

## Test Case Report - E-commerce Website

---

**Project: E-commerce Website built with Sanity, Next.js, and TSX**

**Date: October 26, 2023**

### Summary

This report details the test cases performed on the e-commerce website, outlining the functionality tested, the results (success or failure), and any problems encountered during testing along with their resolutions.

---

### Test Cases

Test Case ID	Description	Status	Problems Faced	Resolution
1	Dynamic Product Display from Sanity after Vercel deployment.	Success	Products initially not displaying after deployment.	Issue identified and resolved. The incorrect API endpoint configuration in production was fixed by updating environment variables.
2	Search bar functionality - product search and display.	Success	Products were searchable, but the search button was not functioning.	Issue identified and resolved. The search button's event handler was missing. This was fixed by adding the necessary event listener.
3	"Add to Cart" button functionality on dynamic product pages.	Success	Amounts being added to the cart were resulting in "NaN" (Not a Number) values.	Issue identified and resolved. The quantity was being treated as a string. Fixed by converting it to a number before calculations.
4	Mobile Responsiveness - overall layout and functionality on mobile devices.	Failed	Empty spaces appearing due to excessive top bar length, causing main page unresponsiveness.	Partial progress made. Attempted to reduce top bar height using CSS media queries. Next steps include refactoring the top bar component for a more responsive layout.

---

# Detailed Test Case Descriptions and Results

## 1. Dynamic Product Display from Sanity

**Description:** Verify that products stored in the Sanity CMS are correctly fetched and displayed on the website after deployment to Vercel.

**Steps:**

1. Deploy the website to Vercel.
2. Navigate to the product display page.
3. Check if the products from Sanity are displayed correctly with their details (name, image, price, etc.).

**Result: Success.** Initially, products were not displayed after deployment due to an incorrect API endpoint configuration in the production environment. The issue was resolved by updating the environment variables with the correct API URL.

---

## 2. Search Bar Functionality

**Description:** Test the search bar functionality to ensure that users can search for products and relevant results are displayed.

**Steps:**

1. Enter a product name or keyword in the search bar.
2. Click the search button.
3. Verify that the relevant products are displayed in the search results.

**Result: Success.** Products were searchable and displayed correctly. Initially, the search button was not working because its event handler was missing. This was resolved by adding the necessary event listener.

---

## 3. "Add to Cart" Button Functionality

**Description:** Test the "Add to Cart" button on dynamic product pages to ensure products are correctly added to the shopping cart.

**Steps:**

1. Navigate to a dynamic product page.

2. Click the "Add to Cart" button.
3. Verify that the product is added to the cart and the cart count updates correctly.

**Result: Success.** Products are added to the cart correctly. Initially, the quantities were being treated as strings, causing "NaN" errors. The issue was resolved by converting the quantity to a number before performing calculations.

---

## 4. Mobile Responsiveness

**Description:** Test the website's responsiveness on different mobile devices and screen sizes. Ensure the layout and functionality remain user-friendly and consistent.

### Steps:

1. Access the website on various mobile devices (or use browser developer tools to simulate different screen sizes).
2. Navigate through different pages.
3. Check for layout issues, overlapping elements, and usability problems.

**Result: Failed.** Empty spaces were observed due to an excessive top bar length, causing significant unresponsiveness across the main pages. Partial progress was made in resolving the issue by reducing the top bar height using CSS media queries. Further improvements are required, including refactoring the top bar component to use a more responsive layout approach.