CODE

POM

```
<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>Automate_Flipkart</groupId>
	<artifactId>Automate_Flipkart</artifactId>
	<version>0.0.1-SNAPSHOT</version>
	<packaging>jar</packaging>

	<name>Automate_Flipkart</name>
	<url>http://maven.apache.org</url>

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

	<dependencies>
		<dependency>
			<groupId>io.github.bonigarcia</groupId>
			<artifactId>webdrivermanager</artifactId>
			<version>5.6.2</version>
		</dependency>

		<dependency>
			<groupId>org.seleniumhq.selenium</groupId>
			<artifactId>selenium-java</artifactId>
			<version>3.141.59</version>
		</dependency>

		<dependency>
			<groupId>org.uncommons</groupId>
			<artifactId>reportng</artifactId>
			<version>1.1.4</version>
		</dependency>

		<dependency>
			<groupId>com.google.inject</groupId>
			<artifactId>guice</artifactId>
			<version>4.2.2</version>
		</dependency>

	</dependencies>
</project>
```

Testing

```
<?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="Automate_Flipkart.Automate_Flipkart.AppTest"/>
      <class name="Automate_Flipkart.App"/>
    </classes>
  </test> <!-- Test -->
</suite> <!-- Suite -->
```

App

```
package Automate_Flipkart;

import io.github.bonigarcia.wdm.WebDriverManager;
import org.openqa.selenium.By;
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.Keys;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.chrome.ChromeOptions;
import org.testng.annotations.AfterTest;
import org.testng.annotations.BeforeTest;
import org.testng.annotations.Test;

import java.util.List;
import java.util.concurrent.TimeUnit;

public class App {

    static WebDriver wd;

    @BeforeTest
    public void setupBraveWebDriver() {
        // Setup Brave browser with specific configurations
        WebDriverManager.chromedriver().setup();
        ChromeOptions options = new ChromeOptions();
        options.setBinary("C:\\Program Files\\BraveSoftware\\Brave-
Browser\\Application\\brave.exe");
        wd = new ChromeDriver(options);
        wd.manage().window().maximize();
        wd.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
    }

    @Test
    public void testFlipkart() {
        wd.get("https://www.flipkart.com/");
        measurePageLoadTime();
        searchProduct("iPhone 15", "Mobile");
        checkImageVisibility();
```

```
        checkPageScroll();
        checkContentRefreshFrequency();
        verifyImageDownload();
        navigateToBottom();
    }

    @AfterTest
    public void tearDown() {
        // Close the browser after the test
        if (wd != null) {
            wd.quit();
        }
    }

    private void measurePageLoadTime() {
        long loadTime = (long) ((JavascriptExecutor) wd)
                .executeScript("return performance.timing.loadEventEnd -
performance.timing.navigationStart;");
        System.out.println("Page Load Time: " + loadTime + " milliseconds");
    }

    private void searchProduct(String productName, String category) {
        WebElement searchBox = wd.findElement(By.name("q"));
        searchBox.sendKeys(productName , Keys.ENTER);

        // Allow time for the search results to load
        wd.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
    }

    private void checkImageVisibility() {
        List<WebElement> images = wd.findElements(By.cssSelector(".product-
image img"));
        for (WebElement image : images) {
            if (isElementInViewPort(image)) {
                System.out.println("Image is visible till the screen
height.");
            } else {
                System.out.println("Image is not visible till the screen
height.");
            }
        }
    }

    private boolean isElementInViewPort(WebElement element) {
        return (boolean) ((JavascriptExecutor) wd).executeScript("var rect =
arguments[0].getBoundingClientRect(); "
                + "return (rect.top >= 0 && rect.bottom <=
window.innerHeight);", element);
    }

    private void checkPageScroll() {
        ((JavascriptExecutor) wd).executeScript("window.scrollTo(0,
document.body.scrollHeight)");
        System.out.println("Page has been scrolled down.");
    }

    private void checkContentRefreshFrequency() {
```

```
        long startTime = System.currentTimeMillis();
        long currentTime;
        do {
            ((JavascriptExecutor) wd).executeScript("window.scrollTo(0,
document.body.scrollHeight)");
            try {
                Thread.sleep(500);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
            currentTime = System.currentTimeMillis();
        } while (currentTime - startTime < 5000);

        System.out.println("Content refresh frequency checked.");
    }

    private void verifyImageDownload() {
        WebElement lastImage = wd.findElement(By.className("_396cs4"));
        long startTime = System.currentTimeMillis();
        long currentTime;
        do {
            try {
                Thread.sleep(500);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
            currentTime = System.currentTimeMillis();
        } while (!isElementInViewPort(lastImage) && (currentTime - startTime
< 5000));

        System.out.println("Image download verified.");
    }

    private void navigateToBottom() {
        ((JavascriptExecutor) wd).executeScript("window.scrollTo(0,
document.body.scrollHeight)");
        System.out.println("Scrolled to the bottom of the page.");
    }
}
```