There were many lessons learned from this class and from making this final project.

Design, implementation, how to read and use a variety of tools and frameworks are just some examples. Let’s get into it.

Designing a website is much more complicated and complex than I had imagined. I thought you could just add some html, style it a little bit and ‘voila-easy-bake website. The truth is designing a website requires the same and intentional design as good code requires. The biggest lesson learned from designing this final project was “design first-write code last”. This is almost a maxim in software design, but I didn’t think that making a website was “real” coding. Fortunately, I had an opportunity to reinforce that law while designing this website. For example, when I started this project, I just went through each element, adding them one by one to the ‘index.html’ page with vanilla html without thinking about how the styling would be implemented. Once I got to the styling, it became very easy to get lost in the weeds and get confused about what was implemented, and what was not. Eventually, I just tossed everything, and drew some pictures of how I would like this website to look like. That exercise and subsequent thoughtful design made it much easier to make it work.

Implementing the css without using bootstrap was much harder than I had imagined. I had many different attempts to make it responsive, and size up appropriately fail miserably. Again, I went back to the literal drawing board. I also used previous homeworks as templates for the final product-using much of the format from homework 4. I just added some different content and some additional stylings as needed.

The first attempt at making this navigation bar was a vanilla piece of work. That seemed a bit too much to style, but it looked good. It did not respond correctly to the resizing of the screen, despite using @media queries. I turned to the bootstrap documentation and copied and pasted the boilerplate template for a navbar, and that worked great, but I could not get the background colors that I wanted to be prevalent in the website. I tried to do some more individual styling for the background, but nothing seemed to be permissible. In the next iteration, I would like to have the navigation bar be used from the bootstrap website, because the output is clean and the hamburger menu can be readily adapted to be more interesting. I think I may need to read more on using background images behind the navigation bar, and that may be the solution to having the background I want displayed.

The next big lesson learned was how to implement a responsive image appropriately. We did use one in the previous homeworks, but these seemed to be more complex and would not stay in line. Again, using bootstrap and following the directions on the website solved many issues. MDN webdocs and w3cschool websites answered and showed appropriate examples for how to style, size and set images appropriately.

The last issue that I still can’t figure out is how to get the contact button from the left to the center. I have styled it such that the element has a variety of ids, and classes associated with it, and have added “align-items: center” in all of them to no avail. I have changed the display from block to flex and nothing has changed. In general, I am not too satisfied with the form documentation. It doesn’t work properly. The aspired for behavior was to have the user click on the submit button and then an email would be sent. I would not have it sent to a real email currently, because of lack of anti-spam tools built into the website, but it does not work. A more careful reading of the documentation at [formsubmit.com](https://formsubmit.co/) will be in future iterations.

Writing clean code is challenging. While looking over the css file, there are many opportunities for refactoring. The number of repetitions counted cursorily would make Martin [Fowler](https://martinfowler.com/) throw this code against the wall. Every initial code project requires much refactoring and this one is no exception. This website will be hard to maintain, and update due to the number of different spots that an element will have to be modified. One example is all of the stylings for the images in the body. They are all the same, and will all have to be updated in the future. The other big issue is how to build tests for this piece of software. There are none. Some good options are out in the wild to test html,css and javascript, but I did not take the time to learn them.