**Phase 1 Assisted practice project**

Demonstrate the features of Selenium

Download Selenium IDE on chrome browser using the below link

<https://chromewebstore.google.com/detail/selenium-ide/mooikfkahbdckldjjndioackbalphokd?pli=1>

Features:

* It is not a complete tool, it is a small extension/plugin which can be downloaded for chrome and firefox browsers
* It is a record and playbook
* It records user actions of manual execution of test and we save that test
* Using selenium IDe, we can play the test case again and again.
* This is a very simple and easy tool
* Suitable for testers who are new to testing
* gives you an idea about what is automation testing
* We may never use this tool for automation testing on major webpages
* You don't write any code or scripts in IDE

Disadvantage:

* only tests on 2 browsers, we cannot do testing on other browsers
* Test data is fixed, if new data has to be passed, you have to enter manually
* IDE does not support reading of data from excel sheets-> Data driven testing no possible
* No wait time in IDE
* Parametrization cannot be handled in IDE
* No reports are generated
* Does not support much with dynamic page testing
* No way to prioritize test cases
* No test scheduling is possible
* You cannot set up any pre-condition or post condition.

1. Selenium WebDriver → Write selenium scripts using Java

=======================================================

Features:

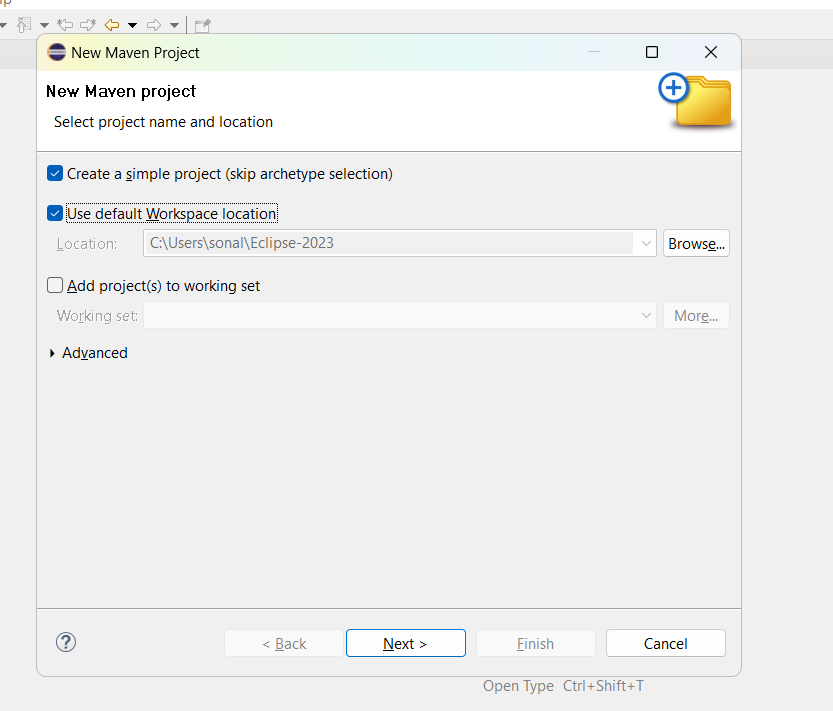
* Selenium webdriver is the main component of Selenium tools
* This tool has no front end, we will install it in our Eclipse
* It is a java library that has to be added in the eclipse java project
  + Add it manually
  + Add it automatically by using Maven project
* Using selenium webdriver, we will write selenium automation scripts
* You write the script once and same works on all browsers
* All the complex automation testing will be done using webdriver component only
* Support testing on all browsers and all OS
* Webdriver can be integrated with other frameworks like testNG, cucumber, apache poi
* Data driven testing, parametrization of scripts
* Prioritization, pre condition & post condition
* Generate test reports

Setting up of Selenium WebDriver:

==============================

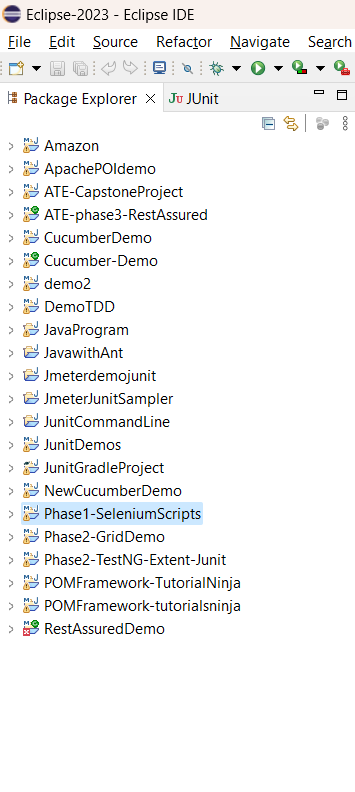
> Go to Eclipse → Go to file→ New project → Maven project → Select checkbox-  Create a simple project -> pls make sure your workspace location is your default workspace location

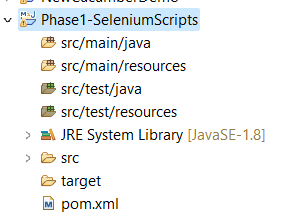
If it is not default → uncheck the second checkbox→ from the drop down select the workspace



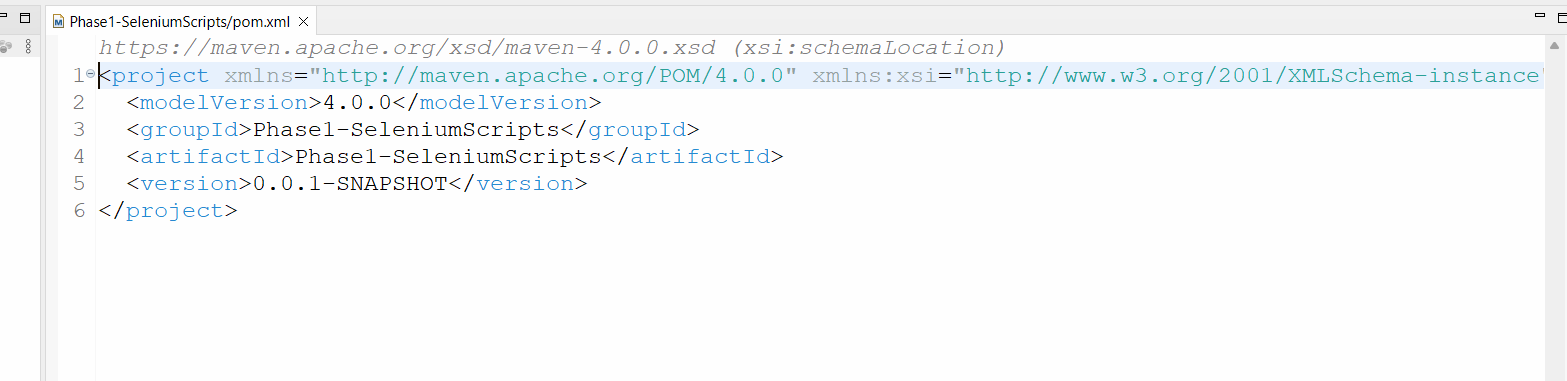
Press Next → enter group id : Phase1-SeleniumScripts → enter artifact id = Phase1-SeleniumScripts -> click on Finish button

You will see your project on the left side:





Double click on POM.xml file and open it



In the Pom file add below dependencies after line number 5 or before the tag </project>

Save the file after adding the dependencies.

**<dependencies>**

**<dependency>**

**<groupId>org.seleniumhq.selenium</groupId>**

**<artifactId>selenium-java</artifactId>**

**<version>4.11.0</version>**

**</dependency>**

**<!-- https://mvnrepository.com/artifact/io.github.bonigarcia/webdrivermanager -->**

**<dependency>**

**<groupId>io.github.bonigarcia</groupId>**

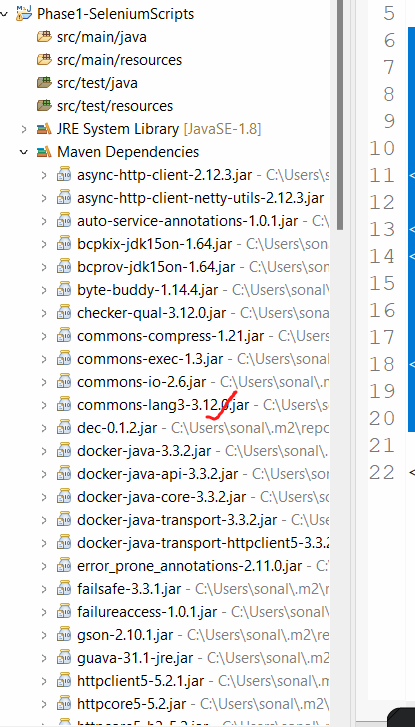
**<artifactId>webdrivermanager</artifactId>**

**<version>5.4.1</version>**

**</dependency>**

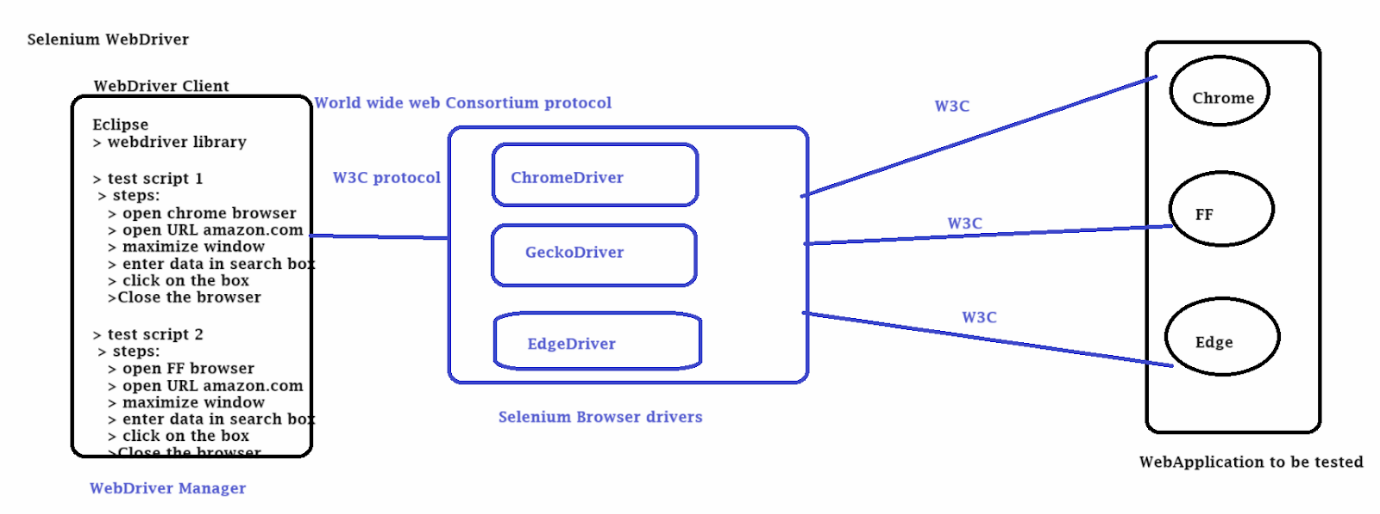
**</dependencies>**





Your setup for selenium is now complete.

Selenium Architecture:

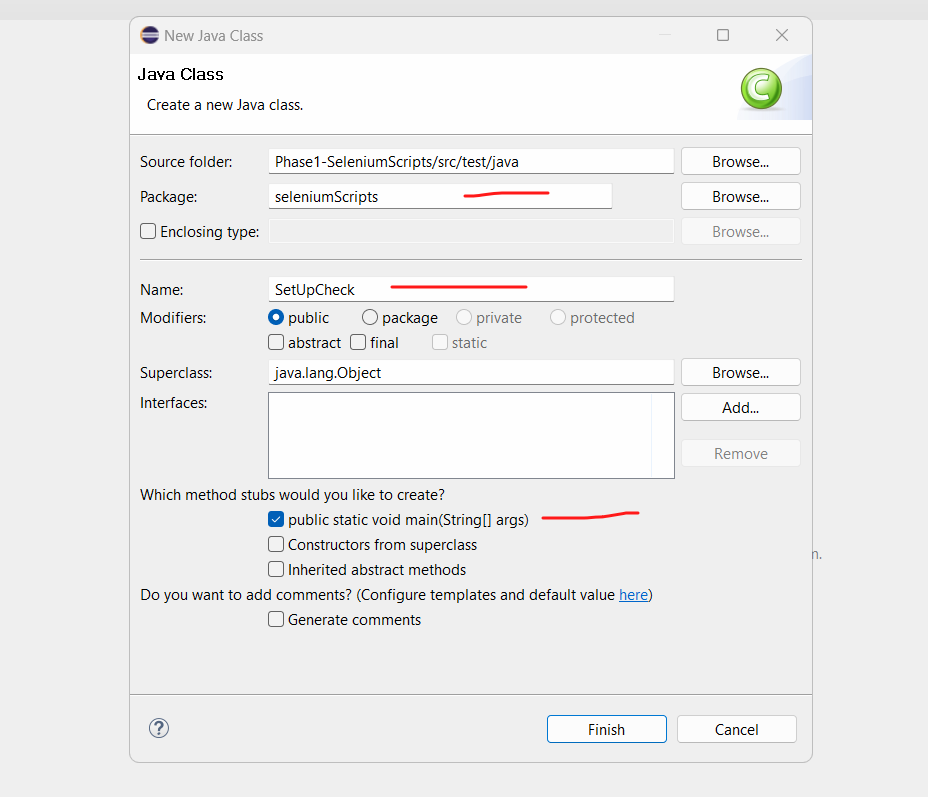


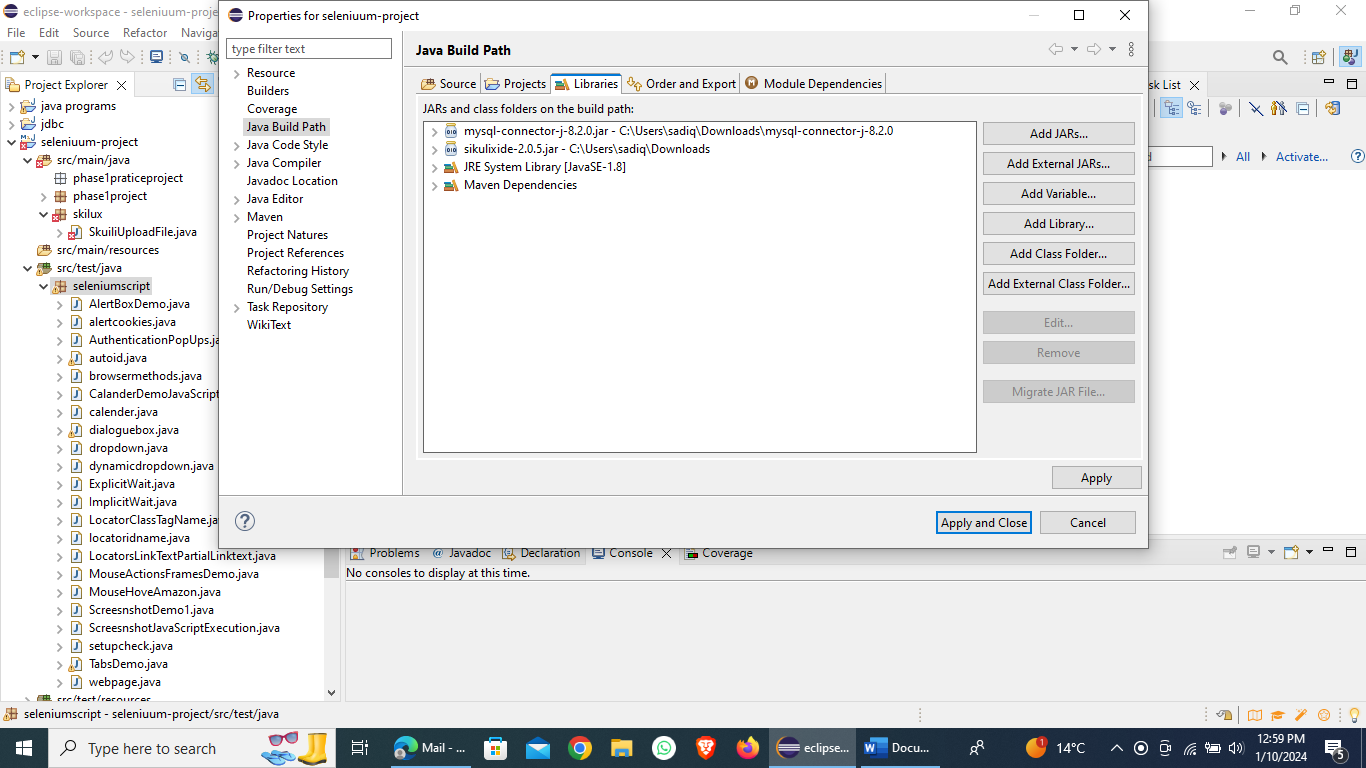
Demo 1:

======================

Let's check the setup on eclipse to check if selenium webdriver is able to create a session with your machine browser.

Create a Java class in the src/test/java folder



****

**package seleniumscript;**

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

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

**public class setupcheck {**

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

**// TODO Auto-generated method stub**

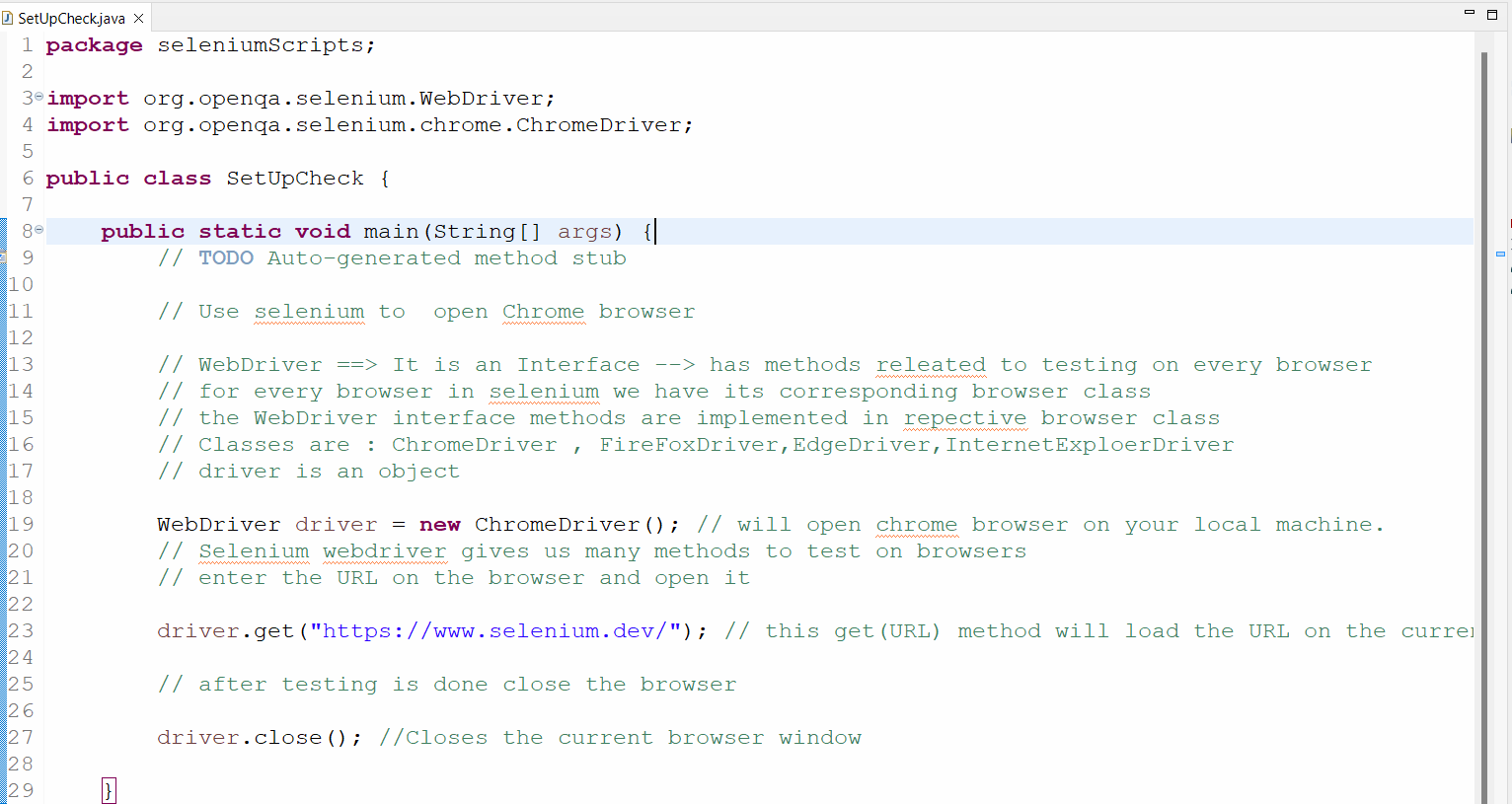
**WebDriver driver = new ChromeDriver();**

**driver.get("https://www.selenium.dev/");**

**driver.close();**

**}**

**}**



**Webpage:-**

**package seleniumscript;**

**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 webpage {**

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

**// TODO Auto-generated method stub**

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

**driver.get("https://www.ironspider.ca/forms/dropdowns.htm");**

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

**// write xpath for the main drop down not of the dropdown option**

**// to handle drop down, selenium gives you a class Select class and method**

**// Select dd = new Select(dropdownelement)));**

**Select dd = new Select(driver.findElement(By.xpath("//select[@name='coffee']")));**

**// selectByIndex(),selectByVisibleText(),selectByValue() methods to select a dropdown option**

**dd.selectByIndex(1); // select with cream**

**Thread.sleep(1500);**

**dd.selectByValue("sugar");**

**Thread.sleep(1500);**

**dd.selectByVisibleText("Crisp (har har...)");**

**// Get all the options from the dropdown and store it in a list object. Print it on console**

**List<WebElement> li = dd.getOptions(); // this method returns a list of elements in the dropdown**

**System.out.println("The size of the list" + li.size());**

**for(WebElement e:li)**

**{**

**System.out.println(e.getText()); // get the each element in the list and print it.**

**}**

**}**

**}**

**package** seleniumScripts;

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

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

**public** **class** BrowserMethods {

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

// **TODO** Auto-generated method stub

// press CTL+Shift+O => to import the packages automatically

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

driver.get("https://www.selenium.dev/");

// maxamize the browser window

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

// delete browser cookies

driver.manage().deleteAllCookies();

// Fetch the title of the page

String title = driver.getTitle();

System.***out***.println("The title of page-1 is : " + title);

// Fetch the URL of the webpage

String url = driver.getCurrentUrl();

System.***out***.println("The URL of page is : " + url);

// Methods to naviagate form one webpage to another webpage

driver.navigate().to("https://www.selenium.dev/downloads/") ;

    String title1 = driver.getTitle();

System.***out***.println("The title of webpage-2 is : " + title1);

// Again naviage to  webpage 3

driver.navigate().to("https://www.selenium.dev/selenium/docs/api/java/index.html") ;

        String title2 = driver.getTitle();

System.***out***.println("The title of webpage-3 is : " + title2);

// Navigation methods : to go forward and backword on browser window

driver.navigate().back();

  String title3 = driver.getTitle();

System.***out***.println("The title of webpage is : " + title3);

driver.navigate().forward();

  String title4 = driver.getTitle();

System.***out***.println("The title of webpage is : " + title4);

driver.navigate().refresh();

driver.close();

}

}

=========================================================

**WebElements:**

====================================================

A webPage includes HTML objects which we can call as WebElements.

A web page is constructed using HTML

These HTML objects are:

==========================

> Text boxes

> Button/Submit button

> CheckBox

> Radio button

> Drop Down

> Date Picker

> Tables

> Links

> Images

As an automation tester you will have to use selenium to interact(action) with these elements

Selenium client library gives us  simple method to locate the element on the webpage and perform an action on it

**User → selenium → findElement(which Element) on webpage → perform an Action(what action)**

Now selenium gives us methods →

**findElement(address of the element) or findElements(address of the element)**

Selenium gives us some action method -**>**

**isDisplayed() -> return true if the element to be tested is visible on the webpage**

**isEnabled() -> return true if the element to be tested is enabled on the webpage**

**isSelected() -> return true if the element to be tested is selected on the webpage**

**sendKeys(“input”)  -> helps us to enter input in a text box**

**click() → helps us to click on button or link**

**getText() → helps us to get the text from webpage**

**getattribute() → helps us to get the attribute value from the webpage**

Demonstrate how elements are located through CSS and XPath

Xpath:-

Whenever the webelements tags don't have any attribute, we can still write Xpath locator for it.

You can write using xpath using xpath axes in which we will take  a tag which has attributes and then from that tag we will traverse upwards to downwards to reach our desired element Tag

<html>

<body>

   <td name=city1>

         <div>element1 </div>

          <div>element2</div>

    </td>

     <td name=country>

         <div>element3</div>

          <div>element4</div>

    </td>

</body>

<html>

Self: xpath will return the current

Syntax of Axes:

**//tagname[@attribute=’value’]/axesname::\***

**OR**

**//tagname[@attribute=’value’]/axesname::tagname[indexnumber]**

Child axes: Xpath will return all children of your current tag

You can then choose the required element tag and give the index number

//div[@class='hmmtctable']/child::ul[3]

|  |  |
| --- | --- |
| descendant | will return all the children and grandchildren of the current node |

Example:

//div[@id='div\_bseindices']/descendant::li[3]

|  |  |
| --- | --- |
| following | will return all the nodes that apppear after the current node has closed |

Example: //div[@id='div\_nseindices']/following::li[4]

Demonstrate how to automate calendars on the web page

**package seleniumScripts;**

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

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

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

**public class CalanderDemo1 {**

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

**// TODO Auto-generated method stub**

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

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

**driver.get("https://seleniumpractise.blogspot.com/2016/08/how-to-handle-calendar-in-selenium.html");**

**// Click on the text box field , so that the calender is displayed.**

**driver.findElement(By.*xpath*("//div[@class='date-posts']/descendant::input")).click();**

**Thread.*sleep*(1500);**

**// On the calander lets inpsect the month and year tab and print the month and year text**

**String monthyear = driver.findElement(By.*xpath*("//div[@class='ui-datepicker-title']")).getText();**

**System.*out*.println(monthyear);  // January 2024**

**// Split the String monthyear in to 2 new strings**

**// Create a new array object that will store the string monath and year**

**String my[] = monthyear.split(" "); // split based on the space**

**String month = my[0]; //January**

**String year = my[1]; //2024**

**// write a loop**

**while(!(month.equals("August") && year.equals("2025")))**

**{**

**//click on the next button in the calender**

**driver.findElement(By.*xpath*("//span[text()='Next']")).click();**

**// come out of the loop**

**monthyear = driver.findElement(By.*xpath*("//div[@class='ui-datepicker-title']")).getText();**

**month = monthyear.split(" ")[0];**

**year = monthyear.split(" ")[1];**

**}**

**// select the date : //a[text()='8']**

**//table[@class='ui-datepicker-calendar']/descendant::a[8]**

**driver.findElement(By.*xpath*("//table[@class='ui-datepicker-calendar']/descendant::a[8]")).click();**

**}**

**}**

Using Selenium WebDriver, write a program to handle alert

package seleniumscript;

import org.openqa.selenium.Alert;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class AlertBoxDemo {

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

// TODO Auto-generated method stub

WebDriver driver = new ChromeDriver();

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

driver.manage().deleteAllCookies();

driver.get("https://mail.rediff.com/cgi-bin/login.cgi");

// just click on Sign in button without enter username and password. An alert box will be displayed

driver.findElement(By.xpath("//input[@type='submit']")).click();

// to perform actions on alertbox, switch to it

Thread.sleep(2000);

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

// get the text from alert

String text = a.getText();

System.out.println(text);

// press ok button

a.accept();

}

}

Demonstrate how screenshots are captured and browser profiles are changed in Selenium.

package seleniumscript;

import java.io.File;

import java.io.IOException;

import org.apache.commons.io.FileUtils;

import org.openqa.selenium.OutputType;

import org.openqa.selenium.TakesScreenshot;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class ScreesnshotDemo1 {

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

// TODO Auto-generated method stub

WebDriver driver = new ChromeDriver();

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

driver.manage().deleteAllCookies();

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

// take screeshot of current window and save it in a file

// Use class TakesScreenshot and method -> getScreenshoAs

File srcFile = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);

// Screesnhot is saved in the object srcFile

// In the current project --> create a folder Screenshot--> create a file with name opera1.png

File destFile = new File("./Screenshot/opera1.png");

FileUtils.copyFile(srcFile, destFile);

/\*Copies a file to a new location preserving the file date.

This method copies the contents of the specified source file

to the specified destination file. The directoryholding the destination file

is created if it does not exist. If the destination file exists,

then this methodwill overwrite it \*/

}

}

Demonstrate installation and configuration of AutoIT.

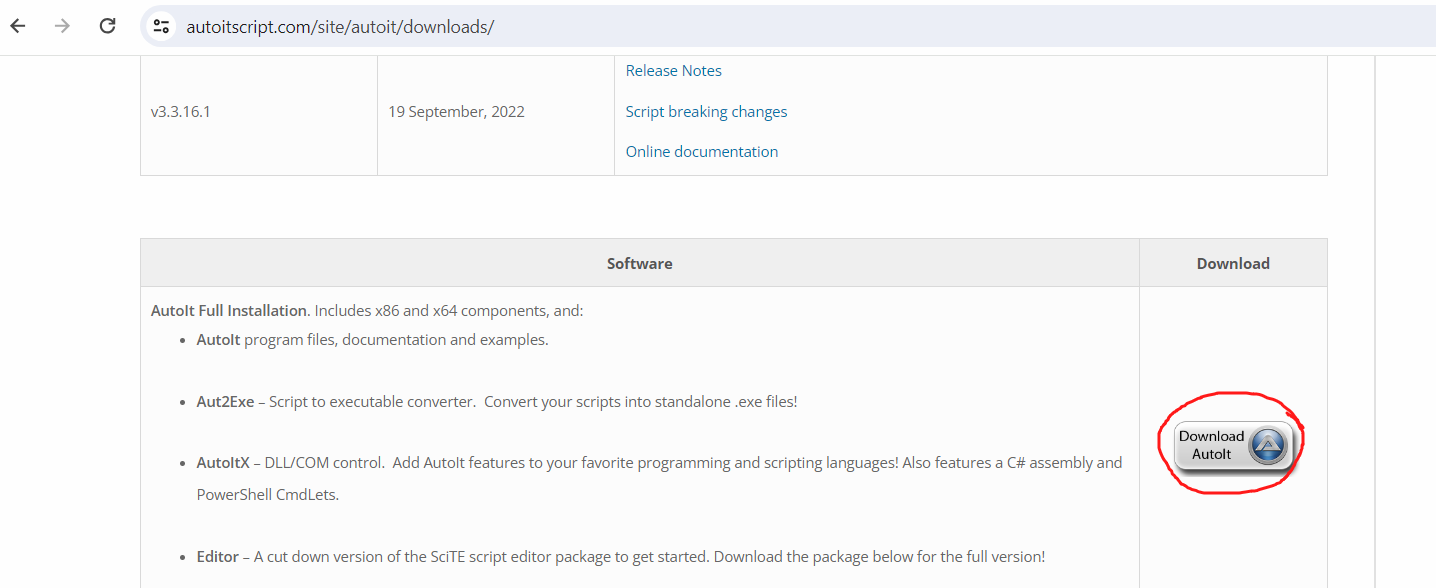
AutoIt tool

=====================================

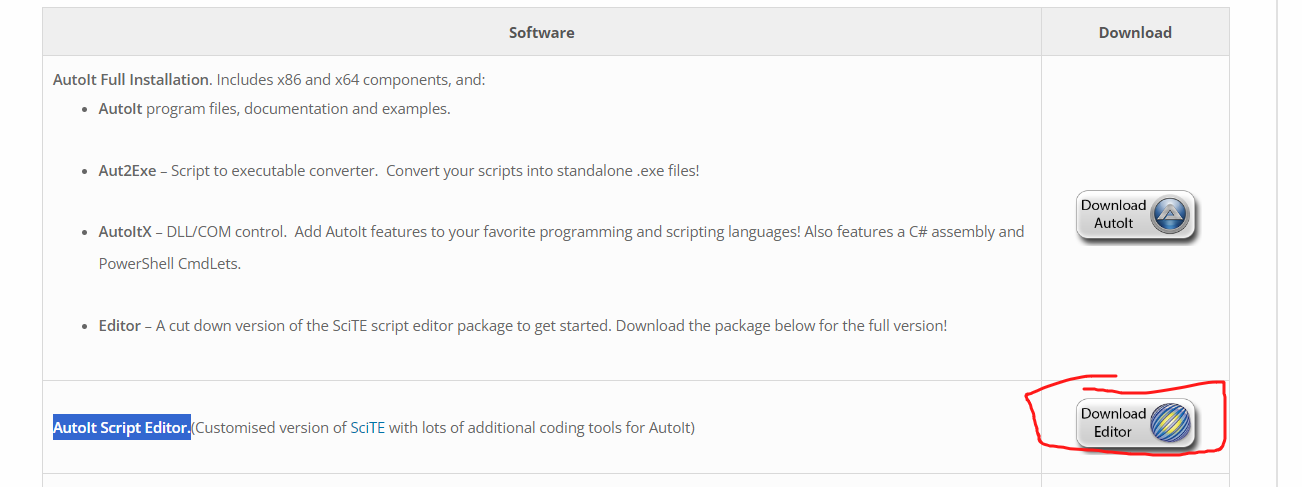
1. Download AutoIT

<https://www.autoitscript.com/site/autoit/downloads/>

Scroll down and click on button as show below



Also download AutoIt Script Editor.



Step 3: Decide the file to be uploaded

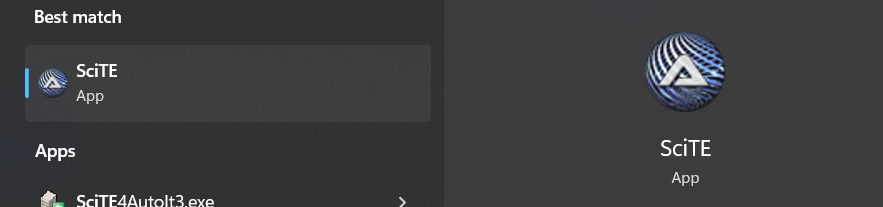
=============

For me it is :  C:\Users\sonal\Documents\ATE-Phase1-SL\JDBCconnectivity.png

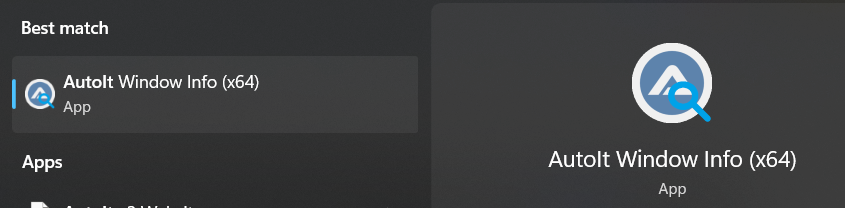
Step 4:

Now we will use AutoIT tool to write script that will first locate the file box and the open button

On the windows machine → open the autoIT editor -> search for app ->Scite



Now open another auto It tool to inspect the file box

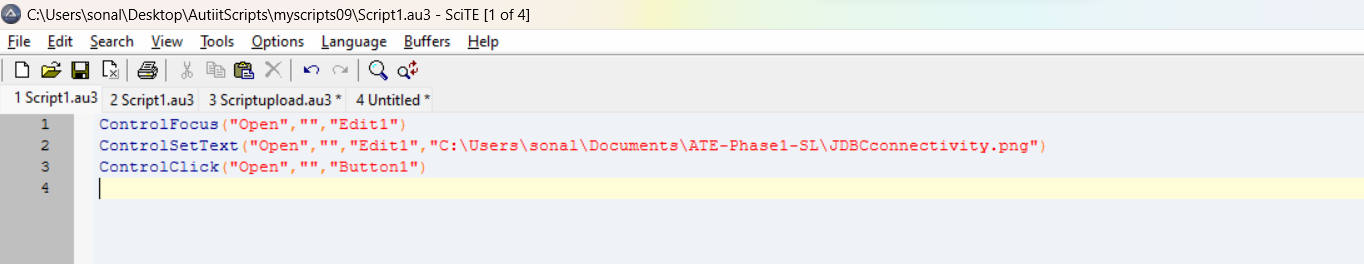


Write the script on the editor

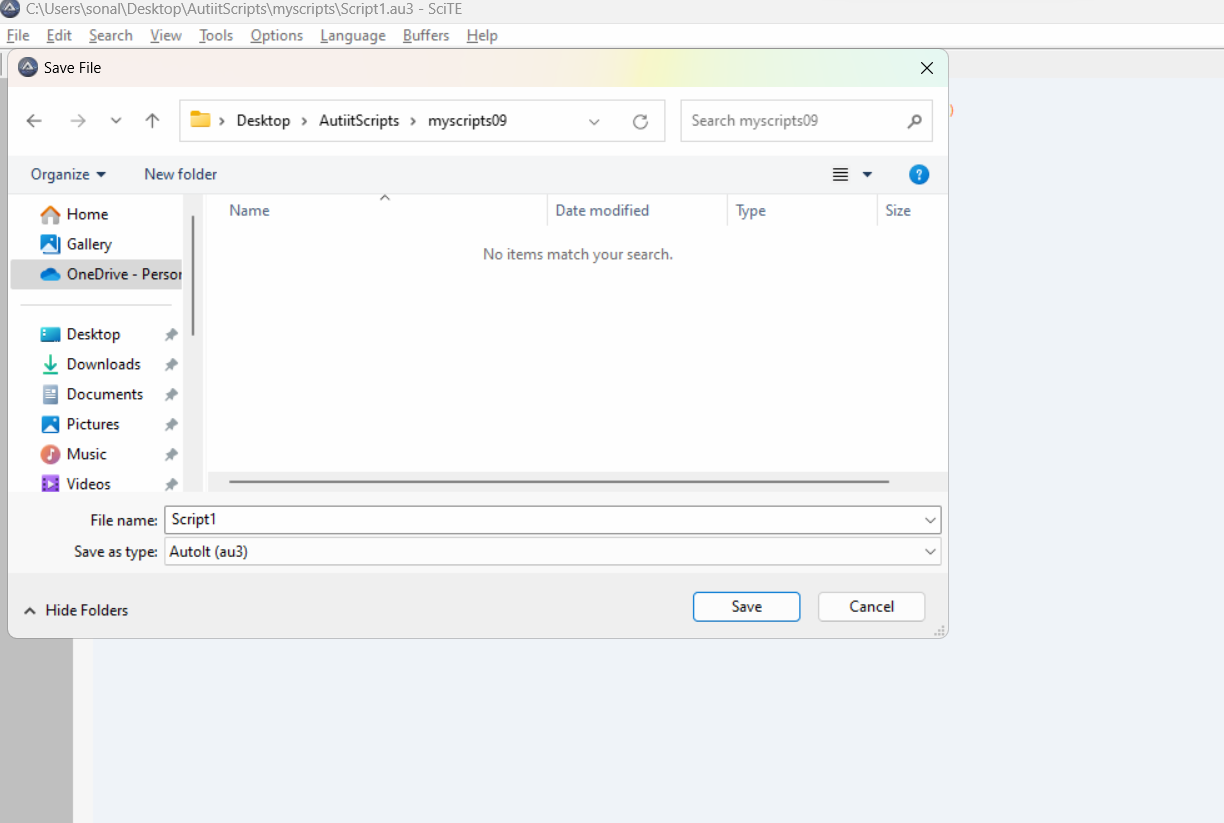
ControlFocus("Open","","Edit1")

ControlSetText("Open","","C:\Users\sonal\Documents\ATE-Phase1-SL\JDBCconnectivity.png")

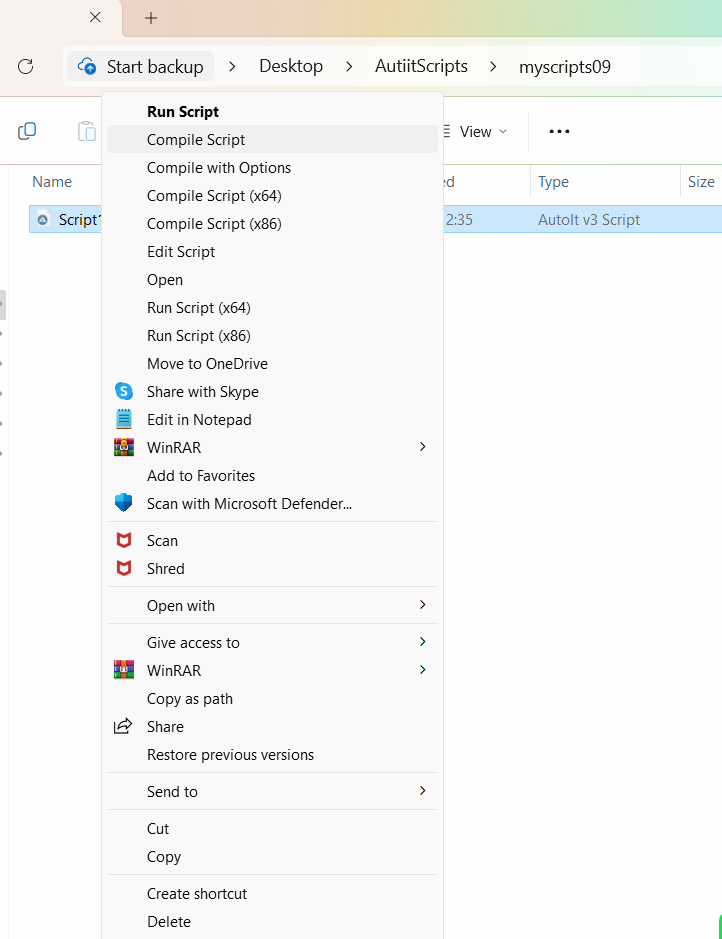
ControlClick("Open","","Button1")



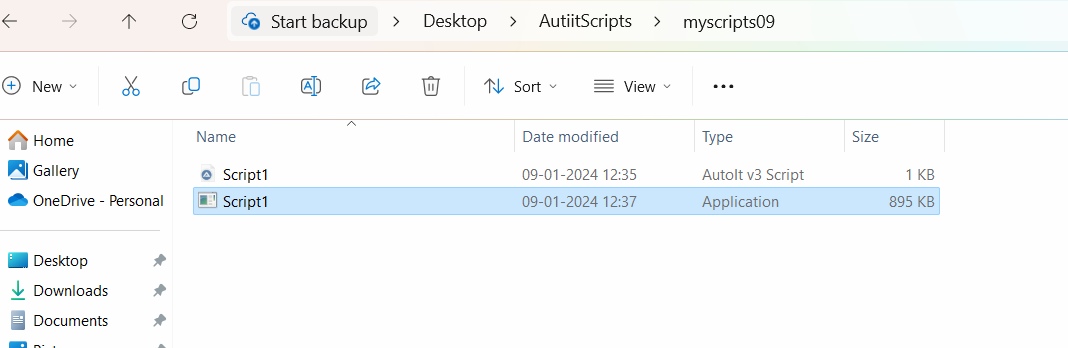
Save the script



Open the folder where the script is → right click on the script —> show more options and compile it



The compiled script will be in the folder:



**package** seleniumScripts;

**import** java.io.IOException;

**import** java.time.Duration;

**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** AutoItdemo {

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

// **TODO** Auto-generated method stub

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

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

       driver.manage().deleteAllCookies();

driver.get("https://www.remove.bg/");

WebDriverWait wait = **new** WebDriverWait(driver,Duration.*ofSeconds*(10));

// wait until the given condition is satisfied

wait.until(ExpectedConditions.*visibilityOfElementLocated*(By.*xpath*("//div[@class='mx-auto w-full px-8 max-w-5xl relative']/descendant::button[1]")));

WebElement e1 = driver.findElement(By.*xpath*("//div[@class='mx-auto w-full px-8 max-w-5xl relative']/descendant::button[1]"));

e1.click();

// selenium to run the autoID compiled script

Runtime.*getRuntime*().exec("C:\\Users\\sonal\\Desktop\\AutiitScripts\\myscripts09\\Script1.exe");

}

}

Demonstrate how file uploads are handled in AutoIT.

package seleniumscript;

import java.io.IOException;

import java.time.Duration;

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 autoid {

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

// TODO Auto-generated method stub

WebDriver driver = new ChromeDriver();

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

driver.manage().deleteAllCookies();

driver.get("https://www.remove.bg/");

WebDriverWait wait = new WebDriverWait(driver,Duration.ofSeconds(10));

// wait until the given condition is satisfied

wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//div[@class='mx-auto w-full px-8 max-w-5xl relative']/descendant::button[1]")));

WebElement e1 = driver.findElement(By.xpath("//div[@class='mx-auto w-full px-8 max-w-5xl relative']/descendant::button[1]"));

e1.click();

// selenium to run the autoID compiled script

Runtime.getRuntime().exec("C:\\Users\\sadiq\\Downloads\\script\\Script1.exe");

}

}

Demonstrate how Sikuli is used for UI testing in Selenium.

package skilux;

import java.time.Duration;

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 com.ibm.icu.impl.TimeZoneGenericNames.Pattern;

import com.tigervnc.rfb.Screen;

public class SkuiliUploadFile {

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

// TODO Auto-generated method stub

WebDriver driver = new ChromeDriver();

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

driver.manage().deleteAllCookies();

driver.get("https://www.remove.bg/");

Thread.sleep(4000);

WebDriverWait wait = new WebDriverWait(driver,Duration.ofSeconds(10));

// wait until the given condition is satisfied

wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//div[@class='mx-auto w-full px-8 max-w-5xl relative']/descendant::button[1]")));

WebElement e1 = driver.findElement(By.xpath("//div[@class='mx-auto w-full px-8 max-w-5xl relative']/descendant::button[1]"));

e1.click();

Screen s = new Screen();

Pattern textfield = new Pattern("C:\\sikuli\\image10.png");

Pattern openButton = new Pattern("C:\\sikuli\\image101.png");

s.wait(20);

s.type(textfield,"C:\\Users\\sonal\\Documents\\ATE-Phase1-SL\\sikulidemo");

s.click(openButton);

}

}

package phase1project;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.SQLException;

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.Actions;

public class SeleniumJDBCTrasaction {

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

// TODO Auto-generated method stub

WebDriver driver = new ChromeDriver();

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

driver.manage().deleteAllCookies();

driver.get("https://www.amazon.in/");

String title = driver.getTitle();

Thread.sleep(2000);

WebElement mobilelink = driver.findElement(By.linkText("Mobiles"));

mobilelink.click();

WebElement mobile = driver.findElement(By.xpath("//div[@id='nav-progressive-subnav']/descendant::span[2]"));

Actions a = new Actions(driver);

a.moveToElement(mobile).perform();

Thread.sleep(2000);

WebElement apple\_link = driver.findElement(By.linkText("Apple"));

String text = apple\_link.getText();

apple\_link.click();

String dburl = "jdbc:mysql://localhost:3306/seleniumtest";

String username = "root";

String password = "root";

Class.forName("com.mysql.cj.jdbc.Driver");

Connection con = DriverManager.getConnection(dburl, username, password);

PreparedStatement ps = con.prepareStatement("insert into endproject values(?,?,?)");

ps.setString(1, title);

ps.setString(2, text);

ps.setString(3, "Yes");

ps.executeUpdate();

}

}