Large Language Models for Web Accessibility: A Systematic Literature Review

Wajdi Aljedaani
Saudi Data and AI Authority
Riyadh, Saudi Arabia
Department of Software Engineering
Rochester Institute of Technology
Rochester, New York, USA
wajdi.j1@gmail.com

Rubel Hassan Mollik
University of North Texas
Denton, Texas, USA
RubelHassanMollik@my.unt.edu

Abstract

Web accessibility aims to ensure that web content and services are usable by people with diverse abilities. In recent years, Large Language Models (LLMs) have been increasingly explored to support accessibility-related tasks on the web, such as content generation, issue detection, and remediation. However, little is known about the characteristics of these approaches, the accessibility issues they target, the standards they follow, and how they are evaluated. In this paper, we present a systematic literature review of 38 peer-reviewed studies that investigate the use of LLMs in web accessibility contexts. We begin by performing a comprehensive search of scientific publications to identify relevant studies. We then conduct a comparative analysis to examine the accessibility tasks addressed, the LLM models and prompting strategies employed, the system architectures adopted, the accessibility issues and guidelines considered, and the evaluation methods used across studies. Our findings show that most studies apply LLMs to text-centric and structurally explicit accessibility tasks, with WCAG serving as the primary reference framework and limited consideration of cognitive accessibility guidelines (COGA). The reviewed approaches predominantly rely on general-purpose LLMs and prompt-based interactions, while evaluation practices vary widely and often lack direct involvement of users with disabilities. We envision this review as a consolidated reference for researchers and practitioners seeking to understand the current landscape of LLM-supported web accessibility, and as a foundation to guide future research and tool development in this area.

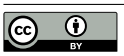
CCS Concepts

• **Human-centered computing** → **Accessibility**.

Keywords

Accessibility, Web Accessibility, Large Language Models, LLM, Accessibility, WCAG, Artificial Intelligence

ACM Reference Format:
Wajdi Aljedaani and Rubel Hassan Mollik. 2026. Large Language Models for Web Accessibility: A Systematic Literature Review. In *Web4All 2026 23rd International Web for All Conference (W4A '26), April 13–14, 2026, Dubai, United Arab Emirates.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3800424.3800452



W4A '26, Dubai, United Arab Emirates

ACM ISBN 979-8-4007-2372-8/26/04
https://doi.org/10.1145/3800424.3800452

1 Introduction

The rapid advancement of Large Language Models (LLMs) has led to their widespread adoption across web development and content creation workflows [8, 66]. These models are now routinely used to generate web code, assist with interface design, produce textual and visual descriptions, and support interactive web-based systems [27, 41, 42]. As LLMs increasingly shape how web content is authored and delivered, their influence extends beyond functionality and aesthetics to fundamental quality attributes of the web, including accessibility.

Web accessibility concerns the extent to which web content and services can be accessed and used by people with diverse abilities [39]. Although accessibility standards and best practices have matured over time, accessibility remains inconsistently implemented across the web [62]. Many websites continue to present barriers for users with disabilities [9, 32], particularly when accessibility considerations are deferred or treated as optional during development. Addressing accessibility often requires specialized expertise, careful evaluation, and sustained maintenance, which can be perceived as time-consuming or difficult to scale within fast-paced development environments [21].

In response to these challenges, researchers have begun investigating whether LLMs can be leveraged to support accessibility-related tasks on the web. Existing studies explore a range of applications, including generating alternative text for images, detecting accessibility violations, assisting with the creation of accessible markup, explaining accessibility guidelines, and supporting users through conversational interfaces [27, 41, 42, 46]. These works demonstrate growing interest in using LLMs to reduce the effort associated with accessibility work. At the same time, reported findings vary widely, with some studies highlighting promising results [27] and others identifying substantial limitations related to accuracy, consistency, and reliability [8].

Despite a growing body of work, research on LLMs and web accessibility remains fragmented. Existing studies often focus on narrow problem settings, such as individual accessibility tasks [40, 44, 45], specific web technologies [19, 24, 27], or isolated evaluation techniques [12, 31, 52]Moreover, accessibility is operationalized inconsistently across the literature, with varying standards, metrics, and validation practices [2, 58, 60]. This fragmentation makes it difficult to develop a cohesive understanding of how LLMs, as

a broader class of systems, contribute to—or potentially undermine—web accessibility beyond isolated tool-level analyses. At the same time, web accessibility poses unique challenges for LLM-based systems, distinguishing it from other application domains. Accessibility-related outputs must adhere to established standards, address diverse user needs, and avoid generating misleading or superficial guidance [23, 49]. Failures such as hallucinated compliance claims or incomplete fixes can have direct and disproportionate consequences for users with disabilities, underscoring the need for a systematic examination of how LLMs are currently evaluated and applied in accessibility-focused web contexts [11, 25, 29, 33].

To address these gaps, we conduct a Systematic Literature Review (SLR) of studies that investigate the role of LLMs in web accessibility. We analyze 33 primary studies spanning accessibility tasks, web technologies, evaluation methods, and application contexts. Our synthesis identifies dominant research trends, commonly reported challenges, and methodological gaps in current LLM-supported accessibility work. This review establishes how LLMs are currently used to support web accessibility, where their limitations persist, and outlines directions for future research toward more reliable and inclusive web experiences.

1.1 Goal & Research Questions

The goal of this study is to provide researchers, practitioners, and accessibility stakeholders with a comprehensive and structured synthesis of existing research on the use of LLMs in web accessibility. By systematically reviewing and analyzing prior work, this study consolidates fragmented evidence on how LLMs are applied to support accessibility-related web tasks, the standards and disability groups they address, and the methods used to evaluate their effectiveness. The findings aim to inform decision-making, highlight methodological and technical gaps, and guide future efforts toward more reliable, inclusive, and standards-aligned LLM-supported web accessibility solutions. To achieve these objectives, we formulate the following research questions (RQs):

RQ_1: What are LLMs being applied to in support of web accessibility?

This research question examines the types of accessibility tasks addressed by LLM-based approaches, such as detection, remediation, content generation, and user support, and characterizes how these models are integrated into web accessibility workflows.

RQ_2: What LLM models, prompting strategies, and system architectures are used in web accessibility research?

This research question focuses on the technical design choices made in prior work, including the selection of LLMs, the use of prompting techniques, and the adoption of architectural patterns. By answering this RQ, we provide insight into prevailing design practices and model diversity within the literature.

RQ_3: What web accessibility issues and WCAG success-criterion violations are most frequently addressed?

This RQ investigates the types of accessibility issues targeted by existing studies and how LLMs are applied to detect, remediate, or generate accessible content for these issues. The findings highlight dominant problem areas and less explored accessibility concerns.

RQ_4: How are LLM-based web accessibility approaches evaluated, and what evidence is used to assess their effectiveness?

This research question examines the evaluation strategies employed across studies, including automated assessments, expert reviews, user studies, and comparative analyses. Addressing this RQ provides insight into the rigor, consistency, and limitations of current evaluation practices.

1.2 Contributions

- A systematic literature review of 33 peer-reviewed studies examining how LLMs are applied to support web accessibility.
- A structured characterization of LLM-based accessibility approaches, including the tasks addressed, solution types, models, and interaction strategies used across studies.
- An analysis of accessibility standards, disability coverage, and reported issues, highlighting dominant trends and gaps in current research.
- A synthesis of evaluation practices and open challenges, offering insights to guide future research toward more reliable and inclusive LLM-supported web accessibility solutions.
- A replication package of our survey for extension purposes [1].

The remainder of this paper is organized as follows. Section 2 reviews prior systematic literature reviews and related work in web accessibility and human–computer interaction, and identifies gaps addressed by this study. Section 3 describes the research methodology, detailing the planning, execution, and synthesis phases of the systematic literature review. Section 4 presents the research findings, structured around the five research questions and summarizing observed patterns in LLM applications, accessibility issues, and evaluation practices. Section 5 discusses the results by synthesizing key themes and challenges identified across the literature. Section 6 outlines the implications of our findings for both research and practice. Section 7 discusses threats to validity and the mitigation strategies employed. Finally, Section 8 concludes the paper by summarizing the main insights and outlining directions for future work.

2 Related Work

A substantial body of work in the HCI and accessibility communities has employed systematic literature reviews to examine web accessibility practices, evaluation methods, and guideline adoption. Several SLRs focus on web accessibility evaluation tools and methods [13, 17, 18, 48], analyzing how automated, semi-automated, and manual approaches are used to assess compliance with WCAG and related standards [61]. These studies consistently report limitations of automated tools in capturing contextual, semantic, and user-experience-dependent accessibility issues, emphasizing the need for complementary approaches. Other SLRs investigate accessibility integration within the software development lifecycle [22], highlighting that accessibility is often addressed late in development and inconsistently applied across design [56], requirements [59],

[1] Will be shared upon acceptance

and testing phases [55]. Collectively, these reviews provide valuable insights into accessibility challenges, tooling limitations, and methodological gaps, however, they predate or only tangentially consider the role of modern generative AI systems.

More recent SLR extend this line of inquiry by examining AI-driven approaches for accessibility, including computer vision, speech technologies, and adaptive interfaces [6]. However, this review typically treat AI as a broad category and does not isolate Large Language Models (LLMs) as a distinct class of systems with unique capabilities and risks. In parallel, SLRs focusing on cognitive accessibility and inclusive design emphasize that COGA-related concerns—such as content complexity, cognitive load, and clarity—remain underrepresented in both research and tooling [14]. While these works underscore important gaps in accessibility coverage and evaluation rigor, none provide a systematic synthesis of how LLMs are applied to web accessibility tasks, how their outputs align with WCAG [16, 63] and COGA guidelines [1], or how LLM-based approaches are evaluated in practice. This gap motivates our systematic literature review, which consolidates evidence across 38 studies to map the landscape of LLM-supported web accessibility research, identify dominant patterns, and highlight underexplored areas within the HCI and accessibility domains.

3 Research Methodology

As a Systematic Literature Review (SLR), this study systematically reviews and synthesizes the existing scientific literature to provide a structured, comprehensive understanding of how Large Language Models (LLMs) [47] are applied in web accessibility [15]. SLRs [37] are widely used in software engineering and human–computer interaction research to consolidate fragmented evidence, identify research trends, and highlight gaps in emerging domains. In this work, we aim to analyze prior studies on the use of LLMs to support, evaluate, or enhance web accessibility and derive insights to inform future research and practice.

While prior reviews have explored LLMs in broader software engineering contexts or examined accessibility independently, this study focuses specifically on the intersection of LLMs and web accessibility. Accordingly, this section describes the methodology we adopted to identify, select, and analyze relevant publications. Our review process consists of three main phases: (1) planning, (2) execution, and (3) synthesis. Each phase is described in detail in the following subsections.

3.1 Planning

The planning phase defines the scope of the review and establishes a systematic strategy for identifying relevant studies. This phase includes selecting digital libraries, defining inclusion and exclusion criteria, and constructing search keywords.

Digital Libraries. To locate publications for our study, we searched six digital libraries, including ACM Digital Library[2], IEEE Xplore[3], Web of Science[4], Scopus[5], Science Direct[6], and Springer Link[7]. These librarieseither contain or index peer-reviewed publications from computer science, software engineering, and human–computer interaction venues, and have been commonly utilized in prior systematic literature reviews (e.g., [32]).

Search Keywords. To identify an appropriate set of search keywords, we conducted a pilot search across two widely used digital libraries: IEEE Xplore and the ACM Digital Library. This process was used to identify commonly used terms and synonyms across studies on Large Language Models and web accessibility. The search query was refined through multiple iterations to improve relevance and coverage. To reduce false positives, we applied the finalized search query only to publication titles and abstracts, rather than the full text. The finalized search string used across all selected digital libraries is presented below.

```
("web accessib*" OR "website* accessib*" OR "web content accessib*" OR "digital accessib*" OR "online accessib*" OR "internet accessib*" OR "accessible web" OR "accessible website*" OR "accessible web page*" OR "accessible web design" OR "web content accessibility guidelines" OR "wcag" OR "wai-aria" OR "a11y" ) AND ("large language model" OR "large language models" OR "llm" OR "llms" OR "llm-based" OR "foundation model" OR "foundation models" OR "chatgpt" OR "gpt" OR "generative ai" OR "generative artificial intelligence" OR "vision-language model" OR "multimodal model")
```

Inclusion/Exclusion Criteria. Inclusion and exclusion criteria [53] are essential for pruning the search space, reducing selection bias, and ensuring the retrieval of relevant, peer-reviewed scientific publications. Publications that meet these criteria serve as the starting point for manual filtering, enabling us to assess whether they align with the scope of this study, namely, the investigation of Large Language Models in web accessibility contexts. The resulting pool of publications also serves as the basis for backward snowballing [35], which involves examining the references cited by the selected studies, and forward snowballing [28], which involves identifying studies that cite the selected publications. Table 1 summarizes the inclusion and exclusion criteria applied in this review.

Regarding the time range, we did not impose a starting date to capture early and foundational work in this emerging area. The end date for the search was set to November 30, 2025. Accordingly, the time range criterion allows the inclusion of any study published on or before this date, provided it satisfies the defined inclusion criteria.

Table 1: Our inclusion and exclusion search criteria.

Inclusion	Exclusion
Peer-reviewed publication	Grey literature (e.g., websites, reports)
Written in English	Non–peer-reviewed work
Addresses LLMs in web accessibility	LLM studies without accessibility focus
Focuses on web-based accessibility tasks	Accessibility outside web context
Available in digital format	Full-text not available online

[2]https://dl.acm.org/
[3]https://ieeexplore.ieee.org/
[4]https://webofknowledge.com/
[5]https://www.scopus.com/
[6]https://www.sciencedirect.com/
[7]https://www.sciencedirect.com/

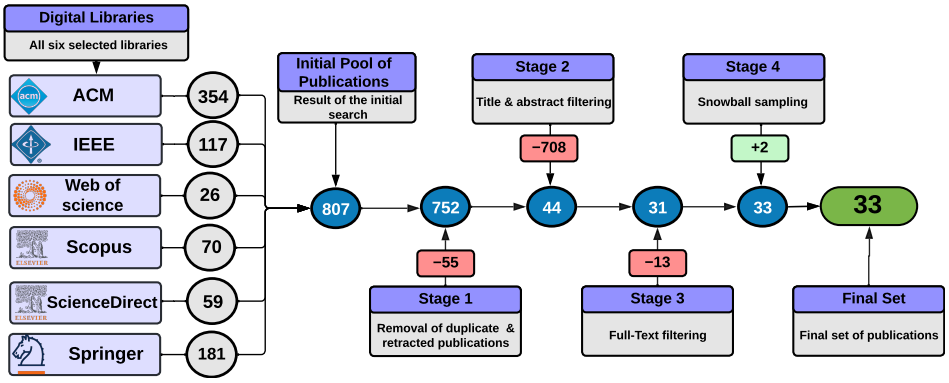


Figure 1: Overview of the volume of publications resulting from our filtering process.

3.2 Execution

The execution phase consists of conducting the literature search and applying a structured, multi-stage filtering process to identify the final set of primary studies. Figure 1 provides an overview of this process and illustrates the number of publications retained at each stage.

In Stage 1, we executed the finalized search query across the six selected digital libraries, yielding an initial pool of 807 publications. To ensure rigor and consistency, all filtering stages were performed manually by multiple authors, with disagreements resolved through discussion until consensus was reached. We then removed duplicate records and retracted publications. This step reduced the initial pool from 807 to 752 candidate publications. In Stage 2, we applied the inclusion and exclusion criteria to the remaining publications' titles and abstracts. Studies that were clearly out of scope, for example, those that did not involve web accessibility or did not consider Large Language Models, were excluded at this stage. After title and abstract screening, 44 publications were retained for further analysis. In Stage 3, we conducted a full-text review of the remaining publications to assess their relevance in greater detail. During this stage, we excluded studies that did not substantively address LLMs in a web accessibility context or failed to meet the inclusion criteria upon closer inspection. This filtering step resulted in 31 publications. In Stage 4, we performed backward and forward snowball sampling on the selected studies by examining their reference lists and identifying papers that cited them. This process led to the identification of 2 additional relevant publications. After completing all filtering stages, we obtained a final set of 33 primary studies, which form the basis of the analysis presented in this review.

3.3 Synthesis

This phase synthesizes the extracted data to answer RQ_1–RQ_5. First, we classified the primary set of publications based on how Large Language Models are applied in web accessibility, distinguishing between studies that propose new LLM-based tools or systems and those that adopt or evaluate LLMs within existing accessibility workflows.

For RQ_1–RQ_3, we manually reviewed the full text of each publication and extracted concrete evidence regarding the accessibility tasks addressed, the use of accessibility standards and guidelines (e.g., WCAG and COGA), the types of accessibility issues targeted, and the technical design choices, including LLM models, prompting strategies, and system architectures. For RQ_4, we synthesized evaluation practices into five distinct categories based on the methods used and the involvement of participants in the reviewed studies.

All RQ-related data collected during the synthesis process were reviewed by two authors to mitigate bias, with disagreements resolved through discussion. The review was conducted by authors with prior research experience in web accessibility, supporting consistent interpretation during the selection and synthesis processes. A shared spreadsheet was used to document the extracted data and facilitate collaboration.

4 Research Findings

This section presents the results of the review. The findings are organized by research question and summarize observed patterns in LLM applications, accessibility issues, and evaluation practices.

RQ_1: What are LLMs being applied to in support of web accessibility?

We found seven application types through which Large Language Models are used to support web accessibility in the reviewed studies as shown in Table 2. To provide a clearer analytical distinction, we categorize these applications by both their primary functional tasks and their method of technological integration into workflows. The results show that LLMs are most frequently applied as automated tools and through prompt-based strategies, each reported in 10 studies. These applications typically position LLMs as assistive components categorized by their functional role: development tasks focus on generating accessible content [5, 20, 40, 45, 51, 54], or assisting with remediation workflows [7, 23, 27, 50, 60] while, evaluation tasks assist in identification of accessibility issues [2, 3, 11, 26, 31, 41, 52]. Regarding technical integration, browser extensions represent the next most common application type, appearing in 5 studies, reflecting efforts to integrate LLM-based accessibility support directly into development or browsing environments [5, 34, 54, 57, 65].

Less frequently, studies apply LLMs within hybrid approaches and LLM agent–based systems, each reported in 3 studies, indicating limited adoption of more complex or autonomous architectures [10, 27, 41]. Only one study frames LLMs as part of an

Table 2: LLM integration methods in web accessibility.

Integration Method	Description	PS#
Automated Tool	LLMs integrated into programmatic frameworks for evaluation or remediation.	PS#2, PS#33, PS#26, PS#14, PS#15, PS#16, PS#30, PS#19, PS#23, PS#25
Prompt Strategy	Direct model interaction via specialized accessibility-oriented instructions.	PS#4, PS#5, PS#8, PS#9, PS#11, PS#12, PS#28, PS#17, PS#24, PS#6
Browser Extension	Integration of AI support directly into the user's browsing or development interface.	PS#7, PS#27, PS#29, PS#18, PS#22
Hybrid Workflow	Multi-stage systems combining LLMs with rule-based or traditional algorithms.	PS#1, PS#3, PS#31
LLM Agent	Autonomous or semi-autonomous multi-agent systems for complex task handling.	PS#10, PS#13, PS#21
Benchmark	Standalone infrastructure for systematic model evaluation and performance tracking.	PS#20
Application	Specialized, end-to-end standalone software for specific accessibility needs.	PS#32

explicit benchmark [12], and one study presents a standalone application [45], suggesting that relatively few works focus on systematic evaluation infrastructures or end-to-end accessibility systems. An examination of the study contexts in Table 7 shows that most applications are evaluated in controlled or task-specific settings, such as standardized prompting, limited datasets, or individual web pages, rather than in long-term or real-world deployment scenarios. This indicates that current applications primarily emphasize feasibility and task performance over sustained integration into production-level accessibility workflows.

RQ$_2$: What LLM models, prompting strategies, and system architectures are used in web accessibility research?

The results reveal a strong concentration around commercial, general-purpose LLMs, particularly GPT-based models. GPT-4 is the most frequently used model, appearing in 18 studies, followed by ChatGPT in 11 studies and Claude in 6 studies. Other models, including Gemini (5 studies), LLaMA (3 studies), DeepSeek (2 studies), and Copilot (2 studies), are used less frequently, while vision–language and multimodal models such as LLaVA, VILA, Phi, MiniGPT appear only sporadically. This distribution indicates that most web accessibility research prioritizes readily available, high-capacity LLMs rather than models specifically trained or fine-tuned for accessibility-related tasks.

A closer examination of Table 7 shows that these models are primarily employed in comparative or head-to-head evaluation settings, where multiple LLMs are tested on the same accessibility task or dataset. In several studies, model performance is assessed relative to baseline tools, human experts, or alternative LLMs, suggesting that model choice is often treated as a key experimental variable. However, only a limited number of studies report systematic exploration of model configuration, fine-tuning, or training data characteristics, indicating that accessibility outcomes are largely evaluated at the level of out-of-the-box model behavior.

Figure 2 summarizes the prompting strategies adopted across the reviewed studies. The most common approaches are instruction-based prompting (10 studies) and role-based prompting (9 studies), followed by zero-shot, structured, and verbosity-controlled prompting (each reported in 6 studies). These strategies are typically used to guide LLMs toward producing accessibility-relevant outputs,

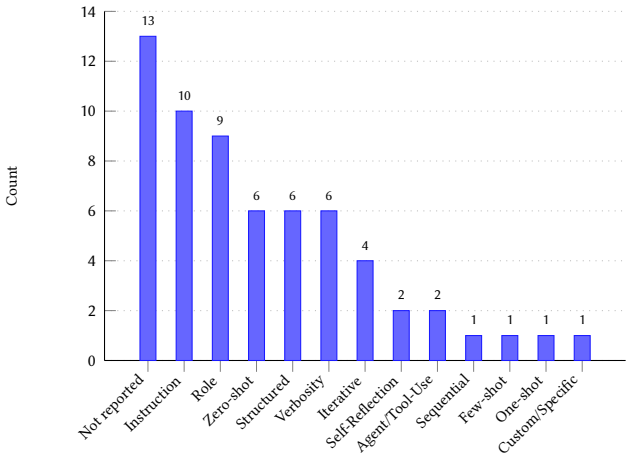


Figure 2: Counts of prompting techniques across the reviewed papers.

such as emphasizing compliance with WCAG success criteria or generating detailed descriptions. More advanced prompting techniques—such as self-reflection, agent-based prompting, ensemble prompting, and chain-of-thought-like strategies—are comparatively rare, appearing in only one or two studies each.

When considered alongside the system descriptions in Table 7, these findings indicate that most studies adopt relatively simple system architectures, in which a single LLM is invoked via carefully crafted prompts to perform specific accessibility tasks. Only a small subset of studies employ multi-agent systems [27], iterative feedback loops [30], or hybrid pipelines [52] that combine LLMs with rule-based tools or accessibility checkers. Overall, the results indicate that current research places greater emphasis on model selection and prompt design than on architectural innovation, with limited exploration of persistent, adaptive, or large-scale accessibility systems built around LLMs

Table 3: Distribution of accessibility guidelines across Primary Studies (PS)

Guideline	Referenced Primary Studies (PS)
WCAG 2.0	PS#3, PS#9, PS#12, PS#15, PS#16, PS#18, PS#21, PS#22, PS#23, PS#25, PS#26, PS#27, PS#28, PS#29
WCAG 2.1	PS#2, PS#5, PS#6, PS#7, PS#8, PS#10, PS#11, PS#19, PS#20, PS#24, PS#30, PS#32, PS#33
WCAG 2.2	PS#1, PS#2, PS#4, PS#17, PS#31
WAI-ARIA	PS#2, PS#27
Ofcom	PS#14, PS#16
EN 301 549	PS#30
Others	PS#1, PS#2, PS#14, PS#18, PS#30
Not Reported	PS#13

Table 4: Accessibility issues and violations identified across the reviewed primary studies.

Code	Issue / Violation	Primary Studies (PS)	Cnt	Code	Issue / Violation	Primary Studies (PS)	Cnt
WCAG SC 1.1.1	Non-text content missing or vague alternative text	PS#1, PS#2, PS#3, PS#6, PS#7, PS#8, PS#9, PS#10, PS#13, PS#15, PS#17, PS#18, PS#20, PS#21, PS#26, PS#28, PS#29, PS#30, PS#31, PS#32, PS#33	21	WCAG SC 1.2.2	Missing or insufficient captions (prerecorded)	PS#10	1
WCAG SC 1.3.1	Improper heading structure	PS#2, PS#4, PS#6, PS#7, PS#9, PS#20, PS#24, PS#26, PS#27, PS#30, PS#33	11	WCAG SC 1.3.1	Incorrect form labeling or association	PS#1, PS#3, PS#8, PS#11	4
WCAG SC 1.3.1	Semantic structure issues	PS#17, PS#20, PS#31	3	WCAG SC 1.4.1	Color used as the sole means of conveying information	PS#8	1
WCAG SC 1.4.3	Insufficient color contrast (minimum)	PS#1, PS#2, PS#3, PS#5, PS#6, PS#8, PS#10, PS#12, PS#13, PS#17, PS#19, PS#21, PS#26, PS#30, PS#31, PS#33	16	WCAG SC 1.4.4	Resize text not supported	PS#3, PS#8, PS#13	3
WCAG SC 1.4.6	Contrast (enhanced) not met	PS#3	1	WCAG SC 1.4.8	Visual presentation issues (brightness/-clutter)	PS#12	1
WCAG SC 1.4.12	Text spacing issues	PS#10	1	WCAG SC 2.1.1	Keyboard navigation / tabindex issues	PS#6, PS#8, PS#10, PS#11, PS#13, PS#17, PS#20, PS#24, PS#30, PS#31, PS#33	11
WCAG SC 2.4.1	Missing navigation aids (bypass blocks)	PS#7, PS#17, PS#27	3	WCAG SC 2.4.2	Missing or incorrect page titles	PS#3	1
WCAG SC 2.4.3	Incorrect focus order	PS#12	1	WCAG SC 2.4.4	Non-descriptive link text	PS#2, PS#3, PS#10, PS#33	4
WCAG SC 2.4.4	Ambiguous link purpose	PS#6, PS#21, PS#26	3	WCAG SC 2.4.6	Unclear or insufficient headings and labels	PS#1, PS#3, PS#8, PS#12	4
WCAG SC 2.4.7	Focus not visible	PS#8, PS#11	2	WCAG SC 2.5.3	Label not included in accessible name	PS#8	1
WCAG SC 2.5.5	Insufficient target size	PS#5, PS#8, PS#10	3	WCAG SC 3.1.1	Missing or incorrect page language	PS#3, PS#8, PS#21	3
WCAG SC 3.1.2	Language of parts not identified	PS#10	1	WCAG SC 3.1.5	Reading level issues (AAA)	PS#25	1
WCAG SC 3.2.4	Inconsistent identification	PS#18	1	WCAG SC 3.3.1	Error identification missing	PS#23	1
WCAG SC 3.3.2	Missing or insufficient labels or instructions	PS#1, PS#2, PS#3, PS#9, PS#10, PS#17, PS#20, PS#26, PS#28, PS#30, PS#33	11	WCAG SC 4.1.1	Parsing errors	PS#3, PS#8	2
WCAG SC 4.1.2	Inaccessible name, role, or value	PS#2, PS#3, PS#7, PS#8, PS#10, PS#12, PS#15, PS#17, PS#19, PS#28, PS#30, PS#32	12	WCAG SC 4.1.3	Missing or improper status messages	PS#8, PS#13, PS#19	3
WCAG 2.2	Adjacent links pointing to same destination	PS#1	1	WCAG SC 1.2.5	Missing audio description (prerecorded)	PS#12, PS#16, PS#29	3
COGA	Clearly identify controls and their use	PS#1	1	COGA	Lack of white space / long uninterrupted text	PS#1	1
COGA	Let users go back or undo actions	PS#1	1	COGA	Make each step clear	PS#1	1
COGA	Missing information about data or fees	PS#1	1	COGA	No icons accompanying headings	PS#1	1
COGA	Static or non-functional buttons and unclear layouts	PS#1	1	COGA	Clear and understandable page structure missing	PS#1	1
Other	Hallucinated image descriptions or objects	PS#13, PS#15, PS#18, PS#29	4	Other	Broken ARIA references	PS#33	1
Other	Content overload / linear navigation burden	PS#22, PS#27	2	Other	Dataset or annotation quality issues	PS#14, PS#16, PS#18, PS#29	4
Other	Design flaws or broken layout	PS#24	1	Other	HTML / CSS validation errors	PS#23, PS#26	2

RQ$_3$: What web accessibility issues and WCAG success-criterion violations are most frequently addressed?

Overall, the reviewed literature demonstrates a strong concentration on a limited set of accessibility issues, primarily those related to content perceivability and textual representation, with most studies explicitly grounding their analyses in established accessibility frameworks such as the Web Content Accessibility Guidelines (WCAG) and, to a lesser extent, the Cognitive Accessibility Guidelines (COGA). As summarized in Table 3, WCAG serves as the dominant reference framework for defining, detecting, and evaluating accessibility issues across the reviewed studies, while COGA guidelines appear in a smaller subset, reflecting a more limited but emerging engagement with cognitive accessibility concerns.

Within this framing, the most frequently addressed issue is missing or vague alternative text for non-text content, which appears in 21 studies (Table 4), making it the most dominant accessibility problem targeted by LLM-based approaches. This emphasis aligns closely with LLMs' strengths in natural language generation and refinement. A second prominent cluster of issues related to visual presentation and perceivability, particularly insufficient color contrast, which is reported in 16 studies and is often treated as a detectable, correctable violation. Together, these findings indicate that current research prioritizes accessibility issues that can be framed as missing textual equivalents or visually measurable deficiencies.

Beyond perceivability, a substantial portion of the literature focuses on structural, semantic, and programmatic accessibility issues that affect how web content is interpreted by assistive technologies. Commonly addressed problems include improper heading structure, missing or insufficient labels and instructions, incorrect form labeling or association, and inaccessible name, role, or value information, each appearing across multiple studies. These issues highlight recurring deficiencies in how semantic relationships and interaction cues are encoded in web interfaces, particularly for screen reader users. In addition, keyboard navigation and focus management issues are addressed in 11 studies (Table 4), indicating moderate attention to operability barriers faced by users who rely on non-pointer input modalities. However, other navigation-related concerns—such as missing bypass mechanisms [57, 60, 65], ambiguous link purposes [38, 41, 58], or inconsistent focus order—are addressed less frequently [19], suggesting uneven coverage even within interaction-focused accessibility research.

In contrast, cognitive accessibility and language-related issues, as captured by COGA guidelines, receive comparatively limited attention in the reviewed literature. Only a small number of studies [2, 4, 7, 43, 51, 58] address issues such as content overload, long uninterrupted text, lack of clear structure, or support for undoing actions, despite their known impact on users with cognitive and learning disabilities. Similarly, issues related to page language identification, reading level, and time-based media accessibility (e.g., captions and audio descriptions) appear sporadically. The table also reveals a set of LLM-specific and emergent accessibility issues that do not map cleanly to individual WCAG success criteria, including hallucinated image descriptions [34, 40, 45], dataset or annotation quality problems [12, 30, 40, 49], broken or incorrect ARIA usage [3, 11, 38, 49], and design flaws introduced during automated generation [29, 38, 51]. Although less frequent, these issues are particularly significant in the context of LLM-based systems, as they represent failure modes introduced or amplified by generative technologies, underscoring that accessibility research involving

Table 5: Categorization of evaluation methodologies used in the primary studies.

Category	Description	PS#
Empirical Performance	Quantitative benchmarking of model intelligence/accuracy via ML metrics (Accuracy, Recall, etc.).	PS#2, PS#4, PS#10, PS#11, PS#19, PS#32, PS#33
Technical Evaluation	Assessing outputs for WCAG/ARIA compliance using automated auditing tools (e.g., WAVE, aXe).	PS#1, PS#6, PS#8, PS#9, PS#12, PS#17, PS#18, PS#21, PS#24, PS#25, PS#30, PS#31
Qualitative Analysis	Descriptive assessments using expert reviews, heuristic evaluation, or thematic quality analysis.	PS#5, PS#23, PS#26, PS#28
User-Centric Study	Human-in-the-loop evaluations focusing on usability, satisfaction, or pedagogical outcomes.	PS#7, PS#13, PS#16
Mixed-Methods	Studies that explicitly integrate multiple lenses (e.g., technical audits + user surveys).	PS#3, PS#14, PS#15, PS#22, PS#27, PS#29

LLMs must address both long-standing web accessibility barriers and new risks associated with automated content generation.

RQ$_4$: How are LLM-based web accessibility approaches evaluated?

Table 5 summarizes the evaluation methods employed across the reviewed studies and shows that evaluation practices are dominated by technical evaluation and empirical performance methods. Technical evaluations are the most common, appearing in 12 studies, followed by empirical performance appearing in 7 studies, mixed methods approach being used in 6 studies. Qualitative analysis was employed by 4 studies, and 3 of the reviewed studies employed a user-centric approach. This distribution indicates that most studies seek to demonstrate feasibility or performance through controlled experiments, structured inspections, or proof-of-concept deployments, rather than through long-term field studies or large-scale longitudinal evaluations.

Table 6 further details the extent and nature of participant involvement in these evaluations. While several studies include human participants, participant-based evaluations vary widely in scale and composition. Many studies rely on authors, students, or developers as participants, often as proxies for end users. A smaller subset explicitly involves accessibility experts, reviewers, or users with disabilities, including blind, low-vision, deaf, or hard-of-hearing participants. Even when users with disabilities are included, sample sizes are typically modest and heterogeneous, with studies often combining multiple participant groups within a single evaluation. Notably, a non-trivial number of studies report no participant information at all, relying instead on automated audits, benchmark datasets, or expert inspection without direct user involvement.

Taken together, these results show that evaluation of LLM-based web accessibility approaches is methodologically diverse but uneven. While many studies employ empirical or experimental designs, direct validation with people with disabilities remains limited and inconsistent. The evidence base is therefore shaped largely by tool-centric and developer-centric evaluations, with comparatively fewer studies providing user-centered or participatory validation. This pattern suggests that current evaluation practices emphasize technical feasibility and comparative performance, while real-world effectiveness and user experience are less systematically assessed.

Table 6: Participant types and total number of participants reported in the primary studies (PS).

Study	Participant Types	# Total
PS#2 [31]	Authors	2
PS#3 [7]	Students	215
PS#5 [30]	Designers/UX Professionals	6
PS#6 [38]	Reviewers/Evaluators	2
PS#7 [65]	Blind (17), Low Vision Users (4)	21
PS#8 [29]	Accessibility Expert	2
PS#9 [52]	Accessibility Expert	1
PS#12 [19]	Blind/Low Vision Users (3), Deaf-/Hard of Hearing Users (3)	6
PS#13 [10]	Students	12
PS#14 [40]	Blind/Low Vision Users(40), Professional Describers (7), Sighted Users(347)	394
PS#15 [44]	Blind/Low Vision Users(8), Reviewers/Evaluators (5), Sighted Users(11)	24
PS#16 [20]	Blind (15) Low Vision Users (5)	20
PS#17 [60]	Low Vision Users (2), Sighted Users(2)	4
PS#18 [54]	Low Vision Users (1), Students (34)	35
PS#19 [11]	Developers/Engineers (88), Students(127)	215
PS#22 [5]	Low Vision Users	22
PS#23 [43]	Reviewers/Evaluators(3), Students(10)	13
PS#27 [57]	Blind (12), Low Vision Users (3)	15
PS#29 [34]	General Participants	3
PS#30 [49]	Authors	2
PS#31 [3]	Accessibility Experts	5
No participant information reported in *PS#1, PS#10, PS#11, PS#4, PS#32, PS#33, PS#24, PS#25, PS#26, PS#28, PS#20, PS#21*		

5 Discussion

Across the reviewed studies, LLMs are predominantly applied as task-oriented assistive components rather than as holistic accessibility solutions. Most approaches focus on supporting narrowly scoped activities—such as generating alternative text [34, 40, 45, 54], identifying accessibility violations [11, 26, 31, 41, 52], or suggesting localized fixes—embedded within existing development or evaluation workflows [27, 29, 50, 60]. This pattern suggests that current research conceptualizes LLMs as augmentation tools rather than as autonomous or end-to-end accessibility systems. While this reflects a pragmatic, risk-aware adoption strategy, it also limits exploration of how LLMs might support broader accessibility lifecycles, including continuous monitoring, iterative remediation, and user-facing accessibility assistance.

Simultaneously, the diversity of application forms, ranging from prompt-based usage to automated tools and browser extensions, indicates growing interest in integrating LLMs at different points of interaction. However, the limited number of standalone systems and benchmarks suggests that the field has not yet converged on shared infrastructures or standardized platforms for deploying and evaluating LLM-based accessibility support at scale.

Table 7: Summary of study contexts, experimental settings, models, and reported results across primary studies.

Study	Objective	Experiment / Setting	Website Type	Models	Best Model	Comparison Type	Result by Model
PS#1 [51]	Evaluate WCAG 2.2 compliance of AI-generated websites & compare ChatGPT vs. DeepSeek accessibility errors	Standardized prompting of LLMs to generate HTML/CSS websites, then evaluated for WCAG and cognitive	E-commerce (online clothing store)	DeepSeek, ChatGPT	DeepSeek	head-to-head	ChatGPT: Generated 54.05% of errors (140 errors); fewer errors in Buttons (38.24% vs 61.76%) and Misc Content (44.62% vs 55.38%). DeepSeek: Generated 45.95% of errors (119 errors); fewer errors in Headings (29.63% vs 70.37%) and Forms (34.21% vs 65.79%).
PS#2 [31]	Introduce GENA11Y, an automated accessibility checker to detect WCAG violations	GenA11y tool vs existing tools and ablation study on datasets and real websites	Real websites (Dataset Dr) and datasets (Da, Dm)	GPT-4o	GPT-4o	GenA11y vs existing tools	Experiment: GENA11Y (GPT-4o): 87.61% recall (516 issues); Next best tool (IBM): 38.20% recall. Ablation: Base Model (LLM with raw HTML): Detected 57 issues; GENA11Y: Detected 127 issues.
PS#3 [7]	Integrate LLM into software eng. courses to improve web accessibility skills.	Participants used ChatGPT to fix/repair identified issues	Participant websites	ChatGP 3.5	ChatGP 3.5	ChatGP 3.5 vs WAVE	ChatGP 3.5: Fixed 47% (AChecker assessment) and 46% (WAVE assessment) of violations.
PS#4 [26]	Detecting heading-related accessibility barriers	Models answered 6 A11y questions for 10 webpages	W3C webpage	Llama 3.1, GPT-4o, GPT-4o mini	GPT-4o mini	head-to-head	GPT-4o mini: Accuracy 73.33%, F-measure 82.61%. Llama 3.1: Accuracy 70.00%, F-measure 79.54%. GPT-4o: Accuracy 70.00%, F-measure 79.07%.
PS#5 [30]	Framework for Human-AI collaboration in UI accessibility via compliance & prompt eng.	Evaluated 50 UIs from 5 AI design tools; tested 4 tools on bad design tasks; and 200 UIs	Not reported	Not explicitly named	Not explicitly named	N/A	AI design tools (aggregate): "modest violations (M=0.47 on an 0-4 scale)". AI design tools (bad design task): Showed "significant resistance". Iterative prompting: Improved through dialogue in 42% of cases.
PS#6 [38]	Assess WCAG compliance of Wix and Framer AI website builders	Comparison of AI website builders (Wix vs Framer) using standardized prompts	E-commerce, Healthcare, School for Blind	Not reported	Wix	tool vs tool	Wix: Average errors M = 33.6 (SD = 10.5); Provided alt text in 100% of cases. Framer: Average errors M = 63.8 (SD = 12.2); Failed to provide alternative text in 100% of cases.
PS#7 [65]	To restructure HTML for accessible websites, validated with screen reader users	User study and automated auditing of extension variants vs original websites	E-commerce (Mercari, Amazon, Nordstrom)	GPT-4o	GPT-4o	baseline vs LLM	Option 1 regenerates HTML to enforce logical order, while Option 2 reorganizes existing tags. Both reduced task time ($p < 0.001$), with Option 1 performing better ($p < 0.001$).
PS#8 [29]	Compare ChatGPT and Claude on WCAG 2.1 compliance using accessibility-oriented prompts	Comparison of two LLMs using accessibility-agnostic and accessibility-oriented prompts, plus self-correction	Banking app homepages	Claude 3.5 Haiku, GPT-4-turbo	Tie	head-to-head | prompt variant	Expert Evaluation: Severity of issues (0-4 scale). Claude 3.5 Haiku consistently produced UIs with lower severity issues than ChatGPT across both prompt types (0.6 vs 1.5 agnostic; 0.2 vs 0.4 oriented). Both models successfully reduced the severity of accessibility issues to 0.0 after the self-reflection step.
PS#9 [52]	Investigate GPT-4 to automate validation of manual WCAG techniques	Comparison of Manual Testing vs Hybrid Testing (Manual + GenAI)	Travel website (skyscanner.net clone)	GPT-4o	GPT-4o	N/A	GPT-4o produced 251 responses, of which 232 were correct (92.4%): 185 successes, 54 failures, and 12 warnings. Across 45 images, TeVal/GPT-4o reported 15 successes, 29 failures, and 1 warning, correctly identifying 11/15 successes and 28/29 failures.
PS#10 [27]	Introduce AccessGuru to detect/fix accessibility violations and AccessCode dataset	AccessGuru multi-agent framework vs Baseline LLM vs Human Professionals	Real-world websites (10 representative sites)	GPT-4, AccessGuru (GPT-4)	GPT-4	Agent vs Baseline	Detection Accuracy: AccessGuru (82%) vs. Standard LLM (31%). Correction Accuracy: AccessGuru (74%) vs. Standard LLM (16%).
PS#11 [24]	Evaluate accuracy of ChatGPT and Copilot in producing accessible web code	Evaluating code generated by chatbots against "ideal" code metrics	Web page features	ChatGPT 3.5; Copilot	ChatGPT 3.5	head-to-head	ChatGPT 3.5: Valid Code 96.30%; Correct Code 77.78%; Screen Reader Compatible 55.56%. Copilot: Valid Code 88.89%; Correct Code 62.96%; Screen Reader Compatible 48.15%.
PS#12 [19]	Propose SceneGenAlly to improve 3D scene accessibility	"Acc3D" agent vs No Agent baseline in a 3D environment	3D web scenes	GPT-4o	GPT-4o	baseline vs LLM	The Acc3D agent enabled an 85% success rate for accessibility tasks, compared to 0% without the agent.
PS#13 [10]	Introduce LectureAssistant (AI chatbot) for blind/low-vision students in lectures	User study with LectureAssistant prototype (thinking aloud, tasks)	Education (Lecture video platform)	Llama 3.1, MiniCPM-V	N/A	N/A	No direct numeric comparison between Llama 3.1 and MiniCPM-V; evaluation focused on the tool's utility versus existing platforms.
PS#14 [40]	Introduce VideoA11y (MLLMs) to generate video descriptions and VideoA11y-40K dataset	Multi-study evaluation (Sighted and BLV users) of VideoA11y method	YouTube and TikTok (video-sharing platforms)	GPT-4V, GPT-4o, Video-LLaVA, LLaVA-OneVision, VILA	GPT-4V		GPT-4V significantly outperformed Video-LLaVA on descriptiveness, accuracy, and clarity ($p<0.05$), with no significant difference for objectiveness ($p>0.05$). BLV users preferred VideoA11y descriptions 90% of the time and rated them significantly higher in satisfaction (8.54 vs 5.43; $p<0.001$).
PS#15 [44]	Introduce TableNarrator using AI to generate alt text for image tables for BLV users	Technical evaluation and User study of TableNarrator system	Not reported	GPT-4, GPT-4V , Claude 3	GPT-4	head-to-head	TableNarrator (GPT-4) outperformed baselines (ROUGE-L = 0.360; selection = 66.86%), exceeding Claude 3 (0.205; 21.14%) and GPT-4V (0.240; 12.00%), and achieved higher row/column accuracy (0.83% vs. 0.05% and 0.19%).
PS#16 [20]	Explore user-driven AI audio descriptions allowing BLV users to control timing/detail	User study evaluating user-driven AD levels (concise vs detailed)	Online video (YouTube, TikTok, Instagram)	GPT-4V	N/A	N/A	GPT-4V (User-driven): Median efficiency=4, effectiveness=4, enjoyability=3; Concise AD activated more often (Mean=5.42) than detailed (Mean=3.58).
PS#17 [60]	Workflow using LLMs to remediate inaccessible web content to "born-accessible"	Generation of remediated webpages	Static university pages	GPT-4o, Gemini	Gemini	head-to-head	Gemini produced significantly fewer WAVE errors ($p = 0.0435$) and alerts ($p = 0.0151$) compared to GPT-4o, though Lighthouse scores showed no significant difference.
PS#18 [54]	AlternAtIve: Browser extension using LLMs to generate alt text for STEM images	Comparison of extension (Gemini) vs other tools	Not reported	Gemini 1.5 Pro , ChatGPT-4	Gemini 1.5 Pro	tool vs tool	AlternAtIve [H] (using Gemini 1.5 Pro) achieved the highest quality scores (Qua1 mean 0.91), significantly outperforming other treatments ($p < 0.01$ in most pairs).
PS#19 [11]	Investigate LLMs' ability to detect accessibility issues in dynamic web content	Analysis of handcrafted code and student projects	Web application components	GPT-4o mini, o3-mini, Gemini	Gemini	head-to-head	Gemini had a much lower hallucination rate (20.1%) compared to GPT-4o mini (69.9%) in detecting violations. o3-mini achieved the highest self-verification accuracy (79%), significantly outperforming GPT-4o mini (34%) and Gemini (30%).
PS#20 [12]	To benchmark LLMs on detecting/remediating code accessibility issues	Students generated components using LLMs	Not reported	GPT-4o mini, o3-mini, and Gemini 2.0 Flash	N/A	N/A	The paper states that GPT-4o mini, o3-mini, and Gemini 2.0 Flash were evaluated in prior work using this dataset, but provides no metrics, numeric results, or comparative outcomes in this article.
PS#21 [41]	Investigate if LLMs can automate manual WCAG success criteria evaluation	LLM-powered scripts evaluated on WCAG SCs	Not reported	GPT-3.5, GPT-4, Claude 2, Google Bard	N/A	N/A	The study reports that LLM scripts achieved 78%/85%/100% success across the three WCAG SC, with 87.18% overall (34/39). SC 1.1.1 (GPT-3.5): 78% successful. SC 2.4.4 (Claude): 85% successful. SC 3.1.2 (GPT-4): 100% successful.
PS#22 [5]	Chrome extension using local LLMs to summarize web content for the visually impaired	Computational evaluation and Usability study	BBC News articles across domains	Llama 3.1, Phi 3, Gemma 2, and Mistral	Gemma 2	head-to-head	Computational evaluation: Gemma 2 led in overall score (0.585) and readability (FRE 52.53), while Mistral excelled in semantic similarity (0.8033). User preference: Gemma 2 was most preferred (35.2%).
PS#23 [43]	Case study using LLMs with interface images to assist in heuristic usability evaluations for ID users	Evaluation of "Add Task" and "Add Habit" screens	Gamified web solution for task management and habit formation	GPT-4o, DeepSeek V3, Claude 3.7 Sonnet	Claude 3.7 Sonnet	head-to-head	Claude (3.7 Sonnet): Found most issues (16 in Exp 1, 17 in Exp 2); Highest unique issues (6 in Exp 1, 10 in Exp 2). ChatGPT (GPT-4o): Unique issues (5 in Exp 1, 6 in Exp 2). DeepSeek (V3): Unique issues (2 in Exp 1, 1 in Exp 2). Gemini (2.0 Flash): Fewest issues found (7 in Exp 1, 12 in Exp 2); Unique issues (2 in Exp 1, 1 in Exp 2).
PS#24 [2]	Benchmark generative-AI coding tools on producing accessible HTML/web	Systematic testing with prompts and follow-ups	Not reported	ChatGPT 4o,Grok 3, Copilot Pro, Claude 3.7 Sonnet	Claude 3.7 Sonnet	head-to-head	Claude 3.7 Sonnet and Copilot Pro tied for highest first-attempt success (72.7%), but Claude was most efficient (fewest follow-up prompts) and had fewer severe violations than Grok 3.
PS#25 [4]	Compare accessibility and readability of ChatGPT vs. Google BARD chatbots	Evaluation of interface accessibility and output readability	Conversational AI chatbot interfaces	GPT 3.5, BARD	Google BARD	head-to-head	BARD passed AChecker but had 26 WAVE ARIA issues, while GPT failed AChecker but had 0 WAVE ARIA issues. Google BARD consistently outperformed GPT in readability, yielding higher scores across most measures.
PS#26 [58]	Analyze education website accessibility for optimization	Case study optimizing a sample page	Education board websites	Claude	N/A	N/A	Claude: Successfully improved accessibility by increasing color contrast (e.g., #ccc to #fff) and adding focus outlines.
PS#27 [57]	Tool using BERT/ChatGPT to generate navigation aids for screen-reader users	User study with simulated prototype and browser extension	Text-only web pages from news domains	BERT, GPT-3	N/A	N/A	BERT achieved the lowest error rate (8%) on long documents, outperforming C99, while GPT-3 produced more coherent, descriptive headers than keyword-based baselines.
PS#28 [23]	Investigate ChatGPT for evaluating and correcting HTML accessibility issues	Qualitative analysis of LLM responses to code snippets	Not reported	GPT3.5	N/A	N/A	GPT-3.5: Correctly identified well-structured forms and missing labels; flagged issues in tables but suggested semantics-altering fixes; correctly noted context for empty alt attributes.
PS#29 [34]	Extension using LLMs to generate context-aware image descriptions	Evaluation of 10 images by participants	News articles	GPT-4o, AI-MCS	SmartCaption AI (GPT-4o)	Tool vs Tool	Experiment A: SmartCaption AI outperformed baselines (8.3/10 vs. 3.6/10 and 1.7/10), which frequently hallucinated objects. Experiment B: Initial latency was high (11.11s) but dropped to 2.54s with caching.
PS#30 [49]	Evaluate the accessibility of three GenAI website builders and provide best practices	Expert and automated testing of generated sites	AI Website Builder Platforms / Artificial Design Intelligence tools.	Dorik.com, Relume.io, and Wix.com	Relume.io	Generative AI Website Builders	While Relume.io was the only tool rated "Partially accessible" in the screen reader walkthrough and produced fewer WAVE alerts than the others, all three platforms exhibited significant accessibility failures, such as high rates of unlabeled buttons and lack of WCAG 2.1 AA compliance.
PS#31 [3]	Evaluate web accessibility of 20 GenAI educational applications	Automated (WAVE) and manual review of 20 tools	20 generative artificial intelligence (AI) applications	ChatGPT, Midjourney, Copilot, Suno	Midjourney	tool vs tool	Midjourney: 1 WAVE error; 6 contrast errors. Suno: 1 WAVE error; 27 contrast errors. ChatGPT: 3 WAVE errors; 0 contrast errors; 1 alert. Copilot: 3 WAVE errors; 0 contrast errors.
PS#32 [45]	To generate alt text and search images for BLV users			MiniGPT-4, BLIP	MiniGPT-4	head-to-head	MiniGPT-4 produced longer alt text and higher top-10 accuracy (68.7%) than BLIP (64.3%) and object-recognition baselines (13.3–24.3%).
PS#33 [50]	Evaluate ChatGPT's effectiveness in fixing WCAG 2.1	Using ChatGPT to fix 39 identified errors	Qatar & British Airways	ChatGPT	ChatGPT	N/A	ChatGPT successfully remediated 37 out of 39 identified errors across the two websites, achieving an accuracy rate of 94%.

Alignment with Accessibility Standards and Disability Coverage. The review shows strong reliance on WCAG as the primary framework for defining and operationalizing accessibility, with WCAG success criteria frequently used to guide task selection, evaluation, and reporting [2, 3, 11, 24, 27, 31, 41, 49, 50, 52, 60]. In contrast, COGA guidelines and other disability-specific frameworks are referenced far less often [51], resulting in uneven coverage across disability groups. Most studies prioritize accessibility needs related to visual impairments and screen reader compatibility, while cognitive, learning, and neurodiverse accessibility needs remain underrepresented.

This imbalance suggests that current LLM-based accessibility research largely mirrors long-standing trends in web accessibility, where visually observable or machine-detectable issues dominate. Although WCAG alignment provides consistency and comparability, limited engagement with COGA highlights an opportunity—and a need—to expand accessibility research beyond compliance-driven perspectives toward more inclusive and cognitively informed design considerations.

Technical Design Choices. From a technical standpoint, the literature is characterized by concentration rather than diversity. Most studies rely on a small set of commercial, general-purpose LLMs and relatively simple prompting strategies, with limited experimentation in model adaptation, architectural complexity, or long-running systems [2, 4, 7, 23, 24, 26, 31, 50, 51]. Prompt engineering is frequently treated as the primary mechanism for improving accessibility outcomes, while architectural decisions—such as multi-agent coordination, feedback loops, or integration with accessibility tooling—are less commonly explored [11, 26, 27, 34, 41, 60].

This focus reflects both the accessibility of current LLM APIs and the exploratory nature of the field. However, it also constrains the types of accessibility problems that can be effectively addressed. More complex accessibility needs—such as context-sensitive guidance, personalization, or sustained interaction—may require architectural designs that go beyond single-prompt, single-response paradigms. The current emphasis on simplicity may therefore limit progress toward robust, real-world accessibility support.

Evaluation Practices and Evidence Quality. Evaluation practices across the literature are methodologically diverse but uneven. While many studies employ empirical or experimental evaluations, these are often conducted in controlled settings using automated tools, expert review, or small-scale participant samples (Table 6). Direct involvement of people with disabilities remains limited, and when present, participant groups are often small or heterogeneous. As a result, much of the reported evidence emphasizes technical feasibility and comparative performance rather than real-world usability or lived experience [2, 4, 26, 27, 34, 40].

For a domain where user impact is central, this raises concerns about the ecological validity of current findings. The limited use of participatory or longitudinal evaluation approaches suggests that accessibility research involving LLMs has yet to fully embrace user-centered and inclusive evaluation practices. Strengthening this aspect of research will be critical to ensuring that LLM-based accessibility tools genuinely benefit the users they aim to support.

Prioritized Accessibility Issues and Emerging Challenges. The synthesis reveals that LLM-based accessibility research concentrates heavily on perceivability and semantic clarity, particularly issues involving missing alternative text, insufficient color contrast, labeling, and structural markup [2, 26, 27, 34, 40, 41, 45, 51, 54]. These issues are well-suited to automated detection and natural language generation, which likely explains their prominence [31, 41, 52]. In contrast, accessibility challenges that involve temporal behavior, interaction complexity, or cognitive load—such as time-based media accessibility, navigation consistency, reading level, and content comprehensibility—are addressed far less frequently [4, 5, 10, 11, 40, 43, 51].

Importantly, the review also identifies emergent, LLM-specific accessibility issues, such as hallucinated descriptions, misleading compliance claims, and incorrect ARIA usage. These problems are not always captured by existing WCAG success criteria, indicating that generative systems introduce new classes of accessibility risks. Addressing these risks may require extending current standards, developing new evaluation heuristics, or rethinking how accessibility compliance is assessed in the presence of AI-generated content.

6 Research Implications

This section discusses the implications of our findings for both research and practice in the context of LLM-supported web accessibility. We highlight directions for future research and practical considerations for responsibly integrating LLMs into accessibility workflows.

6.1 Implications for Research

Future research should broaden the scope of accessibility issues addressed by LLM-based approaches, particularly by engaging more deeply with cognitive accessibility, neurodiversity, and time-based media accessibility. This shift is necessary because current automated tools frequently overlook the subjective and non-code-based requirements of the COGA guidelines [36, 50].

Researchers should also explore architectural diversity beyond single-prompt or single-model designs, including multi-stage pipelines, adaptive systems, and hybrid approaches that combine LLMs with rule-based accessibility tools. Such exploration is essential to understanding how LLMs can support sustained, context-aware accessibility workflows that can handle the dynamic complexity of modern web frameworks [8]. Evaluation practices must move beyond feasibility demonstrations toward rigorous, user-centered, and participatory methods. Involving people with disabilities as primary stakeholders throughout design and evaluation is critical, ensuring that AI-driven solutions address lived experience rather than just technical compliance [60].

6.2 Implications for Practice

For practitioners, the findings suggest that LLMs can be valuable in supporting specific accessibility tasks—such as generating alternative text or identifying common accessibility violations—but should be used with caution. LLM outputs should be treated as assistive suggestions rather than authoritative solutions, particularly given the risk of AI hallucination that technically pass automated checks but fail screen-reader testing [31, 52].

Practitioners should also be aware that current LLM-based tools tend to emphasize compliance with easily detectable WCAG criteria, while offering limited support for cognitive and experiential aspects of accessibility [50, 61]. Integrating LLMs into accessibility workflows, therefore, requires complementary practices, including expert review, user testing, and adherence to established accessibility standards. In general, both researchers and practitioners should approach LLM-based accessibility support as a collaborative, human-in-the-loop process, ensuring that automation enhances rather than undermines inclusive web experiences [64].

7 Threats to Validity

As with any systematic literature review, this study is subject to several threats to validity. We outline these threats below, along with the steps taken to mitigate them.

Selection Validity. Relevant studies may have been missed due to limitations in the search strategy, choice of digital libraries, or keyword formulation. To mitigate this risk, we searched multiple widely used digital libraries, iteratively refined the search strings through pilot searches, and applied backward and forward snowballing. Nonetheless, some relevant studies—particularly those using non-standard terminology or published in non-indexed venues may not have been captured.

Construct Validity. Threats to construct validity arise from variations in how primary studies define and operationalize concepts such as web accessibility, LLM-based approaches, and accessibility issues. To mitigate these threats, we relied on the explicit definitions and descriptions provided by the authors of the primary studies and systematically mapped reported issues to established frameworks such as WCAG and COGA, where applicable, to ensure consistent categorization.

Internal Validity. The manual filtering, data extraction, and synthesis processes may introduce researcher bias or inconsistent interpretation. To mitigate this threat, two authors independently reviewed the extracted data, resolved disagreements through discussion, and documented decisions in a shared spreadsheet. The authors' prior research experience in web accessibility further supported consistency during the review process.

8 Conclusion

This study presents a systematic literature review of 33 peer-reviewed studies investigating the use of Large Language Models to support web accessibility. Through our analysis, we identify the accessibility tasks addressed by these studies, the LLM-based approaches and system designs employed, and the accessibility issues and standards considered. Our findings show that most studies apply LLMs to support a limited set of accessibility tasks—primarily content generation, issue detection, and semantic remediation—with WCAG serving as the dominant reference framework and limited incorporation of cognitive accessibility guidelines such as COGA.

Further analysis highlights common patterns across the reviewed studies, including a strong reliance on general-purpose LLMs, prompt-based interaction strategies, and relatively simple system architectures. Evaluation practices vary substantially, with many studies relying on automated assessments or expert inspection, while fewer involve users with disabilities. We envision that our findings serve as a consolidated reference for researchers and practitioners seeking to understand the current landscape of LLM-supported web accessibility. Future work in this area includes broader coverage of accessibility needs, particularly cognitive accessibility; more rigorous user-centered evaluation; and the development of benchmarks and evaluation frameworks tailored to the accessibility challenges posed by generative AI systems. Additionally, research should investigate the evolution of automated validators as they integrate AI-driven workflows—such as AI-assisted guided testing—to enhance detection and remediation accuracy.

References

[1] 2021. *Making Content Usable for People with Cognitive and Learning Disabilities.* W3C Working Group Note. W3C. https://www.w3.org/TR/coga-usable/

[2] Iyad Abu Doush and Reem Kassem. 2025. Can generative AI create accessible web code? A benchmark analysis of AI-generated HTML against accessibility standards. *Universal Access in the Information Society* 24, 4 (2025), 3483–3506.

[3] Patricia Acosta-Vargas, Gloria Acosta-Vargas, Belén Salvador-Acosta, and Janio Jadán-Guerrero. 2024. Addressing web accessibility challenges with generative artificial intelligence tools for inclusive education. In *2024 Tenth International Conference on eDemocracy & eGovernment (ICEDEG).* IEEE, 1–7.

[4] CP Afsal and KS Kuppusamy. 2024. Comparative Assessment of Accessibility and Readability in Generative AI: A Case Study of GPT and Google BARD. In *International Conference on Emerging Trends and Technologies on Intelligent Systems.* Springer, 397–408.

[5] CP Afsal and KS Kuppusamy. 2025. WEBSumm: A Chrome Extension for Summarizing Web Content Using LLMs for Visually Impaired Users. *SN Computer Science* 6, 2 (2025), 1–15.

[6] Muhammad Ali. 2025. Use of AI in improving website accessibility: a systematic literature review and conceptual framework for digital inclusion. (2025).

[7] Wajdi Aljedaani, Marcelo Medeiros Eler, and PD Parthasarathy. 2025. Enhancing accessibility in software engineering projects with large language models (llms). In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1.* 25–31.

[8] Wajdi Aljedaani, Abdulrahman Habib, Ahmed Aljohani, Marcelo Eler, and Yunhe Feng. 2024. Does chatgpt generate accessible code? investigating accessibility challenges in llm-generated source code. In *Proceedings of the 21st International Web for All Conference.* 165–176.

[9] Wajdi Aljedaani, Rubel Hassan Mollik, Eysha Saad, Marcelo M Eler, and Asmaa Mansour Alghamdi. 2025. Challenges and barriers faced by blind and visually impaired users on social media: a systematic literature review. *Universal Access in the Information Society* (2025), 1–30.

[10] Katharina Anderer, Karin Müller, Lukas Strobel, Matthias Wölfel, Jan Niehues, and Kathrin Gerling. 2025. Making Lecture Videos Accessible for Students who are Blind or have Low Vision through AI-Assisted Navigation and Visual Question Answering. In *Proceedings of the 27th International ACM SIGACCESS Conference on Computers and Accessibility.* 1–15.

[11] Manuel Andruccioli, Barry Bassi, Giovanni Delnevo, and Paola Salomoni. 2025. Leveraging Large Language Models for Sustainable and Inclusive Web Accessibility. *Big Data and Cognitive Computing* 9, 10 (2025), 247.

[12] Manuel Andruccioli, Barry Bassi, Giovanni Delnevo, and Paola Salomoni. 2025. The Tabular Accessibility Dataset: A Benchmark for LLM-Based Web Accessibility Auditing. *Data* 10, 9 (2025), 149.

[13] Jinat Ara, Cecilia Sik-Lanyi, and Arpad Kelemen. 2024. Accessibility engineering in web evaluation process: a systematic literature review. *Universal Access in the Information Society* 23, 2 (2024), 653–686.

[14] Mateo Borina, Edi Kalister, and Tihomir Orehovački. 2022. Web accessibility for people with cognitive disabilities: a systematic literature review from 2015 to 2021. In *International Conference on Human-Computer Interaction.* Springer, 261–276.

[15] Peter Brophy and Jenny Craven. 2007. Web accessibility. *Library trends* 55, 4 (2007), 950–972.

[16] Ben Caldwell, Michael Cooper, Loretta Guarino Reid, Gregg Vanderheiden, Wendy Chisholm, John Slatin, and Jason White. 2008. Web content accessibility guidelines (WCAG) 2.0. *WWW Consortium (W3C)* 290, 1-34 (2008), 5–12.

[17] Milton Campoverde-Molina, Sergio Lujan-Mora, and Llorenc Valverde Garcia. 2020. Empirical studies on web accessibility of educational websites: A systematic literature review. *IEEE access* 8 (2020), 91676–91700.

[18] Milton Campoverde-Molina, Sergio Luján-Mora, and Llorenç Valverde. 2023. Accessibility of university websites worldwide: a systematic literature review. *Universal Access in the Information Society* 22, 1 (2023), 133–168.

[19] Xinyun Cao, Kexin Phyllis Ju, Chenglin Li, and Dhruv Jain. 2025. SceneGenA11y: How can Runtime Generative tools improve the Accessibility of a Virtual 3D

Scene?. In *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. 1–10.
[20] Maryam Cheema, Hasti Seifi, and Pooyan Fazli. 2025. Describe Now: User-Driven Audio Description for Blind and Low Vision Individuals. In *Proceedings of the 2025 ACM Designing Interactive Systems Conference*. 458–474.
[21] Andy Coverdale, Sarah Lewthwaite, and Sarah Horton. 2024. Digital accessibility education in context: expert perspectives on building capacity in academia and the workplace. *ACM Transactions on Accessible Computing* 17, 2 (2024), 1–21.
[22] Mauricio Cruz-Portilla, Juan Carlos Pérez-Arriaga, Jorge Octavio Ocharán-Hernández, and Ángel J Sánchez-García. 2021. Accessibility in the software development life cycle: a systematic literature review. In *2021 9th International Conference in Software Engineering Research and Innovation (CONISOFT)*. IEEE, 97–103.
[23] Giovanni Delnevo, Manuel Andruccioli, and Silvia Mirri. 2024. On the interaction with large language models for web accessibility: Implications and challenges. In *2024 IEEE 21st Consumer Communications & Networking Conference (CCNC)*. IEEE, 1–6.
[24] Iyad Abu Doush and Reem Kassem. 2024. Evaluating ai-generated web code for accessibility compliance: A metric-driven approach. In *Proceedings of the 11th International Conference on Software Development and Technologies for Enhancing Accessibility and Fighting Info-exclusion*. 338–344.
[25] Nikolaos Droutsas, Fotios Spyridonis, Damon Daylamani-Zad, and Gheorghita Ghinea. 2025. Web accessibility barriers and their cross-disability impact in eSystems: A scoping review. *Computer Standards & Interfaces* 92 (2025), 103923.
[26] Carlos Duarte, Miguel Costa, Letícia Seixas Pereira, and João Guerreiro. 2025. Expanding Automated Accessibility Evaluations: Leveraging Large Language Models for Heading-Related Barriers. In *Companion Proceedings of the 30th International Conference on Intelligent User Interfaces*. 39–42.
[27] Nadeen Fathallah, Daniel Hernández, and Steffen Staab. 2025. AccessGuru: Leveraging LLMs to Detect and Correct Web Accessibility Violations in HTML Code. In *Proceedings of the 27th International ACM SIGACCESS Conference on Computers and Accessibility*. 1–22.
[28] Katia Romero Felizardo, Emilia Mendes, Marcos Kalinowski, Érica Ferreira Souza, and Nandamudi L Vijaykumar. 2016. Using forward snowballing to update systematic reviews in software engineering. In *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. 1–6.
[29] Alexandra-Elena Gurita and Radu-Daniel Vatavu. 2025. When LLM-Generated Code Perpetuates User Interface Accessibility Barriers, How Can We Break the Cycle?. In *Proceedings of the 22nd International Web for All Conference*. 124–134.
[30] Alexandra-Elena Guriță. 2025. Exploring the Intersection of UI Accessibility and Generative AI: A Framework for Human-AI Collaboration. In *Companion Publication of the 2025 ACM Designing Interactive Systems Conference*. 116–120.
[31] Ziyao He, Syed Fatiul Huq, and Sam Malek. 2025. Enhancing Web Accessibility: Automated Detection of Issues with Generative AI. *Proceedings of the ACM on Software Engineering* 2, FSE (2025), 2264–2287.
[32] Dylan H Hewitt and Yingchen He. 2021. Internet Accessibility for blind and visually-impaired users: An evaluation of official us state and territory covid-19 websites. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Vol. 65. SAGE Publications Sage CA: Los Angeles, CA, 154–158.
[33] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems* 43, 2 (2025), 1–55.
[34] Gia Ky Huynh and Wenjun Lin. 2024. SmartCaption AI-Enhancing Web Accessibility with Context-Aware Image Descriptions Using Large Language Models. In *2024 International Conference on Computer and Applications (ICCA)*. IEEE, 1–7.
[35] Samireh Jalali and Claes Wohlin. 2012. Systematic literature studies: database searches vs. backward snowballing. In *Proceedings of the ACM-IEEE international symposium on Empirical software engineering and measurement*. 29–38.
[36] Blessing Jerry, Lourdes Moreno, Virginia Francisco, and Raquel Hervas. 2026. LLM-Driven Accessible Interface: A Model-Based Approach. *arXiv preprint arXiv:2601.06616* (2026).
[37] Barbara Kitchenham, O Pearl Brereton, David Budgen, Mark Turner, John Bailey, and Stephen Linkman. 2009. Systematic literature reviews in software engineering–a systematic literature review. *Information and software technology* 51, 1 (2009), 7–15.
[38] Katelyn Leedy, Makenzie Preston, Jinjuan Feng, and Jeba Rezwana. 2025. Accessibility in the Age of Generative AI Web Based Builders: Evaluating Web Design Tools for Inclusive Practices. In *Proceedings of the 28th International Academic Mindtrek Conference*. 304–314.
[39] Sarah Lewthwaite. 2014. Web accessibility standards and disability: developing critical perspectives on accessibility. *Disability and Rehabilitation* 36, 16 (2014), 1375–1383.
[40] Chaoyu Li, Sid Padmanabhuni, Maryam S Cheema, Hasti Seifi, and Pooyan Fazli. 2025. Videoa11y: Method and dataset for accessible video description. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. 1–29.
[41] Juan-Miguel López-Gil and Juanan Pereira. 2025. Turning manual web accessibility success criteria into automatic: an LLM-based approach. *Universal Access in the Information Society* 24, 1 (2025), 837–852.
[42] Shridhar Mehendale and Ankit Walishetti. 2025. DexAssist: A Voice-Enabled Dual-LLM Framework for Accessible Web Navigation. In *Intelligent Human Computer Interaction*, Dhananjay Singh, Jan-Willem van 't Klooster, and Uma Shanker Tiwary (Eds.). Springer Nature Switzerland, Cham, 171–177.
[43] Pedro Afonso F Michalichem, Leonardo Tórtoro Pereira, and Kamila Rios da Hora Rodrigues. 2025. A gamified solution to promote positive habits in children and adolescents with Intellectual Disabilities. *Journal on Interactive Systems* 16, 1 (2025), 833–849.
[44] Ye Mo, Gang Huang, liangcheng li, Dazhen Deng, Zhi Yu, Yilun Xu, Kai Ye, Sheng Zhou, and Jiajun Bu. 2025. TableNarrator: Making Image Tables Accessible to Blind and Low Vision People. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. 1–17.
[45] Yunseo Moon, Hyunmin Lee, SeungYoung Oh, and Hyunggu Jung. 2024. SaGol: using MiniGPT-4 to generate alt text for improving image accessibility. In *Proceedings of the Thirty-ThirdInternational Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization Jeju …, 8745–8748.
[46] Peya Mowar, Yi-Hao Peng, Jason Wu, Aaron Steinfeld, and Jeffrey P Bigham. 2025. CodeA11y: Making AI Coding Assistants Useful for Accessible Web Development. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. 1–15.
[47] Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. 2025. A comprehensive overview of large language models. *ACM Transactions on Intelligent Systems and Technology* 16, 5 (2025), 1–72.
[48] Karla Ordoñez, José Hilera, and Samanta Cueva. 2022. Model-driven development of accessible software: a systematic literature review. *Universal Access in the Information Society* 21, 1 (2022), 295–324.
[49] Sushil K Oswal and Hitender K Oswal. 2024. Examining the accessibility of generative AI website builder tools for blind and low vision users: 21 best practices for designers and developers. In *2024 IEEE International Professional Communication Conference (ProComm)*. IEEE, 121–128.
[50] Achraf Othman, Amira Dhouib, and Aljazi Nasser Al Jabor. 2023. Fostering websites accessibility: A case study on the use of the Large Language Models ChatGPT for automatic remediation. In *Proceedings of the 16th international conference on pervasive technologies related to assistive environments*. 707–713.
[51] Ruchi Panchanadikar, Mitali Shrikant Bhosekar, and Emma Dixon. 2025. Can Generative AI Create Accessible Websites?. In *Proceedings of the 27th International ACM SIGACCESS Conference on Computers and Accessibility*. 1–6.
[52] Fabio Paternò, Manuela Vinci, Marco Manca, and Nicola Iannuzzi. 2025. How an LLM Can Improve Automatic Web Accessibility Validation?. In *Proceedings of the 16th Biannual Conference of the Italian SIGCHI Chapter*. 1–8.
[53] Cecilia Maria Patino and Juliana Carvalho Ferreira. 2018. Inclusion and exclusion criteria in research studies: definitions and why they matter. *Jornal Brasileiro de Pneumologia* 44 (2018), 84–84.
[54] Giacomo Pedemonte, Maurizio Leotta, Marina Ribaudo, et al. 2025. Improving web accessibility with an llm-based tool: A preliminary evaluation for stem images. *IEEE ACCESS* 13 (2025), 107566–107582.
[55] Nataša Rajh, Klaus Miesenberger, and Reinhard Koutny. 2024. Accessibility in the software engineering (SE) process and in integrated development environments (IDEs): a systematic literature review. In *International Conference on Computers Helping People with Special Needs*. Springer, 11–18.
[56] Aylin Sarioğlu, Haydar Metin, and Dominik Bork. 2025. Accessibility in conceptual modeling—A systematic literature review, a keyboard-only UML modeling tool, and a research roadmap. *Data & Knowledge Engineering* (2025), 102423.
[57] Jorge Sassaki Resende Silva, Paula Christina Figueira Cardoso, Raphael Winckler De Bettio, Daniela Cardoso Tavares, Carlos Alberto Silva, Willian Massami Watanabe, and André Pimenta Freire. 2024. In-page navigation aids for screen-reader users with automatic topicalisation and labelling. *ACM Transactions on Accessible Computing* 17, 2 (2024), 1–45.
[58] Utkarsha Singh, Jeevithashree Divya Venkatesh, Anujith Muraleedharan, KamalPreet Singh Saluja, Anamika JH, and Pradipta Biswas. 2024. Accessibility analysis of educational websites using WCAG 2.0. *Digital Government: Research and Practice* 5, 3 (2024), 1–28.
[59] Pedro Teixeira, Celeste Eusebio, and Leonor Teixeira. 2024. Understanding the integration of accessibility requirements in the development process of information systems: a systematic literature review. *Requirements Engineering* 29, 2 (2024), 143–176.
[60] Guillermo Vera-Amaro and Jose Rafael Rojano-Caceres. 2025. Accessible Web Content Generation Using LLMs: An Empirical Study on Prompting Strategies and Template-Guided Remediation. *IEEE Latin America Transactions* 23, 12 (2025), 1230–1239.
[61] Markel Vigo, Justin Brown, and Vivienne Conway. 2013. Benchmarking web accessibility evaluation tools: measuring the harm of sole reliance on automated

tests. In *Proceedings of the 10th international cross-disciplinary conference on web accessibility*. 1–10.

[62] Web Accessibility In Mind WebAIM. 2020. The WebAIM MillionAn annual accessibility analysis of the top 1,000,000 home pages. *Retrieved October* 13 (2020), 2020.

[63] Yeliz Yesilada and Simon Harper. 2019. Web Accessibility. *Human–Computer Interaction Series. Springer London* (2019).

[64] Rui Yu, Sooyeon Lee, Jingyi Xie, Syed Masum Billah, and John M Carroll. 2024. Human–AI collaboration for remote sighted assistance: perspectives from the LLM era. *Future internet* 16, 7 (2024), 254.

[65] Yaman Yu, Bektur Ryskeldiev, Ayaka Tsutsui, Matthew Gillingham, and Yang Wang. 2025. From cluttered to clear: Improving the web accessibility design for screen reader users in e-commerce with generative AI. In *Proceedings of the 27th International ACM SIGACCESS Conference on Computers and Accessibility*. 1–19.

[66] Yehong Zhou and Chun-Hsien Chen. 2025. Examining the Impact of Large Language Models on Design: Functions, Strengths, Limitations, and Roles. *Design and Artificial Intelligence* (2025), 100017.