

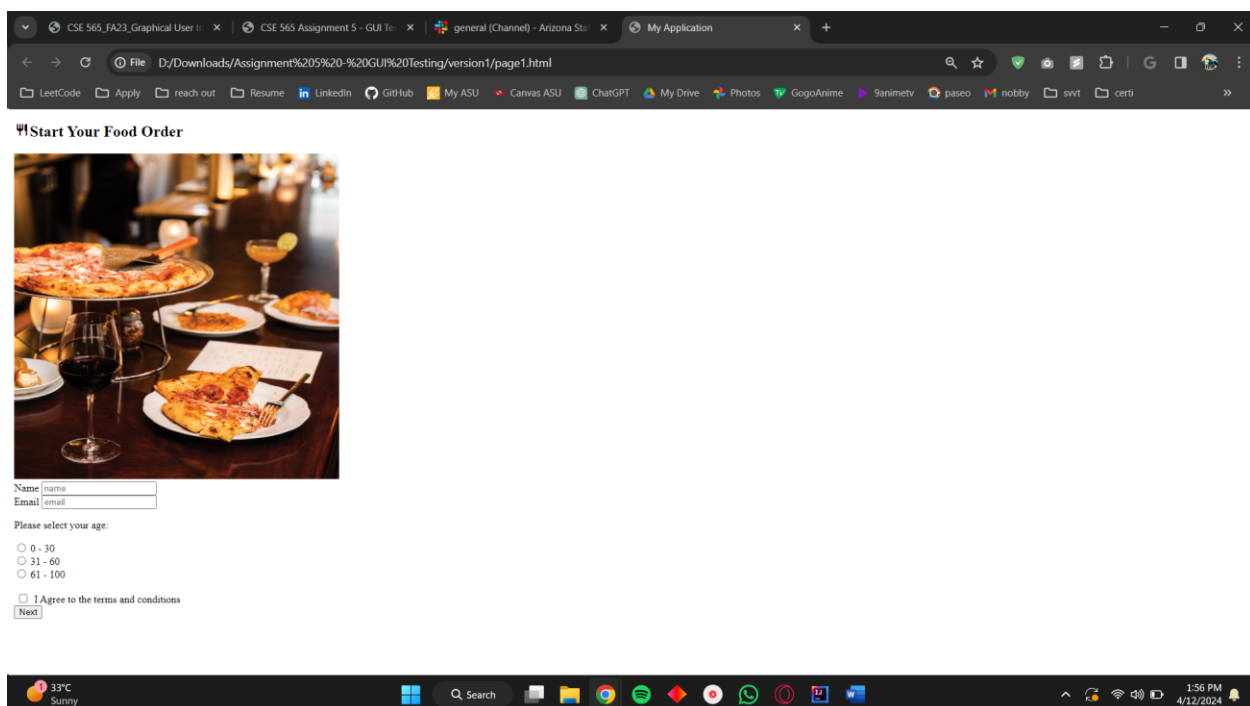
CSE 565: Software Verification and Validation

Graphical User Interface Testing Assignment

1. Description of the Application

A food ordering website has been created using HTML. It has 3 pages which are linked to one another. The following are the screenshots of the application:

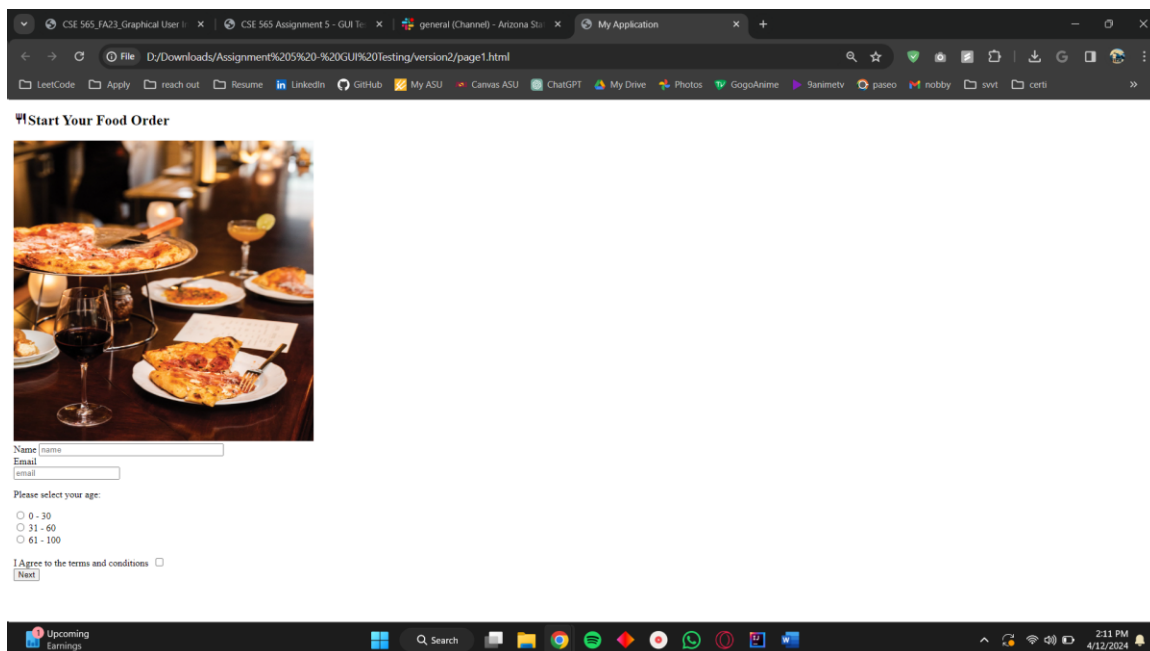
❖ Version 1, Page 1 –



This page contains:

- A heading
- An image
- A label and an input for name and email
- 3 radio buttons for age
- A label and checkbox for terms and conditions
- A button for next page, i.e., Page 2

❖ Version 2, Page 1 –

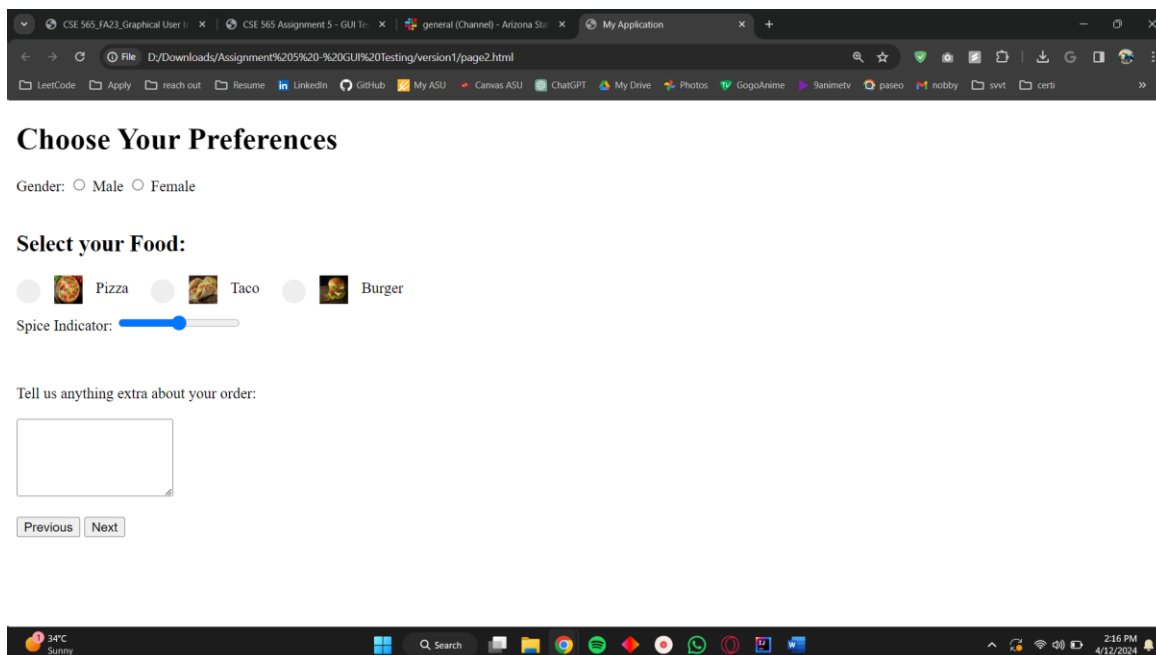


The screenshot shows a web browser window with the address bar displaying the file path: `D:\Downloads\Assignment%205%20-%20GUI%20Testing\version2\page1.html`. The browser tabs include 'CSE 565_FA23_Graphical User', 'CSE 565 Assignment 5 - GUI T...', 'general (Channel) - Arizona St...', and 'My Application'. The page title is 'Start Your Food Order'. The form contains a large image of a pizza on a table. Below the image, there are input fields for 'Name' (with a placeholder 'name') and 'Email' (with a placeholder 'email'). Below these fields, there is a section titled 'Please select your age:' with three radio button options: '0 - 30', '31 - 60', and '61 - 100'. At the bottom of the form, there is a checkbox labeled 'I Agree to the terms and conditions' and a 'Next' button. The Windows taskbar at the bottom shows the system clock as 2:11 PM on 4/12/2024.

Changes made:

- Wider input for name (size) – width set to 300 pixels
- Email input below its label (location) – just added a line break after its label
- T&C checkbox on the right of the label (location) – wrote label code before checkbox
- Additional: “Next” button now links to Page 3 (flow)

❖ Version 1, Page 2 –



The screenshot shows a web browser window with the address bar displaying the file path: `D:\Downloads\Assignment%205%20-%20GUI%20Testing\version1\page2.html`. The browser tabs include 'CSE 565_FA23_Graphical User', 'CSE 565 Assignment 5 - GUI T...', 'general (Channel) - Arizona St...', and 'My Application'. The page title is 'Choose Your Preferences'. The form contains a 'Gender:' label with two radio button options: 'Male' and 'Female'. Below this, there is a section titled 'Select your Food:' with three radio button options: 'Pizza', 'Taco', and 'Burger'. Below the food selection, there is a 'Spice Indicator:' label with a horizontal slider bar. At the bottom of the form, there is a text area labeled 'Tell us anything extra about your order:' and a 'Next' button. The Windows taskbar at the bottom shows the system clock as 2:16 PM on 4/12/2024.

This page contains:

- A heading
- “Male” and “Female” radio buttons for gender
- 3 radio buttons for the choice of food, along with images
- A slider for the spice level
- A label and textbox for extra notes for the food
- 2 buttons for “Previous” and “Next” pages, i.e., Page 1 & 3 respectively

❖ Version 2, Page 2 –

Choose Your Preferences

Gender: ☐ Male ☐ Female

Select your Food: ☐ Pizza ☐ Taco ☐ Burger

Spice Indicator:

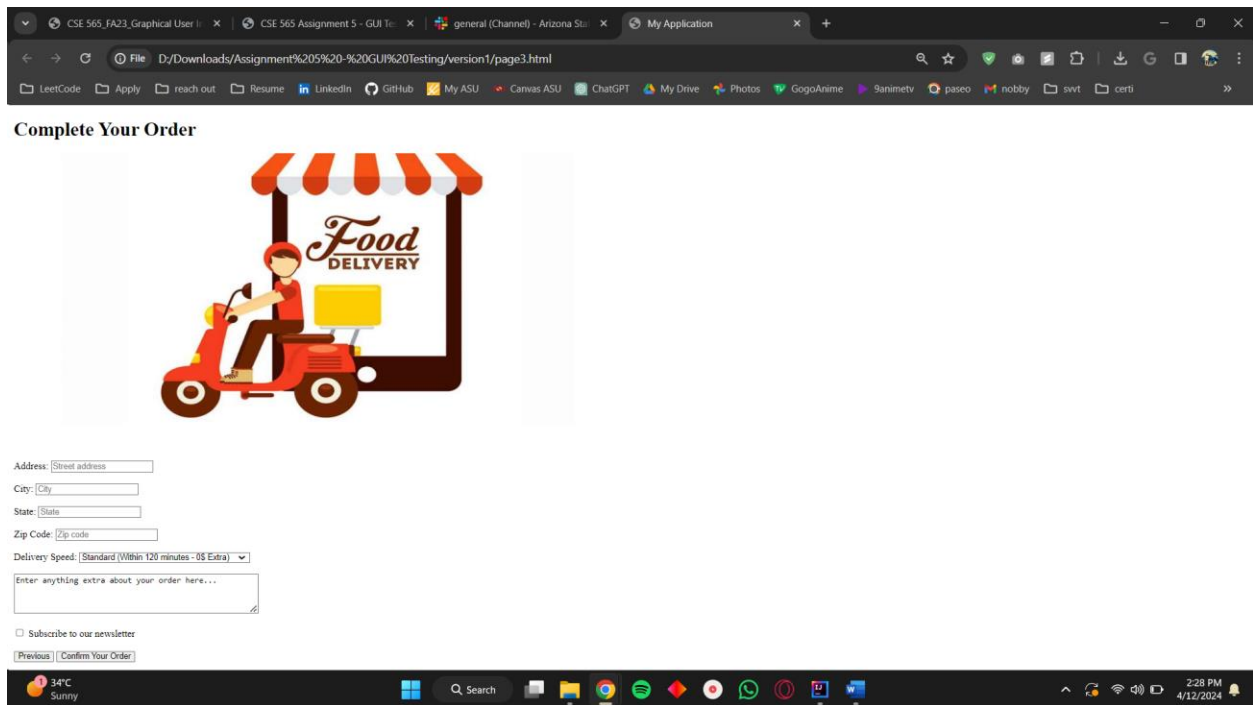
Tell us anything extra about your order:

[Previous](#) [Next](#)

Changes:

- The radio buttons for food are now aligned horizontally with the heading (location) – placed the radio buttons in a grid format
- Wider textbox for extra notes (size) – width set to 100%
- The previous and next buttons are now in the center (location) – text align set to center

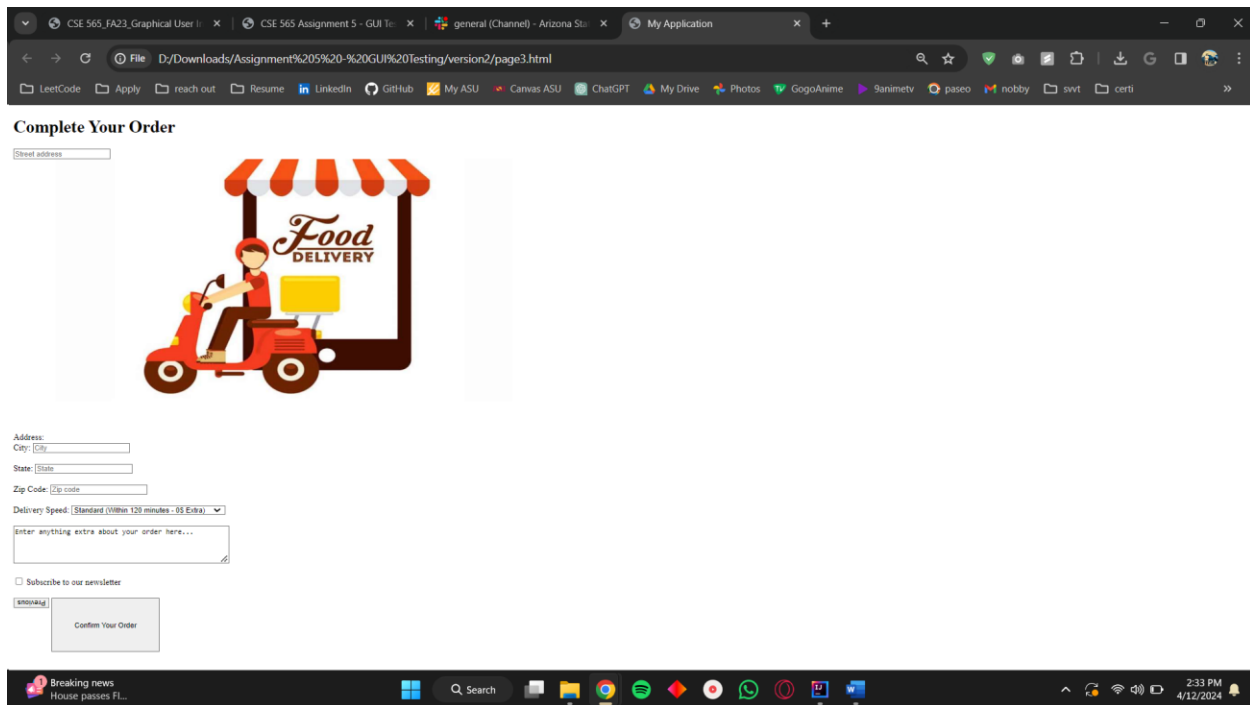
❖ Version 1, Page 3 –



This page contains:

- A heading
- An image
- Labels and inputs for Address, City, State, Zip
- A drop-down list for the delivery speed with 3 options – Standard, Fast, Super Fast
- A textbox for additional notes for the delivery
- A checkbox for the newsletter
- “Previous” button to go to Page 2
- “Confirm Your Order” button to go to the confirmation page, i.e., Page 4

❖ **Version 2, Page 3 –**



Changes:

- “Address” label is as it is, but the input is now above the image (location)
- “Previous” button is upside-down, i.e., rotated by 180 degrees (orientation)
- “Confirm Your Order” button is bigger (size) - height: 100px; width: 200px

2. Description of the Tool for GUI Testing

I have selected Selenium for the GUI Testing of my application. Selenium WebDriver stands out as a widely embraced open-source solution for automating web browsers, pivotal in streamlining interactions with web applications for testing and web scraping purposes. Leveraging a client-server architecture, it facilitates seamless communication with browsers across diverse operating systems.

❖ Key Features:

- Selenium WebDriver boasts a comprehensive suite of APIs, empowering users to automate an array of tasks such as interacting with web elements, managing windows and frames, capturing screenshots, and executing JavaScript.
- It offers support for multiple programming languages including Java, Python, C#, Ruby, and JavaScript.
- Enjoying compatibility with major browsers like Chrome, Firefox, Safari, Internet Explorer, and Edge, WebDriver ensures versatility in testing environments.

- With headless browser testing capabilities, it enables execution of tests sans a graphical user interface.

- Seamlessly integrates with popular test frameworks like JUnit, TestNG, and Cucumber.

❖ **Scope:**

- Selenium WebDriver finds extensive utility in web application testing, spanning functional, regression, integration, and acceptance testing, alongside facilitating cross-browser testing and web scraping endeavors.

❖ **Area of Usage:**

- Streamlining UI testing automation for web applications.

- Conducting comprehensive testing across different browsers and operating systems.

- Ensuring the resilience of existing functionalities through rigorous regression testing.

- Assessing application performance under varying loads to ensure scalability.

- Guaranteeing a seamless user experience across diverse browsers through cross-browser testing.

- Extracting valuable data from web pages through efficient web scraping techniques.

3. Description of the Test Cases

The test cases were developed in such a manner that the difference between the working of Version 1 and Version 2 is highlighted. The following are the test cases created:

Test Case	Description
-----------	-------------

Page 1

testPage1NameLocation()	Checks the name input location w.r.t its original location
testPage1NameSize()	Checks the name input size w.r.t its original size
testPage1EmailLocation()	Checks the email input location w.r.t its original location
testPage1EmailSize()	Checks the email input size w.r.t its original size
testPage1()	Tests the image displayed, input fields for name & email, all the radio buttons for age, and the T&C checkbox

testPage1TnCLocation()	Checks the T&C checkbox location w.r.t its original location
testPage1TnCSize()	Checks the T&C checkbox size w.r.t its original size
testPage1Flow()	Checks the flow from Page 1 to Page 2 on the press of the next button

Page 2

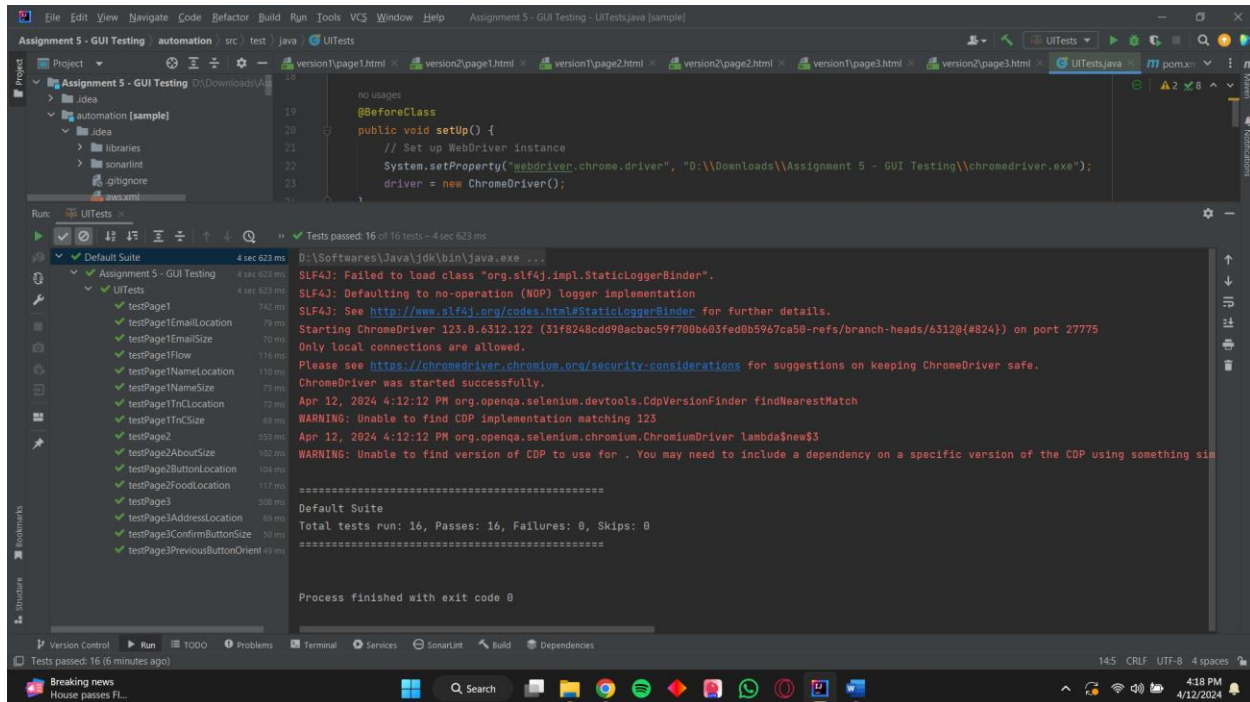
testPage2()	Tests the 2 gender radio buttons, 3 food selection radio buttons, the spice level slider, the previous button, the next button, and the extra notes textbox
testPage2FoodLocation()	Checks the locations of the 3 food radio buttons w.r.t their original locations
testPage2AboutSize()	Checks the size of the notes textbox w.r.t its original size
testPage2ButtonLocation()	Checks the locations of the next and previous buttons w.r.t their original locations

Page 3

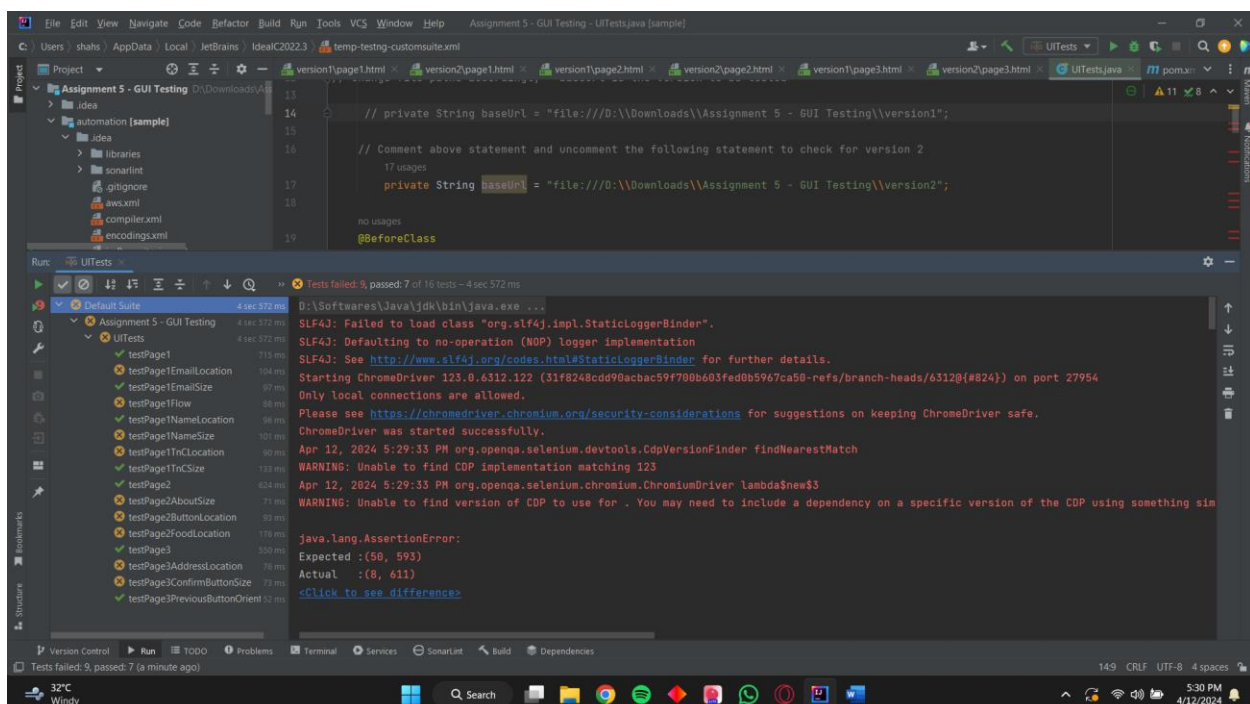
testPage3()	Tests the image displayed, the address/city/state/zip inputs, the delivery speed drop-down list, additional delivery instructions textbox, newsletter checkbox, the confirm button, and the previous button
testPage3AddressLocation()	Checks the address input location w.r.t its original location
testPage3ConfirmButtonSize()	Checks the size of the confirm button w.r.t its original size
testPage3PreviousButtonOrientation()	Checks the location of the previous button w.r.t its original location (there is no other way to check the orientation in Selenium – will explain more about this test case in the next section)

4. Execution of the Test Cases

Version 1 passed all the 16 tests. Hence, we successfully tested the existence of all the GUI elements, correctness of their content/size/location, and correctness of the link (flow from one page to another). Here is the screenshot of the results:



For Version 2, let's see how many test cases it passes.



Version 2 passed just 7 cases and failed 9. Let's first see the tests it passed:

1) testPage1NameLocation()

It passed this test because there was no change in the location of the name input.

2) testPage1EmailSize()

It passed this test because there was no change in the size of the email input.

3) testPage1()

It passed this test because there was no change in the image, the sample name and email inputs were checked correctly, and the 3 age radio buttons and the T&C checkbox were clicked properly.

4) testPage1TnCSize()

It passed this test because there was no change in the size of the T&C checkbox.

5) testPage2()

It passed this test because the gender and food selection radio buttons were clicked properly, the slider sample value and the textbox content were tested correct, and the previous and next buttons correctly linked to Page 1 and Page 3 respectively.

6) testPage3()

It passed this test because the image was right, the sample inputs for address/city/state/zip were tested correct, the delivery drop-down list was clicked and opened correctly, the textbox sample text was correct, the subscribe checkbox was checked correctly, the confirm button was correctly linked with the confirmation page, and finally the previous button correctly linked with Page 2.

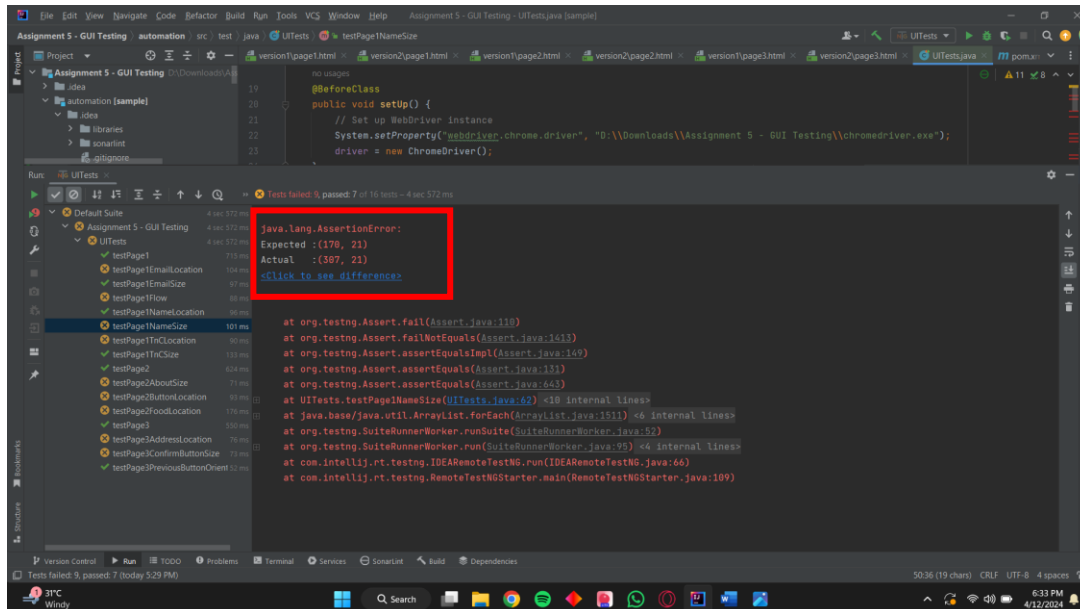
7) **testPage3PreviousButtonOrientation()

This is a crucial test case for this assignment. In Version 2, I rotated the previous button on Page 3 by 180 degrees, without changing its size or location. But since there is no way to test the orientation of a GUI element on Selenium, it passed this test case. This case just tests the location (x and y coordinates) of the button and because rotating a rectangle button just flips it and the text upside-down, there is no change in the coordinates and it passes the test. This demonstrates the drawback of the Selenium GUI testing tool.

Now, let's move onto the test cases it failed:

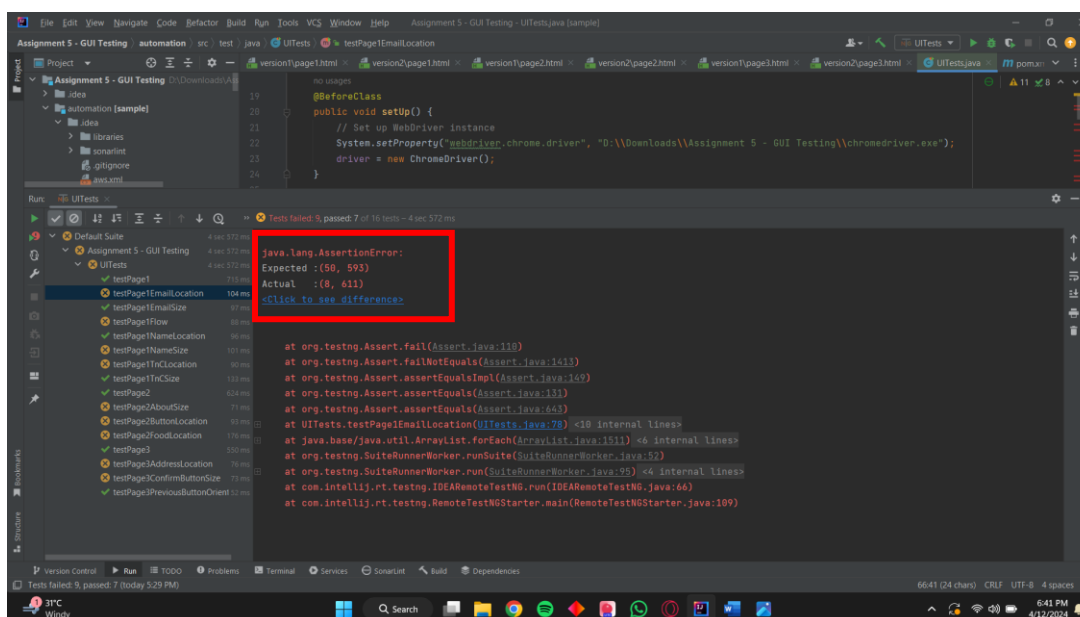
1) testPage1NameSize()

Since we changed the width of the name input to 300px, it failed the size check. The screenshot below shows the expected and actual dimensions.



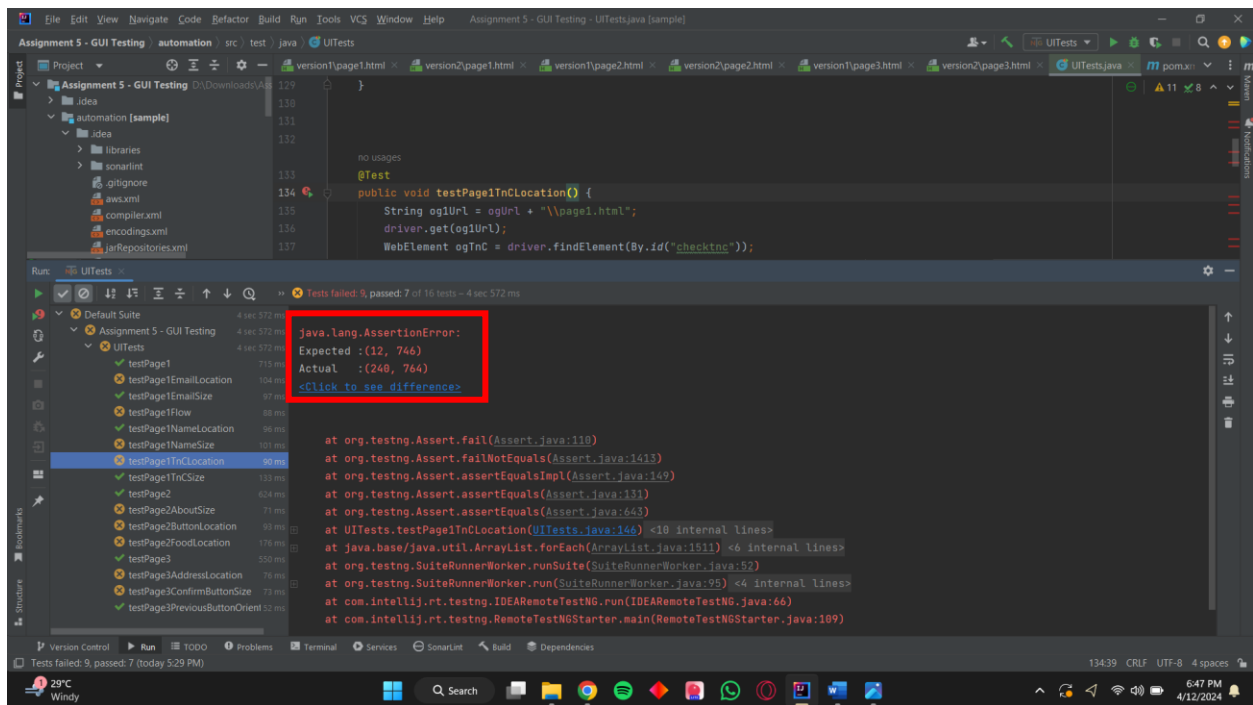
2) testPage1EmailLocation()

Since we added a line break between the email label and input, its location changed and hence it failed the test. See the expected and actual coordinates below.



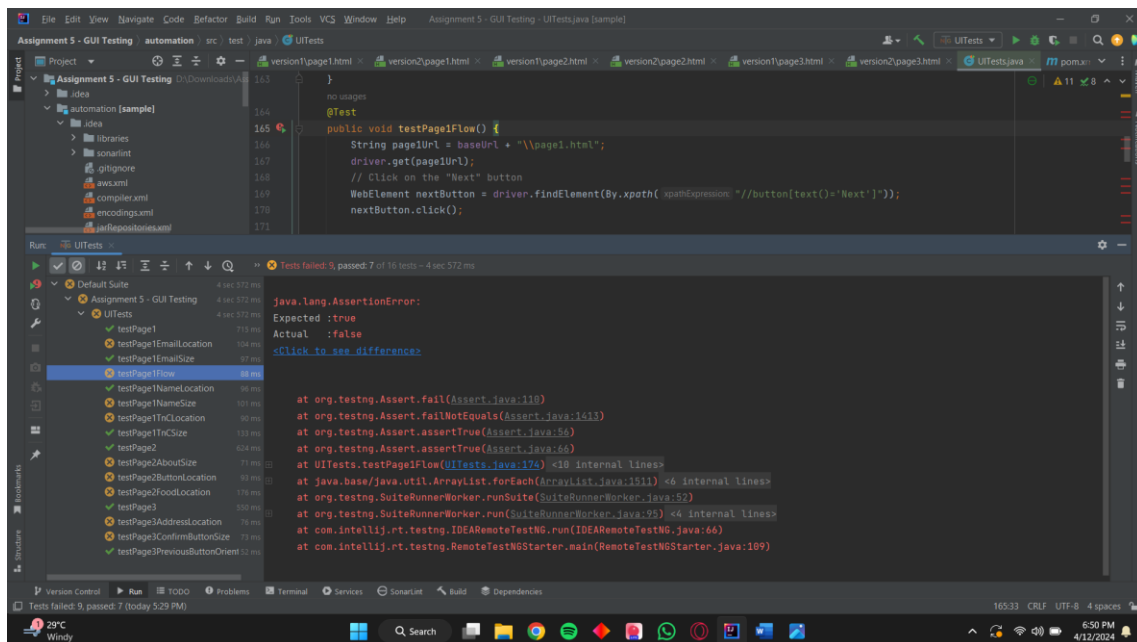
3) testPage1TnCLocation()

Since we wrote the statement for the T&C Label before the checkbox, its location changed.



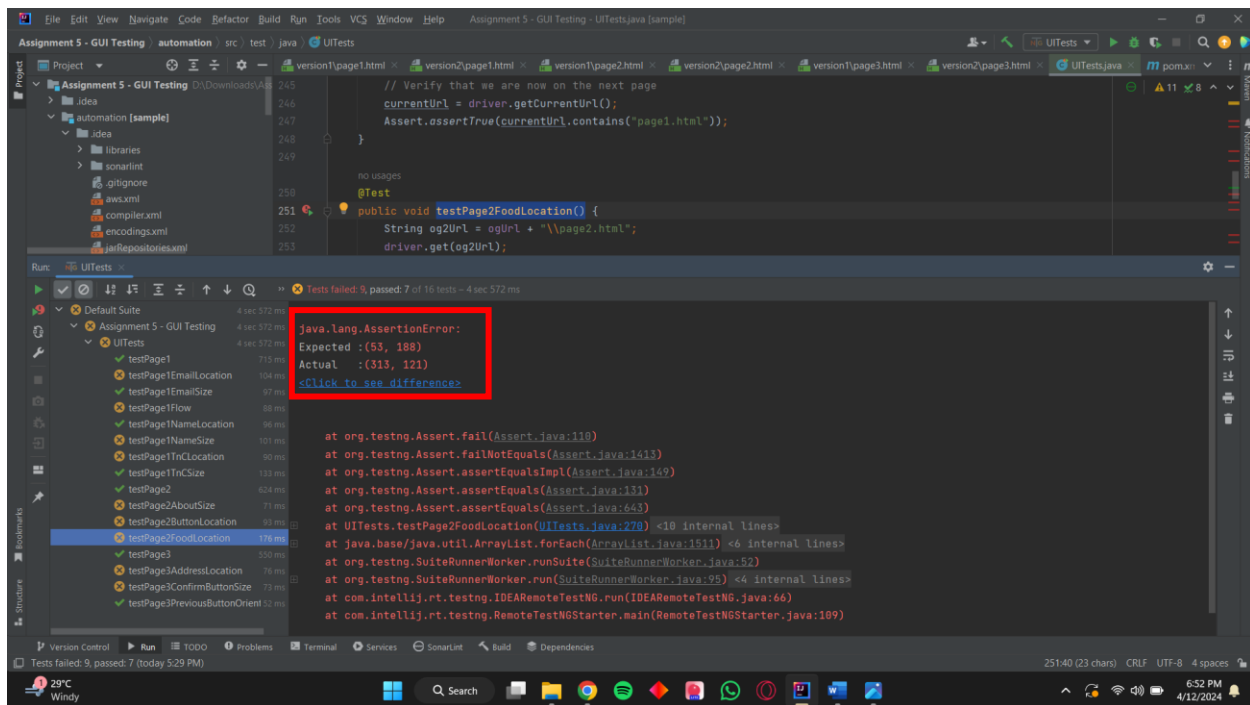
4) testPage1Flow()

Since we changed the link of the next button to Page 3 instead of Page 2, it failed this flow test case.



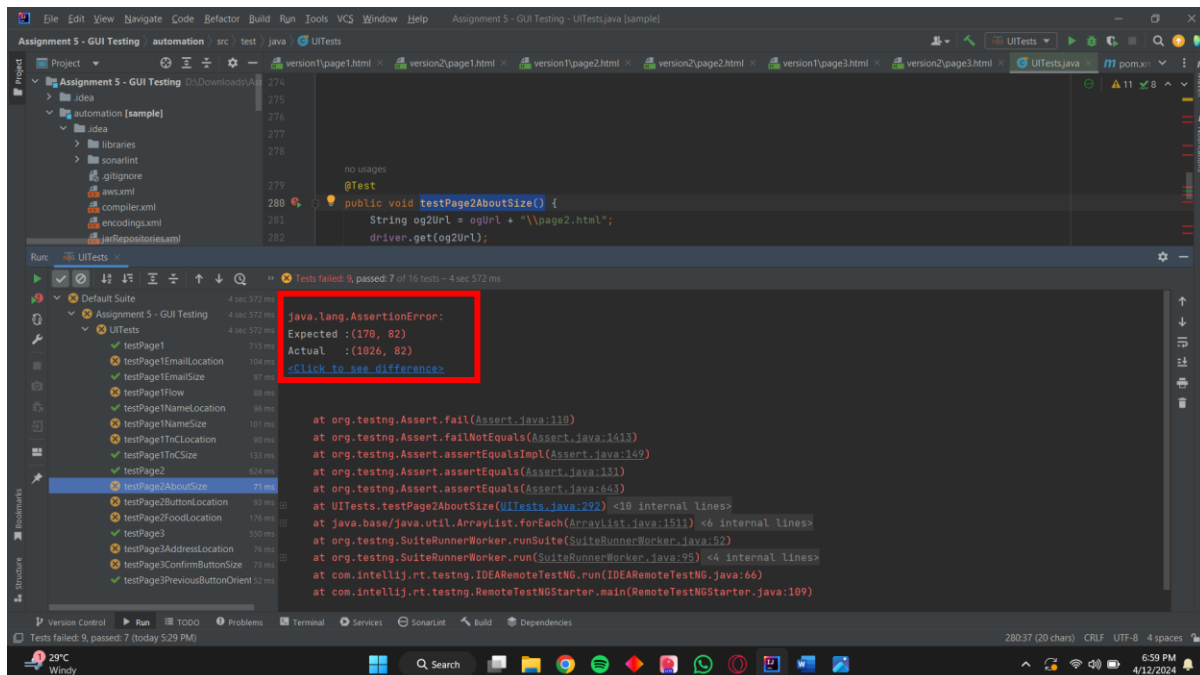
5) testPage2FoodLocation()

Since we changed the location of the 3 food selection radio buttons, it failed this test.



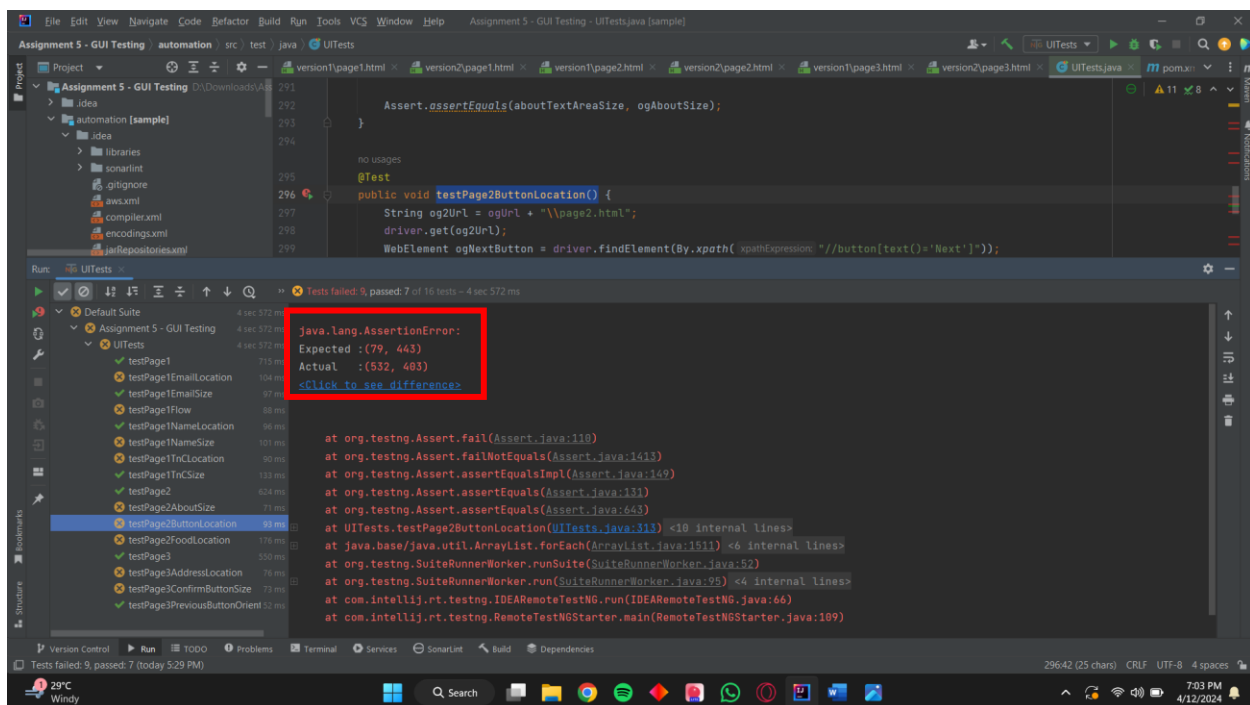
6) testPage2AboutSize()

Since we increased the size of the textbox on Page 2, it failed this test.



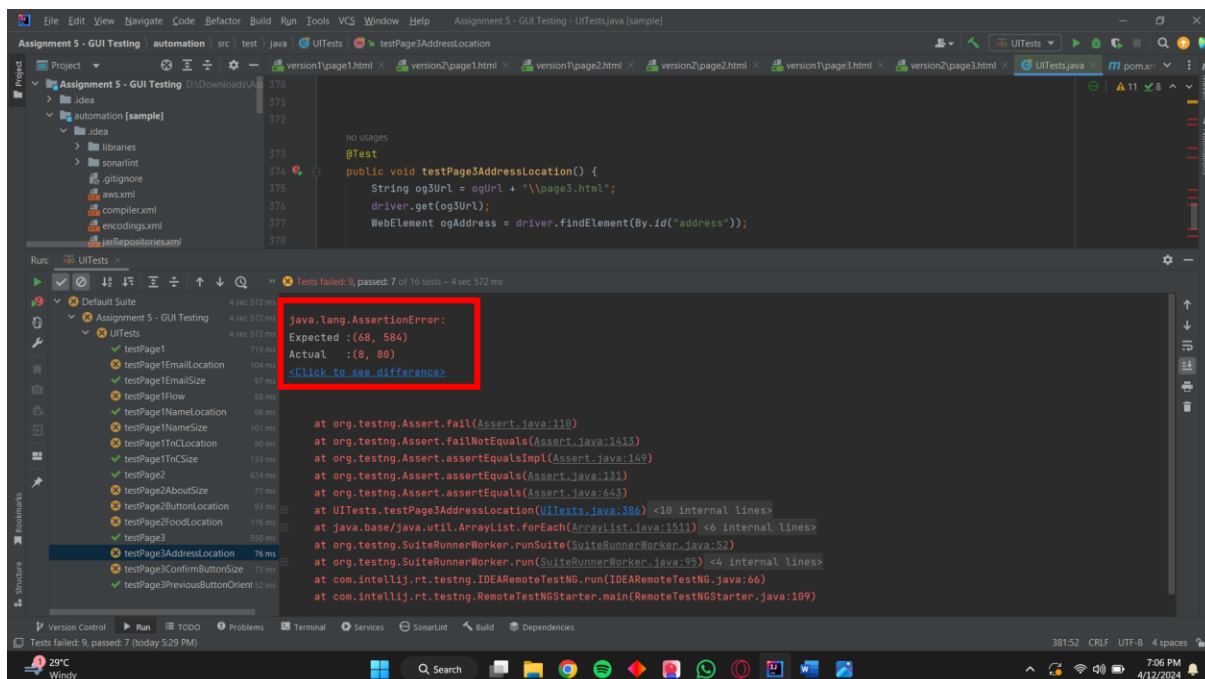
7) testPage2ButtonLocation()

Since we center-aligned the next and previous buttons on Page 2, it failed the location test.



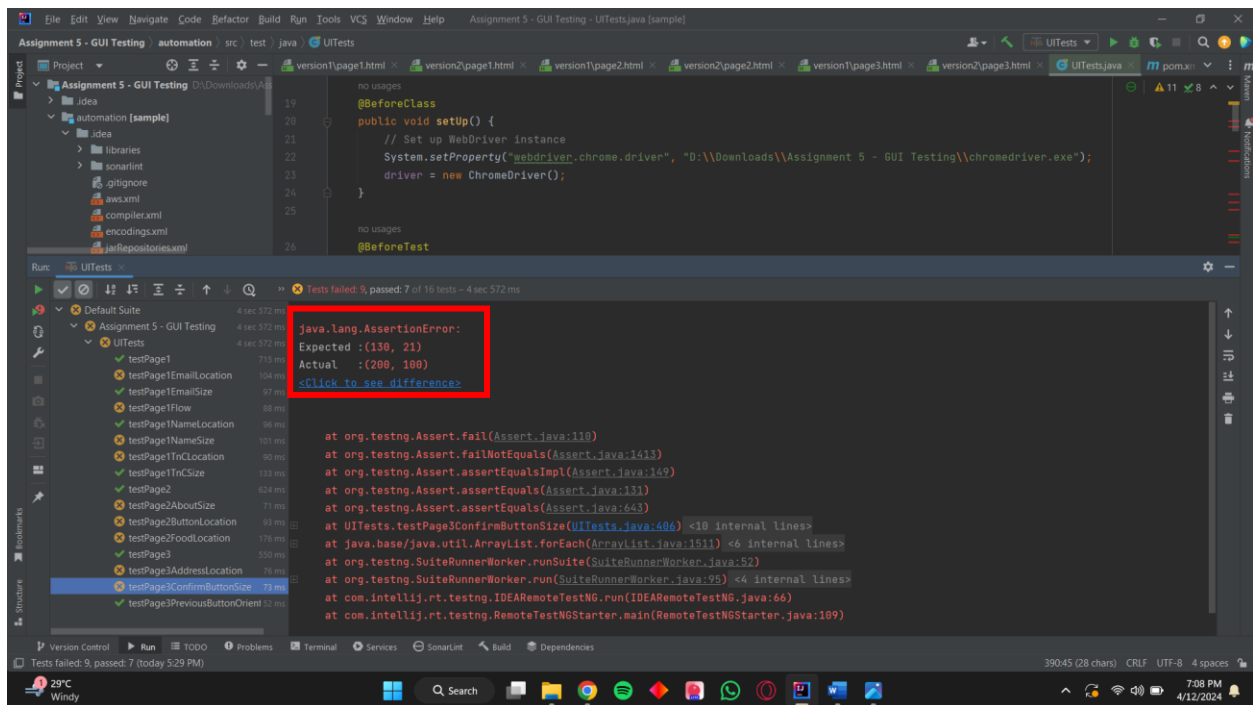
8) testPage3AddressLocation()

Since we placed the address input above its label and image on Page 3, it failed this test.



9) testPage3ConfirmButtonSize()

Since we enlarged the confirm button on Page 3, it failed the test.



I have used TestNG for report generation. It stands as a Java testing framework primarily employed for unit testing. It furnishes functionalities like annotations, grouping, parameterization, and assertions, facilitating streamlined test case creation and management. Furthermore, TestNG yields comprehensive test reports, instrumental in issue identification and progress monitoring.

Here are the reports generated in HTML format:

❖ Version 1 –

Test	Methods Passed	Scenarios Passed	# skipped	# failed	Total Time	Included Groups	Excluded Groups
Assignment 5 - GUI Testing	16	16	0	0	4.8 seconds		

Class	Method	# of Scenarios	Start	Time (ms)
Assignment 5 - GUI Testing — passed				
UITests	testPage1TnSize	1	1712987536406	100
	testPage1EmailLocation	1	1712987535775	95
	testPage1TnCLocation	1	1712987536296	110
	testPage2	1	1712987536507	558
	testPage3FoodLocation	1	1712987537162	105
	testPage3PreviousButtonOrientation	1	1712987538166	74
	testPage3AddressLocation	1	1712987538080	86
	testPage1NameLocation	1	1712987536091	103
	testPage3NameSize	1	1712987536196	100
	testPage3	1	1712987537406	674
	testPage1EmailSize	1	1712987535871	101
	testPage2FoodLocation	1	1712987537268	138
	testPage3PreviousButtonOrientation	1	1712987538241	70
	testPage2AboutSize	1	1712987537066	95
	testPage1	1	1712987535053	721
	testPage1Flow	1	1712987535972	119

Assignment 5 - GUI Testing

UITests:testPage1TnSize

[back to summary](#)

UITests:testPage1EmailLocation

[back to summary](#)

UITests:testPage1TnCLocation

[back to summary](#)

UITests:testPage2

[back to summary](#)

UITests:testPage2ButtonLocation

[back to summary](#)

UITests:testPage3ConfirmButtonSize

[back to summary](#)

UITests:testPage3AddressLocation

[back to summary](#)

UITests:testPage1NameLocation

[back to summary](#)

UITests:testPage1NameSize

[back to summary](#)

UITests:testPage3

[back to summary](#)

UITests:testPage1EmailSize

[back to summary](#)

UITests:testPage2FoodLocation

[back to summary](#)

UITests:testPage3PreviousButtonOrientation

[back to summary](#)

UITests:testPage2AboutSize

[back to summary](#)

UITests:testPage1

[back to summary](#)

UITests:testPage1Flow

[back to summary](#)

❖ Version 2 –

Test	Methods Passed	Scenarios Passed	# skipped	# failed	Total Time	Included Groups	Excluded Groups
Assignment 5 - GUI Testing	7	7	0	0	5.0 seconds		

Class	Method	# of Scenarios	Start	Time (ms)
Assignment 5 - GUI Testing --- failed				
UITests	testPage1TnLocation	1	1712986842378	132
	testPage1Duration	1	1712986841216	118
	testPage1AboutButtonSize	1	1712986842007	68
	testPage1Page3Location	1	1712986841335	167
	testPage1AboutSize	1	1712986842147	99
	testPage2PreviousLocation	1	1712986842247	130
	testPage1AddressLocation	1	1712986843014	72
	testPage1Flow	1	1712986844067	169
	testPage1EmailLocation	1	1712986844064	99
Assignment 5 - GUI Testing --- passed				
UITests	testPage2	1	1712986843095	738
	testPage2Page2Size	1	1712986841655	112
	testPage2ConfirmButtonSize	1	1712986840788	129
	testPage2PreviousButtonOrientation	1	1712986843152	67
	testPage2ConfirmButtonLocation	1	1712986841086	136
	testPage2	1	1712986842711	903
	testPage2	1	1712986841268	774
	testPage2	1	1712986841268	774

Assignment 5 - GUI Testing

UITests:testPage1TnLocation

```
java.lang.AssertionError: expected [[12, 764]] but found [[240, 764]]
    at UITests.testPage1TnLocation(UITests.java:146)
    at java.base/java.util.ArrayList.forEach(ArrayList.java:1511)
    at com.intellij.rt.testing.IDE6RemoteTestNG.run(IDE6RemoteTestNG.java:66)
    at com.intellij.rt.testing.RemoteTestNGStarter.main(RemoteTestNGStarter.java:109)
... Removed 27 stack frames
```

[back to summary](#)

UITests:testPage2FoodLocation

```
java.lang.AssertionError: expected [[63, 188]] but found [[313, 123]]
    at UITests.testPage2FoodLocation(UITests.java:270)
    at java.base/java.util.ArrayList.forEach(ArrayList.java:1511)
    at com.intellij.rt.testing.IDE6RemoteTestNG.run(IDE6RemoteTestNG.java:66)
    at com.intellij.rt.testing.RemoteTestNGStarter.main(RemoteTestNGStarter.java:109)
... Removed 27 stack frames
```

[back to summary](#)

UITests:testPage2AboutSize

```
java.lang.AssertionError: expected [[170, 82]] but found [[1826, 82]]
    at UITests.testPage2AboutSize(UITests.java:192)
    at java.base/java.util.ArrayList.forEach(ArrayList.java:1511)
    at com.intellij.rt.testing.IDE6RemoteTestNG.run(IDE6RemoteTestNG.java:66)
    at com.intellij.rt.testing.RemoteTestNGStarter.main(RemoteTestNGStarter.java:109)
... Removed 27 stack frames
```

[back to summary](#)

UITests:testPage2ButtonLocation

```
java.lang.AssertionError: expected [[79, 443]] but found [[532, 403]]
    at UITests.testPage2ButtonLocation(UITests.java:131)
    at java.base/java.util.ArrayList.forEach(ArrayList.java:1511)
    at com.intellij.rt.testing.IDE6RemoteTestNG.run(IDE6RemoteTestNG.java:66)
    at com.intellij.rt.testing.RemoteTestNGStarter.main(RemoteTestNGStarter.java:109)
... Removed 27 stack frames
```

[back to summary](#)

UITests:testPage3AddressLocation

```
java.lang.AssertionError: expected [[68, 584]] but found [[8, 80]]
    at UITests.testPage3AddressLocation(UITests.java:186)
    at java.base/java.util.ArrayList.forEach(ArrayList.java:1511)
    at com.intellij.rt.testing.IDE6RemoteTestNG.run(IDE6RemoteTestNG.java:66)
    at com.intellij.rt.testing.RemoteTestNGStarter.main(RemoteTestNGStarter.java:109)
... Removed 27 stack frames
```

[back to summary](#)

UITests:testPage1NameSize

```
java.lang.AssertionError: expected [[170, 21]] but found [[107, 21]]
    at UITests.testPage1NameSize(UITests.java:62)
    at java.base/java.util.ArrayList.forEach(ArrayList.java:1511)
    at com.intellij.rt.testing.IDE6RemoteTestNG.run(IDE6RemoteTestNG.java:66)
    at com.intellij.rt.testing.RemoteTestNGStarter.main(RemoteTestNGStarter.java:109)
... Removed 27 stack frames
```

[back to summary](#)

UITests:testPage1Flow

```
java.lang.AssertionError: expected [true] but found [false]
    at UITests.testPage1Flow(UITests.java:134)
    at java.base/java.util.ArrayList.forEach(ArrayList.java:1511)
    at com.intellij.rt.testing.IDE6RemoteTestNG.run(IDE6RemoteTestNG.java:66)
    at com.intellij.rt.testing.RemoteTestNGStarter.main(RemoteTestNGStarter.java:109)
... Removed 26 stack frames
```

[back to summary](#)

UITests:testPage1EmailLocation

```
java.lang.AssertionError: expected [[10, 593]] but found [[8, 611]]
    at UITests.testPage1EmailLocation(UITests.java:79)
    at java.base/java.util.ArrayList.forEach(ArrayList.java:1511)
    at com.intellij.rt.testing.IDE6RemoteTestNG.run(IDE6RemoteTestNG.java:66)
    at com.intellij.rt.testing.RemoteTestNGStarter.main(RemoteTestNGStarter.java:109)
... Removed 27 stack frames
```

[back to summary](#)

UITests:testPage3ConfirmButtonSize

```
java.lang.AssertionError: expected [[130, 21]] but found [[200, 100]]
    at UITests.testPage3ConfirmButtonSize(UITests.java:486)
    at java.base/java.util.ArrayList.forEach(ArrayList.java:1511)
    at com.intellij.rt.testing.IDE6RemoteTestNG.run(IDE6RemoteTestNG.java:66)
    at com.intellij.rt.testing.RemoteTestNGStarter.main(RemoteTestNGStarter.java:109)
... Removed 27 stack frames
```

[back to summary](#)

UITests:testPage1NameLocation

[back to summary](#)

UITests:testPage1

[back to summary](#)

UITests:testPage1TnCSize

[back to summary](#)

UITests:testPage1EmailSize

[back to summary](#)

UITests:testPage3PreviousButtonOrientation

[back to summary](#)

UITests:testPage3

[back to summary](#)

UITests:testPage2

[back to summary](#)

5. Assessment of the Tool

❖ Features and Functionalities:

Selenium offers a comprehensive array of tools for Java-based automated testing, enabling interaction with diverse browsers and mimicking user behaviors such as button clicks and form submissions. It accommodates various test frameworks like JUnit and TestNG, extends support to mobile app testing, and facilitates cross-platform web app automation.

❖ Type of Coverage:

Selenium empowers developers to automate functional, regression, and acceptance testing for web applications, covering end-to-end, UI, and cross-browser testing. It ensures application quality across functionality, compatibility, and usability aspects.

❖ Test Case Reusability:

Selenium enables test case reuse, streamlining testing efforts by allowing WebDriver-based test cases to execute seamlessly across different browsers and platforms. It also integrates with multiple testing frameworks for cross-project test case reuse.

❖ Test Results Produced:

Selenium furnishes detailed test results aiding issue identification and resolution. However, looking at our case, it has one issue of not being able to determine the correctness of orientation of symmetric GUI elements, as it has got no built-in functions to check that. Moving on, the tool can also generate results in HTML, XML, and JSON formats, facilitating seamless integration with CI/CD pipelines. Selenium also offers diverse test reporting tools such as TestNG Reports and Allure Reports for comprehensive result visualization.

❖ Ease of Usage:

Selenium boasts user-friendly automation script creation in Java via Selenium WebDriver, backed by well-documented APIs and support for various programming languages. Its compatibility with testing frameworks like JUnit and TestNG ensures smooth integration into existing testing workflows.

❖ GUI Elements Testing:

Selenium accommodates the testing of an extensive range of GUI elements such as buttons, links, forms, tables, and dropdowns. It extends support to advanced UI components like iframes, canvas, and HTML5 features, while also enabling mobile app testing, enhancing its versatility in automation testing.

TestNG stands out as a prevalent testing framework utilized in Java applications, boasting an array of functionalities including annotations, data-driven testing, parallel execution, and reporting. It simplifies the process of writing and executing unit, integration, and functional tests. Through its annotations, TestNG facilitates a structured approach to test cases and suites, offering multiple options for parallel test execution at the method, class, and suite levels. Additionally, TestNG excels in its reporting capabilities, allowing for the generation of diverse report formats like HTML and XML. In essence, TestNG emerges as a robust and adaptable framework, empowering developers to streamline and automate their testing procedures effectively.

6. References

- <https://www.w3schools.com/tags/>
- https://www.youtube.com/watch?v=mxDG4rzfEOI&ab_channel=HousseemDellai
- <https://www.selenium.dev/documentation/en/>
- <https://www.selenium.dev/selenium/docs/api/java/index.html>
- <https://www.automatetheplanet.com/test-automation-frameworks-selenium-webdriver>
- <https://www.softwaretestinghelp.com/web-elements-locators-in-selenium-webdriver/>
- <https://www.softwaretestinghelp.com/selenium-webdriver-tutorial-for-beginners/>