# Project #3 Graphical User Interface Testing Project

# Online Dealership Website

Fangjie Guo

Arizona State University

Course number: CSE 565 - Software Verification and Validation

Date: 2023.7

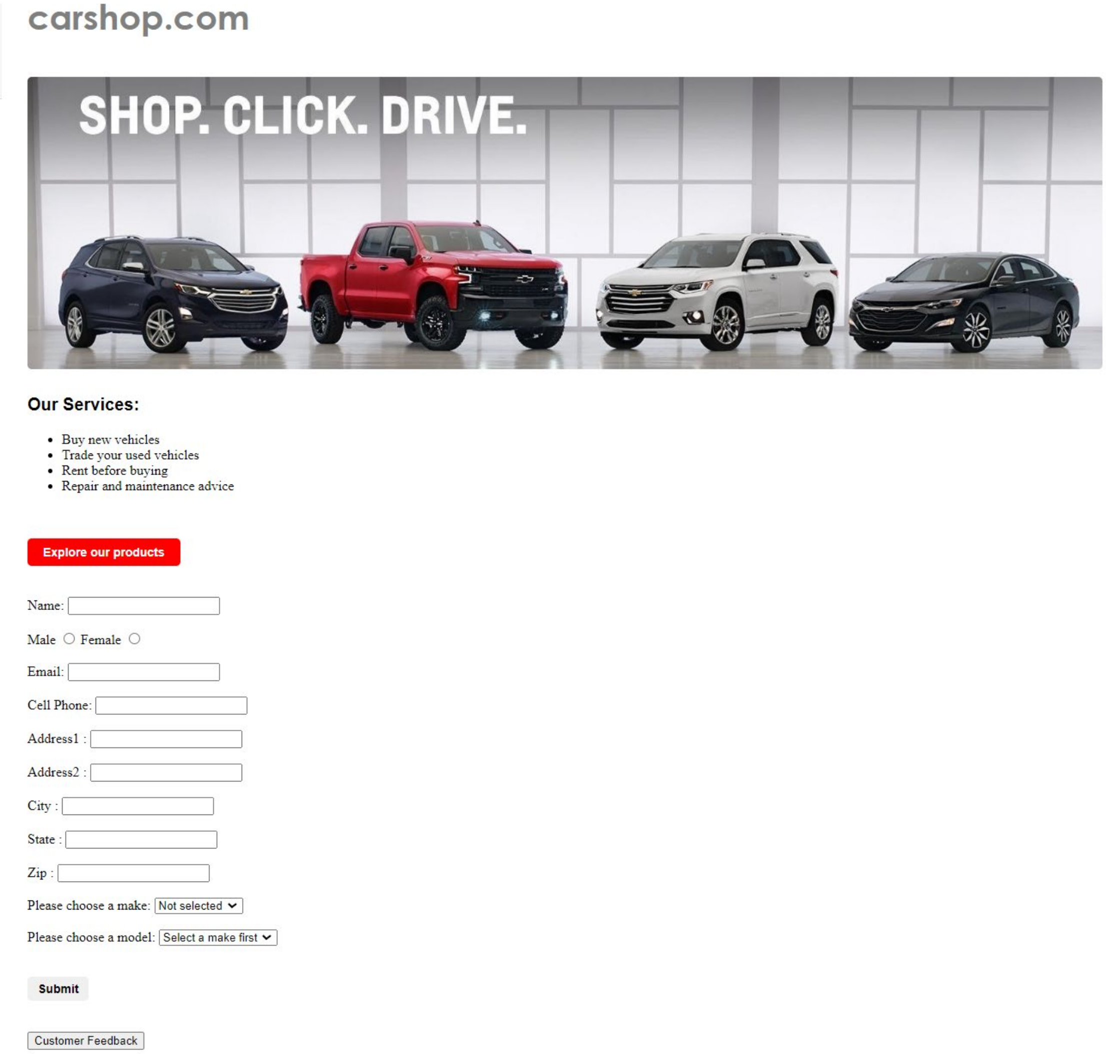
Graphical User Interface Testing Project

## Introduction of Application

The Online Dealership Website is a web-based platform designed to provide an immersive and user-friendly experience for customers seeking to buy or sell vehicles online. The website aims to streamline the process of browsing and purchasing vehicles while offering sellers a platform to showcase their inventory. It has 3 pages: Main Page, Car Images/Listing, Customer Feedback. This section provides an overview of the Online Dealership Website, including descriptions of each page and its main functions/elements.

### Main Page

The Main Page (Figure 1:  Main Page, v1). of the Online Dealership Website serves as the landing page, featuring a stylish and intuitive graphical user interface (GUI). It includes a prominent header displaying the website name, "carshop.com," and a captivating main image. The page provides an overview of the services offered through a concise bullet-point list. It includes interactive elements such as a button to explore products and a customer information form with textboxes and radio buttons for gender selection. To facilitate car selection, the GUI features dropdown menus for choosing the car make and model. Upon selection, the second dropdown dynamically populates with relevant models. A "Submit" button allows users to submit their information, and a separate "Customer Feedback" button encourages user engagement.



**Our Services:**

- Buy new vehicles
- Trade your used vehicles
- Rent before buying
- Repair and maintenance advice

Name:

Male ○ Female ○

Email:

Cell Phone:

Address1 :

Address2 :

City :

State :

Zip :

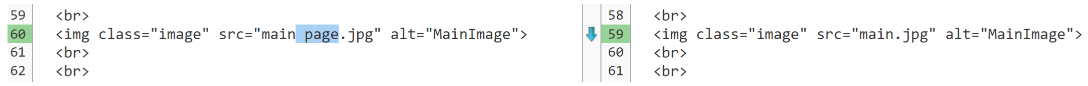
Please choose a make: Not selected

Please choose a model: Select a make first

*Figure 1. Main Page, v1*

The version 2 Main Page (Figure 2:  Main Page, v2) has the following changes:

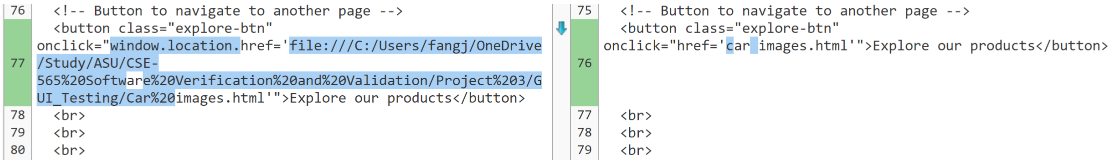
- The head images source name has been changed.



```
<br>
<img class="image" src="main page.jpg" alt="MainImage">
<br>
<br>
```

```
<br>
<img class="image" src="main.jpg" alt="MainImage">
<br>
<br>
```

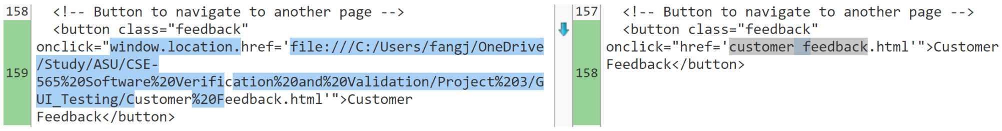
- Under “Explorer our products” button, the onclick reference link has been changed.



```
  <!-- Button to navigate to another page -->
  <button class="explore-btn"
onclick="window.location.href='file:///C:/Users/fangj/OneDrive/Study/ASU/CSE-565%20Software%20Verification%20and%20Validation/Project%203/GUI_Testing/Car%20images.html'">Explore our products</button>
  <br>
  <br>
  <br>
```

```
  <!-- Button to navigate to another page -->
  <button class="explore-btn"
onclick="href='car images.html'">Explore our products</button>
  <br>
  <br>
  <br>
```

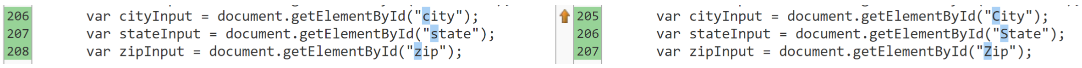
- The Navigation button “Custom Feedback” link address has been changed.



```
  <!-- Button to navigate to another page -->
  <button class="feedback"
onclick="window.location.href='file:///C:/Users/fangj/OneDrive/Study/ASU/CSE-565%20Software%20Verification%20and%20Validation/Project%203/GUI_Testing/Customer%20Feedback.html'">Customer Feedback</button>
```

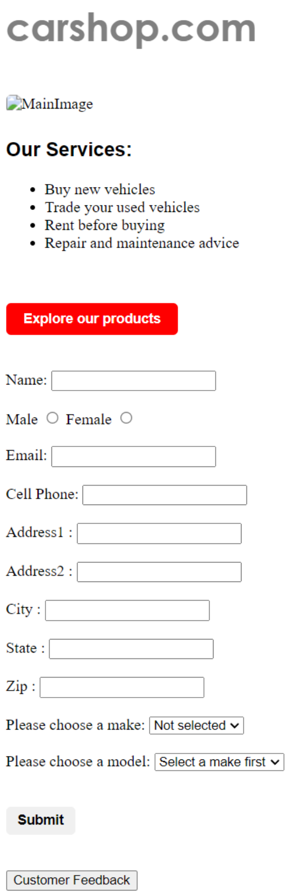
```
  <!-- Button to navigate to another page -->
  <button class="feedback"
onclick="href='customer feedback.html'">Customer Feedback</button>
```

- In Script, function submitForm( ), some reference IDs have been modified.



```
        var cityInput = document.getElementById("city");
        var stateInput = document.getElementById("state");
        var zipInput = document.getElementById("zip");
```

```
        var cityInput = document.getElementById("City");
        var stateInput = document.getElementById("State");
        var zipInput = document.getElementById("Zip");
```



carshop.com

MainImage

**Our Services:**

- Buy new vehicles
- Trade your used vehicles
- Rent before buying
- Repair and maintenance advice

Explore our products

Name:

Male ○ Female ○

Email:

Cell Phone:

Address1 :

Address2 :

City :

State :

Zip :

Please choose a make: Not selected

Please choose a model: Select a make first

Submit

Customer Feedback

*Figure 2. Main Page, v2*

## Dealership Store (Car Images) Page

This “Dealership Store” webpage (Figure 3:  Car images, v1) provides a visually appealing and interactive interface to explore a variety of vehicles available at our dealership. Whether you're searching for a sleek sports car, a robust truck, a spacious SUV, or a fuel-efficient sedan, our store has an extensive collection to meet your needs. The page begins with a prominent heading, "Explore Your Next Vehicle," inviting you to embark on an exciting journey of finding your dream car. As you scroll down, you'll come across a suggestion container where you can select various preferences such as spacious interior, off-road capabilities, luxurious interior, fuel efficiency, hybrid powertrain, and three-row seating. By checking these checkboxes and clicking the "Suggest Car" button, you'll receive tailored recommendations based on your chosen criteria.

Furthermore, if you have a specific make or model in mind, you can enter your preference in the provided text box. The webpage allows you to explore beyond the showcased vehicles, catering to your personal taste and preferences.

As you continue scrolling, you'll encounter a visually appealing display of different vehicles. Each vehicle is presented in a card format, featuring an eye-catching image, the name of the car, and a button that directs you to a search engine with related images. Additionally, accompanying each vehicle is a feature card highlighting its main features, listed in bullet points.

At the bottom of the page, you'll find a convenient "Go Back" link, allowing you to navigate back to the main page of the website.

## Explore Your Next Vehicle

Car Selection Suggestions:
□ Spacious Interior □ Off-road capabilities □ Luxurious interior □ Fuel-efficient □ Hybrid powertrain □ Three-row seating

Suggest Car

Looking for other makes/models? Let us know:

Enter your preferred make/model

Ford Mustang

**Main features:**

- Powerful engine
- Sporty design
- Advanced technology

Ford F-150

**Main features:**

- Robust towing capability
- Spacious interior
- Off-road capabilities

Ford Explorer

**Main features:**

- Spacious and comfortable
- Advanced safety features
- Off-road capabilities

Honda Acura

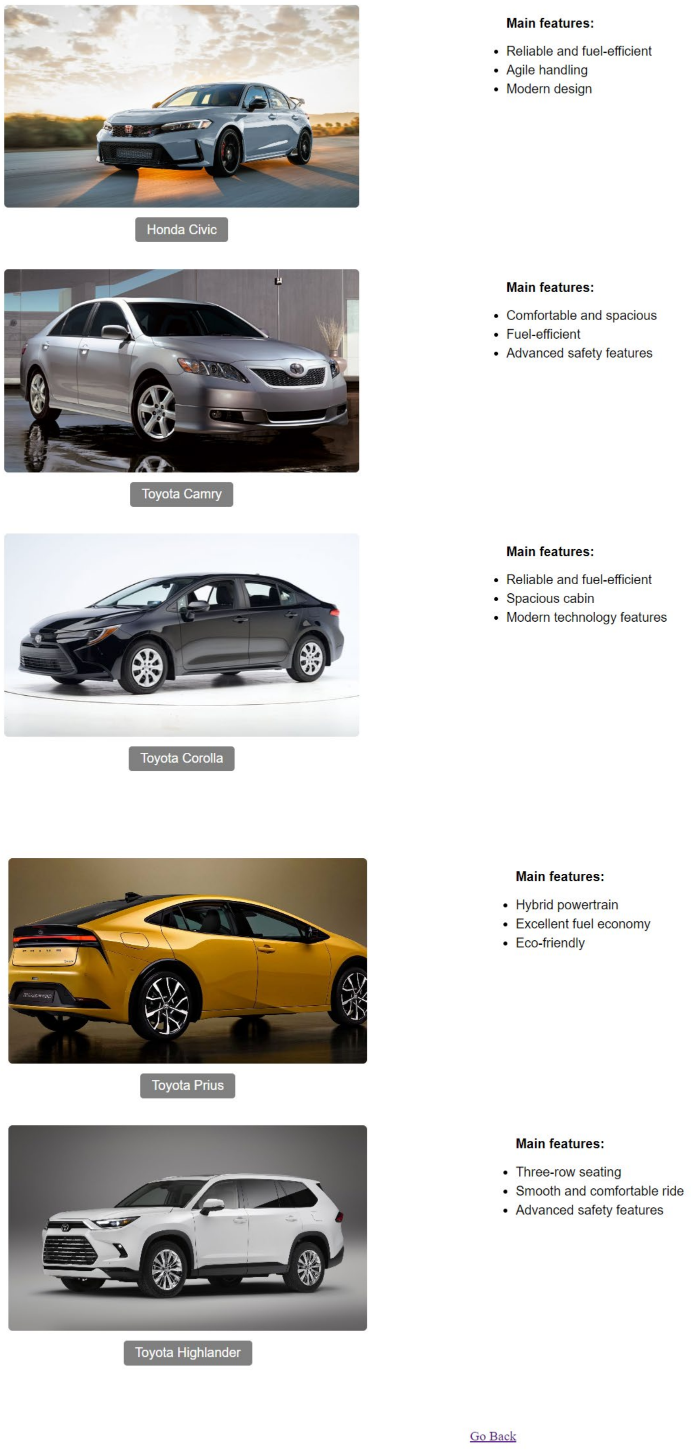
**Main features:**

- Luxurious interior
- Smooth and powerful performance
- Advanced safety features

Honda Accord

**Main features:**

- Fuel-efficient
- Comfortable and spacious
- Advanced safety features



Honda Civic

**Main features:**

- Reliable and fuel-efficient
- Agile handling
- Modern design

Toyota Camry

**Main features:**

- Comfortable and spacious
- Fuel-efficient
- Advanced safety features

Toyota Corolla

**Main features:**

- Reliable and fuel-efficient
- Spacious cabin
- Modern technology features

Toyota Prius

**Main features:**

- Hybrid powertrain
- Excellent fuel economy
- Eco-friendly

Toyota Highlander

**Main features:**

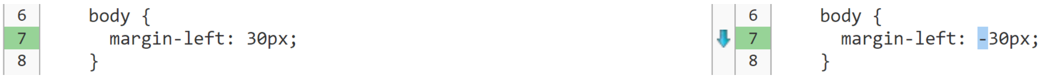
- Three-row seating
- Smooth and comfortable ride
- Advanced safety features

Go Back

*Figure 3: Car images, v1*

The version 2 Main Page (Figure 4: Car images, v2) has the following changes:

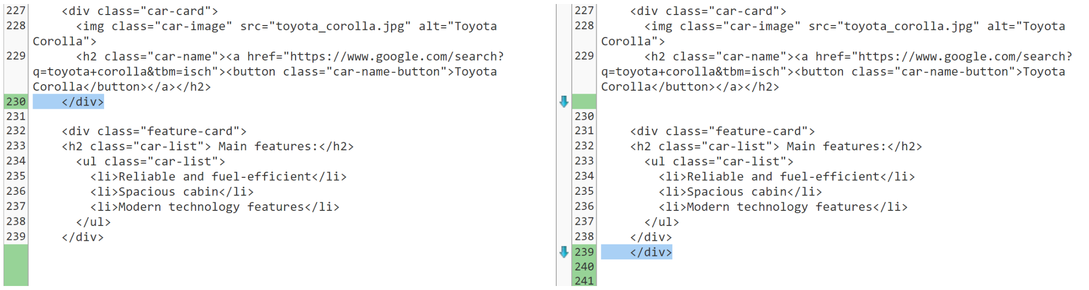
- Page body margin value has been changed.



```
body {
  margin-left: 30px;
}
```

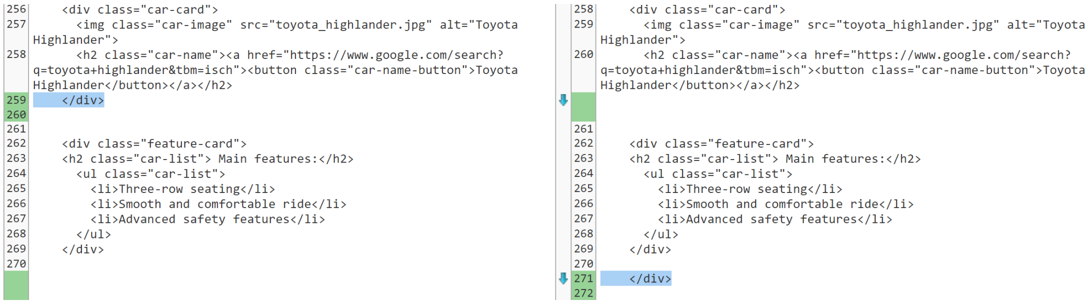
```
body {
  margin-left: -30px;
}
```

- Moved some “feature-card” into “car-card” divisions so that those features lists become part of car image frames.



```
    <div class="car-card">
      <img class="car-image" src="toyota_corolla.jpg" alt="Toyota Corolla">
      <h2 class="car-name"><a href="https://www.google.com/search?q=toyota+corolla&tbm=isch"><button class="car-name-button">Toyota Corolla</button></a></h2>
    </div>

    <div class="feature-card">
    <h2 class="car-list"> Main features:</h2>
      <ul class="car-list">
        <li>Reliable and fuel-efficient</li>
        <li>Spacious cabin</li>
        <li>Modern technology features</li>
      </ul>
    </div>
```

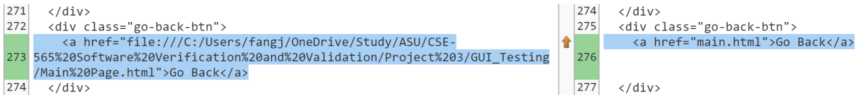
```
    <div class="car-card">
      <img class="car-image" src="toyota_corolla.jpg" alt="Toyota Corolla">
      <h2 class="car-name"><a href="https://www.google.com/search?q=toyota+corolla&tbm=isch"><button class="car-name-button">Toyota Corolla</button></a></h2>


    <div class="feature-card">
    <h2 class="car-list"> Main features:</h2>
      <ul class="car-list">
        <li>Reliable and fuel-efficient</li>
        <li>Spacious cabin</li>
        <li>Modern technology features</li>
      </ul>
    </div>
    </div>
```



```
    <div class="car-card">
      <img class="car-image" src="toyota_highlander.jpg" alt="Toyota Highlander">
      <h2 class="car-name"><a href="https://www.google.com/search?q=toyota+highlander&tbm=isch"><button class="car-name-button">Toyota Highlander</button></a></h2>
    </div>


    <div class="feature-card">
    <h2 class="car-list"> Main features:</h2>
      <ul class="car-list">
        <li>Three-row seating</li>
        <li>Smooth and comfortable ride</li>
        <li>Advanced safety features</li>
      </ul>
    </div>

```

```
    <div class="car-card">
      <img class="car-image" src="toyota_highlander.jpg" alt="Toyota Highlander">
      <h2 class="car-name"><a href="https://www.google.com/search?q=toyota+highlander&tbm=isch"><button class="car-name-button">Toyota Highlander</button></a></h2>


    <div class="feature-card">
    <h2 class="car-list"> Main features:</h2>
      <ul class="car-list">
        <li>Three-row seating</li>
        <li>Smooth and comfortable ride</li>
        <li>Advanced safety features</li>
      </ul>
    </div>

    </div>
```

- Go back to “Main Page” link reference address has been modified.



```
  </div>
  <div class="go-back-btn">
    <a href="file:///C:/Users/fangj/OneDrive/Study/ASU/CSE-565%20Software%20Verification%20and%20Validation/Project%203/GUI_Testing/Main%20Page.html">Go Back</a>
  </div>
```

```
  </div>
  <div class="go-back-btn">
    <a href="main.html">Go Back</a>

  </div>
```

## plore Your Next Vehicle

Selection Suggestions:
pacious Interior ☐ Off-road capabilities ☐ Luxurious interior ☐ Fuel-efficient ☐ Hybrid powertrain ☐ Three-row seating

gest Car

ing for other makes/models? Let us know:

r your preferred make/model

Ford Mustang

**Main features:**

- Powerful engine
- Sporty design
- Advanced technology

Ford F-150

**Main features:**

- Robust towing capability
- Spacious interior
- Off-road capabilities

Ford Explorer

**Main features:**

- Spacious and comfortable
- Advanced safety features
- Off-road capabilities

Honda Acura

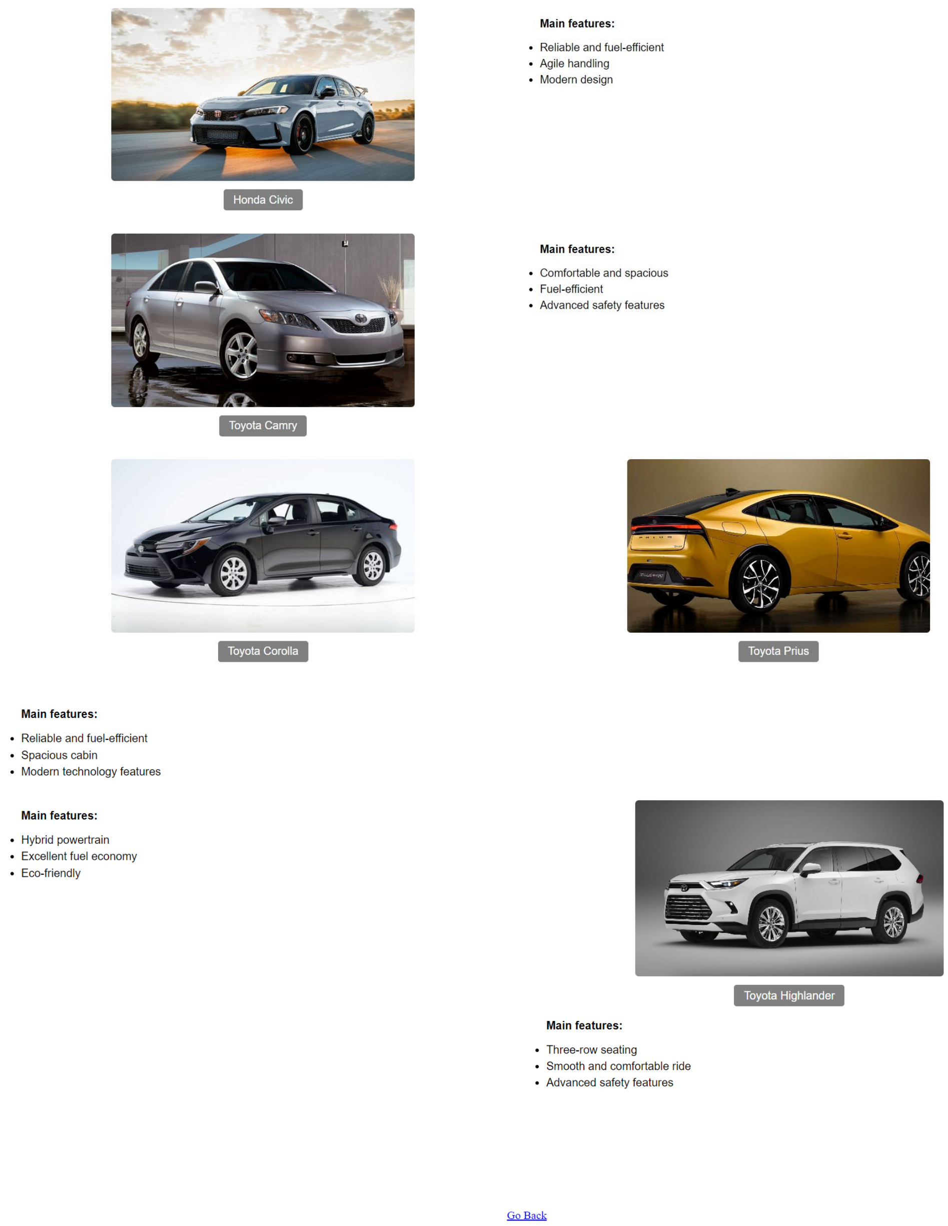
**Main features:**

- Luxurious interior
- Smooth and powerful performance
- Advanced safety features

Honda Accord

**Main features:**

- Fuel-efficient
- Comfortable and spacious
- Advanced safety features



**Main features:**

- Reliable and fuel-efficient
- Agile handling
- Modern design

**Main features:**

- Comfortable and spacious
- Fuel-efficient
- Advanced safety features

**Main features:**

- Reliable and fuel-efficient
- Spacious cabin
- Modern technology features

**Main features:**

- Hybrid powertrain
- Excellent fuel economy
- Eco-friendly

**Main features:**

- Three-row seating
- Smooth and comfortable ride
- Advanced safety features

*Figure 4: Car images, v2*

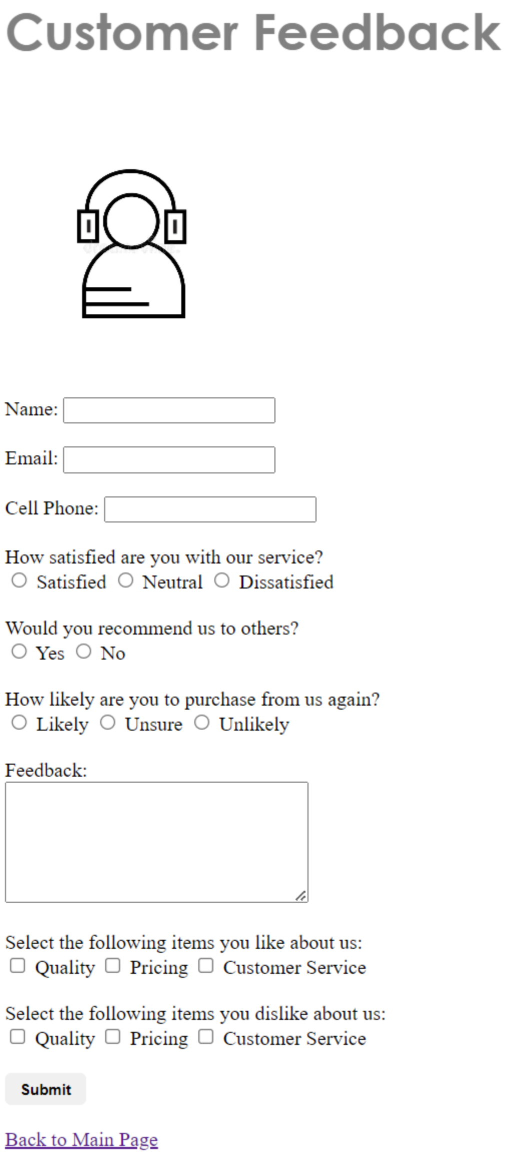
## Customer Feedback page:

The “Customer Feedback” page (Figure 5 Customer Feedback, v1) serves as a user interface for collecting customer feedback. It offers a convenient and structured way for customers to provide their opinions and insights regarding a service or product. The page includes various interactive elements to gather relevant information from users. The page begins with a title "Customer Feedback" to indicate its purpose. It features a heading that clearly states "Customer Feedback" to provide a visual context for users.

To personalize the feedback, the page includes textboxes where customers can enter their name, email address, and cell phone number. This allows for contact details to be collected alongside the feedback. Customers are also presented with a series of radio buttons to answer specific review questions. These questions are designed to gauge satisfaction levels and intentions, such as assessing their satisfaction with the service, whether they would recommend it to others, and their likelihood of making future purchases. In addition to the radio buttons, the page offers checkboxes for customers to express their preferences and provide more detailed feedback. These checkboxes allow customers to indicate specific aspects they appreciate or dislike about the service, such as quality, pricing, or customer service.

To capture customers' detailed comments and suggestions, a feedback text area field is provided. This allows users to enter text-based feedback, offering a space for them to express their thoughts, concerns, or any other relevant information.

Finally, the page includes a submit button that, when clicked, triggers the submission of the feedback. The collected information can then be processed and analyzed to gain insights into customer opinions, identify areas for improvement, and make informed business decisions.



Customer Feedback

Name:

Email:

Cell Phone:

How satisfied are you with our service?
○ Satisfied ○ Neutral ○ Dissatisfied

Would you recommend us to others?
○ Yes ○ No

How likely are you to purchase from us again?
○ Likely ○ Unsure ○ Unlikely

Feedback:

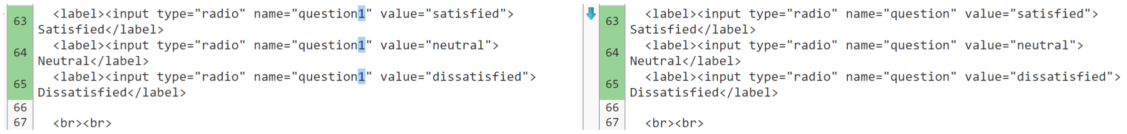
Select the following items you like about us:
□ Quality □ Pricing □ Customer Service

Select the following items you dislike about us:
□ Quality □ Pricing □ Customer Service

*Figure 5 Customer Feedback, v1*

The version 2 Customer Feedback (Figure 6:  Customer Feedback, v2) has the following changes:

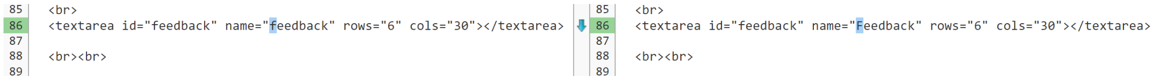
- Radio buttons names have been changed.



```
  <label><input type="radio" name="question1" value="satisfied">
Satisfied</label>
  <label><input type="radio" name="question1" value="neutral">
Neutral</label>
  <label><input type="radio" name="question1" value="dissatisfied">
Dissatisfied</label>

  <br><br>
```

```
  <label><input type="radio" name="question" value="satisfied">
Satisfied</label>
  <label><input type="radio" name="question" value="neutral">
Neutral</label>
  <label><input type="radio" name="question" value="dissatisfied">
Dissatisfied</label>

  <br><br>
```

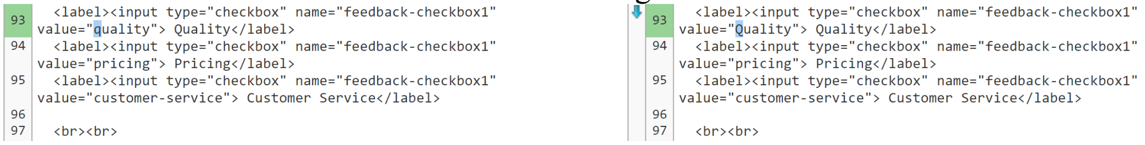
- Feedback text area name has been changed.



```
<br>
<textarea id="feedback" name="feedback" rows="6" cols="30"></textarea>

<br><br>
```

```
<br>
<textarea id="feedback" name="Feedback" rows="6" cols="30"></textarea>

<br><br>
```

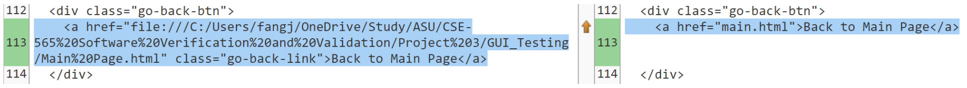
- Feedback-checkbox1 value has been changed.



```
  <label><input type="checkbox" name="feedback-checkbox1"
value="quality"> Quality</label>
  <label><input type="checkbox" name="feedback-checkbox1"
value="pricing"> Pricing</label>
  <label><input type="checkbox" name="feedback-checkbox1"
value="customer-service"> Customer Service</label>

  <br><br>
```

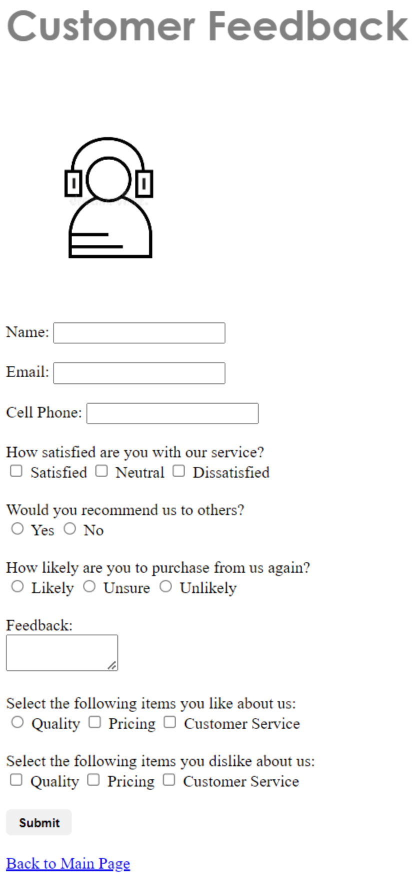
```
  <label><input type="checkbox" name="feedback-checkbox1"
value="Quality"> Quality</label>
  <label><input type="checkbox" name="feedback-checkbox1"
value="pricing"> Pricing</label>
  <label><input type="checkbox" name="feedback-checkbox1"
value="customer-service"> Customer Service</label>

  <br><br>
```

- Go back to “Main Page” link reference address has been modified.



```
  <div class="go-back-btn">
    <a href="file:///C:/Users/fangj/OneDrive/Study/ASU/CSE-
565%20Software%20Verification%20and%20Validation/Project%203/GUI_Testing
/Main%20Page.html" class="go-back-link">Back to Main Page</a>
  </div>
```

```
  <div class="go-back-btn">
    <a href="main.html">Back to Main Page</a>

  </div>
```



Customer Feedback

Name:

Email:

Cell Phone:

How satisfied are you with our service?
□ Satisfied □ Neutral □ Dissatisfied

Would you recommend us to others?
○ Yes ○ No

How likely are you to purchase from us again?
○ Likely ○ Unsure ○ Unlikely

Feedback:

Select the following items you like about us:
○ Quality □ Pricing □ Customer Service

Select the following items you dislike about us:
□ Quality □ Pricing □ Customer Service

Back to Main Page

*Figure 6: Customer Feedback, v2*

## Introduction of Selenium

### Brief Introduction

The Selenium tool is a popular choice for GUI testing, and it has been selected for testing the provided HTML page. Selenium is an open-source framework that offers a range of features and capabilities to automate web browsers. It provides a suite of tools and libraries that enable developers and testers to perform functional testing, regression testing, and cross-browser testing for web applications.

The key features of Selenium include:

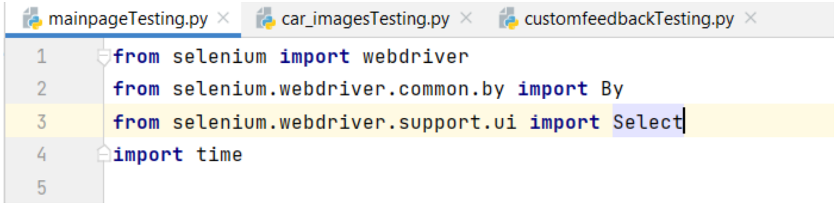
- Browser Compatibility: Selenium supports multiple web browsers, including Chrome, Firefox, Safari, Internet Explorer, and Edge. This allows for comprehensive testing across different browsers to ensure consistent functionality and user experience.
- Cross-Platform Testing: Selenium is a cross-platform tool, meaning it can be used on different operating systems such as Windows, macOS, and Linux. This enables testing teams to verify the application's compatibility and functionality across various platforms.
- Programming Language Support: Selenium supports multiple programming languages such as Java, Python, C#, Ruby, and JavaScript.
- Automation Capabilities: Selenium provides a rich set of APIs and methods for automating browser interactions. Test scripts can simulate user actions like clicking buttons, entering text, selecting options, and navigating through web

pages. This enables the creation of comprehensive test scenarios and ensures accurate and reliable test execution.

- Test Framework Integration: Selenium can be seamlessly integrated with popular test frameworks like JUnit, TestNG, and NUnit. This allows for efficient test management, reporting, and integration with continuous integration (CI) systems, enabling teams to incorporate GUI testing into their existing development processes.
- Extensibility and Customization: Selenium's architecture allows for easy extensibility and customization. Testers can build custom libraries and extend Selenium's functionality to cater to specific testing requirements.

**Usage in this Project**

In this project, three python testing files have been created to test our 3 pages html files: mainpageTesting.py, car_imagesTesting.py, customfeedbackTesting.py. Those files are created in IntelliJ IDEA software and is based on python 3.8 interpreter. First, I installed Selenium in IntelliJ IDEA. Then, I need to import selenium and its APIs I need in each of my testing files (Figure 7:  Selenium import).



```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import Select
import time

```

*Figure 7:  Selenium import*

To fully test my html GUIs, I customized all testing codes against my html styles and functions. In each testing file, I basically have two main parts: test styles of GUI and test all functions. For example, in mainpageTesting.py file, I have the following functions:

- **test_styles():** This function tests the styles of various elements on the main page. It uses Selenium's WebDriver to launch a Chrome browser and navigate to the HTML file containing the main page. Then, it finds specific elements using different locator strategies (such as finding elements by class name) and retrieves their CSS properties using value_of_css_property(). Finally, it compares the retrieved CSS properties with expected values to determine whether the styles are applied correctly.
- **test_button_click():** This function tests the functionality of the "Explore our products" button. It follows a similar approach as test_styles(), but instead of checking styles, it locates the button element using an XPath expression and clicks on it. After a short delay using time.sleep(), it verifies whether the current URL matches the expected URL for the next page.
- **test_dropdown():** This function tests the behavior of dropdown menus. It loads the main page, selects different options from the "makes" dropdown using Selenium's Select class, and then verifies the available options in the "models" dropdown based on the selected make. It compares the actual options with the expected options to determine the correctness of the dropdown behavior.
- **test_radio_select():** This function tests the selection of radio buttons. It loads the main page, locates radio buttons using their IDs, and clicks on them. After each

click, it verifies whether the clicked radio button is selected by using the is_selected() method.

- **test_form_submission():** This function tests the submission of a form on the main page. It defines several test cases, each containing input values for different form fields. The function loads the main page, fills in the form fields using send_keys(), and clicks the submit button. After a short delay, it handles the alert dialog and verifies the text of the alert message against the expected values for the submitted form data.
- **test_customerfeedback():** This function tests the functionality of the "Customer Feedback" button. It loads the main page, finds the button element using an XPath expression, clicks on it, and waits for a short duration. It then verifies whether the current URL matches the expected URL for the customer feedback page.

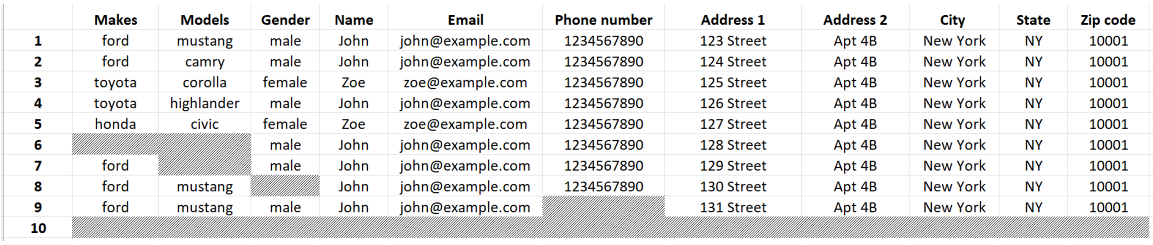
Test cases are only applied in **test_form_submission()** function, which will use different combinations of inputs to get outputs, and then check them against the expected outputs. The other functions don't need any use cases because they only check against the code elements. The other 2 test files have the same logic and structures as this one.

# Testing Process

## Test Cases

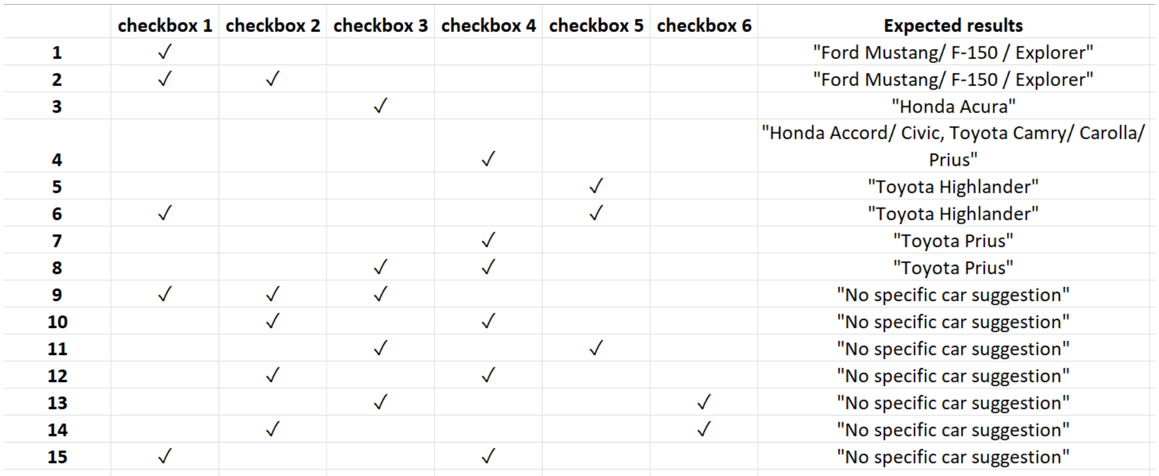
There are 3 sets of test cases used in each of 3 testing files.

- Main Page testing: 10 test cases used in test_form_submission( ) function.



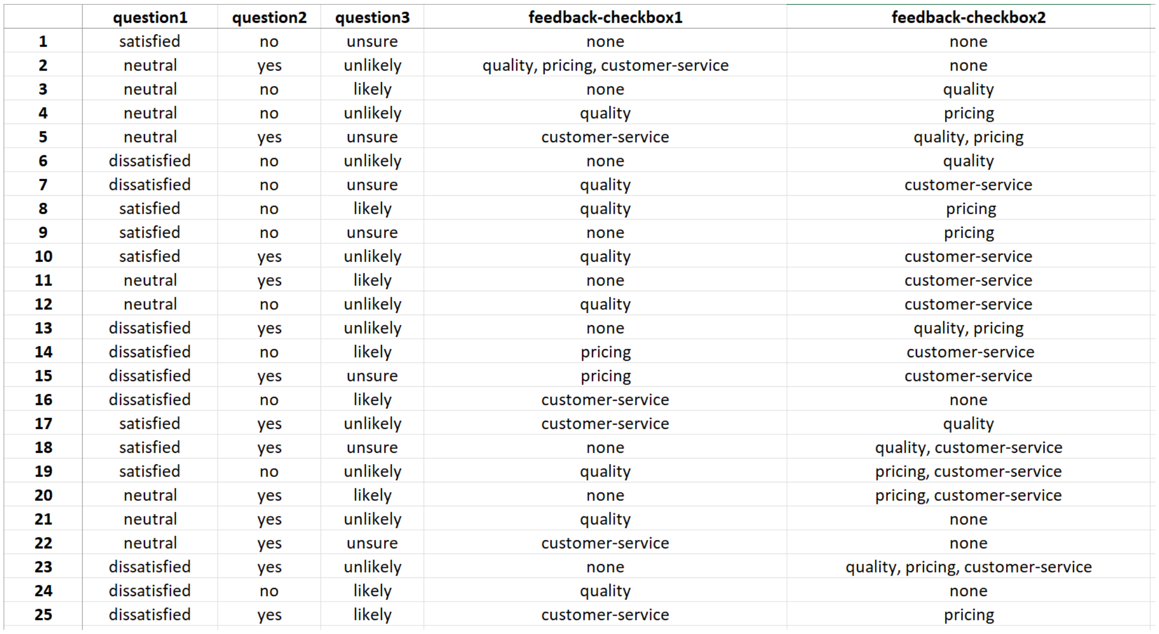
| | Makes | Models | Gender | Name | Email | Phone number | Address 1 | Address 2 | City | State | Zip code |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | ford | mustang | male | John | john@example.com | 1234567890 | 123 Street | Apt 4B | New York | NY | 10001 |
| 2 | ford | camry | male | John | john@example.com | 1234567890 | 124 Street | Apt 4B | New York | NY | 10001 |
| 3 | toyota | corolla | female | Zoe | zoe@example.com | 1234567890 | 125 Street | Apt 4B | New York | NY | 10001 |
| 4 | toyota | highlander | male | John | john@example.com | 1234567890 | 126 Street | Apt 4B | New York | NY | 10001 |
| 5 | honda | civic | female | Zoe | zoe@example.com | 1234567890 | 127 Street | Apt 4B | New York | NY | 10001 |
| 6 | ▨ | ▨ | male | John | john@example.com | 1234567890 | 128 Street | Apt 4B | New York | NY | 10001 |
| 7 | ford | ▨ | male | John | john@example.com | 1234567890 | 129 Street | Apt 4B | New York | NY | 10001 |
| 8 | ford | mustang | ▨ | John | john@example.com | 1234567890 | 130 Street | Apt 4B | New York | NY | 10001 |
| 9 | ford | mustang | male | John | john@example.com | ▨ | 131 Street | Apt 4B | New York | NY | 10001 |
| 10 | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ |

- Car images page testing: 15 test cases used in test_suggestions( ) function based on All Combination method.



| | checkbox 1 | checkbox 2 | checkbox 3 | checkbox 4 | checkbox 5 | checkbox 6 | Expected results |
|---|---|---|---|---|---|---|---|
| 1 | ✓ | | | | | | "Ford Mustang/ F-150 / Explorer" |
| 2 | ✓ | ✓ | | | | | "Ford Mustang/ F-150 / Explorer" |
| 3 | | | ✓ | | | | "Honda Acura" |
| 4 | | | | ✓ | | | "Honda Accord/ Civic, Toyota Camry/ Carolla/ Prius" |
| 5 | | | | | ✓ | | "Toyota Highlander" |
| 6 | ✓ | | | | ✓ | | "Toyota Highlander" |
| 7 | | | | ✓ | | | "Toyota Prius" |
| 8 | | | ✓ | ✓ | | | "Toyota Prius" |
| 9 | ✓ | ✓ | ✓ | | | | "No specific car suggestion" |
| 10 | | ✓ | | ✓ | | | "No specific car suggestion" |
| 11 | | | ✓ | | ✓ | | "No specific car suggestion" |
| 12 | | ✓ | | ✓ | | | "No specific car suggestion" |
| 13 | | | ✓ | | | ✓ | "No specific car suggestion" |
| 14 | | ✓ | | | | ✓ | "No specific car suggestion" |
| 15 | ✓ | | | ✓ | | | "No specific car suggestion" |

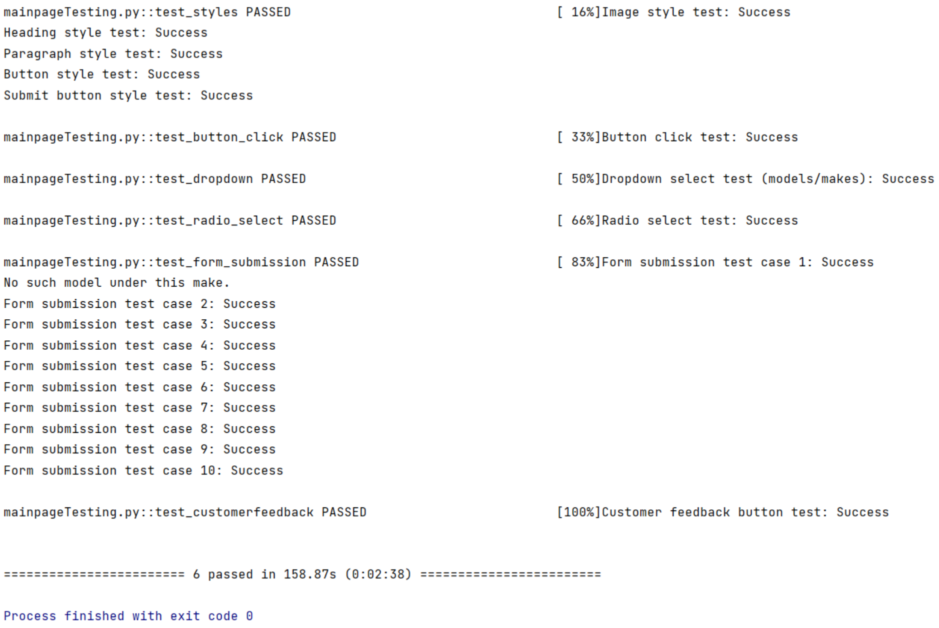
- Customer feedback testing: 25 test cases are applied using Pairwise Combination method.



|  | question1 | question2 | question3 | feedback-checkbox1 | feedback-checkbox2 |
|---|---|---|---|---|---|
| **1** | satisfied | no | unsure | none | none |
| **2** | neutral | yes | unlikely | quality, pricing, customer-service | none |
| **3** | neutral | no | likely | none | quality |
| **4** | neutral | no | unlikely | quality | pricing |
| **5** | neutral | yes | unsure | customer-service | quality, pricing |
| **6** | dissatisfied | no | unlikely | none | quality |
| **7** | dissatisfied | no | unsure | quality | customer-service |
| **8** | satisfied | no | likely | quality | pricing |
| **9** | satisfied | no | unsure | none | pricing |
| **10** | satisfied | yes | unlikely | quality | customer-service |
| **11** | neutral | yes | likely | none | customer-service |
| **12** | neutral | no | unlikely | quality | customer-service |
| **13** | dissatisfied | yes | unlikely | none | quality, pricing |
| **14** | dissatisfied | no | likely | pricing | customer-service |
| **15** | dissatisfied | yes | unsure | pricing | customer-service |
| **16** | dissatisfied | no | likely | customer-service | none |
| **17** | satisfied | yes | unlikely | customer-service | quality |
| **18** | satisfied | yes | unsure | none | quality, customer-service |
| **19** | satisfied | no | unlikely | quality | pricing, customer-service |
| **20** | neutral | yes | likely | none | pricing, customer-service |
| **21** | neutral | yes | unlikely | quality | none |
| **22** | neutral | yes | unsure | customer-service | none |
| **23** | dissatisfied | yes | unlikely | none | quality, pricing, customer-service |
| **24** | dissatisfied | no | likely | quality | none |
| **25** | dissatisfied | yes | likely | customer-service | pricing |

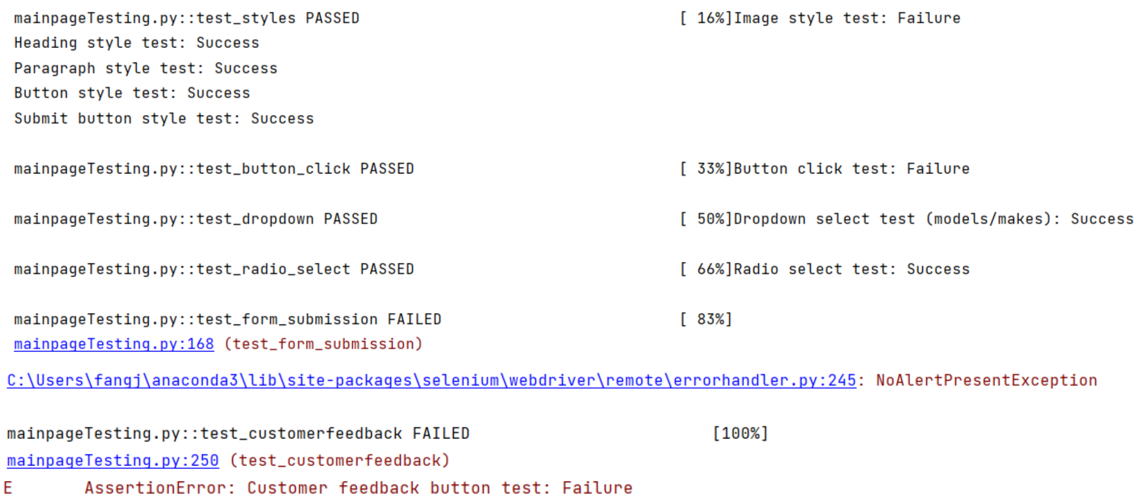
## Test Results

- Main Page testing results:



```
mainpageTesting.py::test_styles PASSED                                  [ 16%]Image style test: Success
Heading style test: Success
Paragraph style test: Success
Button style test: Success
Submit button style test: Success

mainpageTesting.py::test_button_click PASSED                            [ 33%]Button click test: Success

mainpageTesting.py::test_dropdown PASSED                                [ 50%]Dropdown select test (models/makes): Success

mainpageTesting.py::test_radio_select PASSED                            [ 66%]Radio select test: Success

mainpageTesting.py::test_form_submission PASSED                         [ 83%]Form submission test case 1: Success
No such model under this make.
Form submission test case 2: Success
Form submission test case 3: Success
Form submission test case 4: Success
Form submission test case 5: Success
Form submission test case 6: Success
Form submission test case 7: Success
Form submission test case 8: Success
Form submission test case 9: Success
Form submission test case 10: Success

mainpageTesting.py::test_customerfeedback PASSED                        [100%]Customer feedback button test: Success


======================== 6 passed in 158.87s (0:02:38) ========================

Process finished with exit code 0
```

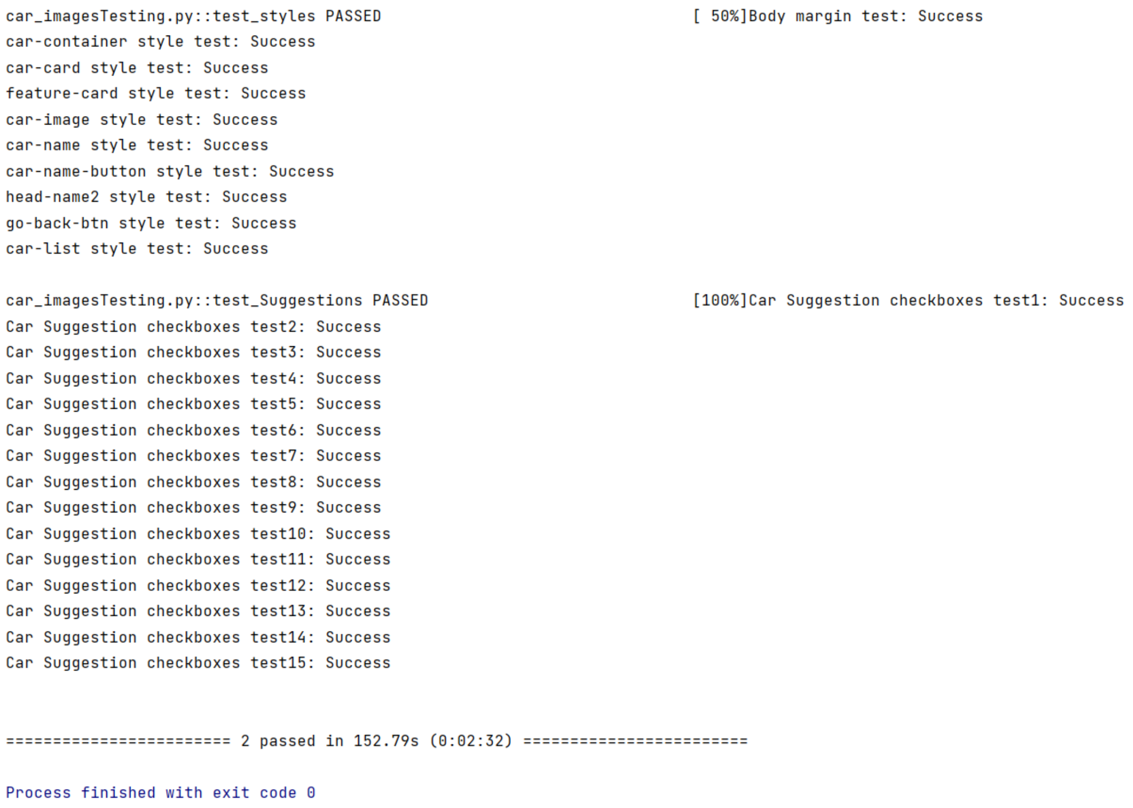
*Figure 8:  Main page (v1) testing result*



```
mainpageTesting.py::test_styles PASSED                              [ 16%]Image style test: Failure
Heading style test: Success
Paragraph style test: Success
Button style test: Success
Submit button style test: Success

mainpageTesting.py::test_button_click PASSED                        [ 33%]Button click test: Failure

mainpageTesting.py::test_dropdown PASSED                            [ 50%]Dropdown select test (models/makes): Success

mainpageTesting.py::test_radio_select PASSED                        [ 66%]Radio select test: Success

mainpageTesting.py::test_form_submission FAILED                     [ 83%]
mainpageTesting.py:168 (test_form_submission)
C:\Users\fangj\anaconda3\lib\site-packages\selenium\webdriver\remote\errorhandler.py:245: NoAlertPresentException

mainpageTesting.py::test_customerfeedback FAILED                      [100%]
mainpageTesting.py:250 (test_customerfeedback)
E       AssertionError: Customer feedback button test: Failure
```

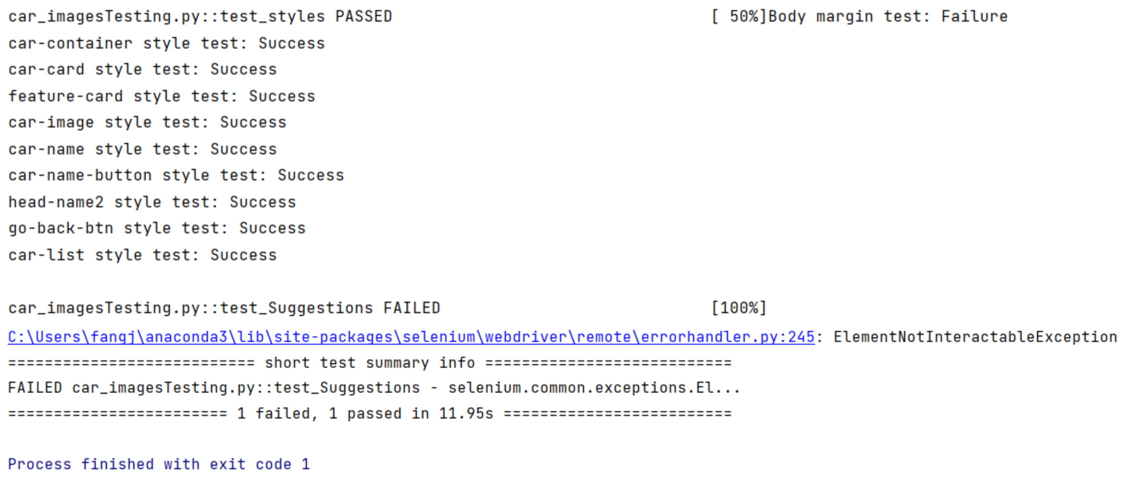
*Figure 9: Main page (v2) testing result*

- Car images/ Dealership Store testing results:



```
car_imagesTesting.py::test_styles PASSED                            [ 50%]Body margin test: Success
car-container style test: Success
car-card style test: Success
feature-card style test: Success
car-image style test: Success
car-name style test: Success
car-name-button style test: Success
head-name2 style test: Success
go-back-btn style test: Success
car-list style test: Success

car_imagesTesting.py::test_Suggestions PASSED                       [100%]Car Suggestion checkboxes test1: Success
Car Suggestion checkboxes test2: Success
Car Suggestion checkboxes test3: Success
Car Suggestion checkboxes test4: Success
Car Suggestion checkboxes test5: Success
Car Suggestion checkboxes test6: Success
Car Suggestion checkboxes test7: Success
Car Suggestion checkboxes test8: Success
Car Suggestion checkboxes test9: Success
Car Suggestion checkboxes test10: Success
Car Suggestion checkboxes test11: Success
Car Suggestion checkboxes test12: Success
Car Suggestion checkboxes test13: Success
Car Suggestion checkboxes test14: Success
Car Suggestion checkboxes test15: Success


======================== 2 passed in 152.79s (0:02:32) ========================

Process finished with exit code 0
```

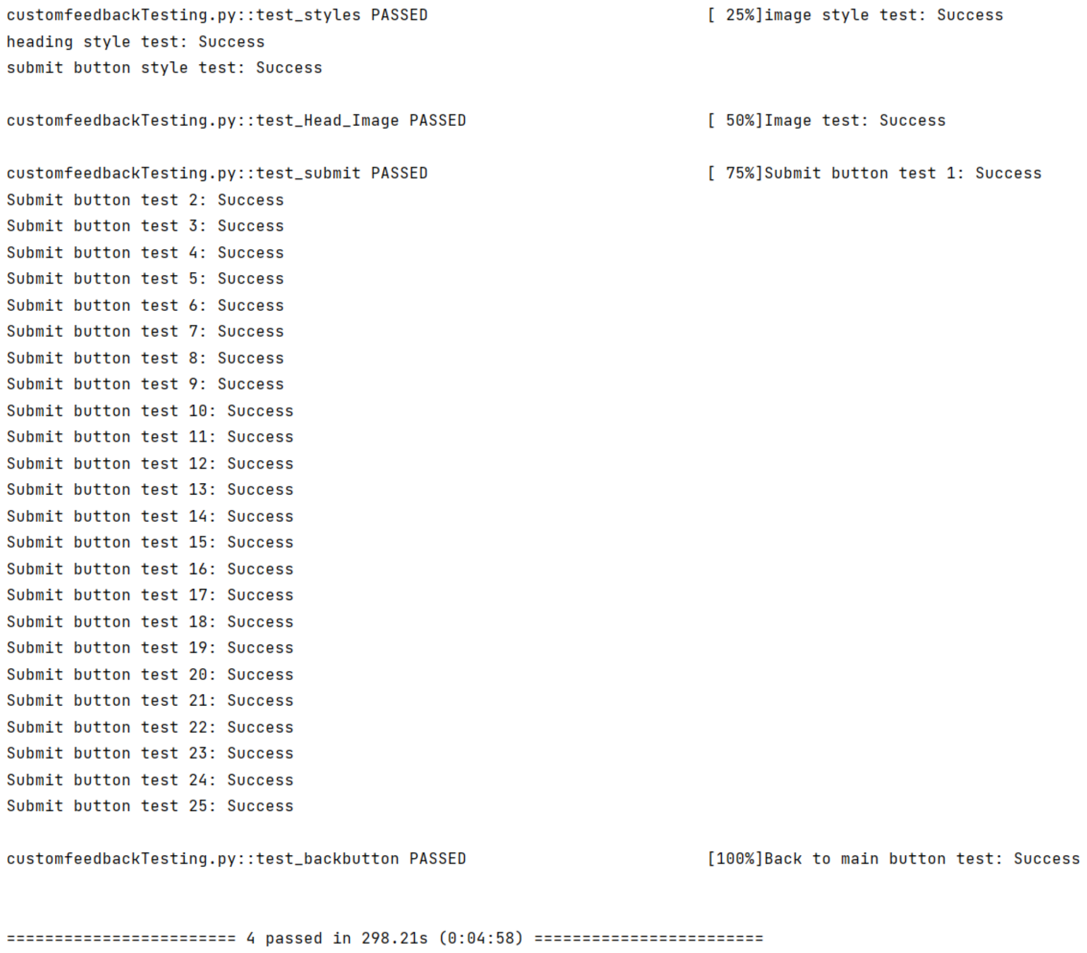
*Figure 10: Car Images page (v1) testing result*



```
car_imagesTesting.py::test_styles PASSED                         [ 50%]Body margin test: Failure
car-container style test: Success
car-card style test: Success
feature-card style test: Success
car-image style test: Success
car-name style test: Success
car-name-button style test: Success
head-name2 style test: Success
go-back-btn style test: Success
car-list style test: Success

car_imagesTesting.py::test_Suggestions FAILED                    [100%]
C:\Users\fangj\anaconda3\lib\site-packages\selenium\webdriver\remote\errorhandler.py:245: ElementNotInteractableException
========================== short test summary info ==========================
FAILED car_imagesTesting.py::test_Suggestions - selenium.common.exceptions.El...
======================= 1 failed, 1 passed in 11.95s ========================

Process finished with exit code 1
```

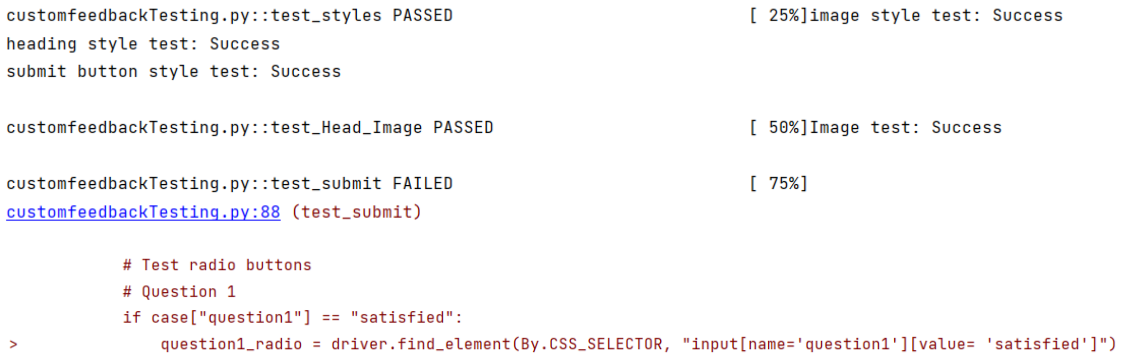
*Figure 11:  Car Images page (v2) testing result*

- Customer Feedback page testing results:

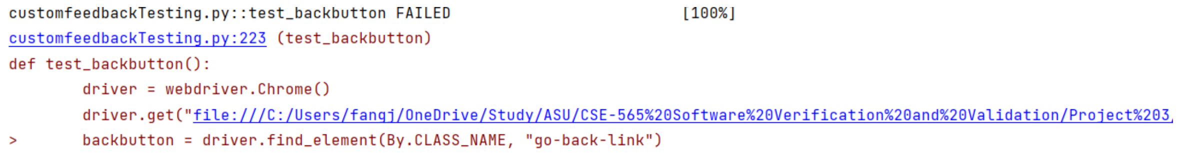


```
customfeedbackTesting.py::test_styles PASSED                     [ 25%]image style test: Success
heading style test: Success
submit button style test: Success

customfeedbackTesting.py::test_Head_Image PASSED                 [ 50%]Image test: Success

customfeedbackTesting.py::test_submit PASSED                     [ 75%]Submit button test 1: Success
Submit button test 2: Success
Submit button test 3: Success
Submit button test 4: Success
Submit button test 5: Success
Submit button test 6: Success
Submit button test 7: Success
Submit button test 8: Success
Submit button test 9: Success
Submit button test 10: Success
Submit button test 11: Success
Submit button test 12: Success
Submit button test 13: Success
Submit button test 14: Success
Submit button test 15: Success
Submit button test 16: Success
Submit button test 17: Success
Submit button test 18: Success
Submit button test 19: Success
Submit button test 20: Success
Submit button test 21: Success
Submit button test 22: Success
Submit button test 23: Success
Submit button test 24: Success
Submit button test 25: Success

customfeedbackTesting.py::test_backbutton PASSED                 [100%]Back to main button test: Success


======================= 4 passed in 298.21s (0:04:58) =======================
```

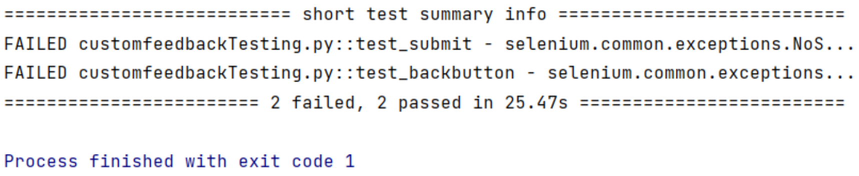
*Figure 12:  Customer Feedback page (v1) testing result*



```
customfeedbackTesting.py::test_styles PASSED                           [ 25%]image style test: Success
heading style test: Success
submit button style test: Success

customfeedbackTesting.py::test_Head_Image PASSED                       [ 50%]Image test: Success

customfeedbackTesting.py::test_submit FAILED                           [ 75%]
customfeedbackTesting.py:88 (test_submit)

            # Test radio buttons
            # Question 1
            if case["question1"] == "satisfied":
>               question1_radio = driver.find_element(By.CSS_SELECTOR, "input[name='question1'][value= 'satisfied']")
```



```
customfeedbackTesting.py::test_backbutton FAILED                             [100%]
customfeedbackTesting.py:223 (test_backbutton)
def test_backbutton():
        driver = webdriver.Chrome()
        driver.get("file:///C:/Users/fangj/OneDrive/Study/ASU/CSE-565%20Software%20Verification%20and%20Validation/Project%203
>       backbutton = driver.find_element(By.CLASS_NAME, "go-back-link")
```



```
=========================== short test summary info ===========================
FAILED customfeedbackTesting.py::test_submit - selenium.common.exceptions.NoS...
FAILED customfeedbackTesting.py::test_backbutton - selenium.common.exceptions...
======================== 2 failed, 2 passed in 25.47s =========================

Process finished with exit code 1
```

**Conclusion and Assessment**

Selenium with IntelliJ IDEA can help write automated tests for HTML applications that interact with web elements, simulate user actions, and verify expected behaviors. Selenium allows us to locate elements on the web page using different strategies such as CSS selectors, XPath, or element IDs. We can perform actions like clicking buttons, filling out forms, and extracting information from the web page.

IntelliJ IDEA offers features like code auto-completion, refactoring tools, and integrated debugging, which can greatly enhance productivity when writing Selenium tests. The IDE also provides support for running tests in different browsers and environments, allowing us to validate the behavior of your HTML applications across multiple platforms.

When executing Selenium tests in IntelliJ IDEA, we can view the test results directly within the IDE, making it easy to identify any failures or errors. The IDE provides comprehensive reporting capabilities, including detailed logs and stack traces, to help you debug and troubleshoot any issues that arise during test execution.

In general, using Selenium with IntelliJ IDEA offers a robust and efficient way to test HTML applications. The combination of IntelliJ IDEA's development features and Selenium's web automation capabilities enables us to create reliable and comprehensive test suites for your HTML applications.

## References

Gundecha, U. (2015). Selenium Testing Tools Cookbook: *Over 90 recipes to help you build and run automated tests for your web applications with Selenium WebDriver*. Packt Publishing.

Richardson, A. (2018). *Selenium WebDriver with Java: Learn automation testing techniques using Selenium WebDriver with Java.* Packt Publishing.

Selenium (n.d.). Documentation. Retrieved from https://www.selenium.dev/documentation/

SeleniumHQ/selenium. (n.d.). *Selenium WebDriver JavaDoc.* Retrieved from https://www.selenium.dev/selenium/docs/api/java/