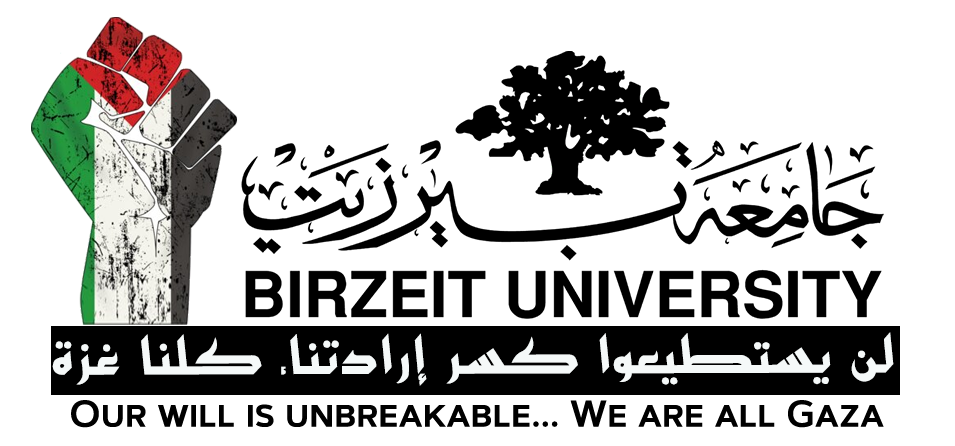
****

**Faculty of Engineering and Technology**

**Department of Computer Science**

**Comp438 – Quality Assurance course**

**Title of Project:** Software testing for a Real-world Software Application

**Group Members with IDs:**

|  |  |
| --- | --- |
| **Name** | **ID** |
| Asmaa Ankoush | 1212410 |
| Dalia Hamed | 1211601 |
| Rayan A Hamayel | 1211073 |

**Doctor Name: Dr. Faisal Shehadeh**

# Abstract

We rely on apps and websites for nearly everything, from shopping to staying connected with friends. Since we use them so often, it’s crucial that they function properly. Imagine trying to make an important purchase and the app crashes, or dealing with a slow, difficult-to-navigate website. It would be frustrating, right? That’s where Quality Assurance (QA) comes in. QA ensures that everything works smoothly, so we can have a positive and hassle-free experience.

Quality Assurance (QA) involves thoroughly checking software to identify and resolve issues before they affect users. QA professionals design test plans, execute tests, and search for bugs, which helps improve the software, ensures user satisfaction, and saves both time and money.

In this project, we will focus on testing a real website like Noon. We’ll explore how the site functions and develop a strategy for testing it. We will create simple test cases and run them to evaluate the website’s performance. Additionally, we will use automation tools to test certain features more efficiently. This project will help us understand how to ensure software runs smoothly and how to identify problems early in the process.

****

Figure 1 Quality Assured.

# Table of Contents

[Abstract 2](#_Toc187069805)

[Table of Contents 3](#_Toc187069806)

[Table of Figure 4](#_Toc187069807)

[List of Tables 5](#_Toc187069808)

[1.Application Selection and Justification: Noon 6](#_Toc187069809)

[2.Requirements Analysis for Noon 7](#_Toc187069810)

[2.1 Functional Requirements: 8](#_Toc187069811)

[2.2 Non - Functional Requirements: 9](#_Toc187069812)

[3.Comprehensive Test Plan 10](#_Toc187069813)

[3.1 Objective 10](#_Toc187069814)

[3.2 Scope 10](#_Toc187069815)

[3.3 Test Objectives 11](#_Toc187069816)

[3.4 Test Strategy 11](#_Toc187069817)

[3.5 Test Schedule 13](#_Toc187069818)

[3.6 Test Resources 13](#_Toc187069819)

[3.7 Test Deliverables 14](#_Toc187069820)

[3.8 Risks and Assumptions 15](#_Toc187069821)

[3.9 Dependencies 15](#_Toc187069822)

[3.10 Exit Criteria 16](#_Toc187069823)

[3.11 Approvals 16](#_Toc187069824)

[3.12 Appendices 16](#_Toc187069825)

[4.Test Cases for Core System Functionalities 17](#_Toc187069826)

[4.1 Test Techniques: Black Box Testing 18](#_Toc187069827)

[4.2 Test Case Management Using the Jira Testing Tool 18](#_Toc187069828)

[4.2.1 Valid Test Case Scenarios 18](#_Toc187069829)

[4.2.2 Invalid Test Case Scenarios 19](#_Toc187069830)

[4.3 Traceability Matrix 20](#_Toc187069831)

[References 22](#_Toc187069832)

# Table of Figure

[Figure 1 Quality Assured. 2](#_Toc187069833)

[Figure 2 main interface of the Noon 7](#_Toc187069834)

[Figure 3 Non-functional requirements 10](#_Toc187069835)

[Figure 4 Agile testing approach 12](#_Toc187069836)

[Figure 5 V-Model Approach 13](#_Toc187069837)

[Figure 6 Test Deliverables 15](#_Toc187069838)

[Figure 7 Black Box Testing 18](#_Toc187069839)

# List of Tables

[Table 1 Traceability Matrix 19](#_Toc187069896)

# 1.Application Selection and Justification: Noon

Noon is a popular online shopping platform in the Middle East, offering many useful features that make it a great choice for software testing. It provides real examples to test different functions, performance, and user experience.

1. **Complete User Journey:** Noon covers everything from browsing products to secure payments and tracking orders.
2. **Different User Scenarios:** The platform supports guest checkout, user accounts, and discount codes.
3. **Third-Party Integration:** Noon works with payment systems, delivery services, and login tools.
4. **Interactive Design:** It includes live product updates and smart search filters.
5. **Works on Web and Mobile(Cross-Platform):** Noon can be tested on different devices and screen sizes.
6. **Regular Updates:** Frequent updates allow us to test new features and ensure stability

# ****2.Requirements Analysis for Noon****

In this section, we analyze the functional and non-functional requirements of Noon based on available documentation, user manuals, and our hands-on experience with the app. This practical experience helped us better understand how the platform operates.

Here is the main interface of the Noon website:

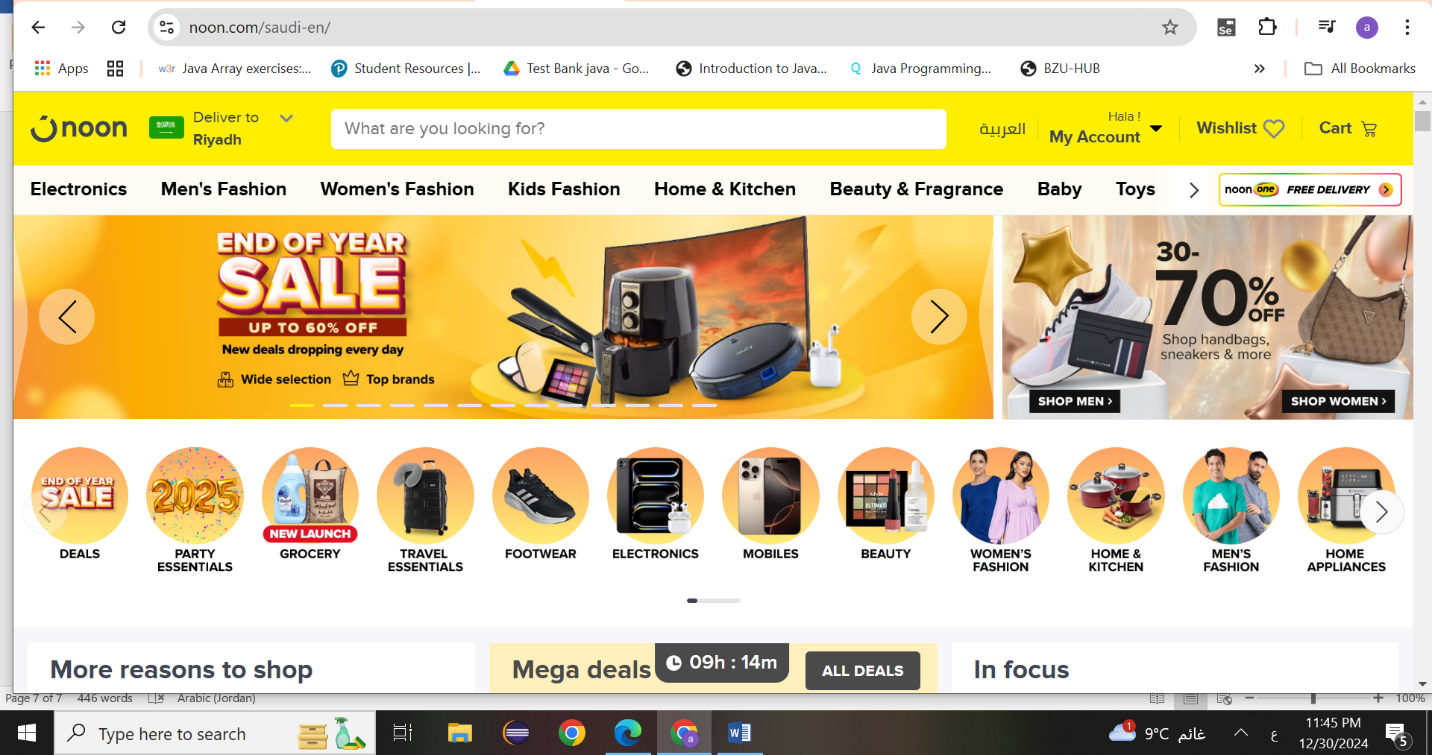


Figure 2 main interface of the Noon

## 2.1 ****Functional Requirements:****

Functional requirements define the essential features and capabilities that the platform must offer to meet user needs and ensure smooth operation.

1. **User Registration and Login**: Users are able to create an account, log in, and manage their profile information.
2. **Product Search and Filtering:** The website allow users to search for products using various filters like category, price range, product rating, and brand.
3. **Shopping Cart and Checkout:** Users are able to add products to the shopping cart, view cart contents, modify quantities, and proceed with checkout.
4. **Order Placement and Payment**: Users are able to place an order and choose from multiple payment options, including credit cards, Additionally, users should be able to pay using more than one bank card, rather than relying on a single bank card.
5. **Delivery Service with GPS Location**: The website allows users to choose a delivery service and specify their delivery address using GPS location, ensuring accurate and efficient delivery to the correct location.
6. **Product Reviews and Ratings:** Users are able to view and submit product reviews and ratings.
7. **Customer Support:** The website provides various channels of customer support, including chat, email, and a help center.
8. **Account Management**: Users are able to reset their passwords, update their contact details, and manage their preferences, ensuring a personalized experience on the platform.
9. **Wish list and Save for Later**: The website allows users to add products to a wish list or save them for later, enabling easy access to items they may want to purchase in the future.
10. **Language Options**: The website allows users to choose between Arabic and English languages for better usability.
11. **Search History**: The website allows users to view their previous searches to easily find products they have looked at before.
12. **Help Center**: The website provides a "Need Help" service where users can get assistance with their issues through various channels such as chat, email, or FAQs.
13. **Return and Refund Policy**: The website provides users with an easy process for returning products and requesting refunds, ensuring customer satisfaction.
14. **Availability of Stock**: The website shows real-time stock availability for products, informing users if an item is out of stock or available for purchase.

## 2.2 Non - ****Functional Requirements:****

Non-functional requirements a set of specifications that describe the system's operation capabilities and constraints.

1. **Performance**: Noon should load quickly (under 3 seconds) and respond fast when you search or place an order.
2. **Scalability**: Noon must handle more users, especially during busy times, without slowing down.
3. **Portability**: Noon works on different devices (phones, tablets, computers) and supports many browsers (Chrome, Safari, etc.).
4. **Compatibility**: Noon shouldn’t have issues working with other apps or tools, like payment systems and delivery services.
5. **Reliability**: Noon should work smoothly with minimal errors or crashes.
6. **Maintainability**: Noon must be easy to update and fix problems quickly without interrupting users.
7. **Availability**: Noon should be available almost all the time, aiming for 99.9% uptime.
8. **Security**: Noon must protect user data with strong security, like encrypted payment details and safe login systems.
9. **Usability**: Noon should be easy to use, with a simple design and easy navigation.

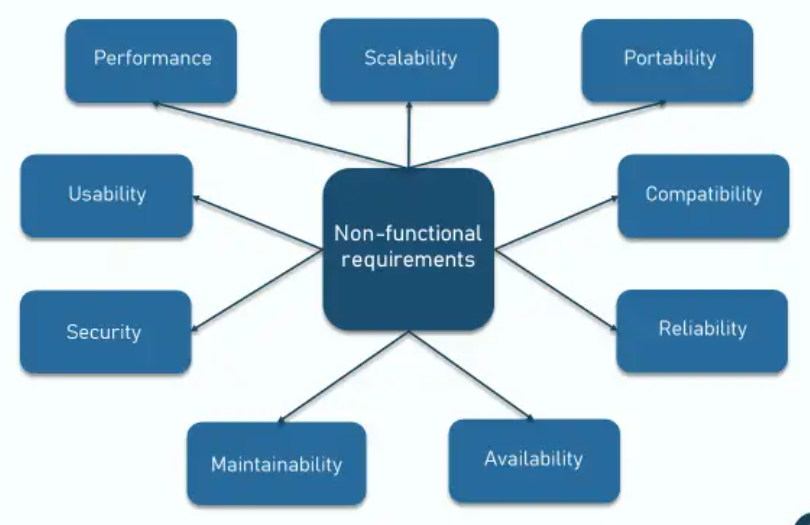


Figure 3 Non-functional requirements

# 3.Comprehensive Test Plan

A Test Plan in software testing is a comprehensive document that outlines the strategy, scope, objectives, resources, and schedule for testing a particular software application. It serves as a roadmap for the testing process and is essential for efficient and effective testing.

## 3.1 Objective

(Asma)

The objective of this test plan is to ensure that the "Noon" e-commerce platform works as expected. It will focus on testing key features like user registration, product search, order placement, payment options, and delivery services. The plan will also check the platform's performance, **Reliability, Maintainability,** and **Usability**, ensuring a smooth experience for users.

## 3.2 Scope

(Dalia)

**Inclusions** Functional testing covers account registration, search functionality, cart management, checkout process, order tracking, and customer support. Performance testing ensures optimal page loading speeds and responsiveness across different browsers and devices. Security testing focuses on data protection, secure login mechanisms. Usability testing evaluates user experience, ease of navigation, and compliance with accessibility standards

**Exclusions**: The test plan does not include mobile applications (iOS and Android), testing of third-party integrations such as payment gateways and shipping APIs, or any experimental or under-development features.

## 3.3 Test Objectives

(Rayan)

**1-Functional Testing Objectives**

* Ensure all primary features, such as user registration, login, shopping cart functionality, and checkout, operate without any errors.
* Validate the GPS-based delivery address system for accuracy and efficiency.
* Test the functionality of multi-card payment options to confirm seamless transaction handling.

**2-Performance Testing Objectives**

* Verify the system\u2019s ability to handle up to 1,000 concurrent users while maintaining response times under 2 seconds.
* Assess the application\u2019s responsiveness and stability during high-traffic events, such as promotional sales.

**3-Security Testing Objectives**

* Perform penetration testing to detect vulnerabilities, including SQL injection and cross-site scripting (XSS).
* Confirm that encryption protocols effectively protect user data during sensitive operations, such as payments and login.

**4-Usability Testing Objectives**

* Evaluate the user interface for simplicity, intuitiveness, and overall user experience.
* Ensure the platform meets accessibility standards, and gather user feedback to identify potential areas for improvement.

## 3.4 Test Strategy

(Asma)

**Approach:** We will use an **Agile testing approach** for the "Noon" e-commerce platform. Agile focuses on **flexibility, teamwork**, and **continuous testing**. It allows us to test **different parts of the system at the same time**, fix issues quickly, and adapt to changes easily. This helps us improve the platform faster and meet user needs effectively.

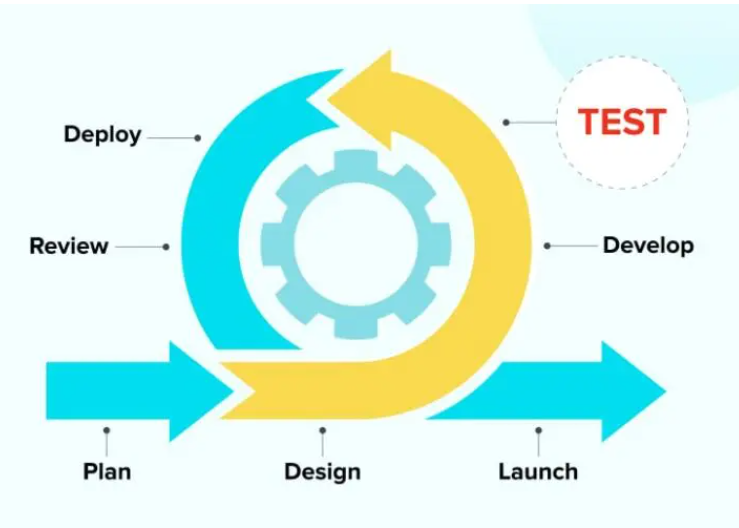
****

Figure 4 Agile testing approach

**Testing Levels (V-Model Approach):**

We will use the V-Model for testing the "Noon" e-commerce platform. In this approach, every development stage has a matching testing stage to catch problems early.

1. **Acceptance Testing**: Checks if the final product meets business goals and user needs.
2. **System Integration Testing**: Ensures that all parts of the system work well together as a complete product.
3. **Component Testing**: Tests how different modules work together based on their design.
4. **Unit Testing**: Focuses on testing small pieces of code separately to make sure they work correctly.

The V-Model helps us test step-by-step, making it easier to find and fix issues early in the process.



Figure 5 V-Model Approach

## 3.5 Test Schedule

(Dalia)

Provide a timeline for testing activities, including start and end dates for each testing phase.Test Schedule for Noon Website

**Examples**:

1. Functional Testing:

Dates: [ January 1 to January 7], Activities: Testing account registration, login, search functionality, cart management, checkout process, order tracking, and customer support.

1. Performance Testing:

Performance testing is scheduled from January 8 to January 10. During this phase, the focus will be on measuring page loading speeds and the responsiveness of the website under normal and high-traffic conditions to ensure smooth performance.

1. Security testing:

Security testing will take place from January 11 to January 12. The primary activities include verifying data protection measures, testing secure login mechanisms.

1. UsabilityTesting**:**  
   Usability testing is planned for January 13. This phase aims to evaluate the overall user experience, ease of navigation, and compliance with accessibility standards to ensure a user-friendly design and functionality.

## 3.6 Test Resources

(Rayan)

**Hardware Resources:**

* Computers and mobile devices (smartphones, tablets) for testing across various platforms.
* A staging server and a separate database server to mimic real-world scenarios.

**Software Resources:**

* Testing tools: Selenium for automated functional testing, JMeter for performance testing, and OWASP ZAP for security testing.
* Browsers (e.g., Chrome, Firefox, Safari) for cross-browser compatibility testing.

**Personnel:**

* QA team members:
  + Asmaa Ankoush (Test Lead)
  + Dalia Hamed (Automation Tester)
  + Rayan A Hamayel (Performance Tester)

**Test Environments:**

* Dedicated test environments for functional, performance, and security testing.
* Devices for testing: Smartphones with iOS and Android, tablets, and desktops.

**Additional Tools:**

* Bug tracking system: Jira for managing reported issues.
* Collaboration tools: Slack for communication and coordination among team members.

## 3.7 Test Deliverables

(Asma)

During testing, we will create key documents to track progress and ensure quality:

1. **Test Plan:** A document explaining what, how, and when testing will happen.
2. **Test Strategy:** A guide on how testing will be done.
3. **Test Cases:** Steps to check if each part of the software works correctly.
4. **Test Data:** Sample information used for testing.
5. **Test Scenario:** Different situations to see how the software reacts.
6. **Requirement Check:** A list to ensure all requirements are tested.
7. **Bug Report:** A record of problems or errors found.
8. **Test Summary:** A short report showing the main results of testing.
9. **Final Report:** A final document saying testing is finished and the software is ready to use.

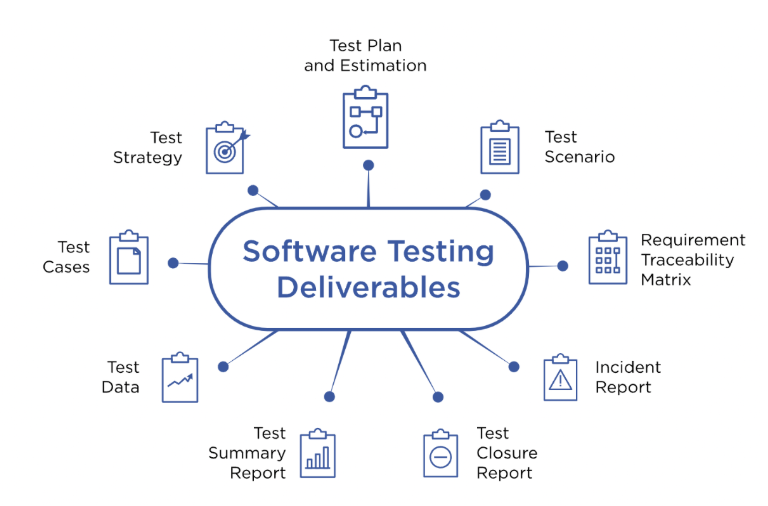


Figure 6 Test Deliverables

## 3.8 Risks and Assumptions

(Dalia)

**Assumptions:** The staging environment will match the production environment in functionality, data, and configurations. Comprehensive and sufficient test data will be provided to cover all required scenarios. All features and modules will be deployed before testing begins, and the development team will promptly address any critical issues. Necessary tools and resources for performance, security, and usability testing will be fully available and functional.

**Risks:** The risks for testing include limited availability of test data, which may hinder the ability to test edge cases effectively. Differences between the staging and production environments could lead to undetected issues. Delays in feature readiness may disrupt the testing schedule, and resource constraints such as limited tools or team members could affect the quality of testing. Unexpected technical issues, such as system errors or downtime, may cause delays in the timeline. Additionally, incomplete documentation for features could result in missed or inaccurate test cases.

## 3.9 Dependencies

(Rayan)

## 3.10 Exit Criteria

(Asma)

**Exit criteria** outline the conditions that must be met before testing can be considered complete. These criteria help determine whether the software is ready for release or if further testing is required. Common exit criteria include:

1. **Test Coverage:** All planned test cases have been executed, covering a specific percentage of key requirements (e.g., user login, payment processing, order tracking).
2. **Defect Resolution**: All critical and high-priority defects (e.g., issues with user registration, data security, payment processing) must be resolved or documented.
3. **Pass Rate**: A minimum percentage of test cases must pass (e.g., 90%).
4. **Documentation Completion**: All testing documentation (test cases, results, defect logs) should be completed and reviewed.
5. **Stakeholder Approval**: Stakeholders must review and approve the testing results and the quality of the product.
6. **Performance Criteria**: The software must meet defined performance benchmarks (e.g., load times, response times).

## 3.11 Approvals

(Dalia)

The test plan for Noon’s website requires approval from:

1. **Project Manager**: To ensure alignment with project goals and requirements.
2. **QA Manager**: To verify the accuracy and completeness of the tests.
3. **Development Team Lead**: To ensure resources and features are ready for testing.

## 3.12 Appendices

(Rayan)

The appendices provide additional resources and references used throughout the project. These supplementary materials support the main content and include the following:

**Glossary**

* **QA (Quality Assurance):** A systematic process to ensure a product meets specific quality standards.
* **Selenium:** An automation tool for web application testing.
* **JMeter:** A performance testing tool used to measure the efficiency of web applications.
* **OWASP ZAP:** A security tool used to identify vulnerabilities in web applications.

**Test Data**

* Sample user data for testing login functionality:
  + Username: testuser1, Password: Test@123
  + Username: testuser2, Password: Test@456
* Sample product data for cart and checkout testing:
  + Product 1: Item Name “Wireless Headphones”, Price: $50
  + Product 2: Item Name “Laptop Sleeve”, Price: $20

# 4.Test Cases for Core System Functionalities

## 4.1 Test Techniques: Black Box Testing

**Black-box testing is a type of software testing in which the tester is not concerned with the software’s internal knowledge or implementation details but rather focuses on validating the functionality based on the provided specifications or requirements, for testing the Noon website, we will perform black-box testing by interacting with the site. We will not conduct white-box testing as we don't have access to the source code.**

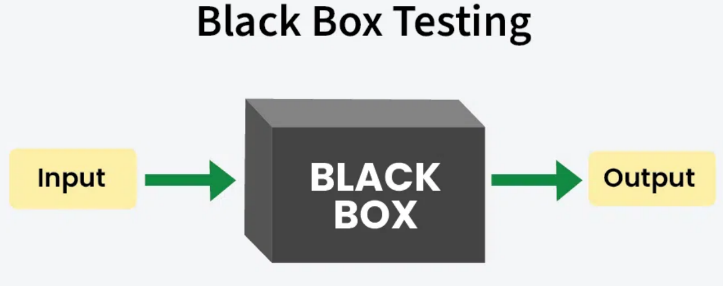
****

Figure 7 Black Box Testing

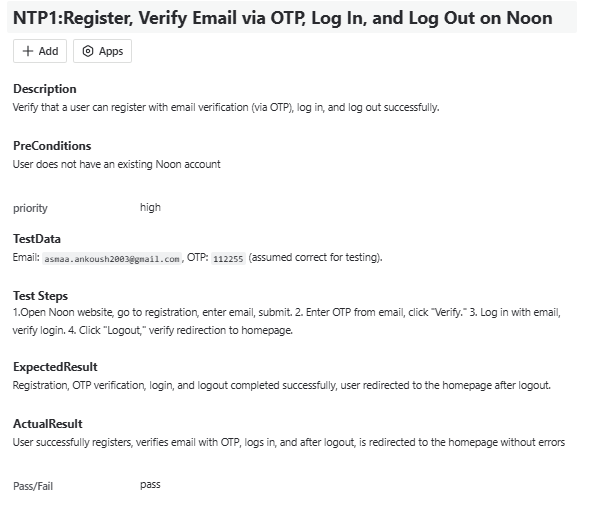
## ****4.2 Test Case Management Using the Jira Testing Tool****

JIRA is a software testing tool developed by the Australian company Atlassian. It is a bug tracking tool that reports all the issues related to your software or mobile apps. In this context, we will use JIRA to manage and organize the test cases for the noon website.

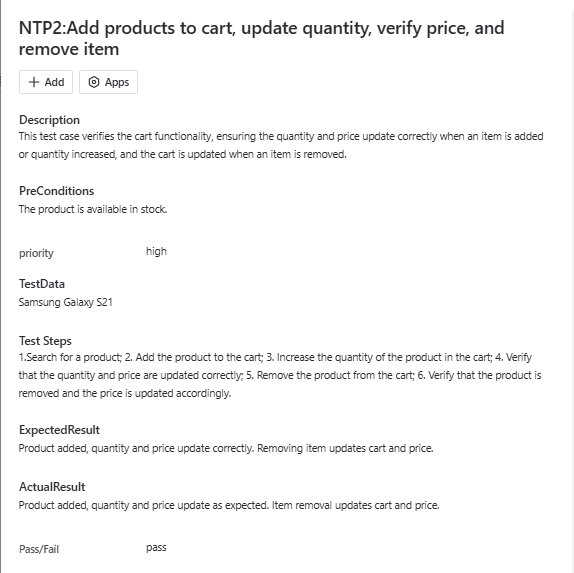
### 4.2.1 Valid Test Case Scenarios

This section covers test cases that should pass, ensuring the system works as expected.

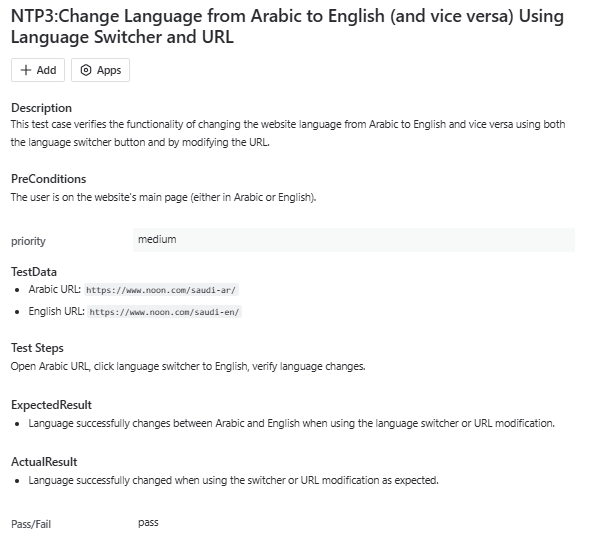
1. Register, Verify Email via OTP, Log In, and Log Out on Noon



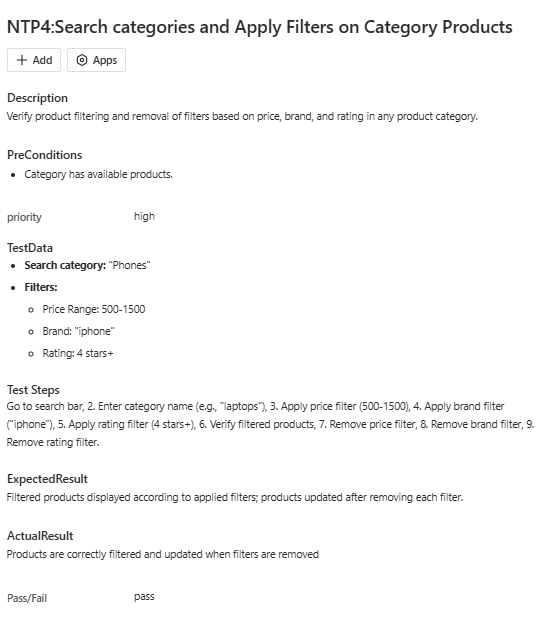
1. Add products to cart, update quantity, verify price, and remove item.



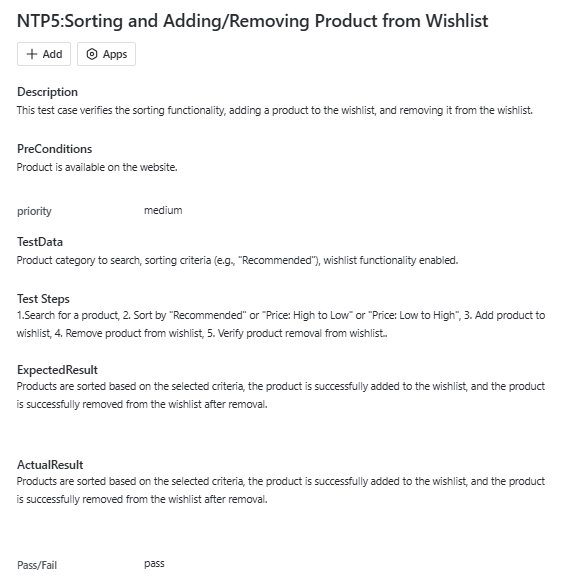
1. Change Language from Arabic to English (and vice versa) Using Language Switcher and URL



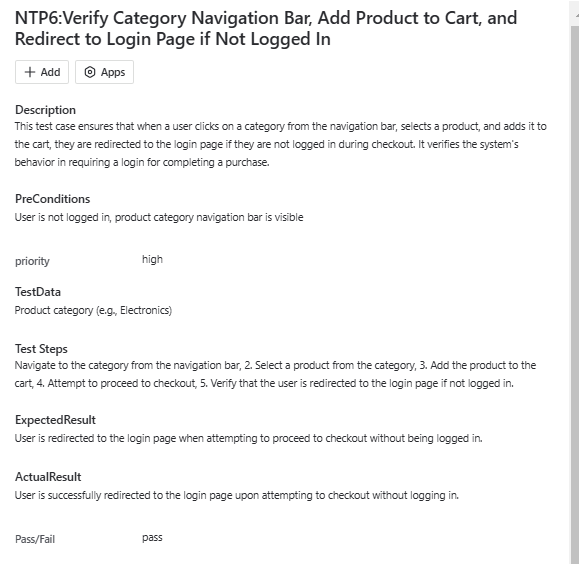
1. Search categories and Apply Filters on Category Products



1. Sorting and Adding/Removing Product from Wishlist.



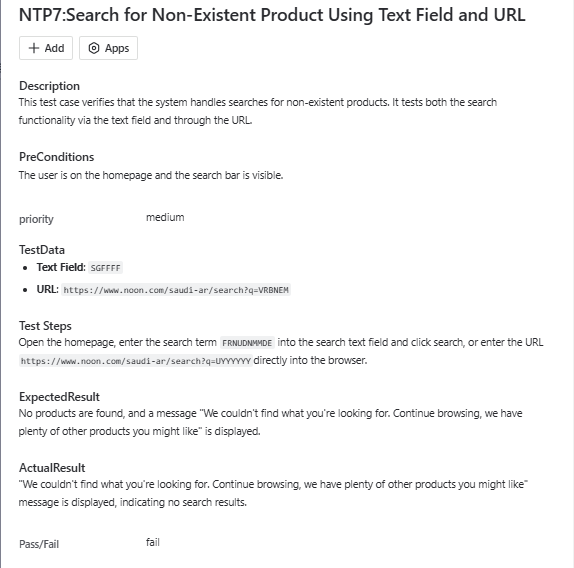
1. Verify Category Navigation Bar, Add Product to Cart, and when go to checkout, redirect to Login Page if Not Logged In

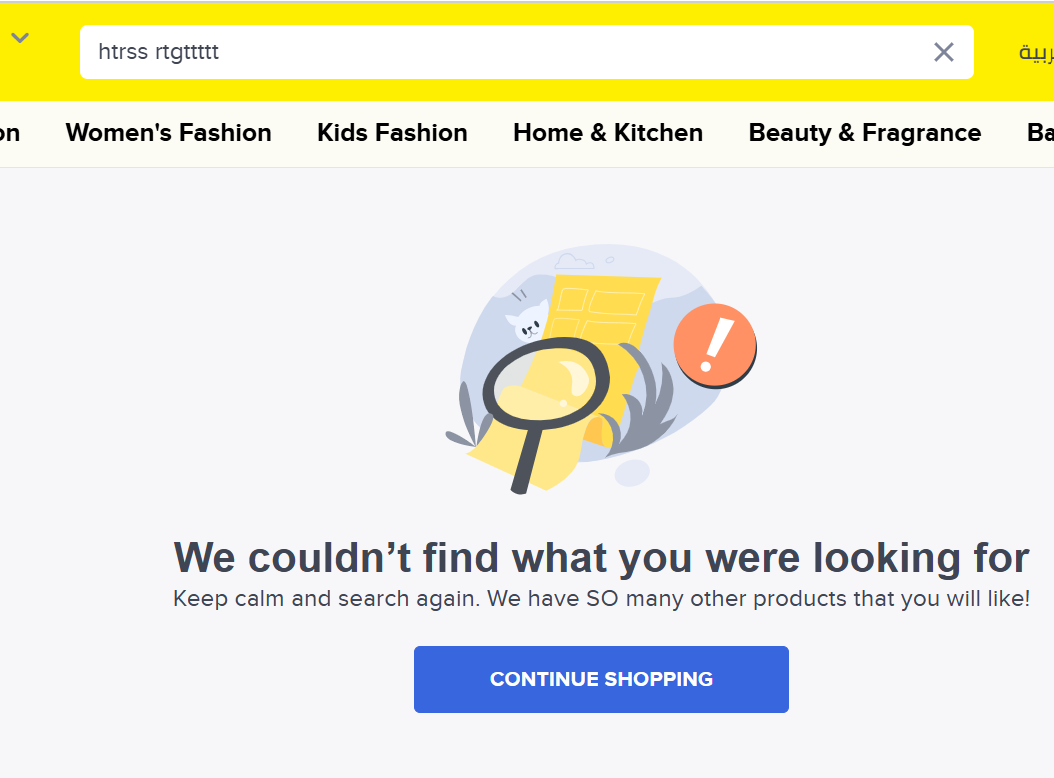


### 4.2.2 Invalid Test Case Scenarios

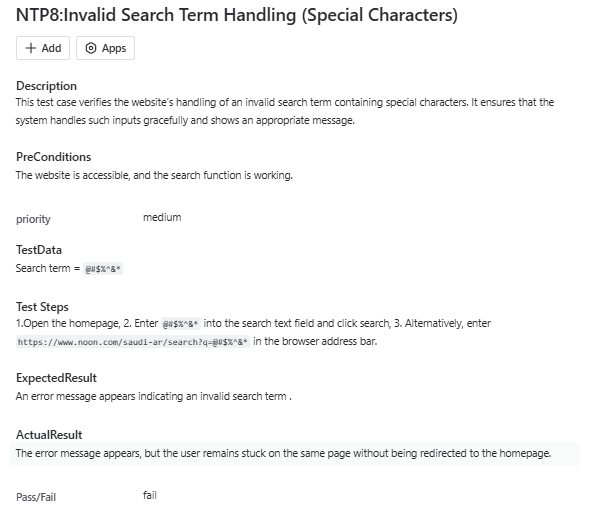
This section covers test cases that should fail, checking for system errors or unexpected behavior.

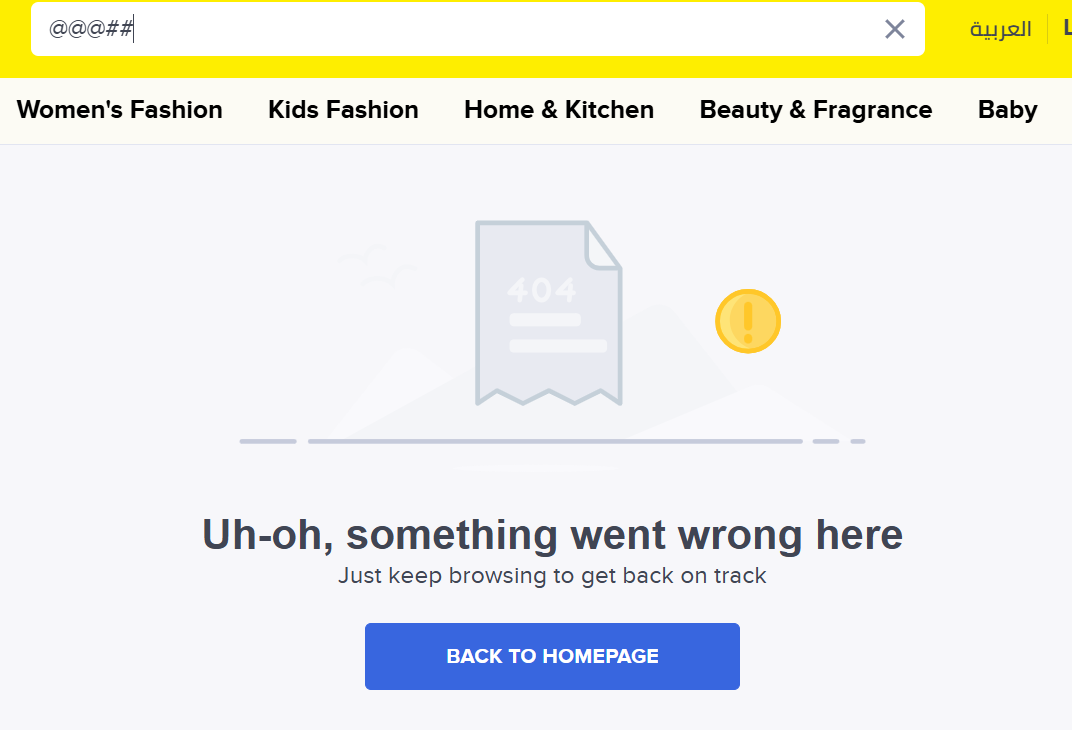
# 1.Search for Non-Existent Product Using Text Field and URL



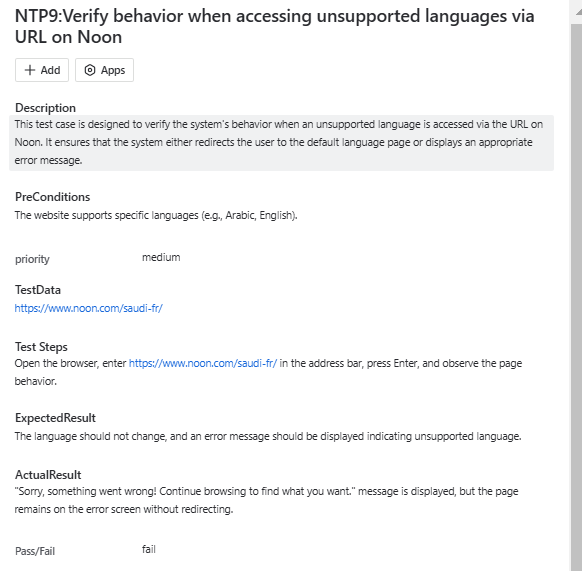


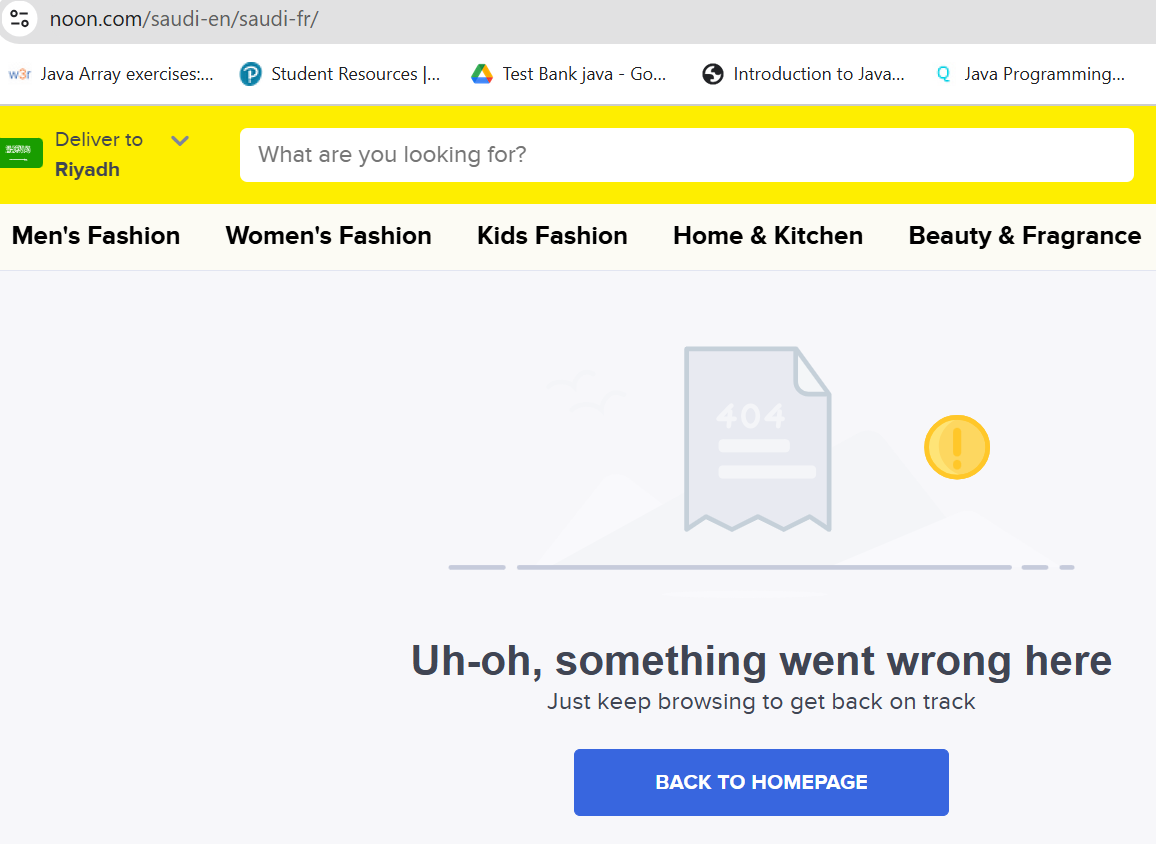
2. Invalid Search Term Handling (Special Characters)



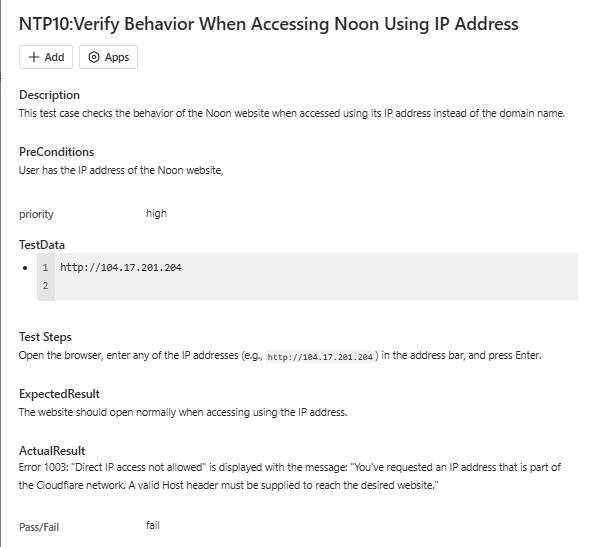
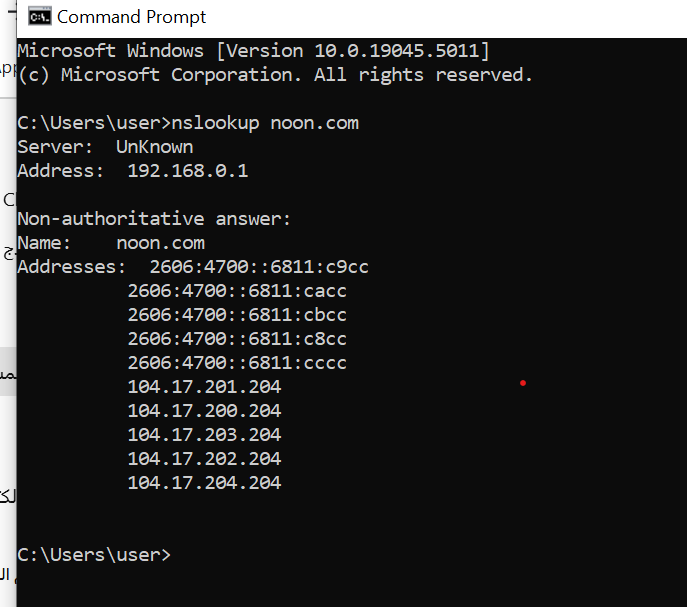


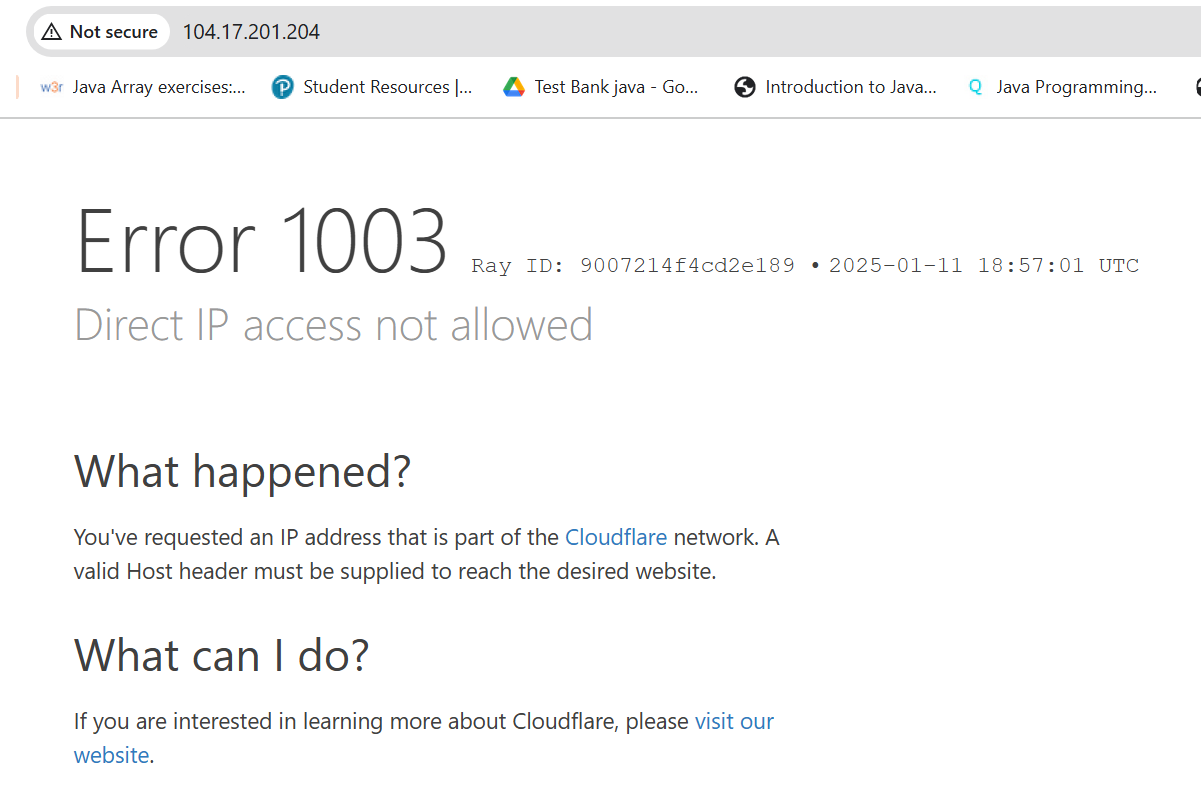
# 3. Verify behavior when accessing unsupported languages via URL on Noon



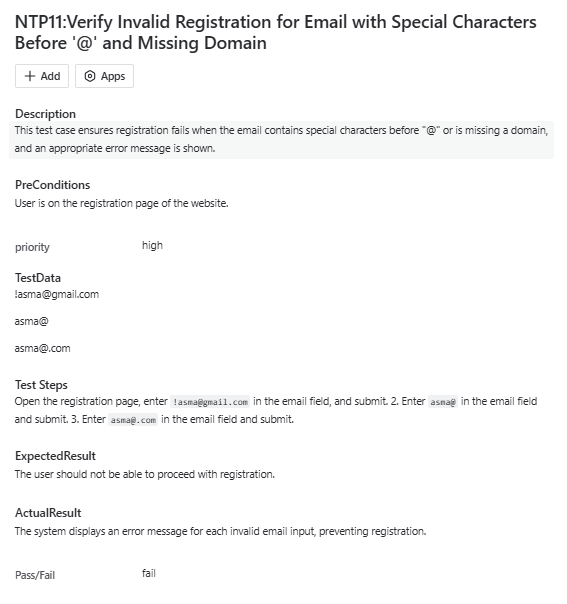


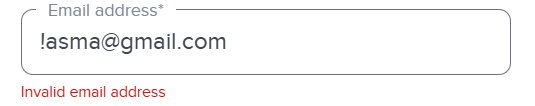
# Verify Behavior When Accessing Noon Using IP Address

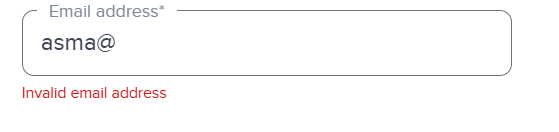
 

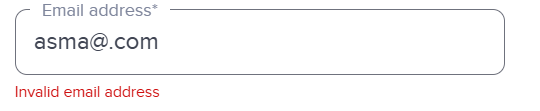


1. Verify Invalid Registration for Email with Special Characters Before '@' and Missing Domain

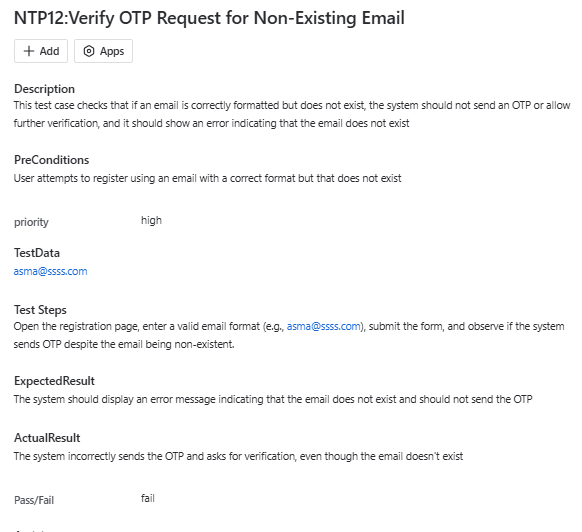


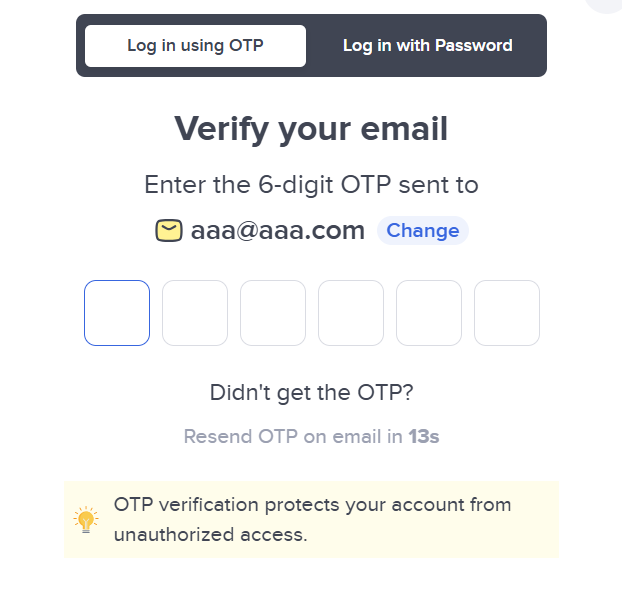






6. Verify OTP Request for Non-Existing Email





## 4.3 Traceability Matrix

The Traceability Matrix links test cases to their corresponding requirements to ensure all requirements are tested.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | **Requirement ID** |  |  | | --- | |  | | |  | | --- | | **Functional Requirement** |  |  | | --- | |  | | |  | | --- | | **Test Case ID** |  |  | | --- | |  | | |  | | --- | | **Test Case Summary** |  |  | | --- | |  | | | **Status** | | --- |  |  | | --- | |  | |
| R1 | |  | | --- | | User Registration and Login |  |  | | --- | |  | | NTP1 | |  | | --- | | Register, Verify Email via OTP, Log In, and Log Out on Noon |  |  | | --- | |  | | PASS |
| R3 | |  | | --- | | Shopping Cart and Checkout |  |  | | --- | |  | | NTP2 | |  | | --- | | Add products to cart, update quantity, verify price, and remove item |  |  | | --- | |  | | PASS |
| R10 | |  | | --- | | Language Options |  |  | | --- | |  | | NTP3 | |  | | --- | | Change Language from Arabic to English (and vice versa) Using Language Switcher and URL |  |  | | --- | |  | | PASS |
| R2 | |  | | --- | | Product Search and Filtering |  |  | | --- | |  | | NTP4 | |  | | --- | | Search categories and Apply Filters on Category Products |  |  | | --- | |  | | PASS |
| R2 | |  | | --- | | Product Search and Filtering |  |  | | --- | |  | | NTP5 | |  | | --- | | Sorting and Adding/Removing Product from Wishlist |  |  | | --- | |  | | PASS |
| R6 | |  | | --- | | User Registration and Login |  |  | | --- | |  | | NTP6 | |  | | --- | | Verify Category Navigation Bar, Add Product to Cart, and Redirect to Login Page if Not Logged In |  |  | | --- | |  | | PASS |
| R2 | |  | | --- | | Product Search and Filtering |  |  | | --- | |  | | NTP7 | |  | | --- | | Search for Non-Existent Product Using Text Field and URL |  |  | | --- | |  | | FAIL |
| R2 | |  | | --- | | Product Search and Filtering |  |  | | --- | |  | | NTP8 | |  | | --- | | Invalid Search Term Handling (Special Characters) |  |  | | --- | |  | | FAIL |
| R10 | |  | | --- | | Language Options |  |  | | --- | |  | | NTP9 | |  | | --- | | Verify behavior when accessing unsupported languages via URL on Noon |  |  | | --- | |  | | FAIL |
| R4 | |  | | --- | | Product Search and Filtering |  |  | | --- | |  | | NTP10 | |  | | --- | | Verify Behavior When Accessing Noon Using IP Address |  |  | | --- | |  | | FAIL |
| R5 | |  | | --- | | User Registration and Login |  |  | | --- | |  | | NTP11 | |  | | --- | | Verify Invalid Registration for Email with Special Characters Before '@' and Missing Domain |  |  | | --- | |  | | FAIL |
| R5 | |  | | --- | | User Registration and Login |  |  | | --- | |  | | NTP12 | |  | | --- | | Verify OTP Request for Non-Existing Email |  |  | | --- | |  | | FAIL |

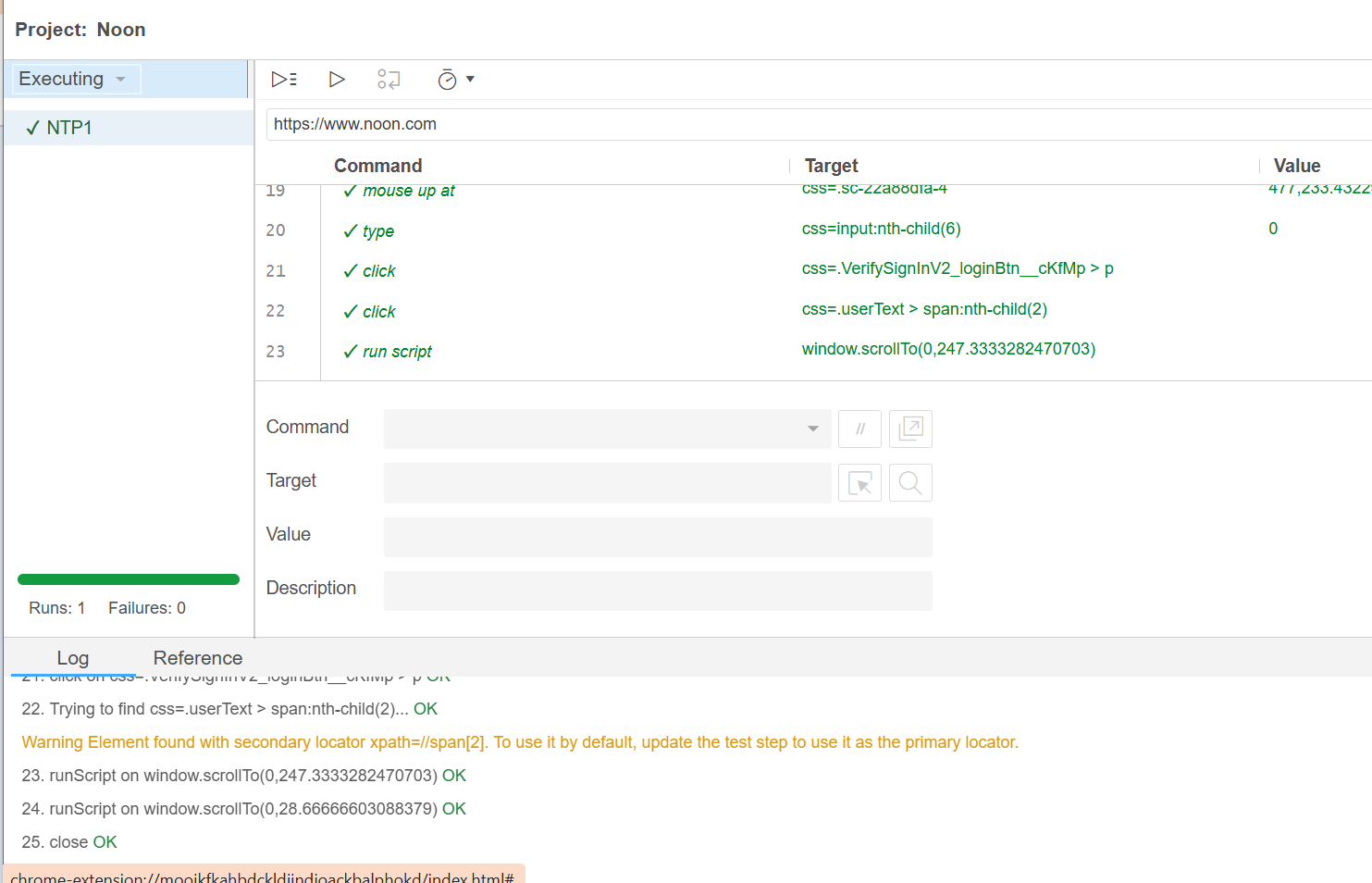
5.Automating Valid Test Cases Using Selenium for GUI Testing.

Selenium is an open-source tool used for automating web browser interactions. It allows testers to write scripts to verify web application functionality across different browsers and platforms.



Valid Test Cases :

* + 1. NTP1: Register, Verify Email via OTP, Log In, and Log Out on Noon



NTP2 : Add products to cart, update quantity, verify price, and remove item

NTP3

NTP4

NTP5

NTP6

# References

1. User manuals and documentation from Noon’s official website.
2. Online resources on QA methodologies and best practices.
3. Tools documentation: Selenium, JMeter, and OWASP ZAP user guides.