• Titles: Contact us Page and Search for “Dress” Testing

• Author: Wahid Kassem

1. **Introduction**

   This documentation outlines the testing processes for a shopping site designed specifically for women. It covers the testing environment, test case structure, and specific scenarios related to the site's features

2. **Testing Environment**
   - Windows 11
   - Visual studio code
   - NIghtwatch framework
   - Chrome
3. **Test Cases**

Test case Name	**1.Contact us page (unsubmitted)**	**2.Contact us page submitted without file attachment**
Steps	• Open website and wait until every element is visible • Click on Contact us • Fill Your Email, Order reference and message. • Click on send button • Error in submitting because mandatory fields empty	• Open website and wait until every element is visible • Click on Contact us • Onen dropdown and choses Webmaster • Fill Your Email, Order reference and message. • Click on send button • Submit the form successfully
Expected Results	An unsuccess message is displayed, and the form data is not submitted	A success message is displayed, and the form data is submitted
Actual Results	An unsuccess message is displayed, and the form data is not submitted	A success message is displayed, and the form data is submitted
Status	passed	passed
Issues and Their Handling	During testing, normal scrolling failed, so I used the (execute) command for manual control. The full-screen functionality did not work as expected so I tried to use (requestFullscreen()), To reliably identify elements, I chose CSS selectors, as they provided a simple and effective way to obtain unique elements for interaction.	Using (waitUntilVisible)with a dropdown element caused the test to fail. To address this, I implemented an explicit wait of 3 minutes to ensure the element would be visible before clicking on it, allowing the test to proceed successfully.

Test case Name	**3.Contact us page submitted with file attachment**	**4.Searching for Dress**
Steps	• Open website and wait until every element is visible • Click on Contact us • Onen dropdown and choses Webmaster • Fill Your Email, Order reference and message. • Attach a file from desktop • Click on send button • Submit the form successfully	• Open website and wait until every element is visible • Click on search engine • Searching for “Dress” • Click on magnificent icon to search • Verify the search
Expected Results	A success message is displayed, and the form data is submitted	displaying dress items.
Actual Results	A success message is displayed, and the form data is submitted	displaying dress items.
Status	passed	passed
Issues and Their Handling	Using (waitUntilVisible)with a dropdown element caused the test to fail. To address this, I implemented an explicit wait of 3 minutes to ensure the element would be visible before clicking on it, allowing the test to proceed successfully.	The Enter key function failed to trigger results when clicked, so I used a clickable icon instead to achieve the desired outcome. To verify the dress results, I opted to search for dress images in the results, ensuring accurate identification of the relevant items.

4. Common Issues

I encountered common issues with version compatibility between Chrome, WebDriver, and Nightwatch while writing scripts. I resolved these by updating all components to their latest versions. While implementing Page Objects, some functions did not work directly due to hardcoded selectors, but I managed to address these issues. Using Page Objects proved to be beneficial, as it made the code clearer and easier to debug. Additionally, it simplified error identification and resolution. This approach greatly enhances the understanding for developers reviewing the code for the first time.

5. Conclusion

This documentation outlines test cases for a shopping site for women, including functionalities like searching for items and sending reports. I ensured that every button is clickable and explored alternative methods when direct approaches did not work. The goal is to verify that all features function as intended.