

Objective:

Create a basic e-commerce web application where users can browse products, view detailed product information, and add or remove items from the cart.

Requirements:

1. Home Page (Product Listing)

- Display a grid of products with essential details (e.g., name, price, thumbnail).
- Each product should link to its **Detail Page**.
- The user should be able to **filter and sort** the products based on the categories (multiple at once). The filter and sorting should be **agnostic to the page refresh** and back button navigation i.e. user should be able to share the link and filters should be applied.

2. Product Detail Page

- Use **dynamic routing** to display the product details based on the product **id** in the URL. e.g., `/product/:id`
- Fetch product data dynamically based on the **id**.
- Display product information, including a title, description, price, and an "Add to Cart" button.

3. Cart Functionality

- Allow users to **add** items to the cart from the Product Detail Page.
- Enable users to **remove** items from the cart.
- Display the total cart value and the number of items in the cart

4. Navigation

- Implement navigation to allow users to move between the **Home Page, Product Detail Page, and Cart**.
- Provide a way to navigate back to the Home Page from the Product Detail Page.

5. Technical Requirements

- **Typescript:** Use typescript to create the assignment
- **React JS:** Utilize React's component-based structure and state management.
- **Routing:** Use **React Router** for navigation between pages.
- **State Management:** Manage cart state using React's **Context** API
- **Data Fetching:** Use an API (<https://fakestoreapi.com/>) to fetch product details as well as category filters dynamically and once filters are applied, data should be refetched using apis for the selected filters. Don't filter locally, always call the APIs.
- **Responsiveness:** Ensure the application is mobile-responsive
- **E2E Testing:** Setup cypress (<https://www.cypress.io/>) or playwright (<https://playwright.dev/>) for basic page testing.

6. Bonus Points

- Implement a feature to persist the cart state using **localStorage**.
- Include animations for transitions between pages or adding/removing items from the cart.
- Ensure accessibility by using appropriate html elements for different blocks.

Submission Guidelines:

- Provide a GitHub repository link with clear instructions on how to set up and run the application in README. **Failure to do so will lead to auto-rejection.**
- Document any assumptions, limitations, or additional features implemented.
- You may use AI tools for syntax lookups or debugging. **Submissions will be automatically rejected if any AI tools are used without putting your own logical efforts at all.**