Fullstack/Frontend Engineer Intern Assignment

Objective

This assignment tests your skills in fetching APIs, handling state, rendering tables, and implementing dynamic filters.

Requirements

1. Build a React application to display and filter data from an API

2. **Get dummy data**
   (E.g. use `https://dummyjson.com/products`
   You'll receive an object with a `products` array containing product data.)

3. Store the dummy data locally, write mock API calls using Promise & setTimeout to get and update the data.

4. **Show data in a table**
   Display the following fields in a table:
   - `Title (Make this field editable, user should be able to edit the product name)`
   - `brand`
   - `category`
   - `price`
   - `rating`

5. **Make each row in the table deletable**

6. **Create filters for each column**
   - Each column should have a drop-down filter.
   - The filter values should be dynamically populated **based on currently available results** (i.e., with all other filters applied).

    - Example: If the user selects a brand, the category dropdown should only show categories available in the filtered result set.

Technical Requirements

- Use **React** (CRA or Vite).
- Use **functional components** and **React Hooks**.
- Styling is not the focus, but keep the UI readable.
- Don't use third-party table or filter libraries (like `Material Table`, `React Table`, etc.).

Bonus (Optional)

- Add a reset filter button.
- Add a loading state and error handling for the API call.
- Display a "No results found" message if filters return no data.

Deployment

- **Hosting:** Deploy the application on a free hosting platform (e.g., Heroku, Vercel, Netlify).
- **Accessibility:** Ensure the deployed application is fully functional, responsive, and accessible via a public URL.
- **Environment Variables:** Securely manage any API keys or environment variables needed for deployment.

VIRTUAL INTERNSHIPS

Deliverables

1. **GitHub Repository:** A public repository containing all source code, organized in a clean and logical structure.
2. **Deployed Application URL:** A live link to the deployed application for testing and review.
3. **README File:** Include setup instructions, the deployed URL, technologies used, and any necessary configurations.
4. **Documentation:** A brief document (approximately one page) explaining your development approach, challenges faced, and solutions implemented.

Submission Instructions

Please submit the following via the Google form link shared in the email:

- **GitHub Repository Link**
- **Deployed Application URL**
- **Documentation File**
- **Loom Video** walking us through your final submission:
    - Codebase Structure:
      Explain component hierarchy, state management.
    - Demo of Working Features :
      Show critical user flows.
    - Key Technical Decisions:
      Why did you choose specific libraries/patterns?

Note: AI generated/supported assignments will not be considered

We look forward to reviewing your application. If you have any questions or need clarifications, feel free to reach out. **Good Luck!**