**How to install and use Pokemon-path**

For capstone project reviewers

The easiest way to access my project is to simply visit it using the internet. There is absolutely no need for you to go through the hassle of trying to get node.js to work on your computer if you aren’t already familiar with it. The entire project is currently live on Heroku right now. Simply click the link below to visit it.

<https://pokemon-path.herokuapp.com/>

Note: My application goes into sleep mode when no one has accessed it within 30 minutes. It may take up to *30 seconds* to wake up. After that it should be fine.

All of the code used for the project can also be accessed on my GitHub:

<https://github.com/jordanmkimball/pokemon-path>

Note: I started laying the foundations for my capstone project back in the Building Functional Prototypes with Node.js Microsoft Professional Program Class. This is why my repository is older than the start of this class. Its always good to start early.

**How to run Pokemon-path locally on your computer**

*You can Skip this section if you just want to view the project through the Heroku link*

This section is for those of you who want to run my project locally on your computer. As previously stated you can access the entire project from the Heroku link provided above.

My project is run using Node.js.

Express.js is used as an application framework. Pug is used as its view engine, CSS is used for styling, PostgreSQL is used as its database, and client-side JavaScript is used to dynamically alter some of the code.

*In order to run the project on your computer you will need to install node.js*

1. Make sure you have the latest version of Node.js installed on your computer. If you don’t you can download Node.js for free here: <https://nodejs.org/en/>
2. Verify that you have the latest version of Node.js installed. In your command line run:

npm -v

This should return: 6.70

But version 5.6 and above should be fine too.

If npm -v does not work this means that node.js is not set up correctly.

1. Once you have verified that node.js is set up properly, navigate to the project folder in command prompt.
2. You will now need to install the project dependencies in the project directory. Enter the following in command prompt:

npm install async

npm install body-parser

npm install compression

npm install cookie-parser

npm install debug

npm install express

npm install helmet

npm install http-errors

npm install morgan

npm install pg

npm install pug

npm install serve-favicon

npm install --save-dev nodemon

1. If you have followed the steps up to this point, you should be ready to run the project. Again make sure you are in the project directory and run:

SET DEBUG=pokemon-path:\* & npm run devstart

1. Open your favorite browser and type the following into the URL:

localhost:3000

1. You should be good to go. Enjoy!

**Scenarios:**

**Paper Prototype YouTube Link:** <https://youtu.be/9WzDLAp5OXM>

## Scenarios

### Scenario 1 – Cardan Figures out what game he should buy next

#### Situation

Cardan is a university student who has been playing Pokémon games since he was a teenager. He has always dreamed of being able to catch one of every Pokémon. He currently owns 8 different Pokémon games including one of the newest games. He isn’t sure if he will be able to catch every Pokémon with his current set of games, or if he should buy a new game.

#### Outcome

Cardan uses the Pokemon-path website to quickly find out which games he should buy next. He starts by checking the boxes of the 7 games he already owns and then hits the submit button. Within seconds all of his questions are answered. He is shocked to learn that despite owning 7 games he still has 85 Pokémon he would need to catch to have obtained one of every Pokémon. The website generates a recommendation to play Pokémon Ultra Moon next as it would net him an additional 58 Pokémon given the games he currently owns. In addition the website gives him the list of all 85 Pokémon he currently can’t catch in game in case he wants to try to obtain those Pokémon through trades with his friends instead of buying a whole new game. Cardan is happy to have a clear path to follow to accomplish his goal.

### Scenario 2 – Taylor finds the optimal way to catch every Pokémon

#### Situation

Taylor is 16 and has been a fan of the Pokémon anime for a long time. She also plays the mobile app Pokémon Go, but hasn’t ever played one of the “core” series video games. She is looking to buy her first game and is excited about getting started and eventually catching one of every Pokémon. She knows there are a lot of games out there but doesn’t want to play every Pokémon game that’s ever been made. She is a busy high school after all. She wants to catch one of every Pokémon using the minimum number of games.

#### Outcome

Taylor uses the Pokemon-path website and directs herself to the new players link. On the website she finds the perfectly optimal path of games to play in order to catch one of every Pokémon. To her shock she learns that she only needs 5 out of the 29 games in order to catch all the Pokémon! She is excited to get started now that she knows exactly which games to purchase.

**How to test the Scenarios**

**Scenario #1**: Click on the big red button when you first enter the site. It will take you to the your path page/ Pokémon path calculator. Check whatever buttons you want, that correspond to the games you own.

*For example:* If you were to check the boxes for Sun, Alpha Sapphire, White 2, SoulSilver, Sapphire, Emerald, Gold, and that you have access to the Dream Walker the site will tell you that:

1. Are able to obtain 729 out of 807 Pokémon.
2. You are unable to obtain 62 Pokémon through in-game means
3. It will recommend 5 potential games to play next that will net you the most new Pokémon. Out of all the games Pokémon Y would net you the most new Pokémon at 49 new Pokémon.
4. It will list the Pokémon that you are currently missing.
5. If you click on any of the missing Pokémon, it will take you to a new page dedicated to that particular Pokémon. It will list every game the Pokémon is available in, and how you would go about obtaining it in that game.

**Scenario #2**: Click on the link that says New Players. It will take you to a static page that lists the optimal path to catch every Pokémon if you are a new player.

**Internationalization Requirements**

**User Inputs must accept Unicode:** Click on the link that says “My Account”. If you enter your name in the box you will find that the text input accepts Unicode.

**Dates must appear in the users local time**: On the My Account page, you should see the current date and time in your browsers local time at the bottom of the screen.

**At least one string must be localizable**: On the My Account Page you should see a sentence that says something to the effect of “This sentence changes based on your browser language setting”. When you first enter the page, if your primary browser language is set to English, you will see the sentence in English. However if your primary language is set to Spanish, French, German, Chinese, Hindi, Arabic, Malay, or Russian, you will see it in that language instead.

You can change translation of the string without changing your browser setting. Just select one of the other languages on the My Account form. This will override your browsers language preference and display the sentence in your preferred language instead.

**Program must store information about the user**: When you submit the My Account Page form, your First Name selection, and your language preference will be stored in local storage. If you go through the “Your Path to Catch’em All” tab after and select your games again, the website will greet you by name. The header will say “YourNameHere’s Path to Catch’em All” instead of just “Your Path to Catch’em All”.

**How the code is organized**

*For those who want to understand what the code is doing*

Naming conventions I used:

Pug files: naming\_convention

JavaScript files: namingConvention

Names of GET and POST functions: naming\_convention

All other JavaScript/Node.js variables: namingConvention

Variables in pug files: naming\_convention

The different Files/Folders:

*app.js*: sets up the modules and routes, and establishes pug as the view engine.

*routes> index.js*: Establishes the routes for the program. This determines what URL routes correspond to which website pages. Call on controllers> mainController.js to actually render the pages.

***controllers> mainController.js***: Holds all the functions that actually render the website pages. This is where most of the app logic is written. This is also the file which connects to the PostgreSQL database.

*My\_modules*: This is where keep all my custom Node.js modules. The functions contained inside them help run the app, create the SQL queries, and keep track of Constant values that are referred to multiple times throughout the website.

*Node\_modules*; This is where all the node modules are stored. I didn’t write any of the code here. You can just ignore this part.

*Local>Stylesheets> style.css*: This is where the CSS file that adds style to the webpage is stored.

*Local>images*: This is the folder that stores all the website pictures.

***Views***: This is where the pug files (the things that eventually turn into html files) are stored. It is also where all the client-side JavaScript files are stored, as well as the moment.js module which is used to format dates. I actually use client-side JavaScript to store any information about the user.

*Package.json*: This lists all the project dependencies.

*Pokemon.db*: The PostgreSQL database.