Introduction

Welcome to the Pokémon Coding Challenge! In this challenge, you will be working with a dataset of Pokémons, each represented as a JSON object with various properties.

The expected time frame for this home assignment is considered to be around 5 hours, please document in the `README` if there were time constraints limiting your work and what you would've done if you'd had more time. Feel free to spend more time on the assignment if you like to.

Dataset

The dataset can be downloaded from here..

Tasks

- Load the JSON dataset into a database of your choice.
  - The database should be used by all endpoints.
- Create an API endpoint that gets a pokémon by its ID.
  - It should return next and previous evolutions as well.
  - If its next or previous evolutions have further evolutions, those should be included as well.
- Create an API endpoint that filters Pokémon by type.
  - The endpoint should accept a type as a parameter and return all Pokémon of that type.
  - It should accept sorting as a parameter. Should be able to sort on most properties (for example weight).
- Create an API endpoint that searches for a pokémon by name.
  - The endpoint should accept a string parameter with a minimum length of three.
  - The parameter should expect the name of a pokémon.
  - Matching the name should be fuzzy.
- Create an API endpoint that returns a suggested pokémon.
  - It should accept a pokémon as a parameter.
  - It should return a pokémon that has a type that the provided pokémon is weak against. The returned pokémon should not be weak vs. the provided pokémon.
- Create an API endpoint to add a pokémon to the database.
  - It should accept a pokémon as a parameter.
  - It should be possible to add a pokémon as the next or previous evolution of a already existing pokémon in the database.

Requirements

- Your application should use Node.js.
- Should be written in Typescript.
- Your code should be organized and easy to read.
- You should handle any potential errors that could occur.

Nice to have

- Validate parameters to API endpoints
- Unit tests
- Dockerize application
- Client to consume the API (React, swagger, etc)

Submission

Please submit your completed application by providing a link to a Git repository containing your code. Include any necessary instructions for running your application in a `README` file.

Good luck!