Test Plan for Orinoco Handmade Teddy Bears

GitHub repo: https://github.com/2Coo85/TeddyBears.git

**Introduction**

This plan will outline the how to test the functionality of the site Orinoco Handmade Teddy Bears. The purpose of this project is to create an e-Commerce website that displays all products available, allows users to add selected items to cart, and fill in a form to submit their order.

This site has the following requirements

1. The home page will display a listing of all products available
2. The single-item page will populate the information from the selected item
3. The cart page will display the item(s) the user has selected with the current total and a form for them to fill in their information; form must be valid before it can be submitted
4. a confirmation page that will thank the user for their purchase, display an order number and final total of their purchase.

**Features to Test**

1. Adding items to cart
   1. Front-End
      1. User clicks on the add-to-cart button
      2. Cart is then populated with the item and quantity
      3. Cart quantity display is updated
   2. Back-End
      1. Calling the API endpoint \_id adds item into database that persists through the session
2. Updating quantity from cart page
   1. Front-End
      1. User clicks the up/down arrow to change the quantity of the item and clicks the update cart button; totals and quantity are updated in the cart
   2. Back-End
      1. Cart total is updated in the localStorage database
3. Deleting the whole cart
   1. Front-End
      1. User clicks on the delete cart button which clears all items from the cart
      2. Cart quantity is returned to zero
   2. Back-End
      1. Database is cleared of all information
4. Opening the home page
   1. Front-End
      1. All products are displayed with the correct price, image and name
   2. Back-End
      1. API request gets the correct data
5. Opening the single-item page
   1. Front-End
      1. Page displays the item the user has selected with the correct information
   2. Back-End
      1. API request returns the correct information for the id endpoint selected
6. Selecting color options
   1. Front-End
      1. Dropdown should display color options for the selected items
7. Validate form data
   1. Front-end
      1. A valid form should meet the following expectations
         1. Be at least 2 characters long
         2. Be alpha-numeric
         3. Be a valid email address
         4. Be a valid address combination
         5. No blank inputs
8. Order page is displayed
   1. Front-End
      1. Page should display a confirmation message with an order id and the total for the order
   2. Back-End
      1. API POST request should populate orderId
      2. Total price should be populated from the URL that holds the information

**Features to Not Test**

These features do not affect the effectiveness of the API therefore do not need to be tested.

1. Static elements and styling
   1. These elements do not affect the pages effectiveness to gather information from the API
2. Validity of data displayed
   1. Data displayed is built into the API information gathered from the JSON file therefore cannot be changed