Objective:

Create a multi-page dashboard with sidebar navigation using **Vue.js or React**, Tailwind CSS, and Pinia Store (or Context API/Redux for React) for managing customer data. The dashboard will include GET, POST, and DELETE functionality using JSON Server as the backend.

Requirements:

1. **Dashboard Design:**
   - Implement a dashboard interface with two pages using Vue.js or React and Tailwind CSS.
   - Use Pinia Store for Vue or Context API/Redux for React to manage customer data and handle state management.
   - Implement sidebar navigation to switch between the two pages:
     - **Page 1: Customer Management**: Displays a list of customers, allows adding and deleting customers.
     - **Page 2: Customer Stats**: Displays additional information or statistics related to the customers.
2. **Pages:**
   - **Customer Management Page:**
     - Display customer data in a structured format.
     - Provide a form to add new customers.
     - Include a delete button for each customer to remove them from the list.
   - **Customer Stats Page:**
     - Display relevant customer-related statistics, such as total customers, recent additions, or other insights.
3. **Sidebar Navigation:**
   - Implement a collapsible sidebar for navigation between the two pages.
   - The sidebar should be responsive and styled using Tailwind CSS.
   - Ensure the sidebar links properly route between the pages using Vue Router (for Vue) or React Router (for React).
4. **Backend: JSON Server**
   - Set up JSON Server as the backend to handle customer data.
   - Use a db.json file to store the customer data.
   - The backend should support:
     - GET: Fetch the list of customers.
     - POST: Add a new customer.

        - DELETE: Remove a customer by ID.

5. **Pinia Store or Context API/Redux:**
    - Use Pinia Store for Vue or Context API/Redux for React to manage customer data on the frontend.
    - The store/context should handle retrieving customer data from JSON Server, adding new customers, and deleting customers.
6. **Frontend:**
    - Use Axios to communicate with JSON Server for data fetching and updates.
    - Implement form handling for adding customers and buttons for deleting them.
    - Use Vue Router for Vue or React Router for React for navigating between the two pages.
    - Style the dashboard using Tailwind CSS to ensure it is fully responsive.

Deliverables:

1. A fully functioning Vue.js or React project with Tailwind CSS, Pinia Store (or Context API/Redux), and JSON Server integration.
2. A multi-page dashboard with sidebar navigation between the Customer Management and Customer Stats pages.
3. Proper instructions to run both the frontend and backend locally.