QA Challenge Test Plan Chicks Application

**July 31st, 2025**

# Objective

This document outlines the QA strategy for evaluating a full-stack web application as part of a **QA challenge for the Chicks Gold QA Engineer position**. The testing is performed by **Ingrid Garcia** and is intended to demonstrate proficiency in both backend and frontend validation, including defect discovery, scenario coverage, and test documentation.

The objective is to verify that the system behaves reliably under normal and edge conditions, ensure user workflows are functional and secure, and highlight any usability or integration issues. Both the API layer (C#) and the UI (ReactJS) are tested across various user flows, with documentation provided for all findings.

# Scope

This test plan covers **comprehensive manual testing** of the web application backend and frontend:

#### Backend (C# APIs)

* **Authentication:** Login, token validation, error handling
* **Product Management:** Creation, duplication handling, bad payload rejection
* **Order Processing:** Order creation, updates, cancellation, invalid flow handling

#### Frontend (ReactJS UI)

* **Login & Dashboard:** Valid/invalid logins, error UI, dashboard visibility
* **Product & Order Display:** Data display and interaction
* **UI Checks:** Responsiveness, session handling, visual feedback

Test scenarios include **happy path**, **negative**, **security**, and **boundary testing**.

# Resources

**Tools**

* Postman
* Swagger UI
* React Dev Server
* Chrome DevTools
* Google Docs & Sheets
* Google Drive
* Screenshot/Video tools

**Environments**

* Backend: <http://localhost:5044>
* Frontend: <http://localhost:3000>

#### Payload Samples

**Login**

{ "username": "string", "password": "string" }

**Product**

{ "id": 1, "name": "Product A", "price": 100 }

**Order**

{ "id": 1, "productName": "Product A", "quantity": 2, "status": "Pending" }

# Risks

| Risk | Impact | Mitigation |
| --- | --- | --- |
| Swagger showed the wrong port (5000 instead of 5044), causing confusion | Medium | Confirmed correct port manually via console and adjusted Postman environment setup accordingly |
| Backend failed to return proper status codes in certain error scenarios (e.g., invalid content-type) | High | Documented as failed test cases; included expected vs actual responses in defect notes |
| Product endpoint allowed duplicate product IDs or duplicate entries with different names | High | Marked as a data integrity issue; backend lacks proper ID validation |
| Token-based authentication did not properly restrict access to protected endpoints | High | Documented failed test cases showing 200 OK returned when token was missing or invalid |
|  |  |  |

# Deliverables

* Full Test Plan
* Backend & Frontend Test Cases
  + Link: <https://docs.google.com/document/d/1NuSQRYZxOzYD_bnQdkqVg7DVVZj7jki1vekXGgvyztA/edit?usp=sharing>
* Backend & Frontend Test Results:
  + Link: <https://docs.google.com/spreadsheets/d/1sm_seiFet8hSSVNLEqwHFPCFb3BVk_6BZcHcs-5V1Qs/edit?usp=sharing>
  + Defect Notes (inside spreadsheet)
  + Proof of Testing: <https://drive.google.com/file/d/1Lq45xq_-XcbQ4gJymd2xrYK9Qau9Csoh/view?usp=sharing>
* Postman collection:
  + <https://ingarciar-4502416.postman.co/workspace/Ingrid-Garcia's-Workspace~79df3770-db8b-43f5-8b85-45ca8b1d79d4/collection/47265845-cad8ec42-8c6d-42f6-85e7-697c72d002a3?action=share&creator=47265845&active-environment=47265845-ed6688ab-1458-43d8-9cb0-d62b405b627d>