**Automate An E-Commerce Web Application**

**SourceCode :-**

**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>E-commerce</groupId>

<artifactId>Automation</artifactId>

< version>0.0.1-SNAPSHOT</ version>

< packaging>war</ packaging>

< name>Automation 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>javax.servlet</groupId>

<artifactId>javax.servlet-api</artifactId>

< version>4.0.1</ version>

< scope>provided</ scope></ dependency>

< dependency>

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

<artifactId>selenium-java</artifactId>

< version>4.10.0</ version></ dependency>

<!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-chrome-driver -->< dependency>

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

<artifactId>selenium-chrome-driver</artifactId>

< version>4.10.0</ version>

</ dependency>< dependency>

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

<artifactId>selenium-firefox-driver</artifactId>

< version>4.10.0</ version>

</ dependency><!-- https://mvnrepository.com/artifact/org.testng/testng -->

< dependency>

<groupId>org.testng</groupId>

<artifactId>testng</artifactId>

< version>7.0.0</ version>

< scope>test</ scope>

</ dependency>

< dependency>

<groupId>commons-io</groupId>

<artifactId>commons-io</artifactId>

< version>2.6</ version>

</ 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\_pa ckaging -->

< 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>

**AutomationFlipkart : package**  ecommerce;

**import** java.io.File;

**import** java.io.IOException;  **import** java.time.Duration;  **import** java.util.concurrent.TimeUnit ;

**import** org.apache.commons.io.FileUtils;  **import** org.openqa.selenium.By;  **import** org.openqa.selenium.JavascriptExecutor;  **import** org.openqa.selenium.OutputType ;  **import** org.openqa.selenium.TakesScreenshot;  **import** org.openqa.selenium.WebDriver;  **import** org.openqa.selenium.WebElement;  **import** org.openqa.selenium.chrome.ChromeDriver;  **import** org.openqa.selenium.edge.EdgeDriver;  **import** org.openqa.selenium.support.ui.ExpectedConditions;  **import** org.openqa.selenium.support.ui.WebDriverWait;  **import** org.testng.annotations.Test;

**public class**AutomationFlipkart{

**private static** String *url* = "https://www.flipkart.com/" ;

@Test (groups = "Chrome" )  **public void** LaunchChrome() {

//System.setProperty("webdriver.chrome.driver","C:\Users\Prudhvi \Downloads\chromedriver\_win32\c hromedriver.exe");

**try**  {

Thread. *sleep* (2000);

}  **catch** (Exception e ) {

e .printStackTrace();

}

}

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

**public void** f() {

WebDriver driver =  **new** ChromeDriver(); driver .manage().window().maximize(); //driver.get("https://www.flipkart.com/"); driver .get(*url* );

*pageLoadtime*( driver , *url* );

// close button

*screenshot*( driver , "output-1" );

driver .findElement(By.*xpath*( "/html/body/div[2]/div/div/button" )).click();

// mobile category

*screenshot*( driver , "output-2" );

driver .findElement(By.*xpath*( "//\*[@id=\"container\"]/div/div[2]/div/div/div[2]/a/div[1]/div/img" )).click();

*checkImageLoaded*( driver );

*ScrollHeight*( driver );

}

@Test (groups = "Edge" )

**public void**LaunchEdge() {

**try**  {

Thread. *sleep* (4000);

}  **catch** (Exception e ) {

e .printStackTrace();

}

}

@Test (groups= "Edge" ,dependsOnMethods= "LaunchEdge" )

**public void**firefox() {

WebDriver driver =  **new** EdgeDriver(); driver .manage().window().maximize(); //driver.get("https://www.flipkart.com/"); driver .get(*url* );

*pageLoadtime*( driver , *url* );

// close button

*screenshot*( driver , "Output-3" );

driver .findElement(By.*xpath*( "/html/body/div[2]/div/div/button" )).click();

// mobile category

*screenshot*( driver , "Output-4" );

driver .findElement(By.*xpath*( "//\*[@id=\"container\"]/div/div[2]/div/div/div[2]/a/div[1]/div/img" )).click();

*screenshot*( driver , "Output-5" ); *checkImageLoaded* ( driver );

*ScrollHeight*( driver );

}

**private static void** ScrollHeight(WebDriver driver ) {

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

}

}  **catch** (InterruptedException e ) { e .printStackTrace();

}

}

**private static void** checkImageLoaded(WebDriver driver ) {

driver .findElement(By.*className*( "\_3704LK" )).sendKeys( "iphone 13" ); driver .findElement(By.*className*( "L0Z3Pu" )).submit();

// iphone13 click

WebDriverWait wait =  **new** WebDriverWait( driver , Duration .*ofSeconds* (8)); wait .until(ExpectedConditions

.*visibilityOfElementLocated* (By.*xpath*( "//\*[@id=\"container\"]/div/div[3]/div[1]/div[2]/div[2]/div/div/div/a/ div[2]/div[1]/div/div/img" )));

driver .findElement(By.*xpath*( "//\*[@id=\"container\"]/div/div[3]/div[1]/div[2]/div[2]/div/div/div/a/div[2]/div

[1]/div/div/img" )).click();

driver .get( "https://www.flipkart.com/apple-iphone-13-midnight-128-gb/p/itmca361aab1c5b0?pid=MO

BG6VF5Q82T3XRS&lid=LSTMOBG6VF5Q82T3XRSOXJLM9&marketplace=FLIPKART&q=iphone +13&store=tyy%2F4io&spotlightTagId=BestsellerId\_tyy%2F4io&srno=s\_1\_1&otracker=search&otra cker1=search&fm=Search&iid=a6d4e97e-4f85-4e7e-b6a3-35911f5ebe0f.MOBG6VF5Q82T3XRS.S EARCH&ppt=sp&ppn=sp&ssid=4kft6zzs6o0000001690806005781&qH=c68a3b83214bb235" );

// identify image

WebElement i =

driver .findElement(By.*xpath*( "//\*[@id=\"container\"]/div/div[3]/div[1]/div[1]/div[1]/div/div[1]/div[2]/div[1]/ div[2]/img" ));

// Javascript executor to check image

Boolean p = (Boolean) ((JavascriptExecutor) driver ).executeScript( "return

arguments[0].complete "

+ "&&typeof arguments[0].naturalWidth != \"undefined\" " + "&&

arguments[0].naturalWidth> 0" , i );

// verify if status is true  **if** ( p ) {

System.***out*** .println( "Image Loaded" );

*screenshot*( driver , "output-6" );

}  **else** {

System.***out*** .println( "Image Not Loaded" );

}

}

**private static void** pageLoadtime(WebDriver driver , String url ) {

**long**  s = System.*currentTimeMillis* ();

// URL launch

driver .get(url );

// verify page is loaded

WebDriverWait wait =  **new** WebDriverWait( driver , Duration .*ofSeconds* (8));

wait .until(ExpectedConditions.*elementToBeClickable* (By.*xpath*( "/html/body/div[2]/div/div/button" ))); // close button

// capture time after page load  **long** e = System.*currentTimeMillis* ();

// compute time

**long**  r = e - s ;

System.***out*** .println( "Page load time in milliseconds: " + r );  *screenshot* ( driver , "output-7" );

}

**public static void**  screenshot(WebDriver driver ,String screenshotName ){

TakesScreenshotts = (TakesScreenshot) driver ; File scr = ts .getScreenshotAs(OutputType .***FILE*** );  **try**  {

FileUtils.*copyFile*(scr ,  **new**  File(screenshotName + ".png" )); System.***out*** .println( "Screenshot taken" );

}  **catch** (IOException e ) {

e .printStackTrace();

}

}

}