PhotoScout App Development Challenge

Project Overview:

You are tasked with developing the PhotoScout app for a social photography startup based in Sydney. The app aims to showcase default scenic photo locations and allows users to add their own locations with additional notes. Your primary focus is on creating a seamless user experience for both viewing and adding locations, along with persistence of user-entered data.

Assumptions:

1. **User Interaction:** Users are likely to interact with the app by exploring the map, discovering locations, and adding personal points of interest. Simplicity and intuitiveness are key.
2. **Location Management:** Users prefer a streamlined approach to manage locations, with the ability to add custom places and edit notes effortlessly.
3. **Persistence Priority:** Persisting user-added locations is crucial. Default locations can be fetched from the provided JSON file, while custom locations should be stored locally.
4. **Technology Stack:** React Native

Implementation:

1. **Map Screen:**
   - Utilize a user-friendly map SDK to display default scenic locations with pins.
   - Allow users to add custom locations directly from the map screen.
   - Provide a clear interface for users to name custom locations.
2. **List Screen:**
   - Present a sorted list of all locations based on distance.

- Distinguish default and custom locations for clarity.
- Enable users to select a location from the list to access details.

3. **Detail Screen:**
    - When a location is selected, display a detailed screen.
    - Include an option for users to input and edit notes about the location.
    - Ensure all information entered is persisted between app launches.
4. **Data Handling:**
    - Fetch default locations from the provided JSON file dynamically.
    - Implement internet connectivity checks to ensure smooth data retrieval.
    - Store user-added locations and notes locally for offline accessibility.
    - Sample JSON - https://pastebin.com/raw/fkAyNYGF
5. **Technical Considerations:**
    - Use third-party libraries judiciously, focusing on enhancing specific functionalities rather than solving the entire problem.
    - Include unit tests to validate critical app functions, ensuring robustness.

Code Submission:

You can share your completed code through a private git repository (e.g., GitHub) or send it directly, including all necessary git files. Keep the code private and refrain from making it publicly accessible.