Innovative: International Multi-disciplinary Journal of Applied Technology
(ISSN 2995-486X) VOLUME 04 ISSUE 07, 2026

EVALUATION OF WCAG CRITERIA IN DIGITAL LEARNING PLATFORMS USING A HYBRID AUDIT APPROACH

Sevinch Jovlieva
Academy of Islamic Studies of Uzbekistan, Third-Year Student in Information Security Management

Abstract:
Ensuring digital accessibility in e-learning platforms is a primary requirement for creating equal educational opportunities for all students. This study examines the practical effectiveness of a hybrid audit approach that combines automated software tools, specifically Axe-core and Google Lighthouse, with manual keyboard testing. As a practical testing ground, a prototype of the DigiLab learning platform developed by the author was used as the research object. While automated scanners are highly useful for the rapid detection of static code errors, their technical algorithms are limited to identifying only about 20 to 25 percent of overall WCAG standard violations. Full evaluation of interactive functionality and real user experience requires dynamic testing beyond mere source code analysis, a limitation openly acknowledged by the tool developers in their official documentation. Manual testing conducted using a standard keyboard successfully revealed these hidden semantic and navigational errors that the software had overlooked. After resolving all identified issues, the platform's initial Lighthouse score increased from 68 to 95 points, while Axe-core recorded zero errors in the final audit. The experiment demonstrates that to achieve authentic digital inclusion in educational resources, automated assessments must be systematically complemented by manual human verification.

Keywords: Web Accessibility, Digital Learning, Axe-Core, Google Lighthouse, Manual Testing, Digilab, WCAG Standards, Semantic Audit, User Experience

Introduction

Polyurethanes Digital learning platforms have today become the primary means of instruction and independent study at institutions of higher education. Students make extensive use of electronic libraries, learning portals, and testing systems. In Uzbekistan as well, as part of the broader digital transformation, ensuring that all educational portals are fully accessible to persons with disabilities has been established as a legal obligation [1]. Yet such platforms frequently fail to fully meet the needs of students with disabilities — for example, users with visual impairments, limited mobility, or those who rely solely on a keyboard rather than a mouse [2]. For this reason, taking into account the needs of users with diverse abilities requires particular attention when building such platforms.

Automated tools that are widely used around the world are typically employed to test the accessibility of websites. These tools can analyze technical indicators within the code in a short amount of time [3]. However, they can detect only certain errors within the code itself. It is therefore necessary to supplement them with manual navigation testing, an approach that makes it possible to connect the correctness of the code with students' actual usage experience. This paper compares automated and manual keyboard testing using the DigiLab platform as a case study, and examines methods for identifying and resolving issues throughout this process.

The question of ensuring accessibility for all in digital education has been extensively studied by international researchers. Sloan and colleagues, examining the accessibility of university websites, showed that adherence to inclusive design principles creates equal opportunity for all students [4]. Otherwise, if online resources fail to meet the needs of students with disabilities, the digital divide widens further. Burbules and Torres likewise emphasize that distance-learning platforms must be open and accessible to all social groups [5].

The most widely recognized benchmark for evaluating website accessibility worldwide is the WCAG (Web Content Accessibility Guidelines) international standard. This framework rests on four core principles and specific rules intended to make digital content accessible to everyone [6]. The table below presents the core principles of the WCAG standard along with their practical aspects:

Table 1. Core principles and directions of the international web accessibility standard [7]

Principle	Core purpose and meaning	Aspects checked in practice
Perceivable	Users must be able to receive site information through their senses	Presence of alt text for images and correct color contrast
Operable	Interface elements and the navigation system must be easy for everyone to operate	Stability of keyboard navigation and a clearly visible focus outline on screen
Understandable	Site content and all control buttons must be understandable to the reader	Correct configuration of the document language and text labels on input fields
Robust	The web resource must continue to function correctly across various future programs and technologies	Correct use of semantic tags and logical attributes within the code

The WCAG-EM methodology defines the stages for evaluating web resources from an accessibility standpoint [8]. Abascal and colleagues, comparing automated testing with manual review methods, showed that projects relying solely on automated tools frequently misjudge the true level of accessibility [9].

The capabilities of tools such as Axe-core and Google Lighthouse have been examined in various studies. Brajnik showed that while automated review software effectively evaluates code quality, it does not fully capture a page's semantic structure or usability [10]. Vigo and colleagues, meanwhile, demonstrated that even the most advanced automated tools detect only a portion of real errors, and that the remaining errors must be found through manual keyboard testing [11]. For this reason, combining both methods is essential when evaluating educational platforms.

Materials and Method

The study used a prototype of the DigiLab platform developed by the author. For the audit, the core sections of the interface were selected — navigation, course sections, the registration window, and the footer. Testing was carried out on the localhost version of the platform.

Axe-core was used to check the code's compliance with international standards, and Google Lighthouse was used to measure the overall web accessibility score. Manual navigation testing was performed using a keyboard, with results monitored in the Chrome browser console. The process was carried out in the following six stages:

Table 2. Stages of the study and their practical content

Work stage	Brief summary of actions performed
Initial automated audit	The Axe-core library was connected to the project and run in development mode, and all software errors in the browser console were logged
Structure-level fixes	The identified code errors were corrected in line with international standard requirements and the program was re-run
Manual functional audit	The entire interface was navigated using only the primary navigation controls, without using a computer mouse
Resolving functional issues	The deficiencies identified through the manual audit were fixed one by one in the interface code
Repeat automated audit	Various component states, including the open-modal view, were re-checked
Final evaluation stage	A final control audit was conducted using both tools, and the resulting quantitative and qualitative findings were documented for analysis

Results

At the initial stage, Lighthouse recorded a score of 68, and Axe-core flagged six errors. These were mainly related to missing alternative text on images and insufficient color contrast. Following this, manual testing was performed using the keyboard in accordance with the WCAG-EM methodology. While the automated tools analyzed the static code well, they did not fully capture the user's real-time interactions with the interface. Manual testing revealed five issues: deficiencies in button elements, an invisible focus outline, and the absence of ARIA attributes in the modal window. In particular, interactive elements in the navigation and course sections had been written as div tags with an onClick attribute, which are not operable via keyboard.

The five identified issues were resolved as follows:

1. Div elements with onclick were converted to <button> elements;
2. A focus outline (outline, box-shadow, color change) was added;
3. The attributes role="dialog", aria-modal, aria-labelledby, and aria-live were added to the modal window;
4. <label> elements along with aria-required and aria-describedby were added to form fields;
5. Focus movement was tracked via a focusin event listener.

After these fixes, a repeat test confirmed that all elements were accessible via keyboard. Subsequently, when re-checking the various states of the components, five further contrast errors were identified. The text-to-background ratio, which had measured between 3.9:1 and 4.8:1 before the fix, ranged from 5.3:1 to 9.2:1 after the colors were readjusted (the WCAG requirement is 4.5:1). These errors were not visible at the initial stage because the relevant elements had not yet been activated at the time of the first scan.

As a result, the Lighthouse score rose to 95, Axe-core recorded zero errors, and full keyboard accessibility was achieved for all elements.

Discussion

The study once again confirmed that automated tools are effective at detecting static code errors but are unable to evaluate the actual user experience. Although Lighthouse gave an initial score of 68, this score was based solely on code structure. This outcome reflects how automated tools function: scanning algorithms rely on clean HTML and JavaScript syntax as a page loads, rather than on its semantic meaning. Correct usage of the project's core tags, compliance of the browser code structure with standards, and the presence of technical meta tags were sufficient for the program to assign a relatively high score. In other words, the initial score was not zero precisely because these tools read the text and code structure only in the format they are designed to understand.

This confirms the "false sense of security" noted by Brajnik [12] and Vigo [13]. It further demonstrates that relying entirely on automated tools is a mistake [14].

The interactive and semantic shortcomings identified through manual testing showed that automated tools check only static code and cannot evaluate the user experience. Once these issues were resolved, the Lighthouse score rose to 95. This confirms that it is essential not to rely on automated tools alone, but to complement them with manual testing — which is precisely the core idea behind the hybrid approach.

The contrast example further supports this: the initial ratio did not meet the WCAG AA requirement (4.5:1), yet the static scanner did not flag it as an error. After the components were re-checked, the ratio rose to a range of 5.3:1 to 9.2:1. This shows that only manual testing can detect errors that occur in dynamic states. From this standpoint, the hybrid audit model appears to be the most reliable method for a complete assessment of web accessibility [15].

The study has a number of limitations. Because testing was conducted on localhost, it may not fully reflect a real user environment. Likewise, since manual testing was performed by a single tester, this may increase the subjectivity of the results. The fact that testing was not conducted using screen-reader software is another factor that limited the scope of the study. Future work involving screen-reader software, the participation of multiple experts, and evaluation of other modules of the platform would help address these limitations.

Conclusion

The findings of this study demonstrate the importance of a hybrid approach — combining automated tools with manual testing — for evaluating web accessibility in digital learning platforms. While automated tools are effective at detecting static code errors, they cannot verify the real-time functional performance of interactive elements. Manual keyboard testing uncovered five issues that the automated tools had failed to detect, and these were subsequently resolved. As a result, the Lighthouse score rose from 68 to 95, and Axe-core recorded zero errors Based on these findings, web developers and IT specialists should not treat automated auditing as a final check, but should instead carry it out in conjunction with manual navigation testing. Educational institutions and responsible organizations should rely not only on automated scores but also on testing that reflects genuine user experience when evaluating digital learning environments. For future research, it would be worthwhile to extend the hybrid audit approach to include screen-reader software and testing involving multiple experts.

References

[1] D. Sloan, P. Gregor, P. Booth, and L. Gibson, "Auditing accessibility of UK higher education web sites," *Interacting with Computers*, vol. 14, no. 4, pp. 313–325, 2002.

[2] N. C. Burbules and C. A. Torres, *Globalization and Education: Critical Perspectives*. New York, NY, USA: Routledge, 2000.

[3] World Wide Web Consortium (W3C), "Website Accessibility Conformance Evaluation Methodology (WCAG-EM) 1.0," 2014. [Online]. Available: W3C WCAG-EM 1.0

[4] J. Abascal, M. Arrue, I. Fajardo, N. Garay, and J. Tomás, "The use of guidelines to automatically verify Web accessibility," *Universal Access in the Information Society*, vol. 3, no. 1, pp. 71–79, 2004.

[5] G. Brajnik, "Beyond conformance: The role of accessibility evaluation methods," in *Web Information Systems Engineering—WISE 2008 Workshops*, S. Hartmann, X. Zhou, and M. Kirchberg, Eds., Lecture Notes in Computer Science, vol. 5176. Berlin, Germany: Springer, 2008, pp. 63–80.

[6] M. Vigo, J. Brown, and V. Conway, "Benchmarking web accessibility evaluation tools: Measuring the harm of sole reliance on automated tests," in *Proc. 10th Int. Cross-Disciplinary Conf. Web*

Accessibility (W4A), 2013, Art. no. 1, pp. 1–10.

[7] President of the Republic of Uzbekistan, “On approving the ‘Digital Uzbekistan–2030’ strategy and measures for its effective implementation,” Decree No. PF-6079, 2020. [Online]. Available: LexUZ

[8] WebAIM, “The WebAIM Million,” 2024. [Online]. Available: WebAIM Million

[9] World Wide Web Consortium (W3C), “Web Content Accessibility Guidelines (WCAG) 2.2,” 2023. [Online]. Available: WCAG 2.2

[10] International Organization for Standardization, *ISO 9241-171:2021—Ergonomics of Human-System Interaction—Part 171: Guidance on Software Accessibility*. Geneva, Switzerland: ISO, 2021. [Online]. Available: ISO 9241-171

[11] N. N. Atabayeva, “Improving inclusive education based on digital technologies: Modern approaches and practical methods,” 2025. [Online]. Available: CyberLeninka

[12] C. Power, A. Freire, H. Petrie, and D. Swallow, “Guidelines are only guidelines: Why WCAG 2.0 is not sufficient,” in *Proc. SIGCHI Conf. Human Factors in Computing Systems (CHI ’12)*, 2012.

[13] A. Hambley, Y. Yesilada, M. Vigo, and S. Harper, “OPTIMAL-EM: Complexity-driven clustering for optimised web accessibility evaluation,” *ACM Transactions on the Web*, 2026.

[14] O. Koller, O. Zargaran, H. Ney, and R. Bowden, “Deep sign: Hybrid CNN-HMM for continuous sign language recognition,” in Proc. Brit. Mach. Vis. Conf. (BMVC), 2016.

[15] A. M. Sincan and H. Y. Keles, “AUTSL: A large scale multi-modal Turkish sign language dataset and baseline methods,” IEEE Access, vol. 8, pp. 181340–181355, 2020.