Access Buddy, [Ai-Based Automated Web Accessibility Testing and Recommendation System]

Suhani Pawade[1], Prof. Sayli Bhosale[2]
[1,2]*Department of Information Technology, Sonopant Dandekar Arts, V.S. Apte Commerce & M.H. Mehta Science College Affiliated to University of Mumbai, Palghar, Maharashtra, India*

***Abstract*—Web accessibility helps ensure that websites can be perceived, understood, navigated, and used by people with different abilities. However, accessibility evaluation can be difficult for developers and testers when it depends mainly on manual inspection or when automated tools present findings in highly technical language. This paper presents Access Buddy, an AI-based automated web accessibility testing and recommendation system designed to make website accessibility evaluation more organized and understandable. The proposed system accepts a website URL and runs a browser-based accessibility rule engine to identify a defined set of supported accessibility issues. For each finding, the system presents information such as severity, relevant Web Content Accessibility Guidelines (WCAG) references, affected elements, HTML details, and an overall accessibility score. An AI-based component is used to explain findings in plain language and suggest possible corrective actions. The system also stores scan history, supports issue-status tracking, enables retesting, and provides dashboard and reporting features. The design follows a complementary approach in which rule-based checks perform the supported detection tasks while AI assists with explanation and recommendations. The system is intended to support developers, testers, students, and website owners; it does not replace manual accessibility evaluation or guarantee detection of every accessibility barrier.**

***Index Terms*—Artificial Intelligence, Web Accessibility, WCAG, Automated Testing, Accessibility Evaluation, Large Language Models, Recommendation System, Access Buddy.**

I. INTRODUCTION

Web applications are widely used for education, communication, public services, commerce, and everyday activities. To be useful to everyone, websites should be designed so that people with different abilities can access information and complete tasks. Barriers such as missing alternative text, unclear form labels, unsuitable heading structures, and keyboard-navigation problems can make a website difficult to use, particularly for people who rely on assistive technologies.

The Web Content Accessibility Guidelines (WCAG), published by the World Wide Web Consortium (W3C), provide a recognized framework for evaluating web accessibility. Automated evaluation tools can check certain technical rules quickly and consistently, but not every accessibility requirement can be decided reliably by automation alone. Some criteria depend on context, meaning, visual presentation, or human judgment. Existing tools may also present results that are difficult for inexperienced users to interpret.

Access Buddy is proposed as an AI-based automated web accessibility testing and recommendation system. A user enters a website URL, runs a scan, and receives structured results with severity information, WCAG references, affected elements, and an accessibility score. The AI component then helps explain the detected issues and proposes possible fixes.

1.1 Problem Statement

Website accessibility checking can be time-consuming when performed entirely through manual inspection. Although automated tools can detect selected accessibility issues, users may find it difficult to understand the technical output, determine the importance of each finding, or decide how to address it. In addition, accessibility work requires a way to retain earlier results and verify whether reported issues remain after changes are made. The problem addressed by Access Buddy is to provide a unified platform that automates supported checks, explains findings in accessible language, recommends possible corrective actions, and organizes scan history and

issue status without claiming to replace comprehensive manual testing.

II. LITERATURE REVIEW AND OBJECTIVES

[1] Markel Vigo and Giorgio Brajnik,2011studied automatic web accessibility metrics and examined their effectiveness in evaluating website accessibility. Their research compared different accessibility metrics based on factors such as validity, reliability, sensitivity, adequacy, and complexity. The study highlighted the importance of selecting appropriate metrics for reliable accessibility evaluation. This research is relevant to Access Buddy because the system uses automated accessibility results and a scoring mechanism to help users understand the accessibility status of a website.

[2] Giorgio Brajnik,2004 investigated methods for comparing the effectiveness of web accessibility evaluation tools. The study emphasized that accessibility tools should be evaluated based on their ability to identify accessibility barriers and provide meaningful results. Different tools may produce different findings depending on their testing methods and supported criteria. This research supports Access Buddy by highlighting the importance of presenting clear accessibility findings, relevant WCAG references, and useful information about detected issues.

[3] Shashank Kumar, Jeevitha Shree D. V., and Pradipta Biswas,2021 evaluated multiple WCAG-based accessibility testing tools and compared their effectiveness in identifying website accessibility problems. Their research highlighted that no single automated tool can detect every type of accessibility issue. Some accessibility requirements need contextual understanding and manual verification. This study is relevant to Access Buddy because the proposed system combines automated accessibility scanning with AI-based explanations and recommendations while recognizing the limitations of automated testing.

[4] Ammar Ahmed, Margarida Fresco, Fredrik Forsberg, and Hallvard Grotli,2025 investigated the use of ChatGPT for designing and improving accessible web pages. Their study explored how AI-generated suggestions could help developers address accessibility issues according to WCAG guidelines. The research indicated that AI can assist with certain accessibility improvements, although complex issues may require additional iterations and human review. This work supports Access Buddy's approach of using AI to explain detected accessibility problems and suggest possible corrective actions.

[5] Peya Mowar, Yi-Hao Peng, Jason Wu, Aaron Steinfeld, and Jeffrey P.,2025 Bigham introduced CodeA11y, an AI-assisted approach designed to support developers in creating accessible web interfaces. Their research examined how AI coding assistants could provide accessibility-related guidance and help developers consider accessibility requirements during development. The study highlighted the value of integrating accessibility suggestions into the software development workflow.

[6] Wajdi Aljedaani and Rubel Hassan Mollik,2016 conducted a systematic literature review on the use of Large Language Models (LLMs) for web accessibility. Their research examined the application of LLMs in accessibility-related tasks, including issue identification, accessible content generation, and remediation recommendations. The review discussed the growing use of WCAG guidelines in accessibility research and highlighted the need for further evaluation of AI-based approaches. This study is relevant to Access Buddy because the system combines WCAG-related accessibility findings with AI-generated explanations and recommendations to support website accessibility improvement.

2.1 Objectives

- To provide a web-based interface for scanning a website URL for a defined set of supported accessibility issues.
- To display structured findings, including issue severity, WCAG references, affected elements, HTML details, and an accessibility score.
- To use an AI component to explain detected issues in plain language and suggest possible corrective actions.
- To store scan history and allow users to review previous testing activity.

- To support issue-status management, including Open, Fixed, and Ignored, and enable retesting to review changes.
- To provide dashboard and reporting features that make accessibility results easier to review.
- To support accessibility improvement while clearly recognizing the limits of automated testing.

III. MATERIALS AND METHODS

3.1 Methodology

Access Buddy follows a workflow that combines rule-based accessibility scanning with AI-assisted explanation. The system is organized around website input, scan execution, result presentation, recommendation generation, and issue tracking. The following steps describe the intended processing sequence.

1 Website URL input and validation: The user enters the address of the website to be evaluated. The application validates the input before initiating a scan.

2 Accessibility scan: A browser-based accessibility rule engine checks the page against the supported rules. The scan is limited to the checks implemented by the system and should not be interpreted as exhaustive WCAG conformance testing.

3 Finding collection: The system organizes the detected findings and records available details such as severity, relevant WCAG references, affected page elements, and HTML information.

4 Accessibility score and result display: The results interface presents an overall score together with individual findings. The score summarizes the scan output and does not independently establish full accessibility.

5 AI-based explanation and recommendations: Detected issue details are provided to an AI component to generate plain-language explanations and possible remediation suggestions. These suggestions should be reviewed by a developer or accessibility specialist before implementation.

6 Storage and issue tracking: Scan results and history are stored so users can review previous scans and maintain issue statuses such as Open, Fixed, and Ignored.

7 Retesting and reporting: Users can scan a website again, compare findings over time, and review or export a report were supported by the application.

3.2 System Architecture

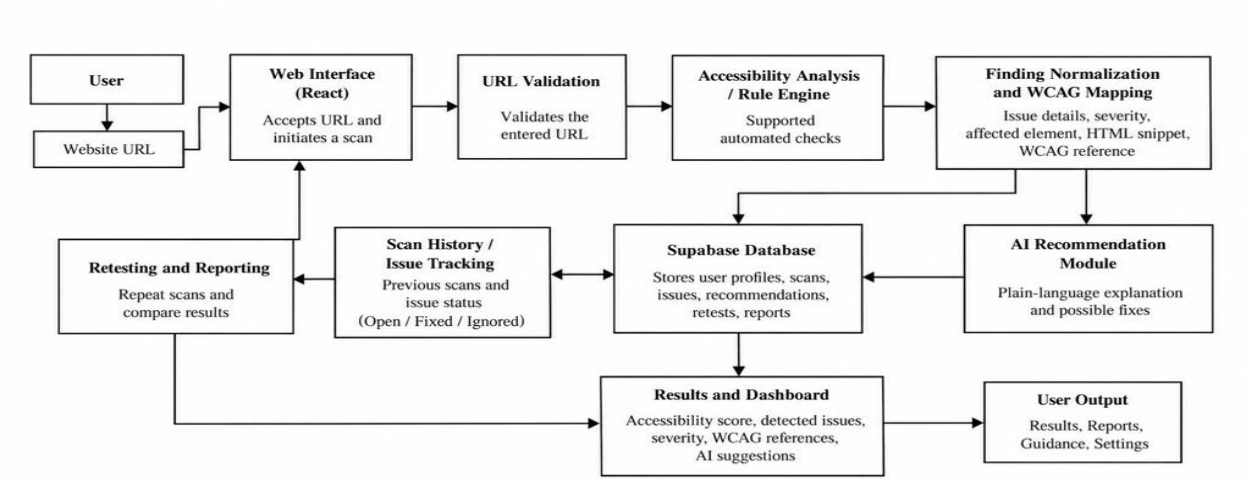


Fig 3.2: System Architecture of Access Buddy

The system architecture of Access Buddy follows a systematic flow from user input to accessibility results. The user enters a website URL through the web interface, and the system validates the URL before starting the accessibility scan. The accessibility analysis module performs supported automated checks to identify issues. The detected findings are then normalized and mapped to relevant WCAG references, including severity levels and affected HTML elements. The AI recommendation module provides plain-language explanations and suggests possible fixes. The scan results, user details, issues, and recommendations are stored in the Supabase database. Finally, the results and dashboard display the accessibility score, detected issues, severity levels, WCAG references, and AI suggestions. The system also supports scan history, issue tracking, retesting, result comparison, and reporting.

3.3 Main System Modules:

Website scanning module: Accepts a URL and executes the supported accessibility checks.

Results and severity module: Displays the score, detected issues, severity levels, WCAG references, affected elements, and HTML details.

AI recommendation module: Produces plain-language explanations and possible corrective suggestions based on scan findings.

History and issue management module: Stores previous scans and allows issues to be tracked as Open, Fixed, or Ignored.

Retesting and reporting module: Supports repeat scans and presents results for review and comparison.

Dashboard and guidance module: Provides scan statistics and access to accessibility and WCAG guidance, profile, and settings features.

3.4 Technologies and Requirements

Access Buddy uses React, TypeScript, Vite, Tailwind CSS, HTML, and CSS for the frontend. Superbase supports authentication, database storage, and backend services. Node.js and npm are used for the development environment and package management. Visual Studio Code is used for development, while a modern web browser and a stable internet connection are required to access and test the application.

IV. RESULTS AND DISCUSSION

Access Buddy integrates website accessibility scanning, AI-based recommendations, dashboard statistics, scan history, issue tracking, and retesting into one interface. Rule-based checks detect supported accessibility issues, while AI provides explanations and possible fixes with relevant WCAG references.

Scan history and retesting help users track issues after making changes. However, automated results alone cannot confirm complete accessibility. Manual testing and human review are also required. Quantitative performance metrics are not included because controlled evaluation data is unavailable in the project documentation.

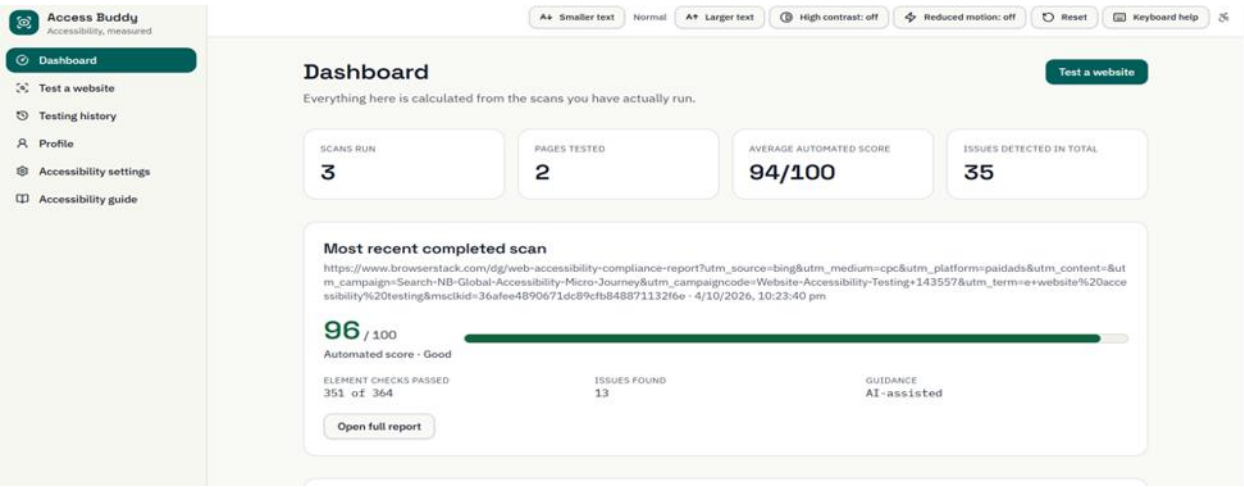


Fig 4.1 Dashboard

The dashboard provides an overview of website accessibility testing results. It displays accessibility scores, detected issues, and issue severity statistics. It helps users understand the overall accessibility status of scanned websites.

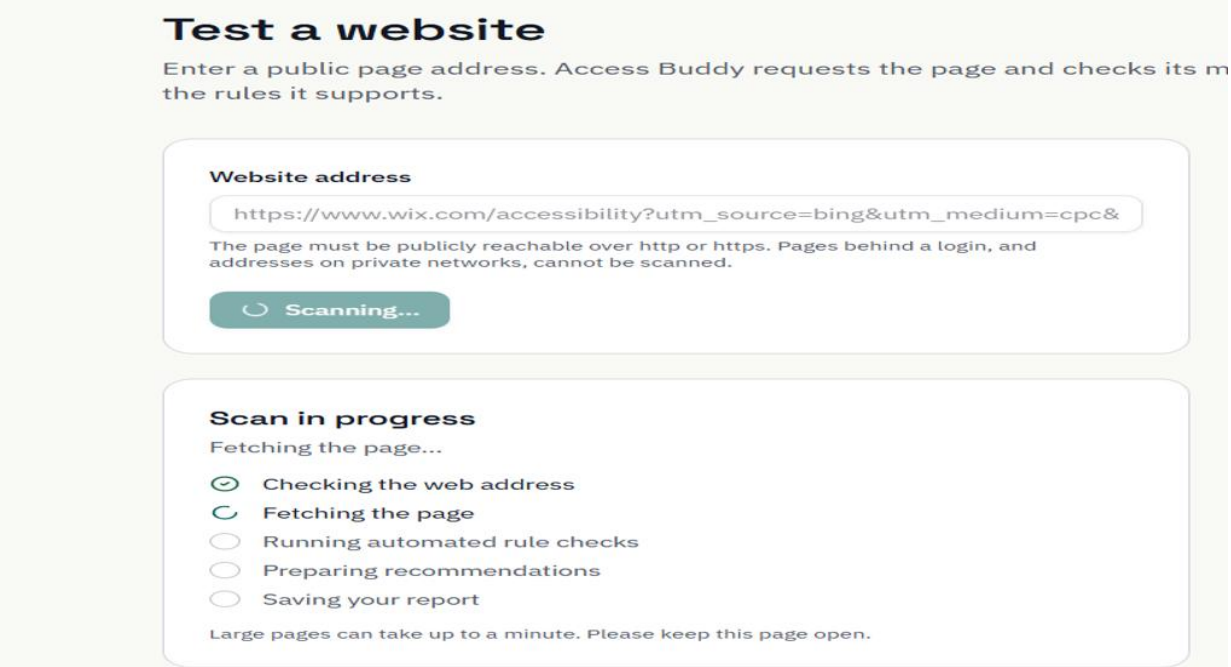


Fig 4.2 Providing Input

The input interface allows users to enter a website URL for accessibility testing. The system validates the entered URL before initiating the scan. After validation, the website is processed by the accessibility testing engine.

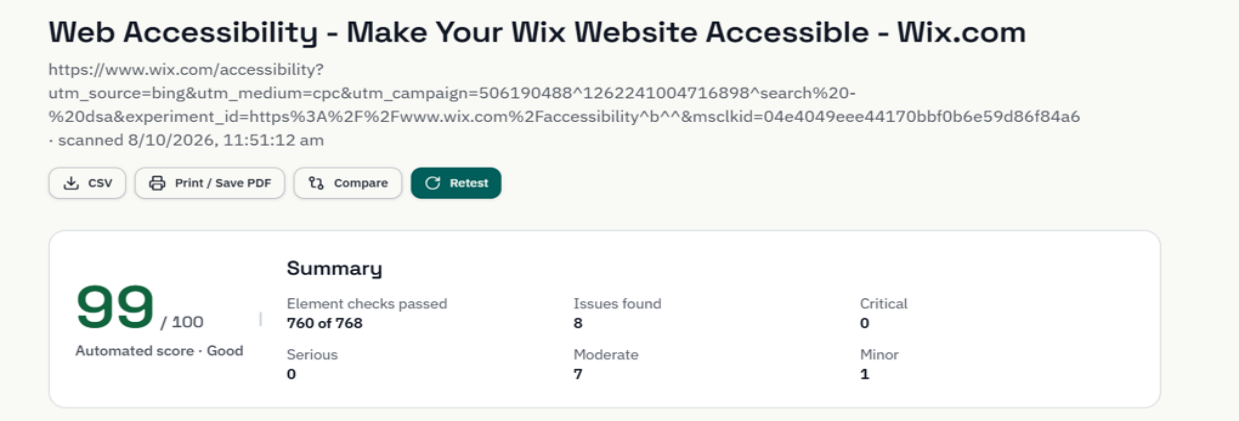


Fig 4.3 Result

The result page displays the accessibility scan findings, including detected issues, severity levels, WCAG references, and affected HTML elements. The AI-based recommendation module provides explanations and possible fixes to help users understand and address the identified issues.

V. CONCLUSION AND FUTURE SCOPE

This paper presented Access Buddy, an AI-based automated web accessibility testing and recommendation system. The system accepts a website URL, performs supported rule-based accessibility checks, presents findings with severity and WCAG-related details, and uses AI to explain issues and suggest possible corrective actions. Scan

history, issue-status tracking, retesting, dashboards, and reporting features are intended to help users organize accessibility improvement activities.

The proposed approach uses AI as a support layer rather than a replacement for established accessibility rules or human judgment. Automated checks cover only the rules implemented by the system, and AI-generated recommendations may require technical and accessibility review. Therefore, Access Buddy should be used as an aid within a broader accessibility evaluation process.

Future work may include expanding the set of supported WCAG checks, improving the transparency of the accessibility score, adding comparisons across repeated scans, and evaluating recommendations with developers and accessibility specialists. Further work could also measure detection coverage against manually reviewed test cases, assess the correctness and usefulness of AI-generated fixes, improve handling of dynamic pages, and provide clearer evidence for each reported issue. These improvements would help establish the system's reliability and clarify where automated testing is most useful.

REFERENCES

[1] Vigo, M., and G. Brajnik, "Automatic web accessibility metrics: Where we are and where we can go," Interacting with Computers, vol. 23, no. 2, pp. 137–155, 2011, doi: 10.1016/j.intcom.2011.01.001.

[2] Brajnik, G., "Comparing accessibility evaluation tools: A method for tool effectiveness," Universal Access in the Information Society, vol. 3, nos. 3–4, pp. 252–263, 2004, doi: 10.1007/s10209-004-0105-y.

[3] Kumar, S., J. Shree D. V., and P. Biswas, "Comparing ten WCAG tools for accessibility evaluation of websites," Technology and Disability, vol. 33, no. 3, pp. 163–185, 2021, doi: 10.3233/TAD-210329.

[4] Ahmed, A., M. Fresco, F. Forsberg, and H. Grotli, "From code to compliance: Assessing ChatGPT's utility in designing an accessible webpage—A case study," arXiv preprint arXiv:2501.03572, 2025. [Online]. Available: https://arxiv.org/abs/2501.03572.

[5] Mowar, P., Y.-H. Peng, J. Wu, A. Steinfeld, and J. P. Bigham, "CodeA11y: Making AI coding assistants useful for accessible web development," in Proc. 2025 CHI Conference on Human Factors in Computing Systems, 2025, doi: 10.1145/3706598.3713335.

[6] Aljedaani, W., and R. H. Mollik, "Large language models for web accessibility: A systematic literature review," in Proc. 23rd International Web for All Conference (W4A '26), 2026, doi: 10.1145/3800424.3800452.

[7] López-Gil, J.-M., and J. Pereira, "Turning manual web accessibility success criteria into automatic: An LLM-based approach," Universal Access in the Information Society, vol. 24, no. 1, pp. 837–852, 2025, doi: 10.1007/s10209-024-01108-z.

[8] Manca, M., V. Palumbo, F. Paternò, and C. Santoro, "The transparency of automatic web accessibility evaluation tools: Design criteria, state of the art, and user perception," ACM Transactions on Accessible Computing, vol. 16, no. 1, Art. no. 3, 2023, doi: 10.1145/3556979.

[9] Ara, J., and C. Sik-Lanyi, "Automated evaluation of accessibility issues of webpage content: Tool and evaluation," Scientific Reports, vol. 15, Art. no. 9516, 2025, doi: 10.1038/s41598-025-92192-5.

[10] World Wide Web Consortium (W3C), "Web Content Accessibility Guidelines (WCAG) 2.2," W3C Recommendation, 2023. [Online]. Available: https://www.w3.org/TR/WCAG22/.