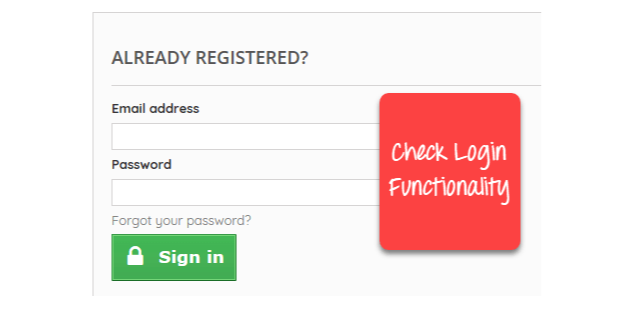
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test** | **Test** |  | **Test Steps** | | **Test** | **Expect** | **Actual** | **Pass** |  |
| **Case** | **Case** |  |  |  | **Data** | **ed** | **Result** | **/Fail** |  |
| **ID** | **Descri** |  |  |  |  | **Results** | **s** |  |  |
|  | **ption** |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| TU01 | Check | 1. | Go to site | | Userid | User | As | Pass |  |
|  | Custo | = | should | Expect |  |  |
|  | mer |  | http://demo. | | guru99 | Login | ed |  |  |
|  |  |  |  |  |  |
|  | Login |  | guru99.com |  | Passw | into an |  |  |  |
|  | 2. | Enter UserId | |  |  |  |
|  | with | ord = | applicati |  |  |  |
|  | valid | 3. | Enter Password | | pass99 | on |  |  |  |
|  | Data | 4. | Click Submit | |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| TU02 | Check | 1. | Go to site | | Userid | User | As | Pass |  |
|  | Custo | = | should | Expect |  |  |
|  | mer |  | http://demo. | | guru99 | not | ed |  |  |
|  | Login |  | guru99.com |  | Passw | Login |  |  |  |
|  | 2. | Enter UserId | |  |  |  |
|  | with | ord = | into an |  |  |  |
|  | invalid | 3. | Enter Password | | glass9 | applicati |  |  |  |
|  | Data | 4. | Click Submit | | 9 | on |  |  |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |



This entire table may be created in Word, Excel or any other Test management tool. That’s all to Test Case Design

**How to Write Test Cases in Manual Testing**

Let’s create a Test Case for the scenario: Check Login Functional

**Step 1)** A simple test case to explain the scenario would be

|  |  |
| --- | --- |
| **Test Case #** | **Test Case Description** |
|  |  |
| 1 | Check response when valid email and password is entered |

**Step 2)** Test the Data.

In order to execute the test case, you would need Test Data. Adding it below

|  |  |  |
| --- | --- | --- |
| **Test** | **Test Case Description** | **Test Data** |
| **Case #** |  |  |
|  |  |  |
| 1 | Check response when valid email | Email: guru99@email.com |
|  | and password is entered | Password: lNf9^Oti7^2h |

Identifying test data can be time-consuming and may sometimes require creating test data afresh. The reason it needs to be documented. **Step 3)** Perform actions.

In order to execute a test case, a tester needs to perform a specific set of actions on the AUT. This is documented as below:

|  |  |  |  |
| --- | --- | --- | --- |
| **Test** | **Test Case Description** | **Test Steps** | **Test Data** |
| **Case #** |  |  |  |
|  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | Check response when valid | 1) | Enter | Email: |
|  | email and password is entered | Email | | guru99@email.c |
|  |  | Address | | om |
|  |  | 2) | Enter | Password: |
|  |  | Password | | lNf9^Oti7^2h |
|  |  | 3) | Click Sign |  |
|  |  | in |  |  |

Documented steps will help him and also facilitate reviews by other stakeholders.

**Step 4)** Check behavior of the AUT.

The goal of test cases in software testing is to check behavior of the AUT for an expected result. This needs to be documented as below

|  |  |  |  |
| --- | --- | --- | --- |
| **Test** | **Test Case Description** | **Test Data** | **Expected** |
| **Case** |  |  | **Result** |
| **#** |  |  |  |
|  |  |  |  |
| 1 | Check response when valid | Email: | Login should |
|  | email and password is entered | guru99@email.c | be successful |
|  |  | om |  |
|  |  | Password: |  |
|  |  | lNf9^Oti7^2h |  |

During test execution time, the tester will check expected results against actual results and assign a pass or fail status

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test** | **Test Case** | | **Test Data** | | **Expected** | **Actual** | | **Pass/** | |
| **Cas** | **Description** | |  | | **Result** | **Result** | | **Fail** | |
| **e #** |  | |  | |  |  | |  | |
|  |  | |  | |  |  | |  | |
| 1 | | Check response when valid | | Email: | | | Login should | |
|  | | email and password is entered | | guru99@email.c | | | be successful | |
|  | |  | | om | | |  | |
|  | |  | | Password: | | |  | |
|  | |  | | lNf9^Oti7^2h | | |  | |

**Step 5)** That apart your test case -may have a field like,

Pre – Condition which specifies things that must be in place before the test can run. For our test case, a pre-condition would be to have a browser installed to have access to the site under test. A test case may also include Post – Conditions which specifies anything that applies after the test case completes. For our test case, a postcondition would be time & date of login is stored in the database

**Experiment 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)");

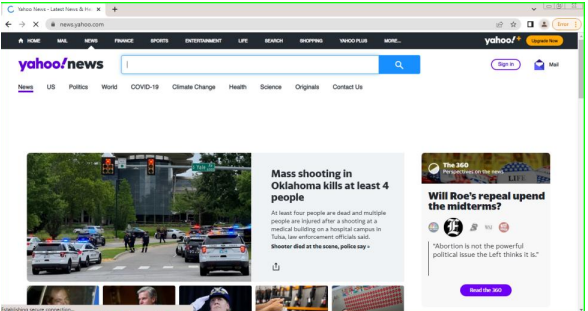
// Click on the Search button

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

}

}

**Output :**

****

**Experiment 3**

**AIM:- Implementing handling multiple frames**

**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** java.util.concurrent.TimeUnit;

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

**import** org.openqa.selenium.chrome.ChromeOptions;

**public** **class** Selenium {

**public** **static** **void** main(String[] args) {

System.*setProperty*("webdriver.chrome.driver","C:\\Users\\spdc\\Desktop\\MCA\_Praticals\\selenium\\chromedriver.exe");

ChromeOptions option = **new** ChromeOptions();

option.addArguments("--remote-allow-origins=\*");

WebDriver driver = **new** ChromeDriver(option);

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

driver.get(url);

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

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

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

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

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

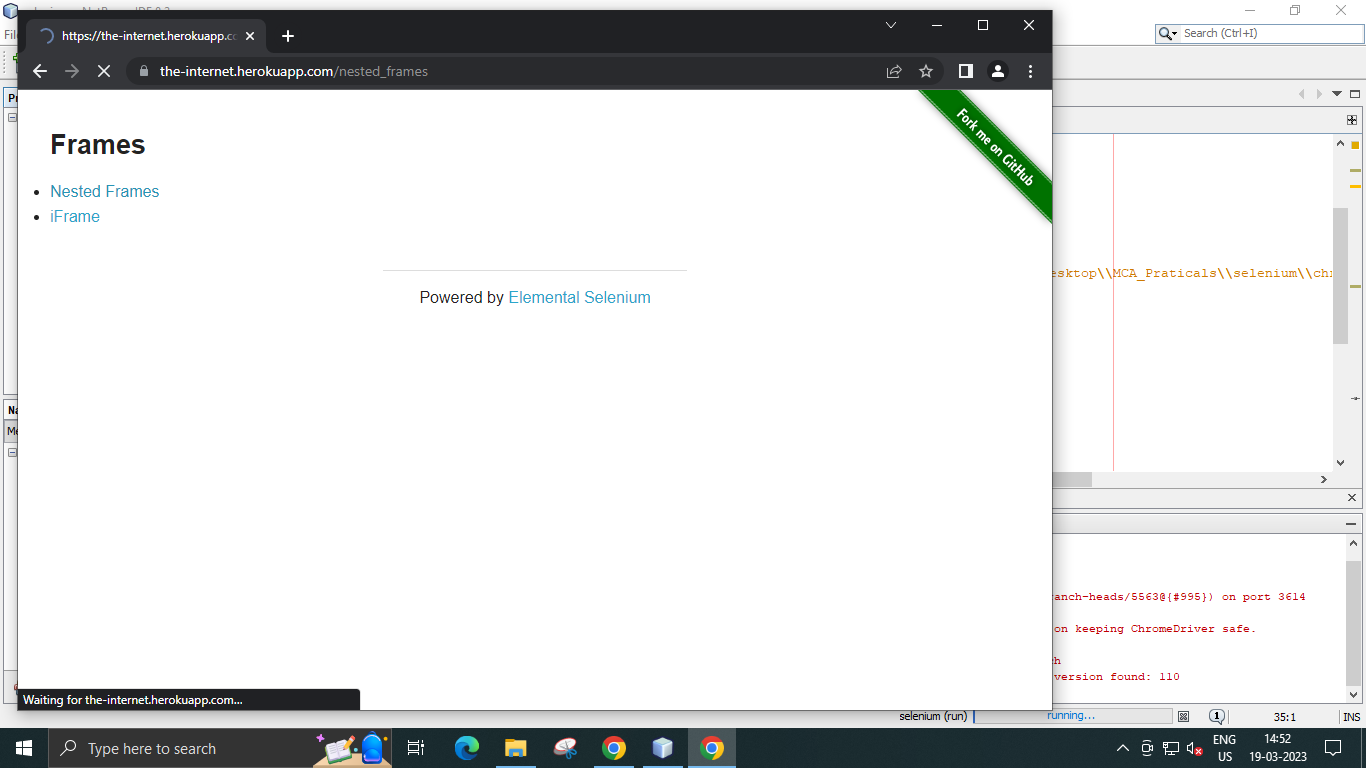
driver.switchTo().defaultContent();

driver.quit();

}

}

**OUTPUT :**



A screenshot of a computer

Description automatically generated

Graphical user interface, text, application

Description automatically generated

**Experiment 4**

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

**package** selenium;

**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** org.openqa.selenium.chrome.ChromeDriver;

**import** org.openqa.selenium.chrome.ChromeOptions;

**import** org.openqa.selenium.\*;

**public** **class** Selenium {

**public** **static** **void** main(String[] args) {

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

"C:\\Users\\spdc\\Desktop\\MCA\_Praticals\\selenium\\chromedriver.exe");

ChromeOptions option = **new** ChromeOptions();

option.addArguments("--remote-allow-origins=\*");

WebDriver driver = **new** ChromeDriver(option);

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);

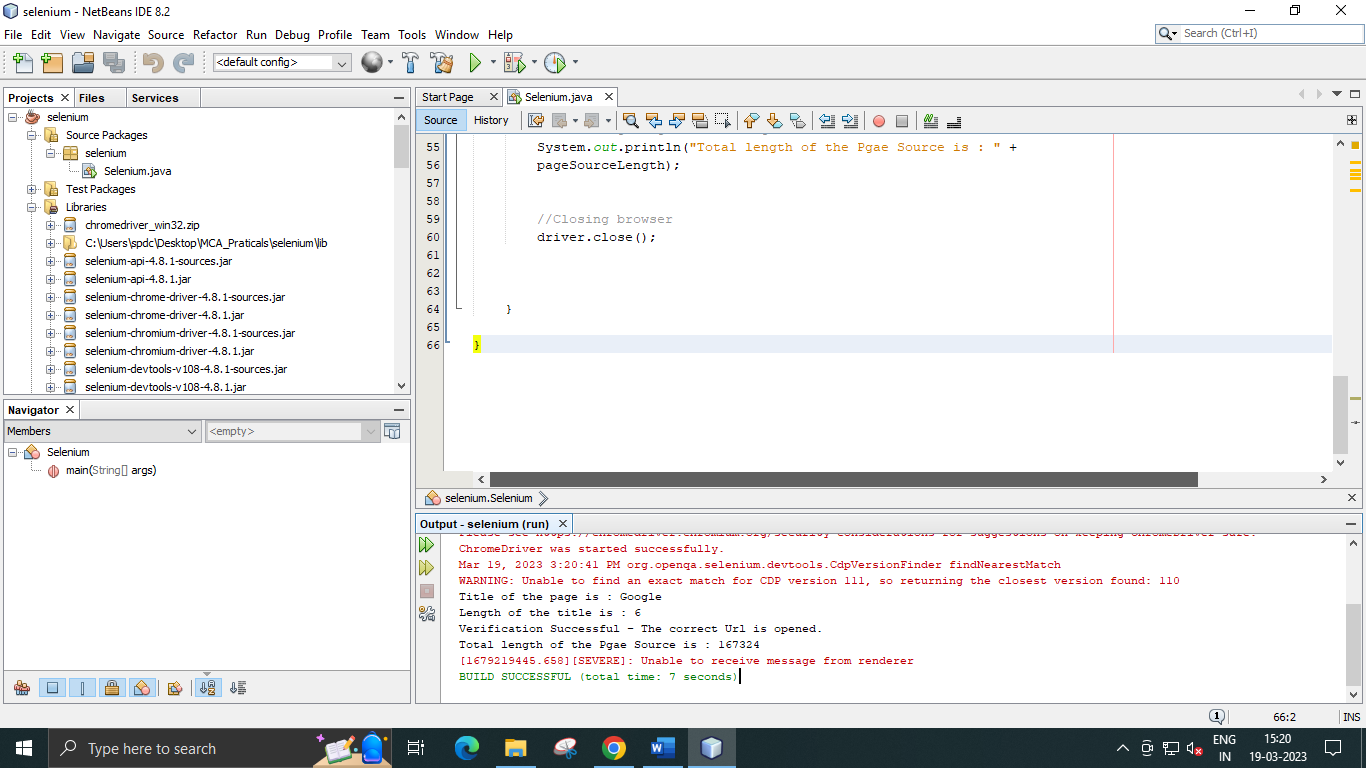
// Closing browser

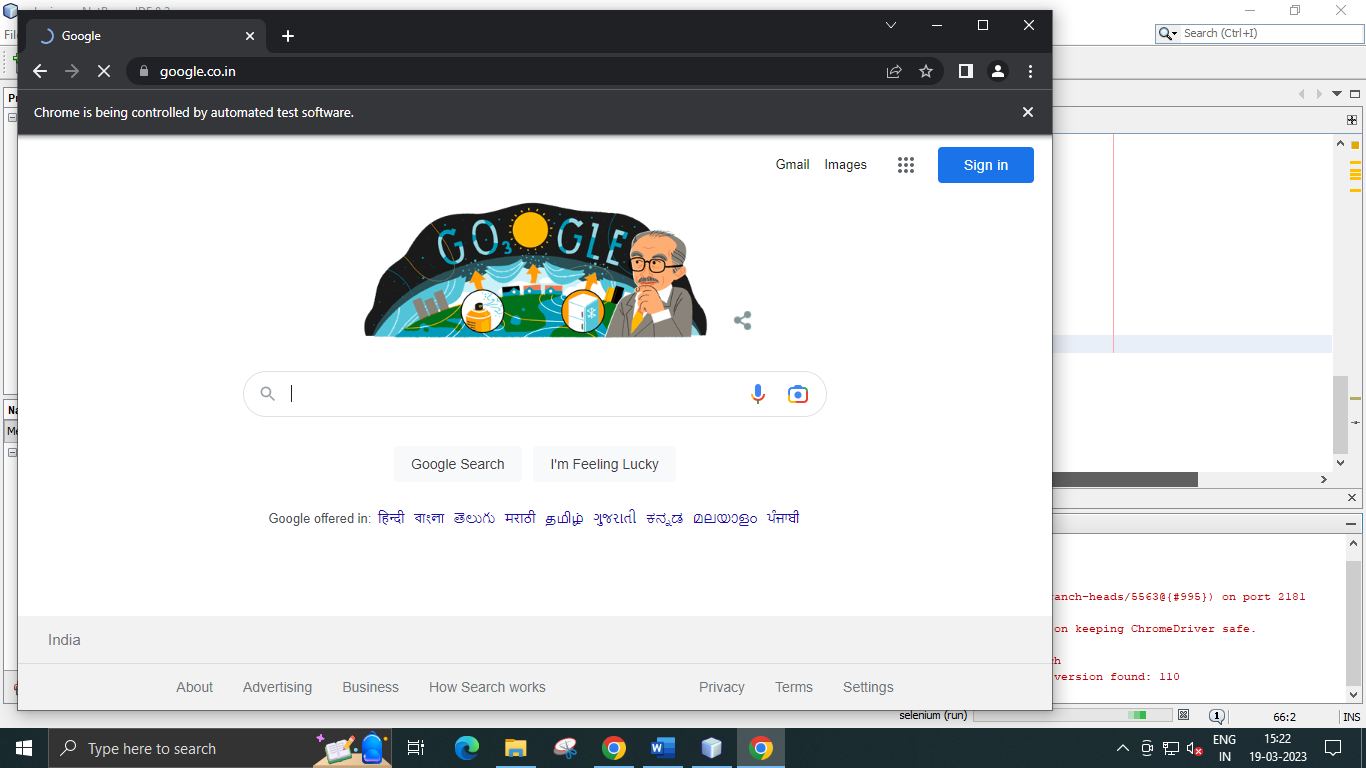
driver.close();

}

}

**OUTPUT :**





**Experiment 5**

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

**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","C:\\Program Files\\Selenium\\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"));

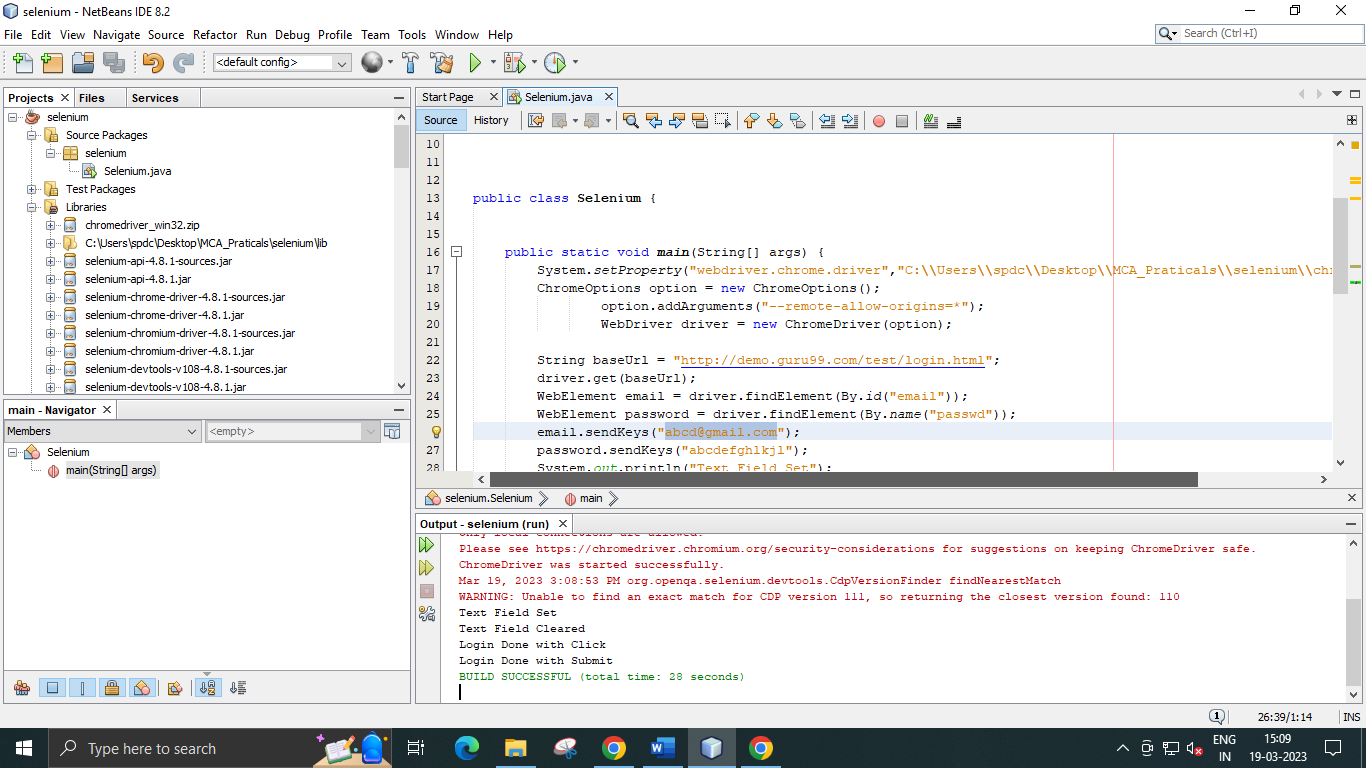
* Using click method to submit form email.sendKeys("abcd@gmail.com"); password.sendKeys("abcdefghlkjl"); login.click(); System.***out***.println("Login Done with Click");

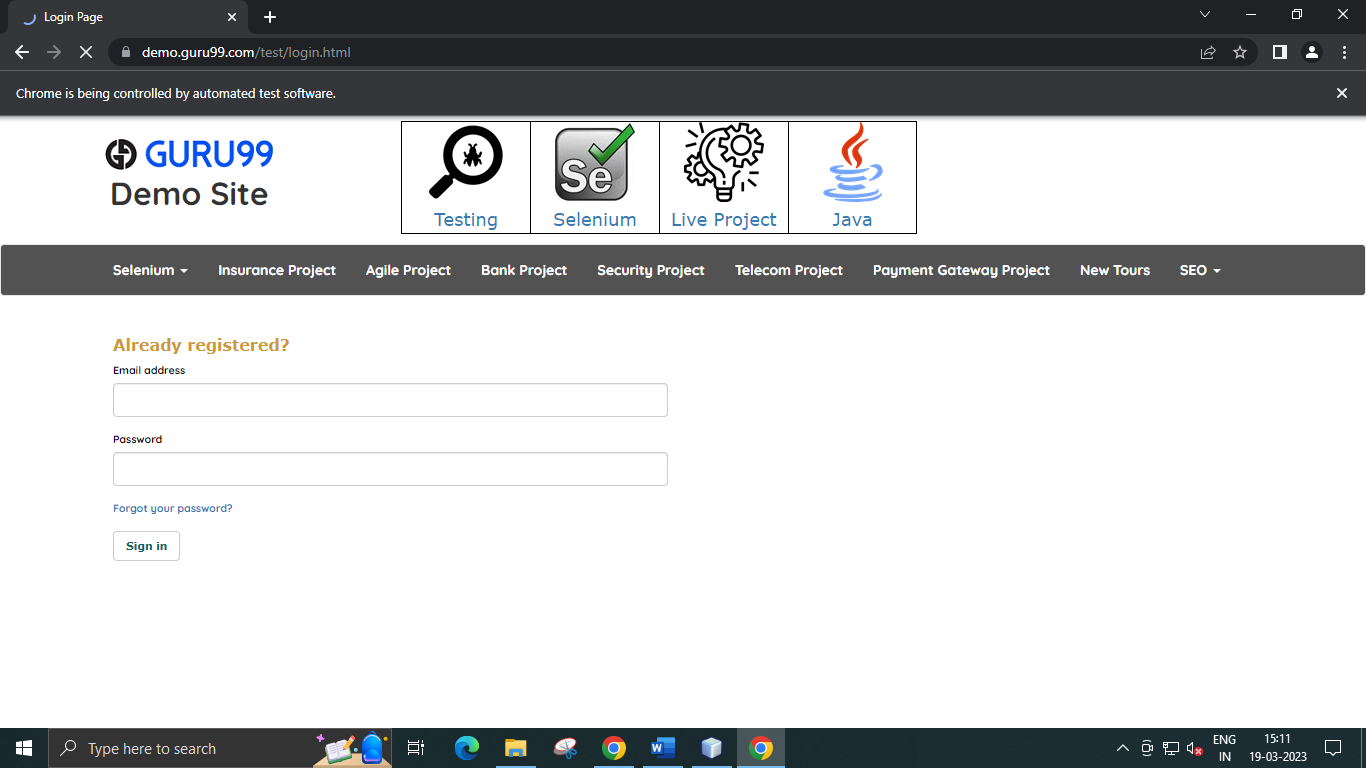
//using submit method to submit the form. Submit used on password field driver.get(baseUrl); driver.findElement(By.*id*("email")).sendKeys("abcd@gmail.com"); driver.findElement(By.*name*("passwd")).sendKeys("abcdefghlkjl"); driver.findElement(By.*id*("SubmitLogin")).submit(); System.***out***.println("Login Done with Submit");

//driver.close();}

}

**OUTPUT :**





A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Experiment 6**

**Aim:- Demonstrate different types of alerts**

**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;

**import** org.openqa.selenium.chrome.ChromeOptions;

**public** **class** NewClass {

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

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

"C:\\Users\\spdc\\Desktop\\MCA\_Praticals\\selenium\\chromedriver.exe");

ChromeOptions option = **new** ChromeOptions();

option.addArguments("--remote-allow-origins=\*");

WebDriver driver = **new** ChromeDriver(option);

// Alert Message handling

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

driver.findElement(By.name("cusid")).sendKeys("53920");

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);

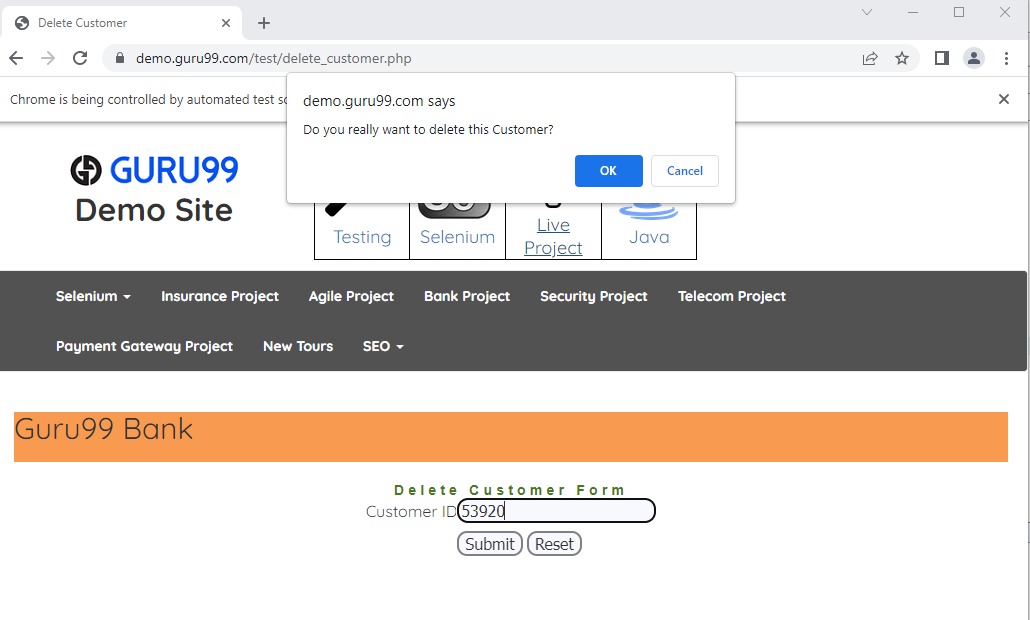
// Accepting alert

alert.accept();

}

}

**Output:-**



**Experiment 7**

**Aim:- Demonstrate CheckBox and Radio Button in Selenium WebDriver.**

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

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

**import** org.openqa.selenium.\*;

**import** org.openqa.selenium.chrome.ChromeOptions;

**public** **class** NewClass {

**public** **static** **void** main(String[] args) {

// declaration and instantiation of objects/variables

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

"C:\\Users\\spdc\\Desktop\\MCA\_Praticals\\selenium\\chromedriver.exe");

ChromeOptions option = **new** ChromeOptions();

option.addArguments("--remote-allow-origins=\*");

WebDriver driver = **new** ChromeDriver(option);

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());

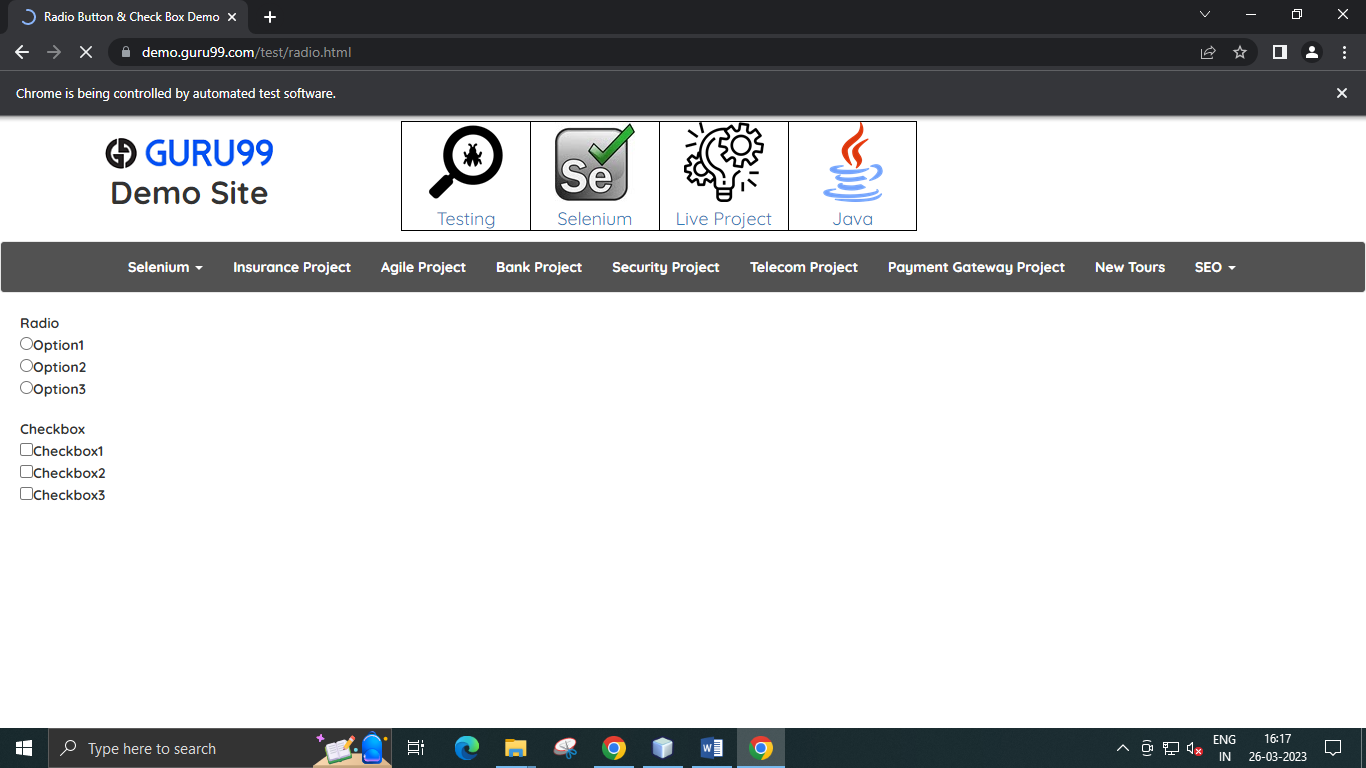
}

driver.close();

}

}

**Output:-**



**Experiment 8**

**Aim:- Demonstrate synchronization in selenium(Implicit wait).**

**package** JavaTpoint;

**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.chrome.ChromeOptions;

**import** org.openqa.selenium.support.ui.ExpectedConditions;

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

**public** **class** NewClass {

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

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

"C:\\Users\\spdc\\Desktop\\MCA\_Praticals\\selenium\\chromedriver.exe");

ChromeOptions option = **new** ChromeOptions();

option.addArguments("--remote-allow-origins=\*");

WebDriver driver = **new** ChromeDriver(option);

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

driver.manage().deleteAllCookies();

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

//pageload timeout

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

// Implicit Wait for 20 seconds

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

driver.findElement(By.xpath("//input[@id='loginusername']")).sendKeys("JavaTpoint.com"); // Finding element and

// sending values

Thread.*sleep*(1000);

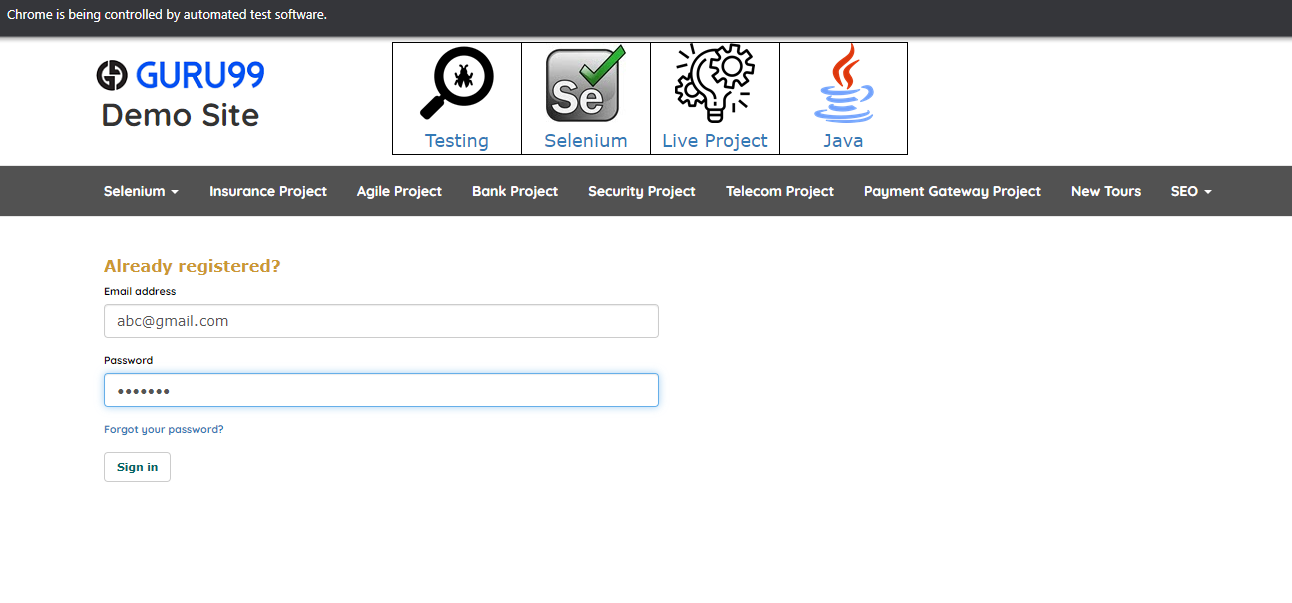
driver.findElement(By.xpath("//input[@id='login-signin']")).click();

//Clicking on the next button if element is located

}

}

**Output:-**



**Experiment 9**

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

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");

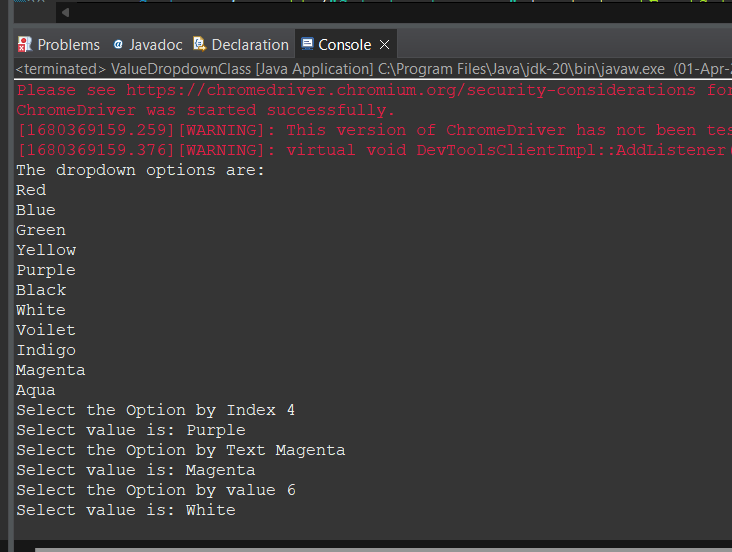
System.out.println("Select value is: " + select.getFirstSelectedOption().getText());

driver.quit();

}

}

Output:-

****

**Experiment 10**

**Aim:- Demonstrate action classes using Selenium Webdriver(Mouse Events)**

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:-

