**Shopping Website Challenge: Build Your Own Online Store!**

**Introduction:**

In this exciting challenge, you'll build a complete online shopping system using JavaScript and the provided API. You’re creating a website like furniture online store, but for your own unique brand!

**Important Information:**

How to install and run to the backend API:

<https://github.com/otemu/project-1-ecommerce-html-scss-javascript>

Figma design:

<https://www.figma.com/file/8XLY2o5YgepfaC2Rw93Iqc/eCommerce-Website-%7C-Web-Page-Design-%7C-UI-KIT-%7C-Interior-Landing-Page-(Community)?type=design&node-id=117-818&mode=design&t=s4vVLKQFosQAHwf0-0>

Where does my code go:

After you clone the project from link above, your see two folders, **backend** and **frontend**, all code should go in the frontend folder.

**The Challenge:**

Your mission is to construct a website with the following functionalities using HTML, SCSS and JavaScript:

**1. Home Page:**

* **Products:** Display a list of products retrieved from the backend API: /api/products/
* **Product Details:** Each product listing should include:
  + An image with a descriptive alt text.
  + The product's price.
  + A clear description of the product.
  + Available colour information.
  + A clickable link that takes you to the dedicated product page.

**2. Product Page:**

* **Focus on One Product:** This page showcases a single item using its unique ID from the API. (e.g., /api/products/33)
* **Details Display:** Show the product's image (with alt text), price, and description.
* **User Interaction:**
  + Allow users to choose the desired colour of the product.
  + Enable them to select a quantity (using a number input).
  + Provide a button for adding the chosen product to their shopping cart.

**3. Shopping Cart Page:**

* **Local Storage Integration:** This page utilizes local storage to keep track of items in the cart.
* **Cart Contents:**
  + Display all products currently added to the cart.
  + Enable users to adjust the quantity of each item. As the quantity changes, update the price accordingly.
  + Offer a button to remove items from the cart.
* **Order Summary:**
  + Calculate and display the total order value (sum of all product prices).
  + Show the total number of items in the cart.

**4. Checkout Page:**

* **Form Creation:** Build a form with the following fields:
  + Title (Mr./Ms./Mrs.)
  + First Name
  + Last Name
  + Email Address
  + Phone Number
* **Form Validation:** Implement validation to ensure users enter valid information before submitting the form.
* **Order Processing:** Upon successful form submission:
  + Post the ordered products and user details (name, email, phone) to a api endpoint API: /products-order
  + Redirect the user to a confirmation page.

**5. Confirmation Page:**

* **Thank You Message:** Display a simple message thanking the user for their order with confirmation order number.
* **Cart Clearing:** Clear the shopping cart after a successful order.

**Design Inspiration:**

For visual design, you can refer to this page [furniture shop design](https://www.figma.com/file/8XLY2o5YgepfaC2Rw93Iqc/eCommerce-Website-%7C-Web-Page-Design-%7C-UI-KIT-%7C-Interior-Landing-Page-(Community)?type=design&node-id=117-818&mode=design&t=s4vVLKQFosQAHwf0-0) . However, focus on building this site is the JavaScript, so don’t worry about getting a pixel perfect looking site.

**Helpful Tools and Techniques:**

* **API Calls:** Use the fetch API to retrieve data from the provided API endpoints.
* **DOM Manipulation:** Leverage methods like querySelector and innerHTML to dynamically modify the web page content.
* **Loops:** Utilize for...in, for...of, or map loops to iterate over product data.
* **Template Strings:** Use template literals (backticks) for cleaner string formatting.
* **Local Storage:** Store and retrieve cart data using the browser's local storage mechanism.

**API Reference: (if port set to 3000)**

* **Get Products:** <http://localhost:3000/api/products/>
* **Get Single Product:** [http://localhost:3000/api/products/[id]](http://localhost:3000/api/products/%5bid%5d) (Replace [id] with the specific product ID e.g <http://localhost:3000/api/products/33>)

**Tips:**

* Start by building the foundation:
  + Focus on fetching and displaying products on the home page.
  + Proceed step-by-step, tackling functionalities one at a time.
* Break down complex problems into smaller, manageable tasks.
* Don't hesitate to research and explore online resources for guidance.
* Test your code thoroughly as you develop each component.

**Ready to take on the challenge? Let's build your online store!**