Creating the Portfolio

# Resources Used

<https://www.feelingpeaky.com/9-principles-of-good-web-design/>

<https://ucsbcarpentry.github.io/2022-01-28-ucsb-webpub-online/01-introduction/index.htmlbrian>

# Introduction

For my first entry into the devlogs, I have chosen to create an article on the creation of the website that these will be hosted on. There’s not much more to be said, so let’s get cracking!

# Requirements

So, **What is it I need my website to do**?

Well, broadly speaking, I need it to act as both a window into my private coding life, and a resume to show to potential employers.

Cool, sounds simple. **So how does that translate into the content requirements of the website?**

Here are the sections of the website that I plan to create in order to meet my needs:

* Homepage – I need a page which draws people further into my site while providing them with an overview of what the site is and has to offer.
* Portfolio section – A page or couple of pages that showcase my projects and connects them to explore their repositories. Ideally I would be able to push visitors towards specific projects that I am particularly proud of.
* DevLog section – A section that acts as a repository for my devlogs. Possibly multiple pages that are sorted in different ways, and may be pinned.
* About Me Page – I want employers to know who I am. I will include some basic information such as my general working area, notable employment history, and interests (mostly those that lie within the field of computer science).

For the DevLog and Portfolio sections, I plan to only get the bare minimum working. If I decide to add features to the sections, I imagine they would be good subjects for future articles with a deeper dive.

On top of that, I would also like to make sure that my website follows a few principles:

1. Each page has a clear purpose and sticks to that purpose.
2. Keep the aesthetics consistent (using a template should take care of this)
3. Keep it easy to navigate
4. No clutter
5. Make sure that information is organized as consistently as possible in a left to right, top to bottom order and with important info prioritized.
6. Keep load times down. (AKA don’t screw up the template)
7. Keep it mobile friendly (Learn from the mobile friendliness of the template to keep the website mobile friendly moving forward.

I wouldn’t necessarily call these principles “requirements”. I consider them more as guidelines that most web designers follow that it would be prudent to also follow in my own endeavor.

# Getting Started

## Which platform to use?

The first thing I have to decide is what platform I wish to host my website on. Thankfully, this is not my first attempt to create my own website, so I am a little bit more aware of what the options are and their pros and cons than I was the first time.

The main contenders I am looking at are Wordpress, Squarespace, Github pages, and Firebase

The first time I attempted to make a personal website (for a non-web-dev undergraduate class) I used Wordpress because it took little effort to make a minimal site for a class. While it had some perks with its simple startup and the ability to host your website for free, I found myself frustrated with the difficulty in customizing my website without forking over a significant sum of money.

It’s important to me to host on a free site as at the moment I cannot justify the cost of ongoing fees to keep a website running while my wife goes through nursing school. So, as a paid service, Squarespace is out for the same reason.

That leaves Github pages and Firebase, which as far as I know are free to host on and are largely customizable without spending lots of money. There are many differences between Firebase and Github pages, but the main one to me is that Github pages forces you to create a static website and Firebase allows for dynamic pages. Firebase comes with some more bells and whistles like integration with the Google Cloud, database storage, and

## Setting up the repository

In order to use Github, I must first set up a repository for the website. This requires that you already have and are logged into a Github account.

Describe process steps

## Choosing a template

Downloaded a few templates from HTML5 UP and played around with what the site needed.

Tried Story

Ended up on Editorial

A screenshot of a computer

Description automatically generated

# Editing the Home Page

First thing I did was edit the titles and welcome messages.

Second thing is to create interest in and links to the portfolio and devlogs.

I then cleaned up the left sidebar and social media links.

I then chose to remove the personal information as I do not have a PO box or professional email that I feel comfortable cluttering. Perhaps I should change that.

I left the search bar unfinished

# Sidebar into Markdown and Jekyll

## What is markdown?

Markdown is a formatting language for plain text. See this article for more information on Markdown.

## What is Jekyll?

Jekyll is a site generator for static sites (which is what Github Pages allows).

It takes an input of Markdown (see above), Liquid (allows for some neat global variable tricks and the inclusion/reuse of code from outside the page), HTML, and CSS and compiles/renders them into that pages of a website.

Jekyll has some quirks that we will get into later, but overall it is an awesome tool that I expect to save lots of typing in the future. See this article for more of the specifics on how to use it.

# Back to the Homepage: Jekyll Edition

## The wall

After I had gotten all of the text pieces into Markdown format, I then went to look at the final result:

Picture of my first attempt with Jekyll.

This wasn’t quite what I imagined. A text wall of code isn’t very… websitey (definitely a technical term, don’t look it up).

Back to research!

A Closer attempt

Ah, it seems that I had some divs placed in the wrong spot on my code. Rookie mistake.efd

# Creating the Portfolio Page

# Creating the DevLog Section of the Site

# Adding this Article and some of my old Projects

# What is next?