

Assignment: Create a Dynamic Pokémon Explorer App with React Router and API Integration

Objective

Build a Pokémon Explorer application where users can browse, search, and view information about different Pokémon dynamically. The app will use React Router for navigation and integrate with a public API to fetch Pokémon data.

Requirements

1. **Homepage:**
 - Display a list of Pokémon fetched from the API.
 - Each Pokémon should display its name, an image, and a "View Details" button.
 2. **Dynamic Pokémon Details Page:**
 - When the user clicks "View Details," navigate to a dynamic route (e.g., `/pokemon/:name`), where `:name` is the Pokémon's name.
 - On this page, display the full details of the Pokémon, including:
 - Name
 - Image
 - Abilities
 - Types
 - Base stats
 3. **Search Pokémon:**
 - Add a search bar to the homepage where users can type the name of a Pokémon to filter the list dynamically.
 4. **Favorites:**
 - Allow users to "favorite" Pokémon from the homepage or details page.
 - Include a separate "Favorites" page (`/favorites`) where users can view all their saved Pokémon.
 5. **Navigation:**
 - Include a navigation bar with links to:
 - Homepage
 - Favorites
 6. **Styling:**
 - Make the app visually appealing using CSS or a library like Tailwind CSS, Bootstrap, or Material-UI.
-

API Details

Use the **PokéAPI** for fetching Pokémon data:

- Base API URL:
<https://pokeapi.co/api/v2/>
 - Endpoints:
 - List Pokémon:
<https://pokeapi.co/api/v2/pokemon?limit=50>
 - Pokémon Details (dynamic):
<https://pokeapi.co/api/v2/pokemon/{name}>
-

Instructions

1. Use React Router for dynamic routing to Pokémon details and favorites pages.
 2. Use the `useParams` hook to access the Pokémon name from the URL.
 3. Use `useState` and `useEffect` hooks to fetch and display API data.
 4. Implement local storage or state management to store "favorited" Pokémon.
-

Bonus Tasks (Optional)

- Add sorting functionality (e.g., by name or base stats).
 - Display Pokémon by type (e.g., Fire, Water, Grass).
 - Add pagination to the homepage for browsing through all Pokémon.
-

Deliverables

- A fully functional React application meeting the requirements.
 - The project should be hosted (e.g., on GitHub, Netlify, or Vercel).
 - Submit the GitHub repository link or deployed application URL.
-

Dynamic Pokémon Explorer App

Grading Criteria

Criteria	Poins
Homepage with fetched Pokémon	20
Dynamic routing for details	25
Search functionality	20
Favorites feature	20
Navigation and Styling	15
Bonus Tasks	10