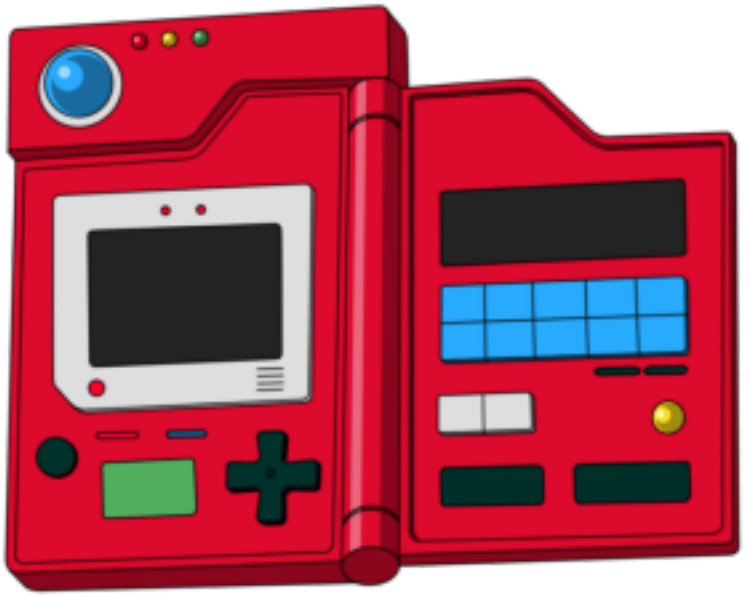
Myriad mobile

Front-End Web Pokédex Project



Updated: Sept 9, 2018

Challenge Overview

Objective

Since we were but wee children, we have all dreamt of being the best Pokémon trainer to walk the earth. A critical tool to that end is the Pokédex -- an interactive catalog of all known Pokémon, containing facts and stats about each creature. Unfortunately, Professor Oak can't afford to hand out free Pokédexes to every wannabe trainer to waltz into his lab. That's where you come in.

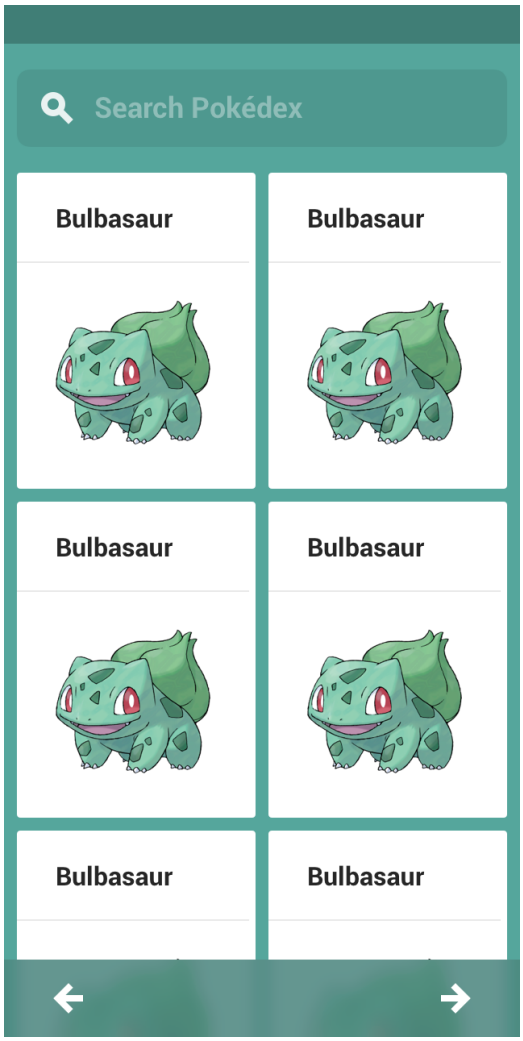
You are tasked with building a Pokédex web app for Professor Oak. He has provided a JSON API for you to use -- you are charged with building a SPA (single-page app) which uses that API.

Outline

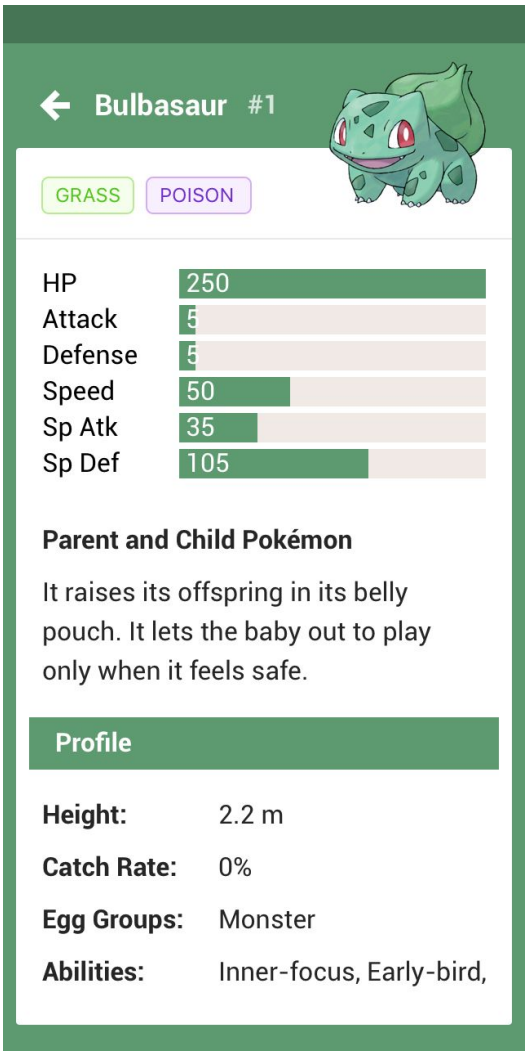
This task outlines specific requirements which must be fulfilled to continue. However, you are not limited to these requirements. You are encouraged to improve your solution in any way you deem fit.

You will need to create a hosted Git repository where you will be expected to push your code so we can track your progress. At minimum, you must commit and push your code upon completion so that it may be evaluated. A few things to note:

- Any work above-and-beyond the given task will be evaluated. Feel free to add any additional features that you would like, as long as all of the required objectives are completed.
- You are not necessarily expected to finish the entire project. Time considerations such as work, class, or other life concerns will be taken into account when evaluating how far an applicant has made it into the project.
- You are not necessarily expected to have the knowledge to complete this project—this is simply a test of your ability to problem solve and find the necessary answers.



- Get paginated list of Pokémon from the API (Documentation below in the Links section)
- Clicking a Pokémon card should load the detail view for that Pokémon
- Number of Pokémon per row should change dynamically with the width of the screen
- Next/Previous buttons should allow the user to progress through the pages
    - A query parameter should be used to store the current page number. When reloading the webpage, the correct page should be loaded and displayed
- Search
    - The new results should be fetched when the user presses enter
    - The search string should be stored in a query parameter



- Each Pokémon detail page should have a unique URL
- Back arrow should return the user to the list view
- Standard web browser navigation controls (i.e. back/forward) should be functional
- For bonus points, use color thief to dynamically set the primary color used on the page depending on the Pokémon's picture

Resources

API Documentation	https://intern-pokedex.myriadapps.com/docs/
UI Specification	https://app.sympli.io/p/ee64004e3d338b3f1d0d8421d860328025be569550
UI Desktop/Mobile Design	https://myriadmobile.invisionapp.com/share/N2Q2EHOBHA7
HTML	https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML
CSS	https://developer.mozilla.org/en-US/docs/Learn/CSS
Git	https://try.github.io/levels/1/challenges/1 https://medium.freecodecamp.org/git-cheat-sheet-and-best-practices-c6ce5321f52
React.js	https://reactjs.org/
Create React App	https://github.com/facebook/create-react-app
React Router	https://reacttraining.com/react-router/
Restful Web Services	http://www.javacodegeeks.com/2014/11/an-introduction-to-rest.html
Axios	https://github.com/axios/axios

Postman
https://www.getpostman.com/docs/introduction

Postman is a Chrome extension that allows you to easily hit API endpoints to test data. This tool is very helpful for you to view each of the API calls and see their responses.