Process

1. Attain some basic HTML/CSS knowledge
    a. Used Team Treehouse and took introductory courses for HTML and CSS (~6 hours)
    b. Mozilla Developer Network was also very helpful
2. Attain basic JavaScript knowledge
    a. Found a series of articles that explain how to utilize APIs with Javascript
        i. https://www.taniarascia.com/how-to-connect-to-an-api-with-javascript/
    b. Need to add JSON view extension to Chrome (done)
3. Choose IDE
    a. Researched IDEs for web development, specifically for HTML/CSS/JavaScript.
    b. This website gave 14 highly rated web development IDEs https://tms-outsource.com/blog/posts/web-development-ide/
    c. Landed on Visual Studio Code (free and highly rated in several other articles for web dev, also has great integration with GitHub)
    d. Watched introductory video on using and setting up VS Code through LinkedIn Learning (~ 2 hours)
        i. Need to remember to cancel in 30 days!
4. Figure out hosting/domain
    a. As I did more research on what is needed to run a website it seemed like every resource told me that I needed the website to be hosted on a server for it to properly display the website. Digging in further, I found that it is possible to get a URL for the source code files through GitHub pages (https://www.youtube.com/watch?time_continue=228&v=CJLb8UUIJPg).
        i. This is great because I was going to use GitHub for version control anyway
    b. Emailed Gabe to see if loading the source files on GitHub would be appropriate, he said that would be OK.
5. Write User stories
    a. Learned about this from my team mates at Hack-A-Pipeline, we wrote out how a specific user might utilize our Alexa Skill then focused on solving a specific story that would yield a proof of concept for the Skill.
    b. Decided that would be a good idea for beginning to design this webpage, found other resources to guide me in writing the User Stories on my own
        i. https://www.romanpichler.com/blog/10-tips-writing-good-user-stories/
        ii. https://www.mountaingoatsoftware.com/agile/user-stories
    c. After writing out some stories I decided that my webpage submission would focus on the Epic “I want to know the weather at a Penrod office right now” for the User “I am a Penrod employee about to leave for work in the morning and I need to see the weather at my local office so that I can dress accordingly for my commute.”
        i. Open Weather provides an API for the current weather based on a City ID, coordinates, or zip code.
        ii. Was concerned about adding on too many features especially since this is my first time putting together a website from scratch. I wanted to choose a concept that would match the requirements provided to me that I could also successfully deliver in a reasonable amount of time.

        iii. Can always add features on later.
6. Determine Website design/flow
    a. Based on the User Stories I decided to focus on:
        i. The user opens the webpage
        ii. The webpage clearly displays its purpose
        iii. The website has buttons in the header portion for each Penrod location City
        iv. When the user clicks on the button the page displays the current weather in that city
            1. Temperature in deg F
            2. Photo of city experiencing current weather (i.e. if it is snowing in Milwaukee currently then a picture of Milwaukee while it is snowing is displayed)
    b. Required to use Lightning Design System for CSS framework
        i. Download files
        ii. Read documentation
        iii. Focus on getting the website to work w/ API calls first, then worry about "making pretty" with CSS
7. Figure out how to use Open Weather API
    a. Need to know how to call to the API and what information it returns
        i. https://www.youtube.com/watch?v=uxf0--uiX0I - tutorial on working with JSON
    b. Need to figure out how to use the returned information to fulfill design requirements
        i. Use JavaScript to dynamically display image based on City selected, the current temperature, and the current condition
            1. Ultimately used a series of if/else statements to get images to display. Would like to refactor to use a hash table to get the image address
                a. Use cityID (from Open Weather JSON) and returned weather condition (clear, rain, thunderstorms) to create hash position ID
        ii. Use Javascript to display the city, temperature, and current condition at the top of the page

8. Add links to this document and the User Stories document