Full-Stack Engineer (Dubai) - Home Assignment

Objective

Create a full-stack application using **React** for the frontend and **Node.js** for the backend. The application will fetch data from the **PokéAPI** and allow users to view and manage a list of Pokémon. Favorites will be managed and persisted through the Node.js backend.

Requirements

Main Features

1. **Frontend**:
    - Fetch and display the first 150 Pokémon in a scrollable list.
    - Clicking on a Pokémon should display its:
        - Abilities
        - Types
        - Evolution options (if available)
    - Add or remove Pokémon from the favorites list through a backend request.
    - Allow users to filter the list to show only their favorite Pokémon.
2. **Backend**:
    - Use Node.js to route requests from the frontend to the **PokéAPI**.
    - Implement a simple favorites management system:
        - **Add a favorite**: Save a Pokémon to the user's favorites list.
        - **Delete a favorite**: Remove a Pokémon from the user's favorites list.
        - **List favorites**: Return the current list of favorite Pokémon.
    - Persist the favorite Pokémon (can be stored in-memory or simple file-based storage)

UI/UX

- Build a clean and intuitive interface.
- Highlight favorite Pokémon in the list (e.g., with a badge or icon).

Data Handling

- Use the Node.js backend as a proxy to fetch data from the **PokéAPI**.
- Handle API loading states and errors gracefully on the frontend.

State Management

- Manage state on the frontend using React (or a library like Redux, if preferred).

Storage

- The backend will persist favorite Pokémon in memory or via a simple JSON file.

Technical Guidelines

1. **Tech Stack**:
    - Frontend: React
    - Backend: Node.js with Express or any other web framework.
2. **Frontend**:
    - Use React's state management for local state.
    - Implement filtering and interaction with the backend for favorites.
3. **Backend**:
    - Proxy requests from the frontend to the PokéAPI.
    - Expose the following REST API endpoints:
        - Fetch the first 150 Pokémon from the PokéAPI.
        - Add a Pokémon to the favorites list.
        - Remove a Pokémon from the favorites list.
        - Get the current list of favorite Pokémon.
4. **Styling**:
    - Use any preferred approach (CSS modules, styled-components, plain CSS, etc.).
5. **Code Quality**:
    - Write clean, modular, and well-documented code.

Bonus Points

1. **Frontend**:
    - Implement a search feature to quickly find a Pokémon by name.
    - Add animations or transitions for improved user experience.
    - Use lazy-loading or infinite scrolling for the Pokémon list.
2. **Backend**:
    - Store favorites in a database (e.g., SQLite, MongoDB) instead of in-memory storage.
3. **Deployment**:
    - Deploy the app (e.g., on Vercel, Netlify for the frontend, and Render for the backend).
    - Include the live link in your submission.

Submission

- Submit your project as a GitHub repository.
- Include a `README.md` with:
  - A brief overview of your approach.
  - Instructions on how to run both the frontend and backend locally.
  - Any additional features or assumptions made.