React.js Intern Assignment

Objective:

Create a simple React.js application that fetches and displays data from a public API. The application should allow users to search and filter the displayed data.

Requirements:

1. **API Integration:**
   - Use the JSONPlaceholder API or any other public API (e.g OpenWeatherMap etc.).
   - Fetch a list of items (e.g., users, posts, weather info) and display them in a structured way (e.g., cards or a table) or create a location based weather website.

2. **Routing:**
   - Implement routing using `react-router-dom`.
   - Create at least two routes;
     a. A home page (`/`) to display the list of items.
     b. A details page (`/item/:id`) to show more details about a specific item when clicked.

3. **Search and Filter:**
   - Add a search bar to filter the list of items based on a specific field (e.g., name, title, or type).

4. **State Management:**
   - Use React's `useState` and `useEffect` hooks for managing state and side effects.
   - Optionally, use `Context API` or a state management library like Redux Toolkit (bonus points).

5. **Styling:**
   - Use a CSS framework like Tailwind CSS to style the application.
   - Ensure the UI is clean, modern, and responsive.

6. **Bonus Features (Optional):**
   - Add pagination or infinite scrolling for the list of items.
   - Use TypeScript for type safety (if familiar).

Instructions:

1. Set up a new React.js project using `create-react-app` or Vite.
2. Use functional components and React hooks (`useState`, `useEffect`, etc.).
3. Use `fetch` or `axios` to make API calls.
4. Implement routing using `react-router-dom`.
5. Host the project on GitHub and share the repository link.
6. Deploy the application using a free service like Vercel, Netlify, or GitHub Pages and share the live link.

Submission:

- Upload the completed project to a public GitHub repository.
- Ensure the repository is well-documented, with proper commit messages.
- Share the GitHub URL for evaluation.
- Include a `README.md` file with:
- Attach screenshots of the website

Evaluation Criteria:

- **Code Quality:** Clean, modular, and maintainable code, with proper use of components and services.
- **Responsiveness:** The app should be responsive and functional across different devices and screen sizes.
- **Interactive Features:** Smooth animations, transitions, and interactive elements that contribute to a professional user experience.
- **Documentation:** Clear documentation and screenshots included in the README file.