Applying Navigability & Accessibility to a Self-Coded Digital Interactive Portfolio

A Master’s Thesis
Presented to
School of Arts & Sciences

In Partial Fulfillment of the Requirements for the Master of Science Degree

State University of New York Polytechnic Institute

By
Robert Zuch
December 2022
SUNY POLYTECHNIC INSTITUTE

DEPARTMENT OF INFORMATION DESIGN & TECHNOLOGY

Approved and recommended for acceptance as a thesis in partial fulfillment of the requirements for the degree of Master of Science in Information Design & Technology

December 12, 2022

DATE

Ana Jofre, Advisor

Ryan Lizardi, Second Reader
ABSTRACT

In this work, I have expanded upon a digital portfolio that I had created from scratch. This improved digital portfolio encompasses several methods of allowing for accessibility and easier navigation of the content within it. This was done through the creation of new features such as a landing page, prevalent ALT text, reduced motion toggles, buttons which filter portfolio content by categories, search functionality, tab key navigation, and more. Improving the visual presentation was also a focus, which is reflected through responsive layouts for mobile users, animated background graphics, and the revising of the portfolio’s visual hierarchy.

Despite the existence of website builders, I have come to the conclusion that self-coding your own website allows you to express yourself beyond the content you upload to it. With enough dedication and alacrity to learn the basics of coding within HTML, CSS, and JavaScript, developers can create their own designs that convey their unique personalities. My revised website can be found at https://zuchr.github.io/, and seeks to serve as an example of that dedication and expression.
# TABLE OF CONTENTS

ABSTRACT .................................................. 3

TABLE OF CONTENTS ................................. 4

LIST OF FIGURES ........................................ 5

1. INTRODUCTION ....................................... 6

2. REVIEW OF RELATED LITERATURE AND RESEARCH ................................. 8
   Categorization, Optimization, and Performance .................................. 9
   Visual Design .................................................................................. 12
   Accessibility Options ................................................................. 14

3. METHODS .................................................. 18

4. RESULTS & DISCUSSION ................................... 25

5. CONCLUSIONS AND FUTURE OUTLOOK ......................................... 32

REFERENCES .................................................. 34
LIST OF FIGURES

1: Projects shown are a lot more digestible when given ample room to breathe. .................19
2: Every entry in the main gallery is loaded in as plain text from a spreadsheet. ...............21
3: This info-box gets populated with spreadsheet information when an item is selected. ......21
4: The WAVE tool spots a contrast error at the top, after the tool itself caused it. ............26
5: Brand new landing page, before and after. .................................................................27
6: Separate page for the main gallery of content. ............................................................28
7: Separate page for expanded projects, with toggle-controlled animated gears. ..............28
8: Separate page for displaying my contact information and resume. .........................28
In the digital age, having a digital portfolio is important for highlighting your skills and expertise. This is well-reflected in modern job applications for creative fields, where it is frequently stated that applicants should provide links or URLs leading to their portfolios. As standing out from the crowd is common advice for job applications, the same philosophy should apply to the presentation of a portfolio. Where many website builders and templates fail in regard to in-depth personalization, the ability to create your own unique digital portfolio is the definitive answer to standing out to employers. But not every employer who visits your website has the same browsing experience as your own. It is important to create proper visual designs and navigation systems that can account for those with disabilities. This project serves as a case-study in self-developing a custom-made digital portfolio website which not only looks good, but is also accessible to as many users as possible.

This sets up the task of creating a visually appealing design and connecting it to systems that make it available and easy to use. In order to make a website that meets the requirements above, the pages must accommodate multiple screen sizes, perform well without placing strain on guests’ computers, and be usable to those who navigate using assistive technologies. Given that the end result will be in the form of a visual interface, the pages should be instantly readable, adhere to proven design principles, and follow an organized color scheme with good contrasting. Masahiro Sakurai described it best in his brief talk about balancing clarity and style. “Key design
points to consider include visual guidance, the size of display elements, window hierarchy, brightness and saturation, choosing an appropriate typeface, and more” (2022).

This case-study documents the redevelopment of my original portfolio website, which can be found at https://zuchr.github.io/oldportfolio/ (formerly hosted on the SUNY Polytechnic web hosting servers, at https://people.sunypoly.edu/~zuchr/). I have archived this version to demonstrate the key differences between both versions of my portfolio. How the new revision improves over the original will be discussed in great detail throughout this paper.
Recently, I embarked on a journey to create a digital portfolio entirely from scratch. I wanted to make it as a culmination of all the design skills I have picked up on my road to getting a Master’s degree. While I was able to accurately meet the vision of my initial design, I soon began to realize that my portfolio website could be more satisfying for users to navigate. With a few ideas and a list of scholarly sources at my fingertips, my main goal was to find answers to this simple yet daunting question: *How can I make my digital portfolio more accessible to viewers?*

Before I discuss the “how”, I need to address the “why”. If I was to build upon my existing digital portfolio, why would it be advantageous for me to do so? This thought had me revisit Heymann’s writeup wherein it was shown that there are many “benefits from the use of an e-portfolio for building on self-awareness through reflection and improving competencies” (Heymann et al., 2022, p. 386). This text has been a big motivator for myself personally, and for those who are seeking employment, I highly recommend seeking out the full version of this writeup for an analysis on digital portfolios and their link to employability and self-evaluation. Heymann’s quote was helpful the first time I heard it, when I was just starting to build my website. But since I have reached a state where I found myself re-evaluating what I have built, the quote carried more weight than it did before. I was confident in my stylish presentation going forward with my first draft, but after viewing what I made with a critical lens, I had started to notice issues that I had neglected to even think about. One such issue was the lack of proper support for those who navigate websites without a mouse. Another issue laid in the layout of my
website, wherein my projects seemed to be fighting for the viewers attention. There were a multitude of issues of which I had begun to notice, which I will expand upon in upcoming pages. Nevertheless, with the newfound ability to properly perceive these issues, and having the proper time to work on fixing those issues, I could make my website function better than ever before. Those themes of self-awareness, reflection, and improving competencies from Heymann’s text would continue to reappear as I continued to analyze and re-analyze what I have built.

Categorization, Optimization, and Performance

As I will discuss further into this paper, an important feature I aspired to add to my site was the ability to distinguish and divide projects into separate categories. I believed this feature could help to make my portfolio stand out to employers from different backgrounds, especially if one employer is interested in my graphic designs, versus an employer who may only be interested in my interactive works. Having less of my works displayed on the page at a given time is imperative to making sure that the viewers are not overwhelmed. As written in a thesis paper by N. B. Dao (2011), it is said that, “in a multi-project environment [...] recognizing commonalities and differences between various types of projects becomes evident to employ efficient management methods. Project categorization systems are utilized for this reason” (p. 1). My works span a lot of different fields (graphic design, 2D & 3D animation, web design, etc.), so I had to be careful to strike the right balance of selectable categories. Not only that, but I needed to consider what would be shown to the viewer when they would first arrive at my gallery of works. My leading thought was to create separate categorization options that would show all of my works versus a selection of featured works. As to not overload the viewer, the page will
default to displaying the featured works. “Too much freedom will increase the probability that important items are not considered and rendered invisible” (p. 19). By showing the select few works that I want the viewer to see when they first land on the page, the user gets a good taste of the type of varied content I can produce before they begin to set their own filters on the content they want to see. With the sound reasoning presented in Dao’s scholarly article, I proceeded with that plan.

In addition to preset categorizations, I wanted to give users the option to filter through my portfolio content directly by searching within the site. But something I considered was how to apply such functionality while not overloading people’s browsers. I had to figure this out as I went on, but some research into the area of users interacting with pages with complex code had been insightful. One of the articles I looked at by Selakovic & Pradel (2015) stated that “code that performs poorly can cause users to perceive an application as unresponsive, which may encourage them to instead use a competitor’s website” (p. 1). I certainly did not want to deter people who might be interested in my work, so optimization was a high priority for me. One of the first things I was doing when I was re-evaluating my website was looking at the code itself, and realizing that there were a multitude of ways that I could reformat the code to optimize it and condense it. The article itself was very interesting because it dives into the patterns of optimization that would get employed on pieces of Javascript coding. “Optimizations of JavaScript code deserve attention because thousands of JavaScript applications rely on responsive and energy-efficient computations” (p. 10). In a way, this article is a celebration of code optimization, which makes it a both warm and fascinating read.
I have thought about changing my existing Javascript libraries to something that can perform as best as it can. But as it turns out, D3, the one I am using, is very well-liked by many scholarly sources. This includes a thesis I found which states that “D3.js aims to provide reusable components to improve accessibility and efficiency. Moreover, additional JS libraries can be utilized to assure backwards compatibility with older browsers” (Guldamlasioglu, 2015, p. 24). This was perfect, since better code performance means my website is more accessible to those with lower hardware specs. I may want to investigate further into the additional libraries being talked about moving forward. It is worth noting that before I fully understood and integrated D3 into my website, I was using a similar Javascript library called “Papa Parse”. This library loaded my spreadsheet data in a very similar way to D3, but I abandoned it due to the fact that I was learning how to use D3 at the same time for a different project. Plus, the features that D3 supports may become of use to my site in the future.

I wanted to look closer into how I could not only refine my code, but get the most out of it as well. That is why, on top of taking a class on D3 this semester, I had been looking over Dewar’s classic guidebook, “Getting Started with D3”. This book shows how to best leverage D3 to perform advanced actions with processed data. Some methods written in this book go over how to use D3 in a way that is less intensive on user’s computers. One such method is through trimming and cleaning data before being processed by D3. “Performing cleaning or data analysis in the browser is not only a frustrating programming task, but can also make your visualization less responsive” (p. 4). While the book mainly has its sights set on data visualization, some of the documentation within the book is very useful to confirm that I am using the D3 library in the correct way.
Visual Design

Regardless of the data, the visual design is always going to be a focus for what I am working on. Lidwell’s book on the “Universal Principles of Design” serves as a great inspiration for making cohesive design choices, and has been for myself personally throughout the entirety of my time at college. The book does a great job of teaching how to make a design appealing, but also how to make it accessible as well. There are so many design principles being utilized everywhere around us, yet people do not need to recognize them or even understand them conceptually to think that something utilizing them looks appealing or functions without a hitch. A good direct example of that comes from the page on the Orientation Sensitivity design principle, where it reads, “the time displayed on a standard analog clock can be quickly interpreted because the numbers are positioned at 30-degree increments around the center. The 30-degree increment happens to correspond to the minimum recommended difference in line orientation required to be easily detectable” (Lidwell et al., 2012, p. 146). The book is a great starting point for understanding what makes designs work and when to use different design principles. It has a plethora of straight-to-the-point design-based information presented in a way that is very easy to understand, making it one of the most useful books on visual design, and I would stress its significance to anyone trying to make an appealing design. Admittedly, I own the condensed version of the book that can fit in a jacket pocket. Despite having shortened explanations, it contains 50 more design principles than the larger print, making “The Pocket Universal Principles of Design” act as a sort of Swiss army knife for designing and planning. Though my copy is not currently residing in any jacket of mine, I always carry the book around in the front of my backpack because it is just that useful to have if I am ever designing something
When I am away from home. It has been my go-to source for planning how I would make my new website features look and feel.

When I was re-assessing my visuals, I had been thinking about branching out to include more interesting colors. I was wondering if my palette of emerald green, pacific blue and muted grays led to an uninspiring combination. However, after taking another look at the “Designer’s Color Manual” by Fraser & Banks (2004), I realized that I was already at a very good spot with my color palette. If I threw more colors into the mix, it would make my palette more confusing and less memorable. And if I were to change my existing colors, I could be giving my website a completely different feeling than what I was aiming for with my existing setup. As reported by the book, my main palette involving a light green color promotes subconscious thoughts such as those of nature, luck, success, and—fittingly for my situation—employment. And as for the light blue in my palette, that color mainly speaks to themes of calmness, beauty, calculation, and introspection (p. 21). The book itself is full of enlightening information that speaks volumes about each color and what they can potentially represent. It is a vital read when deciding what colors work best for your project, as it discusses the distinct psychological effects colors have on people. According to Fraser & Banks, “all theories of color are in some sense theories of language, and how we ‘speak’, ‘hear’, or ‘read’ colors tells us a great deal about how we understand the world” (p. 19). With all of that in mind, thanks to the information from this book, I kept true to my original colors, which is what I still believe is the right decision.
In fact, my chosen colors work excellently with each other in that they contrast remarkably well from one-another. This is great not only for having a versatile color palette, but it also makes my content easy to digest. In a journal published in 2010 calling out the United Kingdom government’s websites, it specifically stated, “guidelines for contrast and luminosity exist to make it easier for people with disabilities to see content by better separation of foreground from background. One way this can be done is by the effective use of color contrast” (Kuzma, 2010, p. 144). Since I have such vibrant colors in my main palette that are easily differentiated from the cool grays of my secondary palette, my website is proven to be very easily readable for viewers with less-than-average eyesight. While this journal was written with the purpose of getting the United Kingdom’s government to fix their websites, it also serves as a great exemplar for how to make a website accessible, and the pitfalls to avoid when aiming to do so. The article also mentions that there are a multitude of tools available for testing and analyzing color contrast in relation to the registered standards and guidelines (p. 144).

Accessibility Options

In a paper by Petrie and co., the idea is put to the test that accounting for accessibility in websites hinders design excellence. According to the study funded by the Disability Rights Commission of Great Britain, when 51 disabled users tried to perform tasks across 100 different websites, it usually turned out to be a success. In fact, it turned out that “some of the most accessible websites had complex visual designs encompassing graphics, complex layouts, and photographs” (Petrie et al., 2004, p. 18). While this was relieving to hear, it is important to note that this study was conducted during the early days of the internet. The standard for “complex
visual designs” on a website was definitely not the same as it is today. But, given that sites such as early 2000’s-era Ebay are listed as the prominent websites used for testing, I had no worries about my comparatively very less complicated portfolio site. I did not plan on making text elements hard to read, as backed up by my research of appropriate color contrasting, as well as my personal preference for intuitive layout design—and I definitely do not have as much content cluttering the screen as early 2000’s-era Ebay! As long as a balance is kept between visual design and a website’s ease of use, anyone should be able to substantiate Petrie’s paper in the modern day. With accessibility as a major focus of my design revision, achieving a great level of usability among disabled users seemed more attainable after reading this.

A cornerstone of making a website accessible to as many people as possible is the inclusion of ‘ALT text’ to supplement images or other media for those who can not see them. The scholarly article “ALT text and basic accessibility” does wonders for describing how to efficiently supply ALT text in a descriptive and useful way. To make sure that people with disabilities could navigate my website, “a text equivalent [for non-text elements] must be provided and fulfil the same function for a disabled person as it does for a person without a disability. This ensures that users relying on assistive technology, such as screen readers, can access the same information as others” (McEwan & Weerts, 2007, p. 72). In an image-oriented project like mine, it can be difficult to tag each and every image with text that describes the image in detail. This is especially true, since a sizable amount of my time has been spent on describing accompanying information about my projects instead of simply describing what is visually right in front of the viewer. Either way, the text inspired me to craft short yet descriptive ALT text for each of the images in my portfolio. In contrast to how I wrote the informational
captions, I had to be careful to not get into lengthy explanations for the ALT text. This is because, as described by McEwan & Weerts, “ALT text should also be kept as accurate and succinct [as] possible. Unnecessarily long ALT text makes it more difficult for users of assistive technology to understand the website content” (p. 73). In addition, there are plenty of images employed in my website which are purely decorative. As I learned through the reading, people need to specify those as decorative elements within their code so that screen readers and website crawlers will not get confused about which elements are imperative to the content, and which elements are merely supplementary / background images. This scholarly article has been very eye-opening in regard to all things surrounding ALT text, housing a lot of useful information.

In fact, that very article encouraged me to investigate how my site could improve to even further accommodate disabled guests of my site. This investigation led me to an excerpt from Chris Northwood’s 2018 book, “The Full Stack Developer”, which recommended methods of making navigation more accessible for those who are more comfortable navigating the web with their keyboards than their mice. “A common interaction mechanism for moving through web pages is to use the tab key to access elements such as form fields and buttons. [... This can be] analogous to hovering over an element with a mouse, and it allows [assistive technology] tools to indicate where a user is on a page” (p. 215). The ability for users to navigate through webpages with the tab key is a standard feature across web browsers, and completely opens up the opportunity to browse any website for those who may not be able to navigate with a mouse. Interactive elements such as links and those listed in the quote above will automatically become available through tab-navigation, and as Northwood continued to discuss, plain elements should be specified as navigable by giving them a number value in a property referred to as a “tabindex”
(p. 215). Information like that is imperative to aspiring web developers, and this book is full of equally helpful accessibility tricks that every HTML coder should know. Northwood has reinforced and further validated related knowledge that I had picked up after conducting interrelated online searches on the subject of web navigation.

On that token, I can safely make the argument that the majority of my sources in developing this project that ultimately helped me the most have been non-academic in nature. Documentative websites and web coding Q&A discussion forums have helped me whenever I ran into a problem or a pressing concern while coding, which had been frequent. I wanted to take this short paragraph as an opportunity to thank these informational outlets, namely W3Schools, MDN Web Docs, and StackOverflow, despite not being professional journalistic sources.
3. METHODS

As I have touched on prior, the main question concerning my project was, "How can I make my portfolio more accessible?". On top of simply improving the look and feel of the website, there are various ways that I have improved the navigation, made the content more findable, and the experience more user-friendly. Below are some of the larger issues that I have addressed.

I have made my website more accessible to viewers through the grouping and categorization of my content. But I have separated my content through multiple steps. For the very first step, users are taken to a landing page where they are presented with multiple choices for viewing different types of content on my website. This page acts as a main menu of sorts. It opens up with the three options of viewing my main gallery of work, seeing more in-depth explanations of larger projects, and showing my resume.

The main gallery of content lends itself to further categorization. For instance, I have many projects listed that might be of interest to very different audiences. In the case that a user only wants to see my article designs instead of seeing my animations, I have coded buttons that allow that content to be filtered out, which you will see described in greater detail in later paragraphs. In addition, in order to be extra sure that the viewer will not be overwhelmed with a slew of content, I have also leveraged the idea of splitting up my content based on the year each piece was created. The next page contains a mockup figure I made to demonstrate how I planned to make the “All Content” listing look after modifying the code that generated the list before it.
Being that my website is a portfolio, the content itself is something that needs to be as findable and navigable as possible. With a plethora of projects to display, it is important that my website does not overload the users’ browsers. With these concerns in mind, I decided it was not illogical to make it so every single entry has its own section, but it would be wiser to have them share a single info-box and swap out its data when the thumbnail of each entry is clicked on.

Thus, the way my content is loaded into the portfolio revolves around the way I set up my portfolio content in a Google Sheets spreadsheet, which gets loaded into the website at runtime. Figure 2, as seen on the page after the next, shows a trimmed view of this spreadsheet, with several rows and columns omitted for visual clarity. The spreadsheet gets loaded into the
Javascript with D3.js and gets converted into an object array, which the webpage keeps stored in memory until the cache is emptied or reloaded. This array contains a thorough set of columns, labeled: "Title", "Info", "Difficulties", "Skills", "Software", "Date", "Client", "Category", "URL", "Type", "Thumb", "Permahash", and "Featured". The first large set of columns, from "Title" to "Client", features purely text-based information that gets shown to the user when they click on the entry for that particular project or work. The "Category" column contains the information which allows the user to easily filter between the different types of projects I have listed. The next paragraph expands on this functionality further. The "URL" column houses the hyperlink to the work itself to be displayed, while the "Type" column dictates how the webpage should process that hyperlink. If the type is an image, or "img", it will have an image generated with the URL as the source. If the type is instead an embedded webpage, or "iframe", the page will generate a frame with which to load the embedded page, once again using the URL next to it as the source. This system is very convenient, as it saves spreadsheet space and computer memory by not having to separate the different types into different columns. Instead, the code decides how to process it just by that simple type indicator. The "Thumb" column was implemented more recently, and uses a smaller condensed version of the work’s main content to be used as a clickable thumbnail image. I will get into how this affects performance in a later paragraph. The "Permahash" column sets how the address bar will point to the work of that specific row. This is for easier shareability, and helps to identify what the work is going to be from nothing more than a glance over some text. For example, the portfolio entry Key Drawn with Stipple Dot Patterns is identified with the simplified hash "stipple-key". Lastly, the "Featured" column is used to set if and where the work or project will show up within the listing for featured content. This field is either left blank or given a number to tell the code what order the content should be displayed in.
Figure 2: Every entry in the main gallery is loaded in as plain text from a spreadsheet. This example is trimmed for easier readability, and shows the core aspects of the data.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Title</td>
<td>Info</td>
<td>Date</td>
<td>Category</td>
<td>URL</td>
<td>Type</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>(WIP) CyberWizard Animation Preview</td>
<td>The TF2 YouTube</td>
<td>October 1, 2022</td>
<td>3D-Animation</td>
<td>../../frames/v_cybiframe</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Expanded Project: Faces in TIME</td>
<td>There's too much</td>
<td>September 25, 2022</td>
<td>Web-Designs</td>
<td>../../frames/bigprofile</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Tournament Player Listing</td>
<td>This is something</td>
<td>September 20, 2022</td>
<td>Web-Designs</td>
<td><a href="https://cdn.discordimg">https://cdn.discordimg</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Game Designer Portfolio Creation: Josh</td>
<td>Seeing the portfo...</td>
<td>June 26, 2022</td>
<td>Web-Designs</td>
<td><a href="https://cdn.discordimg">https://cdn.discordimg</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Videographer Portfolio Creation: Dan</td>
<td>After seeing my</td>
<td>May 25, 2022</td>
<td>Web-Designs</td>
<td><a href="https://cdn.discordimg">https://cdn.discordimg</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Expanded Project: Friday Night Voice</td>
<td>There's too much</td>
<td>April 13, 2022</td>
<td>Games+Demos</td>
<td>../../frames/bigprofile</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Expanded Project: The Factory Times</td>
<td>There's too much</td>
<td>April 13, 2022</td>
<td>Article-Designs</td>
<td><a href="https://cdn.discordimg">https://cdn.discordimg</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Sub-Way</td>
<td>I made all the art</td>
<td>April 3, 2021</td>
<td>Games+Demos</td>
<td><a href="https://static.wii37">https://static.wii37</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Font Exhibit Title: UD Shin Go NT</td>
<td>For class, had to</td>
<td>March 28, 2021</td>
<td>Article-Designs</td>
<td><a href="https://cdn.discordimg">https://cdn.discordimg</a></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 3: This info-box gets populated with spreadsheet information when an item is selected.
The largest new addition, on top of my method of loading content, is through the new content filtering system employed by my website. By attaching specific tags to my content, I have been able to reliably self-code the entire system of searching for specific phrases and categories on my portfolio, with shareable links being automatically generated through the URL in the address bar. Using the "Category" column of the spreadsheet, I have separated each of my projects into groups based on the nature of what type of content they are. Paired with that, I have created a set of clickable content filters based on each type of category listed in the spreadsheet. The filters make it so only the works which match the selected category show up on the page. Separate non-categorical filters listed above those are "All Content", "Featured Works", and a filter with an input field for a custom search. Every one of these filters can be seen back in Figure 1, and the functionality is quite simple.

When clicking on an entry in the main gallery, it generates a hashtag using the "Permahash" column of data from the spreadsheet and applies it to the address bar, letting the page know to load that content. But when using one of the content filters, it adds the search string to the address bar followed by a "?" to notate that the phrase succeeding the question mark is being used to filter data, and narrows down the available options as such. These can also be combined to form a URL that links to one piece of content while bringing related results to the user in the background. For example, "https://zuchr.github.io/main/#nycd-spec-ops?Graphics" will link directly to the NYCD Special Ops Emblem entry in my portfolio, while also showing nothing other than additional graphics I made when the user clicks off of the main work the user was linked to.
The flexibility lent by the system allows the "Search" filter to work just as easily as clicking on one of my premade filters. The key difference is that it searches each applicable column of the spreadsheet for the user-entered phrase instead of just the categories. The "Featured Works" filter functions in a similar fashion to the "Search" filter, only displaying works with a number in the "Featured" column in the spreadsheet, and organizing the order based on the numbers given. With all that, it is safe to say that the content filters do their job in creating a much more navigable experience for the user. And, as an additional navigation feature, I have added a function which makes the info-box’s "Skills" and "Software" sections trigger a search for the text clicked on within them to further simplify the act of searching.

But on top of front-end navigation, I knew I had to be extra careful that my website would not be too much for anyone's browser or computer to handle. To be sure that every user is guaranteed a fast and easy experience on my website, I needed to optimize my code. My first draft of the website worked fine, but looking back at what I had written, I found many unnecessary lines of code that would have an impact on the page's performance. This means that if someone's computer or connection was not powerful enough, my content would load slowly and inefficiently. There are two steps that I have taken to mitigate this. Step #1 was to reformat my code to be more concise, and only include what was necessary while retaining things like comments to give information about certain lines of code (having understandable code is just as important as having it work). Step #2 involved adding a new column to the spreadsheet which my portfolio loads its content from. This column contains links to smaller versions of the images that would be used for the thumbnail of the work. This means that the full-size images, which would take a while to process, would only load when the smaller version is clicked on. This
alone makes my website more efficient than it used to be, and I will continue to make small thumbnail images for future content going forward.

But this was just the Main Gallery page. As mentioned at the beginning of this section, there are two other big pages I had mapped out for the user to navigate to from the home page. The page that describes my more detailed projects is referred to as the Expanded Projects page, and takes the elements that previously dominated the space at the top of my old portfolio (see the first half of Figure 5) and places them into their own page. While the repositioning of these elements into their own page was simple, it was the act of figuring out my page hierarchies that took a lot of planning. I realized that those projects were not at the top of my old portfolio because they were my best work—it was because they had more of a story to tell than they physically could in the same gallery as the rest of the projects. At that point, I figured that it was best to separate the projects into different pages as such, while still prioritizing the content of the gallery as what viewers should gravitate towards on their first visit. Thus, the Main Gallery only contains projects which can be described by a single info-box, and the new Expanded Projects page gives dedicated sub-pages to each project listed. Those sub-pages are more traditionally crafted: vertical and hardcoded, not utilizing any form of spreadsheet data.

The last new page that users can navigate to from the landing page is the Resume/Contact page. This is as simple as it could be, consisting of a short section with my email address and my Linkedin URL, while the rest of the page is made up of a large picture of my resume and a PDF download link. With that, all pages meant for users to navigate to are completed. As it stands, any other pages on my website are iframes, only viewable within the Main Gallery info-box.
4. RESULTS & DISCUSSION

The website is now available to browse at https://zuchr.github.io and is hosted under the umbrella of GitHub Pages. With having the pages and projects properly separated and categorized, on top of adding the ability to traverse clickable objects with the tab key, my website is infinitely more navigable and accessible than it ever was before. The improved graphics and appended options menus contribute to this as well. I can proudly state that the end result greatly matches what I had initially set out to do.

To provide evidence that my accessibility changes meet the standards of what makes web pages accessible, I have screened my website through a tool called the WAVE Web Accessibility Evaluation Tool, found at https://wave.webaim.org/. While the tool can potentially trip over itself with how it moves elements around and judges them afterward (see next page), it still gives imperative assessments on the accessibility of individual elements within a page. It notes good features such as the inclusions of suitable ALT text, labels, and document-wide language specifications. The tool also gives alerts for redundant ALT texts and links, and elements that could potentially be inaccessible without proper handling. The main things that the tool warns its users to watch out for when present are broken links, elements with no content, missing ALT texts, and texts that do not contrast well with their backgrounds. (I should also note that, despite the tool not seeking this feature out, I have implemented buttons that toggle reduced motion on certain animated elements for users who may experience motion sickness.) In the upcoming figure, the WAVE tool has assessed elements of my Main Gallery page, finding zero errors aside from four contrast errors.
Figure 4: The WAVE tool spots a contrast error at the top, after the tool itself caused it.

Note: The tool adds iconography directly to page elements which can stretch them out as shown.

Two of those four errors are a result of its own code-injection into my site upon testing, and the other two are linked to background texts in the info-box that are purely decorative and hidden from screen readers. In previous WAVE testing dating earlier into the project, each of the filter buttons at the top all had significant alerts regarding their accessibility. This is because they were not actually buttons at the time, but instead they were generic clickable elements which could not be navigated to with the keyboard. It was with this tool that I realized how important it was to specify buttons with <button> tags, links with <a> tags, and so on—all because browsers have the built-in capability to automatically make items such as those navigable to users without
a mouse. The Expanded Projects pages give a similar result, all with zero errors after making extensive adjustments to the code of each page. But, in its testing of my landing page, the tool proudly reads, “Congratulations! No errors were detected!” This includes contrast errors, of which none are present on this page.

Below is a direct comparison of what my portfolio website looked like before and after this redevelopment project was completed. On the next page of this document, I have attached one screenshot for each of the webpages that are linked from the new landing page. The info-box seen on the Main Gallery and the old portfolio page has mostly remained the same, aside from new category text and embedded search links, so see Figure 3 for a reference.

<table>
<thead>
<tr>
<th>Before:</th>
<th>After:</th>
</tr>
</thead>
<tbody>
<tr>
<td>A confused landing page that does not know which projects to point to first, squeezing everything in with little regard for hierarchy. The resume is hidden in the “ABOUT” menu.</td>
<td>A very distinct and easy-to-understand landing page. The different content is sectioned off into separately linked pages with clear priorities based on sizing and positioning.</td>
</tr>
</tbody>
</table>

Figure 5: Brand new landing page, before and after.
Figure 6: Separate page for the main gallery of content.

Figure 7: Separate page for expanded projects, with toggle-controlled animated gears.

Figure 8: Separate page for displaying my contact information and resume.
I am especially proud of the results of this project, as it successfully sets my website apart from the plethora of sites that are edited off of existing templates. Before making this, I had tried my hand at creating my portfolio through popular website builders such as Wix, Squarespace, Adobe Portfolio, and more. But through lots of unsuccessful trials, none of them hit the mark.

In the case of Wix, the service gave me full control of element placements and robust visual design decisions, but lacked the responsive design capabilities needed for modern websites to look acceptable across multiple screen sizes. Any elements placed outside of a 600-pixel-wide area are not guaranteed to show up how you want them to across displays, which can make Wix frustrating to use for both the designer and the viewers. In my brief period of time with Squarespace, it would only generate long, vertical pages with little variety in layouts and a limiting choice of overarching visual styles. I had used Squarespace when I was designing a website (with less experience) for an organization, but most of my time was spent coding sections using an external sandbox, which I then embedded into the website instead. That ability to code things yourself would make Squarespace worthwhile if not for the paid subscription. Adobe Portfolio was tailor-made for showcasing portfolio content, but ultimately did not allow for enough expression through both the minimal captioning of content and the empty site design. I had previously made two portfolios with this service, and they both felt painfully simple and nearly identical despite each being made a year apart.

While site builders and templates are convenient for quickly creating pages, none are able to replicate the level of customization that you get from coding your website yourself. The big problem with that is coding an entire website manually can be a daunting task. But as I have
learned over the course of this project, self-coding and designing can be made easy with the endless amount of resources available for free online.

While the literature discussed previously proved very useful for determining how I should plot out the design of my website, it was through writing the code that I needed to find additional resources to guide my work. The Q&A forums at stackoverflow.com have been incredibly helpful for troubleshooting and demonstrating certain scenarios that can be acted out with code, especially regarding JavaScript. Anytime I have found myself stuck with a coding problem that I could not solve on my own, Stack Overflow has consistently had an answer for me. This goes hand-in-hand with w3schools.com, which has every type of documentation I needed to know about HTML and CSS. W3Schools simplifies the information in a way that is easily digestible, and gives multiple interactive examples for each unique element and attribute that is utilized in HTML and CSS.

However, even if you have the coding techniques mastered, it is important that your design choices do not interfere with the user’s abilities to understand and enjoy your website. In addition to the online resources above, I have found another resource that helped to further guide my philosophies regarding the presentation of my website. Very recently, Masahiro Sakurai, the peak inspirator behind my style of graphic design, has released his guide on finding a balance between clarity versus style on his YouTube channel at youtu.be/UjW_TTNtXEM (2022). Even though this guide is primarily directed toward video game design, it goes entirely hand-in-hand with web design, as they are both interactive experiences being traversed by the end user. Striking that balance is something that has been very important to me, and will continue to be as
I design new features beyond this submission and improve on those already discussed. But this resource, among others, should inspire people to make interesting designs that exude personality, without sacrificing that much-needed clarity.
The source code for my website is available to be viewed in the GitHub repository at https://github.com/zuchr/zuchr.github.io/. Though my developments for this project spanned several weeks, I still heavily advocate for the creation of a self-made portfolio. The degree of freedom that comes with coding your own website allows designers to push the boundaries of what a website can look like. I firmly believe that with enough patience and willingness to learn web coding, anyone can make a website for themself.

However, not everyone would want to put themselves through that much coding trouble. While I have mentioned that the templates I found did not meet my personal requirements for the portfolio that I wanted to develop, this does not mean that templates can not be of use to people. In fact, many people would find it more advantageous to leverage existing templates, as they might not have the time or experience needed to develop a website themselves. In developing this project, I have had several thoughts about reformatting this work into its own template, with my professors suggesting this idea to me as well. I would have to define clear directions and guidelines to the public for how to best use what I have coded, but most of the work required to make it a template would be through the simple removal of my own branding, and creating modifiable sample pages and a spreadsheet.

While this project was a large undertaking, it has proved helpful for my own personal use. Now that I have all of these new systems put in place, I can continue to add portfolio content
to my website without modifying the website’s underlying structure. In essence, my website is future-proofed in the way that my new accessibility features are applied so that no manual work is required to make new gallery entries accessible. If I do end up repackaging this website as a template as previously discussed, I would hope that other people will find my design to be just as useful to them as it has been to me.
References


Sakurai, M. [Masahiro Sakurai on Creating Games]. (2022, September 28). *Clarity vs. Style [UI]* [Video]. YouTube. [https://www.youtube.com/watch?v=UjW_TTNtXEM](https://www.youtube.com/watch?v=UjW_TTNtXEM)