Source Code Of Automate an E-Commerce Web Application-insist

**package** com.example;

**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.support.ui.ExpectedConditions;

**import** org.openqa.selenium.support.ui.WebDriverWait;

**import** org.testng.annotations.AfterClass;

**import** org.testng.annotations.BeforeClass;

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

**public** **class** NewTest1 {

**private** WebDriver driver;

@BeforeClass

**public** **void** setUp() {

// Set the path to your ChromeDriver executable

System.*setProperty*("webdriver.chrome.driver", "C:\\Users\\Lenovo\\Downloads\\chromedriver-win32\\chromedriver-win32\\chromedriver.exe");

driver = **new** ChromeDriver();

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

}

@Test

**public** **void** testProductSearch() {

// Navigate to the Flipkart homepage

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

// Perform page load time measurement

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

// Search for "iPhone 13" under the "Mobile" category

WebElement searchBox = driver.findElement(By.*name*("q"));

searchBox.sendKeys("iPhone 13");

searchBox.submit();

// Explicit wait for search results to load

WebDriverWait wait = **new** WebDriverWait(driver, 10);

WebElement productImage = wait.until(ExpectedConditions.*visibilityOfElementLocated*(By.*xpath*("//img[@alt='iPhone 13']")));

// Check if the images are loaded and visible till the screen height only

JavascriptExecutor js = (JavascriptExecutor) driver;

**boolean** isImageVisible = (**boolean**) js.executeScript(

"return arguments[0].complete && typeof arguments[0].naturalWidth != 'undefined' && arguments[0].naturalWidth > 0;",

productImage

);

**assert** isImageVisible : "Product image is not loaded or visible.";

// Check if the page has a scroll feature

**boolean** isScrollable = (**boolean**) js.executeScript("return document.documentElement.scrollHeight>document.documentElement.clientHeight;");

**assert** isScrollable : "Page is not scrollable.";

// Content Refresh Frequency Check

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

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

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

**long** contentRefreshDuration = contentRefreshEndTime - contentRefreshStartTime;

System.***out***.println("Time between content refreshes: " + contentRefreshDuration + " milliseconds");

// Scroll to the position of the product image

js.executeScript("arguments[0].scrollIntoView();", productImage);

// Measure the time it takes for the image to load

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

**boolean** isImageLoaded = (**boolean**) js.executeScript(

"return arguments[0].complete && typeof arguments[0].naturalWidth != 'undefined' && arguments[0].naturalWidth > 0;",

productImage

);

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

**if** (isImageLoaded) {

**long** imageLoadTime = imageLoadEndTime - imageLoadStartTime;

System.***out***.println("Image loaded in: " + imageLoadTime + " milliseconds");

} **else** {

System.***out***.println("Image did not load in time.");

}

// Scroll to the bottom of the page

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

// Implement checks for different browsers and screen resolutions

}

@AfterClass

**public** **void** tearDown() {

driver.quit();

}

}