Project: Book Interiors with LibreOffice Writer

By David Keefer

Table of Contents

Chapter 1 | Assignment: Gen AI First Impressions

Gen AI First Impressions

This week, we learned more about Gen AI and how it can be used in co-authoring documents. The growth of AI will cause technical communicators to integrate AI tools into their authoring such asn translating, creating documentation for code, or editing content for style purposes. Three important facts for technical writers using AI for co-authoring are that *AI should be used as a tool to help, authors should be aware of accuracy and bias issues, and technical writers should know how to effectively use AI in their work*.

AI Should be Used as a Tool

AI needs to be treated like *a tool in the workplace, and not a replacement*. Like in assignment last week, we used AI to brainstorm ideas for a research paper. We gave Microsoft Copilot or Google Gemini prompts regarding what the paper will be about. The **AI tools** gave us responses with multiple topics related to technical writing. Some topics were not the best, best some would have worked as a basic starting point to get ideas. Technical writers should use AI as a tool to brainstorm ideas. Using it to create drafts and document designs take away from human input. The content or document is not as "real" if created by AI. A human writer will make the final decisions on how a document meets the user's needs no matter how well an AI did. AI can never replace human writers and that is why it should be treated as a tool.

Issues of Accuracy and Bias

Responses from AI can *contain poor accuracy and bias*. Technical writers that do not look over the information produced by AI will suffer the consequences of a poor document. AI has the chance of giving false or outdated information. A technical writer that is generating ideas for product instructions need to carefully read over the responses. For example, if a writer is creating instructions for medication and use AI for help, they must make sure the content is correct with what they want the reader to know. AI may give **wrong information** about side effects and if that information gets published in instructions that are printed on medicine bottles, it could lead to serious consequences. Writers have to ensure they take proper editing steps to make sure that the content is true and authentic.

Technical Writers Knowing how to Effectively Use AI

AI can enhance the way technical writers work, but it will make writers *create new editing skills*. AI allows writers to produce content and finish tasks such as summarizing, formatting, and editing quickly. Using AI this way can create more time for writers to focus on task that may require more attention. Writers must then understand how to **effectively use AI**. They will need to learn how to create effective prompts, assess the AI response with detail, and find ways to integrate AI into their tasks. Technical writers will have to learn how to balance the use of AI and their own input when creating content and documents.

Final Thoughts

The use of AI in helping technical writers co-author documents is becoming more prevalent in the workplace. Technical writers must understand that AI should be used as a collaborative tool to assist them and detail reviews of everything AI produces is necessary.

Chapter 2 | Assignment: HTML and CSS First Impressions

HTML and CSS First Impressions

This week, we were learned more HTML and were introduced to CSS. CSS allows a user to style their content in HTML. Three takeaways for me about CSS are the *number of rules and their functions, the box model, and creating a design that is applied across multiple pages*.

Number of Rules in CSS

CSS has *over 800 rules for styling text*. Those rules can change colors, text font, borders, size and position of boxes, spacing between content, images, and animations. The possibilities are almost **endless** for how someone wants to style their content. CSS makes it a lot easier to create designs how you want it. There is a whole list of hex codes for colors if basic ones do not get the job done. W3 Schools offers a page on their website with the 140 colors that are supported on all browsers. In CSS, you can either use the name of the color or the hex code when applying a color. Animations can even be implemented without the need for JavaScript. Animations can create the extra touch of design and creativity an HTML could be missing. CSS styles can be enhanced by the styling of boxes and content.

The Box Model

The use of borders, boxes, and padding allow for *easy content organization*. The box model contains a margin, border, padding, and then the content inside. If you wanted two columns of text on the page, then you can create two boxes with specific margins, padding, and borders. Or if you wanted to stack content boxes you can create multiple ones and then organize them however you like. Boxes make it possible to **make content readable** and create a better experience. An example I can think of is of a usability project from my class last semester. We had to analyze how usable a website is and looked through how content is structured on the webpages and if they were designed well enough for a user to find their desired information. Boxes and organization mean everything when creating a website for a document using HTML and CSS. By using all the design options and effectively using boxes, CSS makes it possible to have consistent designs across multiple HTML files.

Creating Consistent Designs

CSS and HTML together can be used to *create consistent designs across multiple HTML files*. There should be an established colorway for the content that are aesthetically pleasing. This means no light text on a light background or use of heavy, solid colors. The same goes with using fonts. There should be **consistent font choice** throughout. Then there should be consistent layouts. The navigation bar design should be the same on every page, styles of headers and footers should stay similar, and spacing be consistent. It is best to keep fonts, layouts, colors, and navigation consistent because it makes it easier for a user to navigate.

Final Thoughts

CSS lets a designer have endless possibilities when designing content in HTML. The designs and organization of content need to be accessible, readable, and consistent for the audience.

Chapter 3 | Assignment: HTML First Impressions

HTML First Impressions

This week, we were introduced to HTML. It stands for Hyper Text Markup Language and is primarily used in web design. I have a lot of experience using HTML and **VS Code**, so I will write about my thoughts on VS Code. Three interesting takeaways for me about VS Code is it's *versatility, customization, and Git intergration*.

Versatility

VS Code has a *wide array of extensions one could possibly want*. This allows for users to be as versatile in their coding environment however they want. Users can install multiple programming languages, packages, developer tools, and AI coding assistant such as GitHub Copilot. Juptyer notebook can be installed for those who prefer to code Python in that environment and then install certain debuggers for Python. **Web developers** can instal HTML, CSS, and JavaScript to create and design websites. There is even R Studio to let a user process and manipulate data. There are so many possibilities to let users personalize their work and tasks, which makes VS Code more than just a normal editor.

Customization

Second, VS Code has features that allow a user to *customize their code visually through themes*. Themes can change the color of the background, keywords, tags and elements, variables, and functions. This **customization** allows for users to read and understand code easier. When I first started programming in a Python terminal, I hated looking at the white screen with the colors of variables bothering me. I found the themes setting and changed the background to a darker color and the other text to cooler colors. In future coding classes, one of the first things I did when using a new coding environment was to change the themes. In VS Code, I use a black background with white text, tags, elements, and attributes being blue and items such as hyperlinks being a red, orange color. The different colors help me customize my code so I can quickly spot different parts of my code quickly.

Git Integration

VS Code includes extensions to GitHub that allow for *collaboration inside the editor*. Users can have control over document edits and managing the versions. Repositories can be cloned in VS Code and worked on at the same time. Users can still created pull requests, review edits, and create issues. GitHub Copilot can also be used to help write code.

Final Thoughts

VS Code is a versatile code editor that can be used for almost any task. It's customization and many extensions allow a user to truly make VS Code their own personal environment. I am looking forward to keep learning about VS Code and using HTML.

Chapter 4 | HTML and CSS Project: A4.Info Design Portfolio Project

Project Description and Goals:

I chose option **B** for my final project because I wanted to continue to learn more about accessibility and information design but in a different context. I chose to investigate the usability of nine different *University of Wisconsin* school disability and accessibility service websites. I was inspired from the research article I used in the A3 project to do this research.

One of my goals for this project was to conduct my own web accessibility research. I have worked with co
llege students with disabilities and have experience seeing their struggles with web usability. When we started to learn about accessibility in class, I was motivated to further my knowledge in it because it is something I am interested in.

My second goal was to use different tools than I have previously used in other projects. In the
beginning of the course, I noted in one of the first discussion posts that I wanted to learn how to use different technology to further my information design skills. In projects A1 and A2, I used Canva and Excel but I was already familiar with them. In this project I decided to use **WAVE** (Web Accessibility Evaluation Tool), Excel, and RStudio.

What I Investigated:

I investigated nine different *University of Wisconsin* system disability and accessibility service websites. The list of websites and their links are below:

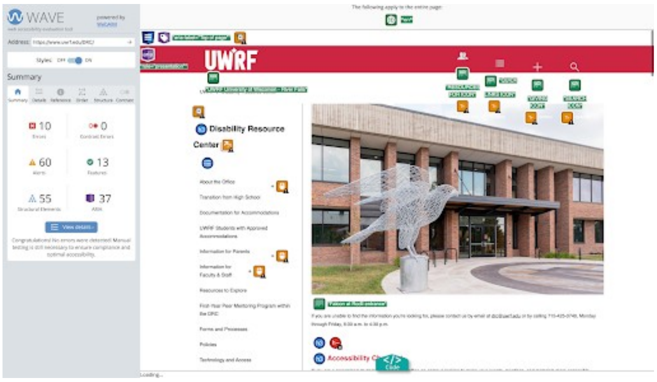
- UW-La Crosse Disability Resource Center
- UW-Milwaukee Accessibility Resource Center

- UW-Eau Claire Services for Students with Disabilities
- UW-Parkside Student Accessibility Services
- UW-Oshkosh Center for Accessibility and Disability Resources
- UW-Platteville Disability Access Center
- UW-River Falls Disability Resource Center
- UW-Green Bay Student Accessibility Services
- UW-Whitewater Center for Students with Disabilities

The *University of Wisconsin System* states that “Our commitment k
to accessibility reflects our belief that
nowledge should never be limited by how someone navigates the digital world”. I wanted to research how well the UW System websites follow through with this commitment. I found that three out of nine have no errors. All websites have many alerts that raise concern. To find my data, I used a tool called **WAVE** (Web Accessibility Evaluation Tool).

WAVE Tool:

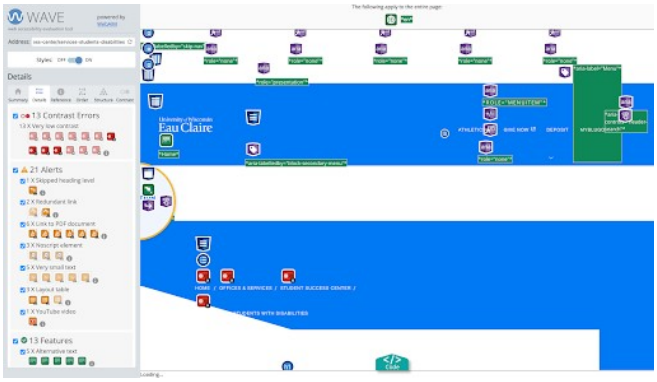
WAVE was created by the group *Web Accessibility in Mind (WebAIM)* at Utah State University. To use this tool, a user can simply paste a website URL into the search box.



From there, the tool will display a summary, breakdown of errors, alerts, and features, and the order
and structure of content. I started on the home page of each website's disability or accessibility service program and then to each subpage usually through a navigation bar. In the example screenshot above from the UW-River Falls site, I started on their Disability Resource Center home page and I went to each link on the left side and recorded data from each page.

How Data was Recorded:

To record the data from **WAVE**, I used **Excel**. I created a sheet for each website and then created a table that kept track of the type of error, contrast error, or alert and the number of times they appeared across each page. The Excel sheet is in the Google Drive folder linked on the first page of this document. Some alerts or errors were the same across multiple pages. One example is in the contrast errors on the UW-Eau Claire website.



This website contained 169 very low contrast errors. The white text displaying the webpage path was caught on every single page. Other websites had high noscript elements, redundant links, redundant title texts, and PDF document alerts. After I entered all of the data in, I used **RStudio** to create my graphs. Below are some definitions of alerts and errors that need more context before discussing the graphs. All definitions are from the **WAVE** website.

- **Noscript Element:** Content within noscript is presented if JavaScript is disabled. Because nearly all users (including users of screen readers and other assistive technologies) have JavaScript enabled, noscript cannot be used to provide an accessible version of inaccessible scripted content.
- **Redundant Title Text:** The title attribute value is used to provide advisory information. It typically appears when the users hovers the mouse over an element. The advisory information presented should not be identical to or very similar to the element text or alternative text.
- **Layout Table:** Layout tables exist merely to position content visually - to create columns, insert spacing, or align content neatly for sighted users. Their content is

not at all tabular in nature. Layout tables should not be used in HTML5. They can introduce reading and navigation order issues. Screen readers may interpret them as data tables (i.e., announcing column and row numbers), especially if they contain table header th cells. This introduces significant overhead on screen reader users.

- **Skipped Heading Level:** Headings provide document structure and facilitate keyboard navigation by users of assistive technology. These users may be confused or experience difficulty navigating when heading levels are skipped.
- **Orphaned Form Label:** An incorrectly associated label does not provide functionality or information about the form control to the user. It usually indicates a coding or other form labeling issues.
- **Possible Heading:** Heading elements provide important document structure, outlines, and navigation functionality to assistive technology users. If heading text is not a true heading, this information and functionality will not be available for that text.
- **Empty Heading:** Some users, especially keyboard and screen reader users, often navigate by heading elements. An empty heading will present no information and may introduce confusion.
- **Missing Form Label:** If a form control does not have a properly associated text label, the function or purpose of that form control may not be presented to screen reader users. Form labels also provide visible descriptions and larger clickable targets for form controls.