Sourcecode:

1)Apptest.java

package com.demo;

import org.testng.annotations.Test;

import org.testng.AssertJUnit;

import org.testng.annotations.Test;

import org.testng.AssertJUnit;

/\*\*

\* Unit test for simple App.

\*/

public class AppTest

{

/\*\*

\* Create the test case

\*

\* @param testName name of the test case

\*/

public AppTest( String testName )

{

}

/\*\*

\* Rigourous Test :-)

\*/

@Test

public void testApp()

{

AssertJUnit.assertTrue( true );

}

}

2)FlipkartTestChrome.java

package com.demo;

import org.testng.annotations.AfterClass;

import org.testng.annotations.Test;

import org.testng.annotations.BeforeClass;

import org.testng.annotations.AfterClass;

import org.testng.annotations.Test;

import org.testng.annotations.BeforeClass;

import org.testng.annotations.Test;

import java.io.File;

import java.io.IOException;

import java.util.concurrent.TimeUnit;

import java.util.function.Function;

import org.apache.commons.io.FileUtils;

import org.openqa.selenium.By;

import org.openqa.selenium.Dimension;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.NoSuchElementException;

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.support.ui.FluentWait;

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

import org.testng.annotations.AfterClass;

import org.testng.annotations.BeforeClass;

import org.openqa.selenium.TakesScreenshot;

public class FlipkartTestChrome {

WebDriver driver;

@BeforeClass

public void beforeClass() {

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

"C:\\Users\\Umera\\Downloads\\chromedriver\_win32\\chromedriver.exe");

driver = new ChromeDriver();

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

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

}

@AfterClass

public void afterClass() {

driver = null;

}

@Test(priority = 1)

public void closeLogin() throws InterruptedException {

try {

System.out.println("\nChrome Browser Result:\n");

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

driver.findElement(By.cssSelector("body > div.\_2Sn47c > div > div > button")).click();

Thread.sleep(1000);

} catch (NoSuchElementException e) {

e.printStackTrace();

}

screenshot(driver,"closelogin");

}

@Test(priority = 2)

public void scroll() throws InterruptedException {

Thread.sleep(2000);

JavascriptExecutor js = (JavascriptExecutor) driver;

js.executeScript("window.scrollBy(0,document.body.scrollHeight)");

System.out.println("\nNavigated to bottom of the page");

Thread.sleep(2000);

js.executeScript("window.scrollBy(0,-document.body.scrollHeight)", "");

System.out.println("\nScroll Feature available");

Thread.sleep(2000);

screenshot(driver,"scroll");

}

@Test(priority = 3)

public void searchProduct() throws InterruptedException {

Thread.sleep(1000);

driver.findElement(By.name("q")).sendKeys("iPhone 13");

Thread.sleep(1000);

By search = By.cssSelector(

"#container > div > div.\_1kfTjk > div.\_1rH5Jn > div.\_2Xfa2\_ > div.\_1cmsER > form > div > button > svg");

driver.findElement(search).click();

Thread.sleep(3000);

By load = By.cssSelector(

"#container > div > div.\_36fx1h.\_6t1WkM.\_3HqJxg > div.\_1YokD2.\_2GoDe3 > div:nth-child(2) > div:nth-child(9) > div > div");

long start = System.currentTimeMillis();

driver.findElement(load).click();

long finish = System.currentTimeMillis();

long totalTime = finish - start;

System.out.println("\nTime to load page in millisecs - " + totalTime);

screenshot(driver,"searchproduct");

}

@Test(priority = 4)

public void loadImage() throws InterruptedException {

String url = "<https://www.flipkart.com/apple-iphone-13-blue-256-gb/p/itmd68a015aa1e39?pid=MOBG6VF566ZTUVFR&lid=LSTMOBG6VF566ZTUVFR2RQLVU&marketplace=FLIPKART&q=iPhone+13&store=tyy%2F4io&srno=s_1_8&otracker=search&otracker1=search&fm=organic&iid=1c0c7402-fe4f-4f45-9aa8-cc59dffe8503.MOBG6VF566ZTUVFR.SEARCH&ppt=hp&ppn=homepage&ssid=i4t60bsv4g0000001665375424769&qH=c3d519be0191fbf8>";

driver.get(url);

Thread.sleep(3000);

//driver.navigate().refresh();

Wait<WebDriver> wait = new FluentWait<WebDriver>(driver).withTimeout(10, TimeUnit.SECONDS)

.pollingEvery(2, TimeUnit.SECONDS).ignoring(NoSuchElementException.class);

wait.until(new Function<WebDriver, WebElement>() {

@Test

public WebElement apply(WebDriver driver) {

WebElement img = driver.findElement(By.xpath(

"//\*[@id=\"container\"]/div/div[3]/div[1]/div[2]/div[9]/div[4]/div[3]/div/div/div/div[1]/img"));

if (img.isDisplayed()) {

System.out.println("\nImage Loaded");

return img;

} else {

System.out.println("\nFluent Wait Fail!, Element Not Loaded Yet");

return null;

}

}

});

screenshot(driver,"pageLoad");

}

@Test(priority = 5)

public void scrollFrequency() throws InterruptedException {

Thread.sleep(2000);

long start = System.currentTimeMillis();

WebElement element = driver.findElement(By.cssSelector(

"#container > div > div.\_2c7YLP.UtUXW0.\_6t1WkM.\_3HqJxg > div.\_1YokD2.\_2GoDe3 > div.\_1YokD2.\_3Mn1Gg.col-8-12 > div.\_1YokD2.\_3Mn1Gg > div:nth-child(7) > div > div:nth-child(3) > div > div > div:nth-child(1)"));

((JavascriptExecutor) driver).executeScript("arguments[0].scrollIntoView(true);", element);

long stop = System.currentTimeMillis();

long totalTime = stop - start;

System.out.println("\nScroll Frequency in millisecs - " + totalTime);

screenshot(driver,"scrollfrequency");

}

@Test(priority = 6)

public void downloadImages() throws InterruptedException {

WebElement img = driver.findElement(By

.xpath("//\*[@id=\"container\"]/div/div[3]/div[1]/div[2]/div[9]/div[4]/div[3]/div/div/div/div[1]/img"));

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

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

if (p) {

System.out.println("\nImage present");

} else {

System.out.println("\nImage not present");

}

screenshot(driver,"downloadImages");

}

@Test(priority = 7)

public void screenResolution() throws InterruptedException {

Thread.sleep(1000);

Dimension dimension = new Dimension(720, 1080);

driver.manage().window().setSize(dimension);

Thread.sleep(3000);

Dimension dimension1 = new Dimension(1280, 800);

driver.manage().window().setSize(dimension1);

Thread.sleep(3000);

Dimension dimension2 = new Dimension(2256, 1504);

driver.manage().window().setSize(dimension2);

JavascriptExecutor js = (JavascriptExecutor) driver;

int contentHeight = ((Number) js.executeScript("return window.innerHeight")).intValue();

int contentWidth = ((Number) js.executeScript("return window.innerWidth")).intValue();

System.out.println("\nHeight: " + contentHeight + " Width: " + contentWidth + "\n");

screenshot(driver,"screenshotResolution");

}

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

TakesScreenshot ts = (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();

}

}

}

3)FlipkartTestChrome1.java

package com.demo;

import org.testng.annotations.AfterClass;

import org.testng.annotations.Test;

import org.testng.annotations.BeforeClass;

import org.testng.annotations.AfterClass;

import org.testng.annotations.Test;

import org.testng.annotations.BeforeClass;

import org.testng.annotations.Test;

import java.io.File;

import java.io.IOException;

import java.util.concurrent.TimeUnit;

import java.util.function.Function;

import org.apache.commons.io.FileUtils;

import org.openqa.selenium.By;

import org.openqa.selenium.Dimension;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.NoSuchElementException;

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.support.ui.FluentWait;

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

import org.testng.annotations.AfterClass;

import org.testng.annotations.BeforeClass;

import org.openqa.selenium.TakesScreenshot;

public class FlipkartTestChrome {

WebDriver driver;

@BeforeClass

public void beforeClass() {

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

"C:\\Users\\Umera\\Downloads\\chromedriver\_win32\\chromedriver.exe");

driver = new ChromeDriver();

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

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

}

@AfterClass

public void afterClass() {

driver = null;

}

@Test(priority = 1)

public void closeLogin() throws InterruptedException {

try {

System.out.println("\nChrome Browser Result:\n");

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

driver.findElement(By.cssSelector("body > div.\_2Sn47c > div > div > button")).click();

Thread.sleep(1000);

} catch (NoSuchElementException e) {

e.printStackTrace();

}

screenshot(driver,"closelogin");

}

@Test(priority = 2)

public void scroll() throws InterruptedException {

Thread.sleep(2000);

JavascriptExecutor js = (JavascriptExecutor) driver;

js.executeScript("window.scrollBy(0,document.body.scrollHeight)");

System.out.println("\nNavigated to bottom of the page");

Thread.sleep(2000);

js.executeScript("window.scrollBy(0,-document.body.scrollHeight)", "");

System.out.println("\nScroll Feature available");

Thread.sleep(2000);

screenshot(driver,"scroll");

}

@Test(priority = 3)

public void searchProduct() throws InterruptedException {

Thread.sleep(1000);

driver.findElement(By.name("q")).sendKeys("iPhone 13");

Thread.sleep(1000);

By search = By.cssSelector(

"#container > div > div.\_1kfTjk > div.\_1rH5Jn > div.\_2Xfa2\_ > div.\_1cmsER > form > div > button > svg");

driver.findElement(search).click();

Thread.sleep(3000);

By load = By.cssSelector(

"#container > div > div.\_36fx1h.\_6t1WkM.\_3HqJxg > div.\_1YokD2.\_2GoDe3 > div:nth-child(2) > div:nth-child(9) > div > div");

long start = System.currentTimeMillis();

driver.findElement(load).click();

long finish = System.currentTimeMillis();

long totalTime = finish - start;

System.out.println("\nTime to load page in millisecs - " + totalTime);

screenshot(driver,"searchproduct");

}

@Test(priority = 4)

public void loadImage() throws InterruptedException {

String url = "<https://www.flipkart.com/apple-iphone-13-blue-256-gb/p/itmd68a015aa1e39?pid=MOBG6VF566ZTUVFR&lid=LSTMOBG6VF566ZTUVFR2RQLVU&marketplace=FLIPKART&q=iPhone+13&store=tyy%2F4io&srno=s_1_8&otracker=search&otracker1=search&fm=organic&iid=1c0c7402-fe4f-4f45-9aa8-cc59dffe8503.MOBG6VF566ZTUVFR.SEARCH&ppt=hp&ppn=homepage&ssid=i4t60bsv4g0000001665375424769&qH=c3d519be0191fbf8>";

driver.get(url);

Thread.sleep(3000);

//driver.navigate().refresh();

Wait<WebDriver> wait = new FluentWait<WebDriver>(driver).withTimeout(10, TimeUnit.SECONDS)

.pollingEvery(2, TimeUnit.SECONDS).ignoring(NoSuchElementException.class);

wait.until(new Function<WebDriver, WebElement>() {

@Test

public WebElement apply(WebDriver driver) {

WebElement img = driver.findElement(By.xpath(

"//\*[@id=\"container\"]/div/div[3]/div[1]/div[2]/div[9]/div[4]/div[3]/div/div/div/div[1]/img"));

if (img.isDisplayed()) {

System.out.println("\nImage Loaded");

return img;

} else {

System.out.println("\nFluent Wait Fail!, Element Not Loaded Yet");

return null;

}

}

});

screenshot(driver,"pageLoad");

}

@Test(priority = 5)

public void scrollFrequency() throws InterruptedException {

Thread.sleep(2000);

long start = System.currentTimeMillis();

WebElement element = driver.findElement(By.cssSelector(

"#container > div > div.\_2c7YLP.UtUXW0.\_6t1WkM.\_3HqJxg > div.\_1YokD2.\_2GoDe3 > div.\_1YokD2.\_3Mn1Gg.col-8-12 > div.\_1YokD2.\_3Mn1Gg > div:nth-child(7) > div > div:nth-child(3) > div > div > div:nth-child(1)"));

((JavascriptExecutor) driver).executeScript("arguments[0].scrollIntoView(true);", element);

long stop = System.currentTimeMillis();

long totalTime = stop - start;

System.out.println("\nScroll Frequency in millisecs - " + totalTime);

screenshot(driver,"scrollfrequency");

}

@Test(priority = 6)

public void downloadImages() throws InterruptedException {

WebElement img = driver.findElement(By

.xpath("//\*[@id=\"container\"]/div/div[3]/div[1]/div[2]/div[9]/div[4]/div[3]/div/div/div/div[1]/img"));

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

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

if (p) {

System.out.println("\nImage present");

} else {

System.out.println("\nImage not present");

}

screenshot(driver,"downloadImages");

}

@Test(priority = 7)

public void screenResolution() throws InterruptedException {

Thread.sleep(1000);

Dimension dimension = new Dimension(720, 1080);

driver.manage().window().setSize(dimension);

Thread.sleep(3000);

Dimension dimension1 = new Dimension(1280, 800);

driver.manage().window().setSize(dimension1);

Thread.sleep(3000);

Dimension dimension2 = new Dimension(2256, 1504);

driver.manage().window().setSize(dimension2);

JavascriptExecutor js = (JavascriptExecutor) driver;

int contentHeight = ((Number) js.executeScript("return window.innerHeight")).intValue();

int contentWidth = ((Number) js.executeScript("return window.innerWidth")).intValue();

System.out.println("\nHeight: " + contentHeight + " Width: " + contentWidth + "\n");

screenshot(driver,"screenshotResolution");

}

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

TakesScreenshot ts = (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();

}

}

}

4)testing.xml

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

<!DOCTYPE suite SYSTEM "<https://testng.org/testng-1.0.dtd>">

<suite name=*"Suite"*>

<test thread-count=*"5"* name=*"Test"*>

<classes>

<class name=*"com.demo.FlipkartTestChrome"*/>

<class name=*"com.demo.AppTest"*/>

<class name=*"com.demo.FlipkartTestChrome1"*/>

</classes>

</test> <!-- Test -->

</suite> <!-- Suite -->