Frontend Engineer - Interview Case Study

Project: Interactive Data Dashboard

Objective

The goal of this case study is to assess your ability to build a clean, well-structured, and interactive frontend application using modern web technologies. We want to see how you approach a small project from start to finish, focusing on code quality, component architecture, and user experience.

The Challenge

You are tasked with building a single-page dashboard that displays a list of products fetched from a mock API. The user should be able to view, filter, and sort these products.

Provided Assets

You will be provided with a mock API endpoint: https://api.example.com/products.
A GET request to this endpoint will return a JSON array of product objects. Each object will have the following structure:

```
{
  "id": "prod_123",
  "name": "Wireless Noise-Cancelling Headphones",
  "category": "Electronics",
  "price": 249.99,
  "rating": 4.5,
  "inStock": true,
  "imageUrl":
"[https://placehold.co/300x300/E0E0E0/555?text=Product](https://placehold.co/300x300/E0E0E0/555?text=Product)"
}
```

Core Requirements

1. **Technology Stack:**
    - Use **React** for the user interface.
    - You may use any state management library you prefer (e.g., Context API, Zustand, Redux) or rely on React's built-in hooks.
    - For styling, please use Tailwind CSS or a CSS-in-JS solution like Styled Components.
2. **Functionality:**

  - **Display Products:** On page load, fetch the product data from the mock API and display the products in a grid or list format. Each product card should show at least its name, image, price, and rating.
  - **Loading & Error States:** Implement clear loading and error states for the API call. Show a loading indicator while data is being fetched and a user-friendly error message if the API call fails.
  - **Filtering:** Add a dropdown menu to filter products by category. The dropdown should be populated dynamically with the categories available from the fetched data.
  - **Sorting:** Add controls (e.g., buttons or a dropdown) to sort the displayed products by price (low to high, high to low) and rating (high to low).
  - **Responsiveness:** The application must be fully responsive and usable on mobile, tablet, and desktop screen sizes.

Expected Deliverables

1. A link to a public Git repository (e.g., GitHub, GitLab) containing the complete source code for your project.
2. A README.md file in the repository with clear instructions on how to install dependencies and run the application locally.

Evaluation Criteria

We will be looking at the following aspects of your submission:

- **Code Quality:** Is the code clean, well-organized, and easy to read?
- **Component Architecture:** Are the components well-structured, reusable, and logical?
- **State Management:** Is the application state handled efficiently and predictably?
- **User Experience:** Is the interface intuitive, responsive, and visually appealing?
- **Best Practices:** Have you followed modern frontend development best practices?

Good luck! We look forward to seeing what you build.