Digital Health Humanities Toolkit: A Minimal Computing Reflection

Anastasia Hutnick

UCSF Archives and Special Collections Minimal Computing Internship

April 11, 2026

Introduction

For the past eight weeks, I have been working as a Minimal Computing Intern for the Digital Health Humanities Toolkit at University of California San Francisco (UCSF) under Dr. Sean Purcell. The Toolkit is the culmination of the Advancing Digital Health Humanities Institute (ADHHI), a two year program cultivating a wide range of Digital Health Humanities projects across the United States. My internship began with the presentation of these projects at the end of the Institute, followed by my main goal of designing and developing the front-end of the Toolkit - a static website publishing the lesson plans and white paper from the Institute. The website would be developed using Jekyll, a framework for static websites built off of the programming language Ruby. One benefit to using Jekyll is that the website could be hosted for free on the platform GitHub Pages, avoiding any server costs.

For context, minimal computing may be understood as a digital design philosophy in which the developers only use as much software as they need for a given project. Minimal computing developers create their projects with intention, whether by choice or necessity. Digital Humanists Roopika Risam and Alex Gil best summarized this philosophy with four questions: “1) "what do we need?"; 2) "what do we have"; 3) "what must we prioritize?"; and 4) "what are we willing to give up?"[1] Now, at the end of my time with the Toolkit, I wish to reflect on my experience through the lens of these questions.

What Do We Have?

Coming into the internship, I had fairly extensive experience working with Jekyll and its digital archive framework offshoot Wax. I had published my undergraduate thesis using Jekyll, and I had worked on a few different Wax digital archives, such as P.O. Box 34 – which published artwork and writing created by people incarcerated during COVID-19, Women in Bandura of North America – a proof of concept archive for programs, history, and sheet music of the traditional Ukrainian instrument, and The New York Bandura School archive – a work in progress that preserves performance and rehearsal audio. I was familiar with the languages that I would need – namely HTML, CSS, and Liquid – as well as deploying websites on GitHub Pages. What I did not have was experience with the Digital Health Humanities specifically.

My internship officially started with the ADHHI Symposium back in February, during which I listened in at the awardee round table reflections and monitored the Zoom stream during the presentations. I was impressed with the vast array of projects, from studying

---

[1] Risam, Roopika and Alex Gil. . 2022. "Introduction: The Questions of Minimal Computing," Digital Humanities Quarterly 16 (2). http://ezproxy.gc.cuny.edu/login?url=https://www.proquest.com/scholarly-journals/introduction-questions-minimal-computing/docview/2681375031/se-2.

experiences of Depersonalization Disorder discussed on Reddit to mapping leprosy cases in the early twentieth century, to examining the relationship between AIDS PSA posters and horror movies to creating an art installation exploring the Opioid Crisis.

I was also struck by the ethical discussions held during the round table, especially when they explored the tensions between using open source yet more labor-intensive software versus commercialized, corporate-owned yet more user-friendly software. As someone who likes programming and often has the bandwidth and desire to learn new software, I tend to favor the former. However, I also believe that Digital Humanities should be accessible to everyone, no matter one's ability to program, project budget, quality of internet access, or other potential limitations. From ADHHI, we not only had the lesson plans and the awardees' project experience as a foundation, but also the opportunity to build a resource that continues to bridge the gap between access and level of experience.

What Do We Need?

As part of the federal grant funding ADHHI, UCSF needed to run the symposium, publish a white paper reflecting on the symposium, and make the learning materials openly available by the end of June 2026. The Toolkit would satisfy the latter two requirements. Furthermore, creating a static website would allow for easier access in areas with slow internet connections due to its lower processing requirements, increasing the availability of the site.

The Toolkit would also need to be easy to hand off and maintain, given that the project is owned by UCSF, and that both my supervisor's and my time there would be finite. Creating a static website using Jekyll that would be hosted for free on GitHub Pages would require minimal upkeep, and adding the UCSF Library team as a collaborator / owner of the GitHub repository was effortless.

Since we would be publishing the website on GitHub Pages, we also needed to build our own Jekyll theme. A theme, required for all Jekyll sites published on GitHub Pages, is essentially a style template, containing basic page layouts and other design elements. Rather than try to make the website's front-end design overlay an existing theme, we would create our own. The theme itself would also be open source, adding another deliverable to the project – a digital learning template for anyone to use.

When designing the theme, I also had to work with the UCSF Style Guide to ensure that the Toolkit would fit the University's brand. This process involved using colors and fonts that were approved by the University, as well as including the general UCSF banner. While the banner was located only in the Toolkit's repository, I included the color scheme and font in with the theme.

The theme definitely provided a bit of a learning curve for me, as I previously would add the necessary elements on top of an existing Jekyll / Wax theme. Over the weeks, a pattern emerged where I would test out a new feature in the Toolkit website repository, then move the code for that feature to the theme's repository when I was happy with it. Of course, when I felt that I had enough code in the theme to officially deploy it and connect it to the Toolkit, much trial and error ensued. For anyone who has ever programmed before, you know the moment when you run through every emotion in rapid fire succession as you stumble your way through a new concept, running on adrenaline and answers that you can stitch together from various Stack Overflow posts. Finally though, the theme and the website connected, the website was up and running, and I could focus on refining the website.

What Must We Prioritize?

Accessibility was our top priority for the website. As soon as I had the theme working, I was running the pages through WAVE, or the web accessibility evaluation tool, to ensure that the website passed web content accessibility guidelines. We didn't just want anybody to be able to find the website – we wanted to remove any barriers we could that would prevent people from using the website. Compliance included ensuring that the website would work on any size screen, that the colors used provided enough contrast for easy reading, that the font size would scale up for easier reading, and that all images had alt text or appropriate labels. As I have learned in my Digital Pedagogy class this past semester, you never know who will be using your project, what accessibility features they may need, or why they need them. When websites are well-designed, everyone benefits.

When writing the code for the Toolkit theme, I always kept in mind that we wouldn't be the only people using this theme. I therefore wanted to make the theme as easy to use as possible, including for people with only a basic understanding of Jekyll. Though I love programming, I firmly believe that programming is not a required skill to be a member of the Digital Humanities community and create DH projects. While Jekyll will always require at least a small amount of programming, I could simplify the development process for theme users by writing easy to follow documentation and following the philosophy of DRY.

DRY is an old programming adage which stands for "Don't Repeat Yourself". For example, why write the same five lines of code ad infinitum in your project when you could write a function and save yourself four lines each time? Rather than ask a theme user to write a whole complex code block each time that they wanted to display an image with alt text and a caption, I would provide them with a function (or "include", in Jekyll) where they could fill in a few prompts and have the image pop up on screen with responsive sizing as well as a caption and alt text.

What Must We Give Up?

With the Toolkit designed to require minimal maintenance, the website also required a limited lifespan. The Toolkit only has a planned lifetime of about five years, after which the website will be taken down and archived. As such, the design for the website had to be simple – which mainly meant no complex JavaScript-fueled interactivity that could break with any given software update.

As a fledgling game video game developer, I was initially interested in adding an element of play to the lessons, by creating a quiz game or something similar to reinforce the material. With this restriction, I turned my attention to the lesson materials rather than the lesson plans on the website itself. For the Command Line lessons specifically, the user would have to download a zip file containing various folders and files for them to navigate. I couldn't place a traditional game on the website, but I could create a scavenger hunt of sorts with the files for students who wished to get some extra practice with the command line functions. I had grand plans of hiding folders and files that would tell a little bit of Digital Health Humanities history should the users find them.

Unfortunately, when considering what we must give up over the course of a project, we don't just need to consider our technical capabilities. We need to consider that we are in fact human. On top of this internship, I already had a full time job and was in school part time. For the bulk of the internship, I was able to juggle the three – until my cat was diagnosed with cancer and died in the same week. You can only do so much in eight weeks, and life unfortunately knocked me down to seven. Needless to say, what was going to be a detailed plan became a one page schematic.

Conclusion

As we discussed during the Symposium, Digital Humanities projects change during the best of times. DH, as I have learned over the course of my studies, is a slow practice. You don't churn out concrete deliverables in two week sprints like in the tech industry. You may have a deadline, imposed by a grant or the end of a semester or by someone's personal schedule, but DH first and foremost centers the people involved. You do what everyone involved can personally handle, what best suits the material you have, and what best benefits the community at the heart of the project.

I would like to thank Dr. Sean Purcell for his guidance and support throughout my time with the project. It has been a privilege.