Coming into this class, I was unsure of what to expect. Computer Science/Computer Engineering and Web Development were very different form each other. Programming meant drawing diagrams to try to understand how to construct the actual program, and worrying about thing like running time or memory accesses. Web design meant drawing sketches to make sure a website is easy to navigate whilst aesthetically pleasing; things like browser compatibility and mobile compatibility were also concerns.

It was easy transitioning in some ways. Good practices like code maintainability, comments, and avoiding long-winded unnecessary code in programming translated well to web design. Understanding what code should be in my HTML files and what should be in my CSS file(s) led to smaller code. The use of classes (or ids, which are similar but bad practice to use) and understanding how the browser parses the code based on classes led to faster code. Drawing wireframes and designs saved a lot of potential headache since it allowed easier visualization of both the webpage(s) and what code there had to be written. Determining what should be a block element and what should be an inline element was the hardest.

The main difficulty in transitioning was from focusing on programming for other programmers to see, to focusing on designing for possible non-programming users. While a minimalist approach is often the best (even if it is overused) approach, a webpage can be too minimalist and often leave a lot of empty undesirable unused space on the page. However, cramming too many things into one page can result in a webpage loading slower (especially with movie images or flash animation). It can also be an eyesore for the user (like the homepage of Craigslist). There's prioritizing what should be linked to another page and what should remain on the page. You don't want the user to be lost trying to click link after link to reach a page, but neither do you want the user to get lost navigating a single page because the link I small and hidden in some obscure part of the page. The coding part is usually easier than the design part.

If I had a better understanding and appreciation for this difficulty earlier in the semester, I would have looked at more websites and study their layout for inspiration. The key is to understand the flaws in a website and why that flaw did not work. Even a good website like Craigslist has a lot of flaws.

Initially I was hesitant to do a complicated design, since I believed that a minimal design is always better. However, looking through the website, I realized that such an approach was not going to work with a website like Craigslist. Craigslist is a very interactive website. It's not like Tumblr or Pintrest where people simply scroll or comment on images. Users search and type a lot on the website. It would be easy for a user to miss an item on the page because they are scrolling too fast or are too inattentive. With that in mind, I aimed to get rid of features I thought Craigslist did not need (there were a lot) and add some features that I believed Craigslist would benefit from. I also took inspiration from ideas given by my classmates, such as an integrated Google Maps and calendar notification system

One of the biggest things I remembered was that users respond to bigger than normal, but not too big text. It makes it easier to read the text and the user won't be straining their eyes trying to read a big block of text. Another thing I remembered was that users also respond well to pictures. A good picture is often worth more than a thousand words. What might take 2 paragraphs to explain can be summed up in a single picture or diagram, which often takes less space on the webpage. When users are scrolling through a search page, they usually don't search past the first page. They will only search 2 at most before refining their search. On the first page, their eyes will focus heavily on the upper left hand side (since most users around the world read left to right, with perhaps the exception of Arabic and Hebrew users). With this in mind, I focused on designing the webpages with regards to easier user navigation, with heavy emphasis on search engines (which I assumed a back-end developer could tackle).

Although this approach solved most problems I believed Craigslist had, I felt there were simply too many items on the pages. Due to limited space, I could not allot some spaces between blocks on the page. For example, on my item page, there was little space between the top of the picture's title and the bottom of the search engine. I may have also overestimated the capabilities of the search engine, since it may require the user to understand how to navigate it. For example, on Google's search engine, if a user types in a sentence with double quotes (like so: "example"), Google searches for pages with those exact quotes. Intricacies like that would result in better searches, but also put more burden on the user to learn, which might turn some new users away. If I had more time, I would have minimized some blocks and make other blocks bigger. I would have also created a user-friendly tutorial page to help newcomers learn. I would have also created a mobile page since a lot of users simply browse on their tablets.