Hotcode Frontend Homework Assignment

Objective: Build a Dynamic Shopping List App

Your task is to create a small React application for managing a shopping list. The goal is to showcase your ability to structure a React app, make architectural decisions, and write clean, maintainable code.

Requirements

1. **Core Features**:
    - Users should be able to:
        - Add an item to the shopping list with a name, quantity, and category (e.g., "Fruits," "Dairy," "Vegetables").
        - Edit an item in the list.
        - Remove an item from the list.
        - Mark an item as "purchased."
2. **Dynamic Categorization**:
    - Display items grouped by their category.
    - Allow users to filter the list by category.
3. **State Management**:
    - Use a state management approach of your choice (e.g., Context API, Redux, or simple React state).
    - Explain in comments or a short README why you chose that particular state management strategy.
4. **Styling**:
    - Use a CSS-in-JS library or CSS Modules. You may also use a UI library like Material-UI, Tailwind, or Bootstrap, but you must customize styles slightly to avoid looking like a boilerplate.
5. **Error Handling**:
    - Handle basic user input errors (e.g., prevent adding items with empty names or negative quantities).
6. **Code Quality**:
    - Your code should be modular and well-organized.
    - Include comments where necessary to explain your thought process.
7. **Optional (Bonus Points)**:
    - Add a simple persistence layer (e.g., save the shopping list to localStorage or a mock API using a library like json-server).

Deliverables

1. A GitHub repository containing your code.
2. A short README file that includes:
    - Instructions on how to run the app.
    - An explanation of your chosen architectural and design decisions.
    - What you would improve if given more time.

Evaluation Criteria

1. **Code Structure**:
    - How well-organized is your code? Are components reusable and modular?
2. **Critical Thinking**:
    - How effectively did you choose and justify your state management solution?
    - Did you anticipate edge cases (e.g., invalid user input)?
3. **User Experience**:
    - Is the app intuitive and visually appealing?
4. **Completeness**:
    - Did you meet all the core requirements? Bonus points for implementing optional features.
5. **Efficiency**:
    - Does the app function smoothly without unnecessary re-renders or performance issues?