**--Test Completion Report**

**1. Summary of testing performed**

In this project, we tested the N11 website and its various functions.

Our tests include the website's main and most important functions, namely the login, search, shopping cart, and advanced search functions, and the website's performance. We've done these tests with using Selenium Java, Apache JMeter, and seleniumIDE.

* The testing of the login function includes scenarios where the user logs in to the website successfully, or enters the wrong email, password, or both.
* The testing of the search functions is the most complex and detailed one, as it includes scenarios for accurate search input, accurate and specific search input, search input with random capitalized letters, search input with typos, search input with only numbers, search input with only special characters, and empty search input.
* The testing of the shopping cart function mainly includes scenarios of adding and removing specific amounts of items from the cart.
* The testing of the advanced search functions includes scenarios of the advanced options of the search function being tested, such as filtering by rating or price.
* The testing of the website's performance includes, or was supposed to include, scenarios of multiple users accessing the website and how the website reacts to it.

**2. Deviations from planned testing**

During the testing process, we encountered some deviations, which forced us to change our methods, codes, or the way we wanted to test the proper functions of the website.

For example, during the testing of the shopping cart functions, we found out that the website does not update the number of items in the basket on the background if you try to remove either the same amount of items that are in the basket or more which also causes the total price of the shopping cart not to be updated. This forced us to change the automated code that was testing this scenario.

Similarly, we found out that the website also does not update the item count of items in the basket during and adding and removing operation. Instead, it completely removes the data, and then proceeds to re-add it. This caused a NoSuchElement exception during the testing, which forced us to change our automated code to adapt to the situation. Though, it was more of an unexpected function of the website rather than a flawed code.

**3. Test completion evaluation**

For the most part, we got the results we expected from the testing process, which means the website works properly as intended and expected. Although most of the results were sufficient, there are some cases where the criteria weren't met.

For example, during the testing process of the advanced search function, we saw that the website's "*sort-by-rating*" function wasn't working properly. For instance, selecting the "*4 or more stars*" option was supposed to give us results with items that have *4 or more stars*, as the name suggests. However, after checking the results, we saw a few items with 3 stars, which meant that the function was either flawed or wasn't working as intended, and was a huge mistake on the website's part.

This error applies to all versions of the sort-by-rating function of the website, except *1 or more stars*, which works as intended.

**4. Factors that blocked progress**

There were several factors that blocked our testing process, which, mostly, forced us to abandon those testing methods.

At the very beginning of the project, there was also a plan to test the register function of the website. However, after writing the automated test function, we encountered a problem with the website: CAPTCHA. The website either had protection for the automated account creation or this as part of its security in general. Due to being unable to bypass CAPTCHA with automated tests, we have abandoned testing the register function of the website.

Another problem we encountered was during the performance testing. For some reason, either for security purposes or something else, the website does not allow any kind of performance tests to the applied to itself, no matter what. Other shopping websites, such as Trendyol and Amazon, responded positively to our performance test attempt, which made us come to this conclusion. We still ended up including our attempts to test this function in our presentation and report.

**5. Test measures**

**A. Test Cases**

* **Total Test Cases**: 30.
* **Executed Test Cases**: 29.
* **Passed Test Cases**: 28.
* **Failed Test Cases**: 1.
* **Blocked Test Cases**: 1.
* **Test Case Execution Rate**: 96.6666666667%

**B. Defects**

* **Total Defects**: 5.
* **Defect Severity**: low

**C. Incidents**

* **Total Incidents**: 5.
* **Resolved Incidents**: 3.
* **Unresolved Incidents**: 2.

**D. Activity Progress**

* **Test Plan Progress**: 100%.
* **Test Design Progress**: 100%.
* **Test Execution Progress**: 96.6666666667%.

**E. Resource Consumption**

* **Time Spent**: 2 weeks.
* **Human Resources**: 2 person.
* **Tool Utilization**: Selenium Java, Apache JMeter, SeleniumIDE.

**6. Test deliverables**

1. **Test Plan**

The purpose of this test plan is to test some basic functionalities of n11.com and to conduct load testing on the site. The primary objectives are to ensure that key features such as user registration, login, product search and adding items to the shopping cart functions process correctly under normal conditions. Additionally, load testing will be performed to simulate high user traffic and evaluate the website's performance, measuring response times and server resource utilization. This comprehensive approach will help identify and address any performance bottlenecks or issues.

1. **Test Case Specifications**

Test Case 1: Login Functionality

Description: Test the login functionality with both valid and invalid credentials. The user logs in with a correct email and password, then attempts to log in with incorrect email and password.

Expected Results: The user should successfully log in and be redirected to the homepage with valid credentials, and an error message should be displayed with invalid credentials.

Actual Result: System responds exactly as expected.

Test Case 2: Search Functionality

Description: Test the search functionality using mixed case letters, special characters, very long strings, and empty strings.

Expected Results: The search results should be case-insensitive and handle special characters appropriately. Very long strings should either return relevant results or an appropriate message indicating no results found. An empty string should prompt the user to enter a search term.

Actual Results: Search bar functions as expected on normal searches. If a string that is too long sent into search bar, it limits the string by 100 characters. When a search is conducted with random strings, the system suggests relevant results; if no relevant results are found, it returns no results. And lastly, if an empty string sent, it returns no results.

Test Case 3: Cart Functionality

Description: Test the cart functionality by adding and removing products, and checking the application of discounts.

Expected Results: Products should be correctly added to and removed from the cart. Any applicable discounts should be correctly applied and reflected in the cart total.

Actual Results: Items can be added to the cart without causing any problems. If stocks are not sufficient for the desired number of items, system warns the user about the number of items in stock and limits the order to that number and do not allow user to increase the number. But when decreasing the number of items, if user uses the decrease number button more than the number of items in cart, it causes an error to occur. System fails to update the number of the item in cart but shows the number as ‘1’. And also, despite reducing the number of items, the price remains the same.Which is a result that is not supposed to happen.

Test Case 4: Filtering Functionality

Description: Test the product filtering functionality in the search results by filtering based on price and rating.

Expected Results: Products should be correctly filtered according to the specified price range and rating, and the search results should update to reflect the applied filters.

Actual Results: When searching for products and filtering by price, the filtering works correctly. However, when filtering by rating, it does not function properly, and products that should be excluded according to the filter are still displayed on the page. This indicates that the filtering function by rating is not working correctly.

1. **Test Procedure Specifications**

**Test Case 1: Login Functionality**

1. Navigate to the login page.
2. Enter valid credentials (correct email and password).
3. Click on the login button.
4. Verify that the user is redirected to the homepage.
5. Repeat steps 1-3 with invalid credentials (incorrect email and password).

**Test Case 2: Search Functionality**

1. Enter mixed case letters, special characters, and very long strings into the search bar.
2. Submit the search query.
3. Verify the search results.

**Test Case 3: Cart Functionality**

1. Add products to the cart.
2. Remove products from the cart.
3. Check the application of discounts.

**Test Case 4: Filtering Functionality**

1. Filter search results by price range.
2. Filter search results by rating.
3. Check for the results.

**7. Lessons learned**

The testing conducted on n11.com provided valuable insights applicable to software testing practices. It underscored the importance of meticulous testing, as even seemingly minor oversights can result in significant issues. User feedback emerged as a critical factor, revealing that features perceived as functional from a developer's perspective may behave unexpectedly from a user's standpoint. The testing also highlighted that even big companies can have simple software bugs that are easy to miss but easy to fix. These bugs, although seemingly insignificant, can sometimes lead to significant issues if not addressed promptly.

**Anıl Abdullah İnce – 21070006006**

**Rıza Toprak Orman - 21070006027**