**Automation.java**

**package** ecommerce;

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

**import** org.openqa.selenium.JavascriptExecutor;

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

**import** org.openqa.selenium.WebElement;

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

**import** org.openqa.selenium.firefox.FirefoxDriver;

**import** org.testng.annotations.Test;

**public** **class** Automation {

WebDriver driver;

@Test(groups = "Chrome")

**public** **void** LaunchChrome() {

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

"C:\\javafsd\\chromedriver\_win32/chromedriver.exe");

**try** {

Thread.*sleep*(2000);

} **catch** (Exception e) {

e.printStackTrace();

}

}

@Test(groups = "Chrome", dependsOnMethods = "LaunchChrome")

**public** **void** chrome() {

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

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

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

System.***out***.println("Flipkart Page is Loaded..");

driver.findElement(By.*xpath*("//button[contains(.,'✕')]")).click();

System.***out***.println("Closed the sign up activity..");

// To find the load time and print on console

Long loadtime = (Long) ((JavascriptExecutor) driver)

.executeScript("return performance.timing.loadEventEnd - performance.timing.navigationStart;");

System.***out***.println("Load time of the website is :- " + loadtime);

// To search the given text in search bar

driver.findElement(By.*className*("\_3704LK")).sendKeys("iphone 13");

driver.findElement(By.*className*("L0Z3Pu")).click();

driver.navigate().refresh();

WebElement i = driver.findElement(By.*xpath*("//img[@class='\_396cs4 \_3exPp9']"));

Boolean p = (Boolean) ((JavascriptExecutor) driver).executeScript("return arguments[0].complete "

+ "&& typeof arguments[0].naturalWidth != \"undefined\" " + "&& arguments[0].naturalWidth > 0", i);

**if** (p) {

System.***out***.println("Image is loaded");

} **else** {

System.***out***.println("Image not loaded");

}

System.***out***.println("The title of the page is :" + driver.getTitle());

**try** {

**long** lastHeight = (**long**) ((JavascriptExecutor) driver).executeScript("return document.body.scrollHeight");

**while** (**true**) {

((JavascriptExecutor) driver).executeScript("window.scrollTo(0, document.body.scrollHeight);");

Thread.*sleep*(2000);

**long** newHeight = (**long**) ((JavascriptExecutor) driver)

.executeScript("return document.body.scrollHeight");

**if** (newHeight == lastHeight) {

**break**;

}

lastHeight = newHeight;

}

**if** (lastHeight > 0)

System.***out***.println("Scrolled & Height of Page : " + lastHeight);

**else**

System.***out***.println("Not Scrolled");

} **catch** (InterruptedException e) {

e.printStackTrace();

}

}

@Test(groups = "Firefox")

**public** **void** LaunchFirefox() {

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

"C:\\javafsd\\geckodriver\_win32//geckodriver.exe");

**try** {

Thread.*sleep*(4000);

} **catch** (Exception e) {

e.printStackTrace();

}

}

@Test(groups="Firefox", dependsOnMethods="LaunchFirefox")

**public** **void** firefox() {

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

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

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

System.***out***.println("Flipkart Page is Loaded..");

driver.findElement(By.*xpath*("//button[contains(.,'✕')]")).click();

System.***out***.println("Closed th singup activity..");

// To find the load time and print on console

Long loadtime = (Long) ((JavascriptExecutor) driver)

.executeScript("return performance.timing.loadEventEnd - performance.timing.navigationStart;");

System.***out***.println("Load time of the website is :- " + loadtime);

// To search the given text in search bar

driver.findElement(By.*className*("\_3704LK")).sendKeys("iphone 13");

driver.findElement(By.*className*("L0Z3Pu")).click();

WebElement i = driver.findElement(By.*xpath*("//img[@class='\_396cs4 \_3exPp9']"));

Boolean p = (Boolean) ((JavascriptExecutor) driver).executeScript("return arguments[0].complete "

+ "&& typeof arguments[0].naturalWidth != \"undefined\" " + "&& arguments[0].naturalWidth > 0", i);

**if** (p) {

System.***out***.println("Image is loaded");

} **else** {

System.***out***.println("Image not loaded");

}

System.***out***.println("The title of the page is :" + driver.getTitle());

**try** {

**long** lastHeight = (**long**) ((JavascriptExecutor) driver).executeScript("return document.body.scrollHeight");

**while** (**true**) {

((JavascriptExecutor) driver).executeScript("window.scrollTo(0, document.body.scrollHeight);");

Thread.*sleep*(2000);

**long** newHeight = (**long**) ((JavascriptExecutor) driver)

.executeScript("return document.body.scrollHeight");

**if** (newHeight == lastHeight) {

**break**;

}

lastHeight = newHeight;

}

**if** (lastHeight > 0)

System.***out***.println("Scrolled & Height of Page : " + lastHeight);

**else**

System.***out***.println("Not Scrolled");

} **catch** (InterruptedException e) {

e.printStackTrace();

}

}

}

**pom.xml**

<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com</groupId>

<artifactId>ecommercewebsite</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>war</packaging>

<name>ecommercewebsite Maven Webapp</name>

<!-- FIXME change it to the project's website -->

<url>http://www.example.com</url>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<maven.compiler.source>1.7</maven.compiler.source>

<maven.compiler.target>1.7</maven.compiler.target>

</properties>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.11</version>

<scope>test</scope>

</dependency>

<dependency>

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

<artifactId>selenium-java</artifactId>

<version>4.4.0</version>

</dependency>

<dependency>

<groupId>org.testng</groupId>

<artifactId>testng</artifactId>

<version>7.0.0</version>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<finalName>Automation</finalName>

<pluginManagement><!-- lock down plugins versions to avoid using Maven

defaults (may be moved to parent pom) -->

<plugins>

<plugin>

<artifactId>maven-clean-plugin</artifactId>

<version>3.1.0</version>

</plugin>

<!-- see http://maven.apache.org/ref/current/maven-core/default-bindings.html#Plugin\_bindings\_for\_war\_packaging -->

<plugin>

<artifactId>maven-resources-plugin</artifactId>

<version>3.0.2</version>

</plugin>

<plugin>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.8.0</version>

</plugin>

<plugin>

<artifactId>maven-surefire-plugin</artifactId>

<version>2.22.1</version>

</plugin>

<plugin>

<artifactId>maven-war-plugin</artifactId>

<version>3.2.2</version>

</plugin>

<plugin>

<artifactId>maven-install-plugin</artifactId>

<version>2.5.2</version>

</plugin>

<plugin>

<artifactId>maven-deploy-plugin</artifactId>

<version>2.8.2</version>

</plugin>

</plugins>

</pluginManagement>

</build>

</project>