**Test Plan**

### ID: Test\_Plan\_001\_QA\_Challenge

# Introduction

This Test Plan will carry out an analysis and testing of the web app and API provided by Chicks Gold, as part of a QA Challenge.

Functional, API and integration tests will mainly be implemented. Test cases will be designed for both the frontend and backend. Their execution will be done manually and with automation, using Selenium with TestNG for the frontend and Postman for the backend. The results obtained will be documented, as well as the defects found.

The sections of the Test Plan are:

* Scope
* Objetives
* Resources
* Test case information
* Risks
* Deliverables
* Terms

# Objectives

The objective is to validate and verify that both the web app and the API work according to the specifications and documentation presented, detecting and reporting the incidents found, in order to increase the quality of the product.

Ensure that any errors found can be fixed before it is used by the end user.

Obtaining software with higher quality and that the users who are going to use it have the best possible experience.

# Scope

Tests will be designed and executed for both frontend and backend functionalities.

The types of tests to be implemented will be only: functional tests, integration tests, accessibility tests, validation tests, API tests.

## Frontend

The following pages will be analyzed: Home page, Login page, Dashboard page, Products page and Orders page.

For these pages, manual test cases will be implemented to validate the content and functionality of the website as well as the integration with the API.

The designed tests will be automated using Selenium, TestNG and RestAssured for the connection to the API during frontend tests.

All tests will be documented and the issues found will be reported. The tests will be finished when these activities are completed.

## Backend

The following entities will be analyzed: User, Product, Order.

For each entity, tests will be designed to validate its funcionality, using Swagger documentation to define which Endpoints to test, specifically:

* User: Login user.
* Product: Create a new product, update a product, delete a product, show all products and show product by ID.
* Order: Create a new order, update an order, delete an order, show all orders and show an order by ID.

These tests will be executed using Postman, the execution of each Endpoint will be automated and test scripts will be implemented. Their results will be documented in Test Cases and the defects found will be reported.

The tests will be finished upon completion of these activities.

# Resources

The resources required to carry out the execution of the Test Plan will be the following:

* Environment: Windows 10, internet access, project running locally (API and web app)
* Tools: Dotnet 8, Node, Java 17, Maven. VS Code, Postman.
* Dataset: access to test data to include in Postman Runners ("Data set for Postman - JSON files" folder).
* Browsers: Firefox, Chrome, Edge, Opera.

Useful links:

* Project repository: <https://github.com/Melisayunis/QA-Challenge-Chicks-Gold>
* QA Challenge repository: <https://github.com/SAMO-Technologies/qa-challenge>
* Postman workspace ID: 67d5cb95-8e19-4799-abe4-899582519bcb

# Test case information

## Priority classification

Priority will be defined as:

* High: highest execution priority, the functionality is a priority.
* Medium: medium execution priority, the functionality is important, but not essential.
* Low: low execution priority, the test will be executed after the previous ones are completed. Its information is required but not blocking.

## Severity classification

The severity of a failure can be classified as:

* Critical: the failure found is in an essential functionality, or is blocking.
* High: the failure found is in a main functionality.
* Medium: the failure found does not limit the functioning of other activities, but is important.
* Low: the failure found does not directly influence a functionality.
* Very low: the failure found does not influence a functionality, mostly visual errors.

# Risks

4. **Risks:** Identify potential risks and mitigation strategies.

Id Riesgo, Nombre, Descripción, Estado inicial, Consecuencias, Probabilidad de ocurrencia, impacto, prioridad, clasificación, síntomas, tolerancias, acciones preventivas y correctivas.

# Deliverables

Upon completion of the design, execution and documentation of the requested tests, the following must be delivered:

* Test plan
* Test set and test cases.
* Bug report.
* Evidence.

API:

* Access to the Postman workspace.
* Test data set to run on the Runners.
* Obtained results.

Web app:

* Access to the repository where the test automation has been implemented.
* Documentation of the execution of the tests.
* Obtained results.

# Terms

**API**: Application Program Interface.

# Implementation

The activities will be developed by:

* Design (test plan, test set, test case): Melisa Yunis
* Test execution: Melisa Yunis
* Reporting and documentation: Melisa Yunis
* API automation: Melisa Yunis
* Web app automation: Melisa Yunis