RESEARCH IN PROGRESS

EVALUATING THE FEASIBILITY OF LLM-BASED AUTOMATION OF MANUAL WCAG COMPLIANCE TESTING

ANNEMAE VAN DE HOEF, KOEN SMIT, SAM LEEWIS, DUAIN CASTRO, FABIAN HARTMAN, JULIANA TODOROVA, NICK KUIPER

HU University of Applied Science Utrecht, Utrecht, the Netherlands
annemae.vandehoef@hu.nl, koen.smit@hu.nl, sam.leewis@hu.nl, duain.castro@hu.nl, fabian.hartman@hu.nl, juliana.todorova@hu.nl, nick.kuiper@hu.nl

Digital inclusion requires that websites and online services are accessible to people with disabilities, yet verifying compliance with the Web Content Accessibility Guidelines (WCAG) remains resource-intensive. Many success criteria require manual auditing, as conventional automated tools detect only a limited subset of violations. This study evaluates the feasibility of using a fine-tuned enterprise large language model (LLM) to (semi-)automate such manual WCAG checks. We developed a Chrome extension that operationalizes selected criteria through structured prompting and tested it against prior expert audits of two benchmark pages. Results show moderate ability to localise problematic code, but weak performance in correctly assigning WCAG criteria and generating reliable improvenent suggestions. While the proof-of-concept demonstrates potential as a audit support tool, its probabilistic outputs currently lack the stability and accuracy required to properly support human accessibility auditing.

DOI
https://doi.org/10.18690/um.fov.4.2026.67

ISBN
978-961-299-152-4

Keywords:
large language models, WCAG, compliance testing, digital accessibility, research in progress

University of Maribor Press

1 Introduction

Web accessibility is essential for ensuring that digital content is usable by people with disabilities, which is an important aspect of a digital inclusive society. Governments and organizations worldwide mandate conformance to the Web Content Accessibility Guidelines (WCAG) (e.g. AA level for most regulations) (see e.g. (European Commission, 2025)). However, achieving and verifying WCAG compliance remains challenging. Full manual auditing by experts (often including people with lived disability experience) is time-consuming, costly, and requires specialized skills. For example, a comprehensive accessibility audit of a website can take hours or days, depending on complexity (López-Gil & Pereira, 2025; Smit et al., 2024). Human auditors may suffer fatigue or overlook issues during repetitive testing of dozens of pages and criteria, as well as the variability across raters when applying WCAG success criteria (Iniesto & Rodrigo, 2024). In contrast, automated tools can scan many pages quickly and consistently, but they only cover a subset of WCAG requirements. However, prior empirical data indicates that conventional scanners reliably detect perhaps only 15–30% of WCAG criteria, while the rest require human judgment (Fischer et al., 2025; Heine, 2025) and coverage was and is still narrow on certain success criteria (Vigo et al., 2013). The risk is, that in practice, a purely automated scan might report a perfect score, whereas a human audit uncovers many issues. Combining both is reported to be favorable.

Given these limitations, human auditors remain the gold standard. Humans can interpret semantics (e.g., whether alt text is meaningful) and assess usability beyond code patterns. But relying solely on experts is expensive and not scalable. To reduce this burden, researchers have investigated hybrid approaches. In particular, recent advances in large language models (LLMs, e.g., GPT-4) suggest a new avenue. LLMs, with their ability to parse and reason about text and code, could help (semi-)automate WCAG checks that currently require manual review. LLMs trained on vast web corpora have demonstrated surprising abilities on diverse tasks, and some studies have begun to apply them to accessibility. For example, López-Gil and Pereira (2025) used an LLM to evaluate three WCAG success criteria (Non-text Content 1.1.1, Link Purpose 2.4.4, Language of Parts 3.1.2) and found that their LLM-based scripts identified issues that traditional tools missed, achieving 87.2% detection across test cases. Similarly, Hinderks et al. (2025) compared GPT-4o to Lighthouse and reported that, with tailored prompts, the LLM found 88% of

induced violations on representative WCAG 2.1 criteria versus only 24% by Lighthouse. Andruccioli et al. (2025) showed that an LLM could reason about dynamically generated HTML and flag accessibility violations beyond the reach of static analyzers, though it also produced false positives ("hallucinated" issues). These early findings suggest LLMs could augment automated testing by bringing semantic understanding to the task (Bassi et al., 2025). To the knowledge of the authors, only the few selected studies identified in this paper focus on applying LLMs for accessibility testing.

However, LLMs are inherently probabilistic and not designed as validators. Unlike rule-based scanners, their outputs can vary with wording and context, and they may "hallucinate" or give inconsistent answers (Li et al., 2025). For example, AI-driven scans might incorrectly flag non-issues or miss subtle errors, requiring extra scrutiny. The question remains as follows: To what extent can a fine-tuned large language model reliably automate manual WCAG 2.x (AA-level) conformance checks? To answer this question, we conducted a feasibility study, for which we have developed a proof-of-concept (PoC) using a fine-tuned LLM to automate WCAG 2.x (primarily AA-level) checks.

2 Background and Related Work

Research on WCAG conformance testing can broadly be grouped into three methodological streams: 1) deterministic rule-based automation, 2) hybrid or AI-augmented extensions of static analysis, and 3) more recent large language model (LLM)-driven evaluative approaches. These streams differ not only in technical implementation but also in their assumptions about what constitutes testable accessibility knowledge.

The first stream consists of rule-based automated tools that operationalize WCAG success criteria as formalized code checks. Applications such as AXE and WAVE analyze the Document Object Model (DOM) and apply predefined rules to detect structural violations (Deque, 2026; WEBAIM, 2026). Their strength lies in determinism and reproducibility, given identical input, the output remains stable and predictable. However, this approach is inherently constrained to criteria that can be expressed as syntactic or programmatic rules. Thus, traditional automation is bounded less by computational power than by formal expressibility. A second

stream attempts to extend automation through hybrid or AI-supported methods. These approaches incorporate machine learning, dynamic rendering analysis, or heuristic reasoning to address limitations of purely static inspection. For example, Andruccioli et al. (2025) demonstrate that model-based systems can analyze dynamically generated HTML and identify accessibility violations that static tools miss (Andruccioli et al., 2025). While such systems broaden technical coverage, they still typically operate within constrained detection paradigms and do not fully replicate the contextual reasoning characteristic of expert human auditors. The third and most recent stream introduces LLMs as evaluative agents in accessibility testing. Unlike rule-based systems, LLMs do not rely on explicitly encoded validation rules; instead, they perform probabilistic inference based on learned representations of language, structure, and context. Empirical studies suggest that LLMs can outperform traditional automated tools on semantically complex WCAG criteria, such as evaluating the meaningfulness of alternative text or the contextual clarity of link purpose (Hinderks et al., 2025; López-Gil & Pereira, 2025). This positions LLMs as potentially capable of automating checks previously categorized as "manual."

However, this paradigm shift introduces methodological and maybe even ethical challenges. Because LLMs are probabilistic, their outputs may vary across runs, depend on prompt design, and occasionally generate hallucinated or unjustified findings (Andruccioli et al., 2025). In compliance-driven domains such as accessibility auditing, where reproducibility, traceability, and defensibility are essential, such variability raises concerns about reliability (Andruccioli et al., 2025; The American Foundation for the Blind, 2025). Consequently, the key research gap is not merely whether LLMs can detect violations, but whether fine-tuned enterprise-scale LLMs can do so with stability and accuracy comparable to human auditors under controlled conditions.

3 Research Method and Preliminary Results

This study adopts a design science approach (Hevner et al., 2004) to develop and evaluate a PoC Chrome extension that automates parts of manual WCAG 2.2 (AA) conformance checking using a locally deployed enterprise LLM. The extension scans a web page, extracts relevant DOM and textual context, and queries the model with criterion-specific prompts. A locally hosted version of Copilot was used to meet institutional security and privacy requirements. Implementation details and prompt

templates are documented in the project repository (Crasto et al., 2026), see Figure 1 for examples of the PoC after the first iteration.

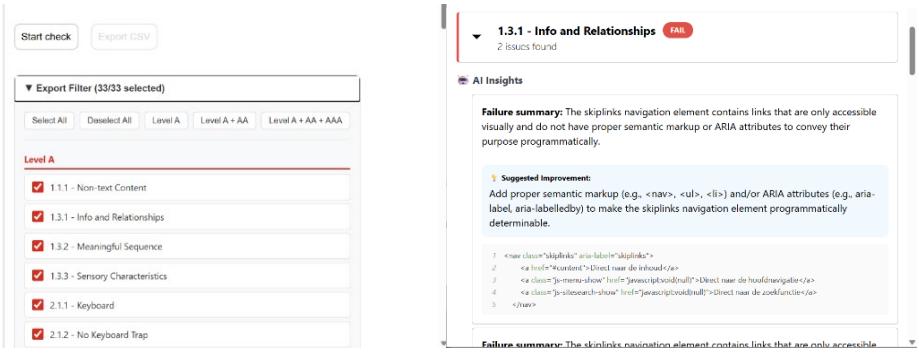

Figure 1: Filter before starting test (left) and results after test (right)

The PoC targets success criteria that typically require manual auditing beyond rule-based tools (Vigo et al., 2013), listed in our dataset (Crasto et al., 2026). Each selected WCAG criterion was translated into a structured prompt containing: (1) a concise guideline description and reference, (2) a task specification, (3) common failure patterns, and (4) explicit pass/fail conditions (White et al., 2023). The model was required to return the relevant DOM location, the WCAG reference, a justification grounded in the guideline, and a suggestion on how to improve the code. Temperature settings were calibrated per analytical category to reduce output variability (lower for structural checks; slightly higher for context-sensitive checks) (Brown et al., 2020). Evaluation used two organizational web pages from the HU University of Applied Sciences that had previously undergone comprehensive manual WCAG audits, which served as the reference standard. For each page, the extension performed a full scan and generated criterion-specific findings. Outputs were evaluated along two dimensions: (1) correct classification of issues (location and WCAG success criterion) and (2) the quality of justification and suggested improvements. For each manually identified issue, it was assessed whether the LLM detected the same issue. Where outputs contained code, matching was performed using normalized HTML snippets to remove variable elements. Exact matches indicated correct localization, while partial matches were manually reviewed for semantic equivalence. If code matched, the corresponding WCAG criterion was verified. For results without explicit code, equivalence was assessed at the guideline

level based on substantive agreement. For correctly classified issues, the LLM's justification was evaluated against the relevant WCAG criterion, followed, if correct, by validation of the suggested improvement. Only criteria for which the manual audit identified at least one issue were included in performance calculations. Precision, recall, and F1-score were computed at criterion level (Sokolova & Lapalme, 2009). For findings where both location and criterion matched the manual audit, we additionally assessed the correctness of the model's justification and improvement advice.

Preliminary results

Across both benchmark pages, the PoC showed moderate ability to localise problematic code but substantially weaker performance in assigning the correct WCAG criterion, see Table 1.

Table 1: Testing results for both pages included in the first iteration of testing the PoC

Performance Dimension	Metric	Page 1 (ÉÉN HU)	Page 2 (HU Voltijdopleidingen)
A. Localisation Accuracy	% Correct Faulty DOM Locations Identified	57.8	60.4
B. Criterion Classification	% Correct WCAG Criterion Assignment	20.3	17.9
	% Precision	4.5	2.4
	% Recall	8.0	17.9
	% F1-score	5.7	4.2
C. Justification Quality*	# Fully Correct Justifications	3	89*
D. Improvement Quality*	# Fully Correct Improvement Suggestions	1	1

*Most correct justifications referred to the same recurring accessibility issue, namely a "favorite" button missing the required aria-checked attribute. Qualitative inspection suggests recurring mismatches between identified symptoms and the precise WCAG success criterion, as well as occasional hallucinated technical details. Overall, the PoC appears promising as a support tool for identifying potentially problematic code regions, but in its current configuration it does not yet provide sufficiently reliable criterion classification or remediation guidance to support or automate human auditing adequately.

4 Discussion and Future Research Directions

These preliminary findings suggest that LLM-based support for manual WCAG auditing is currently more promising for localising potentially problematic code than for making reliable compliance judgements. The PoC showed moderate localisation performance, but weak criterion assignment and limited remediation quality, indicating that the model may detect accessibility-relevant symptoms without consistently translating them into precise WCAG judgements. This aligns with earlier studies showing that LLMs can extend accessibility testing beyond rule-based tools, while still producing unstable or hallucinated results (Andruccioli et al., 2025; Hinderks et al., 2025; López-Gil & Pereira, 2025). At this stage, the PoC therefore seems better suited as a triage aid for human auditors than as a means of automating manual conformance assessment. From an IS perspective, this also suggests that effective AI-supported accessibility auditing depends on socio-technical alignment between AI outputs, human judgement, and compliance governance (Ahmi & Smidt, 2026). The study is limited by the small benchmark set, the time gap between manual and automated audits, and the use of prompting rather than fine-tuning. In addition, prompt calibration could not be performed due to the extensive processing time required for the LLM to handle a single page, combined with the need for subsequent manual validation. Future research should therefore test larger and concurrent datasets, separate localisation from criterion classification, explore criterion-specific or fine-tuned models, and examine whether multi-pass prompting and human-in-the-loop review improve reliability and practical usefulness.

References

Ahmi, A., & Smidt, L. (2026). AI-driven auditing: Trends, themes, and research trajectories. *EDPACS*, 1–20. https://doi.org/10.1080/07366981.2026.2637772

Andruccioli, M., Bassi, B., Delnevo, G., & Salomoni, P. (2025). Leveraging Large Language Models for Sustainable and Inclusive Web Accessibility. *Big Data and Cognitive Computing*, *9*(10), 247. https://doi.org/10.3390/bdcc9100247

Bassi, B., Delnevo, G., Franco, M., Gaggi, O., Gatto, S., Mirri, S., & Olaiya, K. (2025). An Assessment of LLM-Based Auditing and Validation for Web Accessibility. *Proceedings of the 2025 International Conference on Information Technology for Social Good*, 297–305. https://doi.org/10.1145/3748699.3749805

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., … Amodei, D. (2020). *Language Models are Few-Shot Learners* (arXiv:2005.14165). arXiv. https://doi.org/10.48550/arXiv.2005.14165

Crasto, D., Kuiper, N., Hartman, F., & Todorova, J. (2026). *WCAG Tool* [Computer software]. https://github.com/wp7HUDT/AccessibilityCheck

Deque. (2026). The Axe Platform provides all your accessibility testing and monitoring tools in one place. *Deque*. https://www.deque.com/axe/

European Commission. (2025). *Web Accessibility Directive—Standards and harmonisation*. European Commission. https://digital-strategy.ec.europa.eu/en/policies/web-accessibility-directive-standards-and-harmonisation

Fischer, T., Lundell, B., & Gamalielsson, J. (2025). Coverage of web accessibility guidelines provided by automated checking tools. *Universal Access in the Information Society*, *24*(4), 3615–3637. https://doi.org/10.1007/s10209-025-01263-x

Heine, M. (2025, February 28). *Why Automated Accessibility Testing Isn't Enough*. B13. https://b13.com/blog/why-automated-accessibility-testing-isnt-enough

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design Science in Information Systems Research. *MIS Quarterly*, *28*(1), 75–106. https://doi.org/10.2307/25148625

Hinderks, A., Davtyan, A., & Allerkamp, D. (2025). Automated accessibility testing with Large Language Models: A comparative analysis of GPT-40 and traditional tools for WCAG 2.1 conformance testing. *Mensch Und Computer 2025 - Usability Professionals*. Gesellschaft für Informatik eV.

Iniesto, F., & Rodrigo, C. (2024). The use of WCAG and automatic tools by computer science students: A case study evaluating MOOC accessibility. *JUCS - Journal of Universal Computer Science*, *30*(1), 85–105. https://doi.org/10.3897/jucs.101704

Li, Q., Leyang, C., Lingpeng, K., & Wei Bi. (2025). Exploring the Reliability of Large Language Models as Customized Evaluators for Diverse NLP Tasks. *Proceedings of the 31st International Conference on Computational Linguistics*, 10325–10344.

López-Gil, J.-M., & Pereira, J. (2025). Turning manual web accessibility success criteria into automatic: An LLM-based approach. *Universal Access in the Information Society*, *24*(1), 837–852. https://doi.org/10.1007/s10209-024-01108-z

Smit, K., Van De Hoef, A., Kuiper, N., Todorova, J., Hartman, F., Leewis, S., & Gevaert, R. (2024). Digital Accessibility in Higher Education: A Dutch case study. *Proceedings of the 2024 8th International Conference on Software and E-Business*, 76–86. https://doi.org/10.1145/3715885.3715890

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, *45*(4), 427–437. https://doi.org/10.1016/j.ipm.2009.03.002

The American Foundation for the Blind. (2025). *Missing a Human Touch: Why Automated Testing and AI Don't "Solve" Digital Accessibility… Yet*. The American Foundation for the Blind. https://afb.org/blog/entry/missing-human-touch-1

Vigo, M., Brown, J., & Conway, V. (2013). Benchmarking web accessibility evaluation tools: Measuring the harm of sole reliance on automated tests. *Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility*, 1–10. https://doi.org/10.1145/2461121.2461124

WEBAIM. (2026). *WAVE Web Accessibility Evaluation Tools*. WAVE. https://wave.webaim.org/

White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., Elnashar, A., Spencer-Smith, J., & Schmidt, D. C. (2023). *A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT* (arXiv:2302.11382). arXiv. https://doi.org/10.48550/arXiv.2302.11382