

Article

When Cybersecurity Meets Accessibility: A Holistic Development Architecture for Inclusive Cyber-Secure Web Applications and Websites

George Alex Stelea ¹, Livia Sangeorzan ² and Nicoleta Enache-David ^{2,*}¹ Department of Electronics and Computers, Transilvania University of Braşov, 500036 Braşov, Romania; george.alex@stelea.eu² Department of Mathematics and Informatics, Transilvania University of Braşov, 500036 Braşov, Romania; sangeorzan@unitbv.ro

* Correspondence: nicoleta.enache@unitbv.ro; Tel.: +40-746-175-801

Abstract: In today's technology-centric era, digital transformation has made accessibility and cybersecurity critical components for developing inclusive web applications that are essential in daily operations. However, current development models often treat accessibility and cybersecurity separately, leading to solutions that may excel in one area while lacking in the other. This paper addresses this gap by proposing the Holistic Security and Accessibility Layered Architecture (HSALA), a novel architectural model that integrates accessibility and cybersecurity throughout the development process. HSALA consists of five interconnected layers, each paired with a corresponding security layer, ensuring that both accessibility and security are embedded at every level. We conduct a theoretical and analytical case study comparing HSALA with the traditional three-layered architecture, demonstrating HSALA's superiority in addressing contemporary challenges related to accessibility, cybersecurity, scalability, and user experience. The findings indicate that HSALA provides a robust framework for developing secure and inclusive web applications, promoting a more equitable digital ecosystem. We outline a roadmap for future work, including empirical validation and practical implementation strategies. HSALA represents a significant advancement in web application development, fostering inclusivity and trust in an increasingly digital society.



Academic Editors: Ugo Fiore and Steven Furnell

Received: 9 December 2024

Revised: 24 January 2025

Accepted: 27 January 2025

Published: 5 February 2025

Citation: Stelea, G.A.; Sangeorzan, L.; Enache-David, N. When Cybersecurity Meets Accessibility: A Holistic Development Architecture for Inclusive Cyber-Secure Web Applications and Websites. *Future Internet* **2025**, *17*, 67. <https://doi.org/10.3390/fi17020067>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: web accessibility; cybersecurity; digital inclusion; digital transformation; HSALA; inclusive web applications; web application architecture; accessibility and security integration

1. Introduction

1.1. Context and Motivation

In the modern technology-centric era, digital transformation has made accessibility and cybersecurity critical components for the creation of any inclusive platforms which are essential in day-to-day operations [1]. With both the government and business sectors having to find ways to put most of their services online, the call for accessibility and cybersecurity has grown exponentially. The accessibility and cybersecurity of these platforms not only improve user experience but also protect sensitive information while promoting inclusion and trust. Accessible and cyber-safe web interfaces are also crucial for children, students, senior citizens and people who have disabilities, due to the specific needs and vulnerabilities of these groups. Optimizing accessibility and security not only enhances

the overall user experience but also serves as a safeguard against different dangers and risks associated with present-day digital interactions. Nowadays, a wide range of digital technologies is employed by businesses and institutions to facilitate communication, conduct transactions, and store data. Under these conditions, the risks of cyberattacks have also increased [2]. Furthermore, in today's digital age, the terms UX (user experience) and UI (user interface) frequently appear. UI refers to visual aspects, while UX focuses on functionality and user interaction [3]. Both concepts play crucial roles in the development of digital products. Successful digital products require both UX and UI design. UI design emphasizes the visual and interactive elements, making the product visually appealing and engaging. On the other hand, UX design ensures that the product is functional, meets user needs, and provides a seamless experience. Essentially, the UI designer is responsible for creating beautiful and functional interfaces. In the development of web applications, the integration of cybersecurity, UX/UI and accessibility in the best way possible is essential to assure all the users that they can access sensitive information on the Internet in a comfortable and confident way.

1.2. Proposed Solution and Contribution of the Paper

This paper addresses the critical gap between accessibility and cybersecurity in web application development by proposing a novel architectural model called the **Holistic Security and Accessibility Layered Architecture (HSALA)**. HSALA consists of five key interconnected layers: the **Data Layer, Persistence Layer, Application Layer, Content Presentation Layer and Accessibility Enhancement Layer**. Each functional layer is paired with a corresponding security layer, ensuring that security considerations are integrated at every level of the architecture. This holistic approach embodies the integration of accessibility and cybersecurity throughout the development process.

The core contribution of this paper lies in the conceptual and theoretical advancement provided by the Holistic Security and Accessibility Layered Architecture (HSALA). Unlike existing approaches that treat security and accessibility as supplemental features, HSALA presents a structured, systematic methodology that seamlessly integrates both elements from the very beginning of web application development. The following key contributions underscore the architecture's novelty, address pressing contemporary challenges, and pave the way for future empirical validation:

1. **Introduction of the HSALA Model:** We propose a comprehensive architectural framework—HSALA—that merges accessibility and cybersecurity into web application development, directly addressing the gaps between existing accessibility-oriented and security-oriented methodologies. Each functional layer (Data, Persistence, Application, Content Presentation, and Accessibility Enhancement) is paired with a dedicated security layer, ensuring a defense-in-depth approach that explicitly tackles accessibility vulnerabilities alongside typical security concerns.
2. **Comparative Validation:** Through a theoretical and analytical case study, we demonstrate how HSALA outperforms traditional three-layered architectures in tackling modern development concerns, such as scalability, user experience, inclusive design, and regulatory compliance. This comparative assessment highlights the additional value of embedding security and accessibility throughout the development lifecycle.
3. **Implementation Guidance:** Beyond conceptualization, the paper details specific technologies, security safeguards, accessibility features, secure coding practices, and dynamic scaling that practitioners can adopt without sacrificing performance or accessibility. This guidance underscores the practical applicability of the model's design philosophy.

4. **Performance Insights:** We investigate the potential performance ramifications of adding extra layers and propose corresponding mitigation strategies—such as resource allocation, caching, and asynchronous operations—to preserve both efficiency and scalability. This methodological guidance empowers developers to balance robust security, thorough accessibility, and responsive performance.
5. **Foundations for Future Work:** While the paper outlines a roadmap for practical implementation and empirical evaluation, this forward-looking component does not diminish the significance of the current contribution. Many influential architectures start as rigorously developed theoretical models before undergoing validation in multiple real-world settings. Consequently, we advocate future research in user-centered design—particularly for users with disabilities—as well as AI-driven personalization and semantic web approaches to continually refine and enhance HSALA.

1.3. Paper Organization

The following sections of the paper are organized as follows: Section 2 presents the related work and research gap. Section 3 provides an overview of the importance of accessibility, cybersecurity, and UX/UI in web applications, emphasizing the need for integrating these aspects to support vulnerable groups. Section 4 provides a comprehensive introduction to HSALA, the proposed development model, explaining each layer and the corresponding security measures, and how they interact to create a secure and accessible architecture. A comparative analysis between HSALA and the traditional three-layered architecture, highlighting the advantages of HSALA in terms of accessibility, cybersecurity, scalability, user experience, and other critical parameters, is described in Section 5. Finally, Section 6 summarizes the findings, acknowledges the limitations, and outlines future work needed to empirically validate HSALA and enhance its practical applicability.

2. Related Work and Research Gap

Combining accessibility and cybersecurity in web application development is an emerging area of research, reflecting the growing need for inclusive and secure digital platforms. The existing literature has primarily addressed accessibility and cybersecurity as separate concerns, with limited studies exploring their intersection.

Extensive research has been conducted on accessibility in web applications, with a focus on creating guidelines and standards to ensure that digital content is accessible to all users, including those with disabilities.

The Web Content Accessibility Guidelines (WCAG) [4] provided by the W3C's [5] Web Accessibility Initiative (WAI) [6] are widely recognized standards guiding developers in creating accessible web content. These guidelines emphasize principles such as perceivability, operability, understandability, and robustness to improve the user experience for individuals with disabilities.

Rodrigues and Fortes [7] created a checklist to evaluate web accessibility and usability for older adults, emphasizing the importance of inclusive design. Similarly, Kurt S. [8] highlighted the necessity of web accessibility in education to promote universal access to information. The existing research has also explored tools and techniques for assessing and improving web accessibility, such as the use of semantic HTML, WAI-ARIA (Web Accessibility Initiative—Accessible Rich Internet Applications) roles, and assistive technologies [9,10].

Cybersecurity research has focused on protecting web applications from threats like phishing, malware, and data breaches, Ahsan et al. [11] reviewed cybersecurity threats and mitigation strategies using machine learning, highlighting the dynamic nature of cyberattacks and the necessity for robust security measures. Leszczyna [12] conducted a re-

view of cybersecurity assessment methods, emphasizing the importance of comprehensive strategies in safeguarding digital assets.

Preventive cybersecurity strategies aim to anticipate and mitigate potential threats before they materialize [13]. Grigaliūnas et al. [14] proposed a framework which integrates security considerations throughout the development process, promoting proactive defense mechanisms. The increasing complexity of cyberattacks necessitates the continuous improvement and integration of security practices in web application development.

User experience (UX) and user interface (UI) design have crucial roles in developing digital products that are both functional and appealing [15]. The existing research has emphasized the need for cohesive design systems that enhance usability and cybersecurity [16]. Sanchez-Gordon et al. [17] discussed profiling users with disabilities on e-learning platforms, underscoring the importance of considering diverse user needs in design practices.

Design principles, such as alignment, proximity, repetition, and contrast, are essential for creating user-friendly interfaces [18]. The choice of colors and visual elements can significantly impact both accessibility and user engagement. Integrating these principles contributes to improved user experiences and can enhance cybersecurity by guiding users toward safe behaviors [19].

The synergy between cyber-secure and accessible interfaces is essential for enhancing each other's strengths and improving digital interactions [20]. However, in current development models there is a gap between security-oriented and accessibility-oriented approaches, with few solutions proposed to solve this complex challenge [21].

Despite the critical importance of both accessibility and cybersecurity, there is limited research on models that integrate these two aspects holistically. Renaud and Coles-Kemp [21] discussed the complexity of creating accessible and inclusive cybersecurity solutions, emphasizing that security measures often inadvertently exclude users with disabilities.

Kävrestad et al. [22] explored design principles for cognitively accessible cybersecurity training, indicating the need for security practices that are understandable and usable by all user groups. However, a gap still exists in architectural models that effectively integrate accessibility and cybersecurity from the outset.

Traditional web application architectures, such as the three-layered model comprising the Data Access Layer, Application Logic Layer, and Presentation Layer [23], have been widely adopted due to their simplicity and clear separation of concerns. These models focus on enhancing modularity and maintainability; however, they often treat security and accessibility as add-ons rather than integral components of the architecture.

Marinho et al. [24] examined an architecture-based reliability analysis of multi-layered web applications but did not focus on integrating accessibility or security comprehensively. Villamizar et al. [25] compared monolithic and microservice architectures, discussing scalability and performance but not addressing the integration of accessibility and cybersecurity.

The lack of integration between accessibility and cybersecurity in existing models highlights the need for a holistic architectural approach. Fanea-Ivanovici and Pană [26] emphasized the importance of comprehensive strategies in digital transformation initiatives. By embedding accessibility and security considerations into each layer of the architecture, developers can create applications that are both inclusive and secure.

In summary, while significant progress has been made in advancing accessibility and cybersecurity individually, there is a paucity of research on their integration within web application architectures. By reviewing the existing literature, we establish the necessity and relevance of a holistic approach, setting the foundation for our proposed solution.

3. Background

Currently, cybersecurity and accessibility have become key elements of a successful approach to digital transformation. To reach a wider audience for digital content, it is important that the platforms are accessible and embrace inclusivity. Cybersecurity has emerged as a critical concern in every industry due to the imperative of safeguarding sensitive information and preserving business continuity. In this context, both accessibility and cybersecurity form indispensable components of an effective digital strategy. They are fundamental necessities in the era of digital transformation to promote equality, trustworthiness, and confidentiality [27].

By digital accessibility, we mean the ability of IT systems to provide services to those suffering from disabilities, whether temporary or permanent, using assistive technologies. This concept refers to websites, applications, and all user interfaces of the digital world. Web accessibility is essential not only to meet legal obligations [28], but also, especially, to promote an ethical and inclusive approach.

Furthermore, websites and web applications that offer accessible interfaces have a better organization of structural design and source code. These elements also enhance search engine indexing and improve ranking [29].

Regulatory policies have been implemented to ensure that digital products are inclusive and accessible to all users. Some examples of these legal frameworks are the European Accessibility Act (EAA) [30], the Disabilities Act (ADA) in the United States [31] or the Web Content Accessibility Guidelines (WCAG) [4].

The Web Content Accessibility Guidelines (WCAG) is a subset of accessibility recommendations for various types of websites and is published by the Web Accessibility Initiative (WAI), which is part of the World Wide Web Consortium (W3C).

The WCAG are based on four fundamental principles: perceptible, operable, understandable and robust. The four principles are explained below:

- **Perceptible:** The content must be recognizable to users. A person with deafness or hearing loss cannot hear audio clips and a person with blindness or vision impairment cannot see images. If these elements are present on a web application or website, they may not be fully detectable by all users, thereby failing to meet the first principle. Alternatives must therefore be offered for such elements. This can be, for example, a textual description of an image or caption and subtitles for audio video components.
- **Operable:** Web pages must be operable and allow users to perform actions. To fulfill this principle, for example, a website must therefore be completely operable using the keyboard, allowing navigation and interaction for the many Internet users who cannot use a mouse.
- **Understandable:** Users should not have any problems with regard to obtaining the information on the website and how to interact with that information. To comply with this principle, the arrangement of the page has to be as simple as possible, and the language used in the text should be as clear and simple as possible.
- **Robust:** The website can be accessed by users using different browsers and utilities. Using clean, standardized code helps to maintain accessibility over time, even as technologies and devices evolve.

Integrating these design principles not only benefits users with disabilities, but also enhances the overall user experience, promoting inclusion.

Cybersecurity is essential to digital transformation, playing a very important role in protecting sensitive data and ensuring trust in modern digital systems. Information security is the practice of protecting data, networks, systems, and programs against digital attacks, unauthorized access, destruction, or alteration. The practical implications of cybersecurity

are found in almost every digital sector, encompassing an array of methods and techniques designed to secure the digital environment.

One of the essential components in cybersecurity is the protection of user information from unauthorized access and misuse. Preventive cybersecurity is a strategy that is responsible for avoiding cyberattacks [11] before they happen, involving a combination of reactive and proactive measures.

The consequences of an attack following a cyber vulnerability can be serious, causing serious economic and trust damage. The complexity of cyberattacks is continuously developing, making them harder to detect, which can increase the potential damage. By detecting and blocking attack attempts before they reach corporate systems, a company eliminates the potential costs and threats they pose to the business continuity. In addition, regulatory compliance is an important factor in cyberattack prevention efforts. Complying with regulations, such as the General Data Protection Regulation (GDPR) [32] in Europe and the California Consumer Privacy Act (CCPA) [33] in the United States, helps organizations avoid fines and legal issues while aligning with best practices and industry standards.

On 12 February 2024, Romania experienced a significant ransomware cyberattack targeting the production servers that support its Health Information System (HIS). As a result of the attack, the system was down, and files and databases were encrypted. The sources said that 21 medical units from Bucharest, Azuga, Constanta, Iasi, Pitesti, Slobozia, Hunedoara, Calarasi, Rosiorii de Vede, Targoviste, Medgidia, and Fundeni were affected. In all hospitals utilizing this type of computer system, safety measures have been implemented, including internet restrictions and data backups on external media [34].

UX design (user experience design) is both analytical and functional, focusing on the quality of a user's interaction with a service or product. The main aspects analyzed focus on the real problems identified by users. Therefore, it aims to solve these issues by developing or implementing the following:

- functional solutions;
- personalized experiences; and
- wireframes, prototypes, and usability tests to ensure smooth and efficient user flows.

UI design (user interface design) is artistic and focuses on the visual experience and interactivity of the user. It refers to the visual and interactive elements of a product's interface. The main aspects analyzed address real problems related to aesthetics (such as color schemes and visual elements), interactivity (such as how each component functions on the interface), and adaptive design (ensuring the interface is suitable for various devices and platforms).

Creating a cohesive design involves using a common set of design elements such as colors, visual styles, structures, and typography. To create a cohesive, scalable, and efficient design, a comprehensive design system is essential. This system must include a repository of design and development standards, practices, and components that guide a consistent user experience (UX) across platforms and products. These elements encompass visual guidelines, reusable components, interaction patterns, accessibility guidelines, security considerations, and documentation and governance.

Although the idea of a design system is not new, our approach highlights the crucial need to integrate cybersecurity and accessibility standards within it. While most design systems already include libraries of reusable components and a shared language, many organizations neglect to incorporate explicit security and accessibility guidelines at the component level (e.g., standardized secure form patterns, default ARIA attributes, and color contrast checks).

An extended design system combines principles, components, and templates to enhance the design process and ensure consistency across digital products. By consolidating

these resources into a single, governed system, developers and designers can rapidly prototype, maintain consistency in design and coding practices, and ensure that accessibility and security are not overlooked.

The goal of inclusive cybersecurity is to ensure that everyone can safely participate in the digital world. This includes, but is not limited to, children, students, the elderly, and individuals with disabilities, as each have unique vulnerabilities and needs.

For children, accessible web interfaces can support early learning and development by providing age-appropriate interfaces with visual cues and minimal text adapted to their level. Cyber safety for children involves protecting them from online predators, cyberbullying, and exposure to inappropriate content.

Establishing robust, universal child-safety measures on the open web is exceptionally difficult. The key challenges include the following:

- Content filtering limitations: The automatic detection of harmful or inappropriate content is prone to false positives and false negatives. Advanced AI-driven filters can help but are not faultless.
- Parental/guardian involvement: Realistically, adult supervision remains crucial. Technology alone cannot entirely replace oversight and education regarding safe online practices.
- Regulatory differences across jurisdictions: legislation, such as the [35] Australian eSafety framework, or the [36] Children's Online Privacy Protection Act (COPPA) in the U.S., sets stringent compliance rules; however, consistent enforcement and global adoption remain complex.
- Balancing privacy and protection: systems designed to protect children must also respect privacy rights and avoid overreaching surveillance.

Australia has implemented notable measures through agencies like the eSafety Commissioner [37], which provides resources and guidelines for parents, schools, and online service providers to address cyberbullying, online grooming, and the exposure to harmful content. These initiatives highlight the following:

- Multi-stakeholder collaboration: government bodies, educational institutions, social media platforms, and parents share responsibility.
- Ongoing education and awareness: training, digital literacy programs, and safer online curricula for minors.
- Reporting mechanisms: established channels to report illegal or harmful content and rapid response to victim support.

Despite these efforts, no system can guarantee 100% safety. Hence, solutions for "cybersecurity for children" should be seen as mitigations rather than an absolute remedy. A combination of accessible UI design (e.g., simplified parental controls and clear language usage) and robust security measures (e.g., content filtering and safe-browsing APIs) can significantly reduce risks, but ongoing oversight is still necessary.

Another vulnerable group includes students. The educational platforms they access must be accessible so that all of them, including those with disabilities, can fully participate in learning activities. Educational content is made more accessible with features like text-to-speech capabilities, screen reader compatibility, and changeable text sizes. Additionally, to maintain the integrity of students' academic records and protect their privacy, educational platforms need to be secure and resilient against cyberattacks.

Accessible online interfaces are crucial for senior citizens to preserve their independence and social connections. Age-related impairments, like low eyesight, hearing loss, or difficulties with motor skills, are common for individuals in this group [38], which can make using typical web interfaces challenging. Accessible design features, such as larger text,

voice commands, and simplified navigation, can greatly improve their ability to use digital services. In terms of cybersecurity, seniors are often targeted by cybercriminals due to their limited technical knowledge. Phishing scams, identity theft, and fraud are significant risks. Therefore, implementing strong cybersecurity measures, including robust authentication processes and educating users on threat recognition, are necessary to effectively protect them.

UX/UI design is vital for enabling marginalized groups to interact with digital systems in a straightforward, effective, and frustration-free manner. Well-developed interfaces foster self-sufficiency, help reduce barriers, and facilitate improved digital interactions that enhance protection and accessibility for all users. In the context of digital bureaucracy, particularly for elderly or persons with disabilities, optimized interfaces are crucial to make sure that the access to information is offered to everyone.

Cybersecurity is a critical concern for administrative institutions due to their responsibility for managing sensitive data [39]. Among the data contained in bureaucratic and government databases, the most sought-after ones by cyber attackers are the ones containing the personal, economic, and medical information of the citizens. The failure to satisfy protection requirements can lead to identity theft, economic loss, and the erosion of confidence and trust in public institutions.

Within the proposed solution, both cybersecurity and accessibility are integral from the earliest stages of design and development. Rather than a strictly sequential priority (“accessibility first, then security” or vice versa), HSALA advocates a parallel, integrated approach in which the following is fulfilled:

- Security is treated as a baseline concern (e.g., ensuring user data are protected and adopting secure coding practices from the initial architecture planning).
- Accessibility is also treated as a foundational requirement, ensuring that UI components and workflows are inclusive from inception.

This “security by design” and “accessibility by design” philosophy means that each functional layer includes security provisions and accessibility from the start. There is no absolute hierarchical order—the two coexist simultaneously, as neglecting either would result in an application that is either insecure or exclusive. This implies a deep understanding of the needs of users with disabilities and the integration of such needs into each design decision to ensure that accessibility is treated not as an afterthought, but as a core element. In the process, security should also form part of the design concept and process, as well as threat modeling to detect possible security risks early in design. Thus, a two-way priority, both on accessibility and the maintenance of element security integrity, makes sure that accessibility features are robust without causing vulnerabilities. Accessible coding practices encompass the use of semantic HTML to organize content effectively, thereby facilitating accurate interpretation by assistive technologies. Where dynamic content is required, WAI-ARIA properties and roles can be utilized to enhance content accessibility and optimize the experience for users who rely on assistive technologies.

There are four principles that ensure a web page is clear, organized, and professional: alignment, proximity, repetition, and contrast [40]. Additionally, the choice of colors in UX and UI concepts plays a crucial role:

1. Alignment:

- UX/UI: ensures that all elements are arranged coherently, creating a clean and professional layout.
- Cybersecurity: the proper alignment of security elements, such as login buttons and notifications, helps users understand security measures more easily [41].

2. Proximity:
 - UX/UI: elements that are similar or related to each other should be grouped together to improve visual organization.
 - Cybersecurity: grouping security options (e.g., privacy and authentication) in the same section makes it easier for users to manage security settings [42].
3. Repeating:
 - UX/UI: consistent use of styles and visuals to create unity and coherence.
 - Cybersecurity: repeating security messages and notifications helps reinforce safe behaviors and educate users on good security practices [41].
4. Contrast:
 - UX/UI: using contrast to highlight important elements and guide the user's attention.
 - Cybersecurity: using contrast to highlight security warnings and critical messages helps to draw users' attention to potential risks and necessary actions [42].
5. Choice of colors:
 - UX/UI: Colors are essential in design, creating a visually appealing experience and guiding users through content. This concept is based on color psychology, which examines how different colors can evoke specific emotions and behaviors.
 - Cybersecurity: Colors can be utilized to indicate various security levels, such as using red for warnings and green for safe actions. This helps users quickly grasp the security status of an app or website.

Integrating these principles not only improves user experience but also contributes to creating a safer and more protected digital environment.

Moreover, ensuring that all interactive elements are keyboard navigable provides equal access for users operating via keyboards or alternative input devices. To address other common vulnerabilities like cross-site request forgery (CSRF) [43], cross-site scripting (XSS) and SQL injection, secure coding standards should be included in the development process, as well as input validation and sanitization to prevent and block attacks while keeping input mechanisms accessible.

When it comes to the implementation of digital technologies, authentication [44] and authorization are considered as the main mechanisms to guard against cyberattacks; therefore, multi-factor authentication (MFA) [45] must be set up in a way that is accessible to users with disabilities. Data encryption is essential for safeguarding safety and protecting personal information.

Leveraging open-source technologies in the design and development of accessible, cyber-secure web interfaces offers several key advantages. First, it promotes enhanced transparency and security: by making the source code publicly accessible, open-source communities can collectively review, audit, and improve the codebase. This process not only helps to uncover and patch vulnerabilities more rapidly but also encourages best coding practices from the outset. Furthermore, having a broad community contributing to and inspecting the code helps to reduce the likelihood of hidden backdoors or other security flaws.

Second, open-source solutions are typically designed with compliance and standards in mind, facilitating the adherence to established guidelines such as the Web Content Accessibility Guidelines (WCAG) and various cybersecurity protocols. Many open-source frameworks and libraries provide built-in components that already follow recognized accessibility and security standards (for instance, semantic HTML elements or pre-configured HTTPS modules), thus saving development time while ensuring consistent compliance.

Additionally, because these solutions are collectively maintained, updates related to new regulatory mandates often emerge faster than in closed-source systems.

Finally, open-source technologies bring cost-effectiveness and flexibility to web interface development. They are generally license-free, making them financially appealing to both small organizations and large enterprises. Due to the modular nature of open-source software, teams can adopt only the features that fit their needs while retaining the option to customize or extend functionalities as the application evolves. This adaptability not only helps to optimize resource usage and reduce costs but also fosters an environment of continuous innovation, as developers worldwide can contribute enhancements to the shared code. By combining the web design with cybersecurity tools, one can create websites that not only have a nice web design but also protect users and their data.

Moreover, design can significantly contribute to improving cybersecurity in the following ways:

- **Security-oriented design:** Using security principles early in the design phase of a website can help prevent many security problems. For example, ensuring that sensitive data are encrypted or that the site uses HTTPS can help protect user data.
- **Authentication and authorization:** An efficient design can facilitate the implementation of tools for authorization and authentication, such as two-step authentication. This type of authentication, known as two-factor authentication (2FA), uses two steps to verify the identity of a person when accessing a service like social network, email, and bank account, etc.
- **User education:** an intuitive and easy-to-understand design can help users better understand security settings and use them appropriately.
- **Attack prevention:** by implementing design practices such as limiting the number of authentication attempts or displaying generic error messages, knowledge-of-error attacks can be prevented.
- **Updates and maintenance:** a well-thought-out design facilitates regular site updates and maintenance, ensuring ongoing protection against the latest threats [46].

In conclusion, an effective and security-oriented design can play a crucial role in protecting user data and preventing cyberattacks.

4. The Proposed Development Model: Holistic Security and Accessibility Layered Architecture (HSALA)

In a typical web application, the architecture is divided into multiple layers to enhance modularity, maintainability, and scalability [24]. Examples of layered web applications are Data Access Layer—Application Logic Layer—Presentation Layer [23] or Resource Layer—Service Layer—Presentation Layer [47].

We propose a novel architectural model called the **Holistic Security and Accessibility Layered Architecture (HSALA)**, which consists of five key interconnected layers: the **Data Layer**, **Persistence Layer**, **Application Layer**, **Content Presentation Layer**, and **Accessibility Enhancement Layer**.

Each of the abovementioned layers also has an adjacent layer of security: the **Data Security Layer**, **Persistence Security Layer**, **Application Security Layer**, **Content Presentation Security Layer** and **Accessibility Security Layer**. This combination ensures that security considerations are integrated at every level of the architecture, embodying the holistic approach of HSALA. Figure 1, presented below, illustrates the HSALA model and how the layers interact with each other:

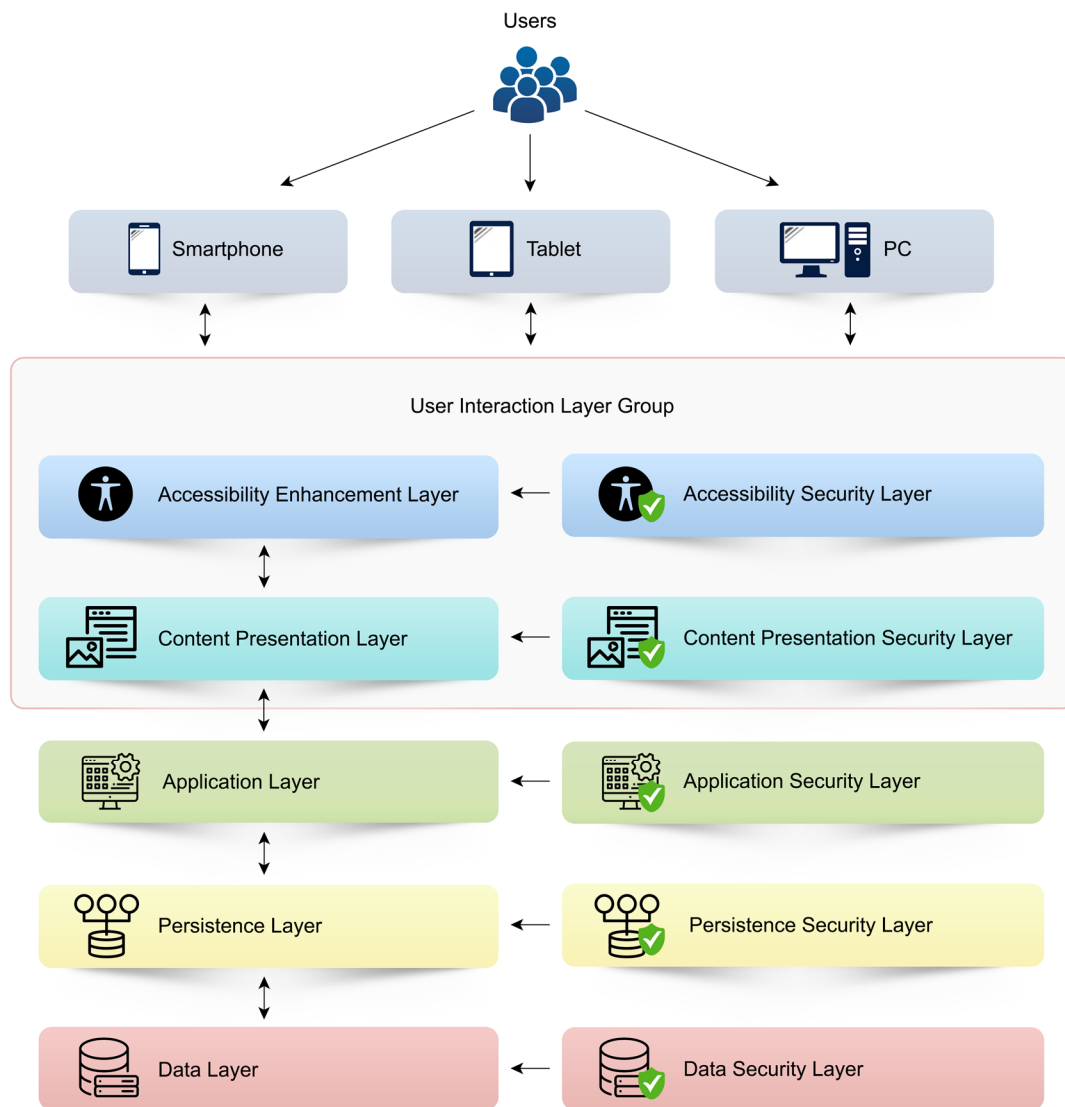


Figure 1. The HSALA model, consisting of five interconnected layers, each paired with an adjacent security layer.

4.1. The Data Layer and Data Security Layer

The application architecture's foundation is referred to as the Data Layer and it deals with the storage and management of data. This layer makes use of databases such as MySQL or PostgreSQL, and NoSQL [48] databases like MongoDB. The Data Layer has the objective of storing and retrieving the information in the safest and fastest way possible.

The layer adjacent to the Data Layer is the Data Security Layer that prevents the unauthorized access of the database by encrypting the data. This layer implements techniques like data masking [49] to protect sensitive information when accessed by non-privileged users by substituting the original data with anonymized values. It also utilizes database encryption and database activity monitoring tools to reinforce data security.

The implementation of the Data Security Layer ensures that confidential and sensitive information is guarded from unauthorized access and breaches. It provides compliance with regulations like GDPR (General Data Protection Regulation) and HIPAA (Health Insurance Portability and Accountability Act) [50], safeguarding data integrity and confidentiality.

Figure 2 presents an overview of the techniques, mechanisms and tools of the above-mentioned layers, and illustrates their interconnections:

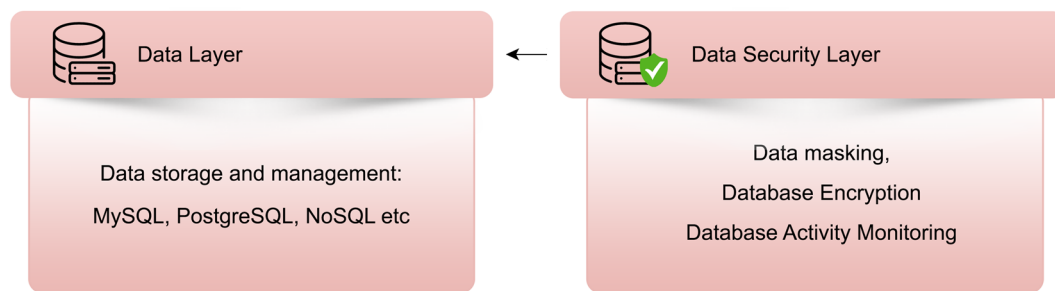


Figure 2. Data layer and data security layer.

4.2. The Persistence Layer and Persistence Security Layer

On top of the Data Layer, we can find the Persistence Layer. This layer handles the logic needed to access and manage the data, making the operations in the database abstract. The Persistence Layer outlines the capabilities and functions made available to the user in order to perform CRUD (Create, Read, Update, Delete) operations [51] on the database assisted by Object-Relational Mapping (ORM).

The Persistence Layer is enhanced with security measures by the Persistence Security Layer, which ensures that the data are accessed securely and only by authorized entities. It prevents common attacks and enforces strict access controls, maintaining data integrity and security.

The Persistence Security Layer is responsible for strong authentication and authorization mechanisms; it defines roles and permissions to limit access according to user roles. This ensures that data access is restricted to authorized users and applications. Methods like SQL Injection Prevention, Secure Access Controls, and Role-Based Access Control [52] are used to achieve this.

Figure 3 illustrates the relationships, technologies, and roles associated with both the Persistence Layer and the Persistence Security Layer:

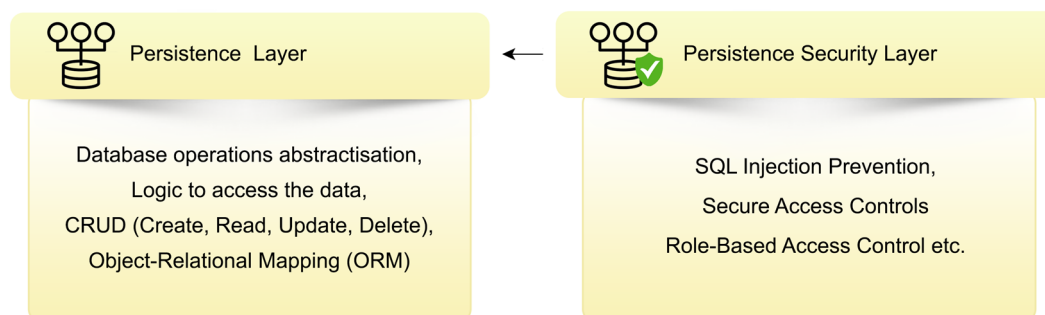


Figure 3. Persistence layer and persistence security layer.

4.3. The Application Layer and Application Security Layer

The Application Layer is situated above the Persistence Layer, which encompasses the basic business logic of the application. It comprises aspects like processing user inputs, enforcing business logic, and maintaining the integrity of the data.

The Application Layer implements the data flow from the user interface to the database and vice versa. Within this layer, frameworks such as Django (for Python) [53], Laravel (for PHP) [54], Spring (for Java) [55], and ASP.NET (for C#) [56] are commonly and widely used.

The Application Security Layer is in charge of protecting the application from outside threats and also maintaining a secure codebase. It implements secure coding practices, dynamic application security testing, and web application firewalls [57].

Integrating security at the Application Layer prevents the exploitation of vulnerabilities and ensures that the application is not compromised easily through any kind of attack; this improves the overall security posture of the application.

Figure 4 presents the Application Layer and the Application Security Layer.

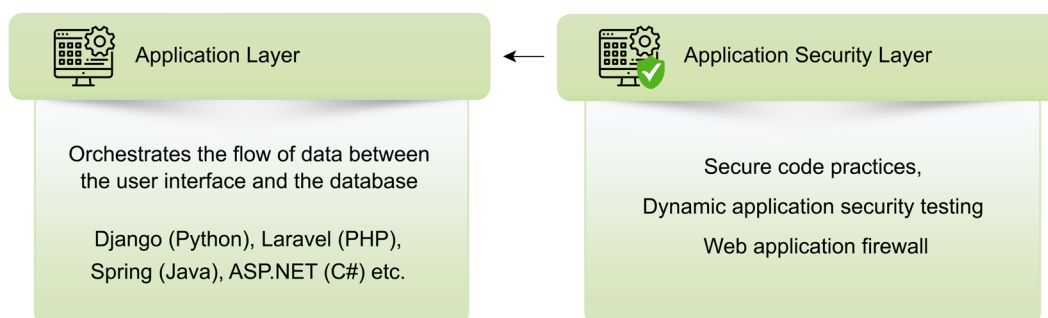


Figure 4. Application layer and application security layer.

4.4. Content Presentation Layer and Content Presentation Security Layer

The Content Presentation Layer is above the Application Layer and manages the web application's interface using web standards: HTML for page structuring, CSS for page styling, and JavaScript to add and control interactive elements [58]. In this layer, UX/UI plays a defining role. It ensures that the content and graphic elements of the application are not only functional but also visually intuitive, encouraging users to interact easily with the web application.

The Content Presentation Security Layer aims, primarily, at protecting the Content Presentation Layer to make it safe for users from client-side attacks and protect the data and its integrity from unauthorized alteration.

The Content Presentation Security Layer enhances the security of all the elements of the user interface by implementing techniques like Content Security Policy [59] or Cross-Site Scripting Prevention [60].

These measures minimize the risk of content being loaded from undesired locations, and include input validation, sanitization, and output encoding. This layer ensures secure communication with the application using the HTTPS protocol, which improves user confidence by making the browsing experience safe and shielding private information from external threats, as illustrated in Figure 5:

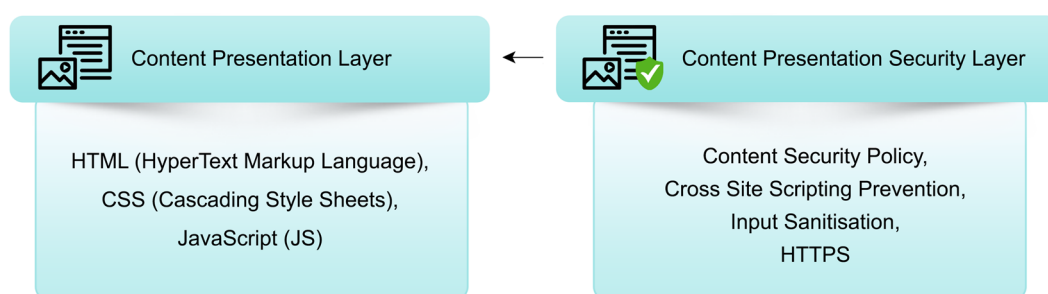


Figure 5. Content presentation layer and content presentation security layer.

4.5. The Accessibility Enhancement Layer and Accessibility Security Layer

The last layer which is located above the whole stack is the Accessibility Enhancement Layer, which increases the web application's accessibility for all the users, especially for those with disabilities.

The Accessibility Enhancement Layer implements different technologies, such as the following:

WAI-ARIA is a specification published by the World Wide Web Consortium (W3C) which aims to improve the accessibility of web applications. It provides supplementary HTML attributes designed to improve the accessibility of web content and applications for individuals with disabilities. It defines metadata like roles, properties, and states that can describe the behavior of interactive and complex elements, making them perceivable and operable by assistive technologies like screen readers [61].

- Using Semantic HTML elements that describe their purpose improves accessibility. Semantic elements such as headings, navigations, and footers are specifically defined, facilitating the usage of assistance technologies like screen readers, which enhances the navigation flow for the users.
- Keyboard Accessibility: It is important to provide keyboard access to all interactive elements (such as links, buttons, and form fields) for those users who cannot use a mouse. This can involve the management of focus order or management of custom controls to make sure that the interactive elements are keyboard navigable.
- Alt Text for Images: providing descriptive alternative text (alt attribute) for images, allowing users who cannot see the images to comprehend their content through assistive technology.
- Color Contrast and Text Size [62]: Ensuring that there is sufficient color contrast between the text and its background facilitates the reading of the content for people with visual impairments. For instance, in the case of text size relative units (such as *em* or *rem*), it allows users to increase or decrease the overall font size according to their preferences.
- Captions and Transcripts for Multimedia Content: captions in videos and transcripts in audio content eliminate the barriers to access such information for individuals with deafness and hearing loss [63].

The model emphasizes a user-centered design and accessibility testing approach, involving users with disabilities throughout the development process to create truly inclusive applications, such as the following:

- User personas and scenarios: Developers create detailed personas reflecting users with diverse disabilities—including visual impairments, hearing loss, motor difficulties, and cognitive challenges—and develop scenarios to examine how these individuals engage with the application. These insights guide design decisions that address their specific requirements.
- Inclusive design principles:
 - Perceptibility: information is presented in multiple formats (e.g., text alternatives for images).
 - Operability: all functionalities are accessible via keyboard and assistive technologies.
 - Understandability: consistent navigation and clear instructions help users comprehend and interact with the application.
 - Robustness: the application is compatible with a wide range of devices and assistive technologies.
- Accessibility testing with real users: Testing involves participants with disabilities using assistive technologies like screen readers, magnifiers, and alternative input devices. The feedback from these sessions informs iterative improvements.
- Continuous feedback loop: A continuous feedback mechanism ensures that user input directly influences design refinements. This iterative process enhances usability and ensures that the application remains accessible as new features are added.

The Accessibility Enhancement Layer is secured by the Accessibility Security Layer, which ensures that accessibility features do not introduce security vulnerabilities and do not compromise the application's security.

The Accessibility Security Layer embeds techniques and practices addressing the accessibility challenges posed by various authentication methods [44], like implementing multi-factor authentication methods that are inclusive, and providing secure login options for all users, including individuals with disabilities. Presented below is a list of challenges and mitigation strategies for several authentication methods:

- Knowledge-Based Authentication (Passwords, PINs):
 - Challenges: Users with cognitive impairments or memory difficulties may struggle to remember complex passwords or PINs. The requirement for frequent password changes can exacerbate this issue.
 - Mitigation Strategies: implement user-friendly password policies, allow the use of passphrases, and provide options for password managers that are accessible.
- Possession-Based Authentication (Hardware Tokens, One-Time Passwords):
 - Challenges: individuals with motor impairments or limited dexterity may find it difficult to handle physical tokens or input codes from devices.
 - Mitigation Strategies: offer virtual tokens, mobile app-based authenticators with accessible interfaces, or alternative methods that do not require physical manipulation.
- Biometric Authentication (Fingerprint, Facial Recognition):
 - Challenges: Users with certain physical disabilities may be unable to provide biometric data. For example, amputees cannot use fingerprint scanners, and facial recognition may not work accurately for users with facial differences.
 - Mitigation Strategies: provide multiple biometric options, such as voice recognition or iris scanning, and ensure that systems are calibrated to meet the needs of a diverse user base.
- Challenge-Response Mechanisms (CAPTCHAs):
 - Challenges: traditional visual CAPTCHAs are inaccessible to users with visual impairments, while audio CAPTCHAs may not be suitable for those with hearing impairments or non-native language speakers.
 - Mitigation Strategies: implement accessible CAPTCHAs, such as text-based challenges, logic puzzles, or interactive tasks that are compatible with assistive technologies.

User choice and flexibility is recommended; users can select from multiple authentication options based on their abilities and preferences, ensuring that security does not become a barrier to access. The instructions for setting up and using multi-factor authentication are provided in accessible formats, including large text, audio descriptions, and simplified language. Additionally, support channels need to be equipped to assist users with disabilities.

By integrating these strategies, HSALA ensures that authentication mechanisms are inclusive and do not inadvertently exclude users with disabilities. This approach upholds the principles of both security and accessibility, reinforcing the model's commitment to creating a universally usable web application architecture.

The layer is also responsible for integrating accessibility scanning (e.g., WAVE), and ensures that ARIA *roles*, *properties* and *attributes* are implemented without introducing security risks, providing a secure and inclusive experience for all users, meeting both accessibility and security standards.

Figure 6 presents the relation, technologies, and roles of the Accessibility Enhancement Layer and Accessibility Security Layer.

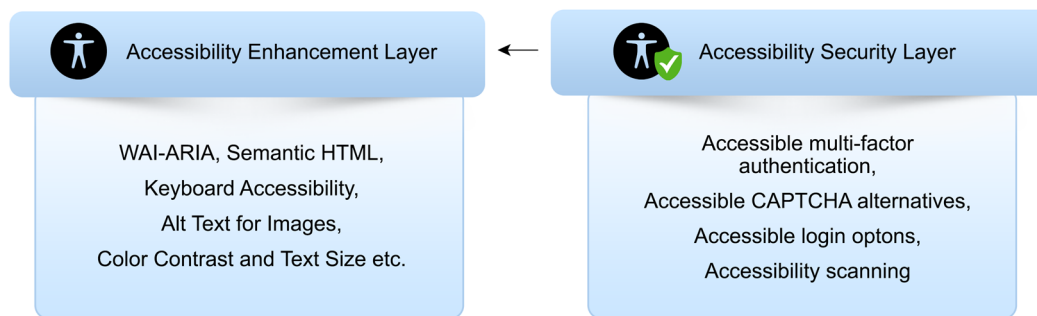


Figure 6. Accessibility enhancement layer and accessibility security layer.

These two interconnected layers, along with the Content Presentation Layer and Content Presentation Security Layer, form the User Interaction Layer Group (as shown in Figure 1). This group represents the interface between the web application and users through devices such as smartphones, tablets, or PCs.

4.6. Novelty and Scope of HSALA

HSALA restructures layered design by systematically weaving accessibility and security into every stage. This results in a noticeably more comprehensive development architecture—one that surpasses traditional models in several important aspects:

1. **Dedicated Security Layers Aligned with Functional Layers:** Instead of having an overarching or optional security module, HSALA integrates security checks, mechanisms, and best practices at each layer (Data, Persistence, Application, Content Presentation and Accessibility Enhancement). This modular approach ensures that no layer is overlooked—a critical departure from traditional architectures.
2. **Explicit Accessibility Layer:** Beyond a simple “Presentation” tier, HSALA adds a dedicated Accessibility Enhancement Layer, which is not typically found in three-tier models. This layer ensures that advanced accessibility measures (e.g., ARIA attributes and multi-factor authentication adaptations for users with disabilities) are consistently designed, tested, and maintained as a core requirement.
3. **Concurrent Accessibility–Security Integration:** A hallmark of HSALA is the recognition that certain accessibility features (e.g., ARIA roles and accessible authentication flows) can introduce new security challenges, and vice versa. By pairing each functional layer with a corresponding security layer that also accounts for accessibility changes, HSALA moves beyond a mere “granular breakdown” and presents a new dimension: a “dual-track” model in which accessibility and security co-evolve.
4. **Holistic Lifecycle Emphasis:** Traditional three-layer models typically address separation of concerns between data, logic, and presentation. HSALA, in contrast, addresses the entire lifecycle—from data storage and secure access to user interface and post-deployment analytics—unifying accessibility and security concerns at each phase. This is not merely a more granular breakdown, but an entirely different vantage point that requires changes in how teams plan, design, and implement each layer.

Consequently, the proposed theoretical architecture is not just an incremental improvement but a significant realignment of layered web application design, ensuring that accessibility and security are prioritized at every stage.

4.7. Interaction of Security Mechanisms Across Layers

In HSALA, security mechanisms are designed to interact seamlessly across the entire architecture, ensuring cohesive security without compromising accessibility. This section details how security measures operate within and across the layers.

Data encryption and access control:

At the Data Layer, sensitive information is encrypted using robust algorithms like AES-256. The Data Security Layer ensures that encryption keys are securely managed, and that data remain encrypted as they move through the Persistence Layer and Application Layer. Access control policies, such as Role-Based Access Control (RBAC), are consistently enforced across layers to ensure that only authorized users and components can access specific data.

Authentication and authorization flow:

Authentication credentials are securely managed within the Application Security Layer during user authentication. Upon successful authentication, authorization tokens or session identifiers are generated and passed through the Persistence Security Layer to the Data Security Layer as needed. These tokens are validated at each layer to maintain the session integrity and prevent unauthorized access.

Consistent input validation:

Input validation is performed at multiple layers to safeguard against malicious inputs:

- Content Presentation Security Layer: implements client-side validation using JavaScript to provide immediate feedback to users.
- Application Security Layer: enforces server-side validation to ensure data integrity and prevent attacks like SQL injection and cross-site scripting (XSS).
- Persistence Security Layer: applies additional validation when interacting with the database to prevent unauthorized data manipulation.

This multi-layered validation ensures comprehensive protection even if one layer's defenses are bypassed.

Error handling and logging:

Secure error handling practices are employed across layers:

- Content Presentation Security Layer: displays user-friendly error messages without revealing technical details.
- Application Security Layer: logs detailed error information securely for developers while ensuring that sensitive data are not exposed.
- Data Security Layer: monitors and logs database activities for auditing and anomaly detection.

Cross-layer communication enables coordinated responses to security incidents, enhancing overall resilience.

Accessibility and security balance:

Accessibility features are integrated with security mechanisms to ensure inclusivity without compromising protection:

- Accessibility Enhancement Layer: implements features like ARIA attributes and keyboard navigation.
- Application Security Layer: ensures that these features do not introduce vulnerabilities, such as ensuring that ARIA roles do not expose sensitive information.

The collaboration between these layers maintains a balance where security measures are accessible and accessibility features are secure.

By integrating the aforementioned layers, the proposed model enables the web application to serve a broader audience and supports inclusive use by all users. Moreover, these protective measures preserve the security of each functional layer. This multi-layered

model not only complies with laws and regulations, increasing the legal protection of the web application, but also enhances overall security with a wide range of functions, making it robust and user-friendly without being intrusive.

4.8. Performance Considerations

While implementing the additional layers and features to enhance accessibility and cybersecurity significantly improve inclusivity and resilience, they may also impact performance due to the increased complexity. The addition of an extra layer could potentially lead to increased processing time and resource consumption. This may involve executing security protocols or accessibility features such as multimedia captioning.

Network latency could also be affected by advanced security mechanisms, like encrypted data transmission and multi-factor authentication, which might slightly increase response times.

To address potential performance issues, the following mitigation strategies could be employed:

- **Efficient code practices:** Writing optimized, well-structured, and reusable code ensures that execution times are minimized. This includes reducing redundancy and applying best practices for secure and accessible coding.
- **Resource allocation:** Leveraging cloud computing services enables dynamic resource scaling based on demand. Tools like AWS Auto Scaling or Microsoft Azure Autoscale can allocate additional resources during peak usage.
- **Monitoring and optimization:** Continuous performance monitoring with tools like New Relic, Grafana, or Prometheus helps identify bottlenecks. Regular profiling and load testing allow developers to fine-tune the architecture.
- **Content Delivery Networks (CDNs):** employing CDNs reduces latency by caching static content closer to users and distributing the server load.

Furthermore, maintaining a seamless user experience for all users, including those with disabilities, is essential. Performance considerations specific to this group include the following:

- **Responsive design:** Accessibility features should not hinder the responsiveness of the application. The efficient use of technologies like WAI-ARIA and semantic HTML ensures fast interaction.
- **Optimized assistive technology support:** testing the application with screen readers, voice input systems, and alternative navigation tools ensures that these technologies integrate smoothly without introducing delays.
- **Lightweight accessibility features:** implementing lightweight alternatives where possible (e.g., static ARIA attributes instead of dynamic updates) minimizes resource use without compromising functionality.
- **Inclusive performance testing:** performance testing with users relying on assistive technologies helps to ensure that their experience remains efficient and frustration-free.

The additional layers in the HSALA model inevitably introduce complexity, but with thoughtful design and optimization strategies, their impact on performance can be minimized. Efficient coding, dynamic resource scaling, and proactive monitoring ensure that response times and reliability remain high. The user experience, including for people with disabilities, remains fast and inclusive, achieving a balance between enhanced functionality and performance. The HSALA demonstrates that accessibility and cybersecurity can coexist with minimal trade-offs in efficiency.

5. Theoretical and Analytical Case Study: Comparing HSALA with the Traditional Three-Layered Architecture

When it comes to web application development, architectural models are key aspects in determining the efficacy, scalability, security, and accessibility of applications. This case study presents a theoretical and analytical comparison between the **Holistic Security and Accessibility Layered Architecture (HSALA)**—which emphasizes the integration of accessibility and cybersecurity throughout the development process—and a simpler, traditional three-layered architecture comprising the **Data Access Layer**, **Application Logic Layer** and **Presentation Layer**. The traditional model represents the conventional approach to web application development, where accessibility and security are often addressed separately from the core architecture.

The objective of this case study is to perform a theoretical validation of HSALA by comparing it with the classic model across critical parameters, such as accessibility features, cybersecurity measures, scalability potential, user experience, and other relevant factors. The analysis aims to demonstrate the superiority of HSALA in addressing contemporary challenges in web application development.

5.1. Description of the Architectural Models

5.1.1. Holistic Security and Accessibility Layered Architecture (HSALA)

HSALA introduces a multi-layered approach that integrates both functionality and security at each level. The model consists of five main functional layers (Data, Persistence, Application, Content Presentation, and Accessibility Enhancement), each paired with a corresponding security layer. Table 1 summarizes the core responsibilities of each layer and its associated security measures.

Table 1. Description of HSALA model.

Layer	Functionality	Security Layer
1. Data Layer	Responsible for data storage and management, utilizing databases like MySQL, PostgreSQL, or MongoDB.	Data Security Layer: Ensures data protection through encryption, data masking, and compliance with data protection regulations (e.g., GDPR).
2. Persistence Layer	Handles data access logic, providing an abstraction over the Data Layer using Object-Relational Mapping (ORM) tools.	Persistence Security Layer: Implements secure data access practices, including authentication, authorization, and protection against SQL injection.
3. Application Layer	Contains the core business logic, processing user inputs, and enforcing application workflows.	Application Security Layer: Secures the application logic through secure coding practices, input validation, and application-level firewalls.
4. Content Presentation Layer	Manages the user interface using web standards (HTML, CSS, JavaScript) and focuses on UX/UI design.	Content Presentation Security Layer: Protects the front-end from client-side attacks, implementing measures like Content Security Policy (CSP) and HTTPS enforcement.
5. Accessibility Enhancement Layer	Enhances the application’s accessibility by implementing standards like WAI-ARIA, ensuring compliance with WCAG.	Accessibility Security Layer: Ensures that accessibility features do not introduce vulnerabilities and that security measures are accessible to all users.

The layered structure of HSALA highlights the comprehensive integration of functionality and security at every level. This ensures that both operational effectiveness and protective measures are inherently built into the architecture. By pairing each functional layer with a corresponding security layer, HSALA provides a robust framework for developing secure and accessible web applications.

5.1.2. Three-Layered Architecture

A simpler, traditional architectural model commonly divides a web application into three primary layers. As shown in Table 2, these layers are the Data Access Layer, Application Logic Layer, and Presentation Layer. While effective in separating core concerns, this conventional model does not inherently integrate accessibility or security throughout its layers.

Table 2. Description of three-layered architectural model.

Layer	Functionality
1. Data Access Layer	Manages direct interactions with the database, performing CRUD operations.
2. Application Logic Layer	Contains the core functionality and application logic, processing data between the Data Access Layer and the Presentation Layer.
3. Presentation Layer	Presents the data and application functionalities to the user, often focusing on the visual aspects of the application.

The three-layered model represents a traditional approach to web application architecture, with a straightforward separation of concerns. However, it does not explicitly integrate security and accessibility within its core layers, potentially leading to gaps in these critical areas.

5.2. Analytical Comparison of the Models

To illustrate the differences between HSALA and the three-layered architecture, we examine several parameters: accessibility, cybersecurity, scalability, user experience (UX/UI), and additional considerations such as technology choices and development overhead.

5.2.1. Accessibility Features

As shown in Table 3, HSALA includes accessibility considerations at each level, with a dedicated Accessibility Enhancement Layer at the top. In contrast, the traditional three-layered model typically treats accessibility as an add-on rather than a core requirement, risking incomplete or inconsistent accessibility support.

HSALA provides a proactive and comprehensive approach to accessibility, embedding it within the architectural framework and ensuring compliance with standards. In contrast, the three-layered model's reactive and fragmented approach can lead to inconsistent accessibility features and potential security vulnerabilities related to accessibility implementations.

Table 3. Comparison of the two architectural models considering accessibility features.

Aspect	HSALA	Three-Layered Model
Integration Level	Accessibility is embedded within the architecture through the Accessibility Enhancement Layer.	Accessibility features are often added after the core development is completed.
Compliance with Standards	Ensures adherence to WCAG and uses WAI-ARIA for assistive technologies.	May not fully comply with accessibility standards due to retrofitting challenges.
Holistic Approach	Accessibility is considered at every stage, from data representation to user interaction.	Lacks a dedicated layer, leading to possible inconsistencies and gaps in accessibility.
Accessibility Security	The Accessibility Security Layer addresses potential security risks introduced by accessibility features, ensuring secure and inclusive access.	Accessibility features might inadvertently introduce vulnerabilities without a dedicated security focus.

5.2.2. Cybersecurity Features

Cybersecurity is another critical area. Table 4 highlights how security is built into each HSALA layer, while the three-layered model often relies on ad hoc or reactive measures.

Table 4. Comparison of the two architectural models considering cybersecurity features.

Aspect	HSALA	Three-Layered Model
Security Integration	Each functional layer is paired with a security layer, providing a defense-in-depth approach.	Security measures are often added reactively in response to identified threats.
Proactive Measures	Incorporates security from the ground up, including data encryption, secure coding practices, and continuous monitoring.	May lack dedicated security layers, leading to gaps in protection and inconsistent security practices.
Regulatory Compliance	Facilitates adherence to data protection laws and industry standards (e.g., GDPR, HIPAA).	Ensuring regulatory compliance can be more complex due to the absence of integrated security considerations.
Security-Aware Development	Developers are guided to consider security implications at every stage due to the integrated security layers.	Security depends heavily on individual developers' awareness and practices, leading to potential oversights.

HSALA's integrated security approach ensures that cybersecurity is a foundational element of the application, reducing vulnerabilities and aiding in the compliance with regulations. The three-layered model's lack of integrated security can result in reactive measures that may not be sufficient to protect against modern threats.

5.2.3. Scalability

The scalability of the two approaches can vary significantly. As shown in Table 5, the multi-layered structure of HSALA offers a higher degree of modularity and adaptability.

Table 5. Comparison of the two architectural models considering scalability.

Aspect	HSALA	Three-Layered Model
Modularity	The multi-layered architecture allows for the independent scaling of each layer based on demand.	Tightly coupled layers can hinder scalability and flexibility.
Maintainability	Easier to update or replace components without affecting the entire system due to the separation of concerns.	Changes in one layer may require significant adjustments in others, complicating maintenance.
Performance Optimization	Security and accessibility features are optimized within their layers to minimize performance impact.	Difficulty in isolating and optimizing specific components can lead to inefficiencies and performance bottlenecks.
Future-Proofing	Adaptable to new technologies and standards due to its flexible structure and modularity.	Incorporating new technologies or standards may be challenging due to the rigid structure.

HSALA's modular design facilitates scalability and adaptability, allowing the application to evolve with changing demands and technologies. The three-layered model may face challenges in scaling and updating due to its less flexible architecture.

5.2.4. User Experience (UX/UI)

Providing a good user experience is essential for the effective use and long-term success of a web application. Table 6 contrasts the UX/UI implications in both architectures.

Table 6. Comparison of the two architectural models considering UX/UI.

Aspect	HSALA	Three-Layered Model
UX/UI Focus	The Content Presentation Layer focuses on delivering an intuitive and responsive user interface with an emphasis on UX/UI design.	The Presentation Layer handles both presentation and interaction, potentially overloading its scope.
Enhanced Accessibility	Improves user experience for all users, including those with disabilities, through the Accessibility Enhancement Layer.	Retrofitting accessibility can compromise design consistency and user experience.
Security Enhancements	The Content Presentation Security Layer ensures that security measures do not disrupt the user experience.	Security prompts or measures may interfere with usability if not carefully integrated due to the lack of dedicated layers.
Consistent Design	Separation of concerns leads to a cohesive and maintainable UX/UI design, facilitating updates and consistency.	Less flexibility in adopting new UX/UI trends due to tightly coupled layers and potential conflicting priorities.

HSALA promotes a superior user experience by dedicating layers to UX/UI and accessibility, ensuring that the interface is both user-friendly and secure. The three-layered model's approach may result in a compromised UX/UI due to conflicting priorities and the insufficient integration of accessibility and security considerations.

5.2.5. Additional Parameters

Several other factors, such as technologies used and development/maintenance overhead, also differ between the two models. Table 7 provides an overview of how technology choices can be integrated, while Table 8 focuses on development and maintenance considerations.

Table 7. Comparison of the two architectural models considering technologies used.

Aspect	HSALA	Three-Layered Model
Layer-Specific Technologies	Allows for the use of the most appropriate technologies at each layer (e.g., different databases, programming languages and frameworks).	May restrict the application to a single technology stack, limiting flexibility.
Innovation-Friendly	Facilitates experimentation with new tools and technologies without overhauling the entire system due to modularity.	Adopting new technologies may require significant rework and pose challenges due to the monolithic structure.
Security Tools Integration	Seamless incorporation of security technologies, like encryption libraries, authentication mechanisms, and security testing tools, at each layer.	Difficulty in integrating advanced security tools due to architectural constraints and lack of dedicated security layers.

Table 8. Comparison of the two architectural models considering development and maintenance.

Aspect	HSALA	Three-Layered Model
Parallel Development	Teams can work on different layers simultaneously, improving development efficiency and reducing the time-to-market.	Layers are often developed in a linear fashion, which can slow down the process and create bottlenecks.
Simplified Maintenance	Isolated layers make it easier to identify and fix issues without impacting unrelated components, enhancing maintainability.	Interdependencies can complicate troubleshooting and updates, increasing maintenance overhead.
Documentation and Standards	Encourages better documentation and adherence to development standards due to the clear separation of concerns and responsibilities.	Without clear separation, development standards may vary across the application, leading to inconsistencies and potential integration issues.

As illustrated in Table 7, HSALA's modular structure accommodates diverse technologies more easily.

HSALA's flexibility in adopting layer-specific technologies enhances innovation and allows for the integration of advanced tools, particularly in security and accessibility. The three-layered model's limitations may hinder technological advancements and the implementation of specialized tools.

Table 8 contrasts the efficiency and maintainability of each model.

HSALA's modularity and separation of concerns streamline development and maintenance processes, enabling more efficient workflows and consistent standards. In contrast, the three-layered model may encounter inefficiencies and inconsistencies due to its less structured approach.

5.3. Theoretical Validation of HSALA

This subsection provides a high-level validation of HSALA against key software engineering principles and accessibility/security standards, along with a detailed analysis of its alignment with best practices.

5.3.1. Alignment with Software Engineering Principles

As shown in Table 9, the HSALA model aligns strongly with modularity, encapsulation, and separation of concerns.

Table 9. HSALA’s alignment with modularity, encapsulation, and separation of concerns principles.

Principle	HSALA Alignment
Separation of Concerns	Divides the application into distinct layers, each with a specific responsibility, reducing complexity and enhancing clarity.
Modularity	Enhances maintainability and scalability by allowing independent development and updates of each layer.
Encapsulation	Each layer encapsulates its functionality and security measures, preventing unintended interactions and side effects between layers.

HSALA aligns closely with core software engineering principles, promoting a well-organized and manageable architecture that facilitates development, maintenance, and scalability.

5.3.2. Compliance with Accessibility Standards

In Table 10, we see how HSALA addresses key accessibility guidelines and legal requirements.

Table 10. HSALA’s compliance with WCAG and legal requirements.

Aspect	HSALA Alignment
WCAG Compliance	The Accessibility Enhancement Layer ensures compliance with WCAG 2.1 guidelines, promoting inclusivity and accessibility for individuals with disabilities.
Legal Requirements	Aligns with regulations like the Americans with Disabilities Act (ADA) and the European Accessibility Act (EAA), mitigating legal risks and ensuring equitable access.

HSALA’s dedicated focus on accessibility ensures that applications not only meet technical standards but also comply with legal requirements, fostering an inclusive digital environment.

5.3.3. Adherence to Security Best Practices

Table 11 illustrates how HSALA embodies the defense-in-depth principle and integrates security throughout the development lifecycle.

HSALA’s integrated security approach aligns with the industry’s best practices, enhancing the application’s resilience against threats and ensuring the compliance with stringent data protection regulations.

Table 11. HSALA’s alignment with security best practices.

Aspect	HSALA Alignment
Defense-in-Depth	Multiple security layers provide comprehensive protection against various attack vectors, embodying the defense-in-depth strategy in cybersecurity.
Secure Development Lifecycle	Security considerations are integrated from the outset, aligning with frameworks like Microsoft’s Security Development Lifecycle (SDL) [64] for robust security practices.
Regulatory Compliance	Facilitates adherence to data protection laws like GDPR and HIPAA, reducing the risk of non-compliance penalties and protecting user data.

5.3.4. Enhancement of User Trust and Satisfaction

As indicated in Table 12, combining solid security with accessibility measures fosters greater user satisfaction and trust.

Table 12. Impact on improved user experience and trust through security.

Aspect	HSALA Impact
Improved User Experience	Prioritizing accessibility and usability enhances user satisfaction and engagement across with diverse user groups, including those disabilities.
Trust through Security	Robust security measures build user trust, which is essential for applications handling sensitive data and transactions, leading to increased user confidence.

By addressing both accessibility and security comprehensively, HSALA fosters a trustworthy and satisfying user experience, which is critical for the success and reputation of web applications.

5.3.5. Risk Mitigation

Finally, Table 13 highlights how HSALA helps reduce technical debt and supports long-term adaptability in a fast-changing digital landscape.

Table 13. HSALA’s benefits in reduction in technical debt and adaptability to change.

Aspect	HSALA Benefit
Reduction in Technical Debt	Proactive integration of accessibility and security reduces future costs associated with retrofitting and patching, leading to long-term savings and efficiency.
Adaptability to Change	The model’s flexibility allows for the easy incorporation of new requirements and technologies, keeping the application relevant and up-to-date in a rapidly evolving landscape.

HSALA addresses the risks related to technical debt and obsolescence through a proactive and adaptable architectural strategy, thereby enhancing the long-term viability and sustainability of the application.

5.4. Summary of Comparative Analysis

The comparison highlights the significant advantages of HSALA over the three-layered model in addressing contemporary challenges in web application development. By integrating accessibility and cybersecurity at each architectural layer, HSALA ensures the following:

- Comprehensive accessibility: meeting or exceeding accessibility standards, enhancing inclusivity for all users.
- Robust cybersecurity: implementing proactive and layered security measures, reducing vulnerabilities and building trust.
- Scalability and flexibility: facilitating growth and the adaptation to new technologies without compromising performance or maintainability.
- Enhanced user experience: providing a seamless, intuitive, and secure interface, balancing aesthetics and functionality.
- Regulatory compliance: aligning with legal and industry standards, mitigating compliance risks and potential legal challenges.

In contrast, the three-layered model, while simpler, may fall short in effectively integrating accessibility and security, potentially leading to fragmented user experiences, security vulnerabilities, scalability issues, and higher long-term costs due to technical debt.

6. Conclusions and Future Work

In the modern age of information, both accessibility and cybersecurity are crucial for successfully embracing digital transformation through enhanced user-centered digital solutions. Cybersecurity not only provides secure access to user data but is also vital for business continuity, while accessibility ensures inclusion and information access to a wide audience. Protecting the most vulnerable individuals is essential for a functional modern society, preventing social isolation and a diminished quality of life for people with disabilities, as well as safeguarding all users from various cyber threats.

Despite the increasing need for digital inclusion and cybersecurity, there exists a gap between security-oriented development models and accessibility-oriented ones. The proposed Holistic Security and Accessibility Layered Architecture (HSALA) aims to improve existing solutions and bridge this gap by embracing the best practices of both fields. By establishing accessible and cyber-safe user interfaces, HSALA ensures resilient, robust, and inclusive web-based applications and websites, thereby promoting digital inclusion. Integrating accessible technologies and techniques into secure, cyber-resilient web interfaces requires a holistic perspective. This approach optimizes both accessibility and security throughout the planning, implementation, and maintenance stages. The model integrates open-source technologies with cybersecurity and accessibility design principles, offering advantages such as transparency, cost-effectiveness, and reusability. Leveraging these benefits allows developers to design web pages that are easy to use and safe, yet strong and flexible enough to meet future challenges. Prioritizing accessibility and security enables the emergence of a more secure and equitable digital ecosystem that benefits all users, regardless of age, ability, or the nature of their interaction with the web.

However, several limitations of the current work must be acknowledged. Although potential performance issues have been discussed theoretically, actual metrics—such as response times, resource utilization, and scalability—have not been measured. Implementing HSALA may present practical challenges. The increased complexity of a multi-layered architecture could lead to higher development and maintenance costs. Developers may require additional training to effectively implement and manage the integrated security and accessibility measures across all layers. Although the model emphasizes user-centered design and the involvement of users with disabilities, actual user testing has not been

conducted. This limitation restricts our ability to validate the model's inclusivity and usability from the perspective of end-users.

To address these limitations and enhance the practical relevance of HSALA, several avenues for future work are proposed. Foremost among these is the empirical validation of the model through real-world implementation. Collaborating with industry partners and organizations to apply HSALA in existing or new web applications will provide opportunities to test the model in diverse environments. Such collaborations would facilitate the collection of empirical data on effectiveness, security, accessibility, and performance. This would provide concrete evidence of the model's applicability.

Performance testing and optimization constitute another critical area for future work. Comprehensive performance assessments are necessary to evaluate the influence of additional layers on response times, scalability, and resource usage. Based on these findings, optimization techniques can be explored to mitigate any negative effects. Strategies such as leveraging microservices to decouple layers, implementing asynchronous processing, and utilizing edge computing or content delivery networks (CDNs) to reduce latency could enhance performance without compromising the model's integrity.

Involving real users, particularly those with disabilities, in the development and testing process is crucial for validating the accessibility features of HSALA. Engaging users through usability studies and focus groups will provide valuable qualitative feedback on usability and user experience. This user-centered approach ensures that the application meets the diverse needs of its intended audience and adheres to the principles of inclusive design.

We also plan to upgrade and broaden the incorporation of semantic web technique [65], such as the Web Ontology Language (OWL) and the Resource Description Framework (RDF), as well as integrate artificial intelligence (AI)-driven personalization and ensure the quality of experience (QoE) [66]. The combination of semantic web technologies, AI, and a focus on QoE can create web interfaces that are not only secure and accessible but also provide a smooth, intelligent, and enjoyable user experience. Features such as fast loading times, intuitive and responsive interfaces, and custom interactions can significantly enhance user satisfaction. Incorporating AI-driven personalization enables interfaces to adapt dynamically to user preferences, thereby enhancing both accessibility and overall user experience. Developing machine learning models capable of predicting user requirements, enhancing security by detecting anomalous behavior, and adjusting content presentation can further strengthen the model's capabilities.

In conclusion, while HSALA represents a significant advancement in integrating accessibility and cybersecurity within web application development, practical implementation and empirical validation are necessary to fully realize its potential. By addressing the identified limitations through the proposed future work, HSALA can evolve into a robust framework that meets contemporary demands for inclusive and secure digital platforms. The ongoing refinement of the model, engagement with real users, and embracing technological innovations will contribute to creating web applications that are resilient, user-friendly, and equitable for all users. Future research and collaboration will not only validate HSALA's effectiveness but also drive progress in developing digital ecosystems that prioritize both security and accessibility. This holistic approach is essential for fostering inclusivity and trust in an increasingly digital society.

Author Contributions: Conceptualization, G.A.S. and L.S.; methodology, L.S. and N.E.-D.; resources, L.S.; writing—original draft preparation, G.A.S., L.S. and N.E.-D.; writing—review and editing, G.A.S., L.S. and N.E.-D.; supervision, G.A.S. and L.S.; funding acquisition, L.S. and N.E.-D. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Transilvania University of Brasov.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Furnell, S.; Helkala, K.; Woods, N. Disadvantaged by Disability: Examining the Accessibility of Cyber Security. In *Universal Access in Human-Computer Interaction. Design Methods and User Experience*; Antona, M., Stephanidis, C., Eds.; Springer: Cham, Switzerland, 2021; Volume 12768. [\[CrossRef\]](#)
2. Chivukula, R.; Jaya Lakshmi, T.; Ranganadha Reddy Kandula, L.; Alla, K. A Study of Cyber Security Issues and Challenges. In Proceedings of the IEEE Bombay Section Signature Conference (IBSSC), Gwalior, India, 18–20 November 2021; pp. 1–5. [\[CrossRef\]](#)
3. Todorov, T.; Dochkova-Todorova, J. Accessible UX/UI Design. In Proceedings of the International Conference Automatics and Informatics (ICAI), Varna, Bulgaria, 5–7 October 2023; pp. 362–366. [\[CrossRef\]](#)
4. Web Content Accessibility Guidelines (WCAG) 2.1. Available online: <https://www.w3.org/TR/WCAG21/> (accessed on 29 December 2024).
5. W3C. Available online: <https://www.w3.org/> (accessed on 29 December 2024).
6. Home | Web Accessibility Initiative (WAI) | W3C. Available online: <https://www.w3.org/WAI/> (accessed on 29 December 2024).
7. Rodrigues, S.S.; Fortes, R.P.d.M. A Checklist for the Evaluation of Web Accessibility and Usability for Brazilian Older Adults. *J. Web Eng.* **2020**, *19*, 63–108. [\[CrossRef\]](#)
8. Kurt, S. Moving toward a universally accessible web: Web accessibility and education. *Assist. Technol.* **2018**, *31*, 199–208. [\[CrossRef\]](#)
9. WAI-ARIA Overview | Web Accessibility Initiative (WAI) | W3C. Available online: <https://www.w3.org/WAI/standards-guidelines/aria/> (accessed on 31 December 2024).
10. Stelea, G.A.; Fernoaga, V.; Gavrilă, C.; Popescu, V.; Murroni, M. Mobile Accessible Rich Internet Web Application Enhanced with AMP Publishing Technology. *Rev. Air Force Acad.* **2019**, *17*, 71–78. [\[CrossRef\]](#)
11. Ahsan, M.; Nygard, K.E.; Gomes, R.; Chowdhury, M.M.; Rifat, N.; Connolly, J.F. Cybersecurity Threats and Their Mitigation Approaches Using Machine Learning—A Review. *J. Cybersecur. Priv.* **2022**, *2*, 527–555. [\[CrossRef\]](#)
12. Leszczyna, R. Review of cybersecurity assessment methods: Applicability perspective. *Comput. Secur.* **2021**, *108*, 102376. [\[CrossRef\]](#)
13. Lee, N. Cyberattacks, Prevention, and Countermeasures. In *Counterterrorism and Cybersecurity*; Springer: Cham, Switzerland, 2024; pp. 295–342. [\[CrossRef\]](#)
14. Grigaliūnas, Š.; Schmidt, M.; Brūzgienė, R.; Smyrli, P.; Andreou, S.; Lopata, A. Holistic Information Security Management and Compliance Framework. *Electronics* **2024**, *13*, 3955. [\[CrossRef\]](#)
15. Stige, Å.; Zamani, E.D.; Mikalef, P.; Zhu, Y. Artificial intelligence (AI) for user experience (UX) design: A systematic literature review and future research agenda. *Inf. Technol. People* **2024**, *37*, 2324–2352. [\[CrossRef\]](#)
16. Azevedo, D.; Ramelot, J.; Legay, A.; Kieffer, S. With or Without U(sers): A Journey to Integrate UX Activities in Cybersecurity. In *HCI for Cybersecurity, Privacy and Trust*; Moallem, A., Ed.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2024; Volume 14728, pp. 212–231. [\[CrossRef\]](#)
17. Sanchez-Gordon, S.; Aguilar-Mayanquer, C.; Calle-Jimenez, T. Model for Profiling Users with Disabilities on e-Learning Platforms. *IEEE Access* **2021**, *9*, 74258–74274. [\[CrossRef\]](#)
18. Kuba, R.; Jeong, A. Demystifying visual design: A sequential analysis of design processes in infographic visual composition. *J. Vis. Lit.* **2023**, *42*, 26–47. [\[CrossRef\]](#)
19. Chaudhary, S.; Kompara, M.; Pape, S.; Gkioulos, V. Properties for Cybersecurity Awareness Posters’ Design and Quality Assessment. In Proceedings of the 17th International Conference on Availability, Reliability and Security (ARES ’22). Association for Computing Machinery, New York, NY, USA, 23–26 August 2022; pp. 1–8. [\[CrossRef\]](#)
20. Renaud, K. Accessible cyber security: The next frontier? In Proceedings of the 7th International Conference on Information Systems Security and Privacy, ICISPP, Online, 11–13 February 2021; Mori, P., Gabriele, L., Furnell, S., Eds.; SciTePress: Setúbal, Portugal, 2021; pp. 9–18. [\[CrossRef\]](#)
21. Renaud, K.; Coles-Kemp, L. Accessible and Inclusive Cyber Security: A Nuanced and Complex Challenge. *SN Comput. Sci.* **2022**, *3*, 346. [\[CrossRef\]](#)
22. Kävrestad, J.; Rambusch, J.; Nohlberg, M. Design principles for cognitively accessible cybersecurity training. *Comput. Secur.* **2024**, *137*, 103630. [\[CrossRef\]](#)

23. What Is Three-Tier Architecture? | IBM. Available online: <https://www.ibm.com/topics/three-tier-architecture> (accessed on 20 November 2024).
24. Marinho, E.H.; Mendonca, A.A.; Rodrigues, G.N.; Alves, V.; Bonifácio, R. Exploring Architecture-Based Reliability Analysis of Current Multi-layered Web Applications. In Proceedings of the 2011 Fifth Brazilian Symposium on Software Components, Architectures and Reuse, Sao Paulo, Brazil, 26–27 September 2011; pp. 51–60. [CrossRef]
25. Villamizar, M.; Garcés, O.; Castro, H.; Verano, M.; Salamanca, L.; Casallas, R.; Gil, S. Evaluating the monolithic and the microservice architecture pattern to deploy web applications in the cloud. In Proceedings of the 10th Computing Colombian Conference (10CCC), Bogota, Colombia, 21–25 September 2015; pp. 583–590. [CrossRef]
26. Fanea-Ivanovici, M.; Pană, M.-C. From Culture to Smart Culture. How Digital Transformations Enhance Citizens' Well-Being Through Better Cultural Accessibility and Inclusion. *IEEE Access* **2020**, *8*, 37988–38000. [CrossRef]
27. Stelea, G.A.; Popescu, V.; Sandu, F.; Jalal, L.; Farina, M.; Murrone, M. From Things to Services: A Social IoT Approach for Tourist Service Management. *IEEE Access* **2020**, *8*, 153578–153588. [CrossRef]
28. Ferri, D.; Favalli, S. Web Accessibility for People with Disabilities in the European Union: Paving the Road to Social Inclusion. *Societies* **2018**, *8*, 40. [CrossRef]
29. Lewandowski, D.; Sünkler, S.; Yagci, N. The influence of search engine optimization on Google's results: A multi-dimensional approach for detecting SEO. In Proceedings of the 13th ACM Web Science Conference (WebSci '21) Association for Computing Machinery, New York, NY, USA, 21–25 June 2021. [CrossRef]
30. European Accessibility Act—European Commission. Available online: https://employment-social-affairs.ec.europa.eu/policies-and-activities/social-protection-social-inclusion/persons-disabilities/union-equality-strategy-rights-persons-disabilities-2021-2030/european-accessibility-act_en (accessed on 30 December 2024).
31. The Americans with Disabilities Act | ADA.gov. Available online: <https://www.ada.gov/> (accessed on 30 December 2024).
32. Regulation-2016/679-EN-gdpr-EUR-Lex. Available online: <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32016R0679> (accessed on 31 December 2024).
33. California Consumer Privacy Act (CCPA) | State of California—Department of Justice—Office of the Attorney General. Available online: <https://oag.ca.gov/privacy/ccpa> (accessed on 31 December 2024).
34. Atac Ciberetic Masiv de Tip Ransomware Asupra Serverelor de Producție Pe Care Rulează Sistemul Informatic HIS—Ministerul Sănătății. Available online: <https://ms.ro/ro/centrul-de-presa/atac-ciberetic-masiv-de-tip-ransomware-asupra-serverelor-de-produc%C8%9Bie-pe-care-ruleaz%C4%83-sistemul-informatic-his/> (accessed on 19 June 2024).
35. Best Practice Framework for Online Safety Education | eSafety Commissioner. Available online: <https://www.esafety.gov.au/educators/best-practice-framework> (accessed on 29 December 2024).
36. Children's Online Privacy Protection Rule (COPPA) | Federal Trade Commission. Available online: <https://www.ftc.gov/legal-library/browse/rules/childrens-online-privacy-protection-rule-coppa> (accessed on 30 December 2024).
37. Online Safety | eSafety Commissioner. Available online: <https://www.esafety.gov.au/> (accessed on 29 December 2024).
38. Fischer, M.E.; Cruickshanks, K.J.; Schubert, C.R.; Pinto, A.A.; Carlsson, C.M.; Klein, B.E.K.; Klein, R.; Tweed, T.S. Age-Related Sensory Impairments and Risk of Cognitive Impairment. *J. Am. Geriatr. Soc.* **2016**, *64*, 1981–1987. [CrossRef]
39. Savaş, S.; Karataş, S. Cyber governance studies in ensuring cybersecurity: An overview of cybersecurity governance. *Int. Cybersecur. Law Rev.* **2022**, *3*, 7–34. [CrossRef]
40. Williams, R. *The Non-Designer's Design Book: Design and Typographic Principles for the Visual Novice*; Peachpit Press: Berkeley, CA, USA, 2014.
41. The Crucial Role of UI/UX in Cybersecurity Products—UX Parrot. Available online: <https://uxparrot.com/the-crucial-role-of-ui-ux-in-cybersecurity-products/> (accessed on 31 December 2024).
42. UX vs. UI: Impact on Cybersecurity—Blue Goat Cyber. Available online: <https://bluegoatcyber.com/blog/ux-vs-ui-impact-on-cybersecurity/> (accessed on 31 December 2024).
43. Zakaria, M.Z.; Kadir, R. Risk Assessment of Web Application Penetration Testing on Cross-Site Request Forgery (CSRF) Attacks and Server-Side Includes (SSI) Injections. In Proceedings of the 2021 International Conference on Data Science and Its Applications (ICoDSA), Bandung, Indonesia, 6–7 October 2021; pp. 85–90. [CrossRef]
44. Buccafurri, F.; De Angelis, V.; Lazzaro, S.; Pugliese, A. Enforcing security policies on interacting authentication systems. *Comput. Secur.* **2024**, *140*, 103771. [CrossRef]
45. Ometov, A.; Bezzateev, S.; Mäkitalo, N.; Andreev, S.; Mikkonen, T.; Koucheryavy, Y. Multi-Factor Authentication: A Survey. *Cryptography* **2018**, *2*, 1. [CrossRef]
46. Ilca, L.F.; Lucian, O.P.; Balan, T.C. Enhancing Cyber-Resilience for Small and Medium-Sized Organizations with Prescriptive Malware Analysis, Detection and Response. *Sensors* **2023**, *23*, 6757. [CrossRef] [PubMed]
47. Lina, L. Personalized e-Learning System Based on Multi-layer Architecture. In Proceedings of the International Forum on Information Technology and Applications, Chengdu, China, 15–17 May 2009; pp. 278–281. [CrossRef]

48. Cuzzocrea, A.; Shahriar, H. Data masking techniques for NoSQL database security: A systematic review. In Proceedings of the 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, USA, 11–14 December 2017; pp. 4467–4473. [\[CrossRef\]](#)
49. Khader, M.; Karam, M. Assessing the Effectiveness of Masking and Encryption in Safeguarding the Identity of Social Media Publishers from Advanced Metadata Analysis. *Data* **2023**, *8*, 105. [\[CrossRef\]](#)
50. Moore, W.; Frye, S. Review of HIPAA, Part 1: History, Protected Health Information, and Privacy and Security Rules. *J. Nucl. Med. Technol.* **2019**, *47*, 269–272. [\[CrossRef\]](#) [\[PubMed\]](#)
51. Chakraborty, S.; Aithal, P.S. CRUD Operation on WordPress Database Using C# And REST API (November 23, 2023). *Int. J. Appl. Eng. Manag. Lett.* **2023**, *7*, 130–138. [\[CrossRef\]](#)
52. Uddin, M.; Islam, S.; Al-Nemrat, A. A Dynamic Access Control Model Using Authorising Workflow and Task-Role-Based Access Control. *IEEE Access* **2019**, *7*, 166676–166689. [\[CrossRef\]](#)
53. The Web Framework for Perfectionists with Deadlines | Django. Available online: <https://www.djangoproject.com/> (accessed on 31 December 2024).
54. Laravel—The PHP Framework For Web Artisans. Available online: <https://laravel.com/> (accessed on 31 December 2024).
55. Spring | Home. Available online: <https://spring.io/> (accessed on 31 December 2024).
56. ASP.NET Core | Open-Source Web Framework for .NET. Available online: <https://dotnet.microsoft.com/en-us/apps/aspnet> (accessed on 31 December 2024).
57. Makiou, A.; Begriche, Y.; Serhrouchni, A. Improving Web Application Firewalls to detect advanced SQL injection attacks. In Proceedings of the 10th International Conference on Information Assurance and Security, Okinawa, Japan, 28–30 November 2014; pp. 35–40. [\[CrossRef\]](#)
58. Stelea, G.A.; Murroni, M.; Popescu, V.; Balan, T.; Fernoaga, V. Enhanced Resource Management for Web Based Thin Clients Using Cross-Platform Progressive Offline Capabilities. In *Cognitive Radio-Oriented Wireless Networks, CrownCom 2019*; Kliks, A., Kryszkiewicz, P., Bader, F., Triantafyllou, D., Caicedo, C.E., Sezgin, A., Dimitriou, N., Sybis, M., Eds.; Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering; Springer: Berlin/Heidelberg, Germany, 2019; Volume 291, pp. 361–372. [\[CrossRef\]](#)
59. Calzavara, S.; Rabitti, A.; Bugliesi, M. Semantics-Based Analysis of Content Security Policy Deployment. *ACM Trans. Web (TWEB)* **2018**, *12*, 1–36. [\[CrossRef\]](#)
60. Dora, J.R.; Nemoga, K. Ontology for Cross-Site-Scripting (XSS) Attack in Cybersecurity. *J. Cybersecur. Priv.* **2021**, *1*, 319–339. [\[CrossRef\]](#)
61. Buzzi, M.; Leporini, B.; Romano, F. Exploring WAI-Aria Techniques to Enhance Screen Reader Interaction: The Case of a Portal for Rating Accessibility of Cultural Heritage Sites. In *Universal Access in Human-Computer Interaction. Design Approaches and Supporting Technologies*; Antona, M., Stephanidis, C., Eds.; Springer: Cham, Switzerland, 2020; Volume 12188, pp. 245–260. [\[CrossRef\]](#)
62. Hristov, H.; Glushkova, T.; Cheresharov, S.; Stoeva, M. A Model for Designing Accessible Color and Contrast for Users with Visual Deficiency and Color Blindness. In Proceedings of the IEEE 11th International Conference on Intelligent Systems (IS), Warsaw, Poland, 12–14 October 2022; pp. 1–7. [\[CrossRef\]](#)
63. Kushalnagar, R. Deafness and Hearing Loss. In *Web Accessibility*; Yesilada, Y., Harper, S., Eds.; Human-Computer Interaction Series 2019; Springer: London, UK, 2019. [\[CrossRef\]](#)
64. Roy, S.; Mudundi, P.S. Implementing a Secure Web-Based Application Using Microsoft SDL. In *Computer and Cyber Security*; Auerbach Publications: Boca Raton, FL, USA, 2018; pp. 355–388. [\[CrossRef\]](#)
65. Berners-Lee, T.; Hendler, J.; Lassila, O. The Semantic Web: A New Form of Web Content that is Meaningful to Computers will Unleash a Revolution of New Possibilities. In *Linking the World's Information: Essays on Tim Berners-Lee's Invention of the World Wide Web*, 1st ed.; Association for Computing Machinery: New York, NY, USA, 2023; pp. 91–103. [\[CrossRef\]](#)
66. Laghari, A.A.; Zhang, X.; Shaikh, Z.A.; Khan, A.; Estrela, V.V.; Izadi, S. A review on quality of experience (QoE) in cloud computing. *J. Reliab. Intell. Environ.* **2024**, *10*, 107–121. [\[CrossRef\]](#)

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.