An Intelligent Chrome Extension for Real-Time Website Accessibility Analysis

[1]Sarika S Jadhav, [2]Suvarna A Bahir, [3]Nikhil Patil, [4]Poonam Bhoi, [5]Aniket Parde
[12345]Department of Computer Science and Engineering
[12345]TSSM's Padmabhooshan Vasantdada Patil Institute of Technology, Pune, India

***Abstract*—With the rapid growth of web applications and digital platforms, ensuring web accessibility has become an essential requirement for inclusive user experiences. Many websites still contain accessibility barriers that prevent users with disabilities from accessing content effectively. Organizations rely on accessibility testing tools to identify compliance issues; however, many existing tools provide limited real-time analysis and lack user-friendly issue visualization capabilities. This paper proposes an Accessibility Tracker that integrates automated accessibility detection techniques with real-time webpage analysis and centralized reporting features. The system performs webpage inspection based on Web Content Accessibility Guidelines (WCAG) to detect issues such as missing alternative text, poor color contrast, improper heading structure, unlabeled form controls, and keyboard navigation barriers. Detected accessibility issues are highlighted directly on the webpage and summarized through an interactive popup dashboard developed using React (Vite). The proposed system provides a structured approach to accessibility testing, issue classification, and compliance monitoring, enabling developers to improve usability and enhance inclusive design practices efficiently. By combining automated analysis with visual feedback and scoring mechanisms, the system improves accessibility evaluation and supports the development of user-friendly web applications.**

***Index Terms*—Web Accessibility, WCAG, Chrome Extension,Automated Testing, Inclusive Design, Accessibility Evaluation, React**

I. Introduction

The increasing reliance on web applications and digital services has significantly highlighted the importance of accessibility in modern software systems. Accessibility barriers in applications can lead to difficulties in navigation, limited content understanding, and poor user experiences for individuals with disabilities. Common issues such as missing image descriptions, low color contrast, improper heading structures, and inaccessible form elements continue to affect many web applications.

Accessibility testing tools play an important role in identifying these usability barriers before websites are deployed to users. Traditionally, accessibility evaluation techniques are categorized into two major approaches: manual testing and automated analysis. Manual testing involves expert inspection and user validation to identify usability concerns, while automated analysis scans webpage elements using predefined accessibility rules and standards.

Although both approaches provide valuable insights into website accessibility, they are often implemented as separate processes that produce isolated reports. Developers frequently need to manually interpret scan results and correct issues detected by multiple tools. This lack of centralized accessibility management makes it difficult to track violations over time and prioritize improvements effectively.

To address these challenges, this paper proposes an **Accessibility Tracker** that integrates automated webpage scanning within a browser-based system. The platform collects accessibility data from

analyzed websites and presents it in a structured format, allowing developers to monitor accessibility compliance and improve user experience over time. An interactive dashboard interface enables users to initiate scans, review detected issues, and analyze accessibility scores efficiently.

The proposed system provides a practical approach for improving accessibility detection and management within web application environments. By integrating automated analysis tools with centralized issue visualization and scoring mechanisms, the system simplifies accessibility testing and improves the efficiency of inclusive web development practices.

II. Literature Survey

Several studies have investigated methods for improving web accessibility and ensuring inclusive digital experiences in modern computing environments. The increasing complexity of websites and web applications has encouraged researchers to explore automated techniques for accessibility evaluation and compliance improvement.

In their systematic review, researchers studying automated accessibility testing frameworks highlight the importance of early accessibility detection and correction strategies in web development environments [1]. Their study emphasizes that proactive accessibility mechanisms can significantly reduce usability barriers for users with disabilities.

Another study focusing on automatic detection and remediation of accessibility issues in web systems presents a comprehensive framework for identifying usability weaknesses across interactive interfaces [2]. The authors emphasize the need for automated accessibility analysis systems that can detect issues before deployment and improve user satisfaction.

Research on artificial intelligence for accessibility testing has also gained attention in recent years. A systematic review on intelligent accessibility evaluation discusses how modern AI techniques can support issue detection in webpage structures and assist developers in generating corrective suggestions [3]. Although such approaches are promising, they require quality datasets and computational resources, making them challenging to deploy in practical systems.

Keyboard navigation barriers, contrast issues, and semantic structure errors in web applications represent another important accessibility concern. A browser-based accessibility detection approach proposed in [4] demonstrates that some accessibility problems may not be directly observable during manual inspection but can appear through automated rule-based scanning processes.

Accessibility challenges also extend to emerging environments such as mobile applications and IoT interfaces. Research on accessible digital systems highlights multiple usability barriers and the need for robust accessibility detection mechanisms to protect equal access for all users and services [5].

The Web Content Accessibility Guidelines (WCAG) provide a standardized method for evaluating the accessibility compliance of digital platforms [6]. WCAG-based testing helps developers prioritize accessibility improvements based on severity, usability impact, and user needs.

Hybrid accessibility detection approaches that combine automated analysis and manual testing have also been proposed. A methodology presented in [7] demonstrates that integrating both evaluation techniques improves accessibility detection accuracy in web applications.

Furthermore, comparative studies of accessibility testing tools show that automated scanners can effectively detect common usability issues but often require centralized management systems to track problems across multiple scans [8].

Finally, research on intelligent web accessibility detection methods highlights the importance of combining multiple analysis techniques to improve accessibility assessment capabilities [9].

These studies collectively demonstrate the importance of integrating accessibility testing techniques with centralized monitoring systems, which motivates the development of the proposed **Accessibility Tracker** platform.

Table I. Comparison with Existing Accessibility Testing Tools

Feature	**WAVE**	**Axe Devtools**	Lighthouse	**Proposed System**
Automated Accessibility Scan	Yes	Yes	Yes	Yes
Real Time Issue Highlighting	Limited	Yes	No	Yes
WCAG Rule-Based Detection	Yes	Yes	Yes	Yes
Accessibility Score Dashboard	Limited	Limited	Yes	Yes
Dark/Light Theme Interface	No	No	No	Yes
Centralized Issue Tracking	No	No	No	Yes

III. Proposed System

The proposed **Accessibility Tracker** is designed to provide an integrated solution for detecting, analyzing, and managing accessibility issues in web applications. The system combines automated webpage scanning with centralized accessibility monitoring to improve the efficiency of accessibility evaluation.

The platform performs accessibility detection using rule-based techniques based on Web Content Accessibility Guidelines (WCAG). The analysis examines webpage elements to identify potential usability issues such as missing alternative text, improper heading hierarchy, low color contrast, unlabeled form fields, and keyboard navigation barriers.

Automated accessibility scanning in the proposed system is performed using browser extension technology, which allows direct interaction with webpage components in real time. The extension automatically inspects HTML structures, styles, and interactive elements to detect accessibility violations. This approach enables fast analysis of websites without requiring external software installations.

Once accessibility issues are detected, the system organizes the scan results and presents them through an interactive popup interface. The interface provides efficient visualization of detected problems, including issue type, severity level, affected element, and improvement suggestions. Users can easily review and understand accessibility gaps.

An interactive dashboard built using **React (Vite)** allows users to monitor accessibility reports and overall compliance scores. Through this dashboard, users can initiate scans, analyze accessibility trends, and review historical results. The integration of automated scanning with centralized issue visualization and scoring creates a comprehensive accessibility management environment.

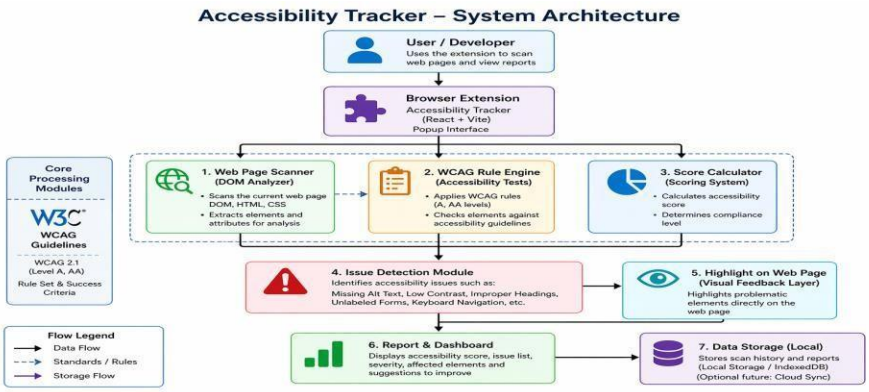

Fig. 1. System Architecture of the Accessibility Tracker Chrome Extension

IV. System Architecture

The architecture of the proposed system consists of multiple layers that work together to detect, analyze, and manage accessibility issues in web applications.

The **user interface layer** provides an interactive environment where users and developers interact with the system. The interface enables users to initiate webpage scans and visualize accessibility reports through an easy-to-use popup dashboard.

The **application control layer** processes user requests and coordinates interactions between system components. When a scan request is received, this layer triggers the appropriate analysis modules and organizes the results for reporting and visualization.

The **accessibility scanning layer** integrates automated webpage inspection modules. The scanner examines webpage structure, HTML elements, CSS properties, and interactive components to detect accessibility barriers such as missing alternative text, improper labels, and contrast issues.

The **rule engine layer** applies Web Content Accessibility Guidelines (WCAG) standards to evaluate webpage compliance. This layer checks detected elements against accessibility success criteria and generates issue severity levels.

The **data storage layer** stores scan history, accessibility reports, and user preferences using local browser storage or IndexedDB. The centralized storage allows issue records to be stored and retrieved efficiently across multiple scans.

Finally, the **analytics and visualization layer** presents accessibility insights through an interactive dashboard interface, allowing users to monitor accessibility scores, review issue severity levels, and track usability improvements over time.

V. Methodology

The proposed **Accessibility Tracker** follows a structured methodology that integrates accessibility detection, centralized issue management, and visualization of usability insights. The methodology focuses on combining automated webpage analysis techniques to improve the detection and monitoring of accessibility barriers in web applications.

The process begins with the identification of a target website or webpage that requires accessibility assessment. The user provides a webpage URL or opens the target site through the browser interface. Once the target is provided, the system initiates the accessibility scanning process by generating a scan request. This request is forwarded to the analysis modules responsible for evaluating the webpage for potential accessibility issues.

The accessibility detection process is performed using automated rule-based techniques. The scanner examines webpage structure to identify issues such as missing alternative text, unlabeled form controls, improper heading hierarchy, insufficient color contrast, and keyboard navigation barriers. Automated analysis is useful for detecting accessibility problems early and helps developers improve usability before deployment.

Real-time accessibility analysis is conducted using browser extension modules which interact directly with the running webpage. The scanner inspects DOM elements, CSS styles, and interactive components to detect barriers while the webpage is active. The combination of structural analysis and live webpage inspection improves accessibility detection coverage and allows the system to identify issues that may not be detectable through a single analysis approach. Similar hybrid approaches combining multiple testing techniques have been shown to improve accessibility evaluation effectiveness in web applications [7].

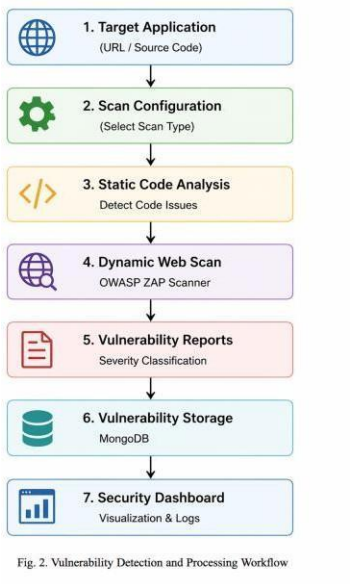

Fig. 2. Web Accessibility Tracking Workflow

Once accessibility issues are detected, the scanning modules generate structured accessibility reports containing details such as issue type, affected element, severity level, and recommended improvement steps. These accessibility records are then processed and stored in local browser storage or IndexedDB. The use of browser-based storage allows efficient storage of semi-structured accessibility data and enables flexible retrieval of historical scan results.

The severity of detected accessibility issues is categorized using established accessibility evaluation standards such as WCAG compliance levels (A, AA, AAA). This classification allows developers to prioritize issues that require immediate correction based on usability impact and accessibility importance.

After accessibility data is stored, the system provides visualization and reporting capabilities through an interactive dashboard built using **React (Vite)**. The dashboard allows users to monitor accessibility statistics, analyze issue distributions, and review historical scan results. Centralized visualization of accessibility data enables organizations to track improvement trends and evaluate website usability over time.

The overall methodology therefore integrates automated webpage scanning, centralized accessibility issue storage, and visualization mechanisms into a unified workflow. This approach enables automated accessibility detection while providing developers with the necessary tools to analyze and manage accessibility issues effectively. The integration of automated testing modules with centralized accessibility management addresses limitations identified in previous accessibility studies and supports more efficient website improvement processes.

VI. Conclusion

This paper presented an **Accessibility Tracker** designed to improve the efficiency of accessibility detection and evaluation in web applications. The proposed system integrates automated webpage scanning techniques with centralized issue storage, real-time highlighting, and visualization capabilities.

Accessibility analysis is performed using browser extension modules, while accessibility data is stored in local browser storage or IndexedDB and visualized through a React (Vite) dashboard. The integration of these technologies enables efficient accessibility monitoring and improves the ability of developers to identify and resolve usability barriers.

The proposed platform provides a structured approach to accessibility testing and management and can serve as a foundation for further research in automated web accessibility evaluation systems.

References

[1] A Survey on Automated Web Accessibility Testing Tools and Techniques.
[2] Web Accessibility Evaluation Using Browser-Based Extensions.
[3] Artificial Intelligence Approaches for Accessibility Issue Detection in Websites.
[4] Detection of Web Accessibility Violations Using Automated Rule Engines.
[5] Accessibility Challenges in Mobile and IoT Interfaces: A Review.
[6] Web Content Accessibility Guidelines (WCAG) 2.1 Specification.
[7] Combining Manual and Automated Accessibility Testing for Better Compliance Results.
[8] Comparative Study of Accessibility Scanning Tools for Web Platforms.
[9] Intelligent Accessibility Detection Based on Machine Learning: A Comparative Study.