Working with External Elements.

ExternalElementsHandlingDemo.java

```java
package com.ecommerce.test;

import java.util.Set;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class ExternalElementsHandlingDemo {

public static void main(String[] args) {
WebDriver driver = new ChromeDriver();
driver.get("https://www.flipkart.com/");

// Handling External Pop-ups
handleExternalPopUps(driver);

// Handling New Tabs and Windows
handleNewTabsAndWindows(driver);

driver.quit();
}

static void handleExternalPopUps(WebDriver driver) {
// Click on the trigger element to open the pop-up
driver.findElement(By.id("triggerElement")).click();
```

```java
// Get a set of window handles
Set<String> windowHandles = driver.getWindowHandles();


// Store the handle of the main window
String mainWindowHandle = driver.getWindowHandle();


// Iterate through the window handles
for (String handle : windowHandles) {
// Switch to the pop-up window
driver.switchTo().window(handle);


// Perform actions on the elements within the pop-up
driver.findElement(By.id("popUpElement")).sendKeys("Text");


// Switch back to the main window
driver.switchTo().window(mainWindowHandle);
}
}


static void handleNewTabsAndWindows(WebDriver driver) {
// Click on the trigger element to open the new tab or window
driver.findElement(By.id("triggerElement")).click();


// Get a set of window handles
Set<String> windowHandles = driver.getWindowHandles();


// Store the handle of the main window
```

```java
String mainWindowHandle = driver.getWindowHandle();


// Iterate through the window handles
for (String handle : windowHandles) {
// Switch to the new tab or window
driver.switchTo().window(handle);


// Perform actions on the elements within the new tab or window
driver.findElement(By.id("newTabElement")).sendKeys("Text");


// Close the new tab or window if needed
driver.close();


// Switch back to the main window
driver.switchTo().window(mainWindowHandle);
}
}
}
```