Creating a Legally Accessible Programming Course: An Experience Report

Ms. Nadia Simone Jean Morrow, University of Florida

Hello, I am Nadia Morrow, a PhD Student at the University of Florida. Currently, my research interest is in working in illuminating and increasing accessible practices happening in higher education computing courses. My pursuit of doctorate is to promote more accessible course experiences in all types of higher education institutions by helping develop technology and techniques that are made for departments' culture and practices.

Jean Louis, University of Florida

Jean is currently pursuing PhD in Computer Science from the University of Florida. Jean also holds a master's in computer science and a bachelor's in computer engineering from the University of Florida. Jean is an NSF Graduate Research Fellow. His research has involved the Internet of Things, Machine learning and IoT education. His current focus is on the impact of running machine learning algorithms on low-resource devices. His work lies at the intersection of embedded devices and machine learning.

Dr. Juan E Gilbert, University of Florida

Juan E. Gilbert is the Andrew Banks Family Preeminence Endowed Professor & Chairman of the Computer & Information Science & Engineering Department in the Herbert Wertheim College of Engineering at the University of Florida where he leads the Human-Experie

Creating a Legally Accessible Programming Course: An Experience Report

Abstract

Due to inaccessibility, disabled students routinely face barriers that require them to do "invisible" work to get to the same starting point as non-disabled students. One of the most persistent barriers for the last decade is inaccessible learning materials. However, in 2024, congress updated the Americans with Disabilities Act (ADA) Title II to adopt the Web Content Accessibility Guidelines (WCAG) 2.1 AA standard. For professors, this update meant that whatever files are uploaded on a learning management system (LMS) must meet these guidelines by April 2026 or April 2027, depending on city population size. These files include videos, audio, PDFs, documents, slide decks, and images, all essential components of a course. While many instructors have offered accommodations before, the new law obliges them to retroactively fit their course materials to the legal standard, regardless of a student asking for this accommodation. Retrofitting presents a well-known accessibility challenge and requires significant time to complete. While the legal standard will help to address some persistent barriers experienced by disabled students, it will not eliminate all. Updating a course should therefore begin with the legal benchmarks and extend beyond them. This experience report offers an in-depth look at the practical work of bringing a course up to the legal standard. The authors kept time diaries that logged the tools used, the effort spent, the costs incurred, and the challenges faced. In total, the author who did the adjustments took about 18 hours over the course of 3 months to complete this endeavor. The authors also describe the obstacles encountered with tools or processes and the adjustments made to achieve compliance in effort to help other instructors do the same for their courses.

1 Introduction

For over a decade, computer science education researchers have documented and attempted to address the inaccessible barriers that disabled students encounter in their courses [1]. One of the most common barriers documented were inaccessible materials [2], [3], [4], [5]. In particular, PDFs are infamous for being a consistent accessibility barrier for disabled students [3]. Due to these barriers, disabled students at all levels of higher education have been forced to engage in invisible work and accept lower grades when unable to make the materials suitable accessible [3], [6], [7], [8]. With the rise of more advanced generative AI and new accessibility legal standards, the decades long issue that disabled students have encountered with inaccessible materials may see progress made nation-wide.

In 2024, Congress updated the Americans with Disabilities Act Title II to incorporate federal accessibility guidelines for the internet for the first time [9]. This update also includes materials uploaded by instructors to learning management systems (LMS), such as Canvas or Blackboard. This new law contrast greatly with the previous legal guidance of "reasonable accommodations" which required instructors to only make materials accessible if a disabled student requested it after going through the disability services process. This process required disabled students to be enrolled, registered with disability services, disclose their disability to their instructor and request for this accommodation, meaning many instructors may never had to make accessible course materials [10]. With this new update, instructors, regardless of if disabled students are enrolled in their course, must make their materials adhere to the Web Content Accessibility Guidelines (WCAG) 2.1 AA standard [11]. The WCAG standard was developed based on the following principles: perceivable, operable, understandable, and robust (POUR) [12]. These principles are meant to ensure that users of online content can perceive, operate, and reliably understand content, with the use of assistive technology if needed, regardless of their abilities. The current WCAG standard, as of writing, is 2.2 with 3.0 in the works so the legal standard is out of date [13]. While the legal WCAG standard will increase the accessibility of the materials, this standard does not guarantee that disabled students will not need further accommodations based on their personal needs.

Due to this new law, instructors will have to engage in retrofitting. Retrofitting is a well-known accessibility challenge and requires significant time to complete [14]. This challenge is compounded by some instructors not knowing what tools to use, how to check the materials, what materials are included, and what is required to make the materials legally accessible. To help minimize this challenge, this experience report offers readers a detailed and documented process that the authors took to make their materials legally accessible. The following sections provide the average time taken to make materials legally accessible, the tools investigated and used, common issues encountered, and tips on how to do specific actions to achieve the legal accessibility standard.

1.1 What materials are included under this updated law?

At the start of this project, the authors realized that there is no clear guidance about which materials need to meet the accessibility standard. After investigation, the authors found that this law applies to any materials that instructors use or upload into LMS systems, including those that the instructor did not create. When thinking about this in practice, the instructor would need to inspect their course LMS page, anything that they uploaded, such as additional readings, explanation videos, articles for reading assignments, and instruction documents for how to complete a task, would all be included in the instructor's responsibility to meet the legal requirements. This requirement is regardless of whether the instructor has a student asking for this accommodation or not.

1.2 Methods

Despite this being an experience report, the authors wanted to provide as comprehensive and accurate information as possible. To achieve this, the author who updated the material kept time diaries that contained the issues addressed, tools used or investigated, and time taken. This task was completed over three months during the school year. Before updating the material, an accessibility computer science education researcher also went through the WCAG 2.1 guidelines to pull out relevant guidelines and checked what they found that applies to course materials to

what LLMs such as Claude, Gemini and ChatGPT produced. This task was to investigate the reliability of these LLMs' responses to see if they were reliable communicators of the standards. The researcher found that all three did a pretty good job at communicating the standards, but did not identify all the guidelines and, therefore, could not be totally relied on for this purpose. This process took about 1 hour and 2 minutes.

1.2.1 Class outline

This experience report is based on the class materials from the course presented in Louis et al. [15]. This course was developed for a 16-week semester, with 14 topics covered. This course included working with cyber-physical systems and allowed students to code in traditional IDEs and design in Tinkercad. Assignments were typically drafted in Word but given in pdf form. PowerPoint was the preferred slide deck development software. For the purpose of this report, the authors focused on three main materials that needed to be adjusted the most for the course, Word documents, PowerPoints, and PDFs. For a more detailed breakdown of the curricular structure and teaching methodology, please refer to Louis et al. [15].

Table 1: Common Accessibility Fixes For Materials

COMMON ADJUSTMENTS	IMPORTANCE	MATERIAL DONE IN
Alternative Text (Alt text) for Images	Provides a non-visual based description to communicate the message of the image.	PDFs Documents Slide Decks
Alt Text for Links	Provides a quick description of where the link will take the user.	PDFs Documents Slide Decks
Adjusting Reading Order	Allows screen readers/text to voice to read the text in the proper order.	PDFs Slide Decks
Hidden Titles	Provides a title for the screen readers to use to help navigate without having to have it be visually shown.	Slide Decks
Adding Headings Style	Provides the structure to the page so that the document will have proper tags associated with the title and headings. It also creates bookmarks so that the document is easier to navigate.	Documents

2 Materials adjusted

Over the course of three months, one author went through the course materials and adjusted all the materials created by the instructor. This author ended up taking approximately 18 hours to finish this process. Table 1 shows the most common adjustments that had to be made across the three materials and how they improve the accessibility of the materials. In the following sections, the authors explain what AI tools were investigated and used, and how they completed the adjustments using the most efficient tools available to them.

Table 2: PDF Time Taken

PDF DETAILS	TIME TAKEN	ADJUSTMENTS MADE
Night Light Assignment - 2 images and 2 links	2hr 18 min	Autotaggedx3, embed font, figure and link alt text, figure not in bound boxes, reading order fixed
ESP32 Setup Assignment - 10 images and 9 links	6hrs	Autotagged, image and link alt text, reading order fixed
Analog vs Digital Read & Write Assignment - 7 links	4 min	Set tab order to page structure
IOT Methodology Report Example - 10 images	43 min	Autotagged, adjusted tags, added title, added language, fixed tags on table, alt text to images
Night Light Assignment - 2 images, 3 links	2 min	Set tab order to page structure, added title
Arduino Cloud - 3 images	2 min	Set tab order to page structure, added title
Analog vs Digital Read & Write - 8 links	2 min	Set tab order to page structure, added title
Azure IoT Assignment - 16 links	2 min	Set tab order to page structure, added title
Communication Assignment 1 – 1 image,	2 min	Set tab order to page structure, added title
ESP32 – Firebase – 9 links, 2 images	2 min	Set tab order to page structure, added title
Literature Review – 9 links	2 min	Set tab order to page structure, added title
NRF24 – 7 links, 5 images	2 min	Set tab order to page structure, added title
Paper Reading Assignment – 9 links	2 min	Set tab order to page structure, added title
Project Final Presentation	2 min	Set tab order to page structure, added title
Project Midpoint	2 min	Set tab order to page structure, added title
Web Socket – 8 links, 6 images	2 min	Set tab order to page structure, added title

2.1 PDFs

PDFs were the most time consuming, issue-ridden, and frustrating material to make accessible. PDFs proved to be either the most intensive or the easiest material to make accessible. The difference laid in how accessible the source document was. Unlike other materials, which have built in accessibility features, if a PDF is not created from an accessible source, the process of making it accessible requires paid software access. For PDFs, an organization will need to invest money to allow for instructors to make WCAG compliant PDFs. While there are free LLM-based applications that claim to achieve legally compliant pdfs, they are currently not a one stop shop. Since the LLM-based applications do not allow for user intervention, either through

prompt engineering or manual override, users cannot fix the mistakes that are made in the system. This means that users must use a paid software to make adjustments. There is limited software available to make the adjustments. The most well-known is Adobe Acrobat. My organization has only invested in Adobe Acrobat, so all the following experiences are based on using this software. The author found that Adobe Acrobat's accessibility checker is insufficient to ensure that your pdfs are legally complaint, so axesCheck's PAC, which is based on the WCAG 2.1 AA guidelines and PDF/UA standards, was used. Despite this application being somewhat robust, several issues do need to be manually checked, such as if the alternative (alt) text is sufficient and if items should be marked for decoration or read aloud. Due to the wide array of support websites to help navigate issues, the author was able to successfully meet the legal guidelines. The support websites proved to be very necessary since the author encountered several glitch issues that Adobe's accessibility software experienced. Even with these support networks, the average time it took to make a PDF, from a non-complaint source, acceptable was over 30 minutes. Most of the time was spent navigating the glitches and finding the right solution to them. For PDFs from a complaint source, it took an average of 2 minutes. For a detailed list of the number of PDFs corrected, time taken for each, and adjustments made, refer to table 2.

Table 3: Common Adobe Glitches

COMMON GLITCH	REMEDIATION STEPS
Won't let you retag with the reading order tool	1. Try to retag the section as a header then use correct tag 2. Go to the tag list section, find the current tag for the section, add a new tag that is the correct tag for the section in the same nest, and move the section that was under the previous tag under the new tag. 3. Delete the current image structure and retag with correct label
Font Not Embedded	Go to preflight, click the wrench button next to the eyeglass button, go to the font section, click on embedded fonts and click fix at the bottom
Figure Not in Bounded Box	Delete the current image structure and retag as figure

2.1.1 Author's PDF remediation process

If the pdf was exported from an accessible source, the author only had a few adjustments to make to meet the guidelines. First, the author went to the "Prepare for Accessibility" tool in the tools section and push the "Check for Accessibility" button. Typically, the author saw, at most, three issues: no title, language not set, and tab order needs to be fixed. Within the report generated from the accessibility check, the author found that these issues could be addressed by right-clicking each of the issues and choosing the fix option. This either automatically fix it for the author or brought up a pop-up that will allowed the author to change the title or add the language. If the pdf was exported from a non-accessible source, the author, typically, encountered a variety of issues. However typically, the author had to add alternative text to images and links, check/adjust the reading order, and have the pdf tagged. The author followed

the following guidance to ensure that the alternative text was sufficient: alt text must describe the image or link by what the creator or user aimed for the reader to notice, not just a simple description. If the PDF was not tagged, the author found that there was no reason to run the accessibility checker first because typically there were too many issues to comprehend. The author found that running the autotag first allowed for less manual intervention and less time investment to get the PDF updated. To get to the autotag tool, the author went to the "Prepare for Accessibility" tool and clicked autotag. After the autotag happened, the author would go to the tag order list in the options available on the menu to the right. In this tab, the author went through the tag tree and ensured that the tags were properly nested. The author found that the nesting logic was similar to indenting lines while coding python. Some tags that the author consistently checked were H1, which represents Heading 1, existed if H2 (heading 2) was used and that H2 was nested under H1. The author followed the same logic for all the subsequent heading numbers (3, 4, 5, etc). Next, the author wanted to ensure that each image had alt text. The author would find the image tag and right click it to check for alt text. Within the image tag right-click menu, the author would locate the image properties option at the bottom of the menu. Then, they would select that option and add alt text in the text box under "alternative text for images". After this, the author would do a similar process for adding alt text for webpage links. First, find the link tags and right-click them. Then, locate the link properties button in the menu at the bottom. Click this option and add alternative text to the second box under "alternative text for links". Finally, the author would typically ensure that the reading order would follow the way that the document would be read by a sighted person. To adjust the reading order, the author went to the reading order menu on the right. For any item that was presented in the wrong order/spot, they clicked and held to move it up or down accordingly. There are many other issues that the author encountered while completing the remediation process and some where caused by the Adobe tool glitching. In Table 3, there is a list of the common issues that happened while the author was adjusting PDFs due to glitching and the methods the author took to try to address them before having to start over and retag the whole document.

Table 4: Word Document Time Taken

DOCUMENT DETAILS	TIME TAKEN	ADJUSTMENTS MADE
Syllabus - 26 links	32 mins	Added heading styles, changed hyperlink color, alt text for links
Webchat AI Integration - 1 link	10 min	Added heading styles, changed font, added title
Wi-Fi Setup - 2 links	3 mins	Updated headers, alt text for links, added title
Final Project: Midpoint Check	3 mins	Updated headers, added title, font updated
Arduino Cloud - 3 images, 8 links	12 min	Alt text for links and images, added title, font update
Analog vs Digital Read & Write - 7 links	3 min	Alt text for links Added heading styles
Paper Reading Assignment - 9 links	3 min	Alt text for links

DOCUMENT DETAILS	TIME TAKEN	ADJUSTMENTS MADE
Paper Reading Assignment Template	30 sec	No issues
Night Light Assignment - 2 images, 3 links	6 min	Alt text for links and images
Literature Review - 9 links	3 min	Alt text for links, Added heading style
NRF24 - 5 images, 7 links	8 min	Alt text for links and images
Communication Protocols Assignment - 1 image	1 min	Marked image as decorative, Added heading styles
Signal Strength Assignment - 6 images, 4 links	3 min	Alt text for links and images, Added heading styles
Web Sockets Assignment Part 1 - 2 images, 7 links	3 min	Alt text for links and images, Added heading styles
Web Sockets Assignment Part 2 - 4 images, 11 links	7 min	Alt text for links and images
Web Sockets Assignment Part 2 Copy - 6 images, 8 links	10 min	Alt text for links and images
MQTT Web Dashboard EC Assignment - 6 images, 14 links	15 min	Alt text for links and images, Font color contrast adjustment
ESP32 Firebase Outline - 2 images, 6 links	6 min	Alt text for links and images, Added headings style
ESP32 Firebase Assignment - 2 images, 9 links	2 min	Alt text for links and images
Azure IoT Exercise - 16 links	3 min	Alt text for links
Final Project: Midpoint Check	30 sec	No issues
Final Project: Presentation 2025	1 min	Alt text for link, Added heading styles
Final Project: Presentation - 1 link	2 min	Alt text to link, Added heading styles

2.2 Word document

The author found that Word documents were the most consistent and monotonous material to remediate**.** For the 22 documents, the most common adjustments that needed to be made were applying the styles to the headers and adding alt text to links and images. On average, the documents took six minutes to make legally compliant. Occasionally, the author would have to adjust colors to ensure distinction and adjust fonts to ensure readability. The following is what the author did to make the headings WCAG approved: went to the styles section on the home ribbon, based on whether the headings should be 1st, 2nd, or 3rd adjusted to the appropriate style, changing the font color and size if needed. The styles available in all Word documents create the equivalent of tags in html, which allows screen readers to provide the same reading flow as a visual reader would have. It also allows for the creation of bookmarks in the navigation pane,

allowing users to more easily navigate the document. The author found that an easy way to know whether the colors would meet the contrast requirements of the WCAG guidelines would be to toggle the high contrast button at the top of the drop down for font color, which is found in the font section of the home ribbon. While WCAG size guidelines do not apply to word documents, due to the ability to zoom within the application, the font was still increased to have a baseline size to 16, to allow for a greater variety of color availability. Based on the guidelines, some colors would meet sufficient contrast if the size of the text was larger. The author found that this could be checked using the WebAIM Contrast Checker.

2.2.1 Author's process for remediating Word documents

Word documents were comparatively more straightforward to adjust than PDFs and slides. First, the authors ensured that the headings had the appropriate styles were applied. The author made sure to apply the style and then change the font or size afterward which ensured that the metadata had proper labels to identify the words as headings. Next, the author would go to the review tab, and click check accessibility. The accessibility checker for Word works well and was reliably used to identify all but two issues, the document having a title and color contrast. The author found that when exporting a Word document to PDF, the document title would not correlate to the document metadata having a title. To give the document a title in the metadata, the author went to the file menu, clicked the info option and added a title. To ensure proper color contrast, the author went to the home tab, click the down arrow on font color and toggle on the "High-contrast only" button. This will only showed colors that met the contrast standard for the guidelines. Other things that the author found that may need to be address but weren't needed for this course are:

- If you make graphs, or tables, ensure that color is not the only way readers can differentiate important information, another visual sign should be used. For example, in a bar graph, the bars can be differentiated by both color and having stripes or dots in the box.
- Adding proper titles to tables and graphs which requires going to the references tab, clicking add caption and choosing the appropriate label.
- If space between paragraphs or tables is needed, do not use a new line. Instead go to the home tab, paragraph section and click the line spacing down arrow. There space can be added before or after the paragraph and adjusted.

Table 5: PowerPoint Time Taken

POWERPOINT DETAILS	TIME TAKEN	ADJUSTMENTS MADE
Introduction – 23 slides	33 min	Alt text for links and images, hidden titles, adjusted reading order
Sensors and Actuators – 40 slides	21 min	Mark images as decorative, adjusted reading order, added slide titles
Circuit Introduction – 10 slides	7 min	Alt text to images, adjusted reading order
Circuits and Schematics – 21 slides	39 min	Alt text for links and images, adjusted reading order, add hidden title

POWERPOINT DETAILS	TIME TAKEN	ADJUSTMENTS MADE
Health Sensors – 7 slides	6 min	Alt text for image, hidden title
IoT Methodology – 2 slides	10 sec	No issues
Project Management and Software Development – 3 slides	40 sec	Hidden title
Research – 27 slides	10 min	Alt text for links and images
Research Design – 10 slides	20 min	Alt text for links and images, hidden titles, adjusted font color contrast
Sustainable Development – 17 slides	23 min	Alt text for images, hidden title
Intro to MCU – 32 slides	21 min	Alt text for links and images, hidden title, checked for captions on video, adjusted reading order
RF Communication – 30 slides	50 min	Alt text for links and images, hidden title, adjusted reading order
Networking – 23 slides	14 min	Alt text for images, added heading style, adjusted reading order
Privacy and Security – 29 slides	30 min	Alt text for links and images, hidden titles, checked for captions on videos
Privacy and Protection – 19 slides	24 min	Alt text for images, hidden titles, checked for caption on video
Man in the Middle – 12 slides	19 min	Alt text for images, adjusted reading order
Web Development 1 – 11 slides	1 min	Marked images as decorative, adjusted reading order
Backend Development – 27 slides	28 min	Alt text for images, adjusted reading order, adjusted font color contrast
Architect and Design – 24 slides	47 min	Alt text for images, adjusted reading order, hidden titles
Cloud Computing Fundamentals – 15 slides	16 min	Alt text for links and images
Intro to Machine Learning and AI – 35 slides	27 min	Alt text for links and images, hidden titles, adjusted reading order
Machine Learning Pt. 2 – 30 slides	24 min	Alt text for links and images
Modern Machine Learning – 30 slides	36 min	Alt text for links and images

2.3 PowerPoint

PowerPoints were also consistent and monotonous but took longer due to the amount of content. On average, PowerPoints took about 33 minutes to remediate. A good indicator of if a

PowerPoint would take longer depended on the number of important images that were in it. The more important images, the more alt text required. For any images that were not important, the author would just mark them as decorative, meaning there is no reason for the screen readers to read this image. Similarly to Word, PowerPoint included the ability to make the necessary adjustments without the need for a third party accessibility checker. The most common adjustments for PowerPoint were adding hidden titles, adding alt text to links and images, and adjusting the reading order. In Table 5, the breakdown of the number of slides in a PowerPoint, time taken to remediate and adjustments made are listed, in order of completion. This allows the readers to see the progression of skill acquirement over time taken.

2.3.1 Author's process for remediating PowerPoints

To adjust PowerPoints, the author, first, went to the review tab and clicked check accessibility. The accessibility checker does point out all but one issue, color contrast, sufficiently. To address the color contrast, the author went to the home tab, font section and click the down arrow on font color. Then, the author toggled on the high contrast button and check the font color on each slide to make sure was selected from the remaining colors shown. Other than the color, the accessibility checker found the issues and the author clicked each on. By clicking on the issue, the checker would take the author through each instance and recommend solutions. These solutions do address the accessibility gaps for the guidelines. So, the author would complete whatever needed to be done manually based on the provided solution.

2.4 Other materials

Other materials that the authors checked or thought about, but at a lower rate or were a part of another material, were videos and audios. For both videos and audios, the guidelines require transcripts to be created from the dialogue. Videos also require captioning to be available. For videos that are taken from YouTube, both transcript and captioning are already available, so the only requirement for the instructor is to double check that they are there and accurate. For audio, there are several machine learning algorithms online that can produce an editable transcript that the user can adjust as needed. While this course did not upload lectures on the LMS, if there were lectures, the authors would have to make both a transcript and captioning available before uploading. The authors found that AI systems that were specified for this service provided a good starting point and could reduce time taken to complete the task. Despite not being in an organization that heavily uses the Google suite, the author did take the time to investigate the native accessibility features found in Google Docs and Slides accessibility features and how accessible their pdf versions are. Google Docs does allow for most of the accessibility fixes to be fixed within their system. The only issues that could not be fixed natively were adding alternative text for links, checking color contrast and marking images as decorative. When exporting the PDF, Google docs retains the accessibility features done with native features. Google Docs does not have an accessibility checker natively so a third party checker is needed. Similarly, Google Slides does not have a native accessibility checker, ability to add alternative text for links, indicator of color contrast, or ability to mark images as decorative. This is a more prominent issue due to the aesthetic nature of slide decks.

3 Advice

Based on this experience, the authors would like to give some advice to readers about how to navigate this process themselves. First, the authors found that it was not helpful to do PDF accessibility first. The process of retrofitting PDFs is arduous and frustrating making it easy to

become demotivated early on. When making PDFs meet the legal requirements, the author noted that the process mirrors coding. By this the author mean that the accessibility checker may only point out one issue, but when you find a way to fix that issue, another one will appear. Due to this issue, the author found it important to allow for a copious amount of time to complete PDF remediation and the need to focus on one issue at a time. Also, the author found it helpful to use the Adobe Acrobat accessibility checker as the first hurtle and fix all the issues found in it before going to the outside checker. If the author did not follow that process, they found that PDF remediation could be overwhelming. Some accessibility requirements require the human eye due to nuance because the algorithms may pick up on something that is not relevant. This meant at times not all issues found were actual issues. The author had to make sure to investigate what issues were found for relevancy. Second, if your organization relies on google products or you prefer google products knowing how to navigate Microsoft would be helpful. Google products do not have as extensive native support as Microsoft and require more work skill to learn. Third, the author found that updating the source Word document to meet accessibility before exporting it to a PDF was much easier and faster than fixing the originally inaccessible PDF. Unless the original version took over 2 hours to find, the author found that they spent less time doing this method than trying to fix the uncompliant pdf. Fourth, the authors recognize that this is tedious work, so they encourage the reader to ensure that they are comfortable and able to spend some time doing this process. The authors emphasize that this process takes time and needs dedicated attention to meet the legal standards. Fifth, this is a reminder that materials not created by instructors but uploaded to the LMS also need to be made legally compliant. Sixth, due to time constraints, the authors were unable to provide time diaries for remediating outside resources uploaded by the instructor to the LMS system, so this gap must be factored into time estimates. Also, this process was done by one author for a singular course so the time estimates may change if there is more than one remediator or more courses to prepare. Seventh, the authors recommend that code be copy and pasted into documents to ensure that the instructor has full control over adjusting for accessibility. If a screenshot of the code is used (which the authors do not recommend), it is important to ensure that the IDE has color contrast turned on so that the text color in the image will meet the color contrast guidelines. Also, remember that the screenshot will need to have alt text where the code will need to be written out. These are some of the lessons that the authors have learned during their time updating their materials. The authors know that this isn't comprehensive and had several limitations, but believe that this experience report can help other instructors and government workers prepare for this new guideline and ensure that the students or the public get the information they need accessibly.

4 Conclusion

Retrofitting past materials to meet the WCAG guidelines will probably not be an easy or fast experience by any means. It is a tedious and at times frustrating experience. As of right now, LLMs are little help in speeding up this process. The biggest issue is that the prompting may take as much time as writing it out by oneself would. The machine learning algorithms incorporated in software as accessibility checkers were able to detect accessibility issues in our course but were not all encompassing and therefore could not be relied on solely. However, with this guide, readers get guidance on what to check for outside of those algorithms to ensure legal compliance. Readers also get a clearer understanding of the time it may take to do this for one course of their own. This process encourages greater critical thinking about how to develop materials, the accessibility of outside materials, and greater intentionality with the use of images. After going through this process, the authors gained a new workflow for materials they will create for any

future endeavour and a more expansive understanding of how the current systems we use should improve to help people more easily meet the legally standards. The authors hope that this paper gives readers a starting point to complete this endeavour on their own. They also want to emphasize that doing these changes to meet the legal requirements does not mean that disabled students will not need further accommodation. The legal guidelines are outdated and incomplete, but incorporating these standards will raise the accessibility of materials for all students.

References

[1] N. S. J. Morrow, A. Irving, J. McKenzie, A. Smith, and J. E. Gilbert, "The Invisible Participants: How Computing Education Research Fails Students with Disabilities," in *Proceedings of the 2024 on RESPECT Annual Conference*, Newark NJ USA: ACM, Jul. 2025, pp. 115–121. doi: 10.1145/3704637.3734739.

[2] V. Borsotti, A. Begel, and P. Bjørn, "Neurodiversity and the Accessible University: Exploring Organizational Barriers, Access Labor and Opportunities for Change," *Proc. ACM Hum.-Comput. Interact.*, vol. 8, no. CSCW1, pp. 1–27, Apr. 2024, doi: 10.1145/3641011.

[3] K. Mack, "Accessible Communication and Materials in Higher Education," in *Proceedings of the 24th International ACM SIGACCESS Conference on Computers and Accessibility*, Athens Greece: ACM, Oct. 2022, pp. 1–6. doi: 10.1145/3517428.3550408.

[4] J. Salmi and A. D'Addio, "Policies for achieving inclusion in higher education," *Policy Rev. High. Educ.*, vol. 5, no. 1, pp. 47–72, Jan. 2021, doi: 10.1080/23322969.2020.1835529.

[5] K. Shinohara, M. McQuaid, and N. Jacobo, "Access Differential and Inequitable Access: Inaccessibility for Doctoral Students in Computing," in *The 22nd International ACM SIGACCESS Conference on Computers and Accessibility*, Virtual Event Greece: ACM, Oct. 2020, pp. 1–12. doi: 10.1145/3373625.3416989.

[6] M. Borrego, A. Chasen, H. Chapman Tripp, E. Landgren, and E. Koolman, "A scoping review on U.S. undergraduate students with disabilities in STEM courses and STEM majors," *Int. J. STEM Educ.*, vol. 12, no. 1, p. 2, Jan. 2025, doi: 10.1186/s40594-024-00522-2.

[7] D. Jain, V. Potluri, and A. Sharif, "Navigating Graduate School with a Disability," in *Proceedings of the 22nd International ACM SIGACCESS Conference on Computers and Accessibility*, Virtual Event Greece: ACM, Oct. 2020, pp. 1–11. doi: 10.1145/3373625.3416986.

[8] K. Shinohara, M. McQuaid, and N. Jacobo, "The Burden of Survival: How Doctoral Students in Computing Bridge the Chasm of Inaccessibility," in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, Yokohama Japan: ACM, May 2021, pp. 1–13. doi: 10.1145/3411764.3445277.

[9] "Fact Sheet: New Rule on the Accessibility of Web Content and Mobile Apps Provided by State and Local Governments," *U.S. Department of Justice Civil Rights Division*, 2024.

[10] "ADA & Post Secondary Education Title 2." Kentucky Govenment. [Online]. Available: https://extranet.personnel.ky.gov/OER%20Documents/ADAPostsecondaryEducation.pdf

[11] "Web Content Accessibility Guidelines (WCAG) 2.1," *W3C Recommendation*, May 06, 2025. [Online]. Available: w3.org/TR/WCAG21

[12] "Introduction to Understanding WCAG 2.2," *Web Accessibility Initiative WAI*, Jul. 15, 2025.

[13] "W3C Accessibility Guidelines (WCAG) 3.0," *W3C Working Draft*, Sep. 04, 2025. [Online]. Available: w3.org/TR/wcag-3.0

[14] A. Foley and B. A. Ferri, "Technology for people, not disabilities: ensuring access and inclusion," *J. Res. Spec. Educ. Needs*, vol. 12, no. 4, pp. 192–200, Oct. 2012, doi: 10.1111/j.1471-3802.2011.01230.x.

[15] J. Louis, N. Morrow, and J. Gilbert, "From Circuits to Cloud: An Experience Report of a Full Stack IoT Course Curriculum," in *2025 ASEE Annual Conference & Exposition Proceedings*, Montreal, Quebec, Canada: ASEE Conferences, Jun. 2025, p. 56604. doi: 10.18260/1-2--56604.