

Ryan Filgas
Intro to Web Development
Final Project

Github Repo: <https://github.com/rfilgas/rfilgas.github.io>

Deployed Website: <https://rfilgas.github.io/>

Getting Started:

Getting started with the website for our project I knew I wanted to start with something simple so I went top down with a navbar first, and side bar for my portrait and simple lines through the center to connect the different sections. As there isn't much time my goal is of course a minimum viable product that I'll build on if I have the time later.

To get started I went to github, created a repo, and added html, css, and javascript files, then pulled the repo down to my local environment to get started. I referenced the class materials for where to put the commands that connect the css and javascript files to the document. Following this I looked at a couple squarespace themes for inspiration, but decided to just build and see what happens. The first thing I did was rough out the general structure of the website. Having decided on a one page site I built the following structure.

Block 1 - Navbar -> Static navbar (won't move) -> Basic links to shift down page. White with Black and grey type.

Block 2 - Intro / about section --> Dark Blue with bold White type

Block 3 - Previous Work -> Basic CV section -> White again with black type and gray accents.

Block 4 - Projects -> Need 3 projects: name, description, link —> lighter blue with black type

Noise Fusion,
Maybe make a computer vision repo with basic algos,
GCP Data Pipeline Infra Project

Block 5 - Contact -> functionality not required —> dark blue block.

Navbar Steps:

Once this structure was worked out I started on getting a navbar operational. I went straight to bootstrap for this as it seemed like the natural thing to do.

<https://getbootstrap.com/docs/4.0/components/navbar/>

Since this is a one page website I didn't like that the navigation would only be available at the top of the page by default, so I found out how to keep the navbar in place using bootstraps "sticky-top" setting, however this led to its own challenges later on. I began roughing in components in html and realized that the menu when scrolled down would be overlapping part of my page. In order to fix this I not only had to add some margin adjustments, I also had to change how to manipulate where on a page targeted links go when a user clicks.

Stack overflow saved the day for this particular issue as well as getting the navbar to close once a link is clicked. I was able to simplify the solution to fit my needs.

<https://stackoverflow.com/questions/10732690/offsetting-an-html-anchor-to-adjust-for-fixed-header>

<https://stackoverflow.com/questions/16680543/hide-twitter-bootstrap-nav-collapse-on-click>

Roughing in the general layout:

Once the navbar was apparently complete I continued roughing in the general layout. First I pulled bootstrap components from earlier assignments, essentially just to make some grids. I realized once it was too late that I could have had much greater flexibility in my design had I used a single grid for everything rather than using nested grids like I did.

For colors I simply googled "color picker" and used the built in tool google has in the browser to select and add new colors. I used these in addition to several shades of bootstrap gray as I was testing styling options. I used these in addition to the colors from our styles sheet from previous assignments.

For the first content I accessed previous discussion posts we wrote for weeks prior to get my first project information in as well as a roughed in introduction. I spent some time trimming the intro down and refining the description for project 1. Following this I chose to create div elements in html to serve as entry points for javascript. As there is going to be a lot of text it seemed more straightforward to insert the paragraphs from the .js file.

Following this I attempted and succeeded in embedding a pdf file, however as this is a one page website it can complicate navigation, and isn't as mobile friendly. For this reason I abandoned the idea, however I did implement successfully implement it using the bootstrap docs.

<https://getbootstrap.com/docs/4.0/utilities/embed/>

Following working with the pdf I decided to create a side panel as the style of the website wasn't harmonizing very well. It took a good amount of time to do this in addition to testing the

browser at various sizes to make sure there weren't any side effects. After some adjusting I was able to get it to a point where I was happy with how responsive it was, however I went to go tweak this later. At this point I had a left panel, a navbar, and different sections, but not too much else.

Adding in Basic Content and More Style Refinement

For this portion I started by completing the building blocks in html for the text blocks that would be getting added later in javascript. I referenced prior class materials for how to embed images and adapted this to html as I would be inserting them with the text blocks there. Following this I checked the mdn docs for how to underline text (and learned why I shouldn't do this for stylistic reasons later on). <https://developer.mozilla.org/en-US/docs/Web/CSS/text-decoration>

Next I wanted to add a design element of a downward arrow, but I wasn't sure quite how to approach making shapes in css, so this was a google problem. I was able to adjust these to my needs, however centering them inside of the container proved trickier than I had hoped. I had a couple issues, the first being that adding an offset changed other areas of the site, and using this elements offset values to make breathing room exposed a square hole in my background in the top right corner, so I had to find ways around this.

https://www.w3schools.com/howto/howto_css_vertical_line.asp

<https://css-tricks.com/snippets/css/css-triangle>

After some time getting the arrows where they needed to be I used these as dividers for my different sections rather than my initial idea of doing color blocks. To tie up adding in basic content I pulled a contact form from bootstrap and dropped it in.

Adding in second and third projects and styling the CV

The first part of this phase was styling the CV. While I ended up reworking this later, this stage convinced me to start the work of realigning most of my elements in the right side of the website, but more on this later. Adding in the other two projects was going to prove a bit more challenging than the first as I had to write content for both of them and upload the last one to a new github repo. For all of them I went through my projects to pull images for the site and then used the same javascript code for the insertion as I did for the first one. Once I had these elements in I spent a good amount of time fighting with the layout so it would be consistent across browser sizes and scale properly on mobile. Following this I had a thought to create a link for my repos, so I downloaded the github logo only to not use it later. Instead I found a bootstrap button, and then used the mdn docs to find a formaction setting so it would directly link to the url. Following this I added all of the github links and styled the buttons.

<https://developer.mozilla.org/en-US/docs/Web/HTML/Element/button>

Cleanup:

Once a good portion of the site was done I went back into each file to organize, syntax check and clean up the code as well as leave comments where necessary. As I worked in a single html file I had a couple random divs where they shouldn't be and in general some things were out of order. I decided on an ordering convention for the javascript file so the styling areas etc. made sense, otherwise it can get easily jumbled. I also had a few tags that didn't include the '/' and this was causing unpredictable behavior before.

Accessibility:

Once I had my code cleaned up and all of my major components completed ran through accessibility audits and found a few problems as expected (I hadn't incorporated labels etc. yet). The biggest areas for improvement I found was actually the h1 – h6 tags. I had them out of order in some places and was using bold <p> tags where I should have been using an h2 for example. In addition I had a couple labels where they shouldn't be. I ran both lighthouse and wave, then fixed all messages. One of the major problems I had was the main menu. Getting it from bootstrap it was initially a list, however when I added anchors to it initially they caused an accessibility problem, so I converted them to divs. I realized this was counter productive to the purpose of accessibility changes, so I reverted to the unordered list and the issues didn't come back. Lastly I did a semi colon check on the .js file to make sure everything was consistent.

Final JavaScript add-ons:

I looked up how to make a deep copy of an object to copy style settings, but didn't end up using it.

https://developer.mozilla.org/en-US/docs/Glossary/Deep_copy

After doing this I worked on my form. The first thing I did was a setTimeout timer on the submit button, then changed the text and color once submitted.

<https://developer.mozilla.org/en-US/docs/Web/API/setTimeout>

After setting the submit button up for the form I switched gears to see what other elements can be added and I made an event to insert my photo on mouseover for all of the arrow components. The arrow components also change color on mouseover, and when my photo gets moved it can be put back by mousing back over my name.

https://developer.mozilla.org/en-US/docs/Web/API/Element/mouseenter_event

Following this I did some validation for the form including checking for an @ symbol in the email, requiring a max length and requiring that the text fields are filled out. I added alerts for all of these validations. Lastly I added a button to download a full pdf resume.

<https://getbootstrap.com/docs/4.0/components/buttons/>

Summary:

Overall I think the site came together well, however if I were to do this again there are some changes I would make. First I would better separate my css classes into categories rather than largely doing individual elements. I also should have separated color styles from structural elements in order to have greater flexibility. On the structure side it would have been useful to do one main grid instead of nested grids. This would have provided ultimate flexibility for this project. One feature I wanted to implement that I couldn't was an image that moves along as I scroll. The scroll implementations I saw while searching seemed complicated enough that I may not understand how they work, so I opted not to do those for this project as I would be using large blocks of someone else's code. At the beginning I wasn't as excited about this project, but towards the end when the final elements were coming together I ended up enjoying it.