Slide 1: Title – Addressing Accessibility in WebPAC

The notes included in this document contains the spoken-portion of the presentation, “Addressing Accessibility in WebPAC”, that was given at the OH-IUG Regional Innovative Users Group Conference on September 20, 2019, in Columbus, Ohio. This document serves as a companion piece to the presentation.

Slide 2: Brief Background

Before diving into the project, I first want to give a brief background on my home university and the project.

Ohio Northern University is a small private university, with about 2,855 FTE as of Fall 2019. There are only two libraries on campus: Heterick Memorial Library, the undergraduate library, and Taggart Law Library, which services the College of Law. As the only Systems Librarian, I service both libraries. As you might also suspect, both libraries utilize Sierra.

The accessibility project originally began as a design revamp project for our library catalog, POLAR. We were looking into revamping POLAR because we learned back in June 2019 that our University’s website would be undergoing a redesign during the Winter Break. As our catalog’s design hadn’t been updated for a number of years, we thought that this would be a good opportunity to do so. While exploring the catalog to see what we might want to change, I opted to turn on an accessibility tool I had available. This tool highlighted a number of accessibility issues in the catalog that we had not known about previously, ranging from missing alt text to problems with linked text. Once these issues were discovered, the priority of the project changed from focusing on redesign, to addressing these issues.

Slide 3: Accessibility Tools Used

I started this project by first exploring guidelines on web accessibility, as it provided guidance on what the level of accessibility we needed to strive for was. The two guidelines I used were WCAG and LibA11y (pronounced “Lib-ally”). Both guidelines have been useful to the project as they have provided a deeper knowledge into making library content accessible. While LibA11y focuses specifically on library content, WCAG provided further background on accessibility as a whole. Additionally, as WCAG is the standard for web accessibility, it was important to use those guidelines as a foundation for the project.

For tools, I used Tota11y (pronounced “totally”) and Contrast Checker. I had learned about Tota11y at ACRL 2019 in Cleveland, Ohio, and had downloaded it following the conference. It’s a free browser extension that can detect various accessibility issues on a webpage, narrowing down the issue to the specific line of code, and offer a potential solution to the issue. Contrast Checker is a tool that allows you to see the color contrast ratios for any combination of colors. Further, it details whether those ratios meet WCAG AA and AAA standards.

I do also want to mention that while I have these resources listed, it’s important to note that these are not the only resources that exist, nor am I stating that these are the only resources you should ever use. These are just what I found and what worked for my purposes. That said I’m always looking for other resources, so if you know of any you’d like to share with me, please reach out to me!

Slide 4: Accessibility Project Workflow

My workflow for addressing the accessibility issues is pretty straightforward and is as follows.
First, I use Tota11y to find issues and impacted code on pages in the catalog. In cases where
the issue involves color contrasting on the page, I also utilize the Contrast Checker. Once I’ve
found an issue, I copy/paste the information from the tools into a Google Sheet. I also add in
additional info regarding the issue, such as recommended solutions, the solution used if it varies
from the recommended, and any comments that arose while fixing the issue. This helps with
tracking of issues, and provides a fairly concise document detailing the issues and solutions in
case we ever need to refer back in the future. Finally, I go into the WebPAC and change the
impacted code.

Because our catalog is built in the WebPAC, most of the accessibility issues I’ve found so far
are not unique to one another. For example, the search results pages all have the same issues,
regardless of what’s being searched for. So thankfully, there hasn’t been a case where I’ve had
to dig through and fix individual records one by one.

Slide 5: Screenshot 1 – Headings Error

In the following screenshots, I want to showcase some of the issues I’ve found in our catalog,
and how Tota11y looks when highlighting accessibility issues. This first screenshot is of our
catalog’s homepage.

Once activated, Tota11y brings up a list on the bottom left-corner of the screen of issues it can
detect. In this first image, I have selected the “Headings” option, which is highlighting our
catalog header. If it detects an issue, Tota11y will display a list of the errors on the bottom right-
corner of the screen. Selecting the dropdown arrow next to any error will provide information
regarding the error, as well as highlighting the relevant code.

In this case, the heading for our page is not using the proper Headings html tag. For some
assistive technology, not using the appropriate heading tag can confuse them, which in turn
provides the patron with incorrect or confusing information.

Slide 6: Screenshot 2 – Missing Alt Text

In this next screenshot of the catalog homepage, Tota11y is detecting that the image on the
page does not contain alt text. Without alt text, screen readers can’t detect the image and
translate the information to the user. While this image is purely decorative, it’s still important to
list it as such in our code, which Tota11y provides the necessary code to denote it as such.

Slide 7: Screenshot 3 – Color Contrast Issue

This next screenshot is of the results page after performing a search. Here, there’s a color
contrast issue with the box housing the OhioLINK catalog button. The error message details the
appropriate contrast ratio for the text based on its size on the page. While Tota11y does offer a
suggested color change, and depending on the page you can also preview that suggestion to
see how it’ll look, I would often use Contrast Checker to get an idea of what similar color options
were available. Since Tota11y’s suggestion is based on the current colors used, Contrast
Checker provided a wider range of acceptable options to use.

Slide 8: Screenshot 4 – Linked Text Issue

Still on the results page, another issue detected is with the linked text on the page. Depending
on how certain linked text is written, it can confuse screen readers, which could prevent the user
from accessing the link. The issue being detected here in particular is a little more complex in
our case, as it’s detecting linked text where there is no text or where a checkbox exists, and involves further investigation into why its flagging them.

Slide 9: Screenshot 5 – Missing Labels

This final screenshot is again of the results page, and the last issue I want to showcase are missing labels. Labels here are referring to attributes that can be given to specific lines of code as a way of identifying it on the webpage. This can be something that labels search boxes, which is currently one of the aspects of the results page being flagged by Total11y. It’s also flagging the material type buttons listed next to each entry in the results page, as they are not being detected as images by the tool, but as another piece of code.

Slide 10: Project Hurdles

While I was able to get the project started, I unfortunately haven’t had much time to continue it, as some hurdles have since come up. The biggest and most recent of which, is that our University’s website updated September 9, instead of over Winter Break. As the Systems Librarian, I’ve had to deal with how this change has impacted our outward facing website, and haven’t had as much time as I’d like to continue work on the catalog.

Another hurdle is that, while I have some familiarity with Sierra, there are still parts that I’m learning. One of which is the WebPAC, and learning where exactly each part of the code is stored, and how each piece plays with one another. And as I mentioned earlier, the other hurdle is finding time in between my other job responsibilities to work on the project. In addition to managing our library systems and website, I’m also the Repository Administrator for our institutional repository, I serve as a liaison to four academic departments on campus, and I teach information literacy classes as needed.

As we’re still pretty early on in the semester, I’m hoping that we’ll slow down enough to finally start the project back up.

Slide 11: Future Goals

Once I’m able to start the project back up, I’d like to do a deeper dive into accessibility testing. I’ve been talking with our University’s Disability Coordinator regarding other accessibility issues at the University, and I’d like to try rolling this into those discussions.

I’d also like to work on a broader accessibility project for all library content, with this as one of the centerpieces. That project would not only involve accessibility testing and fixing any issues that arise, but would also entail the creation of an annual routine for checking accessibility, and developing internal guidelines for ensuring the accessibility of future library content.

And, once we’ve established that the catalog has been made accessible, I’d like to continue the original project of redesigning the catalog.

Slide 12: Questions

If you have any questions regarding the project or this presentation, please feel free to email me at c-deems@onu.edu, or call my office at 419-772-2183. Thank you!