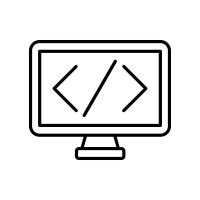
Senior Project – Foundational Learning and Project Plan

Web Development and Design

How Can I Effectively Design and Code a Website That Serves a Practical Purpose?

Cole Manning



# Introduction and Table of Contents

In this paper I will be going through the process that I will take to create my website, while also sharing my learning so far.

In the **Hosting** Section I will go over the role a web host plays and why I choose my host.

In **Software and Programs**, I will start by explaining the use of *Git and GitHub* in Web Development and the important role they play in version-control. Lastly, I will speak on the possible options for source code editors, and why I settled on Sublime Text over all the others.

Moving onto **Coding Languages** I will start with the basic languages of HTML and CSS and their role in web design. I will then move on to the slightly more complicated JavaScript and finish the section off with Databases and their use.

The next section will be around the process of **Designing** a website. I will talk about the use of design programs and some elements of design specific to the web.

In the second part of this paper, the focus will shift from my foundational learning, to the logistics of my Project Plan including: **timeline**, **foundational learning next steps**, and **collaboration**.

Table of Contents

[Introduction and Table of Contents 1](#_Toc535407099)

[Hosting 3](#_Toc535407100)

[Software / Programs 4](#_Toc535407101)

[Programs 4](#_Toc535407102)

[Web Browser 4](#_Toc535407103)

[Git & GitHub 5](#_Toc535407104)

[Text / Source Code Editors 5](#_Toc535407105)

[Coding Languages 6](#_Toc535407106)

[The Big Three (Client-Side) 6](#_Toc535407107)

[HTML 6](#_Toc535407108)

[CSS 6](#_Toc535407109)

[JavaScript 7](#_Toc535407110)

[Server-Side 8](#_Toc535407111)

[Databases 8](#_Toc535407112)

[Structured Query Language (SQL) 9](#_Toc535407113)

[Miscellaneous 9](#_Toc535407114)

[A Note on PHP 9](#_Toc535407115)

[Design / Layout 9](#_Toc535407116)

[The Programs 9](#_Toc535407117)

[Planning 9](#_Toc535407118)

[Making a Site Map 9](#_Toc535407119)

[Wireframing 11](#_Toc535407120)

[Creating the Full Design 11](#_Toc535407121)

[Fonts 11](#_Toc535407122)

[Colors 12](#_Toc535407123)

[Using Artboards 12](#_Toc535407124)

[Notes on Responsive Design 12](#_Toc535407125)

[Foundational Learning Next Steps 14](#_Toc535407126)

[The Product and Content of My Website 14](#_Toc535407127)

[The Idea 14](#_Toc535407128)

[How This Will Demonstrate Learning 15](#_Toc535407129)

[Collaboration 15](#_Toc535407130)

[Timeline 16](#_Toc535407131)

[Bibliography 17](#_Toc535407132)

# Hosting

The web host stores information and code/content for your website, and makes it available on the internet via their servers. The main ways that hosts vary what they offer is how much storage they give you, how fast their connection is (bandwidth), and what price they offer for their services.

You will need a domain for your website, which is what visitors will type into the URL to access your site (ex: example.com).

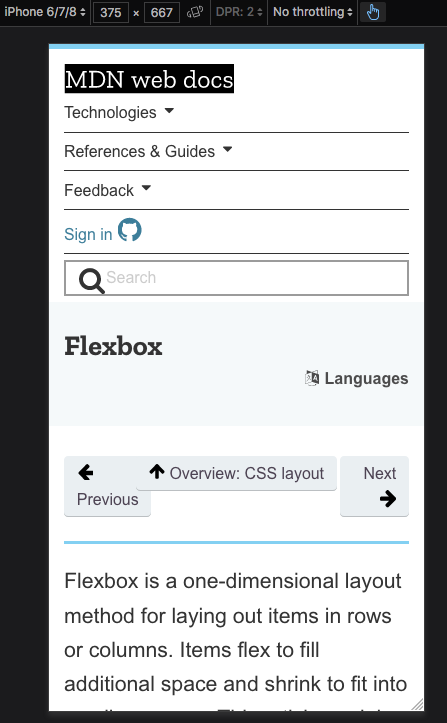
There are many different options for providers of these services. The best way to find them is to just start googling and comparing the features they offer and the cost.

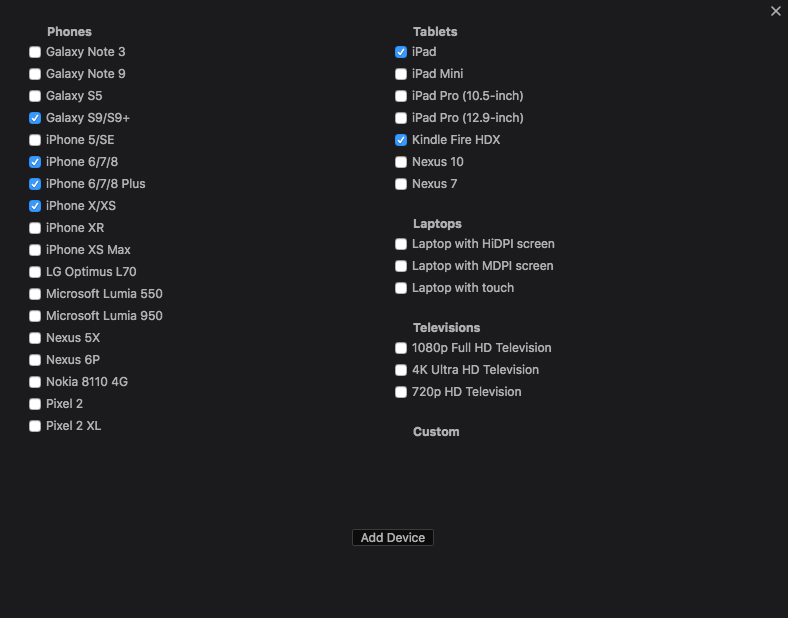
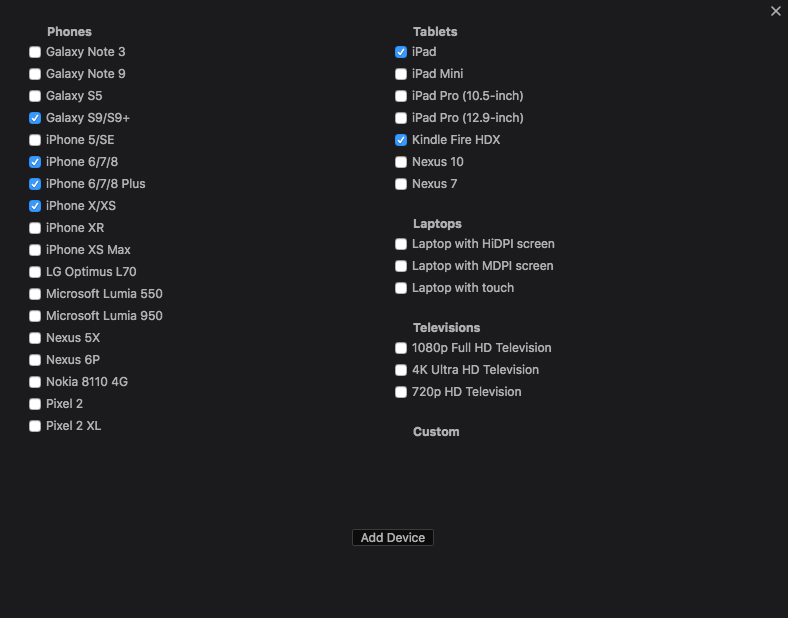
For my project I plan to use Black Host, simply because I already run a service with them, and they allow for the use of addon domains at no additional cost. An addon domain is an additional domain that is stored as a subdomain of the main site but is displayed without the main domain (ex: “subdomain.com” instead of “subdomain.domain.com”) (cPanel, Inc.). This will allow me to create a full site for this project using the host I already have.

# Software / Programs

## Programs

### Web Browser

The web browser is what you browse the web with. It is the software that will interpret the code that you, as the web developer, coded. Most popular web browsers come with a set of “developer tools” to help you in your coding process. Below is pictured Chrome’s (top) and Firefox’s (bottom) dev tools. These tools often include the ability to edit the code of a page and see the results live, debuggers, consoles, and emulators for other devices (Specific Tablets, and Phones). Mozilla Firefox and Google Chrome are by far the most popular choices, bo­­th for general users and developers.



*Mozilla Firefox’s iPhone emulator (left) and possible emulation options (above)*

The emulator tool is useful for making sure a site functions and looks good on other devices.

### Git & GitHub

Git is used to manage multiple documents for a project and record the changes made. This process is called version control, or source control alternatively. It allows you to revert files to a previous state and compare the changes made to the files over time (Straub). Git is also useful in group projects with its ability to sync changes between collaborators and show who committed the changes (Neagoie).

GitHub is a host for what are called the git “repositories,” or the collection of a project’s files and the changelog and past versions of the files (Straub). The host is the server that you upload the changed files to and download the current project files from. A hosting server can be set up on any machine, but GitHub is by far the most used host, and allows for free hosting of public projects, and a paid option for private projects (Neagoie). When starting a site, using Git and GitHub is essential (even when working on a solo project) for being able to keep track of project files and monitor changes and revert potential incorrect edits to a file(s).

### Text / Source Code Editors

A Source code editor is a text editor designed specifically for editing code. Source code editors often have features that make programming easier, such as syntax highlighting, autocomplete for different programming languages, automatic coding-specific indentation and bracket matching, among others. These features are designed to make the coding process smoother and faster, and the code easier to read.

Common text editors include Atom, Sublime Text, Visual Studio Code, and Brackets. Choosing between different text editors really comes down to personal preference. Which one looks better to you? Which one has features that you would find the most useful? I ended up settling on Sublime Text because there were some themes that I thought looked visually pleasing and a package manager for downloading any additional features that I want in the future.

# Coding Languages

## The Big Three (Client-Side)

### HTML

Hyper-Text Markup Language is the baseline layer of the web, and present in all webpages. It is therefore necessary to learn HTML. It provides the structuring and meaning of the content. HTML is the “building block” of the Web (W3Schools). *Hypertext* means text that contains links to other content either within the same site, or inter-site (Mozilla Foundation). Links can go to other pages or to images or files (CSS Stylesheets, JavaScript files) on the same site or across sites. *Markup* refers to the tags given to elements on a webpage, that is the actual code, that tells a browser how the text should be displayed. An element tag in HTML is always within a less-than sign and a greater-than sign. Ex: <head>. Almost all (there are only a few exceptions) opening tags, like the head tag, need to have a closing tag. Closing tags are identical to their opening tag, except they have a slash “/” after the less than symbol. For example, the closing tag for the head element would be </head>. Everything that falls within this element inherits it. For instance, anything within the head tags would be considered part of the “head” of the document, and anything within two “body” tags would be part of the body of the document. Different HTML tags include: <link> which defines a link, <body>, which defines the body of the document, <b>, which defines a bold section, <em>, which declares emphasis (by default in the form of italic text). As you can see, HTML and its tags are typically self-explanatory. They are used on practically all websites as a base structure to simply define the different parts of the page.

### CSS

Cascading Style Sheets is the most common styling language in web development. Almost all projects use it. CSS describes the appearance and defines the design of the web page.

CSS can be in a separate “.css” file linked to through an HTML <link> tag or imbedded directly into the HTML document with the <style> tag. However, because of a developer fundamental often called ‘separation of concerns,’ it is considered a better practice to keep your CSS in a separate document, and not within the same document as your HTML code (Neagoie).

When coding CSS, an HTML element is first declared and the CSS properties to be applied to it are then declared within brackets. In example, say you wanted your paragraph text to be centered on the page and aqua it would look as follows:



Common CSS properties include font-size, font-style (what font the text is), background-image, and padding (is followed by a value and declares how much space is around an element).

There also exist CSS Animations, a more complex CSS module, which allow for the simple animation of web elements (Bytyqi), such as having an element “slide-in,” or “ease-in” from the side of the page.

Without CSS, there would be nothing to tell the browser how your HTML elements will be displayed, and so it is an essential part of every website.

### JavaScript

JavaScript, commonly abbreviated to JS, is considered one of the three core technologies of the web, along with HTML and CSS (W3Schools). It is a client-side language that is used to declare how webpages behave when certain events are triggered (Mozilla Foundation). E.g.; what happens when you click on a button, or hover over some element of the page. When a web page does something more than just displaying static information, it is often JavaScript that is powering it (Mozilla Foundation). Other features of JS include storing information inside variables and committing operations on strings (Codeacademy).

A common use of JavaScript is to run logic tests. A logic test in JavaScript can be, for example, checking if something is equal to another thing and completing an action on the page. JavaScript can also be used to do math with certain variables and make calculators (Codeacademy).

JavaScript follows a more complicated syntax then the other two base web languages, HTML and CSS, and so more time will need to be put into learning it then with the other two simpler languages. However, It’s complexity is worth it as with JavaScript webpages can become much more interactive and fluid.

## Server-Side

### Databases

Any sort of organized data or information can be considered a database. There are two types of databases, Relational and non-Relational (W3Schools).

A Relational Database consists of two or more tables with columns and/or rows. For example, you might have a table called Users. Within this table there is a column for ‘date created,’ ‘username,’ and ‘full-name.’ The values (what the column equals) are the rows (Neagoie).

A Non-Relational database, or a NoSQL Database, is a sort of unorganized database. A non-relational database is like a database of folders. It assembles the data within the specific folders, not in columns and rows like a relational database. The example previously used for a Relational Database can be used for a Non-Relational database, except this time each user will get a separate file, with their information in it. This contrasts the Relational schema of using one document with all the users, and their information in different rows.

Relational databases are much more common than non-relational for web applications. This is because relational databases allow for much fewer database files, granted the files are bigger.

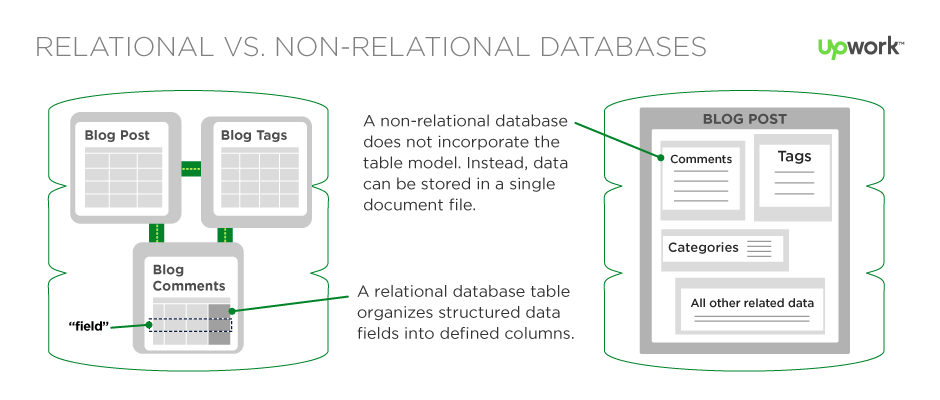


Figure [[1]](#endnote-1) - Differences Between Relational and Non-Relational Databases

The diagram above shows an example of the same application using different database types. The left one (relational) has three different tables that come together to form the app, while the right (non-relational) has all of the info stored in one database entry.

### Structured Query Language (SQL)

SQL is the standard language of web databases (ANSI). It is not the database system itself, but rather is the language that allows for interaction and communication with the database (Neagoie). Database Management Systems (DBMS) such as MySQL and PostgreSQL are used to manage (create, edit, and access) the databases. They use SQL to communicate with the database.

## Miscellaneous

### A Note on PHP

While it might still be a useful skill to have, learning PHP isn’t as important as it used to be. PHP is considered an older programming language in web development, and fewer and fewer web developers are using in PHP their projects (Neagoie).

# Design / Layout

## The Programs

This **Design / Layout** section will primarily be done using Adobe Illustrator Image result for adobe illustrator and Adobe Photoshop 

## Planning

### Making a Site Map

A site map is a flow chart of sorts that shows what pages go where on the website. They are often created in the earlier planning phases of making the website as a way to organize how the full website will be laid out ("Illustrator").

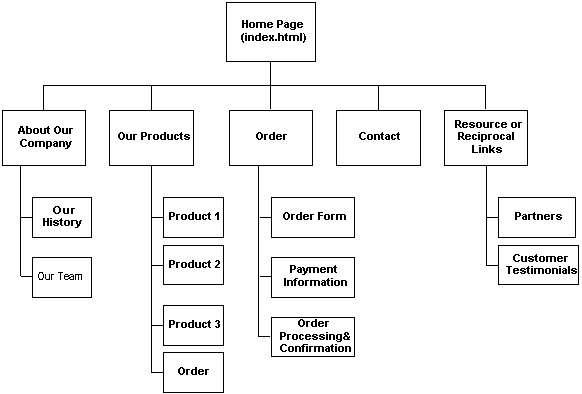


Figure [[2]](#endnote-2) – Website Site Map Example

You can make a sitemap in various Adobe Programs, such as Photoshop, Illustrator or Muse, or you can create a less formal sitemap by just drawing it out on paper ("Illustrator"). Drawing it will be much faster than any of the other options and can be perfectly fine if it is just for informal planning, but it shouldn’t be chosen if there is a need for a more formal site plan.

### https://file.mockplus.com/image/2017/11/e8bbb60f-f2da-49cf-9c3b-64dbefe77b27.pngWireframing

Figure iii – An Example of a completed Wireframe (this example was done in balsamiq mockups).

It’s hard to just jump into a project and start coding what’s in your head. Trying to do this will result in a much longer process and more revisions to your site and its code than necessary. This is where wireframing comes in. A wireframe is a mockup of the page built using placeholder graphics and text ("Illustrator"). It lets you get a basic idea of what the page will visually look like before you jump into coding these designs.

Similar to a sitemap, you can create a wireframe by hand on paper, or using a computer program like Illustrator. The computer created option will always be the neater and more formal of the two options, but that doesn’t mean hand drawing a wireframe is a bad idea. Using a computer program such as Adobe Illustrator or Photoshop allows you to pick the exact colors you want to use and exact sizes of the objects on the page, so when you go to code later in the process it will be much easier.

## Creating the Full Design

Once the wireframe is done, it is time to fully design the website. This can be done using Adobe Illustrator or Photoshop, among others. I will be focusing on the use of both Ps and Ai.

### Fonts

In any design field, you must remember that you cannot just use any font you want. Someone had to have made the font you are using and so it is their intellectual property. For this reason, it is important to make sure you have the right to use a font. [Google Fonts](http://fonts.google.com/) is a great tool for finding free for commercial use fonts. All the fonts that you find through Google Fonts are opensource and free for any use, commercial or not.

### Colors

Color is one of the most important parts of a website’s aesthetic. There should be a constant color scheme throughout the website to help consistency. A helpful tool for picking colors can be found for free at [color.adobe.com](https://color.adobe.com/create/color-wheel/) (Adobe Color CC). It allows you to pick good combinations of colors without knowing any color theory yourself. If you cannot create a color pallet that suits your project, you can click the ‘explore’ tab to find ones made by others that are free for you to use.

### Using Artboards

Artboards (depicted below) are an important part of Ai and Ps when it comes to web design. They allow for the creation of different views of the webpage, so you can design for PCs, Tablets, and iPhones ("Photoshop"). Web accessibility is very important. If your site is not accessible (does not work) for people consuming on an iPhone, you have just lost out on over fifty percent of the internet’s users.

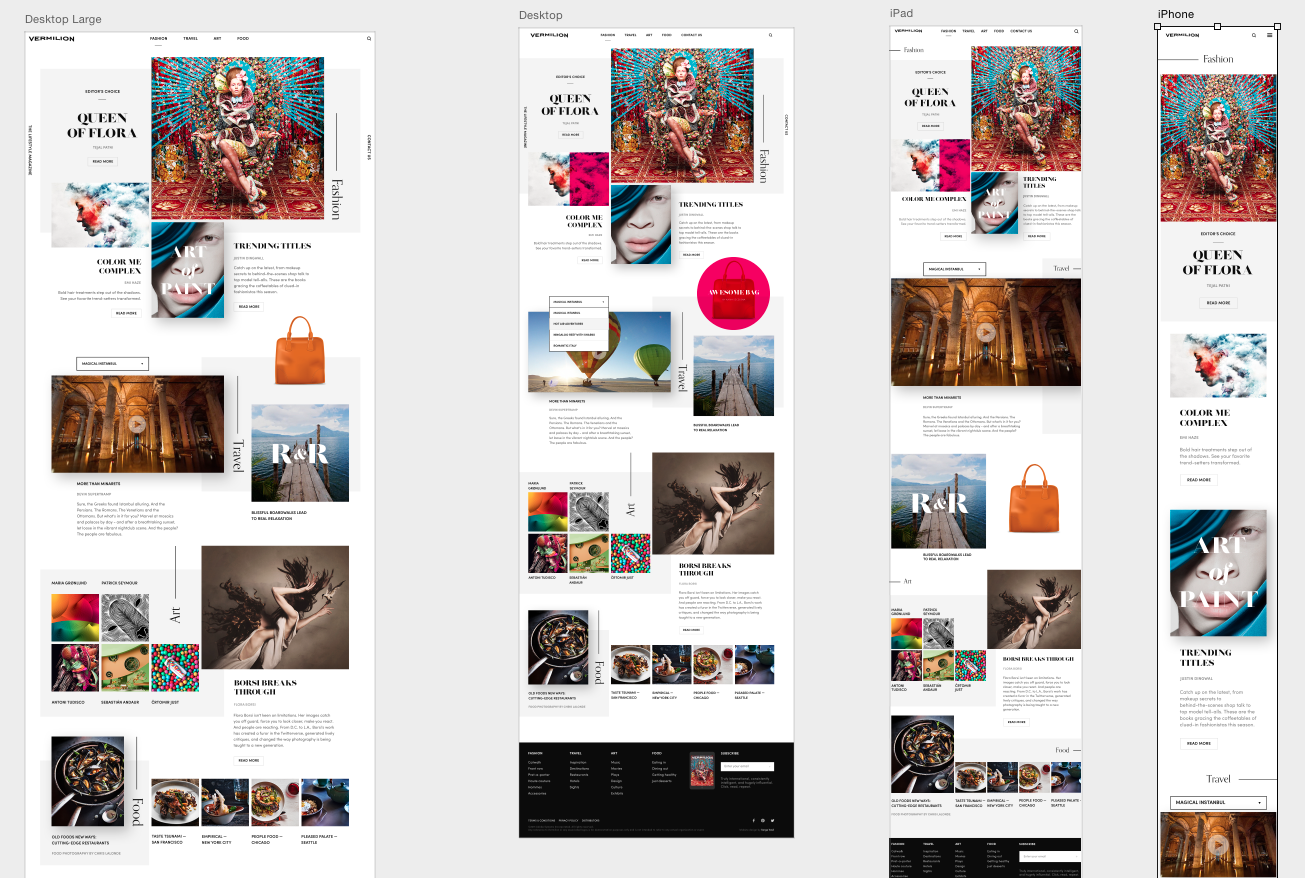


Figure iv – Artboard in Adobe Photoshop

Artboards also allow for the design of your whole site on one Ps or Ai document. Every page of the site can be on its own artboard. This allows for a much cleaner and easier design process, in which you do not need to have a separate file for every page of the site, which would get very messy, especially for bigger projects.

### Notes on Responsive Design

When designing a website, it is very important to keep in mind that the site should be responsive. This means it should be accessible to all different types of web browsing devices and look good on their screen sizes (Neagoie). As I touched on in the previous section on artboards, you can, *and should* use the artboard feature to design how the site will look on different devices. Even if a user on a PC shrinks their browser down, the site should still look nice, and the designer has to plan for that, starting even in the wireframing process.

Part 2: Project Plan

## My Essential Question:

How Can I Effectively Design and Code a Website That Serves a Practical Purpose?

## My Focus:

My essential question includes two parts – Design, and Development. I will be focusing the major part of my project on the development part and much less on the design. This is because I have the goal to complete the ‘web application’ part of the website, as this is the hardest part. This does then lead to having the design aspect fall a bit to the side. The reason I am doing this is because I have come to the realization that focusing on the design aspect and letting the development go potentially incomplete would not work out, as I would feel unfulfilled designing a website that is badly actualized in code. I’ve therefore settled that it would be much better to focus on the practical use of the site and let the design not be as quality.

# Foundational Learning Next Steps

The next steps I need to take in my foundational learning are:

* Fully Learning the Programming Languages
  + Learning SQL
  + Further JavaScript Learning
  + Advanced CSS
  + Potentially Learning React.js and the languages that go with it
* Learning More Web Design (design theory)

I plan to learn the bulk of this through the Udemy course I am taking (It’s supposed to be a 3-month course, and I’ve managed to cram through the majority of it in 3 weeks) and looking through the *Bring Your Own Laptop* design courses that I’ve subscribed to for the content I want to learn.

# The Product and Content of My Website

### The Idea

My product website will be a sort of “showcase” for other senior projects. The plan is to have a webform that other seniors will fill out. This form will contain fields that pertain to what the responder’s project is, and maybe add an option for them to submit an image of their project.

Ideally, the data from this form will be sent to an SQL database upon submission. This data will then be dynamically pulled from the database to complete a full project page for each responder. This will be done using a base template page with empty fields. Upon request the template’s empty fields will be filled with the data from the form.

I recognize that this is an ambitious goal for me to have, and that the only way to know if I can do it is to try. I do however have a backup plan in case I cannot successfully implement the templating feature. If the templating cannot be done, I can still manually create the pages for each responder, but by copying the form submissions, directly into the code. Note that this will have the same end result as the main plan but will lack a web application creation and the use of SQL database. The reason I want to go for the more complicated approach is because I would personally feel that I would have created a much more successful product if I can incorporate more complexity into the finished product.

### How This Will Demonstrate Learning

This product will be a complete synthesis of my learning, using all elements of my research. It will require the use of HTML, CSS, JavaScript, SQL, and potentially React.js, NPM, Node.js, JSX (I’ve yet to learn much about the latter four).

### Collaboration

An important part of being a web developer is being able to use the resources available when coding. Throughout my research, and especially in the “Zero to Mastery” course by Andrei Neagoie, the importance of Q&A, reference, and forum sites had been stressed. Using Sites such as Stack Overflow and The Mozilla Developer Network’s reference documents, to ask and find answers is an important skill for a developer.

# Timeline





# Bibliography

Bytyqi, Besart. Interview. Cole Manning. 24 October 2018.

Coyier, Chris. *CSS Tricks*. n.d. 28 November 2018. <www.css-tricks.com>.

cPanel, Inc. *cPanel Documentation*. n.d. <www.documentation.cpanel.net>.

Draplin, Aaron. *Draplin Design Co.: Pretty Much Everything*. Abrams Books, 2016.

Hoston, Dave. "8 Phases of the Web Design Proces." 28 July 2014. *Print Magazine.* November 2018.

Mozilla Foundation. *MDN Web Docs*. n.d. <www.developer.mozilla.org>.

Neagoie, Andrei. "The Complete Web Developer in 2019." n.d.

Refsnes Data. *W3Schools*. n.d. 1 December 2018. <www.w3schools.com>.

Ryzac, Inc. *Codeacademy*. n.d. 29 November 2018.

Scott, Daniel Walter. "Illustrator - UI & Web Design using Adobe Illustrator." n.d. *Bring Your Own Laptop.*

—. "Photoshop - How to Design a Website in Photoshop." n.d. *Bring Your Own Laptop.*

Straub, Scott. *Pro Git*. Apress, 2014.

Wathan, Adam and Schoger, Steve. *Refactoring UI*. 2018.

1. <https://content-static.upwork.com/blog/uploads/sites/3/2015/06/02170023/relational-vs-nonrelational-databases.png> [↑](#endnote-ref-1)
2. <https://nationalgriefawarenessday.com/wp-content/uploads/2018/01/website-storyboard-template-download-printable-website-storyboard-word-format-sample.jpg>

   iii <https://media.balsamiq.com/img/support/tutorials/responsivedesign/456717.png>

   iv <https://i.ytimg.com/vi/xtgHiVFik0U/maxresdefault.jpg> [↑](#endnote-ref-2)