Universal Design 2024: Shaping a Sustainable, Equitable and Resilient Future for All
K.S. Fuglerud et al. (Eds.)

doi:10.3233/SHTI241041

Exploring the Use of AI for Enhanced Accessibility Testing of Web Solutions

Kristin Skeide FUGLERUD[a,1], Till HALBACH [a], Ingrid UTSETH[a], and Anders Ueland WALDELAND [a]
[a] *Norsk Regnesentral (Oslo, Norway)*
ORCiD IDs: Fuglerud: 0000-0002-5648-0264, Halbach: 0000-0002-9566-7013, Utseth: 0009-0003-6617-4302, Waldeland: 0000-0003-2795-3867

Abstract. Artificial Intelligence (AI) holds significant potential for enhancing accessibility and user experience across digital products and services. However, mainstream web solutions commonly used by the general population still face accessibility barriers, hindering equal participation in the information society for people with disabilities. This article explores several promising applications of AI that can be used to create accessible solutions for people with disabilities. We also present our research, which aims to explore and demonstrate how various AI-based techniques can enhance and streamline accessibility testing for web solutions. We selected four success criteria from the Web Content Accessibility Guidelines (WCAG) that currently require extensive manual work and developed four prototypes using open-source machine-learning models to enhance conformance testing. While the prototypes need further optimization and evaluation, the results suggest that AI-based techniques can significantly reduce the need for manual work in accessibility testing.

Keywords. Accessibility, WCAG success criteria, AI, computer vision (CV), machine learning (ML), natural language processing (NLP), prototypes

1. Introduction

Artificial intelligence (AI) is increasingly recognized for its potential to enhance the accessibility and usability of digital services for a diverse range of users. For example, AI is expected to offer new opportunities that can contribute to a more inclusive work life [1,2]. Moreover, it is pointed out that AI has enormous potential to improve accessibility and user experience for people with disabilities in a wide range of digital products and services [3]. One illustration of this is how automatic speech recognition can be used to create subtitles for video content, and for real-time transcription of conversations and translation into other languages. This can be useful for people with impaired hearing, people with intellectual challenges, and for non-native speakers. Applications that use speech recognition, such as voice control and dictation solutions can provide new opportunities for people with mobility impairments and writing challenges. Image recognition and object recognition can be used to create image descriptions and object descriptions that can increase accessibility for people with impaired vision. AI-generated summaries of digital content can help break down long texts into more manageable parts and can be a benefit for people with reading challenges

[1] Corresponding author: Kristin Skeide Fuglerud, *kristins@nr.no*.

and cognitive challenges. Facial recognition and other biometrics can make authentication more accessible for people who have difficulties using passwords.

In addition to the solutions mentioned above, which can increase accessibility and quality of use in general technology for broad user groups, research is ongoing into how AI can enhance and create new types of aids for specific user groups [3]. For example, AI is being utilized in systems for recognizing gestures and sign language, and in speech and communication solutions for individuals with speech and communication difficulties [4]. Additionally, researchers are exploring how AI can generate easy-to-read versions of texts [5]. Moreover, investigations are underway into various navigation solutions for people with visual impairments and those with poor orientation skills.

However, despite existing legislation[2], individuals with disabilities still encounter barriers when using web-based solutions. To promote the inclusion of people with disabilities in the information society, it is crucial that mainstream solutions prioritize accessibility. In our research we have investigated how AI can enhance and streamline accessibility testing for web solutions, aiming to develop effective tools that can detect and potentially address accessibility issues, benefiting both legal bodies and developers.

In the next section we examine related research on the use of AI to increase accessibility of software and web-solutions for people with disabilities. Subsequently, we present our work with prototypes to explore the use of AI in detecting accessibility issues in web solutions.

2. Related work

AI is utilized in various ways to enhance accessibility in the development of software and web solutions. For example, programmers can use AI to generate code that creates more accessible user interfaces, although the suggestions provided by AI are not always reliable [6]. AI is also employed in tools known as overlays, which are placed between an application and the user for automatic improvements of an application's accessibility [7]. Advancing technical accessibility within publishing tools is also crucial. Better and more advanced built-in support to enhance accessibility makes it easier for content creators to produce accessible materials efficiently. AI can, for example, convert inaccessible PDF documents into text that complies with accessibility requirements [8].

For employees and software developers with disabilities, it is important that AI tools themselves are accessible. A quick check with the browser extension of the Wave Web Accessibility Evaluation tool [9] indicates that widely adopted AI tools, such as ChatGPT and Copilot have accessibility issues. However, social media discussions reveal that these solutions are adopted by people with disabilities. This is likely due to their relatively simple interfaces, which allows for navigation and use despite some accessibility issues. In contrast, solutions that generate still images, audio, and video seem to face greater accessibility challenges, especially for visually impaired people. Fortunately, there are several examples of AI-based solutions that have been developed to be accessible for people with disabilities. For example, a voice interface has been created for ChatGPT [10], and a solution has been developed to make the selection and generation of AI based images more accessible through rich image descriptions [11].

[2] The W3C's Web Accessibility Initiative maintains an extensive list at *https://www.w3.org/WAI/policies/*.

AI can also be used to simulate how users interact with websites, and thus help in optimizing the design, including navigation options [3]. This could lead to more intuitive interfaces, and thus quicker access to information and improved user satisfaction.

When exploring how AI can benefit the development of software, one must also examine potential ethical issues and unintended consequences. Research indicates that today's AI often perpetuates biases, reinforcing stereotypes and discriminating against minorities and people with disabilities [1,3,12]. This could be an issue with user simulation software.

Securing funding for developing more accessible solutions remains a challenge [3]. While current research on AI for accessibility has predominantly centered around solutions for people with visual impairments, it is also important to address the needs of other user groups [3]. To truly enhance inclusivity through the use of AI technologies, further development and adaptations are required in various areas, including managing informal spoken language, dialects, sign language, and minority languages [13].

2.1. Tools for automatic testing of web accessibility

The Web Content Accessibility Guidelines (WCAG), developed by the W3C Web Accessibility Initiative, is designed to ensure that web content is accessible to everyone, including people with disabilities [14]. WCAG has evolved through several versions — 2.0, 2.1, and the latest 2.2 — to address the changing needs of web accessibility. These guidelines are embedded in legislation across many countries, including in the EU [15], underscoring their importance for web accessibility. However, evaluating conformance with the WCAG guidelines manually is time-consuming, especially for large websites. Therefore, various tools to help developers assess WCAG conformance have been developed. The tools, which are often referred to as WCAG checkers, are software programs or online services that help in determining if a web solution meets the WCAG guidelines. The results of existing WCAG checkers differ from one tool to another, and they often cover different subsets of the guidelines [15,16]. While there are great expectations for the use of AI to enhance and extend accessibility testing tools [17], there is currently little research in this area [18].

Researchers with similar ideas to ours conducted a pilot study to investigate the potential of a large language model (LLM) to evaluate certain web accessibility success criteria [18]. They focused on three WCAG criteria that traditionally require manual assessment: 1.1.1 Non-text Content, 2.4.4 Link Purpose (In Context), and 3.1.2 Language of Parts. To this end, they developed LLM-based scripts to evaluate conformance to these criteria and compared the results against existing WCAG checkers. They found that the LLM-based scripts could effectively identify accessibility issues that the current WCAG checkers overlooked. They suggest that further research should attempt to replicate the same tests using open-source models.

3. AI-based WCAG check: Our approach

We selected four accessibility success criteria from the Web Content Accessibility Guidelines (WCAG) that traditional WCAG checkers have been unable to test reliably, but which could be automatically verified using open-source machine-learning models. Moreover, the selected guidelines are particularly important for enhancing accessibility for users who depend on assistive technology. Improved compatibility with assistive technology will have huge benefits for these users [19]. We selected four guidelines, two of which were the same as tested by [18]:

1.1.1 Non-text Content (Level A): Checking whether the use of "alt text" for images is descriptive. This helps users who rely on screen readers to understand visual content. For this, we took advantage of recent advances in multi-modal deep learning, which connects computer vision and natural language processing.

1.4.5 Images of Text (Level AA): Checking whether there is text inside images, and whether or not the text is embedded as regular text elsewhere on the page. If not, assistive technology like screenreaders (which convert a webpage into synthetic voice) cannot detect it. Consequently, the users will not have access to the text. For this, we used methods for Optical Character Recognition (OCR) to detect text in images.

3.1.1 Language of Page (Level A): Checking whether the correct language is specified for the entire page. This is important to ensure that screen readers and other assistive technologies can correctly interpret and vocalize the content. For this we combined traditional testing with Natural Language Processing (NLP) based automatic language detection.

3.1.2 Language of Parts (Level AA): Checking whether the correct language is specified for all text parts. This will help users who rely on assistive technologies to understand multilingual content accurately. For this we combined traditional testing with NLP.

4. Results

For the prototypes, we chose to use several open-source libraries where the AI models are already pre-trained. In the following, we describe the four prototypes that were developed, illustrating the application of the specified techniques:

Prototype 1.1.1 calculates a score for the descriptiveness of alt text in images on a website. We utilized CLIP (Contrastive Language-Image Pre-Training) [20] a neural network that has been trained on image-text pairs. The model consists of an image encoder and a text encoder that were trained on image-text pairs.

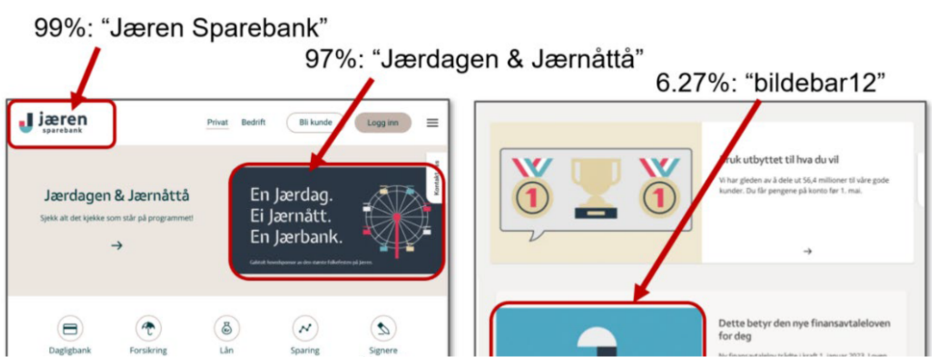

Figure 1: Prototype 1.1.1 Estimating how descriptive an alternative text is

During training, for each image the model is tasked with predicting which text description out of a large set of descriptions belong to the image. In other words, the model is tasked with creating similar text encodings and image encodings for image-text pairs. Using a pre-trained CLIP model, we can extract the image encodings from images from the website and text encodings from the associated alt-text (if available). By comparing the encodings, we can get a measure of how similar the image and the associated alt-text are. For example, in the first image in Figure 1 the similarity score between the alternative text "Jæren Sparebank" and the predicted image description is 99%. For the image in the lower right corner, the similarity score between the meaningless alternative text "bildebar12" and the image of a question mark is only 6,27%. Both similarity scores seem reasonable.

Prototype 1.4.5 extracts text from images, which can then be used to check whether the page includes this text if it is deemed important. Optical character recognition (OCR), extracting machine-readable text from images of text, is used in many different fields and hence there are many models easily available. We chose the EasyOCR Python library, which supports 80+ languages and multiple models for text detection and recognition. When the text is extracted, it can be compared to the text on the website using NLP methods. We found that the models implemented in EasyOCR performed well, meaning that they could convert the text in images to plain text with high reliability.

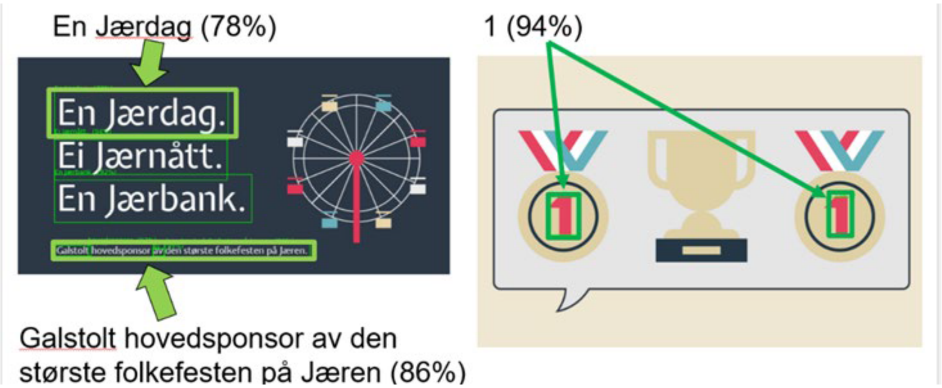

Figure 2: Prototype 1.4.5 Extracting text from images

Figure 2 shows examples of text extracted from images with an estimated detection score in parenthesis, which measures how certain the model is that the text has been correctly parsed. All the text in the example is identified correctly. If the text in the image is very small, such as a logo on a laptop in the background or a quote on a T-shirt, the models may correctly identify that text is present, but the text is not necessarily parsed correctly. However, in most of these instances, the text is not meant to convey information to the reader. Hence, we chose to discard text that covered less than a pre-determined portion of the image. We also discarded very short text snippets, such as the number "1" in Figure 2.

Prototype 3.1.1 detects the language for the main text on a website. This is then compared to the specified language coding, if available, of the web page. We chose to use the LanguageIdentifier Python library, which implements language identification using pre-trained neural models. These models return the detected language as well as a detection score, which can be read as the likelihood that the language was detected correctly. We extracted the text from the websites, divided the text into segments (e.g., lines) and ran these segments through the language detection model. We then compared the detected language against the language attribute in the HTML. This allowed us to determine whether the page has the correct language specification.

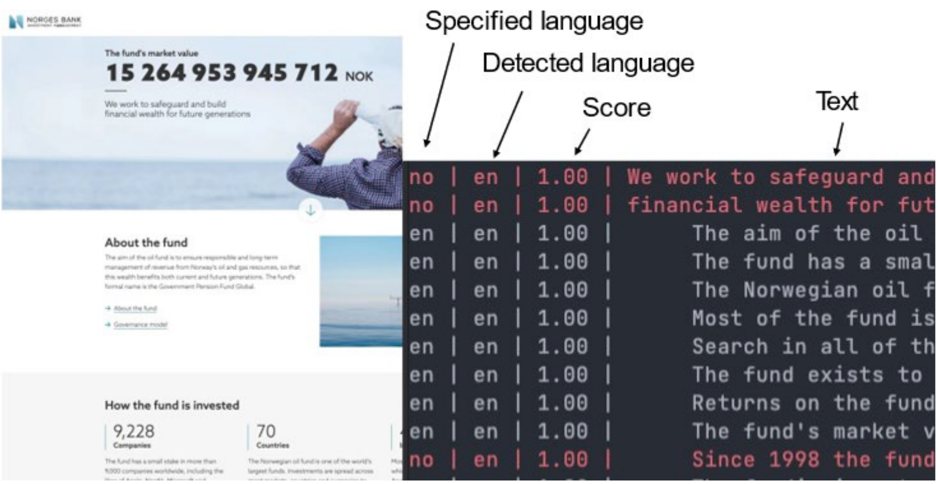

Figure 3: Prototype 3.1.1 Detecting language of text parts and language coding on a web page

Figure 3 shows a screen shot of a web page together with the corresponding table from prototype 3.1.1. For each language part on the web page there is a line in the table with specified language, detected language, a certainty score, and the actual text part. Lines with mismatch between specified and detected language are highlighted in red. We found that the model performed quite well, meaning that in most cases the language could be predicted with 100% confidence.

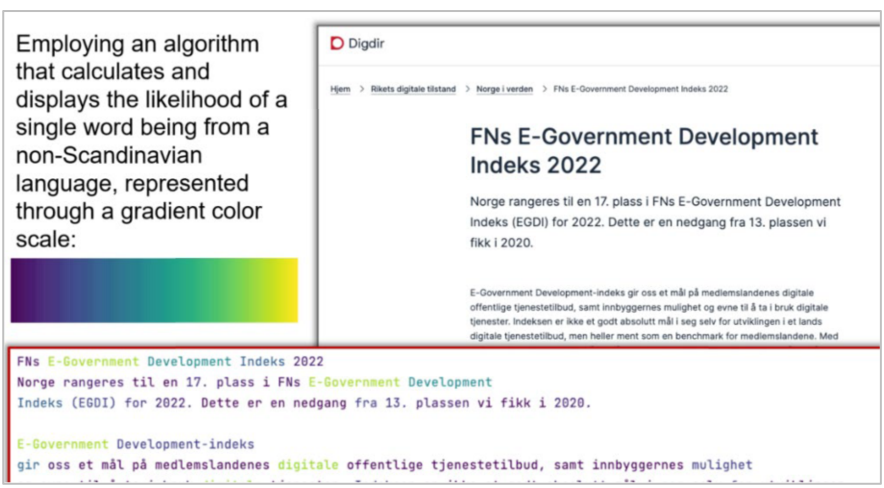

Figure 4: Prototype 3.1.2: Detecting language of text parts on a web site

Prototype 3.1.2 detects the language for text parts on a web site. We used pre-trained language detection models, similar to those in prototype 3.1.1: extracting text segments from the page and checking the language using the LanguageIdentifier library.

The algorithm displays the likelihood of a word being from a non-Scandinavian language using a colour scale, as illustrated in Figure 4. However, the colour scale itself requires some work to be more accessible. We observed that the language identifier model had difficulty distinguishing between Scandinavian languages in short text segments. Therefore, we merged the scores for Norwegian, Swedish, and Danish. Thus, the prototype may be suitable for testing if Scandinavian web pages have text parts in other languages.

5. Discussion

While we obtained good results with prototypes based on open-source machine learning models, they have been tested on a limited number of websites. In further work, a systematic evaluation should be conducted to provide more information about the accuracy of the prototypes. Although the prototypes can be improved in several areas, they demonstrate great potential in enhancing existing WCAG checkers. This is in line with the findings of [18], where LLMs were used to test three WCAG criteria, two of which we also tested.

Despite the significant improvement in automation, human verification of the algorithm results is still advisable. For instance, Prototype 1.4.5 generally detects text in images correctly but cannot always determine its relevance to the webpage content. However, as the fields of computer vision and natural language processing advance, verification accuracy will likely improve.

Furthermore, these techniques can not only be used to check if WCAG criteria are met, but also suggest improvements to accessibility of web sites. For example, the algorithms used in prototype 1.1.1. can be used to suggest better alternative texts.

6. Conclusion

After reviewing AI-enhanced solutions for digital accessibility and usability, we presented four prototypes for identifying accessibility issues on webpages using open-source machine learning models. While operational, these prototypes require further fine-tuning to reduce false positives and negatives. Currently, Prototype 1.4.5 processes OCR images without analyzing if the detected text appears elsewhere on the page.

These prototypes demonstrate the potential of AI-based techniques in developing universally designed ICT solutions. They can estimate the risk of violating specific WCAG criteria and, with further work, suggest improvements. In conclusion, AI-based techniques could significantly reduce the need for manual checks in accessibility testing.

Acknowledgement

This research is partly funded by the Norwegian Research Council (#342640 and #321821).

References

[1] I. Peinado, E. De Lera, J. Usero, C. Clark, J. Treviranus, and G. Vanderheiden, Digital Inclusion at the Workplace Post Covid19, in: Proceedings of the 13th International Joint Conference on Computational Intelligence, SCITEPRESS - Science and Technology Publications, Valletta, Malta, 2021: pp. 460–467. doi:10.5220/0010722900003063.

[2] L.S. Pereira, and C. Duarte, AI and Media Accessibility: An overview, Cost action: Lead-Me, 2023. https://lead-me-cost.eu/resources/LEAD-ME_AIandAccessibility.pdf.

[3] K. Chemnad, and A. Othman, Digital accessibility in the era of artificial intelligence—Bibliometric analysis and systematic review, *Frontiers in Artificial Intelligence*. **7** (2024). https://www.frontiersin.org/articles/10.3389/frai.2024.1349668 (accessed February 24, 2024).

[4] F. Ullah, N.A. AbuAli, A. Ullah, R. Ullah, U.A. Siddiqui, and A.A. Siddiqui, Fusion-Based Body-Worn IoT Sensor Platform for Gesture Recognition of Autism Spectrum Disorder Children, *Sensors*. **23** (2023) 1672. doi:10.3390/s23031672.

[5] M.C. Suárez-Figueroa, I. Diab, Á. González, and J. Rivero-Espinosa, Towards an Automatic Easy-to-Read Adaptation of Morphological Features in Spanish Texts, in: J. Abdelnour Nocera, M. Kristín Lárusdóttir, H. Petrie, A. Piccinno, and M. Winckler (Eds.), Human-Computer Interaction – INTERACT 2023, Springer Nature Switzerland, Cham, 2023: pp. 176–198. doi:10.1007/978-3-031-42280-5_12.

[6] K.S. Glazko, M. Yamagami, A. Desai, K.A. Mack, V. Potluri, X. Xu, and J. Mankoff, An Autoethnographic Case Study of Generative Artificial Intelligence's Utility for Accessibility, in: Proceedings of the 25th International ACM SIGACCESS Conference on Computers and Accessibility, Association for Computing Machinery, New York, NY, USA, 2023: pp. 1–8. doi:10.1145/3597638.3614548.

[7] D. Cane, Task Aware Browsing: AI Assisted Web Access for Blind Users, (2023). http://dspace.calstate.edu/handle/10211.3/225162 (accessed November 20, 2023).

[8] R. van Heusden, H. Ling, L. Nelissen, and M. Marx, Making PDFs Accessible for Visually Impaired Users (and Findable for Everybody Else), in: O. Alonso, H. Cousijn, G. Silvello, M. Marrero, C. Teixeira Lopes, and S. Marchesin (Eds.), Linking Theory and Practice of Digital Libraries, Springer Nature Switzerland, Cham, 2023: pp. 239–245. doi:10.1007/978-3-031-43849-3_21.

[9] WebAIM, WAVE Chrome, Firefox, and Edge Extensions, *WAVE Web Accessibility Evaluation Tool*. (n.d.). https://wave.webaim.org/extension/ (accessed August 9, 2024).

[10] A. Kuzdeuov, S. Nurgaliyev, and H.A. Varol, ChatGPT for Visually Impaired and Blind, (2023). doi:10.36227/techrxiv.22047080.v2.

[11] M. Huh, Y.-H. Peng, and A. Pavel, GenAssist: Making Image Generation Accessible, in: Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology, Association for Computing Machinery, New York, NY, USA, 2023: pp. 1–17. doi:10.1145/3586183.3606735.

[12] F. Bianchi, P. Kalluri, E. Durmus, F. Ladhak, M. Cheng, D. Nozza, T. Hashimoto, D. Jurafsky, J. Zou, and A. Caliskan, Easily Accessible Text-to-Image Generation Amplifies Demographic Stereotypes at Large Scale, in: Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency, Association for Computing Machinery, New York, NY, USA, 2023: pp. 1493–1504. doi:10.1145/3593013.3594095.

[13] S. Skavlid, Kunstig intelligens og sensorteknologi – kunnskapsinnsamling, MediaLT, Oslo, 2022.

[14] W.W.A. Initiative (WAI), WCAG 2 Overview, *Web Accessibility Initiative (WAI)*. (n.d.). https://www.w3.org/WAI/standards-guidelines/wcag/ (accessed August 18, 2024).

[15] A. Bai, K.S. Fuglerud, R.A. Skjerve, and T. Halbach, Categorization and Comparison of Accessibility Testing Methods for Software Development, *Studies in Health Technology and Informatics*. (2018). doi:10.3233/978-1-61499-923-2-821.

[16] R. Ismailova, and Y. Inal, Comparison of Online Accessibility Evaluation Tools: An Analysis of Tool Effectiveness, *IEEE Access*. **10** (2022) 58233–58239. doi:10.1109/ACCESS.2022.3179375.

[17] B. Stewart, A. Batchu, and W. Grant, Market Guide for Digital Accessibility, Gartner Research, 2024. https://www.gartner.com/en/documents/5410363 (accessed August 17, 2024).

[18] J.-M. López-Gil, and J. Pereira, Turning manual web accessibility success criteria into automatic: an LLM-based approach, *Univ Access Inf Soc*. (2024). doi:10.1007/s10209-024-01108-z.

[19] F.H.F. Botelho, Accessibility to digital technology: Virtual barriers, real opportunities, *Assistive Technology*. **33** (2021) 27–34. doi:10.1080/10400435.2021.1945705.

[20] A. Radford, J.W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, Learning Transferable Visual Models From Natural Language Supervision, (2021). http://arxiv.org/abs/2103.00020 (accessed August 19, 2024).