Virtual Internships - Fullstack/Frontend Engineer Intern Assignment

Development Approach:

- The project was constructed with React and Vite to provide a quick and modular development environment.
- State management was managed through React hooks. The app fetches dummy data, keeps it locally in memory, and simulates asynchronous API calls through Promises with setTimeout.
- LocalStorage is employed to keep data and user preferences (such as dark mode) persistent.
- The UI remained simple and responsive through the use of custom CSS.

Key Technical Choices:

1. Utilized Vite for improved build performance and developer experience.
2. Added mock API layer (mockApi.js) to mock fetch/update/delete operations with delay.
3. Persist data and theme preference using localStorage.
4. Applied fully custom dropdown filters per column based on filtered data context.
5. Expanded editing from single-field to full-row editing for added functionality.
6. Made the table sortable by column headers and visual arrows.
7. Included a SummaryBar with dynamic filter chips and clear sort buttons.

Challenges Faced:

1. Managing dynamic filter values based on other active filters.
2. Handling editable rows without third-party libraries with preserved user experience.
3. Preserving UI state and data consistency between refreshes without a backend.

Solutions Implemented:

1. Filters recalculate options on each change of filters or products.
2. Editing employs controlled elements per row and is synchronized with mock API.
3. Employed localStorage wrapper to read/write product information and theme securely and effectively.

Features Delivered :

1. Dynamic filterable & editable table
2. Persistent user preferences and data
3. Dark mode with toggle and memory
4. Responsive, accessible layout

Deployed at: https://vi-react-assignment.vercel.app

GitHub: https://github.com/Akshaj-7264/VI-React-Assignment.git

Loom Walkthrough:
https://drive.google.com/file/d/1UR3JSgR93qrVD_IrfsDd_55fkCKlIgRY/view?usp=sharing