A Web-Based Pokédex

Using React and NodeJS

Dr. Chen

COSC 450-001

12/2/202

Authors:

Andrew Moffett: atmoffett0@frostburg.edu

Ian Turner: ilturner0@frostburg.edu

Table of Contents

[Objective 3](#_Toc1872308982)

[Background 3](#_Toc264169462)

[Team Member Introduction 3](#_Toc1723546891)

[Testbed Setup 3](#_Toc831392083)

[Technologies Used 4](#_Toc1477065670)

[NodeJS 4](#_Toc975729390)

[React 4](#_Toc1157468648)

[HTML and CSS 5](#_Toc349370221)

[Bash Scripting 5](#_Toc2019299093)

[Github 5](#_Toc1579411161)

[System Design 6](#_Toc400309484)

[System Flow 7](#_Toc2082430228)

[Future Work 8](#_Toc1459852327)

[Conclusion 8](#_Toc1045690683)

[Github Project Repository: 8](#_Toc1010309081)

[REFERENCES 9](#_Toc772624825)

# Objective

The objective of this project is to create an interactive web application that serves as a Pokédex where the user can search different types of Pokémon and receive a result displaying the Pokémon's name, type(s), Pokédex number, height, weight, abilities, and base stat values. This app can be used while playing the game(s) to learn more about Pokémon.

# Background

The intent of this project is to gain experience and familiarity with popular front end web development tools such as NodeJS and ReactJS. In addition to learning these tools, we also wanted to gain more general experience with using JavaScript and Application Programming Interfaces in a web application.

# Team Member Introduction

Our team consists of two members, Andrew Moffett and Ian Turner. Andrew handled initial research of the project and the technologies we will be using. Both Andrew and Ian worked together on the coding, with the initial react app creation being handled by Ian. Andrew helped catch bugs, test cases, and UI/UX design choices. Ian created our presentation for the project and set up our live demo. Finally, this document is a group effort from both Ian and Andrew.

# Testbed Setup

For testing purposes, we both used our own personal computers while using Github to share our code. Both machines were similar in that they were both using the Windows 10 operating system, NodeJS version 16.18.0, and ReactJS version 18.2.0. The key relevant difference between the machines is that they used separate browsers, with one using Chrome and the other using Firefox. This could possibly affect the result as different browsers sometimes use different versions of JavaScript. Also, the support for JavaScript APIs differs between browsers as well. In our case, we encountered no such issues and the app worked as intended on both browsers.

# Technologies Used

Our project makes heavy use of JavaScript (JS) as our primary language. JS is used for NodeJS and React primarily. Our app also includes a JSON scripting file to run Bash commands for the app. Finally, we use JS to render and deliver HTML with our given Pokémon data, which uses CSS as styling.

## NodeJS

NodeJS is an asynchronous, event-driven JS runtime. It is designed to build scalable network applications using JS. Unlike more common concurrency models, nodeJS does not use threading, and avoids dead-locking processes by not using locks. It presents an event loop as a runtime construct instead of as a library. Typically, behavior is defined through callbacks at the beginning of a script, and the event loop is entered after executing the input script [1].

## React

React is a component-based library that allows the user to create encapsulated components separately before bringing them together to create a more complex UI. Since the component logic is written using JS, the user can pass rich data through the app and keep the state out of the DOM. React components implement a render() method to take in data and return display information, in our case through the form of HTML and CSS. Each component can maintain internal state data, and when this data changes, the rendered markup will be updated by re-invoking render() [2].

## HTML and CSS

HTML describes the structure of a webpage and is the standard markup language for creating Web pages. HTML elements tell the browser how to display content by labeling pieces of content uniquely. Labeling such as “this is a heading”, “this is a paragraph”, etc. CSS is a method of styling HTML. HTML tags are given names, in JS these are written className =”className” inside of the tag. CSS can then define styling patterns and requirements by referring to the className.

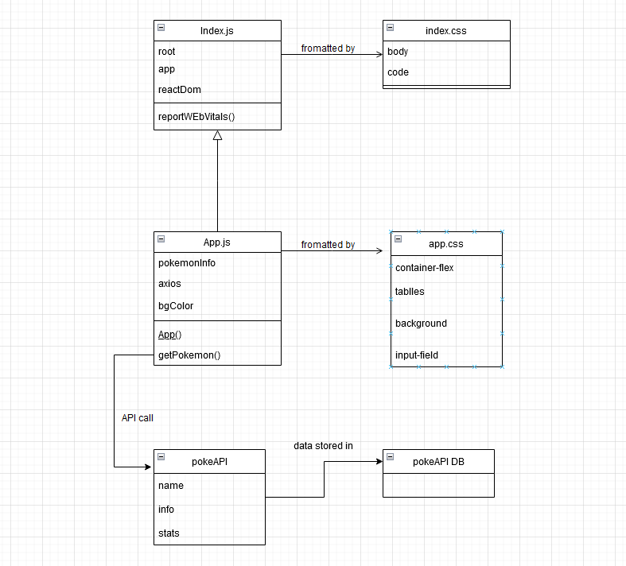
## Bash Scripting

NodeJS implements a package.JSON file that declares dependencies for the project along with information regarding the version data. Inside of this JSON we include script information, such as “npm run server” to run a server that hosts an API to contact a database. Our primary script, “npm start” starts our application. Instead of start we can use “npm build” to build the application. With the server or app running, we can also define scripts to view runtime information, restart the system, or test the application.

## Github

Github is a collaborative tool that allows users to upload code directly to repositories as well as download that code directly into an IDE such as Visual Studio Code. Github also has powerful built-in tools for version management, branching for code comparisons, and allows for strict control over updates. Further, more complex functions such as docker, amazon web services, and more professional applications are available directly through Github. Github also allows for the sharing of open-source code and code spaces that allow users to build their portfolio and skills in a safe and professional environment.

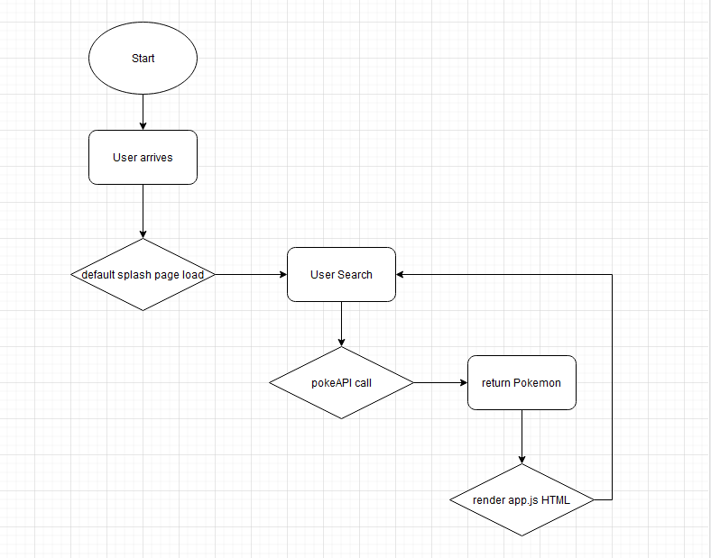
# System Design



Our system only uses a few pieces to create the overall application. Index.js is the nodeJS server information and structure. It makes use of bash commands to allow the application to run, and invokes the application upon startup. Index.css applies some simple styling information to the index.js page.

App.js is the primary application built using React. It contains the primary render() function to display Pokémon information and renders our HTML. App.css provides styling information for the app.js rendered HTML. PokéAPI and its database are third-party systems that we are accessing for the data. Rather than collect the 10s of thousands of pieces of information ourselves, we make use of this open source system designed for training and learning. We use our getPokemon() function from our App.js to make a call to the API. The API then creates a SQL request to the DB and returns a JSON to our App.js.

# System Flow



Our system flow operates a simple loop in which the user can search for and display different Pokémon information. The user starts by accessing the application, currently our app is only found on a local host, but ideally this would be accessible via the internet. Once the user is on the app, they are delivered a static image splash page with a single text entry field to search in. The search will only complete if the user enters an accurately named Pokémon. Once the Pokémon name has been entered, the app will make a call to the PokéAPI for the data and render an HTML page displaying the information. The user can then either end their session or search again. This loop repeats until the user is satisfied and ends their session.

# Future Work

If granted more time, we would like to implement a way for the background color gradient to change dynamically for dual type Pokémon. For example, if a Pokémon is both water and flying type, the background color would display a mixture of the designated colors for those types. We would also like to implement more tables and display more data such as each Pokémon's move set and base experience values. Aside from this, we would take time to improve upon the styling of our page to create as smooth an interface as possible.

# Conclusion

In conclusion, our project met our initial goals of creating a web based Pokédex while providing us with valuable experience working with JS, nodeJS, and React.

# Github Project Repository:

<https://github.com/ilcturner0/450-Project>

# REFERENCES

[1] Node.js. “About.” *Node.js*, <https://nodejs.org/en/about/>.

[2] “React – a JavaScript Library for Building User Interfaces.” *– A JavaScript Library for Building User Interfaces*, <https://reactjs.org/>.