Interview Test

Project Overview

Title: Next.js SQLite API Project with Zustand and useReducer

Objective: Candidates will design and build a small application using **Next.js**, **API routes**, and **SQLite** as the database. They must utilize **Zustand** for global state management and **useReducer** for component-level state. The project will be deployed to **Vercel**, include unit tests, and follow Git best practices.

Requirements

1. **Core Features:**
   - Use **Next.js** for the frontend and backend.
   - Implement at least **one dynamic API route** that interacts with a **SQLite database**:
     - Example: Fetch, add, update, or delete records in SQLite.
   - Demonstrate **CRUD functionality** via API calls and render the results on the UI.
2. **State Management:**
   - Use **Zustand** for global state (e.g., user session, app-wide settings, or shared data).
   - Use **useReducer** for localized state management within a specific component (e.g., form handling, pagination).
3. **Database Setup (SQLite):**
   - Use SQLite to store and query data (use any framework for SQLite).
   - Include a script to initialize and seed the database (e.g., `setupDatabase.js` or `seed.js`).

4. **Frontend Functionality:**
    - Build a user interface to:
        - Display data retrieved from the SQLite database.
        - Enable users to interact with the UI (e.g., add or update data).
    - Ensure responsiveness with **TailwindCSS**, **Bootstrap**, or another CSS framework.
5. **Unit Testing:**
    - Write unit tests for:
        - Components (with Jest or React Testing Library).
        - API routes.
        - Zustand store and reducers.
    - Achieve at least 80% test coverage.
6. **Git Workflow:**
    - Follow meaningful commit practices with a minimum of **5 commits**:
        - Example: Database setup, API development, UI integration, state management, and testing.
    - Encourage branching for feature-specific work.
7. **Deployment:**
    - Deploy the application to **Vercel**.
    - Ensure the SQLite database works seamlessly in production (use filesystem or cloud-based SQLite solutions if needed).
8. **Optional Features: (Bonus)**

    Encourage creativity:

    - Authentication for user access.
    - Enhanced UI/UX with animations or transitions (e.g., Framer Motion).

Submission Deliverables

1. **GitHub Repository:**
    - Public link to the repository.
    - Include a `README.md` with:
        - Project setup instructions.
        - Database setup instructions.
        - Features implemented.
        - Live URL (deployed on Vercel).
2. **Unit Testing:**
    - Include test reports and instructions for running tests.
3. **Video Walkthrough:**
    - A 5-8 minute video explaining:
        - The UI and app functionality.
        - Code structure (Zustand, reducers, SQLite integration).
        - Testing and deployment process.

Example Scenario for Candidates

"Build a web app to manage a simple inventory system. Use **SQLite** to store inventory data and create API routes for CRUD operations. Manage the global state (e.g., user session or selected item) using **Zustand** and local state (e.g., form state) using **useReducer**. Deploy the app to **Vercel**, write unit tests, and explain your work in a video."

Video Script Outline

1. **Introduction (1 min):**
    - Explain the idea and purpose of the project.
    - Highlight the use of **SQLite**, **Zustand**, and **useReducer**.
2. **UI Walkthrough (2-3 mins):**

- Demonstrate the app’s functionality:
  - CRUD operations.
  - State management features (global and local).

3. **Code Structure (3-4 mins):**
   - Walk through the folder structure:
     - **Frontend:** Pages, components, API calls.
     - **Backend:** API routes, SQLite integration.
     - **State Management:** Zustand store and reducers.
   - Highlight key files and logic.
4. **Testing and Deployment (1-2 mins):**
   - Show test coverage and key test cases.
   - Explain deployment to **Vercel**.

Would You Like?

1. **A Sample Project** with Zustand, useReducer, and SQLite for candidates to refer to.
2. **Video Walkthrough** showcasing an example project.
3. **Deployment Demo** live on Vercel.

Let me know how you'd like to proceed! 😊