Arnaldo Medina
Chicks Group Inc.

Test Plan Design

This document outlines the plan for designing, developing, and executing test cases for Chicks Gold, as part of the QA Challenge. Its primary goal is to detail the **scope**, **objectives**, **resources**, identified **risks**, and **deliverables** associated with testing activities. The aim is to ensure a systematic and structured approach to validate that all system components meet the established requirements, minimizing risks and ensuring the quality of the final product. This plan serves as an essential resource to coordinate and document all aspects of the testing process for evaluation purposes.

Scope

This test case design plan covers both the Frontend and Backend components of the challenge under test.

For the Frontend, the scope includes validating the following pages: Login, Dashboard, Products, and Order pages. This involves testing responsive, smooth navigation, compatible across different browsers, and API connection during Frontend tests. All defects will be identified and reported.

For the Backend, the focus is on validating API endpoints, database operations, and internal logic. This includes performance, scalability, and security testing to ensure the system can handle expected and unexpected loads efficiently and securely. These test will be executed with postman.

Objectives

The primary objective of this challenge is to ensure that the web meets all defined functional and non-functional requirements. This includes that each module operate as expected, and interactions are correct. Additionally, it aims to identify defects at an early stage of development to minimize risks and costs associated with errors in production.

Other objectives are ensuring comprehensive test coverage for edgy cases and non-planned scenarios. The plan also aims to generate clear, precise, and easily understandable documentation during the challenge for quick evaluation.

Resources

The resources and tools to be used during the development and execution of test cases are as follows:

- Browsers: Google Chrome version: 131.0.6778.140, Opera version: 115.0.5322.77, Microsoft Edge version: 131.0.2903.99.
- Tools: Visual Studio Code version: 1.95.3, Postman version: 11.22.1, .NET 8.0 SDK version 8.0.404, Node.js version: 22.2.0, Cypress version 13.17.0, GitHub version 3.4.5, Swagger UI version 1.
- Windows: 11 Home 23H2

Links required/expected for the challenge:

- https://github.com/SAMO-Technologies/qa-challenge
- https://github.com/MedinaArnaldo

Risks

Testing is crucial for software development, and test cases play a vital role in this process. Test cases are designed to cover high-priority and high-risk areas of the software. By testing critical functionalities and scenarios, testers can assess the potential impact of defects on the overall system. So, for this challenge, one of the main purposes to plan different test cases is to discover defects or issues in the software. Through the systematic testing of different components, QA can determine whether there are discrepancies between expected and actual behavior. This early detection of defects is important in identifying and fixing issues before the software becomes available to users.

With this clarified, all defects (commonly called bugs), could be classified with they own priority:

- Low: Doesn’t affect functionality, and others defects must be attended before this.
- Medium: Doesn’t limit functionality, but it’s important.
- High: Limits the main functionality, and it’s crucial to fix it.
- Critical: Blocks the main functionality, it’s the highest priority.

Deliverables:

The following deliverables will result from the development of this plan:

- Test Case Plan Design: A detailed document describing the scope, objectives, resources, risks, and deliverables.
- Test Case Development: A well-organized repository including detailed descriptions, expected results, preconditions, and execution steps for all test cases.
- Test Execution and Reporting: Detailed summaries of test case execution, including metrics such as success rates, failures, and blockers.
- Automation: Scripts, configurations, and files associated with automated test cases.
- Access to the repository where the challenge is uploaded.