

**SRI KRISHNA
ARTS AND SCIENCE COLLEGE**
Coimbatore-641 008



RECORD NOTE

DEPT: INFORMATION TECHNOLOGY

NAME: AAKAASH.P

ROLL NO: 19BIT101

PROGRAMME: BSC INFORMATION TECHNOLOGY

CLASS : II – BSC IT – B

COURSE : PROGRAMMING LAB - SOFTWARE TESTING USING SELENIUM

SRI KRISHNA ARTS AND SCIENCE COLLEGE

Coimbatore-641 008



ROLL.NO:19BIT101

Certified bonafide record of work done by AAKAASH.P

during the year 2020 - 2021

Staff In-charge

Head of the Department

Submitted to the Sri Krishna Arts & Science College (Autonomous) end semester

examination held on **05/04/2021**

Internal Examiner

External Examiner

DECLARATION

I **AAKAASH.P** hereby declare that this record of observations is based on the experiments carried out and recorded by me during the laboratory classes of **“PROGRAMMING LAB - SOFTWARE TESTING USING SELENIUM”** conducted by SRI KRISHNA ARTS AND SCIENCE COLLEGE, Coimbatore-641 008.

AAKAASH.P

Date:

Signature of the Student

Name of the Student : **AAKAASH.P**

Roll Number : **19BIT101**

Countersigned by Staff

CONTENTS

| S.NO. | DATE | TITLE OF THE EXPERIMENT | PAGE NO. | SIGN |
|-------|------------|--|----------|------|
| 1 | 11/02/2021 | Open a chrome browser using selenium web driver. | 06 | |
| | | | | |
| 2 | 11/02/2021 | Display that the website is opened successfully wait for 5 seconds | 09 | |
| | | | | |
| 3 | 11/02/2021 | Upload a file with send keys method by web driver | 12 | |
| | | | | |
| 4 | 18/02/2021 | Access a link in selenium web driver bylinktext() and partiallinktext(). | 16 | |
| | | | | |
| 5 | 14/03/2021 | Locate a link by selecting multiple items in dropdown | 19 | |
| | | | | |
| 6 | 19/03/2021 | Develop a test to locate a frame using tag name | 22 | |
| | | | | |
| 7 | 26/03/2021 | Test the case to submit a login from using web driver | 25 | |
| | | | | |
| 8 | 29/03/2021 | Synchronize with an implicit wait | 28 | |
| | | | | |
| 9 | 29/03/2021 | Synchronize with an explicit wait | 31 | |
| | | | | |
| 10 | 01/04/2021 | Identifying and handling a pop-up window by its name | 34 | |
| | | | | |
| 11 | 01/04/2021 | Use locaters with parameters to search for keyword | 37 | |
| | | | | |
| 12 | 03/04/2021 | Dynamic xpath in selenium using double slash | 40 | |
| | | | | |
| | | | | |

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

Ex no : 1

Date : 11/02/2021

**OPEN A CHROME BROWSER USING SELENIUM WEB
DRIVER**

AIM:

To write coding to develop a test to open a chrome browser using selenium web driver:

ALGORITHM:

STEP 1: Open Eclipse IDE

STEP 2: Create a new java project and followed by create a new class named as EX1

STEP 3: Sets the system property to value named webdriver.chrome.driver and the path is mentioned to get the chrome driver

STEP 4: Create an instance of the chromedriver.

WebDriver driver=new ChromeDriver ();

STEP 5: get () method automatically opens a new browser window and fetches the page that you specify inside its. parentheses

driver.get ("http://www.google.com");

STEP 6: Importing Packages Such as:

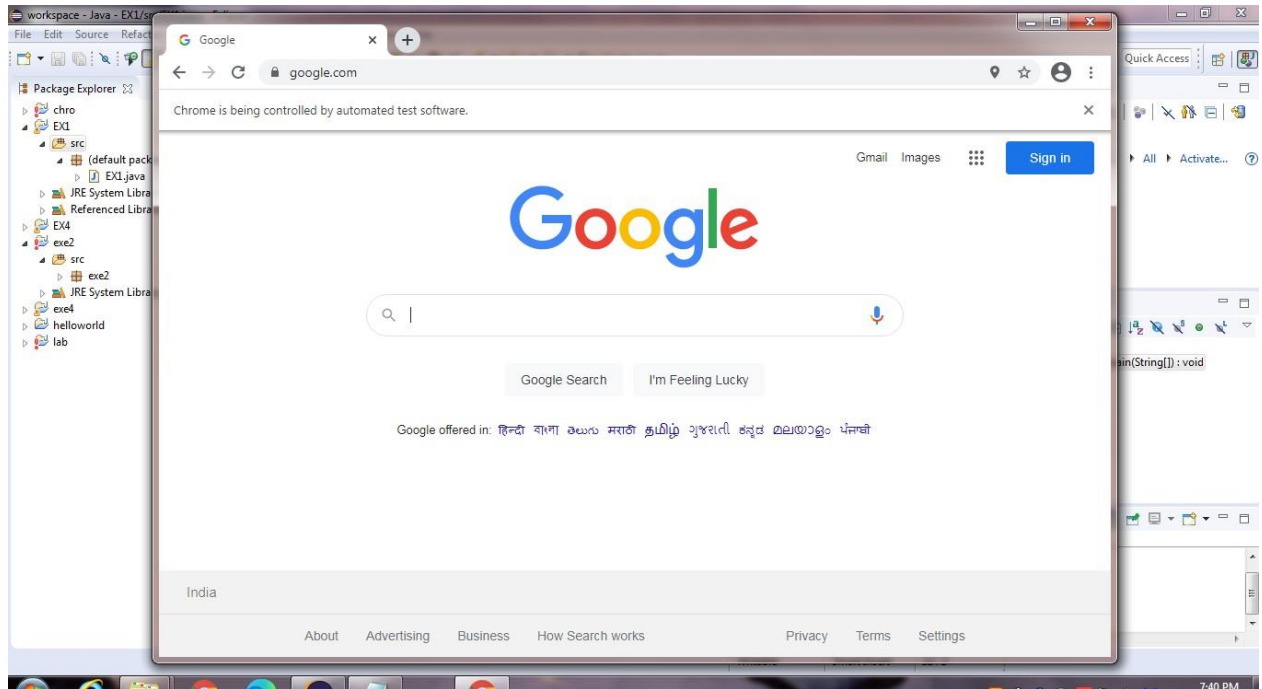
- **org.openqa.selenium.WebDriver** - contains the WebDriver class needed to instantiate a new browser loaded with a specific driver
- **org.openqa.selenium.chrome.ChromeDriver** - contains the ChromeDriver class needed to instantiate a Chrome-specific driver onto the browser instantiated by the WebDriver class

STEP 7: Run the program and the new browser window automatically opened

CODING:

```
package selenium;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class project1
{
    public static void main(String[] args)
    {
        System.setProperty("webdriver.chrome.driver","C:\\Users\\Admin\\eclipse\\java-neon\\eclipse\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("https://www.google.com");
    }
}
```

OUTPUT:



RESULT:

Thus the program has been successfully executed and hence the browser also opened successfully.

Ex no : 2

Date : 11/02/2021

DISPLAY THAT THE WEBSITE IS OPENED

SUCCESSFULLY WAIT FOR 5 SECONDS

AIM:

To develop a test to open a chrome browser using selenium web driver.

ALGORITHM:

STEP 1: Open Eclipse IDE

STEP 2: Create a new java project and followed by create a new class named as EX2

STEP 3: Sets the system property to value named webdriver.chrome.driver and the path is mentioned to get the chrome driver

STEP 4: Create an instance of the chromedriver.

WebDriver driver=new ChromeDriver ();

STEP 5: get () method automatically opens a new browser window and fetches the page that you specify inside its. parentheses

driver.get ("http://www.google.com");

STEP 6: Importing Packages Such as:

□ **org.openqa.selenium.WebDriver** - contains the WebDriver class needed to instantiate a new browser loaded with a specific driver

□ **org.openqa.selenium.chrome.ChromeDriver** - contains the ChromeDriver class needed to instantiate a Chrome-specific driver onto the browser instantiated by the WebDriver class

STEP 7: We have to display the message as opened successfully and need to auto close after 5 seconds

STEP 8: Then close the web driver. Web driver.close();

STEP 9: Run the program and the new browser window automatically opened

Coding:

```
package selenium;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class project2 {

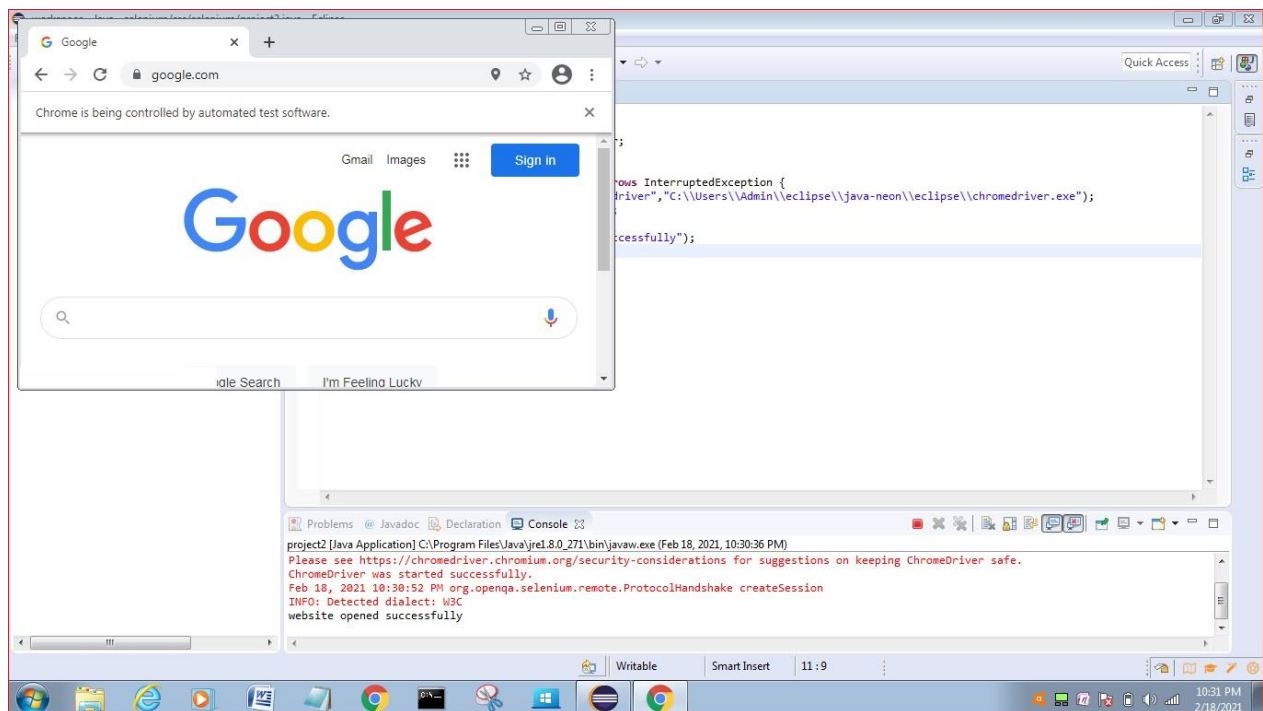
    public static void main(String[] args) throws
    InterruptedException {

        System.setProperty("webdriver.chrome.driver","C:\\Users\\Admin\\e
clipse\\java-neon\\eclipse\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("https://www.google.com");
        System.out.println("website opened successfully");
        Thread.sleep(5000);
        driver.close();

    }

}
```

OUTPUT:



RESULT:

Thus the program has been executed successfully and hence the browser also opened successfully.

Ex.no.3

Date: 11/02/2021

**UPLOAD A FILE WITH SEND KEYS METHOD
BY WEB DRIVER**

AIM:

To develop a test to upload a file with send keys method by web driver.

ALGORITHM:

STEP 1: Open Eclipse IDE

STEP 2: Create a new java project and followed by create a new class named as prg3.

STEP 3: Sets the system property to value named webdriver.chrome.driver and the path is mentioned to get the chrome driver

STEP 4: Create an instance of the chromedriver. WebDriver driver=new ChromeDriver ();

STEP 5: get () method automatically opens a new browser window and fetches the page that you specify inside its. parentheses
driver.get("http://demo.guru99.com/test/upload/");

STEP 6: Importing Packages Such as:

- org.openqa.selenium.WebDriver - contains the WebDriver class needed to instantiate a new browser loaded with a specific driver
- org.openqa.selenium.chrome.ChromeDriver - contains the ChromeDriver class needed to instantiate a Chrome-specific driver onto the browser instantiated by the WebDriver class

STEP 7: the chosen file is uploaded with the help of uploadElement method.

STEP 8: send keys method helps as to type the file name automatically in the editable field.

STEP 9: for checking the "I accept the terms of service" check box find element and click() methods are used.

STEP 10: finally send button is clicked to upload the file with the help of click() method

STEP 11: Program testes successfully.

CODING:

```
package selenium;

import org.openqa.selenium.*;
import org.openqa.selenium.chrome.ChromeDriver;
public class project3A {

    public static void main(String[] args) {
        System.setProperty("webdriver.chrome.driver","C:\\Users\\Admin\\eclipse\\java-neon\\eclipse\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("http://demo.guru99.com/test/upload/");
        WebElement uploadElement = driver.findElement(By.id("uploadfile_0"));

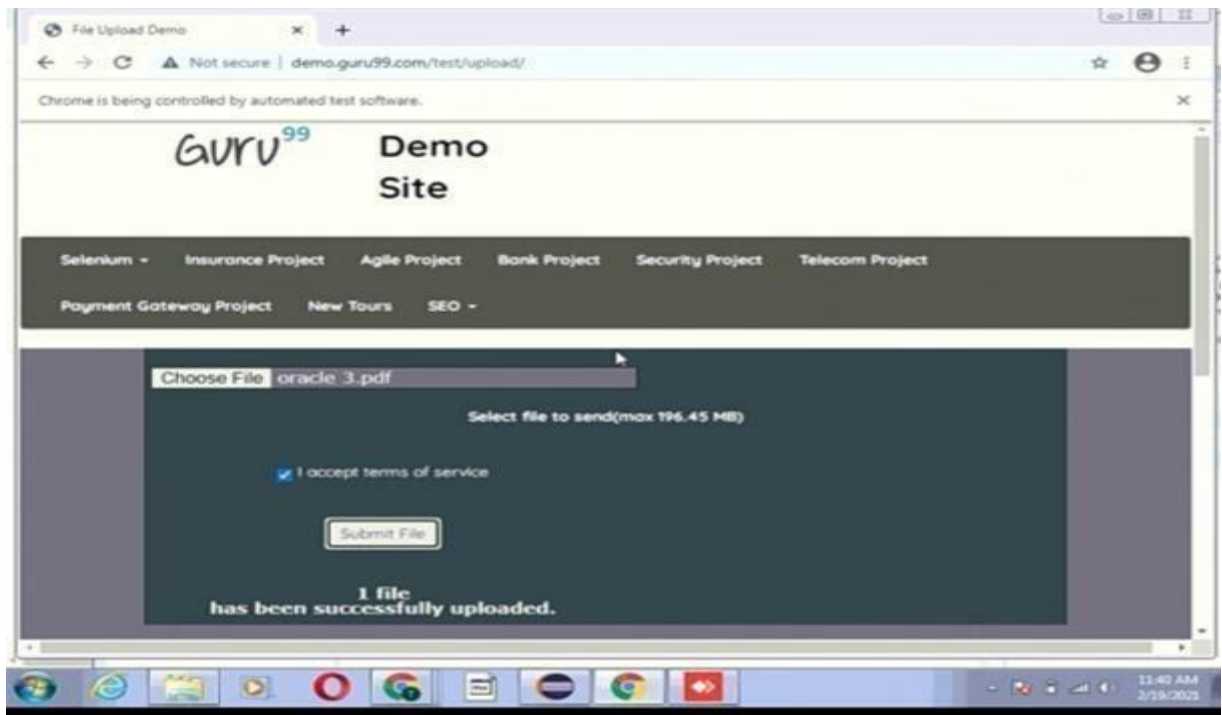
        // enter the file path onto the file-selection input field
        uploadElement.sendKeys("C:\\Users\\Admin\\Documents\\weekend assignment -6");

        // check the "I accept the terms of service" check box
        driver.findElement(By.id("terms")).click();

        // click the "UploadFile" button
        driver.findElement(By.name("send")).click();

    }
}
```

OUTPUT:



RESULT:

Thus the program has been executed successfully and hence the browser also opened successfully.

Ex no : 4

Date : 18/02/2021

ACCESS A LINK IN SELENIUM WEB DRIVER

BY LINKTEXT() AND PARTIALLINKTEXT()

AIM:

Write a coding to automate to access a link in selenium web driver by linktext() and partiallinktext().

ALGORITHM:

STEP 1: Open Eclipse IDE

STEP 2: Create a new java project and followed by create a new class named as selenium4

STEP 3: Sets the system property to value named webdriver.chrome.driver and the path is mentioned to get the chrome driver

STEP 4: Create an instance of the chromedriver.

WebDriver driver=new ChromeDriver ();

STEP 5: get () method automatically opens a new browser window and fetches the page that you specify inside its parentheses.

driver.get ("http://www.google.com");

STEP 6: Importing all necessary packages.

STEP 7: Accessing links using their exact link text is done through the By.linkText() method.

STEP 8: Specify a partial link text that has multiple matches, only the first match will be accessed.

STEP 9: Go to Google link - > Right click on the gmail signin text box -> Select Inspect option -> Select Element option - > Choose text box id "identifierId"

STEP 10: To find the element using the id of "identifierId" and the send emailid to login.

STEP 11: Program testes successfully.

CODING :

```
package selenium;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class project4 {

    public static void main(String[] args) {

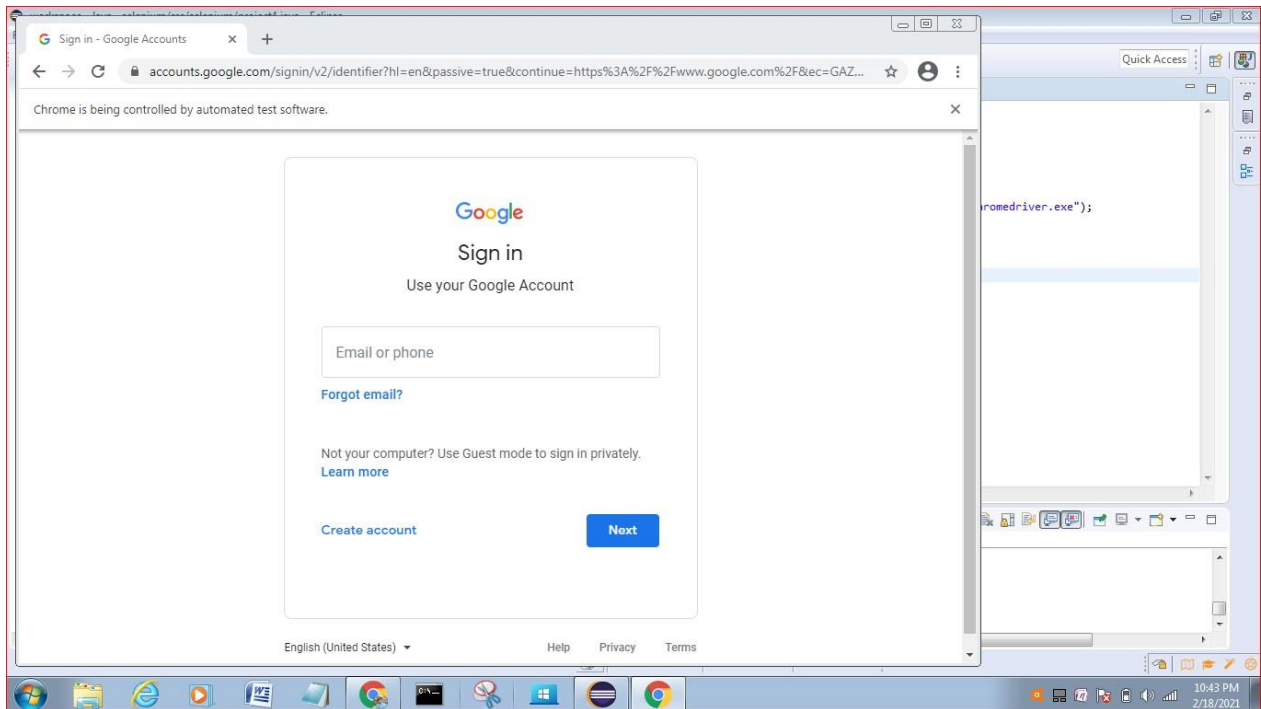
        System.setProperty("webdriver.chrome.driver","C:\\Users\\Admin\\eclipse\\java-neon\\eclipse\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("https://www.google.com");
        driver.findElement(By.linkText("Sign in")).click();
        driver.findElement(By.partialLinkText("Sign")).click();

        driver.findElement(By.id("identifierId")).sendKeys("mithunganesh2001@gmail.com");

    }

}
```


OUTPUT:



RESULT:

Thus the program has been successfully executed and hence the browser also opened successfully.

Ex no : 5

Date: 14/03/2021

**LOCATE A LINK BY SELECTING MULTIPLE ITEMS
IN DROPDOWN**

AIM:

To Write coding for Locate a link by selecting multiple items in a dropdown.

ALGORITHM:

STEP 1: Open Eclipse IDE

STEP 2: Create a new java project and followed by create a new class named as selenium5

STEP 3: Sets the system property to value named **webdriver.chrome.driver** and the path is mentioned to get the chrome driver.

STEP 4: Create an instance of the chromedriver.

WebDriver driver=new Chrome Driver();

STEP 5: get () method automatically opens a new browser window and fetches the page that you specify inside its parentheses.

driver.get ("http://www.google.com");

STEP 6: Importing all necessary Packages.

STEP 7: Accessing links using their exact link text is done through the **By.name()** method.

STEP 8: Specify the index,value and visible text.

CODING:

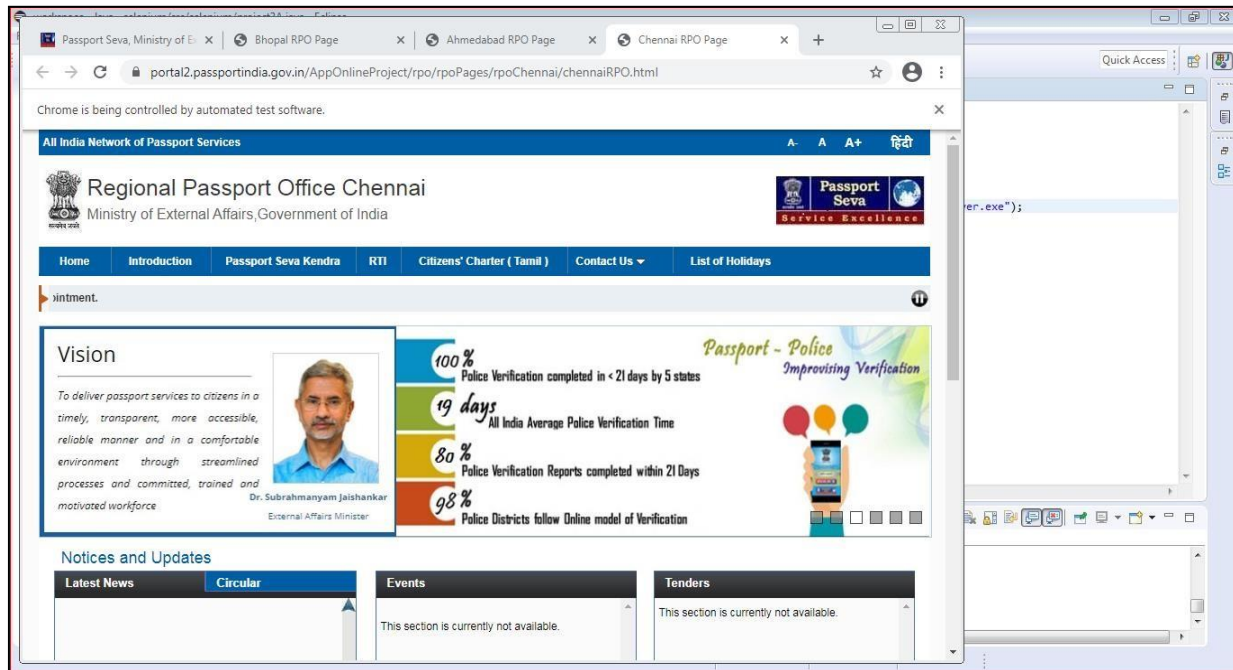
```
package selenium;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.Select;
public class project5 {
    public static void main(String[] args) throws
InterruptedException {
        // TODO Auto-generated method stub

        System.setProperty("webdriver.chrome.driver","C:\\Users\\Admin\\e
clipse\\java-neon\\eclipse\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();

        driver.get("https://portal2.passportindia.gov.in/AppOnlineProject
/user/RegistrationBaseAction&quot;");
        driver.manage().window().maximize();
        Select passportoffice = new Select
(driver.findElement(By.name("dcdLocation")));
        passportoffice.selectByIndex(5);
        Thread.sleep(2000);
        passportoffice.selectByValue("1");
        Thread.sleep(2000);
        passportoffice.selectByVisibleText("Chennai");
        Thread.sleep(2000);

    }
}
```

OUTPUT:



RESULT:

Thus the program has been successfully executed and hence the browser also opened successfully.

Ex no : 6

Date : 19/03/2021

**DEVELOP A TEST TO LOCATE A FRAME USING
TAG NAME**

AIM:

To develop a test to locate a frame using tag name.

ALGORITHM:

STEP 1: Open Eclipse IDE

STEP 2: Create a new java project and followed by create a new class named as selenium7

STEP 3: Sets the system property to value named webdriver.chrome.driver and the path is mentioned to get the chrome driver

STEP 4: Create an instance of the chromedriver.

WebDriver driver=new Chrome Driver();

STEP 5: get () method automatically opens a new browser window and fetches the page that you specify inside its parentheses.

**driver.get
("https://www.w3schools.com/tags/tryit.asp?filename=tryhtml_button_test
"");**

STEP 6: Importing all necessary Packages.

STEP 7: Accessing links using their exact link text is done through the By.xpath() method.

STEP 8:frame();method is used to iframe results.

PROGRAM:

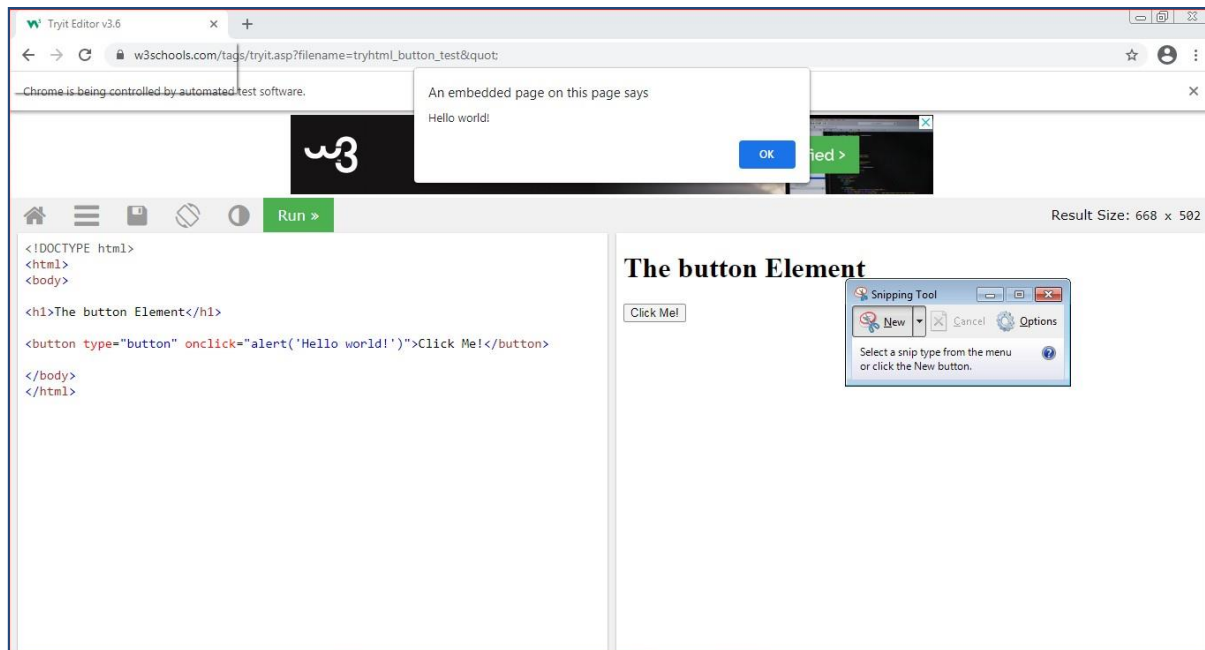
```
package selenium;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class project6 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        System.setProperty("webdriver.chrome.driver","C:\\Users\\Admin\\eclipse\\java-neon\\eclipse\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();

        driver.get("https://www.w3schools.com/tags/tryit.asp?filename=try_html_button_test");
        driver.manage().window().maximize();
        driver.switchTo().frame("iframeResult");
        driver.findElement(By.xpath("/html/body/button")).click();
        driver.switchTo().defaultContent();
    }
}
```

OUTPUT:



RESULT:

Using the above code the program is successfully executed.

Ex no : 7

Date : 26/03/2021

TEST THE CASE TO SUBMIT A LOGIN FORM USING WEB DRIVER

AIM:

To test the case to submit a login form using Web Driver.

ALGORITHM:

STEP 1: Open Eclipse IDE

STEP 2: Create a new java project and followed by create a new class named as selenium8

STEP 3: Sets the system property to value named webdriver.chrome.driver and the path is mentioned to get the chrome driver

STEP 4: Create an instance of the chromedriver.

WebDriver driver=new Chrome Driver();

STEP 5: get () method automatically opens a new browser window and fetches the page that you specify inside its parentheses.

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

STEP 6: Importing all necessary Packages.

STEP 7: Accessing links using their exact link text is done through the By.xpath() method.

STEP 8:sendKeys()method is used to sign in into linked in website.

PROGRAM:

```
package selenium;
import org.openqa.selenium.*;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class project7 {

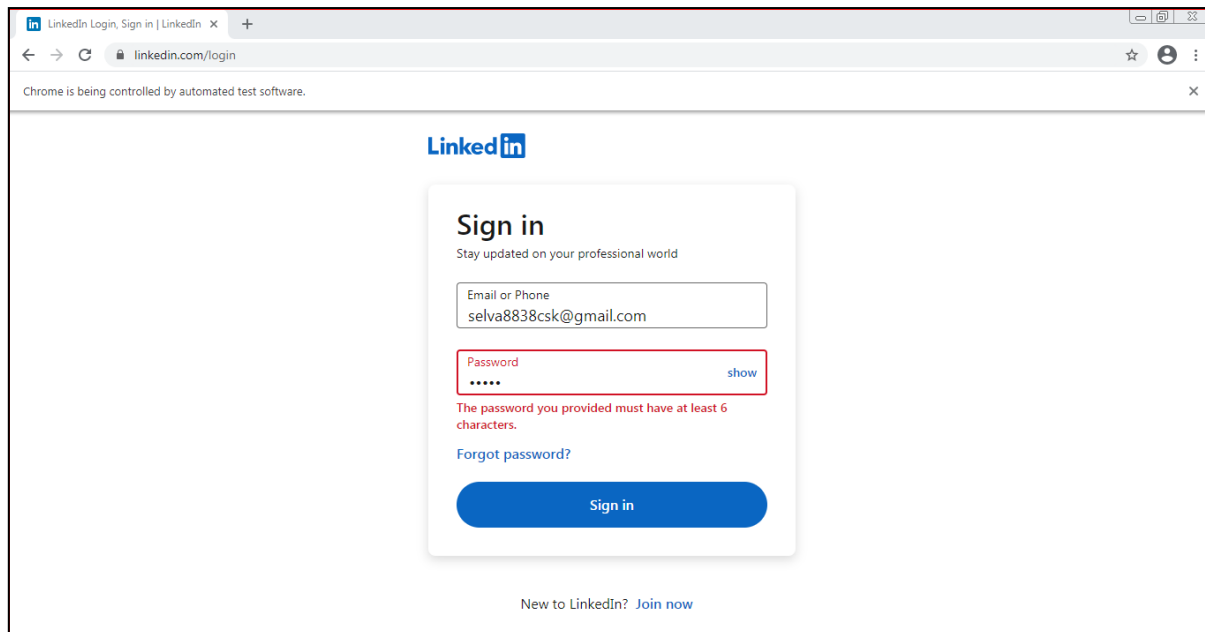
    public static void main(String[] args) throws
    InterruptedException {
        // TODO Auto-generated method stub

        System.setProperty("webdriver.chrome.driver","C:\\Users\\Admin\\e
clipse\\java-neon\\eclipse\\chromedriver.exe");
        WebDriver driver=new ChromeDriver();
        driver.manage().window().maximize();
        driver.get("https://www.linkedin.com/login");
        WebElement username=driver.findElement(By.id("username"));
        WebElement password=driver.findElement(By.id("password"));
        WebElement
login=driver.findElement(By.xpath("//button[text()='Sign in']"));
        username.sendKeys("selva8838csk@gmail.com");
        password.sendKeys("selva");
        login.click();

    }

}
```

OUTPUT:



RESULT:

Using the above code the program is successfully executed.

Ex no : 8

Date : 29/03/2021

SYNCHRONIZE WITH AN IMPLICIT WAIT

AIM:

To write program Develop a test to Synchronize with an implicit wait.

ALGORITHM:

STEP 1 : Open Eclipse IDE

STEP 2 : Create a new java project and followed by create a new class named prg6

STEP 3 : Sets the system property to value Named Webdriver.chrome.driver and the path is Mentioned to get the chrome driver

STEP 4 : create an instance of the Chromedriver.driver=new chromedriver();

STEP 5 : get(); method automatically opens a new browser window and fetches the page that you Specify inside it's parenthesis.

Driver.get(https://www.w3schools.com/tags/tryit.asp?filename=tryhtml_button_test)

STEP 6 : Importing all necessary packages.

STEP 7 : implicit wait(); stays in place for entire duration for which the browser is open.

STEP 8 : close(); method automatically closes from the iopened output window.

CODING:

```
package selenium;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import java.util.concurrent.TimeUnit;

public class project8 {

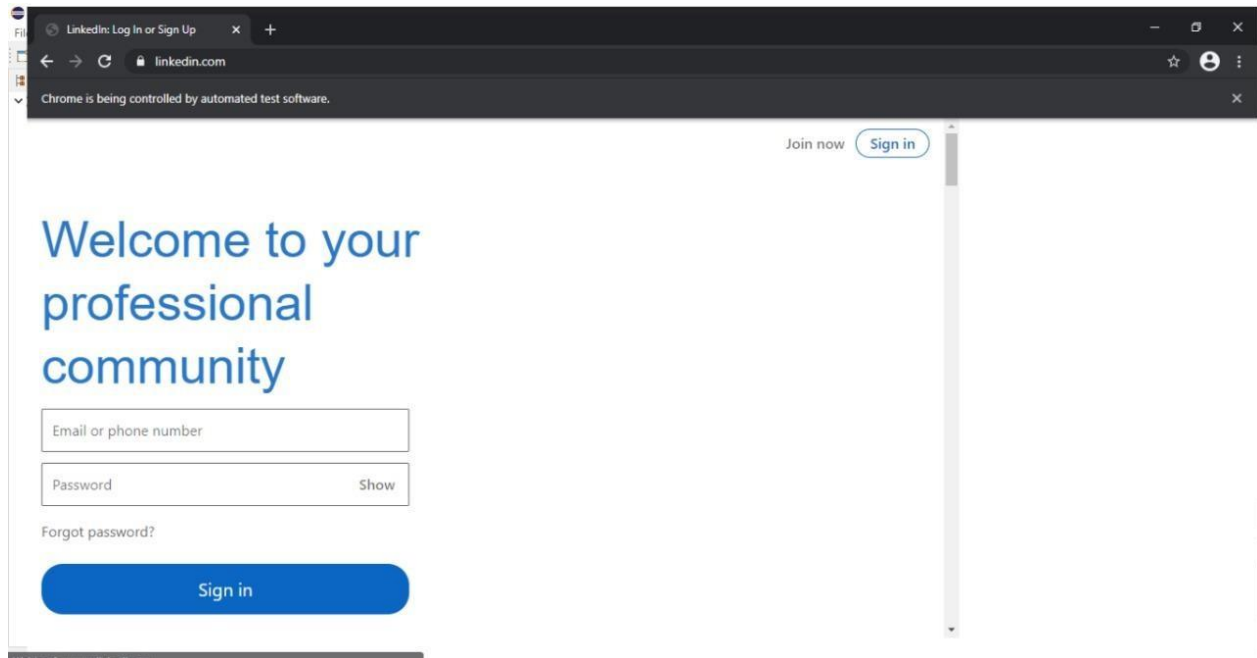
    public static void main(String[] args) throws
    InterruptedException{
        // TODO Auto-generated method stub

        System.setProperty("webdriver.chrome.driver","C:\\Users\\Admin\\e
clipse\\java-neon\\eclipse\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("https://www.linkedin.com/");
        driver.manage().window().maximize();
        driver.manage().timeouts().implicitlyWait(30,
TimeUnit.SECONDS);
        driver.close();

    }

}
```

OUTPUT:



RESULT:

Using the above code the program is successfully executed.

Ex no : 9

Date : 29/03/2021

SYNCHRONIZE WITH AN EXPLICIT WAIT

AIM:

To write a program Develop a test to Synchronize with an explicit wait.

ALGORITHM:

STEP 1 : Open Eclipse IDE

STEP 2 : Create a new java project and followed by create a new class named prg6

STEP 3 : Sets the system property to value Named Webdriver.chrome.driver and the path is Mentioned to get the chrome driver

STEP 4 : create an instance of the Chromedriver.driver=new chromedriver();

STEP 5 : get(); method automatically opens a new browser window and fetches the page that you Specify inside it's parenthesis.

Driver.get(https://www.w3schools.com/tags/tryit.asp?filename=tryhtml_button_test)

STEP 6 : Importing all necessary packages.

STEP 7 : explicit wait(); insists the webdriver to wait for certain conditions or maximum time before throwing exception.

STEP 8 : Run the program and the new browser window automatically opened.

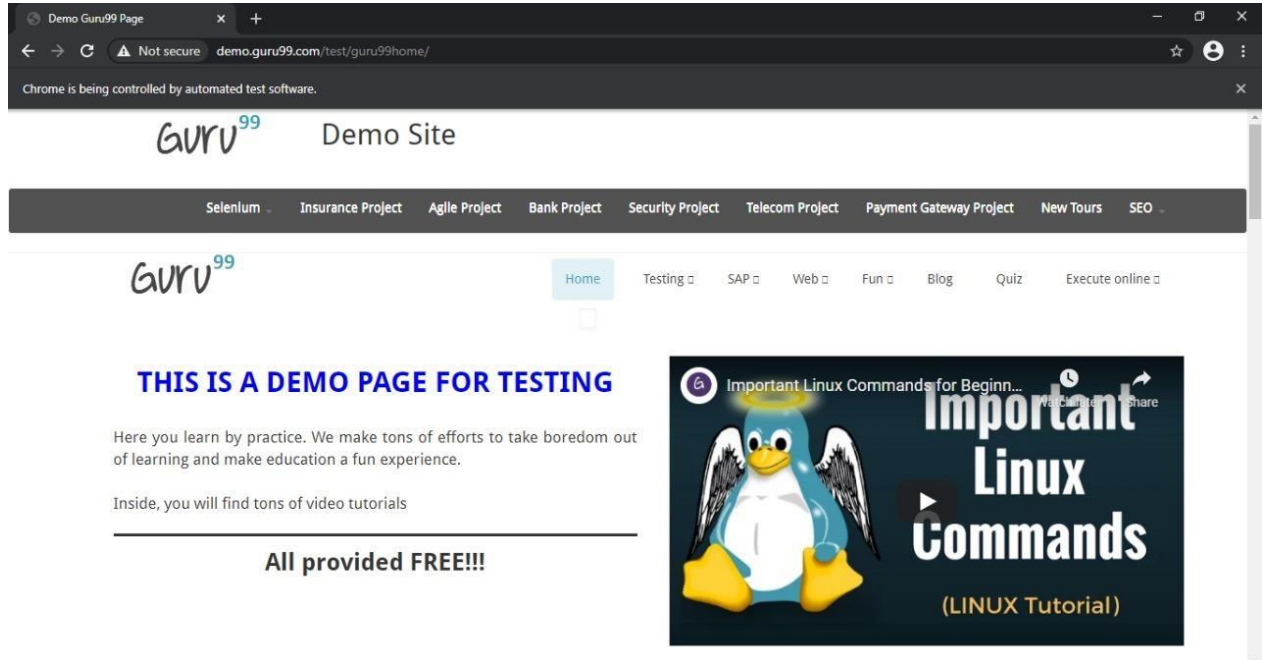
CODING:

```
package selenium;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.By;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.ui.WebDriverWait;
public class project9 {
    public static void main(String[] args) throws InterruptedException{
        // TODO Auto-generated method stub

        System.setProperty("webdriver.chrome.driver","C:\\Users\\Admin\\eclipse\\java-neon\\eclipse\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        WebDriverWait wait=new WebDriverWait(driver, 20);

        driver.get("http://demo.guru99.com/test/guru99home/");
        //Maximizes the browser window
        driver.manage().window().maximize() ;
        //get the actual value of the title
        WebElement guru99seleniumlink;
        guru99seleniumlink=
        wait.until(ExpectedConditions.visibilityOfElementLocated
            (By.xpath("/html/body/div[1]/section/div[2]/div/div[1]/div/div[1]
            /div/div/div/div[2]/div[2]/div/div/div/div/div[1]/div/div/a/i")));
        guru99seleniumlink.click();
    }
}
```

OUTPUT:



RESULT:

Using the above code the program is successfully executed.

Ex no : 10

Date : 01/04/2021

**IDENTIFYING AND HANDLING A POP-UP WINDOW BY
ITS NAME**

AIM:

To write a Test the case by Identifying and handling a pop-up window by its name.

ALGORITHM:

STEP 1 : Open Eclipse IDE

STEP 2 : Create a new java project and followed by create a new class named prg6

STEP 3 : Sets the system property to value Named Webdriver.chrome.driver and the path is Mentioned to get the chrome driver

STEP 4 : create an instance of the Chromedriver.driver=new chromedriver();

STEP 5 : get(); method automatically opens a new browser window and fetches the page that you Specify inside it's parenthesis.

Driver.get(https://www.w3schools.com/tags/tryit.asp?filename=tryhtml_button_test)

STEP 6 : Importing all necessary packages.

STEP 7 : findelement() finds a single web element and returns as a webelement selenium objects

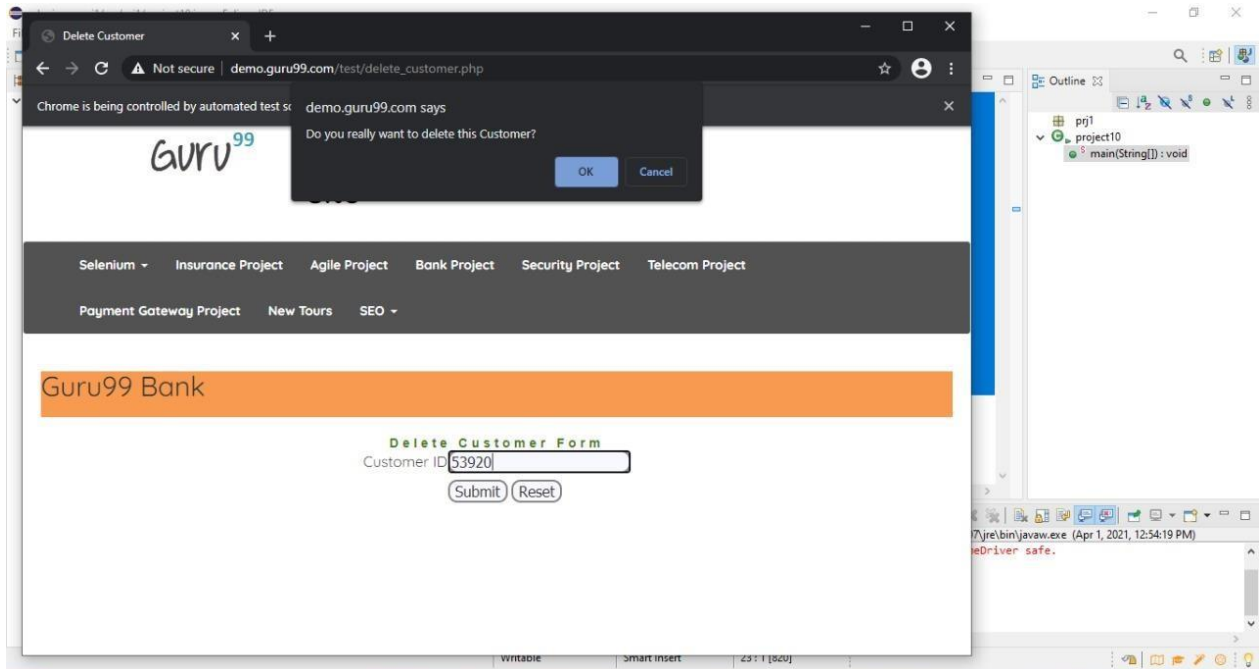
STEP 8 : sendkeys() used to enter editable content in text fields during test execution.

STEP 9 : alert() displays some information or warning on the screen.

CODING:

```
package selenium;
import org.openqa.selenium.By;
import org.openqa.selenium.Alert;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class project10 {
    public static void main(String[] args) throws InterruptedException {
        // TODO Auto-generated method stub
        System.setProperty("webdriver.chrome.driver","C:\\Users\\Admin\\eclipse\\java-neon\\eclipse\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("http://demo.guru99.com/test/delete_customer.php");
        driver.findElement(By.name("cusid")).sendKeys("53920");
        driver.findElement(By.name("submit")).submit();
        Alert alert = driver.switchTo().alert();
        String alertMessage= driver.switchTo().alert().getText();
        System.out.println(alertMessage);
        Thread.sleep(5000);
        alert.accept();
    }
}
```

OUTPUT:



RESULT:

Using the above code the program is successfully executed.

Ex no : 11

Date : 01/04/2021

**USE LOCATORS WITH PARAMETERS TO SEARCH
FOR KEYWORD**

AIM:

To write a program Develop a test plan to use locators with parameters to search for keyword.

ALGORITHM:

STEP 1 : Open Eclipse IDE

STEP 2 : Create a new java project and followed by create a new class named prg6

STEP 3 : Sets the system property to value Named Webdriver.chrome.driver and the path is Mentioned to get the chrome driver

STEP 4 : create an instance of the Chromedriver.driver=new chromedriver();

STEP 5 : get(); method automatically opens a new browser window and fetches the page that you Specify inside it's parenthesis.

Driver.get(https://www.w3schools.com/tags/tryit.asp?filename=tryhtml_button_test")

STEP 6 : Importing all necessary packages.

STEP 7 : findelement() finds a single web element and returns as a webelement selenium objects

STEP 8 : sendkeys() used to enter editable content in text fields during test execution.

STEP 9 : action() method handles keyboard and mouse events.

STEP 10 : Run the program and the new browser window automatically opened.

CODING:

```
package selenium;
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.interactions.Actions;
import org.openqa.selenium.support.ui.Select;
public class project11 {

    public static void main(String[] args) throws
InterruptedException{
        // TODO Auto-generated method stub

        System.setProperty("webdriver.chrome.driver","C:\\Users\\Admin\\e
clipse\\java-neon\\eclipse\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("https://google.com");
        driver.findElement(By.name("q")).sendKeys("passport seva");
        driver.findElement(By.name("q")).sendKeys(Keys.ESCAPE);

        driver.get("https://portal2.passportindia.gov.in/AppOnlineProject
/user/RegistrationBaseAction");
        driver.manage().window().maximize();
        Select passportoffice = new
Select(driver.findElement(By.name("dcdRLocation")));
        passportoffice.selectByIndex(17);
        driver.findElement(By.id("emailloginSameyes")).click();
        driver.findElement(By.id("hintQues")).click();
        Select hintques = new Select
(driver.findElement(By.id("hintQues")));
        hintques.selectByVisibleText("Birth City");
        driver.findElement(By.xpath("//*[@id='passOffice']"));
        Actions actions = new Actions(driver);

        actions.moveToElement(driver.findElement(By.xpath("//*[@id='passO
ffice']"))).perform();
        Thread.sleep(1000);

        actions.moveToElement(driver.findElement(By.id("passportoffices2"
))).click();
        actions.perform();
        Thread.sleep(1000);
    }
}
```

OUTPUT:

The screenshot shows a web browser window with the URL `portal2.passportindia.gov.in/AppOnlineProject/mission/mission`. The page is titled "Mission/ Post Abroad : Passport" and is part of the "Passport Seva" portal, which is the Consular, Passport & Visa Division of the Ministry of External Affairs, Government of India. The page is dated Thursday, April 01, 2021, at 12:56:09 PM. The navigation bar includes links for Home, About Us, Passport Offices, Consular / Visa, RTI, Citizens' Charter, Contact Us, and What's New. A search bar is also present. The main content area is titled "Mission/Post Abroad" and features a "Country" dropdown menu with a "Go" button. A message at the top wishes a Very Happy International Women Day to all Passport Seva Kendras (PSKs) and Post Offices. The left sidebar contains an "Information Corner" with links to Getting Started, Passport Act and Rules, FAQs, Locate Passport Seva Kendra, Locate Common Service Centers, Fee Calculator, Appointment Availability Status, Know your Police Station, Quick Guides, Tatkaal Appointment Opening Time, Instructions Booklet, and All India Network of Passport Services.

Thursday, April 01, 2021 | 12:56:09 PM

Skip to main content | Screen Reader Access | Sitemap | Home | A- | A+ | A++

Passport Seva
Consular, Passport & Visa Division
Ministry of External Affairs, Government of India

Home | About Us | Passport Offices | Consular / Visa | RTI | Citizens' Charter | Contact Us | What's New | Search...

Wishing you a Very Happy International Women Day! All Passport Seva Kendras (PSKs) and Post Offices

You are here : Home > Passport Offices > Mission/Post Abroad

Mission/Post Abroad

Country* Fields marked with asterisk (*) are mandatory

Go

Information Corner

- Getting Started
- Passport Act and Rules
- FAQs
- Locate Passport Seva Kendra
- Locate Common Service Centers
- Fee Calculator
- Appointment Availability Status new!
- Know your Police Station
- Quick Guides
- Tatkaal Appointment Opening Time
- Instructions Booklet
- All India Network of Passport Services

RESULT:

Using the above code the program is successfully executed.

Ex no : 12

Date : 03/04/2021

DYNAMIC XPATH IN SELENIUM USING DOUBLE SLASH

AIM:

To write a program Automate the test case by dynamic XPath in Selenium Using Double Slash.

ALGORITHM:

STEP 1 : Open Eclipse IDE

STEP 2 : Create a new java project and followed by create a new class named prg6

STEP 3 : Sets the system property to value Named Webdriver.chrome.driver and the path is Mentioned to get the chrome driver

STEP 4 : create an instance of the Chromedriver.driver=new chromedriver();

STEP 5 : get(); method automatically opens a new browser window and fetches the page that you Specify inside it's parenthesis.

Driver.get(https://www.w3schools.com/tags/tryit.asp?filename=tryhtml_button_test")

STEP 6 : Importing all necessary packages.

STEP 7 : findelement() finds a single web element and returns as a webelement selenium objects

STEP 8 : sendkeys() used to enter editable content in text fields during test execution.

STEP 9 : action() method handles keyboard and mouse events.

STEP 10 : Run the program and the new browser window automatically opened.

CODING:

```
package selenium;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.interactions.Actions;
public class project12 {

    public static void main(String[] args) throws
    InterruptedException {
        // TODO Auto-generated method stub

        System.setProperty("webdriver.chrome.driver","C:\\Users\\Admin\\eclipse\\java-neon\\eclipse\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();

        driver.get("https://portal2.passportindia.gov.in/AppOnlineProject/user/RegistrationBaseAction");
        driver.manage().window().maximize();
        Actions actions = new Actions(driver);

        actions.moveToElement(driver.findElement(By.xpath("//*[@id='passOffice']"))).perform();
        Thread.sleep(1000);

        actions.moveToElement(driver.findElement(By.id("passportoffices2"))).click();
        actions.perform();
        Thread.sleep(1000);

    }
}
```


OUTPUT:

The screenshot shows a web browser window with the URL `portal2.passportindia.gov.in/AppOnlineProject/mission/mission`. The page is titled "Passport Seva" and is part of the "Consular, Passport & Visa Division, Ministry of External Affairs, Government of India". The page layout includes a top navigation bar with links like "Home", "About Us", "Passport Offices", "Consular / Visa", "RTI", "Citizens' Charter", "Contact Us", and "What's New". A search bar is also present. The main content area is titled "Mission/Post Abroad" and features a form with a "Country*" dropdown menu and a "Go" button. A sidebar on the left contains an "Information Corner" with links to various services like "Getting Started", "Passport Act and Rules", "FAQs", "Locate Passport Seva Kendra", "Locate Common Service Centers", "Fee Calculator", "Appointment Availability Status", "Know your Police Station", "Quick Guides", "Tatkaal Appointment Opening Time", "Instructions Booklet", and "All India Network of Passport Services".

Thursday, April 01, 2021 | 12:57:17 PM

Skip to main content | Screen Reader Access | Sitemap | Home | A- | A | A+

Passport Seva
Consular, Passport & Visa Division
Ministry of External Affairs, Government of India

Home | About Us | Passport Offices | Consular / Visa | RTI | Citizens' Charter | Contact Us | What's New | Search...

Wishing you a Very Happy Int

You are here : Home > Passport Offices > Mission/Post Abroad

Mission/Post Abroad

Country*

Go

Fields marked with asterisk (*) are mandatory

Information Corner

- Getting Started
- Passport Act and Rules
- FAQs
- Locate Passport Seva Kendra
- Locate Common Service Centers
- Fee Calculator
- Appointment Availability Status
- Know your Police Station
- Quick Guides
- Tatkaal Appointment Opening Time
- Instructions Booklet
- All India Network of Passport Services

RESULT:

Using the above code the program is successfully executed.

