

## **QA Take-Home Assignment – Full Report**

### **Part 1: Manual Testing – Test Cases and Bug Reporting**

Website Under Test: <https://www.demoblaze.com>

Test Focus: Browsing for a product → Adding to cart → Viewing the cart → Placing an order

Test Environment: Google Chrome, Desktop

Date: 05/29/2025

#### **Test Case 1:** Browse and click on a product from the homepage

Test Case ID: TC-001

Feature: Product browsing

Preconditions: User is on the homepage

Steps:

1. Open the homepage
2. Click on a product

Expected Result: Product detail page is displayed with title, description, and price

Actual Result: As expected

Priority: High

Status: Pass

#### **Test Case 2:** Add a product to the cart

Test Case ID: TC-002

Feature: Cart Functionality

Preconditions: User is on a product detail page

Steps:

1. Click 'Add to cart'
2. Accept alert if shown

Expected Result: Product is added to the cart and alert is confirmed

Actual Result: As expected

Priority: High

Status: Pass

#### **Test Case 3:** View cart after adding a product

Test Case ID: TC-003

Feature: Cart Functionality

Preconditions: Product has been added to the cart

Steps:

1. Add a product to caet
2. Click 'Cart' from the top menu

Expected Result: Cart page shows the added product with name, price, and delete option

Actual Result: As expected

Priority: High

Status: Pass

#### **Test Case 4:** Add various products to the cart

Test Case ID: TC-004

Feature: Cart Functionality

Preconditions: On homepage

Steps:

1. Add two different products
2. Navigate to the cart

Expected Result: All products are shown in the cart with accurate names and prices

Actual Result: As expected

Priority: Medium

Status: Pass

#### **Test Case 5:** Submit order with invalid data

Test Case ID: TC-005

Feature: Checkout Validation

Preconditions: At least one item in cart

Steps:

1. Go to cart
2. Click 'Place Order'
3. Fill in all the fields with invalid values
4. Click 'Purchase'

Expected Result: Form should validate input and block invalid submissions

Actual Result: Order accepted (Bug: allows orders with invalid entries)

Priority: Medium

Status: Fail

Test Case Summary Chart

Test Case ID	Title	Status
TC-001	Browse and click on a product from the homepage	Pass
TC-002	Add a product to the cart	Pass

TC-003	View cart after adding a product	Pass
TC-004	Add multiple different products to the cart	Pass
TC-005	Submit order with invalid data	Fail

## Bug Report

Bug ID: BUG-001

Title: Accepts invalid customer information for placing an order

Severity: Medium

Priority: Medium

Status: Open

Reported By: Beyza Ozer

Date Reported: 05/29/2025

Environment: Windows 10, Chrome v136.0.7103.11, Desktop

### Description:

The application allows users to submit the 'Place Order' form with invalid or nonsensical data such as 'xxx' for name, 'yyy' for country, and '000' for the credit card number. There is no input validation on the frontend to prevent submission of incomplete or malformed data.

### Steps to Reproduce:

1. Go to <https://www.demoblaze.com>
2. Add any product to the cart
3. Click on 'Cart' and then 'Place Order'
4. In the form that appears, enter invalid values such as:
  - Name: xxx
  - Country: yyy
  - City: zzz
  - Credit Card: 000
  - Month: 00
  - Year: 00
5. Click the 'Purchase' button

Expected Result:

The form should perform validation to prevent submissions with invalid values.

Actual Result:

The form is submitted successfully and a confirmation message appears, despite all values being invalid.

#### **Impact:**

This issue compromises data quality and backend processing. It may lead to invalid or unusable orders being placed. It also negatively affects user trust.

#### **Suggested Fix:**

Implement frontend validation for all input fields in the 'Place Order' form. Ensure required formats for fields like credit card, month/year, and use input types and patterns.

#### **Bug Severity Justification:**

The purchase form allows customers to complete orders with invalid data. It does not block the core purchase flow. There is no crash or data loss, and users can still input valid data if desired. However, the lack of proper validation undermines data integrity and could lead to incorrect order records or poor UX. It can affect all the users and allow invalid submissions, but since the flow completes without error, it is classified as a 'Medium' severity issue.

### **Part 2: SQL Validation**

#### **Scenario:**

The product 'MacBook Pro' is shown on the front-end with a price of '\$1100' under the 'Laptops' category. Assume the backend stores product information in a MySQL table named 'products' with the following schema: product\_id (INT), product\_name (VARCHAR), category (VARCHAR), price (DECIMAL).

SQL Query to validate product info:

```
SELECT * FROM products WHERE product_name = 'MacBook Pro' AND price = 1100 AND category = 'Laptops';
```

### **Part 3: Automation (Cypress)**

Test script file is provided as email attachments and in the GitHub repository provided.

<https://github.com/bozer6125/candidly>

### **Part 4: API Testing (Postman) Live Session**