Zach Brown 4/30/2018
Professor Mark IGME 330

Project 3 – Weather Watch API

Name: Weather Watcher

Application Definition Statement: A weather tracker for casual people trying to figure out what to wear.

Features: Weather of the area, clothing to wear depending on the temperature and precipitation and use of node.js to find the current location and specific weather data depending on the city.

A list of clothing to wear on a specific day (below 40 degrees long sleeve, above 40 degrees short sleeve shirt, to wear a heavy jacket or a snow/ice storm.)

Effects of what weather is will be apparent and possible icons of what weather will be on a map API.

Design Decisions

Decided to go with the same format as my prototype 2, having 2 buttons which someone can search for a city or current location. What I changed, however, is that I added flex movement and flex controls to the map container. This means when the search and current location button is pressed, the map has a smooth transition to the left side of the screen and the right side of the screen shows the weather data from the API as well as what to wear depending on the weather.

Technical Decisions

Used 2 API's, the Google Maps API and a Open WeatherMap API. I also implemented the Node.js implementation with Heroku and building up the server as well to implement this. Decided not to use sound effects for my project do to it bringing away from the main objective of my application, a simple tool that shows weather data and what to wear depending on that data given and sound would possibly take away from that aspect and become a distraction if someone is playing music on their computer for an example. Also the information loads up slowly due to Heroku being a free source and not being the fastest when it comes to loading up data and having to connect to the server, so the application may take around 15 seconds before a search could be made. Did not use any other JS Libraries.

HTML5 Technology Used

Used containers, used flex controls on the map container and information container, inserted 2 API's, google maps API and OpenWeatherMaps API. Also put in icons using fontawesome, a source where I was able to insert icons for weather like clear skies, rain, snow and clouds. Also used the Quicksand font for the Google Maps API and weather data. Separated CSS and Javascript into different files (styles.css and script.js) to keep my information organized.

Extras

Node.js implementation, icons depending on weather patterns, current location button and implementation, weather specifics based upon data, clean and modern design that's easy to use.

Grade Yourself

I believe for this project I should get a 95% due to fulfilling all the requirements, but going a tad beyond by getting node.js to work, building up my server code, and making a application that people are most likely to use in an every day circumstance (figuring out what to wear).