React Coding Challenge (Shopping Cart)

Objective

The goal of this challenge is to correctly assess your knowledge of React and your approach to componentizing the proposed solution.
It is important that you apply your best practices, and if you make any decisions that deviate from standard practices, please explain why.
Additionally, we will positively value the creation of unit tests.

Instructions:

You are required to build a shopping store using React with Next.js as the framework.
Visit the Next.js website for more information: Next.js.

To display products, use the following API: Fake Store API

We also provide the design of the two of the three screens to be implemented. We expect you to faithfully reproduce the design without getting caught up in minor details.

Main Page:

On the home page, you will find:

- A header with a logo, a search bar, and a badge/icon showing the number of items added to the cart.

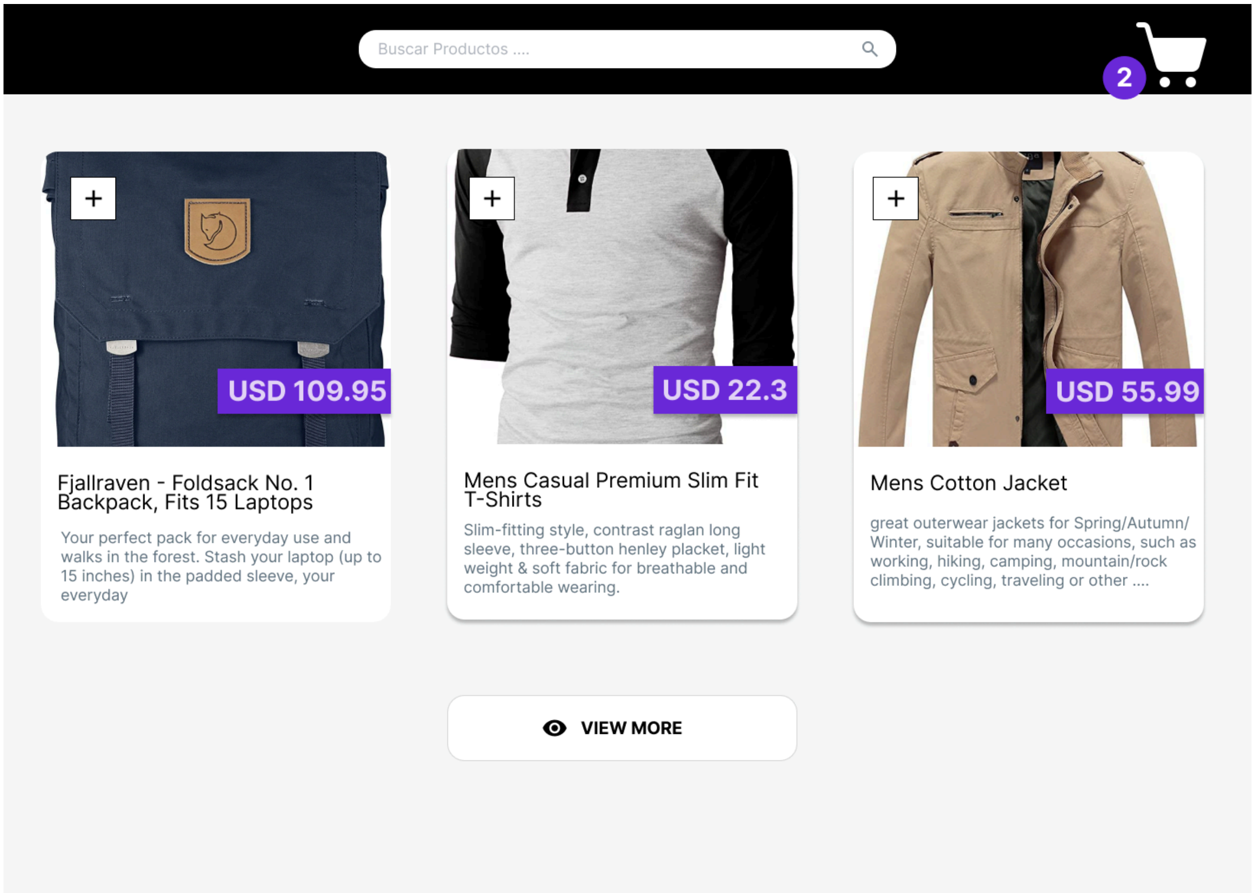
- Product cards displaying the title, description, price, and an action button to add the product to the cart.

Each time a product is added to the cart:

- Update the global cart quantity displayed in the header.

- Synchronize the cart content with the Shopping Cart view.

The home page includes a search bar that filters products that have already been loaded on the page.



Fjallraven - Foldsack No. 1 Backpack, Fits 15 Laptops

Your perfect pack for everyday use and walks in the forest. Stash your laptop (up to 15 inches) in the padded sleeve, your everyday

Mens Casual Premium Slim Fit T-Shirts

Slim-fitting style, contrast raglan long sleeve, three-button henley placket, light weight & soft fabric for breathable and comfortable wearing.

Mens Cotton Jacket

great outerwear jackets for Spring/Autumn/ Winter, suitable for many occasions, such as working, hiking, camping, mountain/rock climbing, cycling, traveling or other ....

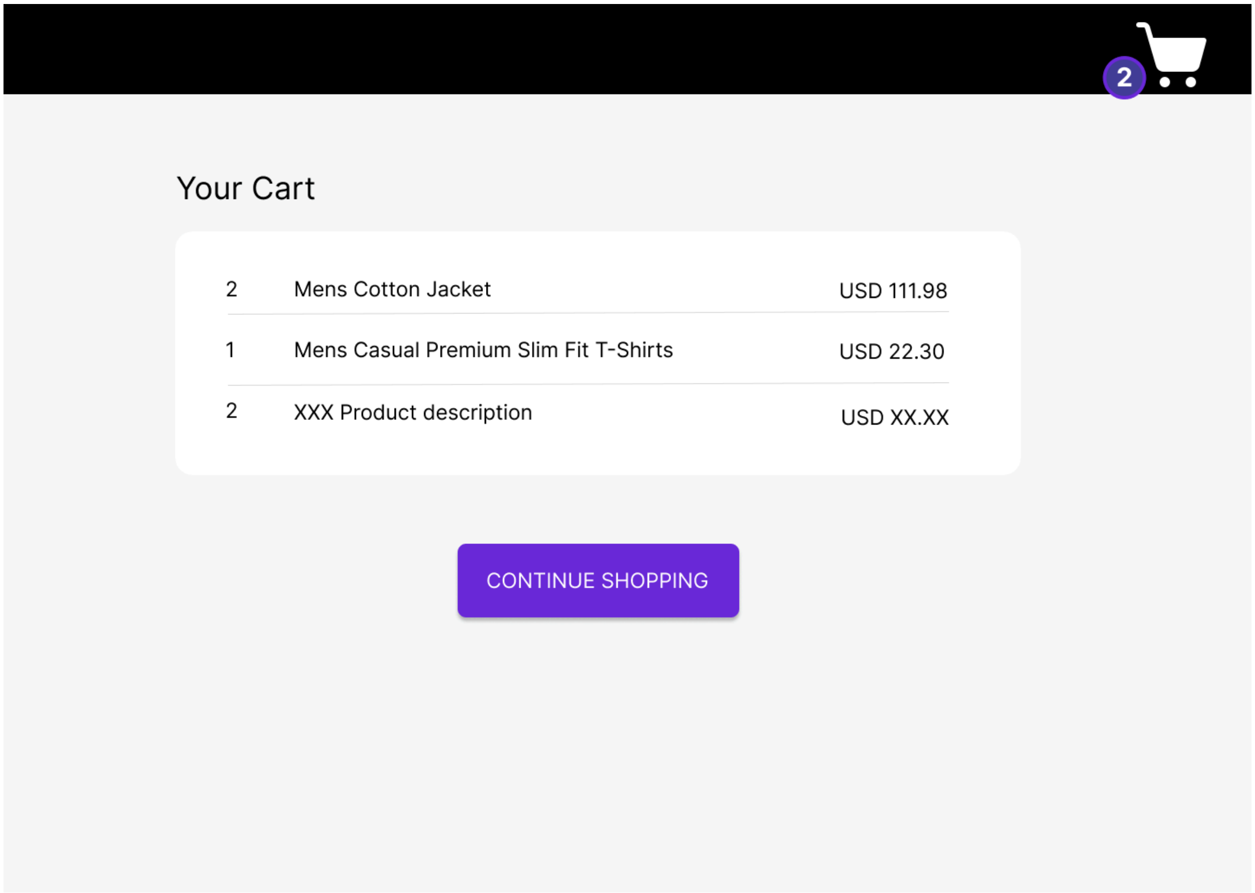
Additionally, a "See More" button should be implemented to request more products (implement local pagination).
By default, only 3 products (one row) should be displayed.

Shopping Cart Page:

The user will navigate to the shopping cart view by clicking on the cart icon. This page should display the details of the items in the cart, including:

- Quantity
- Title
- Total Amount

The "Continue Shopping" button should navigate back to the main page.



Your Cart

2	Mens Cotton Jacket	USD 111.98
1	Mens Casual Premium Slim Fit T-Shirts	USD 22.30
2	XXX Product description	USD XX.XX

Product Details Page:

Design and develop a user-friendly product details page that displays key product information, including an image, title, description, and price. Also this page should have the availability to add the product to the ca

The layout should be clean and intuitive, ensuring that users can quickly understand the product offering. The page must support responsive design, maintain consistency with the overall UI, and be optimized for readability and engagement.

Important Considerations:

While building the application, you will need to make the following technical decisions:

- Managing the application state (products, cart, added items, etc.).
- Choosing a styling framework (SASS, LESS, Styled Components, Tailwind, Pure CSS, etc.).
- Keeping the main page updated with the latest products.
- Handling errors and communicating them to the user.
- Implementing custom hooks if necessary.
- Implementing local pagination
- Implement unit testing in the most important components.
- You can use AI assisted engineering tools to complete the challenge.

Some UX aspects were intentionally left out of the proposed experience, but we would like you to identify and implement at least one improvement.

Submission information:

- **Duration**: We've estimated that this solution should take **around 2 hours to complete.** That said, feel free to take more time if you'd like to make it more complete or explore additional complexity: we welcome thoughtful approaches!
- **Deadline**: We ask that you submit your solution **within 72 hours**. If you **need more time, just let us know** — and **earlier submissions are appreciated**.
- Send an email to lzapata@goempirical.com with a copy to cflores@goempirical.com
  Email subject: "React Coding Challenge - [Your Name]"
- Include the public link to the GitHub repository containing your solution, or grant access to: GitHub - Leandro Zapata

  Note: You may include any details about your implementation in the email or the README file of the repository.