

Software Testing Quality Assurance Lab

Subject Code: MCAL35

A Practical Journal Submitted in Fulfilment of the Degree

of

MASTER

In

COMPUTER APPLICATION

Year 2024-2025

By

Mr. Agrawal Yash Gopal

(Application Id: - 53715)

Semester- III (CBCS)



Institute of Distance and Open Learning

Vidya Nagari, Kalina, Santacruz East – 400098.

University of Mumbai

PCP Centre

[Vidyavardhini's College of Technology – Vasai Road, Palghar 401202]



Institute of Distance and Open Learning,

Vidya Nagari, Kalina, Santacruz (E) -400098

CERTIFICATE

This to certify that, **Mr. Agrawal Yash Gopal** appearing **Master in Computer Application (Semester III - CBCS) Application ID: 53715** has satisfactorily completed the prescribed practical of **MCAL35 - Software Testing Quality Assurance Lab** as laid down by the University of Mumbai for the academic year 2024-25.

Teacher in charge

Examiners

Coordinator IDOL, MCA
University of Mumbai

Date: -12/01/2025

Place: - Vasai

Index

Sr. No.	Practical	Signature
1.	Implementing WebDrivers on Multiple Browser ie.Chrome	
2.	Implementing handling multiple frames	
3.	Implementing Selenium WebDriver – Browser Commands	
4.	Implementing Selenium WebDriver – find element command, Locator (id, css selector, Xpath), Input Box, Buttons, Submit Buttons	
5.	Demonstrate different types of alerts	
6.	Demonstrate CheckBox and Radio Button in Selenium WebDriver	
7.	Demonstrate synchronization in Selenium (ImpliciY Wait)	
8.	Select Value from DropDown using Selenium WebDriver	
9.	Demonstrate action classes using Selenium WebDriver (Mouse Events)	
10.	Functional Testing using Quality Assurance Equivalence Partitioning	

Practical 1

AIM: Implementing WebDrivers on Multiple Browser ie.Chrome

SOURCE CODE:

```
package PP;

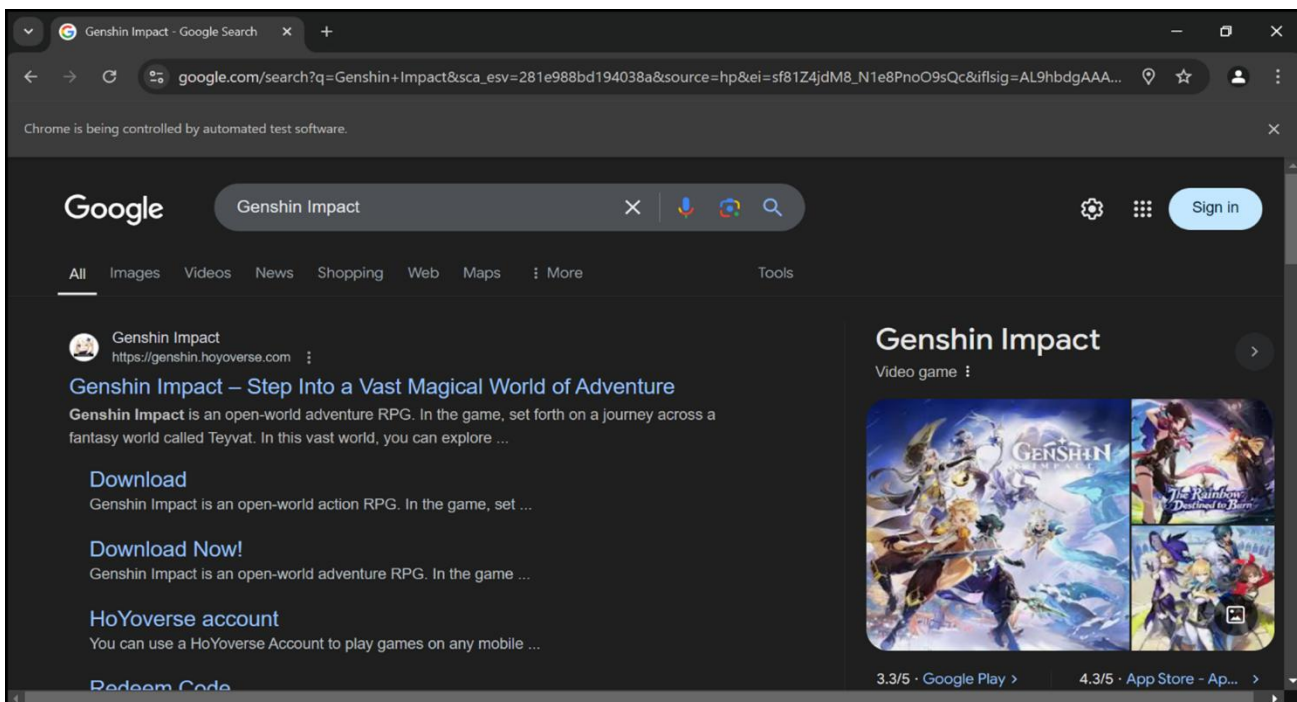
import org.openqa.selenium.By;
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.Keys;
import org.openqa.selenium.chrome.ChromeDriver;

public class PP1 {
    public static void main(String[] args) {
        ChromeDriver driver = new ChromeDriver();
        driver.get("https://www.google.com/");
        driver.manage().window().maximize();

        // Scroll down the webpage by 5000 pixels
        JavascriptExecutor js = (JavascriptExecutor)driver;
        js.executeScript("scrollBy(0, 5000)");

        driver.findElement(By.name("q")).sendKeys("Genshin Impact", Keys.ENTER);
    }
}
```

OUTPUT:



Practical 2

AIM: Implementing handling multiple frames

SOURCE CODE:

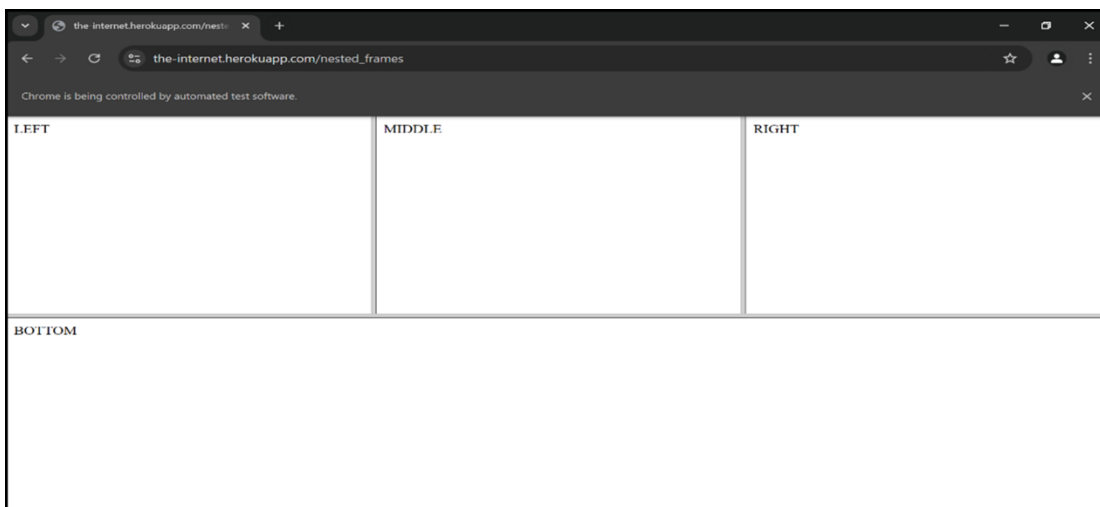
```
package PP;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
public class PP2 {
    public static void main(String[] args) {
        ChromeDriver driver = new ChromeDriver();
        driver.get("https://the-internet.herokuapp.com/frames");
        driver.manage().window().maximize();

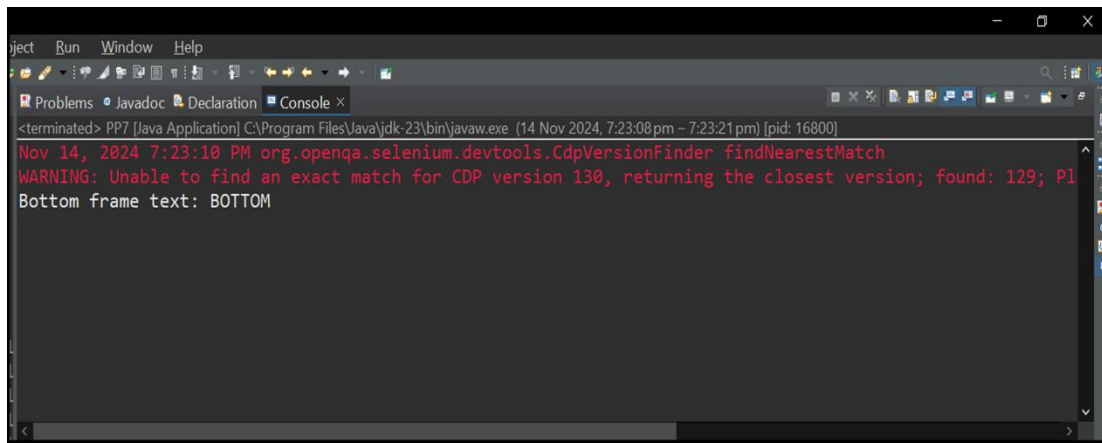
        // Click on the "Nested Frames" link to load the nested frames page
        driver.findElement(By.linkText("Nested Frames")).click();

        // Switch to the bottom frame by frame name and identify the text inside it
        driver.switchTo().frame("frame-bottom");
        WebElement l = driver.findElement(By.cssSelector("body"));
        System.out.println("Bottom frame text: " + l.getText());

        // Switch back to the main page
        driver.switchTo().defaultContent();
    }
}
```

OUTPUT:





The screenshot shows an IDE console window with the following text:

```
<terminated> PP7 [Java Application] C:\Program Files\Java\jdk-23\bin\javaw.exe (14 Nov 2024, 7:23:08 pm - 7:23:21 pm) [pid: 16800]  
Nov 14, 2024 7:23:10 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch  
WARNING: Unable to find an exact match for CDP version 130, returning the closest version; found: 129; Pl  
Bottom frame text: BOTTOM
```

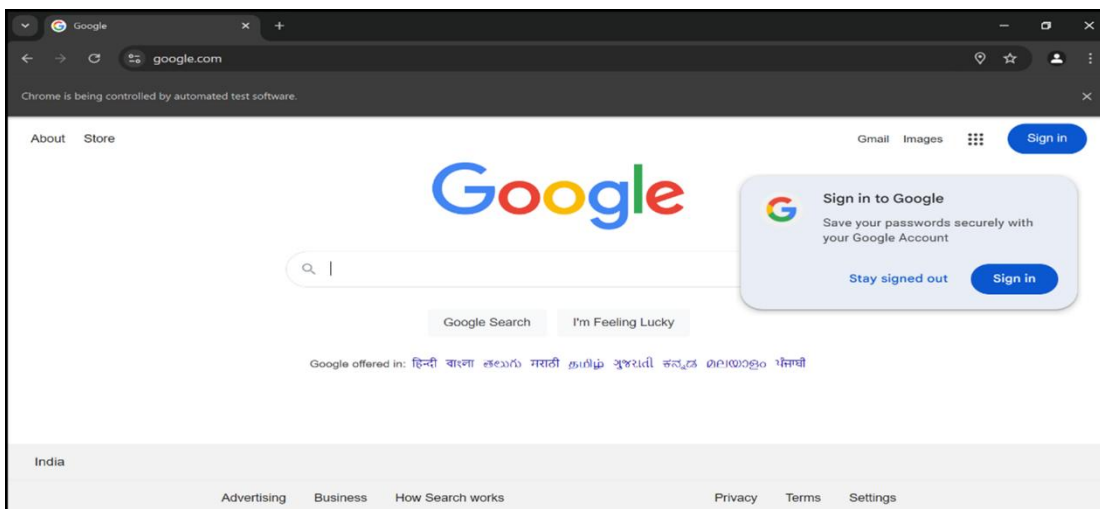
Practical 3

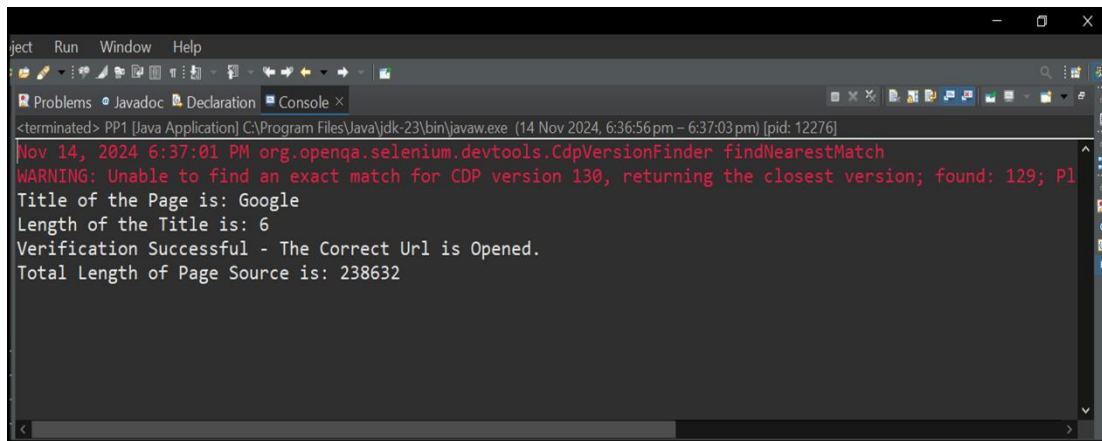
AIM: Implementing Selenium WebDriver – Browser Commands

SOURCE CODE:

```
package PP;  
import org.openqa.selenium.chrome.ChromeDriver;  
public class PP3 {  
    public static void main(String[] args) {  
        ChromeDriver driver = new ChromeDriver();  
        driver.get("https://www.google.com/");  
        driver.manage().window().maximize();  
  
        String title = driver.getTitle();  
        System.out.println("Title of the Page is: " + title);  
  
        int titleLength = driver.getTitle().length();  
        System.out.println("Length of the Title is: " + titleLength);  
  
        String actualUrl = driver.getCurrentUrl();  
        System.out.println("Verification Successful - The Correct Url is Opened.");  
  
        String pageSource = driver.getPageSource();  
  
        int pageSourceLength = pageSource.length();  
        System.out.println("Total Length of Page Source is: " + pageSourceLength);  
    }  
}
```

OUTPUT:





The screenshot shows an IDE's console window with the following text:

```
<terminated> PP1 [Java Application] C:\Program Files\Java\jdk-23\bin\javaw.exe (14 Nov 2024, 6:36:56 pm - 6:37:03 pm) [pid: 12276]  
Nov 14, 2024 6:37:01 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch  
WARNING: Unable to find an exact match for CDP version 130, returning the closest version; found: 129; Pl  
Title of the Page is: Google  
Length of the Title is: 6  
Verification Successful - The Correct Url is Opened.  
Total Length of Page Source is: 238632
```


Practical 4

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

SOURCE CODE:

```
package PP;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
public class PP4 {
    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();
        driver.get("http://demo.guru99.com/test/login.html");
        driver.manage().window().maximize();

        // Get the WebElement for Email and Password fields
        WebElement email = driver.findElement(By.id("email"));
        WebElement password = driver.findElement(By.name("passwd"));

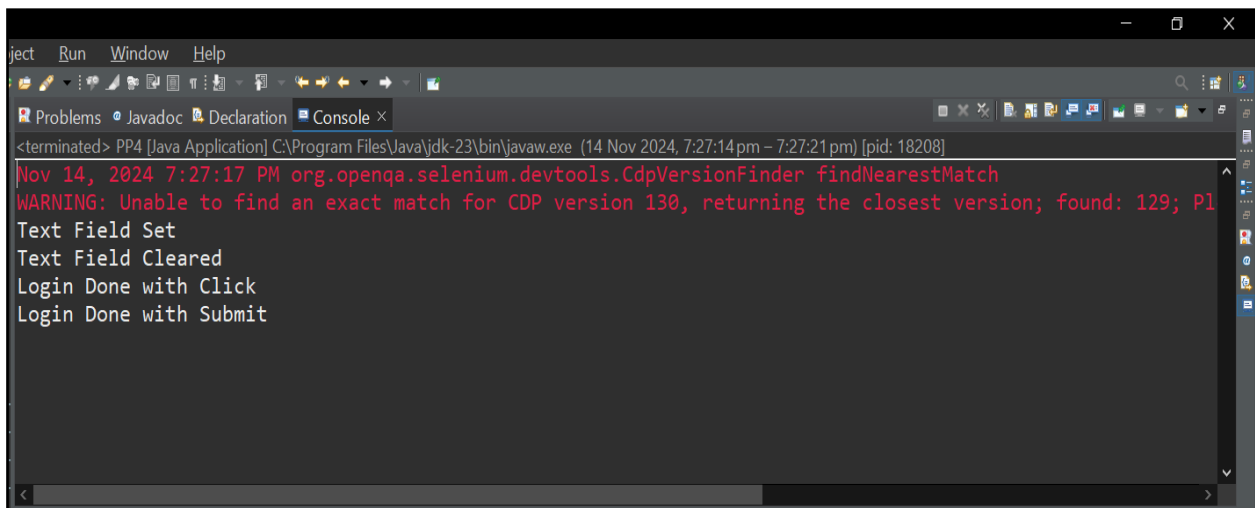
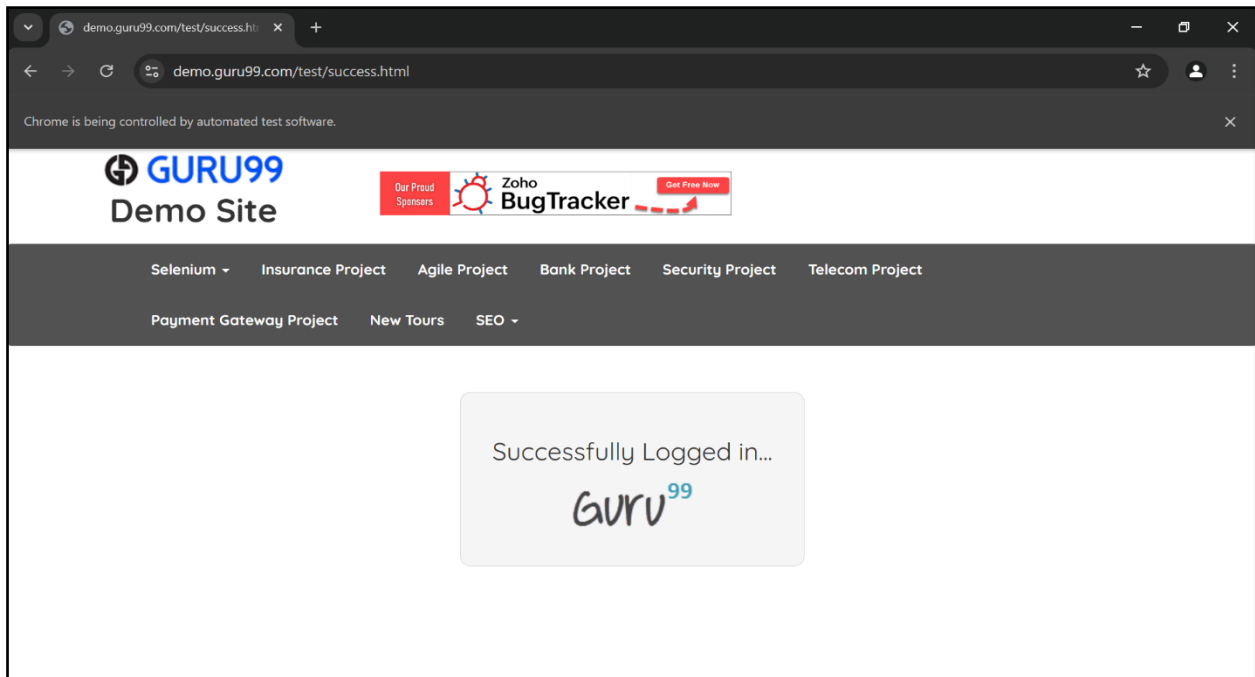
        // Enter email and password
        email.sendKeys("abcd@gmail.com");
        password.sendKeys("abcdefghijkl");
        System.out.println("Text Field Set");

        // Clear the text fields
        email.clear();
        password.clear();
        System.out.println("Text Field Cleared");

        // Enter email and password again and submit the form with click
        email.sendKeys("abcd@gmail.com");
        password.sendKeys("abcdefghijkl");
        WebElement login = driver.findElement(By.id("SubmitLogin"));
        login.click();
        System.out.println("Login Done with Click");

        // Reopen the page to demonstrate the submit method
        driver.get("http://demo.guru99.com/test/login.html");
        driver.findElement(By.id("email")).sendKeys("abcd@gmail.com");
        driver.findElement(By.name("passwd")).sendKeys("abcdefghijkl");
        driver.findElement(By.id("SubmitLogin")).submit();
        System.out.println("Login Done with Submit");
    }
}
```

OUTPUT:



Practical 5

AIM: Demonstrate different types of alerts

SOURCE CODE:

```
package PP;
import org.openqa.selenium.Alert;
import org.openqa.selenium.By;
import org.openqa.selenium.chrome.ChromeDriver;
public class PP5 {
    public static void main(String[] args) throws InterruptedException {
        ChromeDriver driver = new ChromeDriver();
        driver.get("https://demo.guru99.com/test/delete_customer.php");
        driver.manage().window().maximize();

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

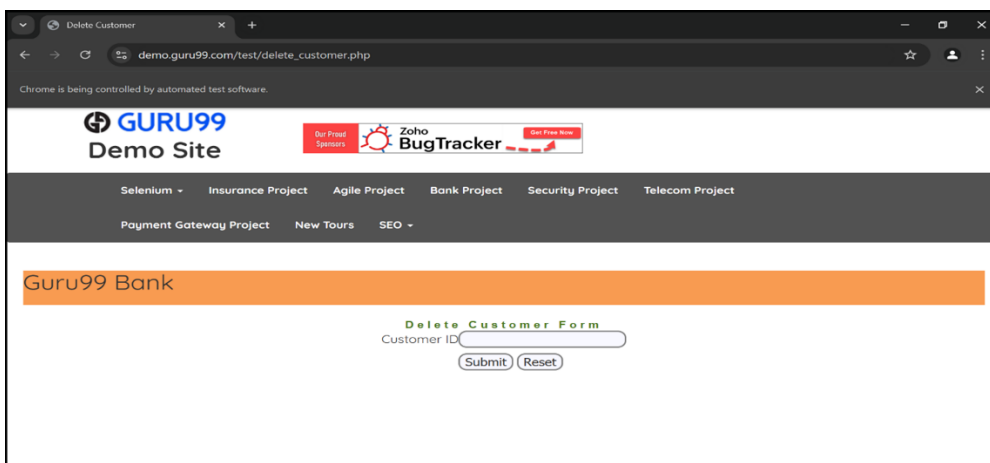
        // Switching to Alert
        Alert alert = driver.switchTo().alert();

        // Capturing alert message.
        String alertMessage= driver.switchTo().alert().getText();

        // Displaying alert message
        System.out.println(alertMessage);
        Thread.sleep(5000);

        // Accepting alert
        alert.accept();
    }
}
```

OUTPUT:



Practical 6

AIM: Demonstrate CheckBox and Radio Button in Selenium WebDriver

SOURCE CODE:

```
package PP;
import org.openqa.selenium.By;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
public class PP6 {
    public static void main(String[] args) {
        ChromeDriver driver = new ChromeDriver();
        driver.get("http://demo.guru99.com/test/radio.html");
        driver.manage().window().maximize();

        WebElement radio1 = driver.findElement(By.id("vfb-7-1"));
        WebElement radio2 = driver.findElement(By.id("vfb-7-2"));

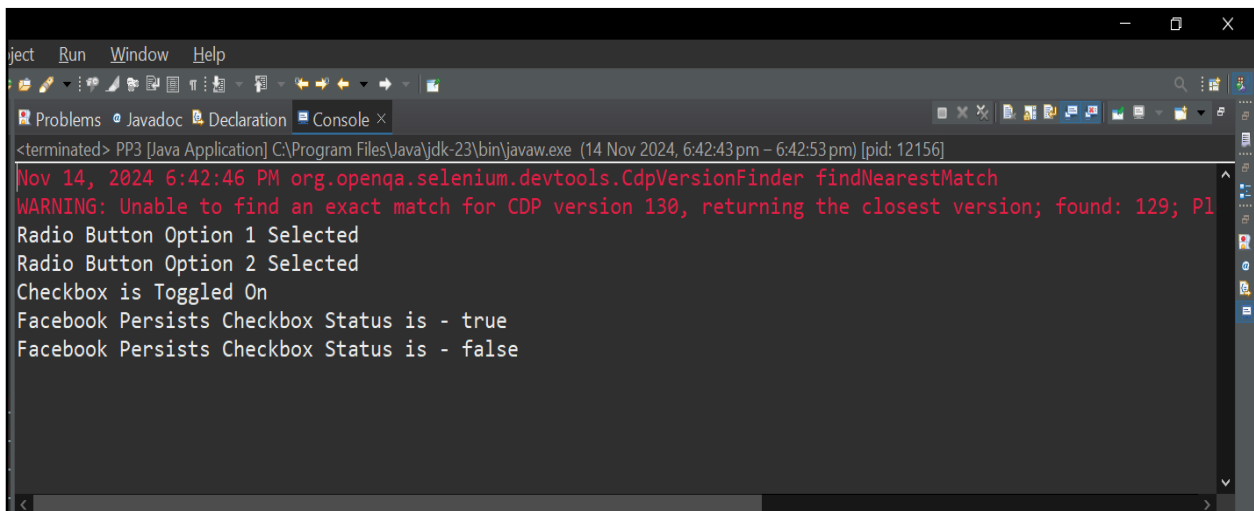
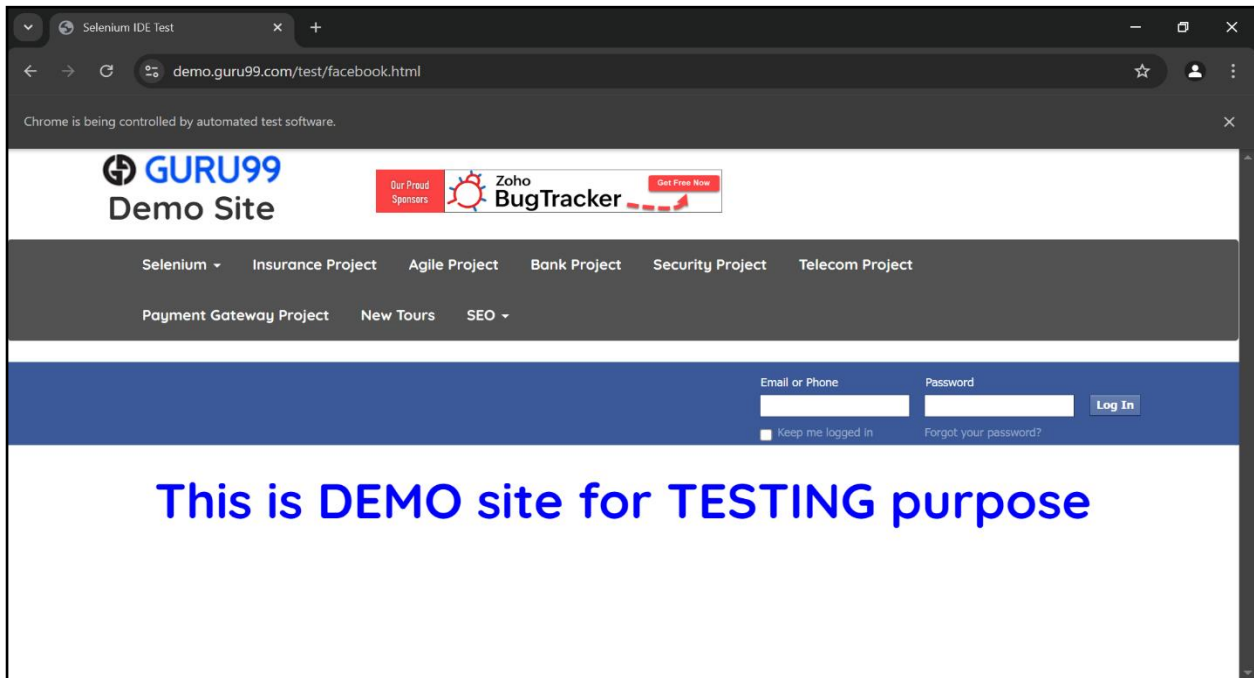
        //Radio Button1 is selected
        radio1.click();
        System.out.println("Radio Button Option 1 Selected");

        //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");
        }

        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());
        }
    }
}
```

OUTPUT:



Practical 7

AIM: Demonstrate synchronization in Selenium (Implicit Wait)

SOURCE CODE:

```
package PP;
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class PP7 {
    public static void main(String[] args) throws InterruptedException {
        ChromeDriver driver = new ChromeDriver();

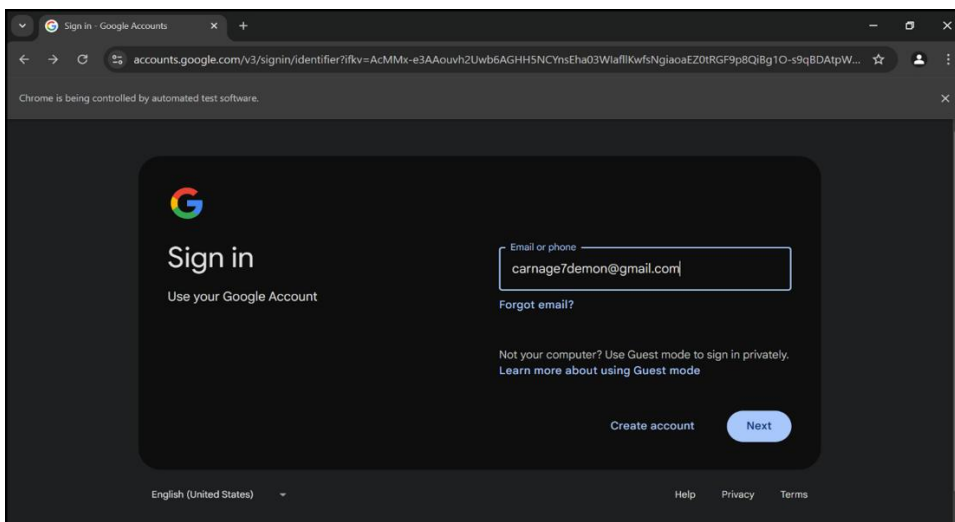
        // Maximize window and delete all cookies
        driver.manage().window().maximize();
        driver.manage().deleteAllCookies();

        // Set page load timeout and implicit wait
        driver.manage().timeouts().pageLoadTimeout(40, TimeUnit.SECONDS);
        driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS);

        // Open the Google login page
        driver.get("https://accounts.google.com/signin");

        // Enter username and click on the next button
        driver.findElement(By.id("identifierId")).sendKeys("carnage7demon@gmail.com");
        Thread.sleep(1000); // Pause for a second before clicking next
        driver.findElement(By.xpath("//span[text()='Next']")).click();
    }
}
```

OUTPUT:



Practical 8

AIM: Select Value from DropDown using Selenium WebDriver

SOURCE CODE:

```
package PP;
import java.util.List;
import org.openqa.selenium.By;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.Select;
public class PP8 {
    public static void main(String[] args) {
        ChromeDriver driver = new ChromeDriver();
        driver.get("https://demoqa.com/select-menu");
        driver.manage().window().maximize();
        Select select = new Select(driver.findElement(By.id("oldSelectMenu")));

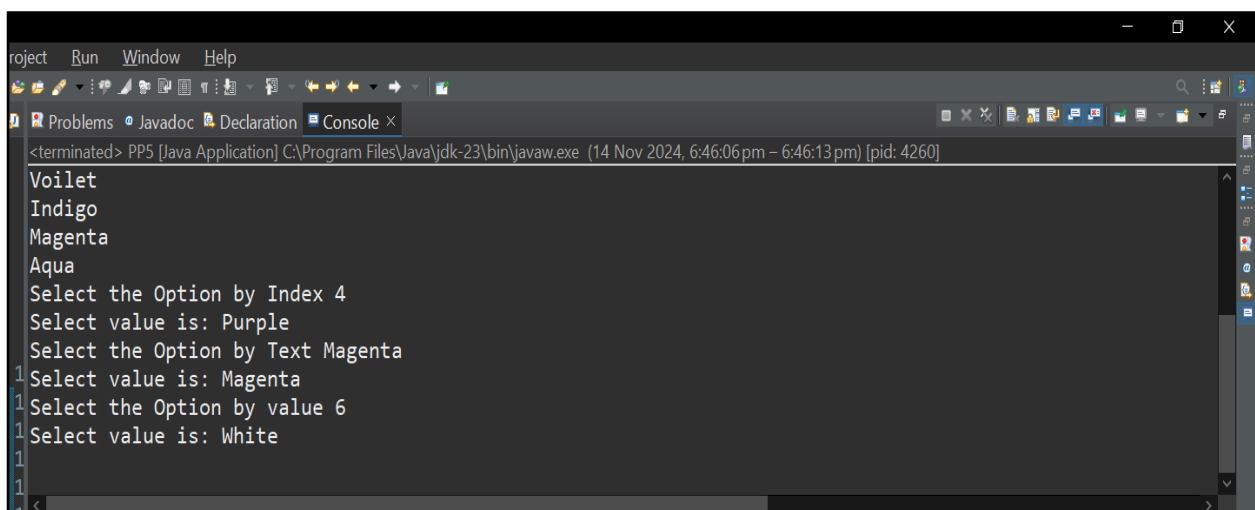
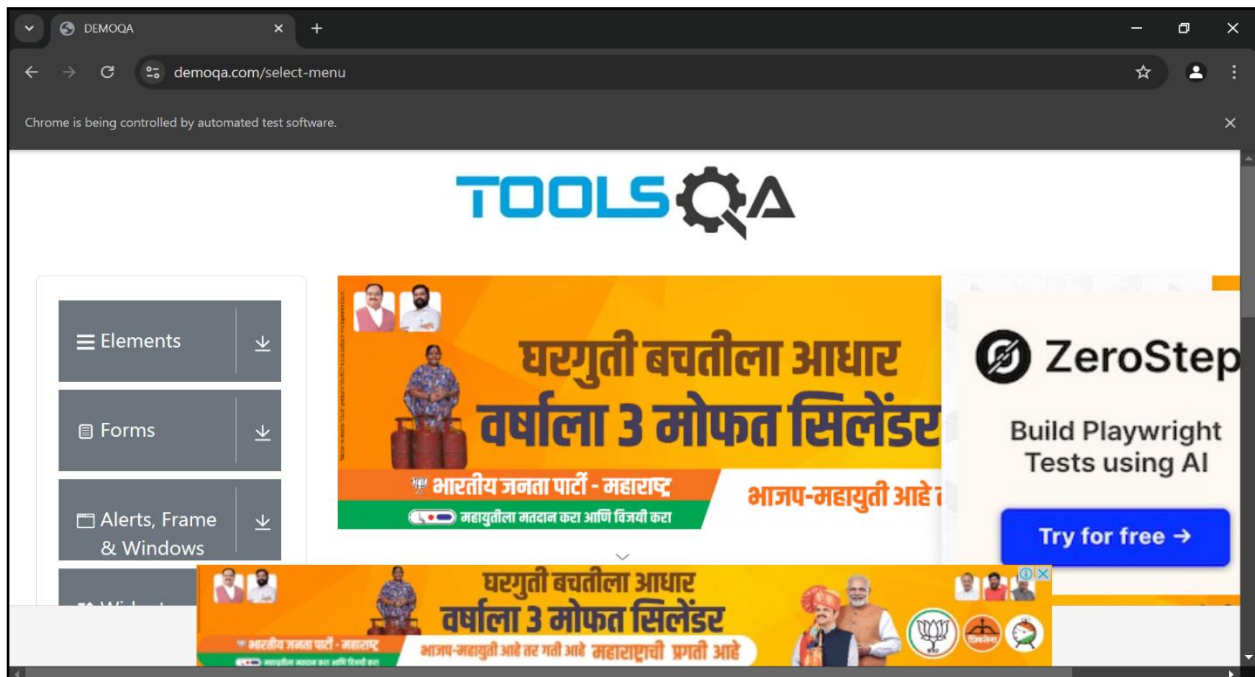
        List<WebElement> lst = select.getOptions();
        System.out.println("The dropdown options are:");

        for(WebElement options: lst)
            System.out.println(options.getText());
        System.out.println("Select the Option by Index 4");

        select.selectByIndex(4);
        System.out.println("Select value is: " + select.getFirstSelectedOption().getText());
        System.out.println("Select the Option by Text Magenta");

        select.selectByVisibleText("Magenta");
        System.out.println("Select value is: " + select.getFirstSelectedOption().getText());
        System.out.println("Select the Option by value 6");
        select.selectByValue("6");
        System.out.println("Select value is: " + select.getFirstSelectedOption().getText());
    }
}
```

OUTPUT:



Practical 9

AIM: Demonstrate action classes using Selenium WebDriver (Mouse Events)

SOURCE CODE:

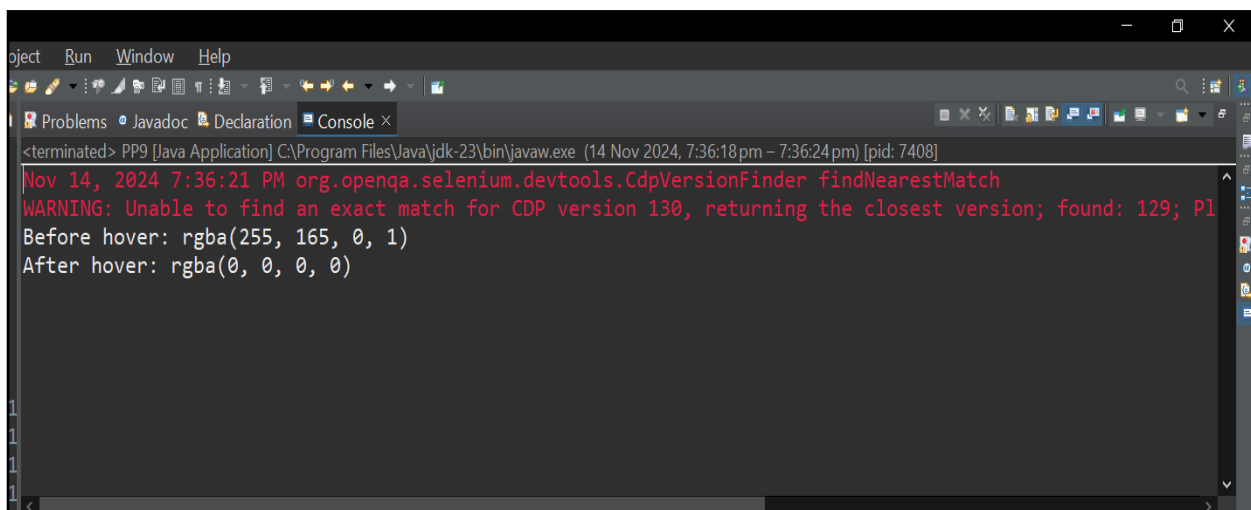
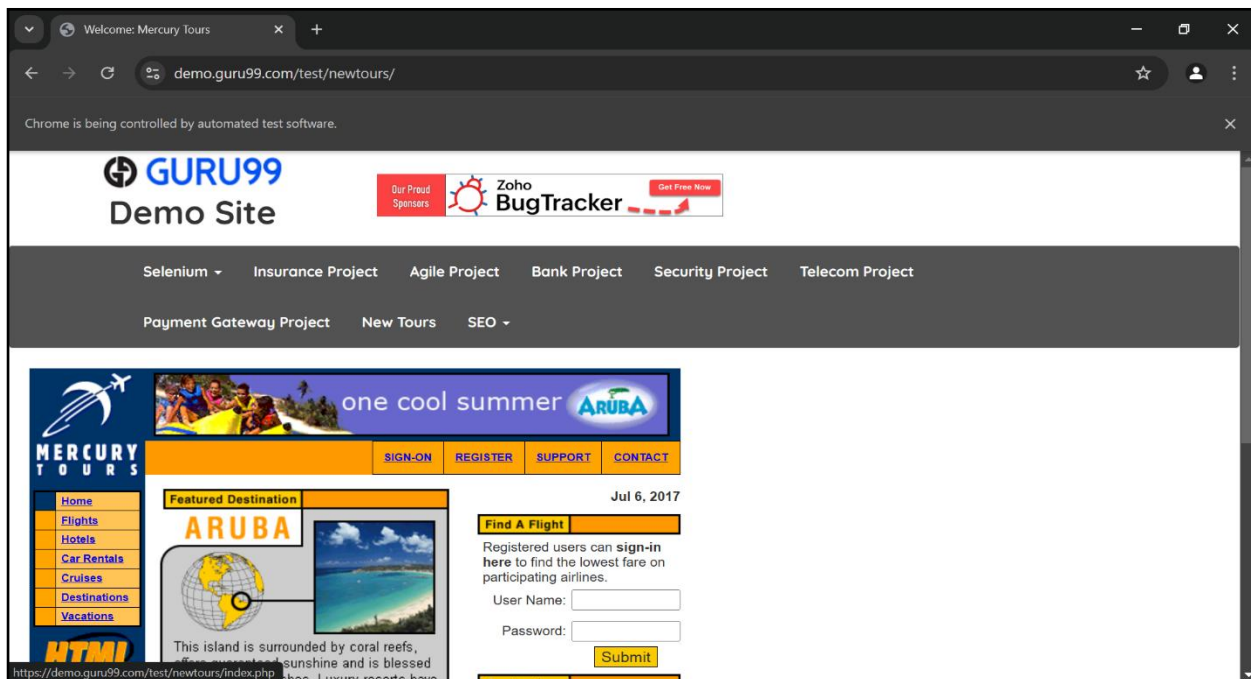
```
package PP;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.interactions.Action;
import org.openqa.selenium.interactions.Actions;
public class PP9 {
    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();
        driver.get("http://demo.guru99.com/test/newtours/");
        driver.manage().window().maximize();

        // Locate the "Home" link and surrounding table cell for background color check
        WebElement link_Home = driver.findElement(By.linkText("Home"));
        WebElement td_Home =
driver.findElement(By.xpath("//table/tbody/tr/td/table/tbody/tr/td/table/tbody/tr/td/table/tbody/tr"));

        // Initialize Actions class and create a mouse over action
        Actions builder = new Actions(driver);
        Action mouseOverHome = builder.moveToElement(link_Home).build();

        // Get and print background color before and after hover
        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);
    }
}
```

OUTPUT:



Practical 10

AIM: Functional Testing using Quality Assurance Equivalence Partitioning

QUESTION:

3% rate of interest is given if the balance in the account is in the range of \$0 to \$100, 5% rate of interest is given if the balance in the account is in the range of \$100 to \$1000, and 7% rate of interest is given if the balance in the account is \$1000 and above, we would initially identify three valid equivalence partitions and one invalid partition as shown below.

Partition 1: balance 0-100

Valid Inputs - 0-100- ≥ 0 And ≤ 100

Invalid Input- <0 , \$, #, @ A-Z

Partition 2: balance 100-1000

Valid Input - 100-1000 (>100 And ≤ 1000)

Invalid Input- \$, #, @ A-Z

Partition 3: balance >1000

Valid Input- >1000

Invalid Input- \$, #, @ A-Z

Invalid Partition	Valid Partition	Valid Partition
<0 and 0-100	100-1000	>1000
3% Interest	5% Interest	7% Interest

Test Case:

Test Case ID	Test Input Account Balance	Expected Output
B001	-90	Invalid Input
B002	50	3% Interest
B003	900	5% Interest
B004	2500	7% Interest
B005	A	Invalid Input
B006	\$	Invalid Input