Pokemon

Instruction

Build a Pokemon catalogue web app using the Angular Framework.

Pokemon API

Use the Pokemon API to display Pokemon with their avatars. https://pokeapi.co/

Landing page

The user must be presented with a landing page where they can input their “Trainer name”. Trainers can collect Pokemon to train later. The Trainer name must be stored locally. When the user returns to the web app, they must not be asked for their trainer name again, but simply redirected to the Pokemon page.

Trainer page

The trainer page must display all the Pokemon the Trainer has collected. (*See the Pokemon detail section for more info on collecting Pokemon*).

It should display a list of Pokemon with their avatar. Each item in the list must be clickable and take the user to the Pokemon Detail page for the specific Pokemon that was clicked.

Pokemon Catalogue

Users may *not* have access to this page unless they have entered a trainer name.

The Pokemon catalogue must be a list of “*card style*” Pokemon presented to the user. The Pokemon should have an image (sprite) and name displayed.

Each Pokemon card must be clickable and take the user to the Pokemon detail page. See “Pokemon detail” section for more detail.

Side notes: Pagination is optional

Pokemon detail

A Pokemon detail page should also be built. This should display an image of the Pokemon along with its abilities, stats, height, weight and types.

There must also be a button to “Collect” this Pokemon for the logged in Trainer. This should then be stored locally and used to display the collected Pokemon in the Trainer page.

The detail profile should be divided into sections:

Base Stats

- Image
- Types
- Base Stats
- Name

Profile

- Height
- Weight
- Abilities
- Base Experience

Moves

- A list of moves

Minimum Requirements:

- Use the latest Angular with the Angular CLI
- Use Components to
    - Create “Root” or “Parent” components for pages
    - Create reusable pieces of UI
- Use the Angular Router to
    - Selectively display “parent” components based on the URL’s path
- Use the Angular Guard pattern to
    - Restrict access to pages
- Use Services to:
    - Share data between components
    - Make HTTP Requests using the HttpClient