

Assignment 5 – GUI Testing

Bhavana Priya Kanumuri

Description of the application with screenshots of GUI:

I have used some open-source sample HTML code I found online to modify it according to the requirements of our assignment. It is a simple HTML website that has 3 pages. All three pages have images, text labels, paragraph labels, listed items, checkboxes, etc. The first page to launch the app is index.html.

The screenshot is attached below:

The screenshot shows a web browser window with the title "Fountain of Fruit". The address bar displays "File | C:/Users/bkanumur/Desktop/fall22/565/assignment5/index.html". Below the address bar, there are several icons for various services like Gmail, YouTube, Maps, News, Translate, and a login link. The main content area of the browser shows the "Fountain of Fruit" page. At the top, there are three buttons: "Home", "Bananas", and "Apples". The "Bananas" button is highlighted. Below these buttons, there is a checkbox labeled "I love fruits!!!". Underneath the checkbox, the word "Bananas" is written in bold. A blue link "Click here to view History of the Banana" is present. Below the link, a green note says "Alternatively click on the image below to see the Banana page." An image of a yellow banana on a pink background is displayed. At the bottom of the page, a green note reads "How many types of banana are there? Well it turns out there are thousands!" followed by two bullet points of placeholder text. A blue link "Click here for an external site on bananas." is located at the bottom left. The word "Apples" is underlined and bolded at the bottom right.

Fountain of Fruit

[Home](#) [Bananas](#) [Apples](#)

I love fruits!!!

Bananas

[Click here to view History of the Banana](#)

Alternatively click on the image below to see the Banana page.



How many types of banana are there? Well it turns out there are thousands!

- Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.
- Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

[Click here for an external site on bananas.](#)

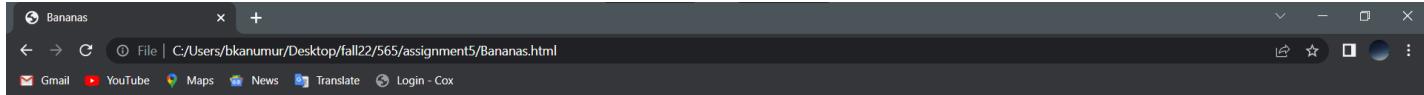
Apples

Fuji Apples



Index.html has a navigation bar that can take the user to the other two pages called Bananas.html and Apples.html. Additionally index.html has hyperlinked text and image which on clicking will redirect the user to the Bananas.html page.

In a similar way, Bananas.html also has a button which will redirect the user to Apples.html page.



This is a page about bananas.

[Home](#) [Bananas](#) [Apples](#)



[This would be text about the history of the banana.](#)

 Lorem ipsum dolor sit, amet consectetur adipisicing elit. Provident fugit corporis excepturi doloremque magni. Rem repellendus numquam aut beatae officiis porro totam amet accusamus eligendi? Nostrum et at deleniti minima! **Things you didn't know about the banana.**

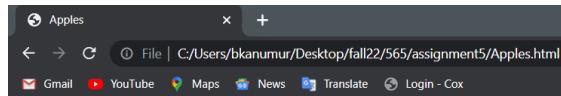
1. "Many believe the Forbidden Fruit consumed by Adam and Eve in the Garden of Eden was an apple. Others think it may have been a fig or a pomegranate. And then there is the speculation that the fruit may have been a banana.
2. There are over 14 different types of banana in the world!!

Alternatively click on the image below to see the Apples page. [Click to view some apples!](#)



[Go back to Home Page](#)

Apples.html is as shown below:



This is a page about apples.

[Home](#) [Bananas](#) [Apples](#)

Apples



I love apples!!!

Apples Big!!!!

- Apples big,
- Apples small,
- Apples green,
- Apples red,
- Apples yellow,



I love apples!!!

Apples Big!!!!

- Apples big,
- Apples small,
- Guess what?
- I like them all!

Worm Wiggle Into An Apple

Worm wiggle into an apple,
Worm wiggle into the core,
Worm wiggle out of the apple,
I just can't eat anymore.

[Go back to Home Page](#)

At the end of Bananas.html and Apples.html there is a button that redirects the user to the home page.

VERSION 2:

I have modified the website so that the output looks like the screenshots below:

The screenshot shows a Microsoft Edge browser window with two tabs open: 'Fountain of Fruit' and 'Fountain of Fruitsss'. The main content area displays the 'Bananas' page under the 'Fountain of Fruit' heading. It includes a checkbox for 'I love fruits!!!', a link to 'Click here to view History of the Banana', and an alternative link to 'Alternatively click on the image below to see the Banana page.' Below this is a large image of a yellow banana on a pink background. A caption below the image reads: 'How many types of banana are there? Well it turns out there are thousands!' followed by two bullet points of placeholder text. At the bottom, there is a link to 'Click here for an external site on bananas.' and a section for 'Apples' featuring a small image of two apples.

Fountain of Fruit

Home Bananas Apples

I love fruits!!!

Bananas

[Click here to view History of the Banana](#)

Alternatively click on the image below to see the Banana page.

How many types of banana are there? Well it turns out there are thousands!

- Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.
- Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

[Click here for an external site on bananas.](#)

Apples

Fuji Apples





This is a page about bananas.

[Home](#) [Bananas](#) [Apples](#)



This would be text about the history of the banana.

 Lorem ipsum dolor sit, amet consectetur adipisicing elit. Provident fugit corporis excepturi doloremque magni. Rem repellendus numquam aut beatae officiis porro totam amet accusamus eligendi? Nostrum et at deleniti minima! **Things you didn't know about the banana.**

1. "Many believe the Forbidden Fruit consumed by Adam and Eve in the Garden of Eden was an apple. Others think it may have been a fig or a pomegranate. And then there is the speculation that the fruit may have been a banana.
2. There are over 14 different types of banana in the world!!

Alternatively click on the image below to see the Apples page. [Click to view some apples!](#)



[Go back to Home Page](#)



This is a page about apples.

[Home](#) [Bananas](#) [Apples](#)

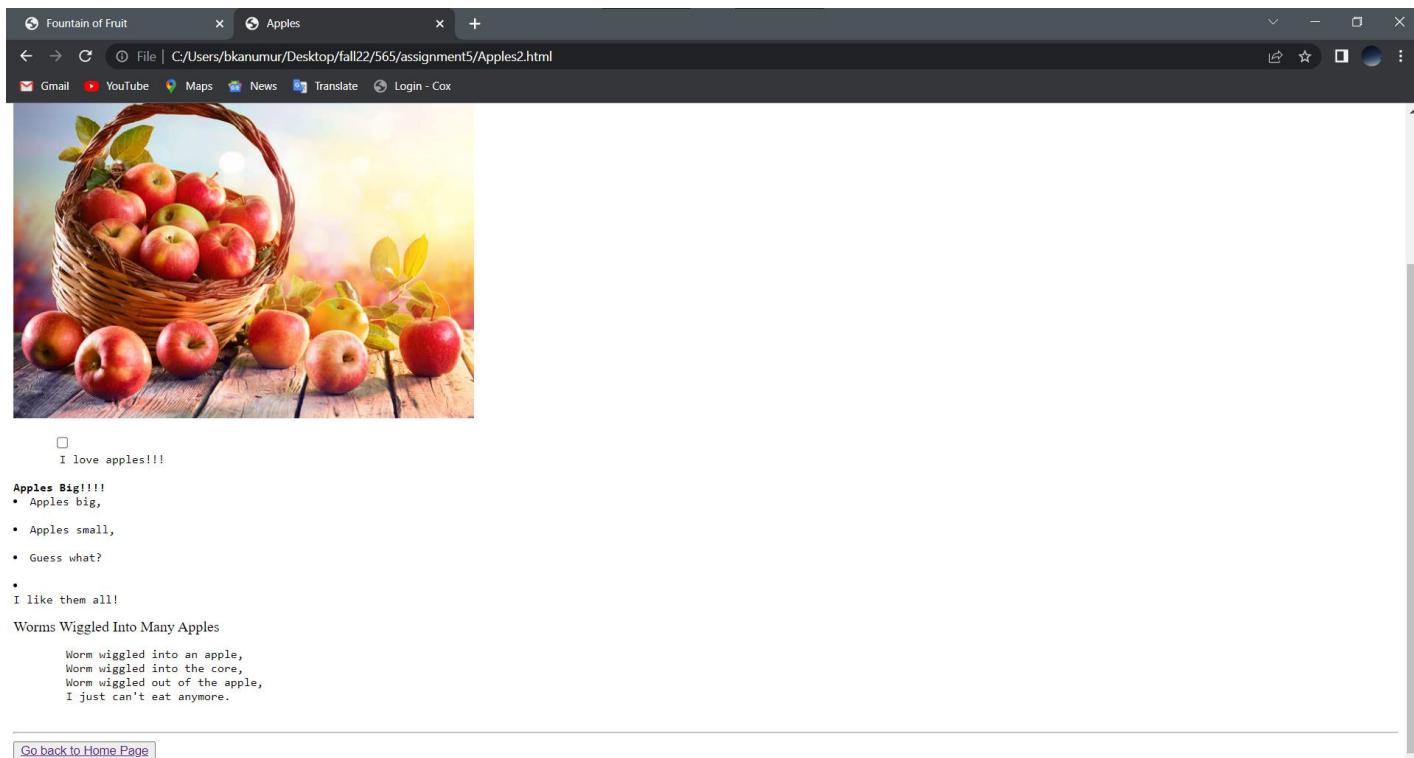
Apples



I love apples!!!

Apples Big!!!!

- Apples big,
- Apples small,
- Guess what?



Description of the tool selected for GUI testing:

I have used the **Selenium WebDriver** for performing the GUI testing. I have set up Selenium and TestNG as dependencies in my Maven project. Selenium WebDriver is a very efficient tool for performing automated testing on web applications. The WebDriver can be used to navigate both local files and remote websites to automate testing effectively. The WebDriver is a very compact object-oriented API.

It is able to detect and assert user actions such as inputs, keyboard strokes, mouse clicks, wheel scrolls on the mouse, etc. It enables testing over multiple different web browsers like Mozilla Firefox, Google Chrome, Internet Explorer, etc.

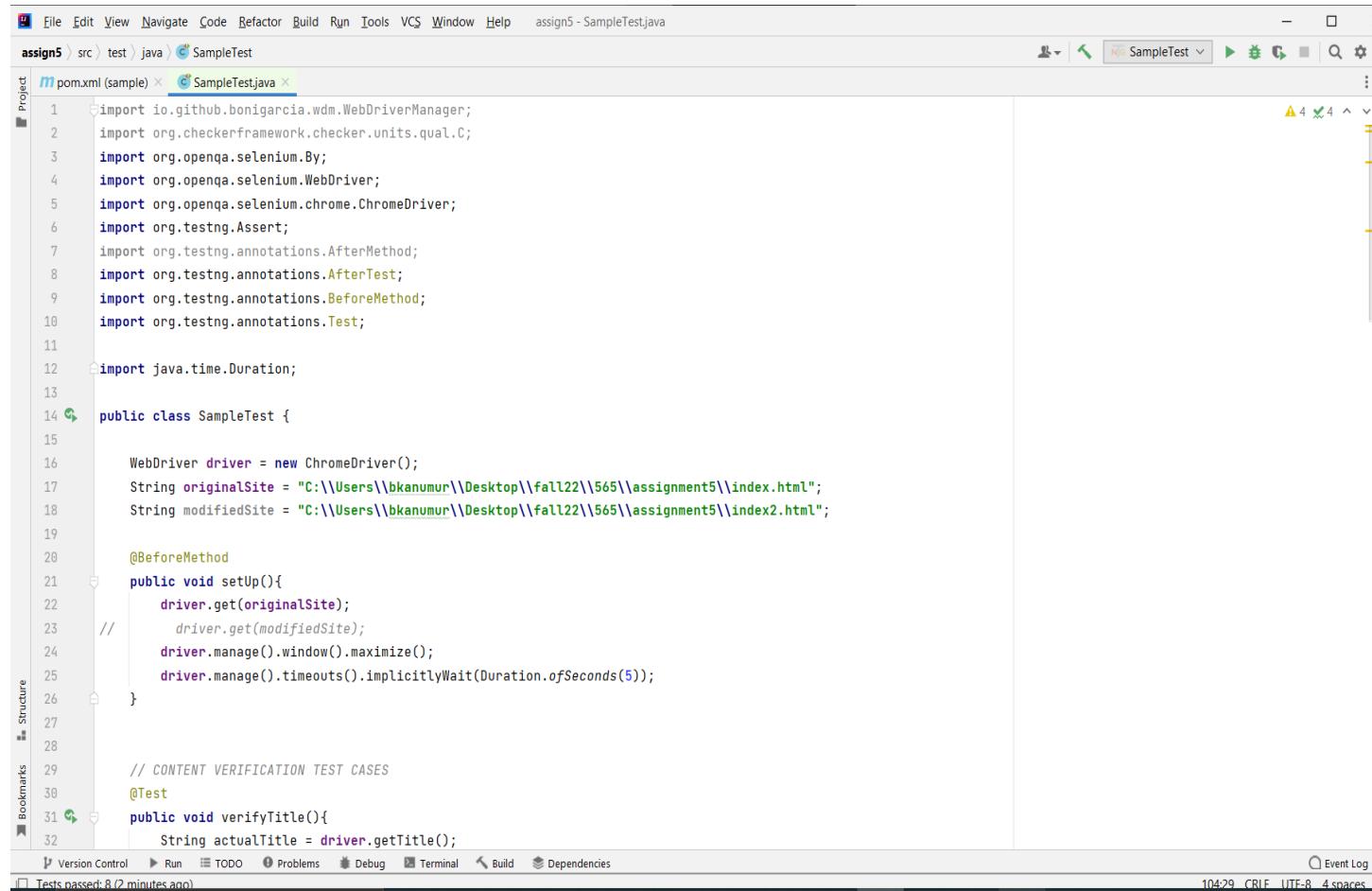
It is also very compatible with a lot of programming languages such as Java, Python, C#. The WebDriver is actually the better version of the selenium RC. However, webdriver is not capable of handling window components. Other tools such as Sikuli can be used in order to cover those test cases as well.

In this assignment I made use of the Chrome browser driver.

I have also used **TestNG Framework** for creating a test class and for generating test reports for the two different versions of the web app. It is a testing framework that is typically used for unit testing and integration testing. In the TestNG framework we make use of a few ‘annotations’ such as @Test to declare a testcase, @BeforeMethod to declare the logic that is supposed to run before the execution of every method and @AfterMethod to declare the logic that is supposed to run after the execution of every method. @BeforeTest and @AfterTest are executed before and after the test is run.

Description of the test cases developed:

I have developed a total of 8 test cases in order to test for navigating different pages, change in content, change in size of image, existence of elements, modified html elements, etc. Below I have attached the list of test cases I have used.



The screenshot shows an IDE interface with the following details:

- File Bar:** File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, VCS, Window, Help, assign5 - SampleTest.java
- Toolbars:** Standard toolbar with icons for file operations.
- Toolbar Buttons:** Includes icons for search, refresh, and other development tools.
- Project Explorer:** Shows the project structure with pom.xml and SampleTest.java selected.
- Code Editor:** Displays the Java code for SampleTest.java. The code imports various Selenium and TestNG annotations and classes, sets up a WebDriver, and contains a verifyTitle() method.

```
1 import io.github.bonigarcia.wdm.WebDriverManager;
2 import org.checkerframework.checker.units.qual.C;
3 import org.openqa.selenium.By;
4 import org.openqa.selenium.WebDriver;
5 import org.openqa.selenium.chrome.ChromeDriver;
6 import org.testng.Assert;
7 import org.testng.annotations.AfterMethod;
8 import org.testng.annotations.AfterTest;
9 import org.testng.annotations.BeforeMethod;
10 import org.testng.annotations.Test;
11
12 import java.time.Duration;
13
14 public class SampleTest {
15
16     WebDriver driver = new ChromeDriver();
17     String originalSite = "C:\\Users\\bkanumur\\Desktop\\fall22\\565\\assignment5\\index.html";
18     String modifiedSite = "C:\\Users\\bkanumur\\Desktop\\fall22\\565\\assignment5\\index2.html";
19
20     @BeforeMethod
21     public void setUp(){
22         driver.get(originalSite);
23         // driver.get(modifiedSite);
24         driver.manage().window().maximize();
25         driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(5));
26     }
27
28
29     // CONTENT VERIFICATION TEST CASES
30     @Test
31     public void verifyTitle(){
32         String actualTitle = driver.getTitle();
```

- Bottom Status Bar:** Version Control, Run, TODO, Problems, Debug, Terminal, Dependencies.
- Event Log:** Shows 10429 CRIF UTF-8 4 spaces.

The screenshot shows an IDE interface with the following details:

- Project:** assign5
- File:** SampleTest.java
- Code Content:**

```
28 // CONTENT VERIFICATION TEST CASES
29
30     @Test
31     public void verifyTitle(){
32         String actualTitle = driver.getTitle();
33         String expectedTitle = "Fountain of Fruit";
34         Assert.assertEquals(actualTitle, expectedTitle);
35     }
36
37     //verifying that the text content is the same
38     @Test
39     public void verifyTextInApples(){
40         driver.findElement(By.id("appleBtn")).click();
41         String text = driver.findElement(By.id("appleTxt")).getText();
42         System.out.println(text);
43         Assert.assertEquals(text, expected: "Worm Wigglee Into An Apple");
44     }
45
46     //EXISTENCE OF ELEMENT TEST CASES
47
48     //verifying that the banana image exists on the banana page
49     @Test
50     public void verifyBananaImgExists(){
51         driver.findElement(By.id("bananaBtn")).click();
52         String imgSrc = driver.findElement(By.id("bananaImg")).getDomAttribute( name: "src");
53         Assert.assertEquals(imgSrc, expected: "https://i.ytimg.com/vi/SpAHPXNhAwk/maxresdefault.jpg");
54     }
55
56     //verifying that the apple image exists on the home page
57     @Test
58     public void verifyAppleImgExists(){
59         String imgSrc = driver.findElement(By.id("appleimg")).getDomAttribute( name: "src");
60         Assert.assertEquals(imgSrc, expected: "https://www.realfoods.co.uk/ProductImagesID/16061_1.jpg");
61     }
```

- Toolbars and Menus:** File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, VCS, Window, Help
- Status Bar:** Tests passed: 8 (3 minutes ago), 27.5 CRIF, UTF-8, 4 spaces

The screenshot shows an IDE (likely IntelliJ IDEA) displaying a Java test class named `SampleTest.java`. The code is annotated with `@Test` and uses `driver` to interact with a web page. The IDE has a vertical toolbar on the left with icons for Project, Structure, Bookmarks, and others. The top bar includes File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, VCS, Window, Help, and a tab for `assign5 - SampleTest.java`. The bottom bar features tabs for Version Control, Run, TODO, Problems, Debug, Terminal, Build, and Dependencies, along with an Event Log window.

```
//ELEMENTS MODIFIED TEST CASES
@Test
public void verifyBananaImgWidth(){
    String width = driver.findElement(By.id("bananaImg")).getDomAttribute( name: "width");
    Assert.assertEquals(width, expected: "400");
}

//NAVIGATION TEST CASES
//verifying that clicking on the banana image on the home page redirects it to banana.html page
@Test
public void verifyBananaImg(){
    String src = driver.findElement(By.id("bananaImg")).getDomAttribute( name: "src");
    Assert.assertEquals(src, expected: "https://encrypted-tbn0.gstatic.com/images?q=tbn%3AAnd9GcSoGtRNlYb9s_NcmTE0BZu8NeJa82efGXn2Yr6l0AY");
}

//verifying that the link on bananas page takes it to apples page
@Test
public void applesBtntoBanana(){
    driver.findElement(By.id("bananaBtn")).click();
    String link = driver.findElement(By.id("clickApple")).getDomAttribute( name: "href");
    Assert.assertEquals(link, expected: "Apples.html");
}

// To verify correct navigation from first page to the third and back to first page
@Test
public void verifyNavigation(){
    driver.findElement(By.id("toBananas")).click();
    driver.findElement(By.id("toApple")).click();
    driver.findElement(By.tagName("a")).click();
    String title = driver.findElement(By.tagName("h1")).getText();
    Assert.assertEquals(title, expected: "Fountain of Fruit");
}
```

The screenshot shows an IDE interface with the following details:

- Project:** assign5
- File:** SampleTest.java
- Code Content:**

```
75     }
76 }
77
78 //verifying that the link on bananas page takes it to apples page
79 @Test
80 public void applesBtntoBanana(){
81     driver.findElement(By.id("bananaBtn")).click();
82     String link = driver.findElement(By.id("clickApple")).getDomAttribute( name: "href");
83     Assert.assertEquals(link, expected: "Apples.html");
84 }
85
86 // To verify correct navigation from first page to the third and back to first page
87 @Test
88 public void verifyNavigation(){
89     driver.findElement(By.id("toBananas")).click();
90     driver.findElement(By.id("toApple")).click();
91     driver.findElement(By.tagName("a")).click();
92     String title = driver.findElement(By.tagName("h1")).getText();
93     Assert.assertEquals(title, expected: "Fountain of Fruit");
94 }
95
96 @AfterTest
97 public void breakdown(){
98     driver.quit();
99 }
100 }
```
- Toolbars and Menus:** File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, Window, Help, Maven
- Bottom Status Bar:** Tests passed: 8 (a minute ago), 24:36, CRLF, UTF-8, 4 spaces, Event Log

Let me describe all of these test cases one by one.

For verifying the content I created 2 test cases named verifyTitle() and verifyTextInApples().

- verifyTitle() checks if the title of the webapp on the home page is same. The original version's title is called “Foundation of Fruits”. This testcase is passed for the first version and failed for the second version of the application as the text is modified.
- verifyTextInApples() checks if a particular text (“ Worm Wiggle Into An Apple ”) is the same or not. This testcase is passed for the first version and failed for the second version of the application as the text is modified.

For checking the existence of html elements I created 2 test cases named verifyBananaImgExists() and verifyAppleImgExists().

- They check if the source of the image is same as the one in the original website.
- verifyAppleImgExists() is passed by both versions of the website
- verifyBananaImgExists() is not passed by second version of the website as the source of the image does not match the original source of the image.

For checking if a html element is modified, I developed a test case verifyBananaImgWidth() which checks if the width of the image is equal to 400.

- This test case is passed for original website and not for the second version of the website as I have changed the width to 600.

For checking if the navigation of the driver works well, I have developed 3 testcases verifyBananaImg(), applesBtnToBanana() and verifyNavigation()

- verifyBananaImg() checks if the image on home page redirects user to Bananas.html or not. This testcase is passed by both version of the web app.
- applesBtnToBanana() checks if the button on the Bananas.html page redirects the user to the Apples.html page or not. The original version passes this test case but the second version does not.
- verifyNavigation() checks if the user is able to navigate from index.html to Bananas.html to Apples.html. This testcase is passed for both versions of the web app.

In addition to these test cases I used a few methods and global declarations to set up and teardown the selenium web driver using the chrome webdriver.

Test results created by the tool after test execution:

For the developed test cases, for the original version of the web app all of the test cases are passed as shown in the DefaultSuite.

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help assign5 - SampleTest.java

C:\Users\bkanumar\AppData\Local\JetBrains\IdeaC2021.3\temp-testing-customsuite.xml

Project pom.xml (sample) SampleTest.java SampleTest

Run: SampleTest.verifyBananalImg SampleTest

Default Suite

- assign5
 - SampleTest
 - applesBtntoBanana
 - verifyAppleImgExists
 - verifyBananalImg
 - verifyBananalImgExists
 - verifyBananalImgWidth
 - verifyNavigation
 - verifyTextInApples
 - verifyTitle

Tests passed: 8 of 8 tests – 2 sec 234 ms

```
"C:\Program Files\Java\jdk-17.0.2\bin\java.exe" ...
SLF4J: No SLF4J providers were found.
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See https://www.slf4j.org/codes.html#noProviders for further details.
Starting ChromeDriver 107.0.5304.62 (1eec40d3a5764881c92085aaee66d25075c159aa-refs/branch-heads/5304@{#942}) on port 3770
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
https://www.realfoods.co.uk/ProductImagesID/16061\_1.jpg
Worm Wiggle Into An Apple

=====
Default Suite
Total tests run: 8, Passes: 8, Failures: 0, Skips: 0
=====

Process finished with exit code 0
```

Version Control Run TODO Problems Debug Terminal Build Dependencies

86:47 CRLF UTF-8 4 spaces Event Log

When the same test cases are run for the second version of the web application only 3 of the test cases are passed and 5 others are failed.

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help assign5 - SampleTest.java

C:\Users\bkanumar\AppData\Local\JetBrains\IdeaC2021.3\temp-testing-customsuite.xml

Project pom.xml (sample) SampleTest.java SampleTest

Run: SampleTest.verifyBananalImg SampleTest

Default Suite

- assign5
 - SampleTest
 - applesBtntoBanana
 - verifyAppleImgExists
 - verifyBananalImg
 - verifyBananalImgExists
 - verifyBananalImgWidth
 - verifyNavigation
 - verifyTextInApples
 - verifyTitle

Tests failed: 5, passed: 3 of 8 tests – 2 sec 841 ms

```
"C:\Program Files\Java\jdk-17.0.2\bin\java.exe" ...
SLF4J: No SLF4J providers were found.
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See https://www.slf4j.org/codes.html#noProviders for further details.
Starting ChromeDriver 107.0.5304.62 (1eec40d3a5764881c92085aaee66d25075c159aa-refs/branch-heads/5304@{#942}) on port 25656
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.

java.lang.AssertionError: |
  Expected :Apples.html
  Actual   :Bananas2.html
<Click to see difference>

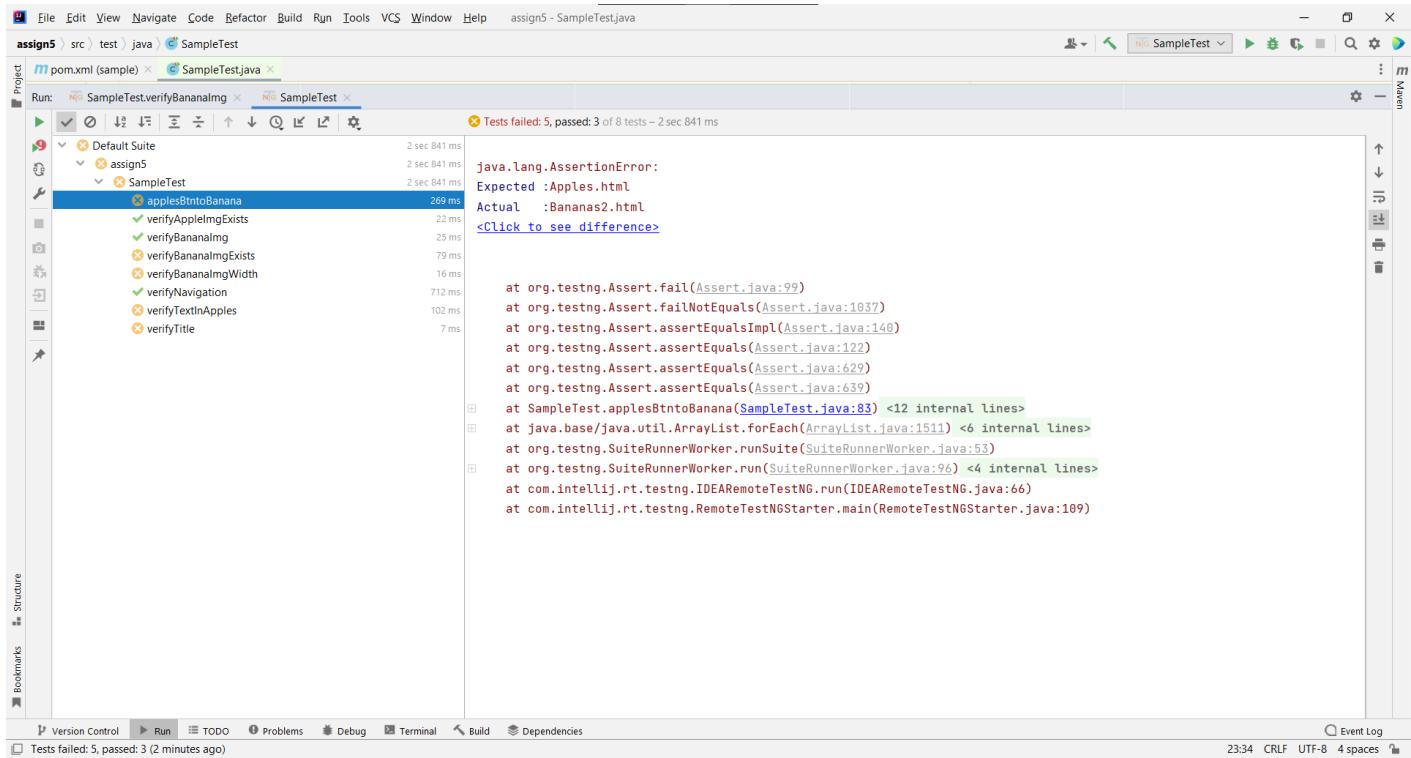
  at org.testng.Assert.fail(Assert.java:99)
  at org.testng.Assert.failNotEquals(Assert.java:1037)
  at org.testng.Assert.assertEqualsImpl(Assert.java:140)
  at org.testng.Assert.assertEquals(Assert.java:122)
  at org.testng.Assert.assertEquals(Assert.java:629)
  at org.testng.Assert.assertEquals(Assert.java:639)
  at SampleTest.applesBtntoBanana(SampleTest.java:83) <12 internal lines>
  at java.base/java.util.ArrayList.forEach(ArrayList.java:1511) <6 internal lines>
  at org.testng.SuiteRunnerWorker.runSuite(SuiteRunnerWorker.java:53)
  at org.testng.SuiteRunnerWorker.run(SuiteRunnerWorker.java:96) <4 internal lines>
  at com.intellij.rt.testng.IDEARemoteTestNG.run(IDEARemoteTestNG.java:66)
  at com.intellij.rt.testng.RemoteTestNGStarter.main(RemoteTestNGStarter.java:109)

https://www.realfoods.co.uk/ProductImagesID/16061\_1.jpg
```

Version Control Run TODO Problems Debug Terminal Build Dependencies

Tests failed: 5, passed: 3 (a minute ago) Event Log

The reason for these test cases failing in the second version of the website are as shown below:



The screenshot shows the IntelliJ IDEA interface during a TestNG test run. The top menu bar includes File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, VCS, Window, Help, and assign5 - SampleTest.java. The title bar shows "assign5 > src > test > java > SampleTest". The left sidebar has tabs for pom.xml (sample) and SampleTest.java, with the latter being the active tab. The main pane displays a test run summary: "Tests failed: 5, passed: 3 of 8 tests – 2 sec 841 ms". Below this, the "Default Suite" section is expanded, showing the "SampleTest" suite with its child test methods: applesBtntoBanana, verifyAppleImgExists, verifyBananaImg, verifyBananaImgWidth, verifyNavigation, verifyTextInApples, and verifyTitle. The "applesBtntoBanana" method is highlighted in blue, indicating it failed. A detailed error message follows:

```
java.lang.AssertionError:  
Expected :Apples.html  
Actual   :Bananas2.html  
<Click to see difference>  
  
at org.testng.Assert.fail(Assert.java:99)  
at org.testng.Assert.failNotEquals(Assert.java:1037)  
at org.testng.Assert.assertEqualsImpl(Assert.java:140)  
at org.testng.Assert.assertEquals(Assert.java:122)  
at org.testng.Assert.assertEquals(Assert.java:629)  
at org.testng.Assert.assertEquals(Assert.java:639)  
at SampleTest.applesBtntoBanana(SampleTest.java:83) <12 internal lines>  
at java.base/java.util.ArrayList.forEach(ArrayList.java:1511) <6 internal lines>  
at org.testng.SuiteRunnerWorker.runSuite(SuiteRunnerWorker.java:53)  
at org.testng.SuiteRunnerWorker.run(SuiteRunnerWorker.java:96) <4 internal lines>  
at com.intellij.rt.testng.IDEARemoteTestNG.run(IDEARemoteTestNG.java:66)  
at com.intellij.rt.testng.RemoteTestNGStarter.main(RemoteTestNGStarter.java:109)
```

The bottom status bar shows "Tests failed: 5, passed: 3 (2 minutes ago)" and the system time "23:34".

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help assign5 - SampleTest.java

Project pom.xml (sample) SampleTest.java SampleTest

Run: SampleTest.verifyBananalmg SampleTest

Default Suite

- assign5
- SampleTest
 - applesBntoBanana
 - verifyAppleImgExists
 - verifyBananalmg
 - verifyBananalmgExists
 - verifyBananalmgWidth
 - verifyNavigation
 - verifyTextInApples
 - verifyTitle

Tests failed: 5, passed: 3 of 8 tests – 2 sec 841 ms

java.lang.AssertionError:
Expected :<https://i.ytimg.com/vi/SpAHPXNhAwk/maxresdefault.jpg>
Actual :
[Click to see difference](#)

```
at org.testng.Assert.fail(Assert.java:99)
at org.testng.Assert.failNotEquals(Assert.java:1037)
at org.testng.Assert.assertEqualsImpl(Assert.java:140)
at org.testng.Assert.assertEquals(Assert.java:122)
at org.testng.Assert.assertEquals(Assert.java:629)
at org.testng.Assert.assertEquals(Assert.java:639)
at SampleTest.verifyBananalmgExists(SampleTest.java:52) <12 internal lines>
at java.util.ArrayList.forEach(ArrayList.java:1511) <6 internal lines>
at org.testng.SuiteRunnerWorker.runSuite(SuiteRunnerWorker.java:53)
at org.testng.SuiteRunnerWorker.run(SuiteRunnerWorker.java:96) <4 internal lines>
at com.intellij.rt.testng.IDEARemoteTestNG.run(IDEARemoteTestNG.java:66)
at com.intellij.rt.testng.RemoteTestNGStarter.main(RemoteTestNGStarter.java:109)
```

Version Control Run TODO Problems Debug Terminal Build Dependencies

Tests failed: 5, passed: 3 (3 minutes ago)

Event Log 57:17 CRLF UTF-8 4 spaces

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help assign5 - SampleTest.java

Project pom.xml (sample) SampleTest.java SampleTest

Run: SampleTest.verifyBananalmg SampleTest

Default Suite

- assign5
- SampleTest
 - applesBntoBanana
 - verifyAppleImgExists
 - verifyBananalmg
 - verifyBananalmgExists
 - verifyBananalmgWidth
 - verifyNavigation
 - verifyTextInApples
 - verifyTitle

Tests failed: 5, passed: 3 of 8 tests – 2 sec 841 ms

java.lang.AssertionError:
Expected :400
Actual :600
[Click to see difference](#)

```
at org.testng.Assert.fail(Assert.java:99)
at org.testng.Assert.failNotEquals(Assert.java:1037)
at org.testng.Assert.assertEqualsImpl(Assert.java:140)
at org.testng.Assert.assertEquals(Assert.java:122)
at org.testng.Assert.assertEquals(Assert.java:629)
at org.testng.Assert.assertEquals(Assert.java:639)
at SampleTest.verifyBananalmgWidth(SampleTest.java:67) <12 internal lines>
at java.util.ArrayList.forEach(ArrayList.java:1511) <6 internal lines>
at org.testng.SuiteRunnerWorker.runSuite(SuiteRunnerWorker.java:53)
at org.testng.SuiteRunnerWorker.run(SuiteRunnerWorker.java:96) <4 internal lines>
at com.intellij.rt.testng.IDEARemoteTestNG.run(IDEARemoteTestNG.java:66)
at com.intellij.rt.testng.RemoteTestNGStarter.main(RemoteTestNGStarter.java:109)
```

Version Control Run TODO Problems Debug Terminal Build Dependencies

Tests failed: 5, passed: 3 (3 minutes ago)

Event Log 57:17 CRLF UTF-8 4 spaces

assign5 - SampleTest.java

Run: SampleTest.verifyBananalmg

Project: assign5

SampleTest

Tests failed: 5, passed: 3 of 8 tests – 2 sec 841 ms

Worms Wiggle Into Many Apples

java.lang.AssertionError:
Expected :Worm Wiggle Into An Apple
Actual :Worms Wiggle Into Many Apples
[Click to see difference](#)

at org.testng.Assert.fail(Assert.java:99)
at org.testng.Assert.failNotEquals(Assert.java:1037)
at org.testng.Assert.assertEqualsImpl(Assert.java:140)
at org.testng.Assert.assertEquals(Assert.java:122)
at org.testng.Assert.assertEquals(Assert.java:629)
at org.testng.Assert.assertEquals(Assert.java:639)
at SampleTest.verifyTextInApples(SampleTest.java:42) <12 internal lines>
at java.base/java.util.ArrayList.forEach(ArrayList.java:1511) <6 internal lines>
at org.testng.SuiteRunnerWorker.runSuite(SuiteRunnerWorker.java:53)
at org.testng.SuiteRunnerWorker.run(SuiteRunnerWorker.java:96) <4 internal lines>
at com.intellij.rt.testng.IDEARemoteTestNG.run(IDEARemoteTestNG.java:66)
at com.intellij.rt.testng.RemoteTestNGStarter.main(RemoteTestNGStarter.java:109)

Version Control Run TODO Problems Debug Terminal Build Dependencies

Tests failed: 5, passed: 3 (3 minutes ago)

Event Log

assign5 - SampleTest.java

Run: SampleTest.verifyBananalmg

Project: assign5

SampleTest

Tests failed: 5, passed: 3 of 8 tests – 2 sec 841 ms

Fountain of Fruit

java.lang.AssertionError:
Expected :Fountain of Fruit
Actual :Fountain of Fruitsss
[Click to see difference](#)

at org.testng.Assert.fail(Assert.java:99)
at org.testng.Assert.failNotEquals(Assert.java:1037)
at org.testng.Assert.assertEqualsImpl(Assert.java:140)
at org.testng.Assert.assertEquals(Assert.java:122)
at org.testng.Assert.assertEquals(Assert.java:629)
at org.testng.Assert.assertEquals(Assert.java:639)
at SampleTest.verifyTitle(SampleTest.java:33) <12 internal lines>
at java.base/java.util.ArrayList.forEach(ArrayList.java:1511) <6 internal lines>
at org.testng.SuiteRunnerWorker.runSuite(SuiteRunnerWorker.java:53)
at org.testng.SuiteRunnerWorker.run(SuiteRunnerWorker.java:96) <4 internal lines>
at com.intellij.rt.testng.IDEARemoteTestNG.run(IDEARemoteTestNG.java:66)
at com.intellij.rt.testng.RemoteTestNGStarter.main(RemoteTestNGStarter.java:109)

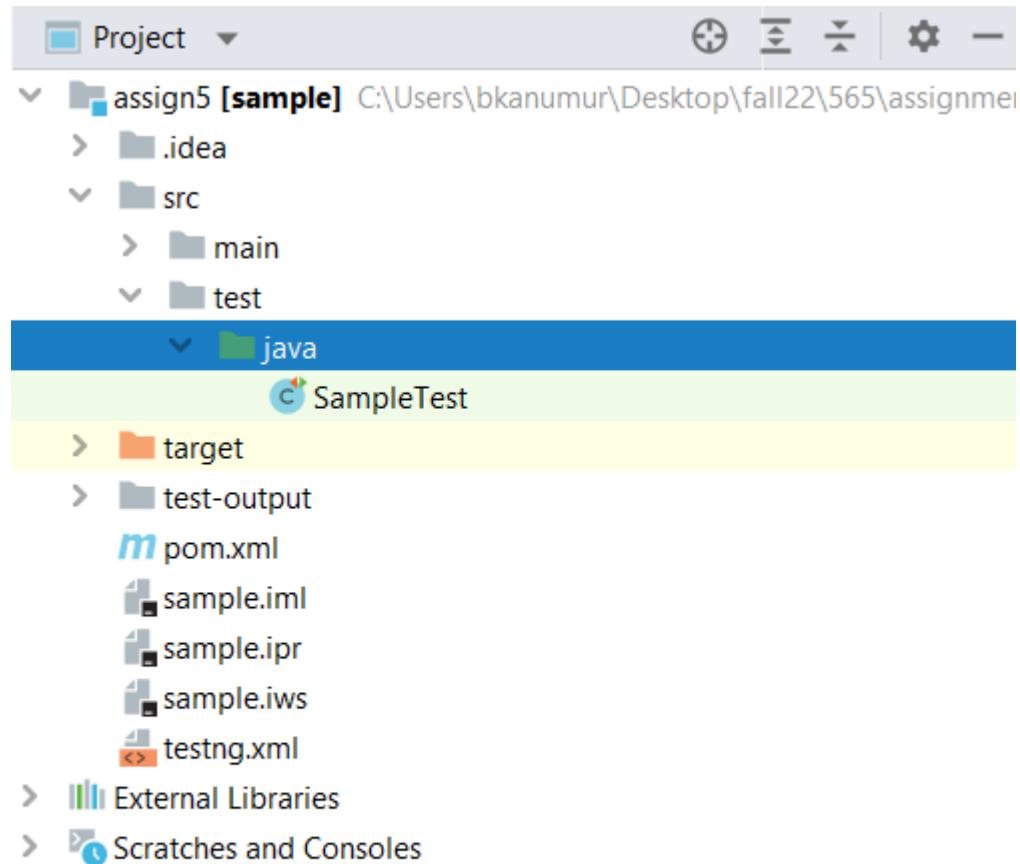
Version Control Run TODO Problems Debug Terminal Build Dependencies

Tests failed: 5, passed: 3 (3 minutes ago)

Event Log

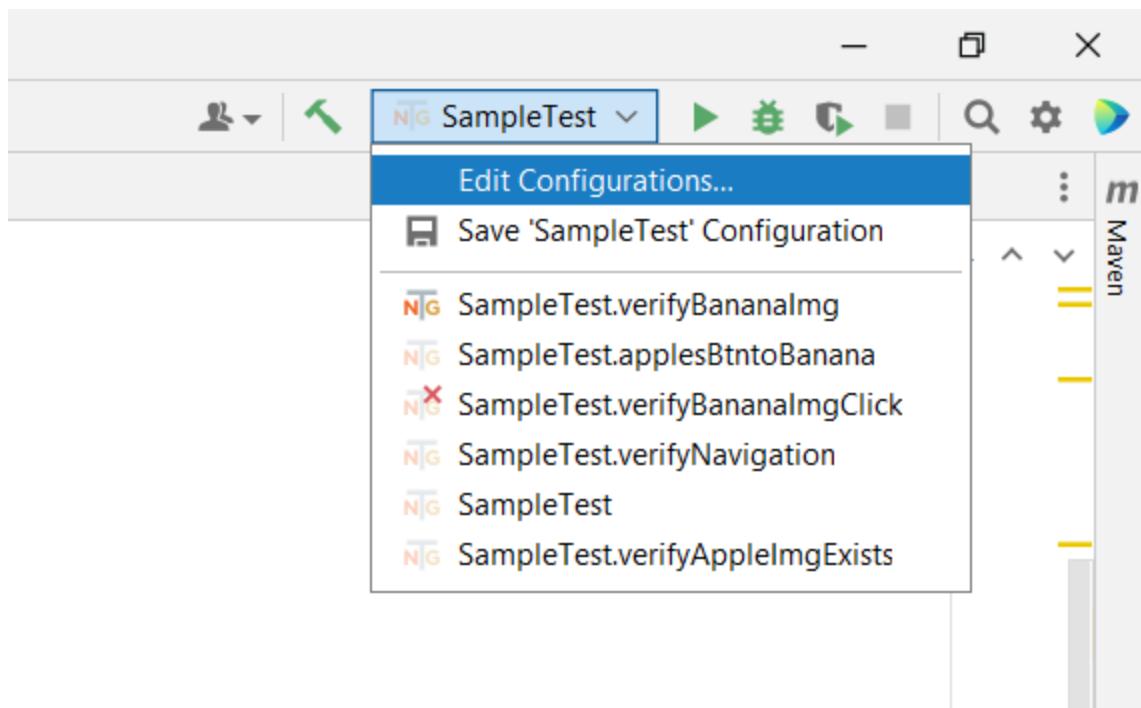
For generating an external report:

I have used the TestNG framework as it provides a robust framework for generating test reports. I have modified some settings on the IntelliJ IDE that I have used for this assignment, to generate these test reports to an external folder “test-output” as shown in the screenshot below:

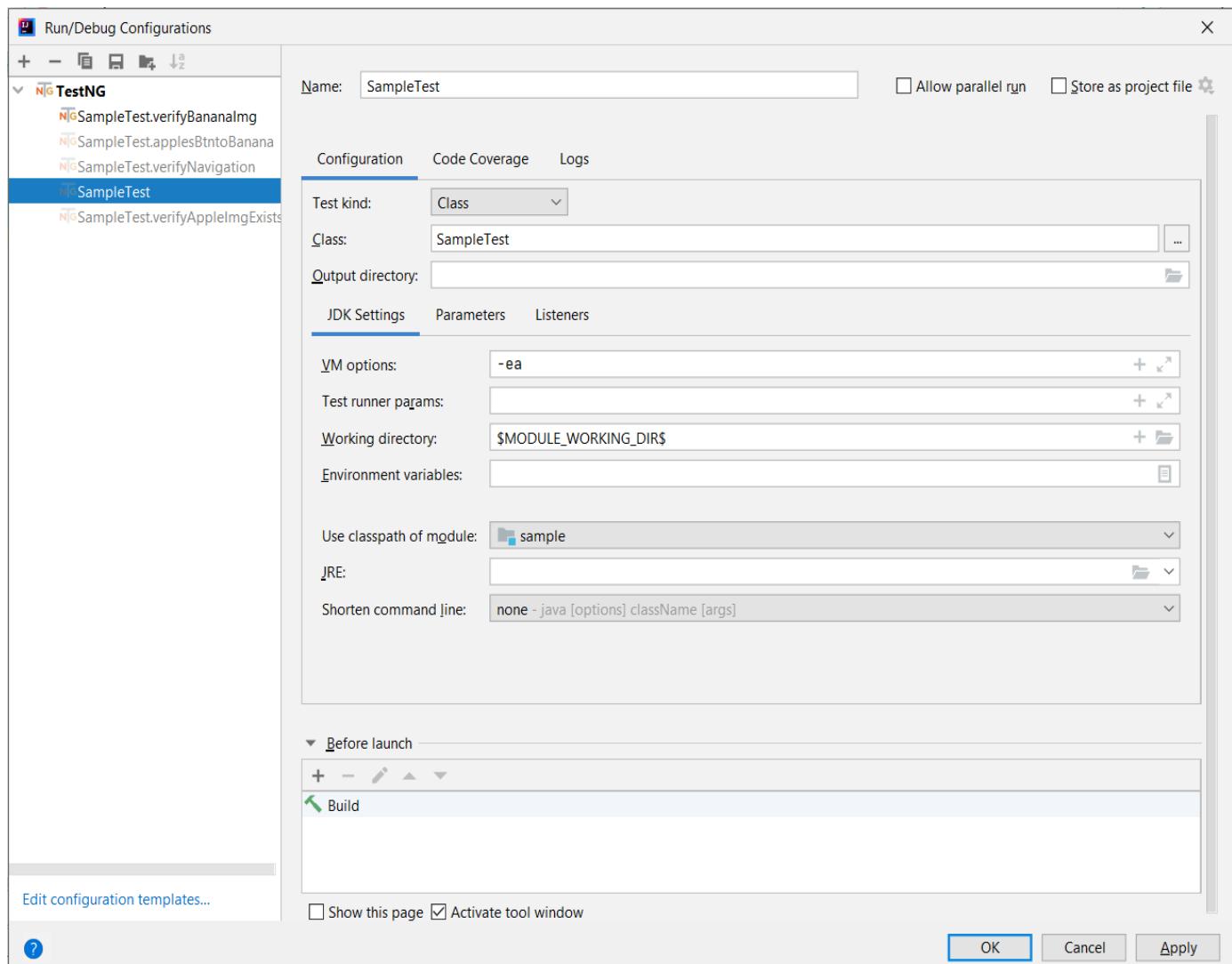


To change the settings I have edited the configuration of the testNG test class called SampleTest (which is the class where I have developed my test cases).

Select SampleTest > Edit Configurations

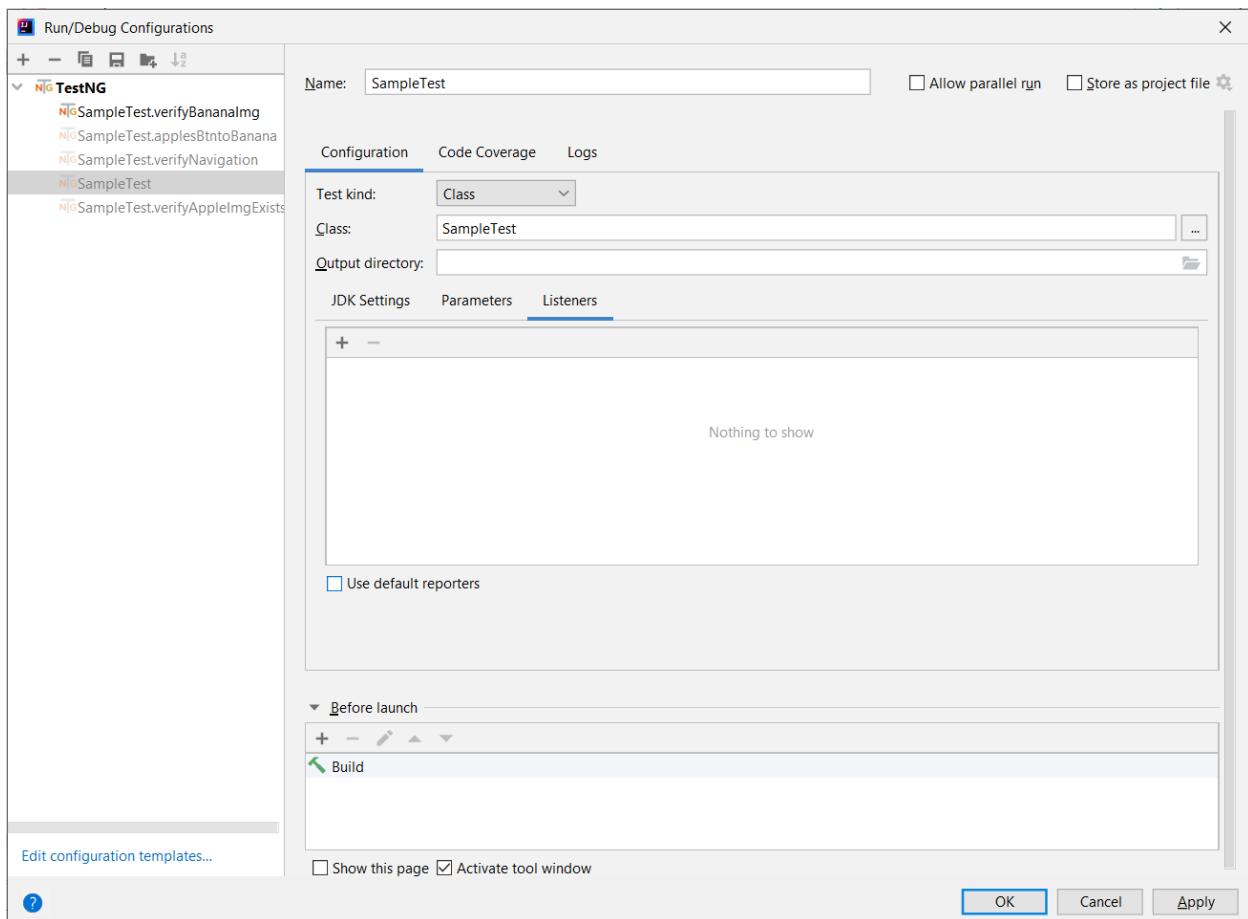


A dialog box will open:



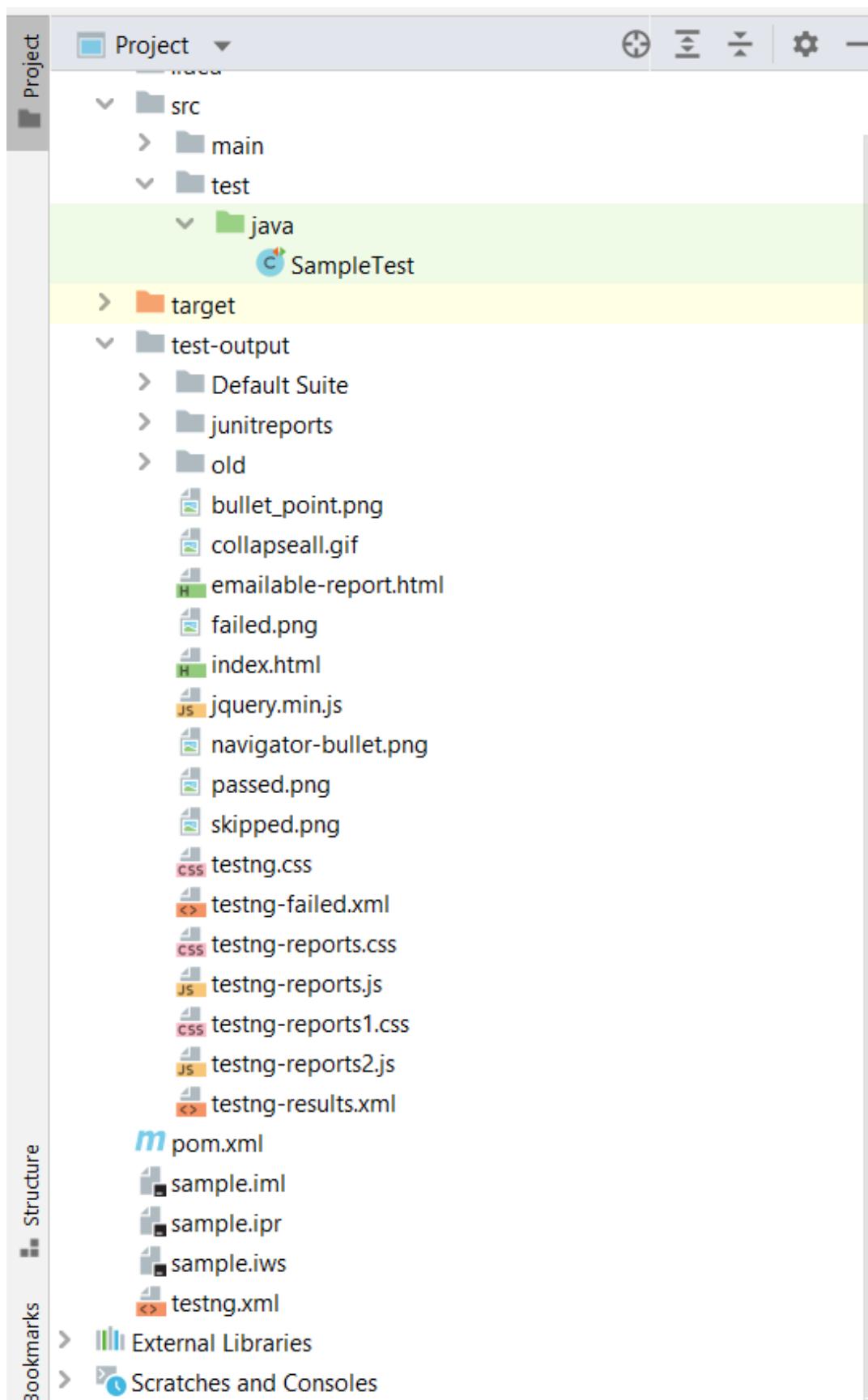
Under Configuration we have 3 options: JDK Settings, Parameters and Listeners.

Select Configuration > Listeners



Select the Use default reporters checkbox

Click Ok > Apply



You will notice a directory called test-output with all these corresponding files and directories. The test report that is generated is in a file named ‘emailable-report.html’

Generated Test Reports:

1. First version:

TestNG Report						
localhost:63342/assign5/sample/test-output/emailable-report.html?_jt=50s9i1o6d7jldfr953bl9mf8b&_ij_reload=RELOAD_ON_SAVE#summary						
Gmail YouTube Maps News Translate Login - Cox						
Test # Passed # Skipped # Retried # Failed Time (ms) Included Groups Excluded Groups						
Default Suite						
assign5 — passed						
Class Method Start Time (ms)						
Default Suite						
assign5 — passed						
SampleTest						
appleBtntoBanana 1669092021776 298						
verifyAppleImgExists 1669092022100 19						
verifyBananaImg 1669092022146 19						
verifyBananaImgWidth 1669092022189 75						
verifyNavigation 1669092022292 17						
verifyTextInApples 1669092022335 290						
verifyTitle 1669092022649 90						
verifyTitle 1669092022759 5						

assign5

SampleTest#appleBtntoBanana

[back to summary](#)

SampleTest#verifyAppleImgExists

[back to summary](#)

SampleTest#verifyBananaImg

[back to summary](#)

TestNG Report						
localhost:63342/assign5/sample/test-output/emailable-report.html?_jt=50s9i1o6d7jldfr953bl9mf8b&_ij_reload=RELOAD_ON_SAVE#summary						
Gmail YouTube Maps News Translate Login - Cox						
Test # Passed # Skipped # Retried # Failed Time (ms) Included Groups Excluded Groups						
Default Suite						
assign5 — passed						
SampleTest						
SampleTest#appleBtntoBanana						
SampleTest#verifyAppleImgExists						
SampleTest#verifyBananaImg						
SampleTest#verifyBananaImgExists						
SampleTest#verifyNavigation						
SampleTest#verifyTextInApples						
SampleTest#verifyTitle						

[back to summary](#)

The link to access the test report for the original version of the website is:
<http://localhost:63342/assign5/sample/test-output/emailable-report.html?ijt=6dp4u8c16ru5e5r4bfv5hr91em>

2. Second Version:

TestNG Report

localhost:63342/assign5/sample/test-output/emailable-report.html?ijt=6dp4u8c16ru5e5r4bfv5hr91em&ij_reload=RELOAD_ON_SAVE

Test	# Passed	# Skipped	# Retried	# Failed	Time (ms)	Included Groups	Excluded Groups
Default Suite							
assign5	3	0	0	5	3,601		
Class	Method	Start	Time (ms)				
Default Suite							
assign5 — failed							
SampleTest	applesBtntoBanana	1669093133908	841				
	verifyBananaImgExists	1669093134870	71				
	verifyBananaImgWidth	1669093134969	14				
	verifyTextInApples	1669093136070	84				
	verifyTitle	1669093136177	7				
assign5 — passed							
SampleTest	verifyBananaImgExists	1669093134781	19				
	verifyBananaImgWidth	1669093134825	21				
	verifyNavigation	1669093135010	1027				

assign5

SampleTest#applesBtntoBanana

Exception

```
java.lang.AssertionError: expected [Apples.html] but found [Bananas2.html]
  at SampleTest.appleBtntoBanana(SampleTest.java:83)
  at java.base/java.util.ArrayList.forEach(ArrayList.java:1511)
  at com.intellij.rt.testng.IDEARemoteTestNG.run(IDEARemoteTestNG.java:66)
  at com.intellij.rt.testng.RemoteTestNGStarter.main(RemoteTestNGStarter.java:109)
... Removed 30 stack frames
```

[back to summary](#)

SampleTest#verifyBananaImgExists

Exception

```
java.lang.AssertionError: expected [https://i.ytimg.com/vi/SpAHPXNhAuk/maxresdefault.jpg] but found []
  at SampleTest.verifyBananaImgExists(SampleTest.java:52)
  at java.base/java.util.ArrayList.forEach(ArrayList.java:1511)
  at com.intellij.rt.testng.IDEARemoteTestNG.run(IDEARemoteTestNG.java:66)
  at com.intellij.rt.testng.RemoteTestNGStarter.main(RemoteTestNGStarter.java:109)
... Removed 30 stack frames
```

[back to summary](#)

SampleTest#verifyBananaImgWidth

Exception

```
java.lang.AssertionError: expected [400] but found [600]
  at SampleTest.verifyBananaImgWidth(SampleTest.java:67)
  at java.base/java.util.ArrayList.forEach(ArrayList.java:1511)
  at com.intellij.rt.testng.IDEARemoteTestNG.run(IDEARemoteTestNG.java:66)
  at com.intellij.rt.testng.RemoteTestNGStarter.main(RemoteTestNGStarter.java:109)
... Removed 30 stack frames
```

[back to summary](#)

SampleTest#verifyTextInApples

Exception

```
java.lang.AssertionError: expected [Iwore Wiggled Into Apple] but found [Iwore Wiggled Into Many Apples]
  at SampleTest.verifyTextInApples(SampleTest.java:42)
  at java.base/java.util.ArrayList.forEach(ArrayList.java:1511)
  at com.intellij.rt.testng.IDEARemoteTestNG.run(IDEARemoteTestNG.java:66)
  at com.intellij.rt.testng.RemoteTestNGStarter.main(RemoteTestNGStarter.java:109)
... Removed 30 stack frames
```

[back to summary](#)

SampleTest#verifyTitle

Exception

```

SampleTest#verifyTextInApples
Exception
java.lang.AssertionError: expected [Worm Wiggle Into An Apple] but found [Worms Wiggle Into Many Apples]
at SampleTest.verifyTextInApples(SampleTest.java:42)
at java.base/java.util.ArrayList.forEach(ArrayList.java:1511)
at com.intellij.rt.testng.IDEARemoteTestNG.run(IDEARemoteTestNG.java:66)
at com.intellij.rt.testng.RemoteTestNGStarter.main(RemoteTestNGStarter.java:109)
... Removed 30 stack frames

back to summary

SampleTest#verifyTitle
Exception
java.lang.AssertionError: expected [Fountain of Fruit] but found [Fountain of Fruitssss]
at SampleTest.verifyTitle(SampleTest.java:33)
at java.base/java.util.ArrayList.forEach(ArrayList.java:1511)
at com.intellij.rt.testng.IDEARemoteTestNG.run(IDEARemoteTestNG.java:66)
at com.intellij.rt.testng.RemoteTestNGStarter.main(RemoteTestNGStarter.java:109)
... Removed 30 stack frames

back to summary

SampleTest#verifyAppleImgExists
back to summary

SampleTest#verifyBananaImg
back to summary

SampleTest#verifyNavigation
back to summary

```

The link to access the generated test report is:

[http://localhost:63342/assign5/sample/test-output/emailable-report.html? jt=fo0oj93bhof8q7gbdiaB00vtppg& ij_reload=RELOAD_ON_SAVE](http://localhost:63342/assign5/sample/test-output/emailable-report.html?jt=fo0oj93bhof8q7gbdiaB00vtppg&ij_reload=RELOAD_ON_SAVE)

Assessment of the tools:

Selenium WebDriver:

Although the WebDriver does not have the record and playback function, that is probably the only disadvantage I found in using the Selenium WebDriver vs. using the Selenium IDE. The Selenium WebDriver is very robust in its features and uses. A huge advantage is that since it doesn't have any core engine, it can be used natively by simply using a web browser. The API documentation for the WebDriver was very exhaustive and overwhelming. However the community support for this tool seemed pretty great as I found the correct resources without a lot of effort. The WebDriver also supports listeners so it checks for any kind of input from the user.

It is an open-source project and free of cost due to that reason. Since it supports a lot of different programming languages, it can be widely adapted by a lot of

developers who program in different languages. It also caters to different website browsers such as Internet Explorer, Google Chrome, Mozilla Firefox, etc. Hence users with different browsers can use it or developers can program and enable performing automation testing on users that use different types of browsers. It also supports different operating systems such as Windows, Mac, linux/unix.

Another huge advantage is that it can mimic a real user's experience by triggering mouse clicks, mouse scrolls, keyboard inputs, etc.

TestNG Framework:

One of the greatest advantages of using the framework was that it made generating the test reports very very easy. A html report was generated. The annotations that I used in the test script made it very easy for me to replicate a user's experience. It was a very 'light-to-use' application that enabled parallel testing. It allowed me to generate logs of the program. It was simple to integrate in the use of the framework. I had to just write the logic of the code and make use of the annotations to add structure to the test cases being run. The documentation provided for TestNG was also surprisingly lightweight and very comprehensive.

I found all the annotations that catered to my particular use just by reading the framework documentation. I would highly recommend using this framework alongside the selenium webdriver for automating any sort of web application testing.

References Used:

1. <https://www.selenium.dev/documentation/webdriver/>
2. https://www.selenium.dev/documentation/webdriver/getting_started/install_drivers/
3. <https://testng.org/doc/documentation-main.html>
4. <https://mvnrepository.com/>
5. <https://www.browserstack.com/guide/testng-framework-with-selenium-automation>
6. <https://www.browserstack.com/guide/testng-annotations-in-selenium>