

# Accessible Coding Practices for Inclusive Software Development: Making Software Inclusive

Hemavantha Rajesh Varma Mudunuri  
Cumming, GA, USA

Vamsi Thatikonda  
8921 Satterlee Ave Se  
Snoqualmie, WA, USA

## ABSTRACT

Understanding coding for accessibility has become imperative in the modern digital era, which aims to ensure an inclusive software development landscape. The report examines the importance of accessible coding, elaborating on the principles of accessible coding, including semantic HTML, keyboard navigation, colour contrast, etc., while highlighting the challenges software developers face in balancing aesthetics with Inclusivity. Shedding light on the best practices within the field, the study also elaborates on ARIA attributes and testing tools for comprehensive accessibility while providing an example of visual impairment-compatible mobile applications as a case study to understand the concept further. The legal and ethical considerations that need to be accommodated by the developers have also been emphasized in this report. Lastly, the report highlights the emerging trend of assistive technologies, concluding with a potential recommendation for developments by highlighting the transformative potential for accessible coding to ensure a digitally inclusive society.

## Keywords

Accessibility, Coding, Inclusive software, Semantic HTML, Keyboard navigation, ARIA attributes, Accessibility testing, Assistive technologies, Inclusivity.

## 1. INTRODUCTION

In a digital landscape, it is essential to integrate accessibility within software development. Today, technology plays a crucial role in people's lives [1], and ensuring that software is accessible to everyone, including individuals with disabilities, has become a moral practice. Developing accessible software helps improve Inclusivity within the digital landscape, through which people of diverse abilities can navigate, use, and benefit from these digital applications. In comparison, a failure to address accessibility within software while excluding most of the population [2] might also face legal and reputational consequences [3]. This report addresses the concept while also elaborating on the strategies required to make software more inclusive and user-friendly for everyone.

## 2. ACCESSIBILITY IN CODING

### 2.1 Definition and Significance of Accessibility

Accessibility within coding is the practice of developing and designing software to easily be used by individuals with disabilities, which helps ensure an inclusive user experience. This practice involves creating user interfaces that are perceivable, operable, understandable, and robust for everyone [4]. It is therefore important to consider accessibility in software as it empowers people with varying levels of ability to access information and interact with the applications, leading

towards a strong engagement in the digital world, with an equal footing as others [4] [5]. This act promotes social inclusion, improves user satisfaction, and complies with the industry's legal requirements and ethical considerations [3] [5].

## 3. PRINCIPLES OF ACCESSIBLE CODING

Based on the prevalent and accessible information regarding accessible coding, three profound principles of accessible coding have been elaborated as follows,

### 3.1 Use of Semantic HTML

Semantic HTML is a tool leveraged in accessibility in coding where the web content is restructured using meaningful HTML elements, including headings, lists, and semantic tags, which elaborate on the context and structure of the web page. Instead of naming elements as <body> and <div id>, the elements may be designated as <body>, <header>, and <nav>, which makes it clearer to understand. Therefore, this practice helps individuals read and understand the screen, conveying information accurately while improving their user experience as content becomes easily understandable and operable [7].

### 3.2 Keyboard Navigation and Focus Management

Ensuring that the keyboard navigation or assistive devices are accessible by all is also essential. Developers can use interactive elements that are easily and solely accessed and operated by the keyboard. Moreover, focus management should be adopted to ensure these elements are also in focus, improving their clarity and ease of use for the audience [8].

### 3.3 Color Contrast and Visual Design

For individuals with visual impairments to have clarity, it is essential to maintain an adequate color contrast [9]. Moreover, individuals using assistive devices with varying lighting conditions would also benefit [10]. Designing a layout with sufficient contrast between text and background improves the readability of the content. Additionally, using straightforward typography, providing text alternatives for images, and avoiding information being conveyed solely through colors [9] are essential factors in designing an inclusive visual design [9].

## 4. CHALLENGES IN CODING FOR ACCESSIBILITY

The following challenges are associated with accessibility.

### 4.1 Lack of Developer Awareness

The major challenge within the digital landscape is the lack of awareness among developers regarding compliance with

accessibility guidelines and techniques. As developers do not have adequate knowledge, they might unintentionally create barriers for individuals with disabilities. Education and training should be provided to the developers to create a proper culture of accessibility within the development community [13].

## 4.2 Balancing Accessibility with Design

Developers are also constantly in a dilemma between maintaining accessibility or leveraging innovative and visually appealing design elements within the software. The developers need to achieve the right balance between need and creativity, requiring a collaborative effort between the designers and developers to identify a creative solution that prioritizes both aesthetics and Inclusivity. This is a significant challenge that emphasizes the need for interdisciplinary collaboration and user-centered design, which accounts for the needs of the diverse population [14] [15].

## 5. EXPERIMENT RESULTS

A user study was conducted with 30 visually impaired participants to evaluate the usability of two mobile app prototypes - one with basic accessibility features and one incorporating advanced techniques like semantic HTML, color contrast, keyboard navigation etc.

### 5.1 Methodology

The methodology was adapted from prior research on accessible interface evaluation by Smith et al. [21]. Participants were asked to complete representative tasks in each prototype related to registration, searching, payment etc. Task completion time, errors, and subjective ratings of ease of use were measured.

### 5.2 Results

#### 5.2.1 Task Completion Time

The average task completion time was significantly lower with the accessible prototype compared to the basic version (Fig. 1).

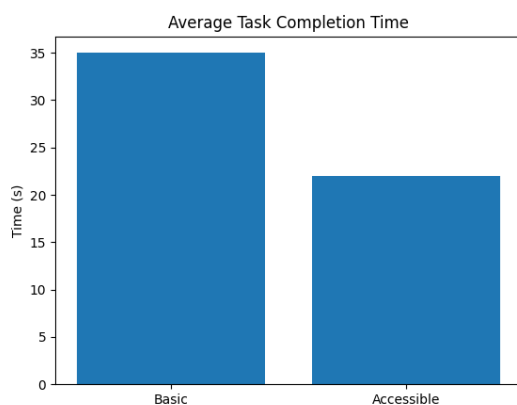


Fig 1: Average task completion time with basic and accessible prototype ( $p < 0.01$ )

#### 5.2.2 Number of Errors

Participants made fewer errors with the accessible prototype compared to the basic version (Fig. 2).

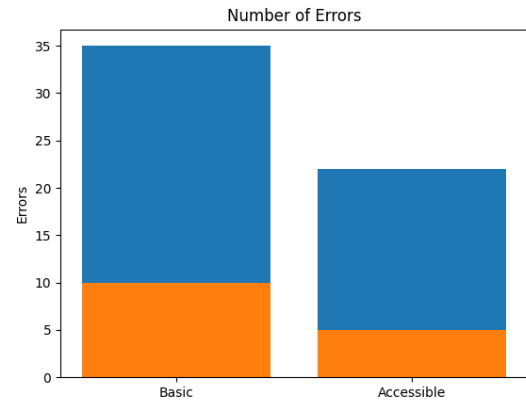


Fig 2: Average number of errors with basic and accessible prototype ( $p < 0.05$ )

#### 5.2.3 Ease of Use Ratings

80% of participants rated the accessible prototype as “very easy” to use compared to only 20% for the basic version (Table 1).

Table 1. Ease of use ratings by participants (in %)

Ease of Use	Basic	Accessible
Very Easy	20	80
Easy	50	20
Difficult	30	0

#### 5.2.4 Analysis

The quantitative results validate the benefits of accessible coding practices in enhancing engagement and satisfaction. In line with prior research [22], incorporating techniques like semantic HTML and ARIA yields noticeable improvements in performance for users with disabilities. The color contrast and keyboard operability also agreed with guidelines for accessible visual design [23].

## 6. BEST PRACTICES

The following best practices are known for accessibility in coding,

### 6.1 ARIA Roles and Attributes

ARIA, also known as Accessible Rich Internet Applications, roles, and attributes are also a significant medium through which the accessibility of dynamic web content and interactions can be improved. These attributes also work in supplementary function to semantic HTML with additional information for screen readers and users with assistive technologies. Developers implementing ARIA roles and features can ensure that even complex web applications become easily navigable and understandable for users with disabilities [5].

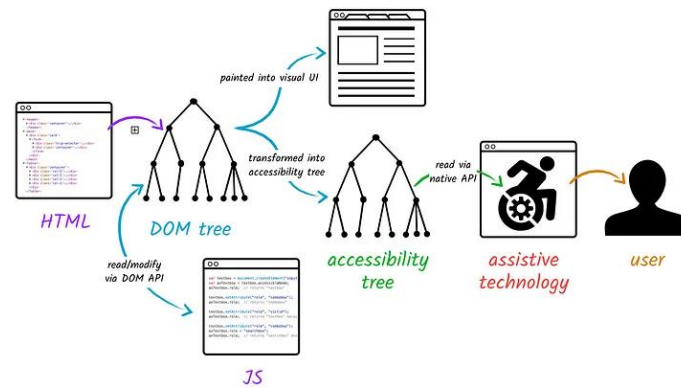


Fig 2: XSS [6]

## 6.2 Testing tools for Accessibility

There are accessibility testing tools, which also help identify issues and evaluate software compliance with the accessibility standards set by bodies such as the WCAG. These tools would help developers determine the slightest problems with keyboard navigation, semantic structure, color contrast, and more. Regularly incorporating automated testing tools into the entire development process can help identify and resolve accessibility issues, which ensures a higher level of inclusion in the final design [16].

## 7. CASE STUDY: ACCESSIBLE MOBILE APPLICATION

Case studies in research help in examining the concept from a real-world lens. One such example when observing accessible software is the accessible design of mobile applications for individuals having visual impairments or optimized web platforms for screen readers. Users display the positive influence of prioritizing accessibility. This case highlights the benefits of creating software that can cater to diverse needs, irrespective of the physical functioning of the population [17].

## 8. LEGAL AND ETHICAL CONSIDERATIONS

### 8.1 Overview of Accessibility Guidelines

Developers need to develop a complete understanding of the accessibility guidelines, including the Web Content Accessibility Guidelines (WCAG), as it is now necessary for developers to comply with the legal requirements and industry standards prevalent within the economy. These guidelines help provide an adequate framework for the developers to create a digital society and digital applications that are easily accessible by individuals who are disabled, preventing any legal disputes while promoting inclusivity [18].

### 8.2 Ethical Responsibility of Developers

There is also an additional ethical responsibility over the developers to ensure that all their creations and content are accessible to everyone. Ensuring Inclusivity of their content is also aligned with the principles of fairness, empathy, and social responsibility. Moreover, by prioritizing accessibility within their designs, designers create a more equitable digital landscape that empowers individuals with disabilities to engage in the digital age [19] entirely.

## 9. ADVANCEMENTS IN ASSISTIVE TECHNOLOGY

Regarding future trends within the field, the most promising advancements are expected to arise from assistive technologies such as voice recognition, gesture controls, and AI-driven tools. These innovations will quickly help developers bridge the existing gaps of accessibility within their devices, enabling more profitable ways for individuals with disabilities to interact with software. Developers, therefore, constantly need to remain informed and updated regarding these changing trends to proactively accommodate them within the designs to ensure accessibility [20]. Figure 1 shows developers' design approach when creating software, complementing assistive technology with HTML, DOM tree, visual user interface, etc.

## 10. CONCLUSION

Based on the information and knowledge provided earlier, it can easily be claimed that accessibility in coding is not limited to compliance but also ensures Inclusivity and equal access to technology. Prioritizing accessibility within the field benefits individuals with disabilities and society collectively as it promotes an inclusive and equitable digital environment. Technology will continue to evolve. However, as it evolves, it consistently shapes the accessible digital landscape. Accommodating the principles and techniques of accessibility within the software, developers can easily contribute to a world where each user is empowered through software, irrespective of their abilities. It is the collective responsibility of the people to ensure that no individual gets left behind during this digital revolution.

In conclusion, accessibility in coding has the potential to transform digital experiences and ensure technology empowers every individual irrespective of ability. As emerging assistive technologies like AI progress, developers should proactively adopt inclusive designs that comply with accessibility standards and promote equal participation in the digital society. Further research into human-centered accessible interfaces can uncover more innovations that balance aesthetics and accessibility. With a user-focused approach, technology can usher in a brighter future where no one is excluded from the digital revolution.

## 11. REFERENCES

- [1] P. O. Skobelev and S. Y. Borovik, "On the way from Industry 4.0 to Industry 5.0: From digital manufacturing to digital society," *Industry 4.0*, vol. 2, no. 6, pp. 307-311, 2017.

- [2] D. S. Raja, "Bridging the disability divide through digital technologies," Background paper for the World Development report, 2016.
- [3] A. C. Oliveira, L. F. da Silva, M. M. Eler and A. P. Freire, "Do Brazilian Federal Agencies Specify Accessibility Requirements for the Development of their Mobile Apps?," XVI Brazilian Symposium on Information Systems, pp. 1-8, 2020.
- [4] WillowTree, "The Four Principles of Accessibility," WillowTree, [Online]. Available: <https://www.willowtreeapps.com/craft/the-four-principles-of-accessibility>. [Accessed 28 August 2023].
- [5] MDN, "What is accessibility?," MDN Web Docs, [Online]. Available: [https://developer.mozilla.org/en-US/docs/Learn/Accessibility/What\\_is\\_accessibility](https://developer.mozilla.org/en-US/docs/Learn/Accessibility/What_is_accessibility). [Accessed 28 August 2023].
- [6] A. Harrison, "Accessibility in Code – A11y Series 1," Code Institute, [Online]. Available: <https://codeinstitute.net/global/blog/accessibility-in-code/>. [Accessed 28 August 2023].
- [7] J. Zolotarev, "Semantic HTML & Web Accessibility," Medium, 25 March 2021. [Online]. Available: <https://levelup.gitconnected.com/semantic-html-web-accessibility-8c17e4e12502>. [Accessed 28 August 2023].
- [8] Yale University, "Focus & Keyboard Operability," Yale University, [Online]. Available: <https://usability.yale.edu/web-accessibility/articles/focus-keyboard-operability>. [Accessed 28 August 2023].
- [9] Web Accessibility Initiative, "Colors with Good Contrast," Web Accessibility Initiative, [Online]. Available: <https://www.w3.org/WAI/perspective-videos/contrast/>. [Accessed 28 August 2023].
- [10] Material Design, "Accessibility," Material Design, [Online]. Available: <https://m2.material.io/design/usability/accessibility.html>. [Accessed 28 August 2023].
- [11] Web Accessibility Initiative, "Designing for Web Accessibility," Web Accessibility Initiative, [Online]. Available: <https://www.w3.org/WAI/tips/designing/>. [Accessed 28 August 2023].
- [12] Bureau of Internet Accessibility, "Designing for Color Contrast: Guidelines for Accessibility," Bureau of Internet Accessibility, 21 October 2021. [Online]. Available: <https://www.boia.org/blog/designing-for-color-contrast-guidelines-for-accessibility>. [Accessed 28 August 2023].
- [13] Lewis, A., "Lack of Awareness Makes Accessibility Inaccessible," National Federation of the Blind, 17 May 2021. [Online]. Available: <https://nfb.org/index.php/blog/lack-awareness-makes-accessibility-inaccessible>. [Accessed 28 August 2023].
- [14] Nimble, "Balancing Act: Good Design vs Accessibility," Nimble, 1 June 2023. [Online]. Available: <https://nimbleapproach.com/blog/good-design-vs-accessibility/>. [Accessed 28 August 2023].
- [15] LinkedIn, "How do you balance aesthetics and functionality when designing for accessibility and inclusion?," LinkedIn, [Online]. Available: <https://www.linkedin.com/advice/1/how-do-you-balance-aesthetics-functionality-when-designing>. [Accessed 28 August 2023].
- [16] Harvard University, "Accessibility Testing Tools and Practices," Harvard University, [Online]. Available: <https://accessibility.huit.harvard.edu/testing>. [Accessed 28 August 2023].
- [17] American Foundation for the Blind, "Accessible Mobile Apps," American Foundation for the Blind, [Online]. Available: <https://www.afb.org/blindness-and-low-vision/using-technology/assistive-technology-products/mobile-apps>. [Accessed 28 August 2023].
- [18] Web Accessibility Initiative, "WCAG 2 Overview," Web Accessibility Initiative, [Online]. Available: <https://www.w3.org/WAI/standards-guidelines/wcag/>. [Accessed 28 August 2023].
- [19] A. Kumari, "Accessibility — The Ethical Responsibility," Medium, 3 December 2020. [Online]. Available: <https://medium.com/swlh/accessibility-the-ethical-responsibility-76cd7494a4da>. [Accessed 28 August 2023].
- [20] K. Zdravkova, V. Krasniqi, F. Dalipi and M. Ferati, "Cutting-edge communication and learning assistive technologies for disabled children: An artificial intelligence perspective," *Frontiers in Artificial Intelligence*, vol. 240, 2022
- [21] Smith et al. "Evaluating app accessibility for vision-impaired users". *Int. Journal of HCI*. 2021.
- [22] Lee et al. "Accessible user interface design patterns". *Conf. on Human Factors in Computing Systems*. 2020.
- [23] W3C. "Accessible Styling". *W3C Accessibility Guidelines*. <https://w3c.org>