PART 1 QA CHALLENGE: Andrés Vergara

TEST PLAN DESIGN

Scope

The scope defines the areas of the system that will be tested, ensuring that both the backend and frontend meet functional and non-functional requirements.

The scope of the ticket includes backend validation through testing for user authentication, product management and order processing, covering correct responses, error handling, performance under load, and security aspects such as encryption and credential management. On the frontend, the interface needs to be evaluated by verifying the authentication flow, dashboard functionality, dynamic product navigation and loading, and the order processing flow, ensuring an intuitive and efficient user experience.

More detailed:

Backend: Testing will be conducted covering user authentication, product management, and order processing. These tests will include endpoint validation to ensure correct responses for valid requests and proper error handling for invalid ones. Performance under simulated load conditions will also be evaluated and proper credential handling.

Frontend: The user interface will be validated, focusing on the complete user authentication flow, the correct functionality of the dashboard, dynamic loading and navigation of the product listing, and the order processing flow. The user experience will be assessed to ensure it is intuitive, fast and functional.

Objective

The objectives outline the goals of the testing process, aimed at ensuring the product's quality in terms of functionality, security, performance and usability.

The main objective is to ensure that the APIs and user interface operate without errors and meet the required specifications. And the implementation must be a satisfactory user experience.

Resources

Resources include tools, environments, and datasets necessary to perform the tests efficiently and effectively.

- **Tools:** Postman will be used for backend functional testing.For the frontend, Cypress will be used for test automation, and ensure compatibility across different browsers

  Browsers like: Chrome, Opera and Microsoft Edge **(Ensure you have the latest version as of the date of reading this)**

- **Testing Environment:** The testing environment will consist of a staging server with access to the APIs and the frontend. This environment will replicate production conditions as closely as possible to ensure accurate results.

  In this case, we will ensure to test it in the corresponding environment, but it is always necessary to confirm that we are in the correct environment for testing; in this instance, it will be a testing host.

- **Datasets:** Test data will include user accounts with varied permissions, products in different states, and orders in various phases (processed, pending, canceled). This will allow for thorough coverage of multiple scenarios during testing.

  Different scenarios can occur depending on whether you are an admin or not, for example. Therefore, visualizing and trying to gather all possible data related to what needs to be tested opens the doors to different testing opportunities

Risk

QA should be aware of risks like delays in getting test environments or data, which can be managed by setting up local environments or using mock APIs. Another risk is missing critical bugs, so it's important to test key functionalities early. And compatibility issues.

Risks will always exist when testing, and it is essential to have a clear expected result to observe them in the best way. Understanding this will help in decision-making based on the current result. There are different levels of risks that could be encountered. A critical bug that directly affects the functionality being tested might require an immediate decision to fix it. Additionally, there could be a low-priority bug, such as a design issue, which can be addressed later. Or, there might be a bug that affects functionality but does not prevent its correct completion.

So, the QA should always be aware of the level of risk involved and also be part of the decision-making process based on these risks. Having a clear expected result will increase the identification rate and help reduce the risk of critical bugs in production.

Deliverables

At the end of the process, a detailed report of test cases and their results, including identified defects and proposed solutions, will be delivered. Automation scripts will also be provided to enable efficient regression testing. Additionally, a report will document identified issues and mitigation strategies.

Note: Depending on the type of testing, it is important to document the tested feature/software. If it is a new behavior within the project, providing documentation and specific steps for the new feature can be valuable. This documentation serves as a backup and ensures that anyone can understand how to execute it. It can also be used for various purposes, such as future reference, troubleshooting, or onboarding new team members.