Sample Project Self Evaluation

Week 6 - Overview

I started working on my project this week. I had an idea of what I wanted to do but I did spend a little bit of time looking at https://www.programmableweb.com for other API that I might use. The hardest part of the deliverable for this week was the UI MockUp. I wanted to use a tool like Balsamiq but the learning curve was a little much for given the rest of my workload. I ended up using copy and paste from similar web pages to create my MockUp.

I spent about 4 hours this week.

Week 7 – API

Working with API documentation is a little overwhelming. I find it hard to read the API documentation and decoding the API results JSON objects is complicated too. I had used both of the API in the weather app so I already had API keys and some idea how to get started.

I spent about 4 hours this week.

Week 8 – Static HTML / CSS

I AM NOT GOOD AT CSS! Creating the static HTML pages for sample content was not hard but getting the pages to look professional was really hard for me. I'm not happy with where I ended up but I know I can work on the look and feel more if I have time after I've completed the JavaScript.

I'm not really sure how do the static graphs, given that chart.js creates the graphs. I ended up creating the tabs like we did in the Event app for class and then hard coding the JSON objects for weather data and writing a tiny bit of JS code to make the charts show up on the page as you click the tabs.

I also set up the JS development environment this week. That basically means that I created a folder, copied and pasted the package.js and webpack.config.js files from the Event app for class, edited a couple of things and started working. Everything appears to be working ok so far.

I spent about 12 hours during these 2 weeks. Most of that time was CSS and I feel like I wasted a lot of time.

Weeks 9 and 10 – JavaScript Implementation

The implementation was easier than I thought it would be, but I worked for a couple of hours every day and I asked a ton of questions. I was glad that I got good at debugging and at working with the debugger in the browser as I worked on the labs earlier in the term because I spent a TON of time debugging my JavaScript code. I was also glad that I spent time in each lab cleaning up and writing comments in my lab code. I was able to find sample code for lots of the things I had to do and understood the code well enough to modify it as a result.

At this point everything is working but the history feature. My app writes to local storage as the user enters a zip code or city/state but it doesn't retrieve that information from local storage, display it on the page or allow the user to click on a location to get weather for that location directly.

I spent more time than I wanted cleaning up and documenting my code when I was finished with my implementation. In retrospect, I should have cleaned up my code every time I finished a feature. I think that would have saved me time in the long run.

I spent about 30 hours during the last 2 weeks working on JavaScript implementation.

Summary

I spent a total of 50 hours on my project and am generally happy with it.

Requirements

- Include one or more web pages, have a clear objective and be user-friendly. DONE.
- Have a professional-looking UI that uses Bootstrap or another CSS framework with which you are familiar. DONE, although it's not as professional-looking as I would like.
- Use client-side code written in JavaScript. Each page must be encapsulated using an ES6 style class. DONE.
- Include at least one AJAX get request. DONE. Uses google maps and open weather map.
- Include one or more of the other JS technologies introduced in the course such as: local storage, html canvas, regular expressions, chart.js, google maps API. DONE. Use local storage and chart.js.
- Be hosted on citstudent.lanecc.net. Production code should be transpiled to a version of JavaScript that can be interpreted by modern browsers. DONE.

Deliverables – 76 / 80 points

- Project overview / description and UI MockUp – 10 / 10
- AJAX call documentation – 10 /10
- Static HTML / CSS implementation – 16 / 20
- JavaScript implementation – 30 / 30 – I hope
- Self-evaluation – 10 / 10