Pokemon Dex and Game

The App is divided in 2 parts, a pokedex explorer and a Pokemon battle game.

Read carefully the instructions in each task and also please, before starting, check also the "Notes" list at the end of the tasks listing for more instructions.
By implementing the **<u>BONUS POINT/TASK</u>** you will stand out with your delivery.

Delivery limit is 5 days

1. Create a development environment using a bundling tool(webpack, rollup etc..), and integrate any unit tests framework (jest, mocha, cypress etc..).
   BONUS POINT: use storybook to preview and develop reusable components;

2. Setup apollo client using https://graphql-pokemon2.vercel.app/ for the graphql interaction;

   You will need to use this endpoint in the next step to fetch the pokemons from the graphql api.
   NOTE: you can play around using graphQL explorer at the same url in your browser.

3. Create the app UI, the app consists of 2 screens (see mockup below)

   - when the user opens the app it should display a grid view of all Pokemon's;
   - we should be able to do a fuzzy search of pokemons using a search element;
   - when selecting a Pokemon it should open a sidebar drawer with a inspect view of;
   Pokemon, including bigger image, characteristics and evolutions - should be clickable and update the current drawer to the new Pokemon selected;
   BONUS POINT: use redux state; use postcss;

4. After doing the battle screen lets create the battle game logic.

In this task(4th) you have the creative freedom to take this game to the next level and define your own game mechanics/logic, by looking at Pokemon types in graphql we can see that we have some data available which we can use to create a simple attack style game using the attacks data, like damage, also the Pokemon being attacked could be resistant or weak to those types of attacks.

Requirements:
The battle needs to be User vs Computer.
Each of the Pokemons that are available in the battle should have a maximum of 4 attacks available or less, being at least 1 fast and 1 special.
The number of pokemons on each side at the beginning of the battle should be between 3 and 5.
After the battle is over we should have a dialog/popup/screen with the result of the battle and couple buttons to rematch or exit to pokedex.
Don't repeat pokemons on each team.

Ideas:
- Allow to make a team by hand picking the pokemons for each battle. Give a score to the team selected and generate a opponent(Computer) team of similar level to be even. (Maybe add a difficulty level selector?)

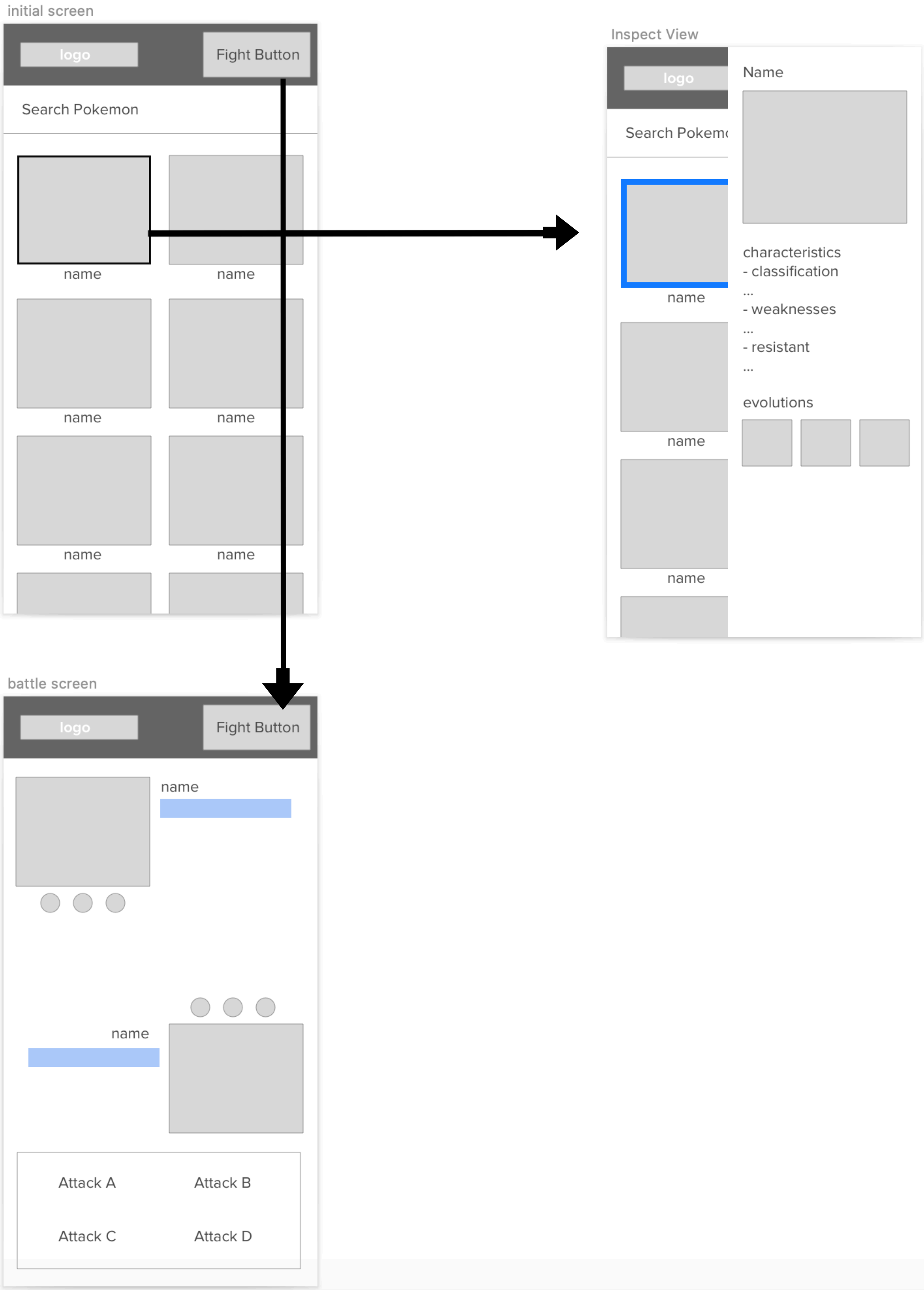
- Allow switching pokemons during battle.
- Use data from height, weight and CP power to add some chances of deflecting the attacks.
- keyboard shortcuts for triggering attacks
- animation effects
- sounds
- extra items eg: potions etc…
- Make it based on turns, player - computer. Or spammable attacks with attack coldowns.

BONUS POINT: make unit tests for the game mechanics/logic.

5. **BONUS TASK:** go a step further and make this app isomorphic by doing a express server to send the initial html response;

NOTES:

- use git and make your code available via github/bitbucket/gitlab;
- the web design should be responsive;
- you can start the project by using any boilerplate tool eg: create react app, next, razzle etc..
- use appropriate tools, libraries and frameworks to make your life easy;


initial screen
logo
Fight Button
Search Pokemon
name
name
name
name
name
name
Inspect View
logo
Search Pokem
name
name
name
Name
characteristics
- classification
...
- weaknesses
...
- resistant
...
evolutions
battle screen
logo
Fight Button
name
name
Attack A
Attack B
Attack C
Attack D