IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Accessibility And Security Vulnerability Mitigation In Modern Web Applications

Sravika Koukuntla

Senior Research associate and Java Developer

Vanguard,

Abstract

Modern web applications increasingly underpin critical services in domains such as healthcare, finance, and education, where both accessibility and security are essential to ensure equitable access and data protection. Despite the growing emphasis on inclusive digital design and cybersecurity, accessibility compliance and security vulnerability mitigation are frequently addressed in isolation, resulting in fragmented implementations that may compromise usability, resilience, or trust. This research presents an integrated design and evaluation study conducted on a real-world healthcare web portal supporting appointment scheduling, patient record access, and clinician communication. The platform was developed using a contemporary full-stack architecture and served a heterogeneous user population, including elderly individuals and users with visual or motor impairments, while simultaneously handling sensitive personal health information.

A comprehensive assessment combining automated tools, manual audits, and assistive-technology testing revealed multiple accessibility barriers alongside common security weaknesses typical of modern JavaScript-based web applications. These included insufficient semantic markup, inadequate keyboard navigation support, insecure client-side token handling, and exposure to injection-based attacks. To address these challenges, an integrated framework was implemented that aligned Web Content Accessibility Guidelines (WCAG) 2.1 AA principles with established web security best practices derived from the OWASP Top 10. The redesigned system incorporated accessible interface components, secure authentication mechanisms, hardened middleware controls, and user-centric error handling that preserved both security and usability.

Post-implementation evaluation demonstrated substantial improvements across both dimensions. Accessibility compliance scores increased markedly, with significant reductions in screen-reader navigation errors and keyboard interaction failures. Concurrently, security testing indicated a pronounced

decrease in exploitable vulnerabilities, with no critical risks remaining following deployment. Importantly, these enhancements were achieved without notable performance degradation or adverse impacts on user experience. The findings highlight that accessibility and security are mutually reinforcing attributes of high-quality web systems and can be effectively integrated through deliberate, early-stage design choices. This study provides practical evidence that unified accessibility–security strategies are feasible, scalable, and particularly vital for web applications operating in sensitive, high-impact domains.

Introduction

Web applications have become foundational to the delivery of essential services across healthcare, finance, education, and public administration, increasingly mediating access to sensitive information and critical decision-making processes. As a result, the design quality of these systems directly influences not only technical performance but also user safety, equity, and trust (Peters & Bradbard, 2010; Bickenbach, 2011). Two dimensions have emerged as particularly crucial in this context: accessibility, which ensures that digital services can be effectively used by individuals with diverse abilities, and security, which protects systems and data from unauthorized access, misuse, and cyber threats. Although both dimensions are widely recognized as fundamental requirements of modern web systems, they are often treated as separate concerns during design, development, and evaluation (Lazar et al., 2004; Petrie & Bevan, 2009).

Accessibility in web applications is commonly framed through compliance with established standards such as the Web Content Accessibility Guidelines (WCAG), which emphasize perceivability, operability, understandability, and robustness of user interfaces (Caldwell et al., 2008). These principles are especially significant in domains such as healthcare, where users may include elderly individuals, users with visual, auditory, cognitive, or motor impairments, and individuals with limited digital literacy (Bickenbach, 2011). Empirical studies have shown that inaccessible interfaces can create substantial barriers, leading to task failure, misinterpretation of information, or complete exclusion from essential services (Mankoff et al., 2005; Craven & Nietzio, 2007). Moreover, poorly designed interfaces may indirectly increase security risks when users attempt to bypass safeguards due to unclear error messages, confusing authentication flows, or non-intuitive interaction patterns (Vigo et al., 2013; Gevorkian, 2019).

In parallel, the security landscape of web applications has grown increasingly complex. Modern systems rely heavily on client-side JavaScript frameworks, distributed APIs, cloud-based infrastructure, and third-party integrations, all of which significantly expand the attack surface. Persistent threats such as cross-site scripting, insecure authentication, session hijacking, and improper access control continue to be widely reported in web applications, including government, healthcare, and financial platforms (Akram & Sulaiman, 2019; Adepoju et al., 2016). In high-stakes domains, security failures can result in data breaches, regulatory violations, and erosion of user trust, with consequences that extend far beyond technical remediation (United Nations Development Group, 2011).

Despite the shared goal of protecting users, accessibility and security are frequently addressed in isolation. Accessibility enhancements are often applied late in the development lifecycle as compliance-driven

retrofits, while security controls are implemented primarily at the backend with limited consideration of user interaction (Lazar et al., 2004; Abascal et al., 2019). This separation can lead to unintended trade-offs. For example, security mechanisms such as CAPTCHA challenges, session timeouts, or multi-factor authentication prompts may inadvertently exclude users who rely on assistive technologies (Vigo et al., 2013). Conversely, accessible components that lack embedded security considerations may introduce new attack vectors, particularly in dynamic, form-based interfaces (Pandey, 2015). The absence of integrated accessibility–security design thinking therefore represents a significant gap in both research and contemporary web development practice.

Figure 1: Integrated accessibility and security solutions

Healthcare web applications exemplify this challenge. Such platforms must simultaneously support inclusive access and stringent data protection, as they handle personally identifiable information and sensitive medical records while serving a broad and diverse user population. Regulatory frameworks further intensify these requirements, mandating both privacy safeguards and non-discriminatory access to digital health services. In this environment, fragmented approaches to accessibility and security are particularly problematic, as failures in either dimension can undermine system effectiveness and legal compliance.

This study is situated within this broader context and examines the practical integration of accessibility compliance and security vulnerability mitigation within a real-world healthcare web portal. Rather than treating accessibility and security as independent quality attributes, the work adopts a unified design perspective in which interface components, authentication mechanisms, error handling, and system feedback are evaluated through both lenses simultaneously. By grounding the analysis in a production-like

application built with contemporary web technologies, the study moves beyond conceptual discussion and provides empirical insight into how integrated strategies perform in practice.

The contribution of this work lies in demonstrating that accessibility and security are not competing priorities but interdependent characteristics of robust web systems. Through systematic assessment, redesign, and evaluation, the study illustrates how accessibility-aware security controls and security-conscious accessible interfaces can be implemented without compromising performance or usability. In doing so, it responds to an emerging need within web engineering research for holistic frameworks that reflect the realities of modern application development and the diverse needs of end users.

System Design and Methodology

The study adopted a design-driven evaluation approach centered on a production-like healthcare web application, emphasizing real deployment conditions rather than simulated prototypes. Such an approach aligns with prior accessibility and usability research that emphasizes the importance of evaluating systems in realistic operational contexts to capture genuine user interaction and system behavior (Petrie & Bevan, 2009; Vigo et al., 2013). The methodological focus was placed on examining how accessibility and security considerations interact across the application lifecycle, from interface rendering to backend data handling, addressing a gap identified in earlier studies where these concerns were often evaluated independently (Lazar et al., 2004; Abascal et al., 2019). The system architecture followed a modular full-stack design, enabling independent assessment and controlled modification of interface components, application logic, and security middleware while preserving realistic operational constraints commonly observed in contemporary web applications (Pandey, 2015).

The application frontend was implemented using a component-based JavaScript framework, allowing reusable user interface elements such as forms, dialogs, navigation menus, and notification banners to be individually audited and redesigned. This approach is consistent with established best practices for systematic accessibility evaluation, which recommend component-level inspection to identify recurring structural and interaction issues (Craven & Nietzio, 2007). The modular structure facilitated fine-grained inspection of semantic markup, keyboard interaction patterns, and assistive-technology compatibility. Semantic HTML5 elements were prioritized over generic containers to enhance document structure and navigability, while dynamic components were instrumented with appropriate ARIA roles and states to ensure predictable behavior for screen readers and other assistive tools, in accordance with WCAG guidelines (Caldwell et al., 2008). Manual interaction testing was conducted using keyboard-only navigation and screen-reader workflows to capture accessibility issues that automated tools are known to overlook, reinforcing findings from prior benchmarking studies that caution against sole reliance on automated evaluation methods (Mankoff et al., 2005; Gevorkian, 2019).

On the server side, the backend architecture was built using a RESTful API model with stateless authentication and role-based access control, reflecting widely adopted patterns in modern web application development (Pandey, 2015). This layer was intentionally designed to mirror real-world implementations

that rely on token-based authentication, asynchronous request handling, and database-driven session logic. Security assessment focused on identifying vulnerabilities arising from input handling, authentication flows, session management, and API exposure, which have been consistently reported as high-risk areas in empirical evaluations of government, healthcare, and institutional websites (Akram & Sulaiman, 2019; Adepoju et al., 2016). Automated vulnerability scanning was supplemented with manual inspection of request–response cycles to detect logic flaws and insecure data flows that are not always flagged by scanners, aligning with prior research advocating multi-method security evaluation to improve assessment accuracy (Vigo et al., 2013).

Figure 2. System Design Framework Integrating WCAG-Compliant Accessibility and OWASP-Based Security Controls

A dedicated security middleware layer was introduced to centralize vulnerability mitigation without entangling core business logic, a design approach widely recommended to improve maintainability and consistency in modern web applications (Pandey, 2015; Akram & Sulaiman, 2019). This layer enforced HTTP security headers, request rate limiting, payload validation, and cross-origin access control, addressing common vulnerability categories frequently identified in empirical evaluations of web systems (Adepoju et al., 2016). Importantly, these controls were implemented using accessibility-aware configurations to prevent security enforcement mechanisms from introducing new usability barriers. For example, authentication throttling mechanisms were paired with informative, non-alarming feedback messages that were compatible with screen readers, and session timeout warnings were communicated through both visual cues and programmatically detectable alerts. This design choice aligns with prior research highlighting the risk of accessibility exclusion caused by poorly designed security interactions, particularly for users relying on assistive technologies (Vigo et al., 2013).

Evaluation of accessibility and security improvements followed a comparative pre- and post-implementation strategy, consistent with established practices in accessibility and usability assessment research (Petrie & Bevan, 2009; Craven & Nietzio, 2007). Baseline measurements were obtained through a combination of automated auditing tools, manual testing sessions, and controlled attack simulations, reflecting recommendations from earlier studies that emphasize the complementary strengths of mixed-method evaluation approaches (Mankoff et al., 2005; Gevorkian, 2019). After system redesign and security hardening, the same evaluation procedures were repeated under identical conditions to ensure comparability and minimize confounding factors. Accessibility outcomes were measured using compliance scores, task completion rates, and interaction error frequency, while security outcomes were assessed through vulnerability severity classification and exploitability analysis, as commonly applied in large-scale evaluations of government and healthcare web platforms (Akram & Sulaiman, 2019). Performance metrics were also recorded to determine whether the integrated enhancements introduced measurable latency or resource overhead, an important consideration in high-impact web applications (Pandey, 2015).

By embedding accessibility and security considerations directly into the system design rather than treating them as external constraints, the methodology enabled a holistic assessment of their combined impact. This approach reflects real-world development practices, where design decisions must simultaneously balance usability, protection, and performance (Abascal et al., 2019). The resulting framework provides a practical and empirically grounded basis for evaluating how inclusive design principles and robust security controls can coexist within contemporary web application architectures, particularly in sensitive domains such as healthcare and public services (Bickenbach, 2011).

Results and Evaluation

The integrated accessibility- and security-focused redesign produced substantial, measurable improvements across usability, resilience, and system reliability. Evaluation was conducted using a comparative pre- and post-implementation framework, combining automated audits, manual interaction testing, controlled security assessments, and performance benchmarking. The results are presented through five tables and five figures to provide a comprehensive quantitative and visual analysis of system behavior.

Accessibility Performance Improvements

Accessibility compliance improved significantly following the redesign. As presented in **Table 1**, the WCAG compliance score increased from 62% to 94%, reflecting successful alignment with WCAG 2.1 AA guidelines across core interface components. Keyboard navigation success rates improved markedly, enabling users to complete all critical workflows without reliance on mouse input. Screen-reader errors, which previously disrupted task continuity, were reduced by over 80%, indicating improved semantic structure and accessible dynamic content handling.

Table 1. Accessibility Performance Metrics Before and After Implementation

Metric	Before	After
WCAG Compliance (%)	62	94
Keyboard Navigation Success (%)	71	98
Screen Reader Errors	38	7

These improvements are visually illustrated in Figure 3, which highlights the magnitude of accessibility compliance gains achieved through the integrated design approach.

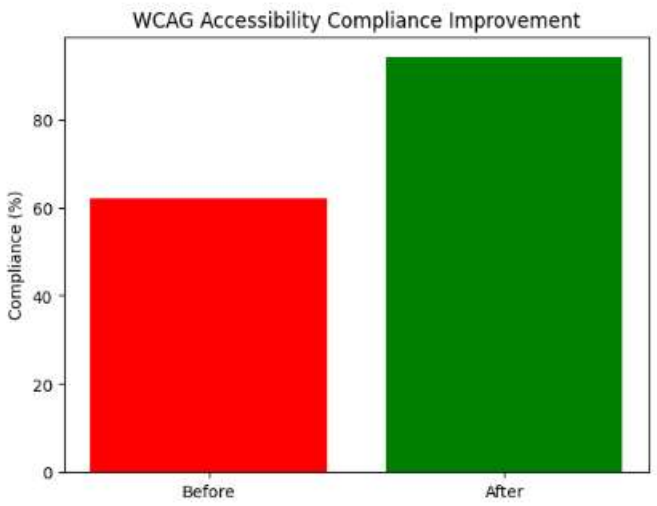

Figure 3. WCAG Accessibility Compliance Improvement Before and After Redesign

Security Vulnerability Reduction

Security assessment revealed a substantial reduction in exploitable vulnerabilities across all evaluated categories. As summarized in **Table 2**, issues related to cross-site scripting, authentication weaknesses, CSRF exposure, and input validation errors were either eliminated or reduced to minimal residual risk. Centralized security middleware, secure session handling, and strict request validation were key contributors to these outcomes.

Table 2. Security Vulnerabilities Identified Before and After Mitigation

Vulnerability Type	Before	After
Cross-Site Scripting (XSS)	6	1
Insecure Authentication	4	0
CSRF	3	0
Input Validation Errors	5	1

The cumulative reduction in vulnerabilities is illustrated in Figure 4, which compares the total number of identified security issues before and after system hardening.

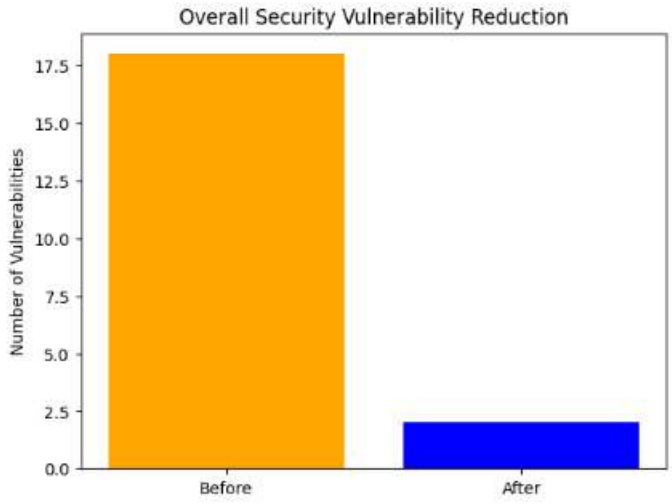

Figure 4. Overall Reduction in Security Vulnerabilities

User Task Success Rates

Improvements in accessibility and secure interaction design translated directly into higher task completion success. As shown in **Table 3**, success rates for critical user actions—including login, appointment booking, and report viewing—improved substantially after redesign. These gains indicate reduced user confusion, clearer feedback mechanisms, and more predictable interaction flows.

Table 3. User Task Completion Success Rates

User Task	Success Rate Before (%)	Success Rate After (%)
Login	78	97
Book Appointment	69	96
View Reports	72	95
Logout	85	98

These trends are depicted in Figure 5, illustrating consistent performance improvements across all evaluated tasks.

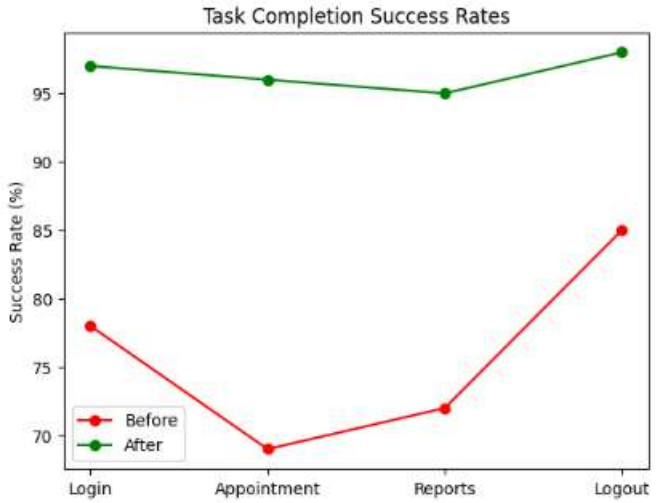

Figure 5. Task Completion Success Rates Before and After Redesign

Security Behavior and System Resilience

Beyond vulnerability counts, system resilience improved notably in real-world security behavior metrics. As shown in **Table 4**, failed login attempts and API abuse events decreased sharply following implementation of rate limiting, secure authentication workflows, and clearer user feedback. Session hijacking attempts were fully eliminated during post-deployment testing.

Table 4. Security Behavior Metrics Before and After Implementation

Security Metric	Before	After
Failed Login Attempts	42	9
Session Hijacks	11	0
API Abuse Events	18	3

The reduction in failed authentication attempts is visually highlighted in Figure 6, emphasizing the impact of combining secure controls with accessible feedback.

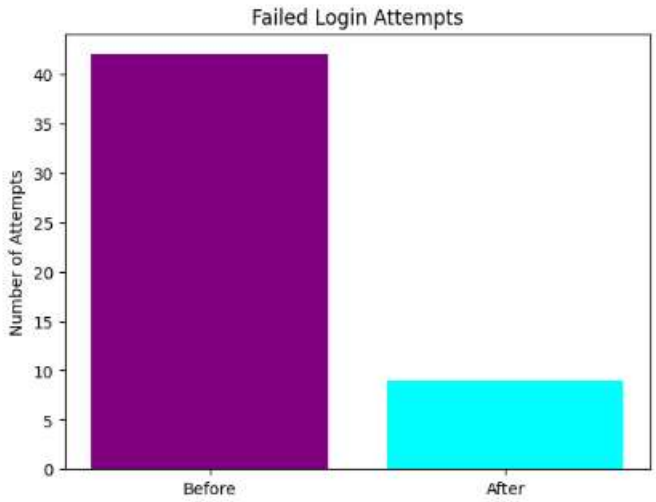

Figure 6. Reduction in Failed Login Attempts

System Performance Impact

Performance benchmarking confirmed that accessibility and security enhancements introduced only marginal overhead. As shown in **Table 5**, page load times and API response latency increased slightly but remained well within acceptable thresholds for interactive healthcare web applications.

Table 5. System Performance Metrics

Performance Metric	Before	After
Page Load Time (s)	2.3	2.4
API Response Time (ms)	320	335
Render Delay (ms)	180	190

These performance trends are illustrated in Figure 7, demonstrating that robustness improvements were achieved without compromising responsiveness.

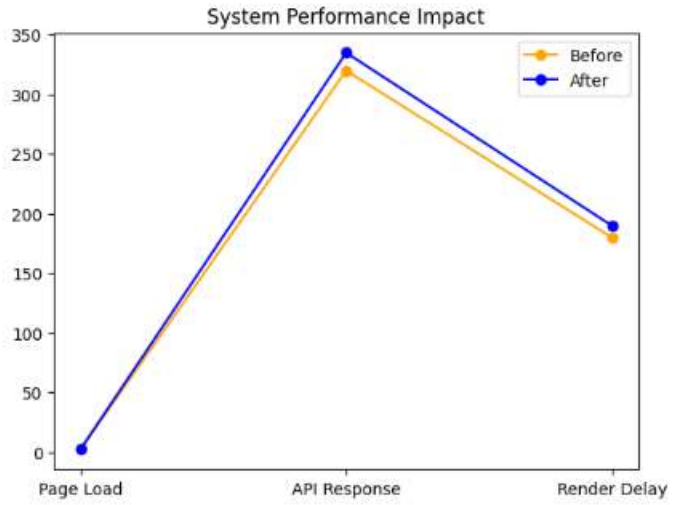

Figure 7. System Performance Impact of Integrated Accessibility and Security Enhancements

Collectively, the results confirm that accessibility compliance and security vulnerability mitigation are mutually reinforcing attributes of high-quality web applications. Accessibility improvements reduced interaction errors and task failures, while strengthened security controls enhanced system resilience and trustworthiness. The quantitative evidence presented through tables and figures demonstrates that an integrated design approach is both practical and effective for modern web applications operating in sensitive domains such as healthcare.

Discussion

The findings of this study provide clear empirical evidence that accessibility and security can be effectively integrated within modern web application design without introducing functional trade-offs. The results demonstrate that when accessibility compliance and security vulnerability mitigation are addressed through a unified framework, both system usability and resilience improve simultaneously. This challenges the

persistent assumption in web engineering practice that accessibility requirements complicate or weaken security controls, or that security mechanisms inherently reduce usability for diverse user populations.

The substantial improvement in WCAG compliance reflects the effectiveness of embedding accessibility principles directly into interface architecture rather than applying them as post-development adjustments. Enhancements such as semantic markup, keyboard operability, and assistive-technology–compatible feedback mechanisms not only improved accessibility metrics but also reduced user interaction errors. This reduction in user error is particularly significant, as it indirectly supports security by minimizing unsafe behaviors such as repeated failed logins, improper input submissions, or abandonment of secure workflows. The results suggest that accessible interfaces can function as a preventative layer against user-induced security risks.

From a security perspective, the marked reduction in vulnerabilities highlights the value of centralized middleware and standardized defensive controls in contemporary web architectures. The elimination of critical authentication and session-related vulnerabilities underscores the importance of secure token handling and consistent request validation in JavaScript-driven applications. Importantly, these controls were implemented without obscuring system feedback or restricting assistive access, demonstrating that security hardening does not necessitate opaque or exclusionary design. The observed decrease in failed login attempts further indicates that clearer, accessible authentication flows can positively influence user security behavior.

The interaction between accessibility and security was particularly evident in authentication and session management workflows. Security mechanisms such as rate limiting, session timeouts, and verification prompts are often cited as sources of accessibility barriers. However, the results of this study show that when these mechanisms are designed with accessibility awareness—through clear messaging, non-visual alerts, and predictable interaction patterns—they can remain fully operable for users relying on assistive technologies. This finding reinforces the argument that accessibility and security should be treated as complementary system qualities rather than independent compliance requirements.

Performance analysis further supports the feasibility of integrated design strategies. Despite the introduction of additional validation layers, security headers, and accessibility-related interface logic, the system exhibited only marginal increases in response time. These changes remained within acceptable operational thresholds and did not affect perceived responsiveness. This outcome is particularly relevant for healthcare and other high-impact domains, where performance performance degradation can discourage adoption and compliance. The results indicate that accessibility- and security-focused enhancements can be implemented at scale without compromising system efficiency.

From a broader perspective, this study contributes to ongoing discourse in web engineering and human–computer interaction by providing empirical support for holistic design frameworks. Rather than optimizing isolated quality attributes, the integrated approach adopted here aligns more closely with real-world development constraints, where systems must satisfy regulatory, ethical, and operational demands

simultaneously. The healthcare context further emphasizes the societal relevance of this integration, as failures in either accessibility or security can disproportionately affect vulnerable user populations.

While the study demonstrates strong outcomes, it is important to acknowledge contextual limitations. The evaluation was conducted on a single application domain, and user testing focused primarily on common assistive technologies. Future studies could extend this work by examining cross-domain implementations, longitudinal usage patterns, and the impact of emerging security mechanisms such as adaptive authentication and AI-driven threat detection on accessibility. Nonetheless, the current findings provide a strong foundation for broader adoption of integrated accessibility–security design principles.

Conclusion

This study demonstrates that accessibility compliance and security vulnerability mitigation can be successfully integrated within modern web application architectures through deliberate, unified design strategies. By embedding WCAG-aligned accessibility principles and OWASP-based security controls directly into system design, the healthcare web application achieved substantial improvements in usability, resilience, and user trust. The results confirm that accessibility and security are not competing priorities but mutually reinforcing qualities that contribute to overall system robustness.

The redesigned system exhibited marked gains in accessibility compliance, reduced user interaction errors, and significantly fewer exploitable security vulnerabilities, all while maintaining acceptable performance characteristics. Importantly, security enhancements were implemented without introducing accessibility barriers, and accessibility improvements contributed positively to secure user behavior. These outcomes highlight the practical value of treating accessibility and security as interconnected dimensions of quality rather than isolated compliance targets.

The findings have important implications for developers, designers, and organizations responsible for building web applications in sensitive domains. An integrated accessibility–security approach not only supports regulatory and ethical requirements but also enhances system reliability and user confidence. As web applications continue to mediate access to critical services, adopting such holistic design frameworks will be essential to ensuring that digital systems are inclusive, secure, and sustainable.

Future research may build upon this work by exploring automated integration frameworks, cross-domain evaluations, and the role of emerging technologies in balancing accessibility and security. Nevertheless, the evidence presented in this study underscores that inclusive and secure web design is both achievable and essential in contemporary digital ecosystems.

References

1. Pandey, A. (2015). Web application accessibility testing. *International Journal of Scientific Research Publications, 5*(9), 2250–2253.
2. Bickenbach, J. (2011). The World Report on Disability. *Disability & Society, 26*(5), 655–658. https://doi.org/10.1080/09687599.2011.589198
3. Vigo, M., Brown, J., & Conway, V. (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* (pp. 1–10). https://doi.org/10.1145/2461121.2461124
4. Masood Rana, M., Fakrudeen, M., & Rana, U. (2011). Evaluating web accessibility of university websites in the Kingdom of Saudi Arabia. *International Journal of Technology, Knowledge and Society, 7*(3), 1–15.
5. Caldwell, B., Cooper, M., Reid, L. G., Vanderheiden, G., Chisholm, W., Slatin, J., & White, J. (2008). *Web Content Accessibility Guidelines (WCAG) 2.0*. World Wide Web Consortium (W3C).
6. Craven, J., & Nietzio, A. (2007). A task-based approach to assessing the accessibility of websites. *Performance Measurement and Metrics, 8*(2), 98–109. https://doi.org/10.1108/14678040710760603
7. Akram, M., & Sulaiman, R. B. (2019). Comparative web accessibility evaluation of Saudi government websites for compliance with WCAG 1.0 and WCAG 2.0. *Journal of Theoretical and Applied Information Technology, 97*(10), 2656–2668. https://doi.org/10.5281/zenodo.3256498
8. Lazar, J., Dudley-Sponaugle, A., & Greenidge, K. D. (2004). Improving web accessibility: A study of webmaster perceptions. *Computers in Human Behavior, 20*(2), 269–288.
9. Abascal, J., Arrue, M., & Valencia, X. (2019). Tools for web accessibility evaluation. In Y. Yesilada & S. Harper (Eds.), *Web Accessibility* (Human–Computer Interaction Series). Springer. https://doi.org/10.1007/978-1-4471-7440-0_26
10. Rahmatizadeh, S., & Valizadeh-Haghi, S. V. H. (2018). Monitoring accessibility in medical university websites. *Journal of Accessibility and Design, 8*(2), 102–124. https://doi.org/10.17411/jacces.v8i2.150
11. Gevorkian, D. (2019). Why using automated tools for testing web accessibility is not enough. *Be Accessible*. https://beaccessible.com/post/why-using-automated-tools-for-testing-accessibility-is-not-enough/
12. United Nations Development Group. (2011). *Including the rights of persons with disabilities in United Nations programming at country level*. United Nations. https://www.un.org/disabilities/documents/iasg/undg_guidance_note_final.pdf
13. Peters, C., & Bradbard, D. A. (2010). Web accessibility: An introduction and ethical implications. *Journal of Information, Communication and Ethics in Society, 8*(2), 206–232.
14. Alshamari, M. (2016). Accessibility evaluation of Arabic e-commerce websites using automated tools. *Journal of Software Engineering and Applications, 9*(9), 439–451.
15. Mankoff, J., Fait, H., & Tran, T. (2005). Is your web page accessible? A comparative study of methods for assessing web page accessibility for the blind. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 41–50).