**Reflection on Resumé Project**

During the duration of this project, I picked up some very important new skills and modern design practices. In my opinion, the most useful modern layout and design practice that I got firsthand practice with is CSS Grid. I learned how to setup a grid with the grid-template-columns and grid-template-rows properties in my CSS stylesheet, as well as how to allot specific elements into the different grid cells/units with the grid-template-areas property, which is extremely powerful for reflowing a site that uses a mobile first approach so that it looks appropriately styled for a tablet and/or a modern desktop monitor. I also learned how to implement Flexbox designs into my resume with the display: flex property which is great for styling elements on a single line, or that you would like to have wrapped down to the next line, with the flex-wrap property, on resizing of the browser window with media queries. Flexbox also allows me to set different spacing between the elements that are displayed flex using the justify-content property.

Another very important thing we learned about is microdata implementation. Microdata can be implemented in at least two ways that we have learned about, Schema and JSON-LD. These approaches offer two unique implementations but accomplish basically the same thing. They allow for the key information in your site to be parsed and laid out on search results pages. Schema uses the approach of embedding itemprop’s, itemscope’s, and itemtype’s in the HTML of your site, while JSON-LD uses an externally linked JavaScript document approach that is linked from your js folder of the project, and contains an “@context”: “property context url” and “@id”: “id property url”, followed by several “name of property”: “field related to property” fields. As mentioned above, both methods allow for crawlers to parse the document and display the results on search results pages and accomplish essentially the same goal.

With the lessons I learned during this project out of the way, here is how I implemented my project solution. I started by going through the given schema layout and filling it with my data. As requested, I verified that Google’s Structured data testing tool was able to parse the document and populate my information into an ordered layout, which it was able to do. I next went about styling my documents ‘mobile first’ approach with some key starting points demonstrated to us in class. For example, the standard fixes, typography, fluid images, design time styles, and global styles ‘mobile first’ approaches. This allowed me to easily layout the site in a mobile friendly, logical, way. Some of the notable implementations in my mobile first design are my use of flexbox for a navigation element and CSS Grid for the dl element that listed my personal information. Once I’d finished my mobile first approach, I moved into my first media query breakpoint for a tablet view. For my tablet view I reflowed the personal details section of my site with CSS Grid to be a 2x2 Grid that displayed elements more horizontally than the vertical display of the mobile first design (increased screen width means increased container max-width of course). In my tablet view, I also reflowed my skills, experience, and education sections to a horizontal display as I mentioned above for the personal details section. However, for the experience section, I decided to reflow my current job to be above my two previous jobs and added a border around it to highlight it as my current occupation. I implemented that change with CSS Grid. With my tablet view looking good, I moved on to the desktop view. For this media query, I again increased the max-width of my container to fill more of the screen. I kept most of the flow for my site the same as the tablet view with the increased max-width of course but did go ahead and reflow my experience to a more visibly appealing, to me, layout of 3 vertical columns, through CSS Grid once again. Other than that, my desktop version pretty much mirrors my tablet version as I’d deemed it a well flowed design. Now that I have outlined my implementation, I’ll move into what I would do differently and the changes I made based on classmate feedback. The biggest change I would make in the future would be involving my tablet media query. I made the mistake of using “@media screen and (min-width: 725px) and (max-width: 900px)”. I’m sure you can already see the problem here, but to be abundantly clear, when I set the max-width property for the media query, it prevents any of my previous design implementations from being present above 900px. Long story short, I had to completely restyle my site to what it was before the breakpoint by copy and pasting a lot of the tablet view styles as well as adding some new properties to get the site to look how I wanted it to. In the future, I would use the media query “@media screen and (min-width: 725px)” so that none of my styles would be lost at the onset of my next breakpoint. Another change that I did get the chance to go back and fix was the use of CSS Grid for practically everything. While CSS Grid worked without issue in those places, it was more appropriate and much easier to instead use Flexbox. For example, things like my main navigation and my related links section which are only on one line and distributed evenly, it makes more sense to use flexbox rather than a 1x4 or 1x3 grid.

Now for the changes made upon feedback from a classmate. The biggest changes I made from classmate feedback were increasing my heading fonts as well as underlining them so that they were clearly distinguishable from the content in their respective sections. Another change that I made was to change my unordered list of skills to display flex instead of having them displayed vertically and centered. Both of those recommended changes I agreed with and was happy to implement. One change that I heard out but decided not to implement was to have my education heading moved to left aligned because it looked like it disrupted the flow of my site to the classmate. I understand what she was saying, but upon making the change I felt it looked slightly better the other way. Nonetheless, it’s always great to get feedback from others because we aren’t creating sites for just ourselves, and just because I may love my sites design, others may find it terrible.

To conclude, I would like to reflect on the strengths and weaknesses of my overall design. The biggest strength, in my opinion, is the use of CSS Grid to reflow the order and layout of elements at each media query. This implementation ensures that no matter the users screen/viewport size, they will be able to view my content clearly. Another strength I would say is the use of an SVG graphic rather than a png or jpeg. There are of course benefits and issues from using an SVG image but, I felt for this project it was best with there being three breakpoints, and the need for a high resolution, fluid image. For the weaknesses of my implementation, I’d say, for older browsers anyway, the use of CSS Grid is a weakness and a strength. Many older browsers do not support CSS Grid and that could be a serious issue with it’s prevalence in my design and implementation. Another weakness is the fact that I didn’t include @supports, aka feature queries, into my solution. The reason this is a weakness here is that it could be used to solve the problem on older browsers that CSS Grid could cause. With that being said, I think I should go ahead and wrap this up, this is no longer a “short reflective statement”, and for that I apologize, I got a bit carried away. During this project, I learned tons of new things, and I can’t wait to see what the next project has in store for us!