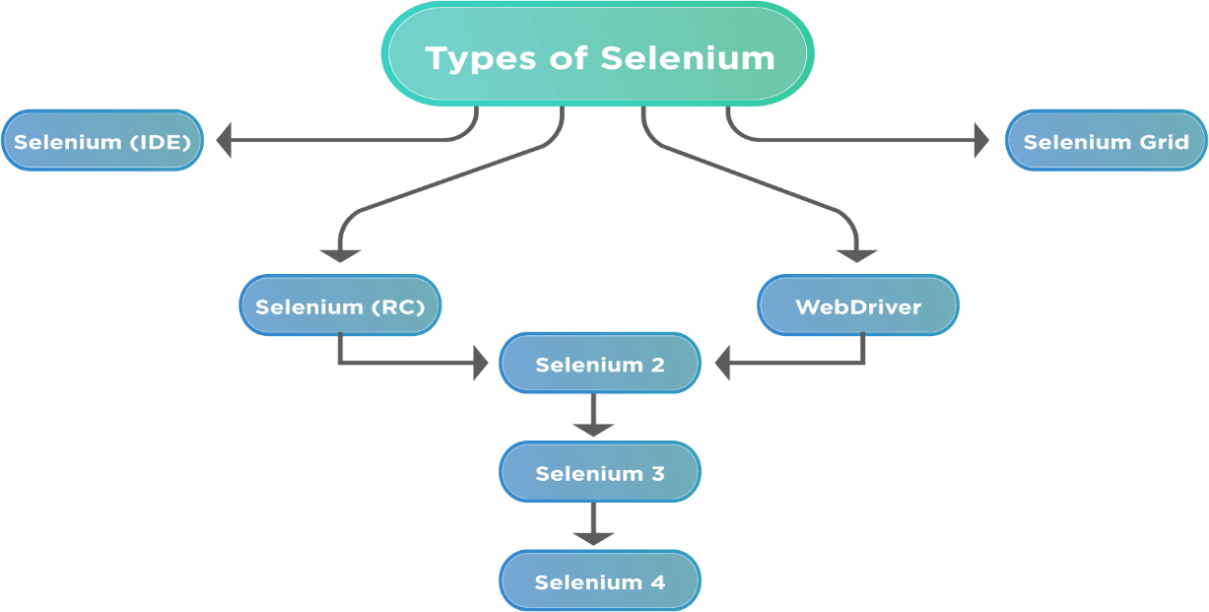
SELENIUM TESTING

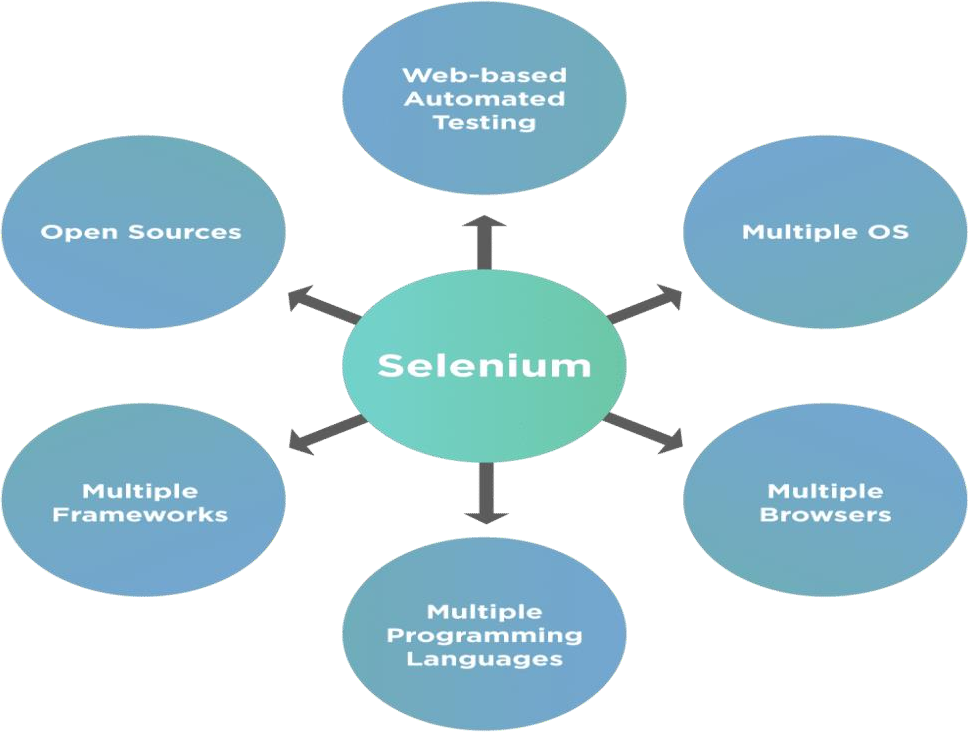
**Selenium**

Selenium is a free (open-source) automated testing framework used to validate web applications across different browsers and platforms.



#### Features of selenium :

* Multi-Browser Support. ...
* Multi-Language Compatibility. ...
* Easy Identification and Use of Web Elements. ...
* Performance and Speed. ...
* Dynamic Web Elements. ...
* Open Source. ...
* Portability (Ability to work with different Operating Systems) ...
* Reusability and Extras.



#### Locators By ID:

WebElement elementById = driver.findElement(By.id("elementId"));

#### By Name:

WebElement elementByName = driver.findElement(By.name("elementName"));

#### By XPath:

WebElement elementByXPath = driver.findElement(By.xpath("//input[@id='elementId']"));

#### By CSS Selector:

WebElement elementByCssSelector = driver.findElement(By.cssSelector("#elementId"));

#### By Class Name:

WebElement elementByClassName = driver.findElement(By.className("className"));

#### By Tag Name:

WebElement elementByTagName = driver.findElement(By.tagName("input"));

#### By Link Text:

WebElement elementByLinkText = driver.findElement(By.linkText("Link Text"));

#### By Partial Link Text:

WebElement elementByPartialLinkText = driver.findElement(By.partialLinkText("Partial Link Text"));

**Types of Xpath In Selenium**

**Absolute Xpath**

/html/body/div[2]/div[1]/div/h4[1]/b/html[1]/body[1]/div[2]/div[1]/div[1]/h4

[1]/b[1]

**Relative Xpath:**

Relative XPath: //div[@class='featured-box cloumnsize1']//h4[1]//b[1]

**Dynamic XPath In Selenium**

1. Basic Xpath

Xpath=//input[@name='uid']

2. Contains()

Xpath=//\*[contains(@name,'btn')]

3. Using OR & AND

Xpath=//\*[@type='submit' or @name='btnReset']

Xpath=//input[@type='submit' and @name='btnLogin']

4. Xpath Starts-with

Xpath=//label[starts-with(@id,'message')]

5. XPath Text() Function

Xpath=//td[text()='UserID']

**XPath axes methods**

1. Following

a. Xpath=//\*[@type='text']//following::input

b. Xpath=//\*[@type='text']//following::input[1]

2. Ancestor

a. Xpath=//\*[text()='Enterprise Testing']//ancestor::div

b. Xpath=//\*[text()='Enterprise Testing']//ancestor::div[1]

3. Child

a. Xpath=//\*[@id='java\_technologies']//child::li

b. Xpath=//\*[@id='java\_technologies']//child::li[1]

4. Preceding

a. Xpath=//\*[@type='submit']//preceding::input

b. Xpath=//\*[@type='submit']//preceding::input[1]

5. Following-sibling

Xpath=//\*[@type='submit']//following-sibling::input

6. Parent

a. Xpath=//\*[@id='rt-feature']//parent::div

b. Xpath=//\*[@id='rt-feature']//parent::div[1]

7. Self

Xpath =//\*[@type='password']//self::input

8. Descendant

a. Xpath=//\*[@id='rt-feature']//descendant::a

b. Xpath=//\*[@id='rt-feature']//descendant::a[1]

#### isDisplayed, isEnabled, isSelected Methods in Selenium isSelected:

* isSelected is a method applicable to checkboxes, radio buttons, and options within a dropdown.
* It checks if an element is selected or not, primarily used with input elements like checkboxes and radio buttons.
* Returns a boolean value - true if the element is selected, false if not.

#### isEnabled:

* isEnabled is a method applicable to input elements like text fields, buttons, etc.
* It checks if an element is enabled for interaction or not, considering factors like being visible, not disabled, etc.
* Returns a boolean value - true if the element is enabled, false if not.

#### isDisplayed:

* isDisplayed is a method applicable to all web elements (e.g., buttons, links, text fields, etc.).
* It checks if an element is currently visible on the page or not.
* Returns a boolean value - true if the element is displayed, false if not.

**PROGRAMS**

1. ALERT

package selenium;

import org.openqa.selenium.Alert;

import org.openqa.selenium.By;

import org.openqa.selenium.NoAlertPresentException;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class alert {

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

System.setProperty("webdriver.chrome.driver","C:/Users/Admin/Downloads/chromedriver-win64/chromedriver-win64/chromedriver.exe");

WebDriver driver = new ChromeDriver();

// Alert Message handling

driver.get("https://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);

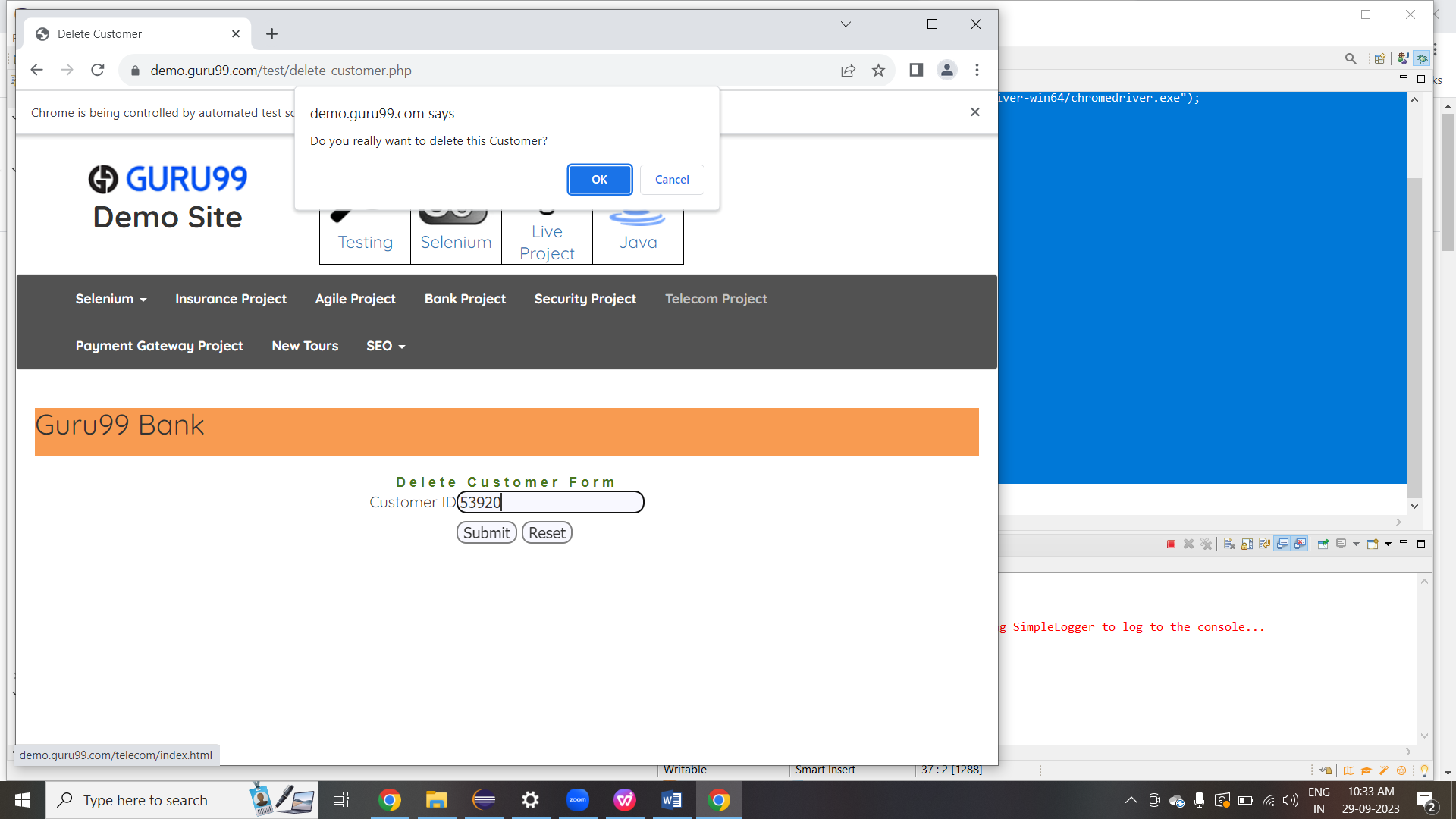
// Accepting alert

alert.accept();

}

}

OUTPUT :



1. CHECK BOX :

package selenium;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class Check {

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

System.setProperty("webdriver.chrome.driver","C:/Users/Admin/Downloads/chromedriver-win64/chromedriver-win64/chromedriver.exe");

WebDriver driver=new ChromeDriver();

driver.get("https://demoqa.com/checkbox");

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

WebElement checkBoxSelected = driver.findElement(By.cssSelector("#tree-node > ol > li > span > label > span.rct-checkbox > svg"));

boolean isSelected = checkBoxSelected.isSelected();

System.out.println(isSelected);

// performing click operation if element is not selected

if(isSelected == false) {

checkBoxSelected.click();

}

/\*WebElement checkBoxDisplayed = driver.findElement(By.cssSelector("#tree-node > ol > li > span > label > span.rct-checkbox > svg > path"));

boolean isDisplayed = checkBoxDisplayed.isDisplayed();

System.out.println(isDisplayed);

// performing click operation if element is displayed

if (isDisplayed == true) {

checkBoxDisplayed.click();

}

WebElement checkBoxEnabled = driver.findElement(By.cssSelector("#tree-node > ol > li > span > label > span.rct-checkbox > svg > path"));

boolean isEnabled = checkBoxEnabled.isEnabled();

System.out.println(isEnabled);

// performing click operation if element is enabled

if (isEnabled == true) {

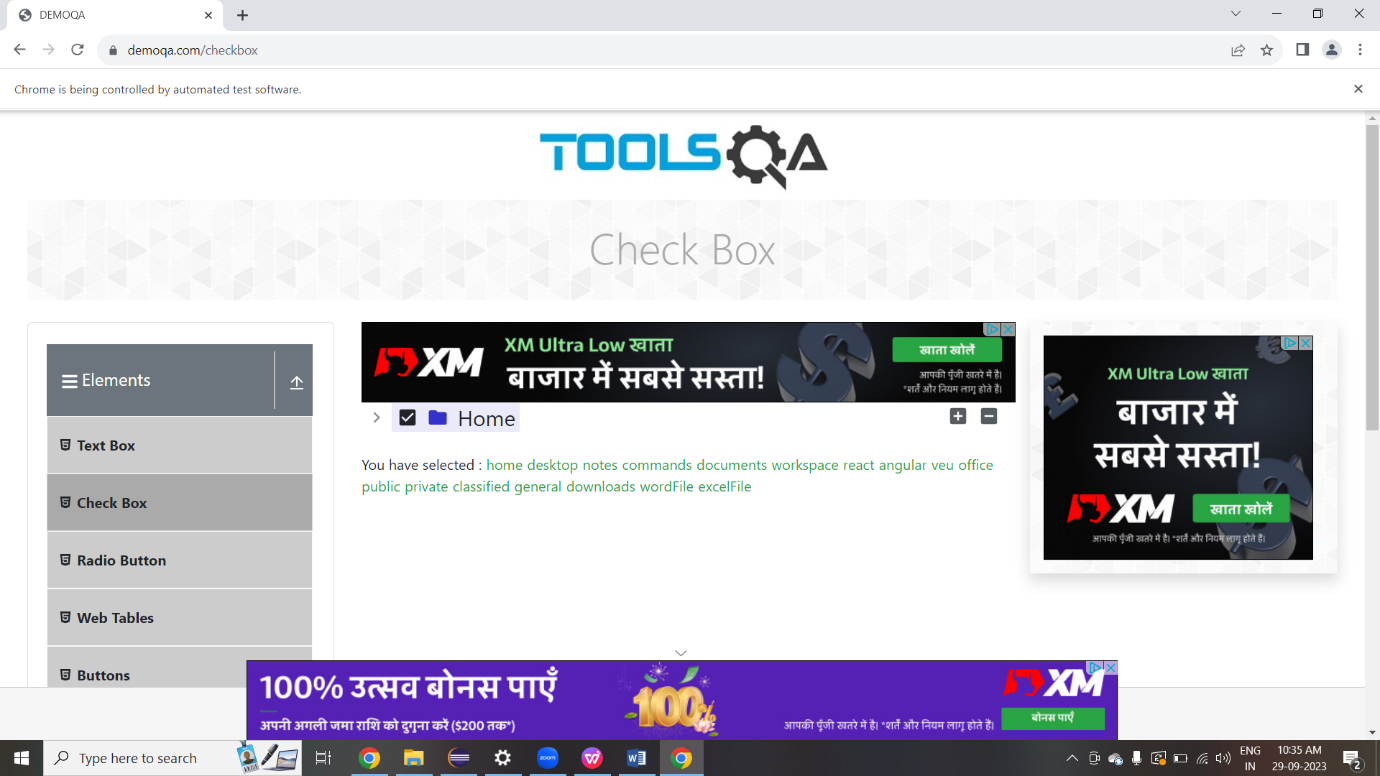
checkBoxEnabled.click();

}

\*/

}

}



1. DROP DOWN :

package selenium;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

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

import org.openqa.selenium.By;

public class Dropdown {

public static void main(String[] args) {

System.setProperty("webdriver.chrome.driver","C:/Users/Admin/Downloads/chromedriver-win64/chromedriver-win64/chromedriver.exe");

WebDriver driver=new ChromeDriver();

driver.get("https://www.facebook.com/campaign/landing.php?");

Select drpCountry = new Select(driver.findElement(By.name("Date of Birth")));

drpCountry.selectByVisibleText("11");

//Selecting Items in a Multiple SELECT elements

driver.get("http://jsbin.com/osebed/2");

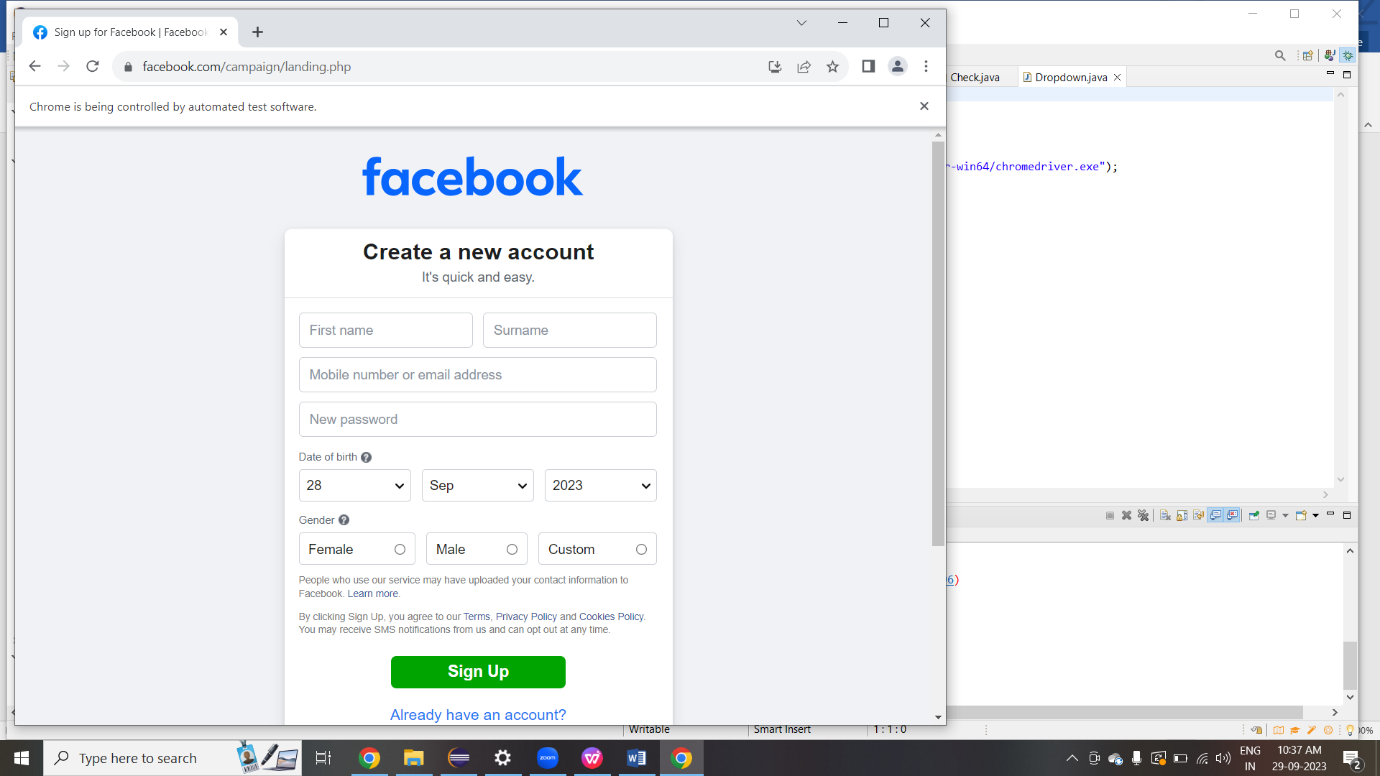
Select fruits = new Select(driver.findElement(By.id("fruits")));

fruits.selectByVisibleText("Banana");

fruits.selectByIndex(1);

}

}



1. EXCEL SHEET :

package selenium;

import org.apache.poi.ss.usermodel.\*;

import org.apache.poi.xssf.usermodel.XSSFWorkbook;

import java.io.\*;

import java.util.\*;

public class Excel {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter your username: ");

String username = sc.nextLine();

System.out.print("Enter your password: ");

int password = sc.nextInt();

sc.close();

try {

boolean loginValid = isLoginValid("D://login1.xlsx", username, password);

// Append the result to the output Excel file

writeToResultExcel(loginValid);

if (loginValid) {

System.out.println("Login successful!");

} else {

System.out.println("Login failed!");

}

} catch (IOException e) {

e.printStackTrace();

}

}

private static boolean isLoginValid(String filePath, String username, int password) {

// TODO Auto-generated method stub

return false;

}

public static boolean isLoginValid(String filePath, String username, String password) throws IOException {

FileInputStream file = new FileInputStream(new File(filePath));

Workbook workbook = new XSSFWorkbook(file);

Sheet sheet = workbook.getSheetAt(0);

Iterator<Row> rowIterator = sheet.iterator();

while (rowIterator.hasNext()) {

Row row = rowIterator.next();

Cell usernameCell = row.getCell(0); // username is in the first column

Cell passwordCell = row.getCell(1); // password is in the second column

String storedUsername = usernameCell.getStringCellValue();

String storedPassword = passwordCell.getStringCellValue();

if (storedUsername.equals(username) && storedPassword.equals(password)) {

workbook.close();

file.close();

return true; // Login successful

}

}

workbook.close();

file.close();

return false; // Login failed

}

public static void writeToResultExcel(boolean loginValid) throws IOException {

FileInputStream resultFile = null;

Workbook workbook = null;

try {

// Open the existing result Excel illana it create automatiaclly

File outputFile = new File("D://login1\_result.xlsx");

if (outputFile.exists()) {

resultFile = new FileInputStream(outputFile);

workbook = new XSSFWorkbook(resultFile);

} else {

workbook = new XSSFWorkbook();

}

Sheet sheet = workbook.getSheet("Login Result");

if (sheet == null) {

sheet = workbook.createSheet("Login Result");

}

int lastRowNum = sheet.getLastRowNum();

Row row = sheet.createRow(lastRowNum + 1);

Cell cell = row.createCell(0);

if (loginValid) {

cell.setCellValue("Pass");

} else {

cell.setCellValue("Fail");

}

FileOutputStream fileOut = new FileOutputStream(outputFile);

workbook.write(fileOut);

fileOut.close();

} finally {

if (workbook != null) {

workbook.close();

}

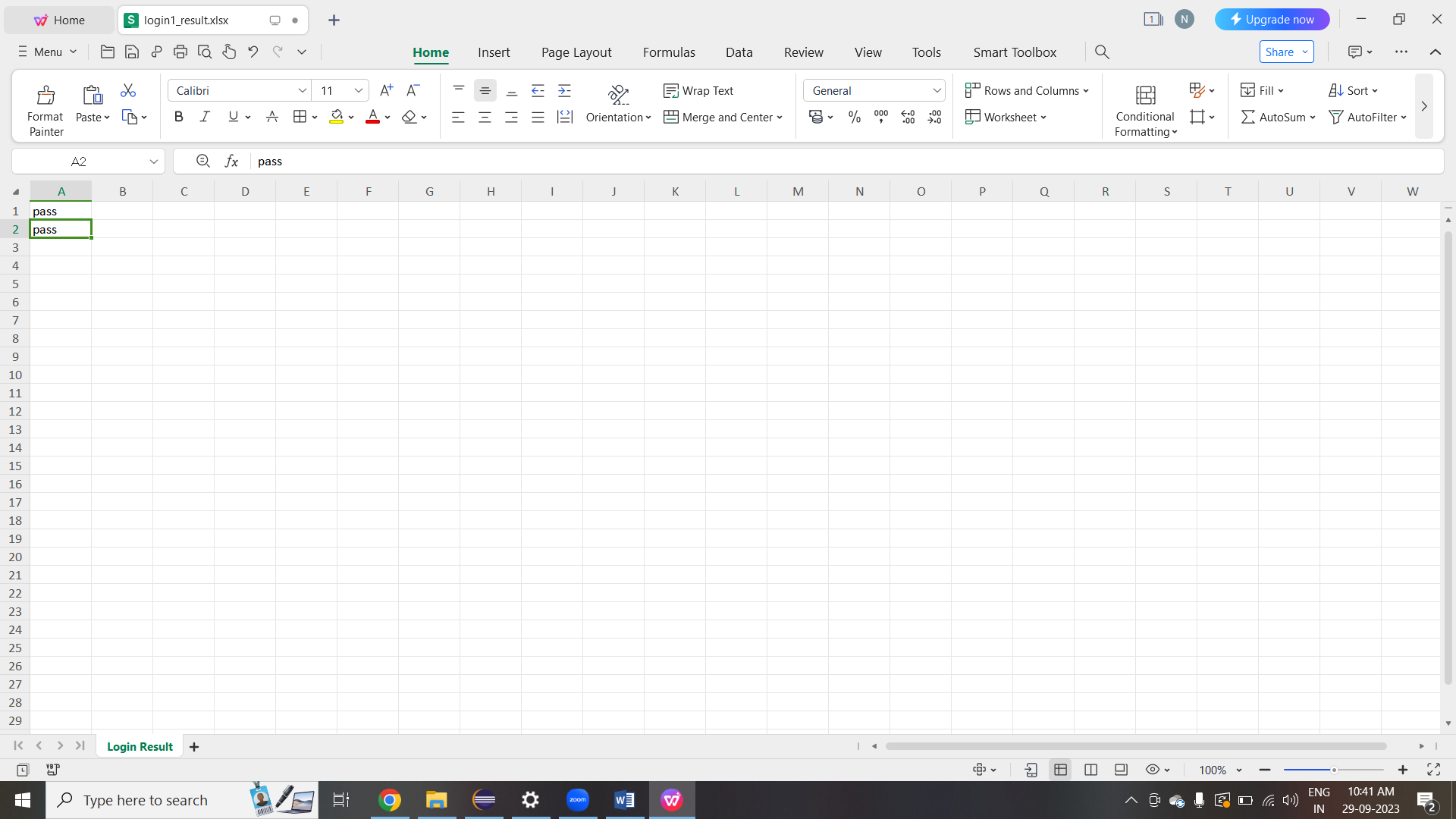
if (resultFile != null) {

resultFile.close();

}

}

}

}

1. Explicit and Implicit :

package selenium;

import java.time.Duration;

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 Explicit {

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

System.setProperty("webdriver.chrome.driver", "C:/Users/Admin/Downloads/chromedriver-win64/chromedriver-win64/chromedriver.exe");

WebDriver driver = new ChromeDriver();

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

driver.manage().deleteAllCookies();

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

driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);

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

WebElement firstname= driver.findElement(By.name("firstname"));

WebElement lastname= driver.findElement(By.name("lastname"));

sendKeys(driver, firstname, 10, "Nithya");

sendKeys(driver, lastname, 20, "Mani");

WebElement forgotAccount= driver.findElement(By.linkText("Forgotten account?"));

clickOn(driver,forgotAccount, 10);

driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);

}

private static void clickOn(WebDriver driver, WebElement forgotAccount, int i) {

// TODO Auto-generated method stub

}

private static void sendKeys(WebDriver driver, WebElement firstname, int i, String value) {

// TODO Auto-generated method stub

}

public static void sendKeys(WebDriver driver1, WebElement element, Duration timeout, String value){

new WebDriverWait(driver1, timeout).until(ExpectedConditions.visibilityOf(element));

element.sendKeys(value);

}

public static void clickOn(WebDriver driver1, WebElement element, Duration timeout){

new WebDriverWait(driver1, timeout).until(ExpectedConditions.elementToBeClickable(element));

element.click();

}

}

package selenium;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class Implicit{

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

System.setProperty("webdriver.chrome.driver", "C:/Users/Admin/Downloads/chromedriver-win64/chromedriver-win64/chromedriver.exe");

WebDriver driver = new ChromeDriver();

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.yahoo.com/");

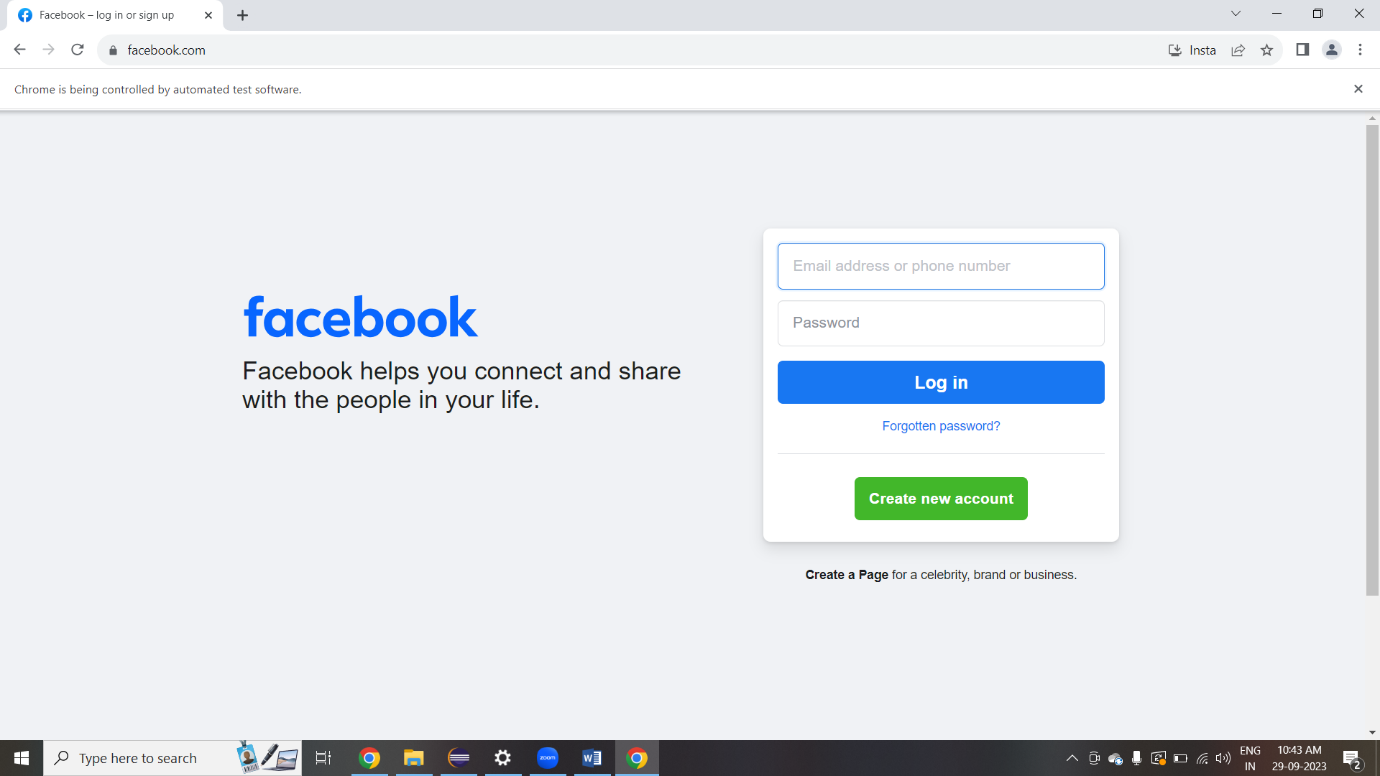
driver.findElement(By.xpath("//input[@id='login-username']")).sendKeys("nithya@yahoo.com"); //Finding element and sending values

Thread.sleep(1000);

driver.findElement(By.xpath("//input[@id='login-signin']")).click(); //Clicking on the next button if element is located

}

}



1. Locators :

package selenium;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class Locators {

public static void main(String[] args) {

System.setProperty("webdriver.chrome.driver","C:/Users/Admin/Downloads/chromedriver-win64/chromedriver-win64/chromedriver.exe");

WebDriver driver = new ChromeDriver();

// id locator

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

driver.findElement(By.id("login-username")).sendKeys("nithya@yahoo.com"); //id locator for text box

WebElement searchIcon = driver.findElement(By.id("login-signin"));//id locator for next button

searchIcon.click();

//name locator

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

driver.findElement(By.name("username")).sendKeys("nithya@yahoo.com"); //name locator for text box

WebElement searchIcon1 = driver.findElement(By.name("signin"));//name locator for next button

searchIcon.click();

// link test

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

driver.findElement(By.linkText("Trouble Signing in?")).click();//linkText locator for links

// css selector

driver.findElement(By.cssSelector("#login-username")).sendKeys("edureka@yahoo.com");

driver.findElement(By.cssSelector("#login-signin")).click();

//x path

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

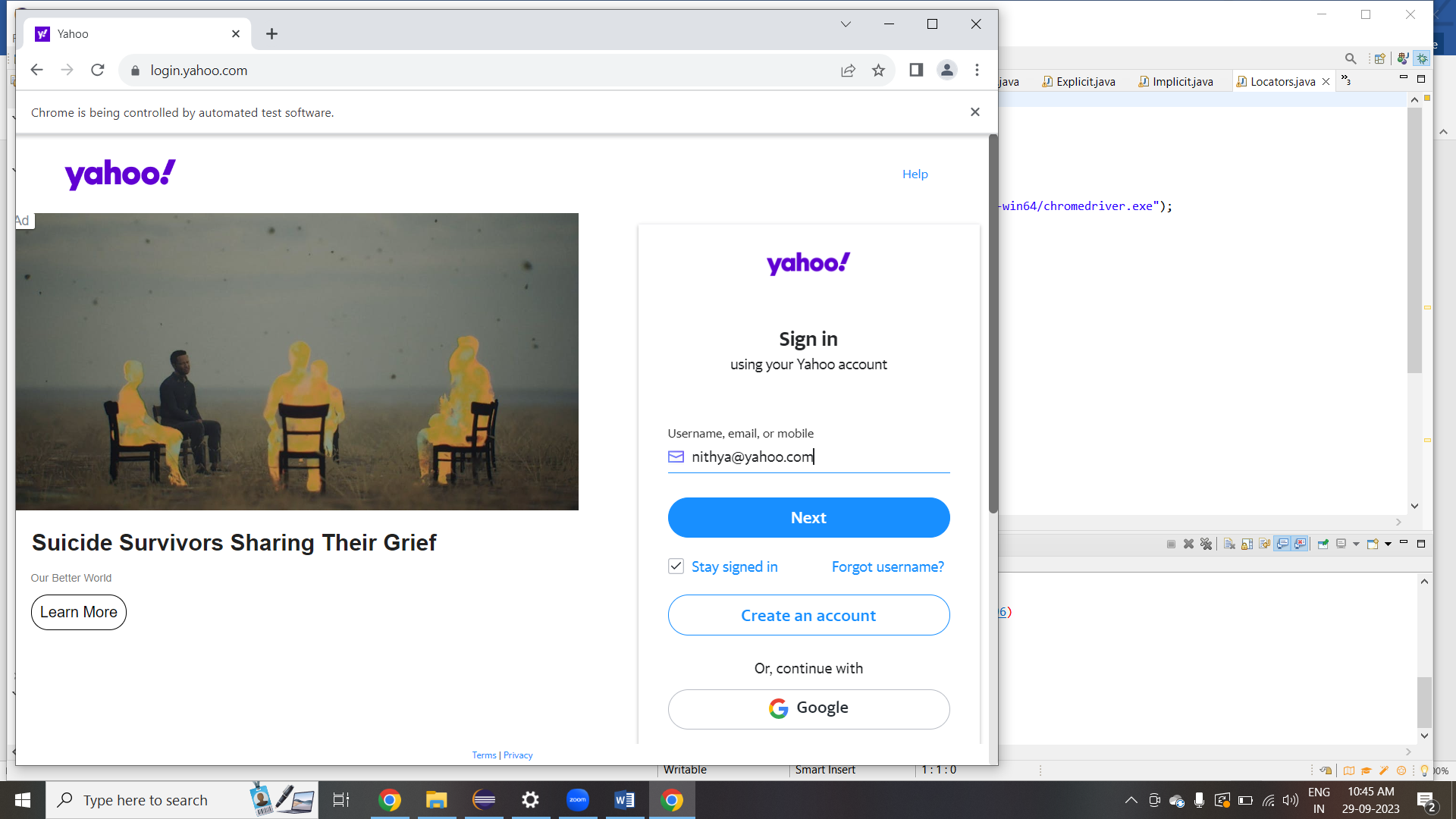
driver.findElement(By.xpath("//input[@id='q']")).sendKeys("Selenium"); //xpath for search box

WebElement searchIcon2 = driver.findElement(By.xpath("//input[@id='Google Search']"));//xpath for search button

driver.close();

}

}



7.navigate :

package selenium;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class Navigate {

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

//system property of chrome driver

System.setProperty("webdriver.chrome.driver","C:/Users/Admin/Downloads/chromedriver-win64/chromedriver-win64/chromedriver.exe");

//instantiate a chromedriver class.

WebDriver amaz = new ChromeDriver();

amaz.get("https://www.flipkart.com/");

System.out.println(amaz.getTitle());

amaz.navigate().to("https://www.amazon.in/");

System.out.println(amaz.getTitle());

amaz.navigate().back();

System.out.println(amaz.getTitle());

amaz.navigate().forward();

System.out.println(amaz.getTitle());

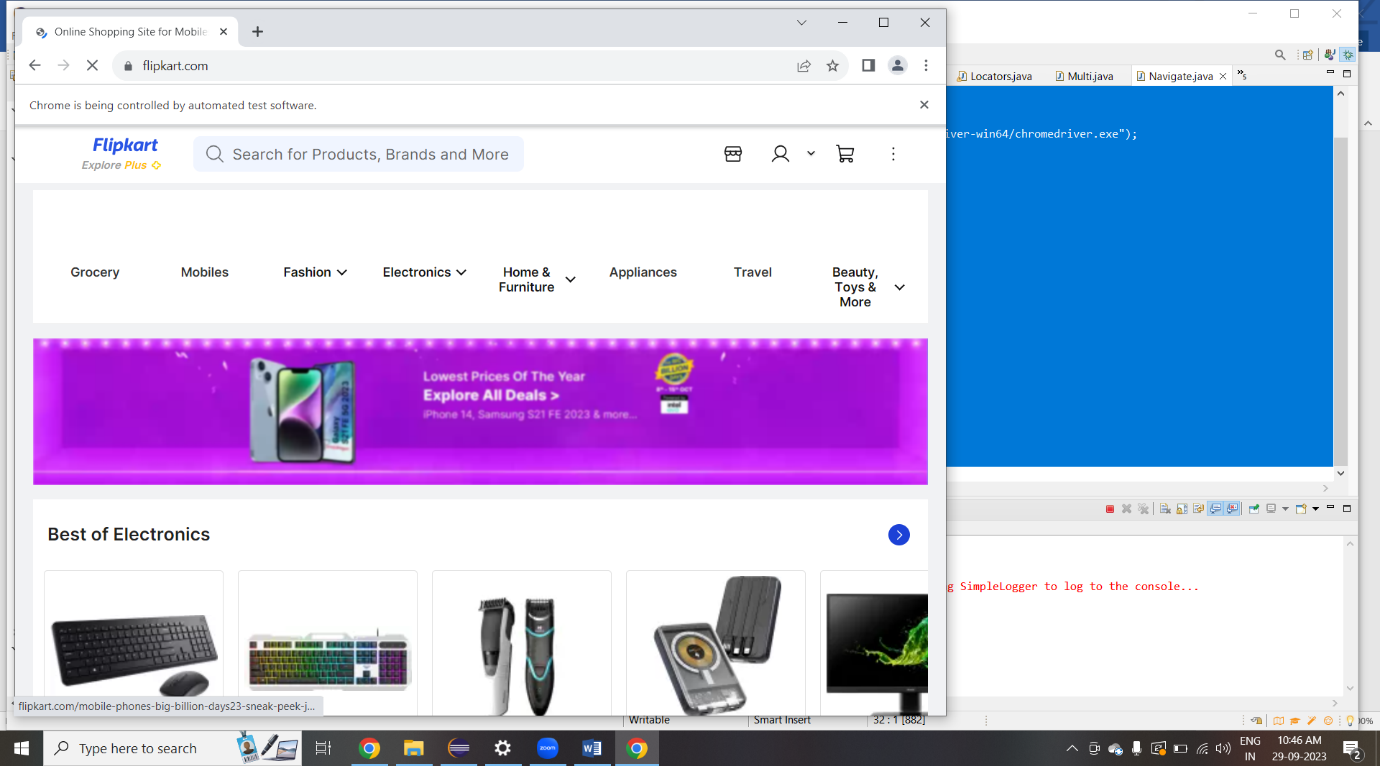
amaz.navigate().refresh();

System.out.println(amaz.getTitle());

amaz.close();

}

}



8. x path :

package selenium;

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;

public class X {

public static void main(String[] args) {

System.setProperty("webdriver.chrome.driver","C:/Users/Admin/Downloads/chromedriver-win64/chromedriver-win64/chromedriver.exe");

WebDriver driver = new ChromeDriver();

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

driver.manage().deleteAllCookies();

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

driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);

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

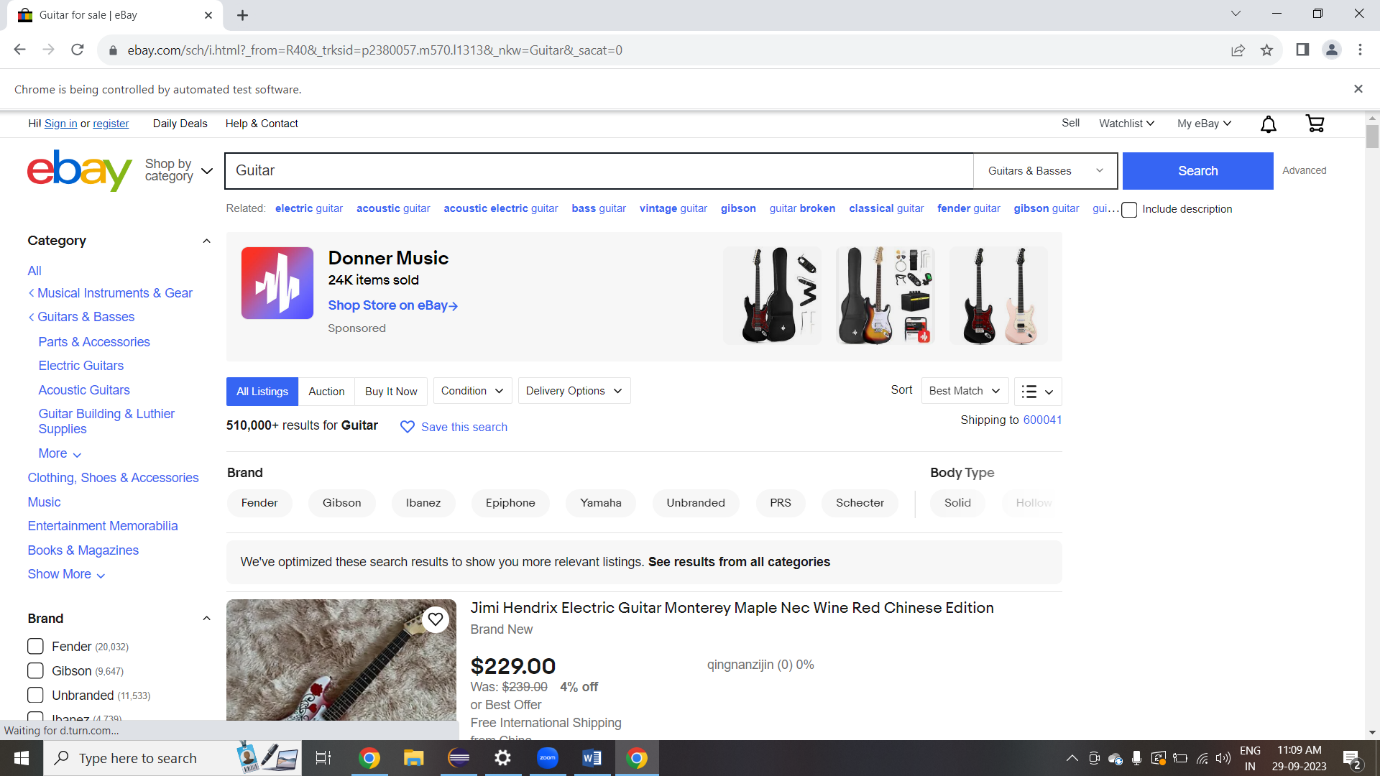
driver.findElement(By.xpath("//input[@id='gh-ac']")).sendKeys("Guitar"); //xpath for search box

WebElement searchIcon = driver.findElement(By.xpath("//input[@id='gh-btn']"));//xpath for search button

searchIcon.click();

}

}



9.radio button :

package selenium;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.\*;

public class Radio {

public static void main(String[] args) {

// declaration and instantiation of objects/variables

System.setProperty("webdriver.chrome.driver","C:/Users/Admin/Downloads/chromedriver-win64/chromedriver-win64/chromedriver.exe");

WebDriver driver = new ChromeDriver();

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

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

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

chkFBPersist.click ();

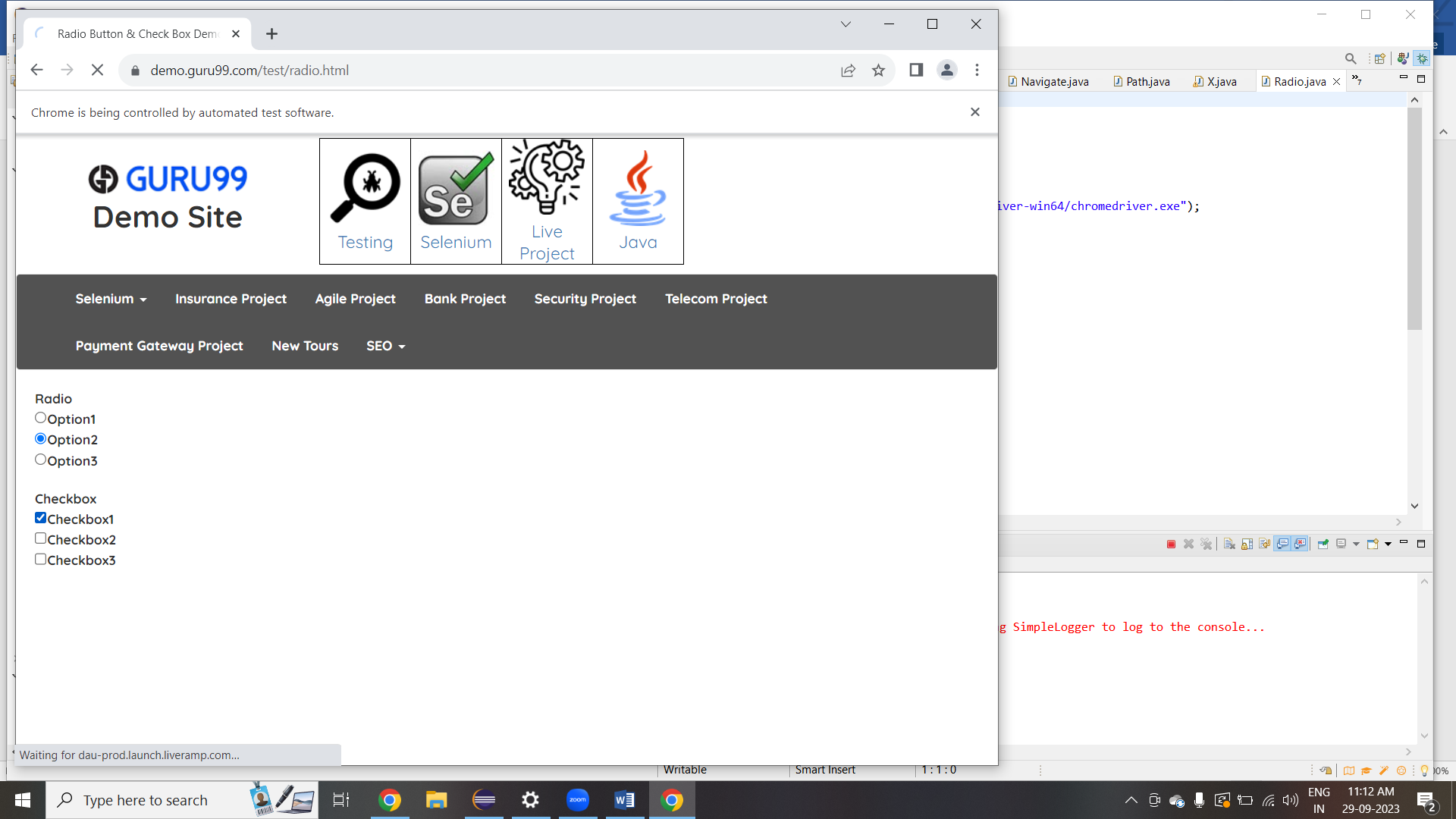
System.out.println("Facebook Persists Checkbox Status is - "+chkFBPersist.isSelected());

}

//driver.close();

}

}



10.screen shot

package selenium;

import org.apache.commons.io.FileUtils;

import java.io.File;

import java.io.IOException;

import org.openqa.selenium.OutputType;

import org.openqa.selenium.TakesScreenshot;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.chrome.ChromeOptions;

public class Screenshot {

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

// TODO Auto-generated method stub

System.setProperty("webdriver.chrome.driver","C:/Users/Admin/Downloads/chromedriver-win64/chromedriver-win64/chromedriver.exe");

ChromeOptions options = new ChromeOptions();

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

//creating obj for web driver

WebDriver driver=new ChromeDriver(options);

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

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

//creating reference is the first step

TakesScreenshot tk=(TakesScreenshot)driver;

//taking screenshot and storing it in the temporary file

File source=tk.getScreenshotAs(OutputType.FILE);

//creating destination file

File des=new File("D:/yahoo.png");

//moving the file from source to destination

FileUtils.copyFile(source, des);

}

}



11.Get command

package selenium;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

/\*Invoke Chrome Browser

Open URL: https://www.google.co.in/

Get Page Title name and Title length

Print Page Title and Title length on the Eclipse Console

Get page URL and verify whether it is the desired page or not

Get page Source and Page Source length

Print page Length on Eclipse Console.

Close the Browser\*/

public class Getcommand {

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

//system property of chrome driver

System.setProperty("webdriver.chrome.driver","C:/Users/Admin/Downloads/chromedriver-win64/chromedriver-win64/chromedriver.exe");

//instantiate a chromedriver class.

WebDriver hotstar=new ChromeDriver();

// launch the url

hotstar.get("https://www.hotstar.com/in/home?ref=%2Fin");// open the url in the browser.

String a= hotstar.getTitle();// getting tittle name inside browser

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

int alength=hotstar.getTitle().length();// getting title length.

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

String c=hotstar.getPageSource();// getting page source

System.out.println(c);

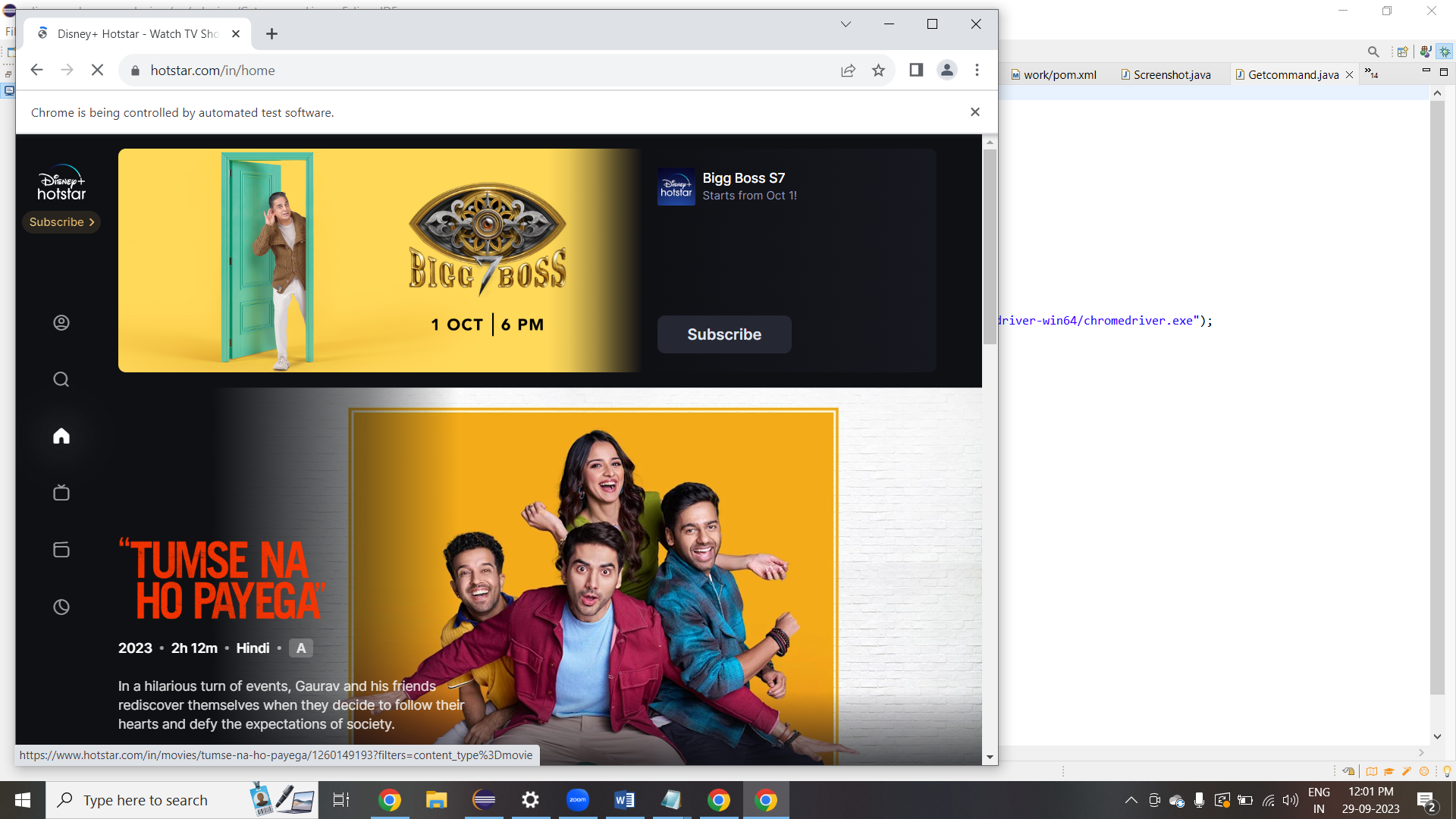
hotstar.close();// its terminate the browser window

hotstar.quit();//its terminate the all windows

Thread.sleep(6000);

}

}



## TEST-NG

**Definition**

* ***TestNG*** is a testing framework inspired

from ***JUnit*** and ***NUnit*** but introducing some new functionality that makes it more powerful and easier to use.

* It is an open-source automated testing framework; where ***NG*** of Test**NG** means **N**ext **G**eneration.

* TestNG is similar to JUnit but it is much more powerful than JUnit but still, it's inspired by JUnit.

* It is designed to be better than JUnit, especially when testing integrated classes. Pay special thanks to *Cedric Beust who is the creator of TestNG*.

### USES

TestNG eliminates most of the limitations of the older framework and gives the developer the ability to write more flexible and powerful tests with help of easy annotations, grouping, sequencing & parametrizing.

### What are the Benefits of TestNG:-

1. *It gives the ability to produce* ***HTML Reports*** *of execution*
2. ***Annotations*** *made testers life easy*
3. *Test cases can be* ***Grouped & Prioritized*** *more easily*
4. ***Parallel*** *testing is possible*
5. *Generates* ***Logs***
6. *Data* ***Parameterization*** *is possible*

### Test Case Writing process in TestNG

* + **Step 1** - Write the business logic of the test
  + **Step 2** - Insert TestNG annotations in the code
  + **Step 3** - Add the information about your test (e.g. the class names, methods names, groups names, etc...) in a testng.xml file
  + **Step 4** - Run TestNG

### What are the different Annotations are present in TestNG?

* + **@BeforeSuite**: The annotated method will be run before all tests in this suite have run.
  + **@AfterSuite**: The annotated method will be run after all tests in this suite have run.
  + **@BeforeTest**: The annotated method will be run before any test method belonging to the classes inside the tag is run.
  + **@AfterTest**: The annotated method will be run after all the test methods belonging to the classes inside the tag have run.
  + **@BeforeGroups**: The list of groups that this configuration method will run before. This method is guaranteed to run shortly before the first test method that belongs to any of these groups is invoked.
  + **@AfterGroups**: The list of groups that this configuration method will run after. This method is guaranteed to run shortly after the last test method that belongs to any of these groups is invoked.
  + **@BeforeClass**: The annotated method will be run before the first test method in the current class is invoked.
  + **@AfterClass**: The annotated method will be run after all the test methods in the current class have been run.
  + **@BeforeMethod**: The annotated method will be run before each test method.
  + **@AfterMethod**: The annotated method will be run after each test method.
  + **@Test**: The annotated method is a part of a test case.

PROGRAMS :

package testNG;

import java.time.Duration;

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;

import org.testng.Assert;

import org.testng.annotations.AfterClass;

import org.testng.annotations.BeforeClass;

import org.testng.annotations.Test;

public class gmail{

private WebDriver driver;

@BeforeClass

public void setUp() {

System.setProperty("webdriver.chrome.driver", "path to chromeDriver");

driver = new ChromeDriver();

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

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

// driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

}

@Test

public void testGmailLogin() {

WebElement usernameInput = driver.findElement(By.id("identifierId"));

usernameInput.sendKeys("sample@gmail.com");

WebElement nextButton = driver.findElement(By.id("identifierNext"));

nextButton.click();

WebDriverWait wait = new WebDriverWait (driver,Duration.ofSeconds(30));

WebElement passwordInput = wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//\*[@id =\"password\"]/div[1]/div / div[1]/input")));

passwordInput.sendKeys("Pass");

WebElement passwordNextButton = driver.findElement(By.id("passwordNext"));

passwordNextButton.click();

System.out.println("Login success");

}

@AfterClass

public void tearDown() {

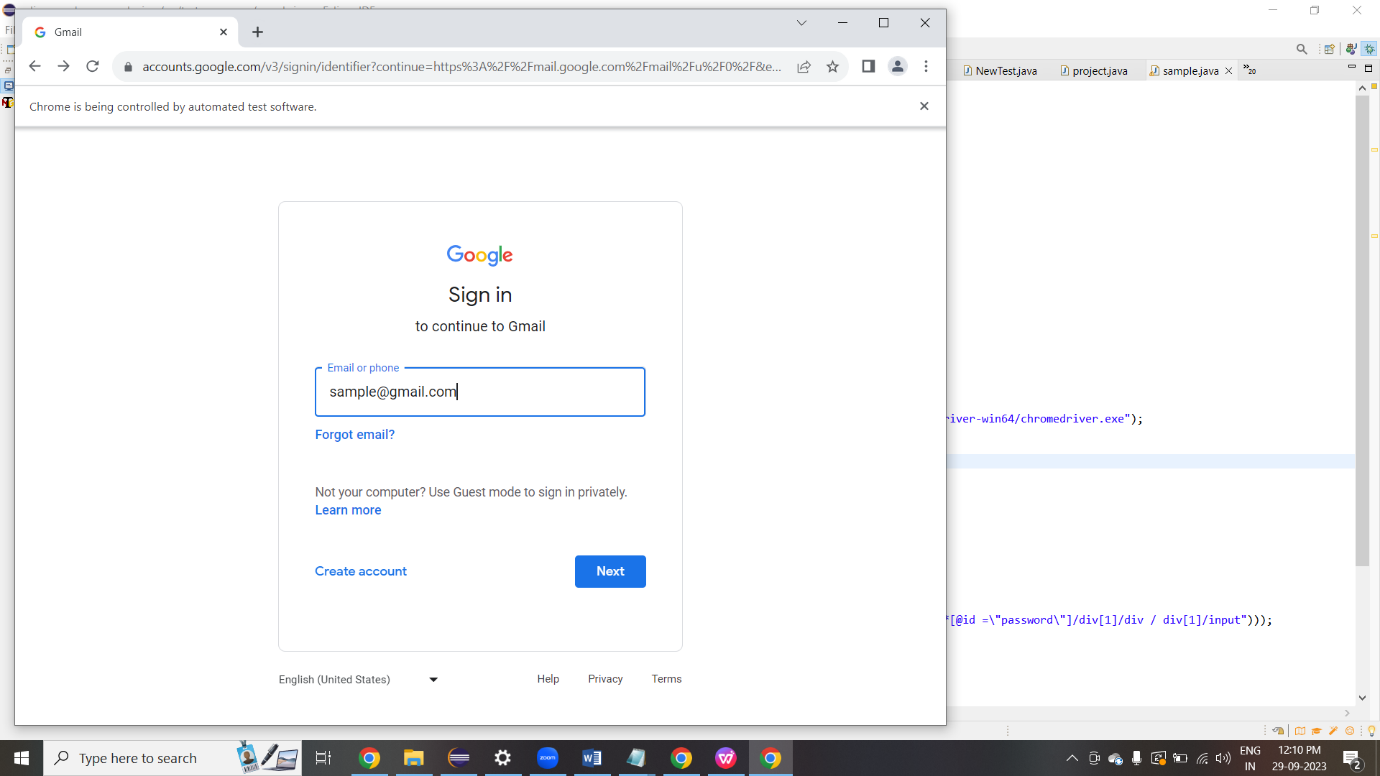
if (driver != null) {

driver.quit();

}

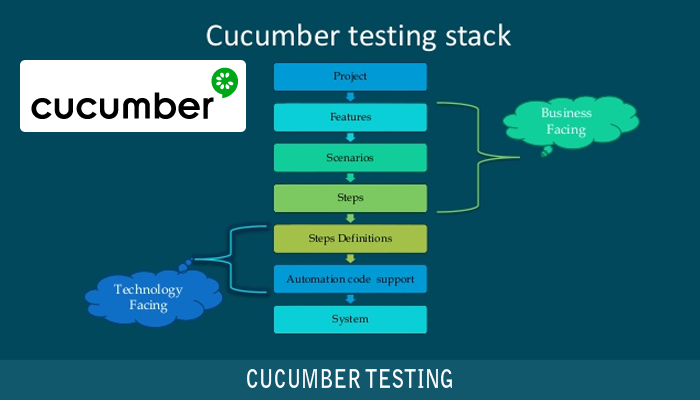
}

}



CUCUMBER

Cucumber framework uses Gherkin ( A simple plain text language parser) to describe expected software behaviors logically, resulting in better communication and collaboration among technical and non-technical team members. Cucumber is compatible with popular software platforms like Selenium, Watir, Ruby, and others.



Programs :

**Feature File**

Feature: youtube search functionality

Scenario: search news in youtube search box

Given User launch chrome browser

When User click to search youtube

Then User enter news in search box

**StepDefinition**

package stepdefinition;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import io.cucumber.java.en.Given;

import io.cucumber.java.en.Then;

import io.cucumber.java.en.When;

public class PrintFacebook {

public static WebDriver driver=null;

@Given("User launch chrome browser")

public void chromeLaunch() {

System.setProperty("webdriver.chrome.driver","C:/Users/Admin/Downloads/chromedriver-win64/chromedriver-win64/chromedriver.exe");

driver =new ChromeDriver();

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

}

@When("User click to search youtube")

public void searchButton() {

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

}

@Then("User enter news in search box")

public void searchButtons() {

if(driver.findElement(By.id("search")).isEnabled()) {

System.out.println("test pass");

}

else {

System.out.println("test fail");

}

//driver.findElement(By.id("search")).sendKeys("News");

//driver.findElement(By.id("search-icon-legacy")).click();

}

}

**RunnerClass**

package runnertest;

import org.junit.runner.RunWith;

/\*import cucumber.api.CucumberOptions;

import cucumber.api.junit.Cucumber;\*/

import io.cucumber.junit.Cucumber;

import io.cucumber.junit.CucumberOptions;

@RunWith(Cucumber.class)

@CucumberOptions(

features = {"src/main/java/feature/first.feature"},

publish = true,

glue = {"stepdefinition"}

)

public class TestRuner {

}

Output:

