**Github Repo: https://github.com/BrianWCaton/OC\_PJCT\_5.git**

**Comprehensive E-Commerce Application Test Plan**

**1. Product Display and Navigation**

**Test Case 1: Product API Fetch**

**Steps:**

1. **Load the homepage**
2. **Open the browser console**
3. **Check if product data is logged correctly**

**Expected Result: Products should be displayed on the page without errors in the console.**

**Test Case 2: Product Rendering**

**Steps:**

1. **Verify that each product has an image, name, description, and link**
2. **Click on a product to ensure it navigates correctly**

**Expected Result: Clicking a product should take the user to product.html?id=productID.**

**2. Cart Functionality**

**Test Case 3: Adding Products to Cart**

**Steps:**

1. **Click on a product**
2. **Choose quantity and click "Add to Cart"**
3. **Open localStorage (F12 > Application > Local Storage)**
4. **Check if the product is stored in cart**

**Expected Result: The product should be saved in localStorage with the correct quantity.**

**Test Case 4: Displaying Cart Items**

**Steps:**

1. **Navigate to the cart page**
2. **Verify that added items appear with their details (image, name, color, price, and quantity)**

**Expected Result: The cart should display all products with the correct information.**

**Test Case 5: Updating Quantity**

**Steps:**

1. **Change the quantity of an item**
2. **Refresh the page**
3. **Check if the updated quantity is saved**

**Expected Result: The quantity should remain updated after refresh.**

**Test Case 6: Removing Items from Cart**

**Steps:**

1. **Click the "Delete" button for an item**
2. **Check if it is removed from localStorage**
3. **Refresh the page**

**Expected Result: The item should be removed from the cart.**

**Test Case 7: Empty Cart Handling**

**Steps:**

1. **Remove all items from the cart**
2. **Try proceeding to check out**

**Expected Result: The checkout should be blocked or show a message that the cart is empty.**

**3. Edge Cases and Quantity Limits**

**Test Case 8: Adding Maximum Quantity**

**Steps:**

1. **Select a product and set the quantity to the maximum allowed (e.g., 100)**
2. **Attempt to add it to the cart**
3. **Check if localStorage updates correctly**

**Expected Result: The cart should store and display the correct quantity without errors.**

**Test Case 9: Adding More Than Maximum Quantity**

**Steps:**

1. **Manually enter a quantity greater than the maximum allowed**
2. **Try adding the product to the cart**

**Expected Result: An error or validation message should prevent adding more than the max allowed.**

**4. Cart Persistence**

**Test Case 10: Cart Persistence After Page Reload**

**Steps:**

1. **Add items to the cart**
2. **Reload the page**
3. **Verify that the items remain in the cart**

**Expected Result: Items should persist after a refresh.**

**5. Order Form Validation**

**Test Case 11: Required Field Validation**

**Steps:**

1. **Try submitting the form with empty fields**

**Expected Result: Error messages should appear under each empty field.**

**Test Case 12: Invalid Email Format**

**Steps:**

1. **Enter an invalid email (e.g., user@)**
2. **Submit the form**

**Expected Result: The form should display an error for an invalid email format.**

**Test Case 13: Submitting Form with Spaces Only**

**Steps:**

1. **Enter spaces in required fields**
2. **Submit the form**

**Expected Result: The form should trigger validation errors.**

**6. Order Placement and Confirmation**

**Test Case 14: Successful Order Placement**

**Steps:**

1. **Fill in all fields correctly**
2. **Submit the form**
3. **Check if the API request is sent (Network tab in DevTools)**
4. **Verify redirection to confirmation.html?orderId=XYZ**

**Expected Result: The order is placed, stored in the database, and the user is redirected.**

**Test Case 15: Handling API Errors**

**Steps:**

1. **Temporarily disable the API (stop the backend server)**
2. **Try placing an order**

**Expected Result: An error message should be displayed.**

**Test Case 16: Order Confirmation Page Validation**

**Steps:**

1. **Successfully place an order**
2. **Verify if the confirmation page displays the correct order ID**

**Expected Result: Order ID should match the one returned from the API.**

**7. Performance Testing**

**Test Case 17: Large Cart Handling**

**Steps:**

1. **Add a large number of items to the cart**
2. **Test responsiveness and loading speed**

**Expected Result: The page should function without performance issues.**