IMY 220
Assignment 6
TypeScript

Due: Wednesday 29 October at 23:59

The submission instructions are available on ClickUP. Any deviation from these instructions will result in a 10% deduction from your mark.

Instructions

- This is an individual, take-home assignment.
- This assignment focuses on using TypeScript for static type checking in React components.
- The objective is to create a React web app that lets users search for a Pokedex by name and displays information retrieved from the Pokemon API.

Project setup

- Create a React app using Webpack, Babel and ExpressJS as taught in class.
- Add TypeScript to your React app to add static type checking throughout your app.
- You can refer to this link for more information: https://github.com/Microsoft/TypeScript-React-Conversion-Guide#typescript-react-conversion-guide

Hint: be sure to use the .tsx extension for all React component files (including index.tsx) so TypeScript properly handles JSX and avoids compiler errors.

Pokedex search app

In a file called api.ts:

- Export an asynchronous function that queries the Pokemon API: https://pokeapi.co/
- The function should take in a string, and query the PokeAPI for a **pokedex** by **name**.
- The API request should be made by using one of the methods taught in the lecture videos and slides which returns a Promise.
- Specify the **parameter** and **return type** of the function (:any is not allowed).
- You should do adequate error handling (for example, if the API request fails or if the response returned is empty) and you should return and display appropriate and informative error messages to the user (not just in the console).

In a file called Pokemon.tsx:

- Create a component that displays the results of the API request that is passed to the component. It should display the following regarding the Pokedex in question:

  - The id
  - The name
  - The english description
  - A list of the pokemon entries (name and url for each)
- Create an interface for the component’s props with the above data. Select the correct TypeScript types for each field (**:any is not allowed**).

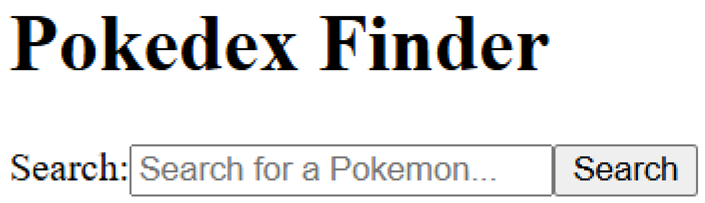
In a file called Search.tsx:

- Create a component that handles the user input for a search.
- Users should be able to search for a Pokedex.
- The search should not execute if the input is **empty**, or if the input contains any **numbers**.
- This component should not make API calls, but should just handle user input of a search value that is then passed to the parent component.
- Specify the **types** of the **props** passed in to your Search component (:any is not allowed)

In PokemonApp.tsx:

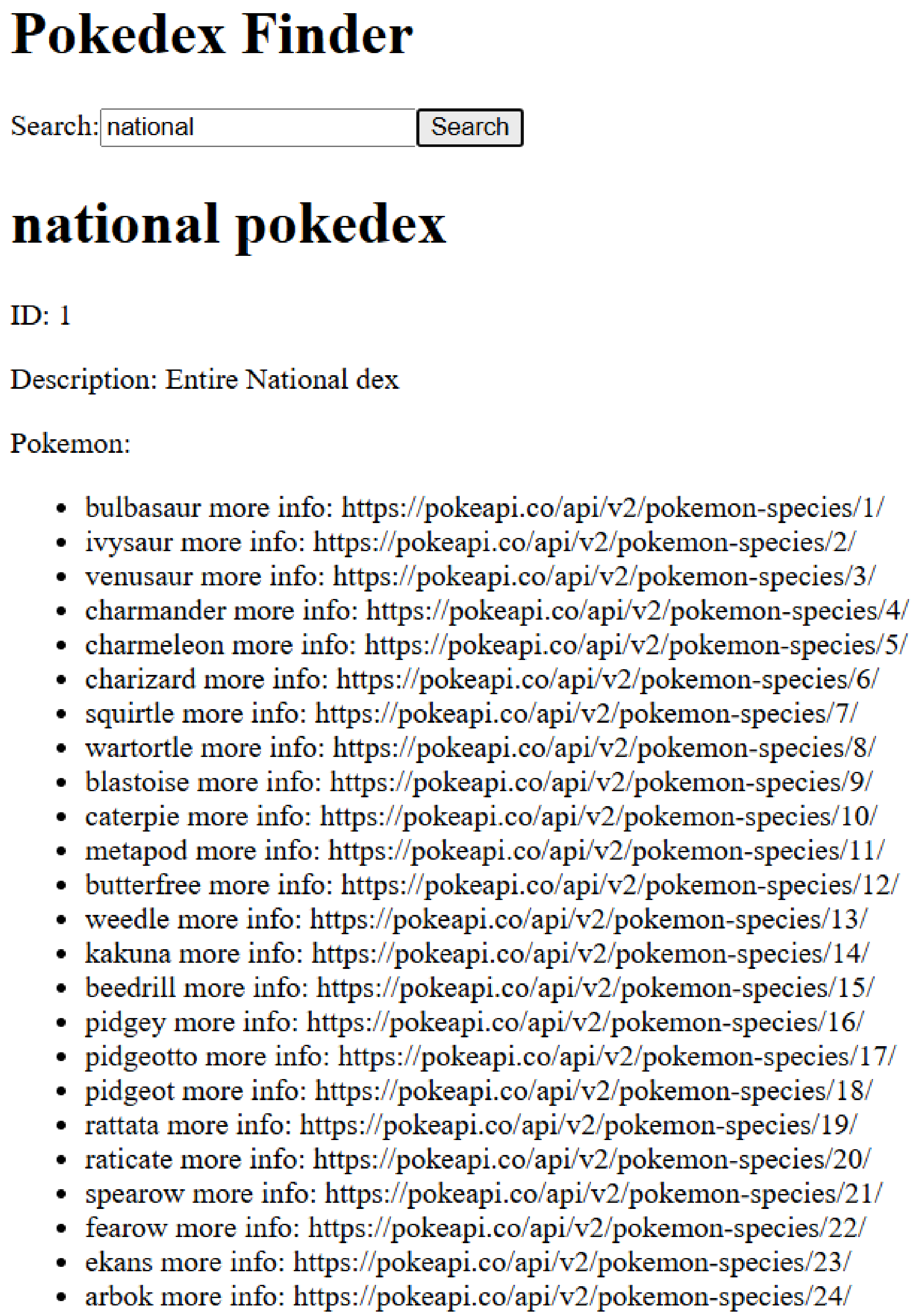
- Create a component to act as the parent component for the Pokemon and Search components, and should handle making requests to the PokeAPI using the function you defined in api.ts.
- When a user searches for a specific Pokemon, a search query to the PokeAPI should be made and the Pokemon component updated with the result of the search.
- Render this component in the “#root” div in the browser. This should be the root of your application.

The final result should look like this:
When the page first loads:



Pokedex Finder

If the user searches for “national”:



Pokedex Finder

Search: national

national pokedex

ID: 1

Description: Entire National dex

Pokemon:

- bulbasaur more info: https://pokeapi.co/api/v2/pokemon-species/1/
- ivysaur more info: https://pokeapi.co/api/v2/pokemon-species/2/
- venusaur more info: https://pokeapi.co/api/v2/pokemon-species/3/
- charmander more info: https://pokeapi.co/api/v2/pokemon-species/4/
- charmeleon more info: https://pokeapi.co/api/v2/pokemon-species/5/
- charizard more info: https://pokeapi.co/api/v2/pokemon-species/6/
- squirtle more info: https://pokeapi.co/api/v2/pokemon-species/7/
- wartortle more info: https://pokeapi.co/api/v2/pokemon-species/8/
- blastoise more info: https://pokeapi.co/api/v2/pokemon-species/9/
- caterpie more info: https://pokeapi.co/api/v2/pokemon-species/10/
- metapod more info: https://pokeapi.co/api/v2/pokemon-species/11/
- butterfree more info: https://pokeapi.co/api/v2/pokemon-species/12/
- weedle more info: https://pokeapi.co/api/v2/pokemon-species/13/
- kakuna more info: https://pokeapi.co/api/v2/pokemon-species/14/
- beedrill more info: https://pokeapi.co/api/v2/pokemon-species/15/
- pidgey more info: https://pokeapi.co/api/v2/pokemon-species/16/
- pidgeotto more info: https://pokeapi.co/api/v2/pokemon-species/17/
- pidgeot more info: https://pokeapi.co/api/v2/pokemon-species/18/
- rattata more info: https://pokeapi.co/api/v2/pokemon-species/19/
- raticate more info: https://pokeapi.co/api/v2/pokemon-species/20/
- spearow more info: https://pokeapi.co/api/v2/pokemon-species/21/
- fearow more info: https://pokeapi.co/api/v2/pokemon-species/22/
- ekans more info: https://pokeapi.co/api/v2/pokemon-species/23/
- arbok more info: https://pokeapi.co/api/v2/pokemon-species/24/

Submission Files

Submit only the following file(s) according to the submission instructions.

- All of your project files besides the node_modules folder

Submission Instructions

- ZIP all submission files (using a tool like 7Zip or WinRar) into a ZIP file called {position-a1}.zip, where position is your position in the grade center.
- Submit your ZIP file before the deadline.