**Web Application Vulnerability Scanning**



Braveil Robinson

25846726

25846726@students.lincoln.ac.uk

School of Computer Science

College of Science

University of Lincoln

Submitted in partial fulfilment of the requirements for the

Degree of BSc(Hons) Computer Science

*Supervisor:* Dr. Saeid Pourroostaei Ardakani

April 2022

**Acknowledgements**

Firstly, I want to thank somebody, and somebody else. Here is another thing.

**Abstract**

Here is the abstract for this project report.

**Table of Contents**

[Introduction 6](#_Toc1014023241)

[1.1 aims and objectives 7](#_Toc1075020218)

[1.2 Figures and Tables 7](#_Toc1909800212)

[1.3Referencing 8](#_Toc793364672)

[1.3.1Ludography 8](#_Toc386676326)

[Literature Review 8](#_Toc255440797)

[2.1 Aims & Objectives 9](#_Toc288184468)

[Requirements Analysis 9](#_Toc358647062)

[Design & Methodology 11](#_Toc625428168)

[4.1 Software development projects 11](#_Toc1299181326)

[4.2 Research not involving human participants 11](#_Toc628344767)

[4.3 Research involving human participants 12](#_Toc209212695)

[Implementation 13](#_Toc647310710)

[Results & Discussion 14](#_Toc1701354224)

[Conclusion 15](#_Toc1980108475)

[References 15](#_Toc1444658122)

**List of Figures**

1. A picture of the Brayford from Google Images. . . . . . . . . . . . . . 5

**List of Tables**

1. Here is a table. . . . . . . . . . . . 5

**Chapter 1**

# Introduction

## 1.1 motivation

The broad adoption of web applications has become essential to contemporary business and communication in todays digital era. Researchers have conducted substantial research in this subject to improve security measures. (Huang et al., 2004) conducted research on the use of static analysis to detect and eradicate vulnerabilities, emphasising the importance of taking a proactive approach to online application security. This study points out the key need to improve security measures as we increasingly rely on digital solutions. There are serious cybersecurity risks which may be associated with our reliance on online web applications in our daily lives as Most of the web applications have resulted in the increased adoption of both frequency and sophistication in cyber threats. Web application vulnerabilities provides chances for malicious assaults, which can have disastrous financial and reputation effects (Li et al., 2016). These assaults can involve data breaches and unauthorised access into personal information. The scope of cyber threats is dynamic at every new change, which requires improved security measures.(Maimon et al.,2017) propounded that a need to determine situational motivations in an online environment calls for monitoring as a crucial mechanism for prediction and prevention of cyber-dependant crimes. These vulnerabilities give rise to both the technical and financial challenges effecting reputational damage. The increasing complexity and frequency of these cyberattacks highlight the essential need for more advanced security measures, particularly at a time when people are relying more and more on digital solutions. This is what motivated me to take this initiative and develop my web application vulnerability scanner. This dissertation aims to showcases the creation of an automated Web application security scanner that scans websites for their URL and notifies the user of the potential vulnerabilities in that online application. What makes this tool unique is that it is one of a kind, with dual capacity software, easily understandable by non technical users, yet robust enough to be uses by the capability of a cyber security professional. Therefore In comparison to previous scanners from the past this scanner intends to provide a user friendly interface which extends its capabilities to detect a diverse range of vulnerabilities. This scanners user friendly interface, which is made to accommodate both security experts and people with less technical knowledge, is one of its main differentiators. This approachability increases the tools usefulness and expands its application without sacrificing its functionality at any cost. The scanner represents a major advancement in the accessibility of complex cybersecurity technologies to a broader audience due to its efficiency in analysing online applications and providing findings in an understandable fashion. In addition to scanning clickjacking, and XXS the scanner will be able to focus on basic SQL vulnerabilities including union-based, time-based, error-based, and blind/out-of-band injections. This strategy aims to provide web applications with a more comprehensive defence mechanism that considers the constantly evolving environment of cyber threats (Zhao and LV, 2007).

## 1.2 report Outline

This dissertation evolves into a thorough literature review after the introduction, which places the reader in the context of the urgency and inventiveness behind the creation of the web application vulnerability scanner. This section examines the most recent research on web application security from both academic and industrial sources. The literature study will aim to highlight and identify the current weaknesses in web application security by analysing earlier studies and commercially available tools. The dissertation then shifts into the requirement analysis, moving from theory to specifics. This chapter carefully describes the needs for the security scanner, dividing them into requirements that are both functional and non-functional. Non-functional criteria highlight the qualities of the scanner such as its functionality and usability, whereas functional requirements specific the fundamental activities the scanner must carry out, such as identifying different vulnerabilities. The security professionals who require depth and the users who need simplicity of use are the two main user groups whose viewpoints are taken into consideration in this research.

In the design and methodology, the project's organised plan is outlined in this section, considering the tools and methods of software development that were carefully chosen and their justifications. This section shows that strict technical standards are upheld in addition to a dedication to a user-centric design philosophy. Along the way, this section later on reveals all of the decisions made for the project and gives a thorough explanation of each, supported by arguments that support the projects objectives.

In the implementation section you will be taken through a narrative of the artifacts creations. This chapter details the real-world difficulties encountered and the calculated detours taken to maintain the projects objectives. With illustrative code snippets and examples that highlight the difficulties and success of the development process, it illustrates the iterative process of creating the user interface and back-end components.

The results and discussions are presented in the penultimate section of the dissertation. This vital chapter describes the rigorous testing phase and presents quantitative and qualitative statistics that evaluate the effectiveness of the scanner in relation to the predetermined goals. A careful consideration of the scanner degree of target-meeting is included in the nuanced discussion, along with insightful suggestions for enhancements. The purpose of this retrospective analysis is twofold: it confirms the effectiveness of the scanner and considers the projects contribution to the larger field.

A thorough synopsis that captures the accomplishments of the project is included in the conclusion. The conclusion looks ahead, considering future directions for research as well as possible improvements , rather than merely looking back. Instead of viewing the security scanner as a static solution, it sees it as a first step towards a dynamic, ever-evolving defence against cyberthreats of the future. This last section, then, closes the entire narrative and sets a stage for future innovations in web application security.

**Chapter 2**

# Literature Review

The ubiquity of web applications in sectors from e-commerce to healthcare underscores the pressing need for robust security measures. Current scanner, while integral, often miss nuanced vulnerabilities or overwhelms users with false positives. This literature review dives into the nature of my web application security scanner, providing insights into how important and pertinent it is in the context of today's cybersecurity environment. Due to the explosive growth of web applications, fresh avenues for cyberattacks have arisen and web applications are now vulnerable to a variety of threats. These threats include SQL injections, cross-site scripting (XSS), and security misconfigurations. (Saeedi, Ghazisaeedi and Rezayi, 2021).

**2.2 Challenges in securing modern web applications**

Complex, cross-platform web applications offer a large attack surface that is difficult to defend against manually, for comprehensive security analysis to find vulnerabilities throughout this complicated environment, automated vulnerability scanners such as the one proposed by (Ladan, 2010) are essential. These scanners must contend with not only traditional security risks but also with the novel challenges introduced by web services – challenges that necessitate a reconsideration of traditional security models and controls. Due to the intricate architecture of web applications, automated vulnerability scanners are indispensable. Proposed by scholars like (Ladan,2010), these tools are crucial for navigating the vast array of traditional and emerging security risks including WSDL and UDDI attacks to XML schema tampering. These challenges are exacerbated within agile development environments that Favour rapid updates and iterations, demanding integration of security measures directly into development pipelines for effective risk management (EL Moussaid & Toumanari, 2014). Traditional end-stage security testing is inadequate within the agile development paradigm, which prioritizes frequent updates and iterations. In order to improve development process agility and lower late-stage security risks, security scanners must be integrated into development pipelines. This enables prompt vulnerability identification and mitigation (EI Moussaid & Toumanari, 2014).

Modern web applications have complex architecture, bringing together multiple sets of technologies, frameworks, and third-party services. This complexity is not a byproduct of technological advancement – it has risen as an architectural necessity to be able to meet multifarious and dynamic demand in web, mobile, and cloud platform environments. These days the modern web application will often include the wide range of technology that compromises everything from front-end frameworks like React or Angular to back-end systems like Node.js or python Django which introduces multiple vulnerabilities. These front-end frameworks are developed to support the building of a dynamic and responsive UI, while the business logic and data management, among others, are handled by a backend system. Each technology layer, either handling the UI or backend data processes, must be rigor secure to prevent breaches that could stem from inadequate data validation or flawed authentication processes(Aruna,2016) There is a possibility of having security gaps from the interaction of these different systems, especially if integration is not done with very strict rigorous security protocols(Aruna,2016) points out that web services often face unique challenges dues to their distributed nature and cross-platform accessibility, complicating the maintenance of data confidentiality and integrity The integration of these disparate technologies and platforms, as seen in modern applications, often leads to gaps in security coverage, especially when external systems interact through APIs or other web services.( Kearney., 2005) elaborates on these interactions, highlighting the necessity for robust message-level security measures such as encryption and digital signatures to safeguard data exchanges across insecure networks, particularly in scenarios involving collaborative business processes that rely on web services. For example, front end applications interacting with back-end services through APIS based on. Json, have to be designed secure, ensuring effective data validation along with processes of authentication and authorization. Any oversight in these areas could lead to vulnerabilities like SQL injection, Cross-site scripting (XXS), or cross-site request forgery, each one taking advantage of a different thing wrong with how the application processes the user data or requests (Awuor,2023). Each layer or component of it carries potential vulnerabilities-those inherent in its security features and the way they have been stitched with the safeguarding of the overall application. For example, a web application may rely on notup to datee third party plugins or libraries, cross-checked against security vulnerabilities', adding therefore adding risks difficult to identify. Every part of the architecture of a web application is built with its inherent security measures, which may not be par with those used in the other arts being utilized in the application. Furthermore, the complexity of securing these applications is not just about managing external interactions but also about ensuring that all internal components are protected against emerging threats.(.Curphey and Arawo., 2006) discuss the importance of employing comprehensive web application security assessment tools that can detect vulnerabilities across a web applications entire architecture. These tools are essential for uncovering hidden threats that could bypass traditional security measures. (Zhang & Xu, 2005) delve deeper into the security of instance-level interactions within webservices, which is particularly critical in distributed and heterogenous computing environments where different services and instances interact. Ensuring the security of these interactions is crucial for maintaining the overall integrity of the application.

(Ojamaa and Duuna, 2012) explore the security challenges of Node.js, particularly vulnerabilities in its event-driven architecture with non-blocking input/output. they outline best practices for secure application development in Node.js environments, emphasizing the importance of validating user inputs and securely managing dependencies. react application may apply client-side rendering and therefore, be exposed to XXS if its not sanitized underscoring the need for comprehensive security strategies that encompass every interaction within the applications system. A Node.js application, in this case would be exposed to a remote code execution attack whenever its inputs from the outside environment are not well validated.(Staicu et al., 2018 ) highlight the susceptibility of Node.js applications to injections attacks, given the platforms extensive use of modules that may execute untrusted code. Often, the integration of these technologies requires bridging in a way that overlapping with existing measures for the different components does not occur. The security strategies implemented had to cover the data in transit between the client and server, the data at rest, and the data being processed ensuring uniform application of encryption, safely handling tokens, and best practices of access controls at every layer is like a tightrope walk and requires constant vigil, together with regular updating.

(van Ginkel et al., 2019) introduce NODESENTRY, a security architecture for securely integrating third-party libraries into the server-side JavaScript ecosystem. This system employs strict web hardening techniques and access control policies, crucial for incorporating third-party libraries that may not adhere to strengthen security practices. Third-party services for things like payment gateways, data analytics, or social media integration just make the picture more complex. Often, such services require overly broad permissions, giving to them, in order to able to interact with the main application, thus creating potential attack surface if that third-party service is compromised. Most modern web applications highly leverage third-party plugins and libraries to fats-track development and introduce advanced functionalities without necessarily reinventing the wheel. This, however, may introduce a lot of risks more so if such third-party elements are not updated or thoroughly vetted, ensuring there are no points of vulnerability any case, this plugin or library increases the attack surface of the application even further, ad its vulnerabilities may compromise the whole application. For examples npm package has some critical vulnerabilities, such that thousands of dependent applications are at great risks if exploited. The challenge lies not only in selecting reliable and secure third-party components but sold maintaining them. It is, therefore, very important that the assessment is made continuously for those new vulnerabilities in order to understand the security practices and update them upon the release of patches.

Some of the problems that are associated with web-based applications, especially in cross-platform implementation, are the strong security measures it requires. For instance, a complex nature of security configuration may have to be put into place so that there can be proper management and monitoring across platforms(Khalid et., 2021).The highest level of vulnerability happens where there is data flow from one ecosystem to another and the caching of data on different devices, which may , therefore require separate security mechanisms put in place such as my proposed web application scanner. For example, an application that stores some data in the cloud might cache some of that data onto mobile devices. Will need separate security measures to ensure that data does not get leaked from cache and, more fatally, from the hardware. Every single architectural decision to the development of the web application is about to take has potential security consequences. Decisions, especially those referring to state management, session handling, and inter-component communication, must be taken with a mindset of security first. For example, state management strategies using sensitive information to unauthorized access in cases where such information is not encrypted, or the mechanism of storage is not tamper-proof. In short, todays moder web application demands a very comprehensive security approach across every layer or component of the application.

It is also an integration process for many security practices, tools and protocols at each stage of software development, with constant monitoring, updating, and assessing the security posture of the application against emerging threats. Therefore, the issues call for advanced solutions in the security that can change with the dynamic landscape of web application security to assure that applications are still strong even in the presence of old and newly discovered cyber threats. Defending these applications requires an overlapping security strategy to ensure that the effectiveness of the layers is as maximal as possible across all platforms that deploy the application. This may entail putting in place security tools that are specifically meant for surveillance and defense of threats against all parts of the application ecosystems in real-time. However, the complexity level of these tools and the expertise level required to manage them could become significant for organization's.

Lastly, the continuous evolution of web application framework necessitates an ongoing assessment of their security features (R.Oliveira et al, 2020) propose a benchmarking methodology that evaluates and compares the security capabilities of various web services Frameworks. This approach helps in identifying potential weaknesses and establishing trustworthiness, ensuring that security measures are both effective and up to data. By synthesing these scholarly insights, it becomes event that defending modern web applications require a layered and comprehensive security strategy .This strategy must encompass everything from securing data transmissions and internal processes to evaluating third-party components and ensuring the secure integration of various technologies. Advanced security solutions that can dynamically adapt to the changing threat landscape, such as the web application scanners discussed, are therefore indispensable. These tools and methodologies not only enhance the detection of vulnerabilities but also ensure that security measures evolve in tandem with technological advancements.

The web application scanner I propose is specifically designed to address these multifaceted challenges. It performs comprehensive scanning of both front-end and back-end components, identifying vulnerabilities from cross-site scripting to SQL injection and click jacking, thus providing a holistic solution. The scanner is quipped with advanced algorithms to detect and alert on inconsistencies or outdated components in real-time, facilitating prompt updates or patches. Moreover, it ensures rigorous security compliance by verifying that all data transactions comply with the latest encryption standards and that authentication are robust against security breaches. By integrating this scanner into the development and maintenance phases of a web application lifecycle, organizations can ensure that their application are not only compliant with current security standards but also prepared to adapt to emerging threars.is proactive approach significantly reduces the risk of data breaches and enhances the overall web application, ensuring that it remains robust across all operating environments.

**2.1 The evolution and effectiveness of security testing**

In exploring the evolving landscape of web application security, (yuan yuan pan, 2019) sheds light on the emerging paradigm of interactive application security testing(IAST) which synthesizes the strengths of static application security and dynamic applications security testing to offer a robust framework for identifying vulnerabilities in web applications. While SAST provides early detection of potential security issues in source code, DAST contributes insights into the applications behavior during runtime. Despite its comprehensive approach, IAST faces challenges such as integration complexity and resource-intensive demands, particularly for developers lacking in-depth security knowledge. This research shows the need for the development of IAST solutions that are not only efficient ad user friendly but also align with agile development methodologies, thereby underscoring a significant area of focus in my research endeavors. The recent study by (Setiawan et.al.2020) extends the discussion on the effectiveness of interactive application security testing by implementing it within the context of governmental web applications. The research emphasizes the utility of IAST in discerning vulnerabilities that may remain undetected by either SAST or DAST alone. The study reveals that IAST, implemented using tools such as ZAP and API, identified 94 vulnerabilities conforming to the OWASP top ten security risks. This contrasted with 81 vulnerabilities detected by SASRS and only 13 by DAST, thereby demonstrating IASTS enhanced capability to pinpoint security. These statistics not only highlight IASTS superior detection capabilities but also reinforce the necessity for an integrated approach in the face web services unique security challenges (Ladan, 2010). (Patil , 2023) provided a detailed review of the OWASP top 10 web application security risks, highlighting the prevalence of risks such as broken access control, cryptographic failures and injection vulnerabilities. These risks underscore the importance of continuously updating security measures and conducting thorough security audits as part of the regular development process. The persistent emergence of new security challenges necessitates the development of sophisticated tools that can not only detect but also anticipate ad adapt to the changing threat landscape. The proposed scanner aims to address this research gap in creating solutions that efficiently integrate into development workflows while providing comprehensive security coverage. Therefore, the scanner will be designed to combat a wide array of vulnerabilities, from injection attacks and broken authentication.

2.3 Cloud based security considerations

As web applications transition to cloud-based environments, the security landscape evolves which has presented both opportunities and challenges. Cloud computing, lauded for its quick deployment and cost-effective solutions, enables organization to scale resources dynamically and reduce the in-house staffing burden(Corrado & Moulaison,2010).However this migration is not without drawbacks and challenges as security still remains a paramount concern, with issues such as secure authentication frameworks and the safe guarding of data transmission at the forefront (Kim, 2009). The lack of transparency from cloud providers about their security infrastructure and policies contributes to the complexity of trust. (Corrado & Moulaison, 2010). The constraints on customization and integration with in-house IT operations underscore the need for adaptable security solutions that can operate with the clouds flexible yet controlled environment. ((Corrado & Moulaison, 2010). The scanner presented in this study aims to close the gap created by these cloud-based service difficulties. It strives to provide a security solution that not only matches with the agility of cloud services, but also strengthens the security posture while maintaining the inherent benefits of cloud computing. The scanner ensures customized security monitoring, SLA adherence, and performance integrity, facilitating a blend of cloud and on-premises services.it aims to bolster trust in cloud security, tackling issues noted by (Corrado and Moulaison, 201). Itss integration into existing infrastructure intends to surpass traditional security measures, proving vital in defending web applications against evolving threats. This automated scanner is a crucial development in web application security aidingorganizationss and developers in vulnerability detection.

If we take an example of cloud computing, improved cost-effectiveness of the market trend would mean a real huge impact in organizational budgeting and deployment of resources. Cloud services present a opportunity through a “pay as you go” subscription model that pushes an organization from large capital expenses normally associated with old physical IT infrastructure to smaller operational expenses. This has been identified by (Azodolmolky, Wieder, & Yahyapour, 2013) , bringing about optimization of financial resources through the relocation of those that would have been used for large capital costs towards operational costs. Beyond budgeting, it is basically another phenomenon defining new approaches to how IT infrastructure is maintained and updated in the cloud. In doing so, cloud services bypass traditional resourceful processes of time and money, leaving organizations off the hook from the continuous treadmill of hardware upgrades and system upkeep.

However, these cloud computing benefits are accompanied by new challenges in cybersecurity. Each change in the cloud environment brings another security problem in what, by all accounts, is an insecure landscape, considering that the security models were alien to most. It meant that, for organizations to enjoy benefits from the efficiency that cloud computing brings along, they had to grapple with securing their data and applications in such distributed environments, for which they needed advanced cybersecurity solutions customized to the peculiarities of cloud computing. (Jensen, Schwenk, Gruschka, & Lo Iacono, 2009) builds upon this by underscoring that the level of security provided by traditional models is simply unacceptable within the cloud environment and thus requires very fresh, innovative approaches to such critical components as data confidentiality and integrity, along with trust. Cloud computing brings the decentralization of data storage and processing; the protection of this now therefore becomes a complex issue.(wyld.,2009) further observes the growing use of cloud computing in the public sector, accompanied by an increase insecurity risks and benefits for government organizations. This, therefore, is an indication not only of the importance of very meticulous security within government-based cloud resources but also of another necessity for tailor-made security solutions. The proposed web application scanner intend to cover up these gaps, in which it will provide a security solution for dynamism of services in the cloud, involving traditional threats in cybersecurity and peculiarities only for services based on clouds. With the unique challenges to cybersecurity in cloud computing, measures like full web application scanners are essential. They secure against the traditional and specific threats to clouds related to data protection and threats that are a management of the authentication framework. This means that the organization has to assure itself that they are not compromising their security at the same time as they benefit simultaneously from the cost of saving and efficiency(khan et.,al 2012). Scanners are set to support the whole way, from data protection during transmission to the secure management of authentication frameworks for complexities of cloud-based web applications. For any organization using cloud computing ensuring a strong security is highly recommended to deploy as it integrates with the cloud environment as the base against next-generation cyber threats.

2.4 Aims and objectives

This project's main objective is to provide a web application vulnerability scanner that can quickly and accurately identify typical security risks like clickjacking, SQL injection, and cross-site scripting (XXS). The tool's design prioritizes ease of use, making it possible for users, regardless of experience level, to browse and utilize it efficiently to enhance the security of their web applications.

Objective 1: Design the user interface with an intuitive design that simplifies the complexity of operations such as scanning. This design will make the scanner accessible even to users without extensive cybersecurity knowledge. Additionally, interactive elements will be incorporated within the interface to guide users through the scanning process, providing useful instructions and support at every step.

Objective 2: Detailed Scanning Engine - To precisely identify SQL injections, XXS, and clickjacking, a strong scanning engine will be created. Sophisticated algorithms will be incorporated to reduce false positives and offer thorough diagnostics, assisting users in comprehending and remediating issues.

Objective 3 :Accessibility and Usability - To guarantee that all users can efficiently use the scanner, it will be made available and useful on a variety of platforms and devices. The development of the web application vulnerability scanner will be organized into sprint cycles, each aimed at achieving specific goals that collectively enhance the tools capabilities

Objective 4: Real-time Support and input - Throughout scans, consistent and current input that highlights developments and outcomes will be given. A support function will provide thorough resources and immediate assistance for fixing identified issues.

Objective 5: Comprehensive User Testing - To obtain input on the scanner's usability from a wide spectrum of user demands, including both technical and non-technical users, extensive user testing will be carried out. The tool will be improved to improve the user experience and streamline the UI in response to this feedback, making the scanner even more user-friendly.

If these objectives are accomplished, a web application vulnerability scanner that excels in user-friendliness and meets strict technical requirements for security scanning will be produced. With the help of this scanner, web security scanning will become a more approachable and crucial aspect of ensuring web application security for a wide range of users. By accomplishing these objectives, the project hopes to redefine the way security products communicate with consumers and open up advanced security scanning to a far wider audience.3.1 functional requirements

**3 Requirement analysis**

The effectiveness of the web application vulnerability scanner is anchored in its capability to meet diverse cybersecurity needs. It must align with the varied requirements of its user base, providing an intuitive and easy-to-use interface while maintaining a thorough level of security analysis. This tool is envisioned as a digital guardian, offering reassurance to both business and individuals by identifying potential security flaws with precision and clarity.

In this study, we will outline the systems' requirements, divided into functional and non-functional categories. The functional requirements will specify the key actions the scanner must perform, such as accurately detecting SQL injections, cross-site scripting (XXS), and clickjacking vulnerabilities. These requirements form the foundation of the tool, detailing the essential tasks it must carry out to be considered effective. These capabilities as highlighted by (Haibo Chen et al.2020), integrate information collection with vulnerability detection to enhance efficacy.

Conversely, the non-functional requirements will focus on the system's performance criteria, highlighting attributes such as speed, reliability, availability, and user experience. These factors, though not directly linked to the scanners core functions, are crucial in ensuring that the tool is not only capable but also efficient, reliable, and adaptable to the ever-changing landscape of web security threats (Fong & Okun, 2007).note that effective web applications scanners must offer robust functionalities that are easily testable and verifiable to maintain their r relevance in a rapidly evolving cyber threat landscape. These functionalize ensures that the new scanner can keep pace with the rapidly evolving landscape of cyber threats and provide thorough security analyses. Below is a detailed breakdown of the specific functions required for the scanner to achieve its intended outcomes, which include maintaining a user-friendly interface alongside rigorous scanning capabilities.

**3.1 functional requirements**

F1.1: accept a uniform resource locator (URL)from users as input to initiate the scanning process.

F1.2: validate and process the URL to ensure it conforms to the URL standards before beginning a security scan.

F2.1: conduct comprehensive scans to detect common vulnerabilities such as SQL injections, cross-site –scripting (XXS), and clickjacking. Research by (Haibo Chen et al ., 2021) emphasizes that a well-rounded vulnerability scanner should have capabilities that span across a wide variety of vulnerabilities.

F2.2: Leverage continuously updated databases of known vulnerabilities to enhance the accuracy and effectiveness of the scanning process.

F3.1: Analyze the results of the scan to accurately distinguish between actual vulnerabilities and false positive.

F3.2 Compile and provide a detailed report that outlines identified vulnerabilities, their severity, and recommended mitigation strategies.

F4.1 Design a clear and navigable user interface that is accessible to both technical and non-technical users.

F4.2 includes educational resources within the interface to aid users in understanding the implications of detected vulnerabilities and how to address them.

F5.1: implement a system to generate real-time alerts for users when critical vulnerabilities are detected during scans.

F5.2 : provide customizable notification setting, allowing users to tailor alerts based on their specific needs and preferences.

F6.2: Facilitates seamless integration with other tools and services commonly used in web development and maintenance workflows.

These functional requirements are designed to equip the web application vulnerability scanner with the necessary tools and capabilities to effective secure web applications against potential threats, thereby supporting business and developers in maintaining robust digital infrastructures. By fulfilling these requirements, the scanners will not only provide critical security insights but also enhance the overall security posture of web applications it is tasked to protect.

**3.2 Non-functional Requirements**

For the web application security scanner, these requirements are crucial in ensuring it not only functions effectively but also deliver consistent, reliable, and user-friendly performance. Here, we outline the essential non-functional standards that the scanner must uphold.

**Usability**

N1: The interface should be intuitive and easy to navigate for users of all skill levels. Provide clear instructions and guidance for new users to enhance learnability.

**Performance**

N2: ensure the scanner operates with optimal efficiency and minimal latency and perform scans and return results promptly to ensure a smooth user experience, avoiding disruptions in workflow.

N3: optimize the scanners code and architecture for fast processing speeds, accommodating complex scans without significant delays.

N4: The scanner must be scalable to accommodate a growing number of users and scans. The system architecture should be designed to handle increased loads smoothly, ensuring it can scale without performance degradation.

N5: uphold the highest security standards to protect user data and the integrity of the scanning process. Secure coding practices should be employes and conduct regular security audits to prevent vulnerabilities.

**Compatibility**

N6: Ensure the scanner is compatible with a variety of browsers and devices and adhere to web standards for cross-browser functionality, enabling consistent performance across different platforms.

**Documentation**

N7:Provide comprehensive documentation for the system including detailed user manuals and FAQs to support end-users and maintain an updated online knowledge base for troubleshooting and support, assisting in resolving issues efficiently.

These non-functional requirements are integral to the overall user experience and operational quality of the web application vulnerability scanner. By meeting these standards, the scanner ensures not only functionality but also long-term reliability, usability, and adaptability, making it a vital tool in the cybersecurity toolkit.

**Chapter 4 Design and methodology**

In the developing of a sophisticated web application scanner, considering specific threats such as SQL injections, cross-site scripting (XXS), and clickjacking, the kind of development methodology to be used speaks of precision, stability, and user-friendliness. As Sun Tzu said, “in the midst of chaos, there is also opportunity”. Such a concept is elicited with full development of security tools within this totally dynamic digital landscape. This project will develop a scanner that adequately identifies and mitigates the predefined categories of risks, using an approach that will optimize the tool's efficacy within the operational parameters it has been designed for (Singh et al., 2015).

**4.1 Justification of methodology choices**

Choosing the right development methodology involved weighing several critical factors, such as the project's scope, the complexity of the security features to be developed, and the importance of incorporating feedback throughout the process. Agile principles will underpin this project, emphasizing functionality, security, and a seamless user experience. Research findings by (Hema et al., 2020) on the scrum framework suggest it will enable me to review the projects direction through scheduled sprints, allowing for quick adjustments in response to new insights or change in the cyber threat landscape. The purpose of the web vulnerability scanner within the application is to provide a working and efficient tool that can find and fix specific known vulnerabilities in a very systematic way, such as SQL injection, XSS, clickjacking, etc. Therefore, since Scrum is iterative and incremental, it aligns properly with projects that must be developed and perfected specifically, rather than adapted to new threats. It ensures that the project remains specifically focused on the improvement of the scanners' ability to detect the given vulnerabilities (Subih et al., 2019). The SCRUM sprint cycles allow the flexibility of being able to follow-up with the development, testing, and enhancement of functionalities that are defined in a systematic and structured manner (Halani & Jhajharia, 2022). This goes far to enhance the chances of the scanner offering high performances in its major functionalities for the detection of SQL injections, cross site scripting, and clickjacking, which must be constantly improved to prove useful against the more complex techniques used in these types of attacks. With regular sprint reviews, it allows for stakeholder feedback, ensuring the development conforms to user requirements and expected expectation of security.

**4.1.1 Alternative methodologies considered**

During the search for an appropriate development methodology to apply in the project for the web application scanner, several approaches were taken into consideration. Some of the methodologies that came up in the process included waterfall and spiral.

**Water fall methodology**



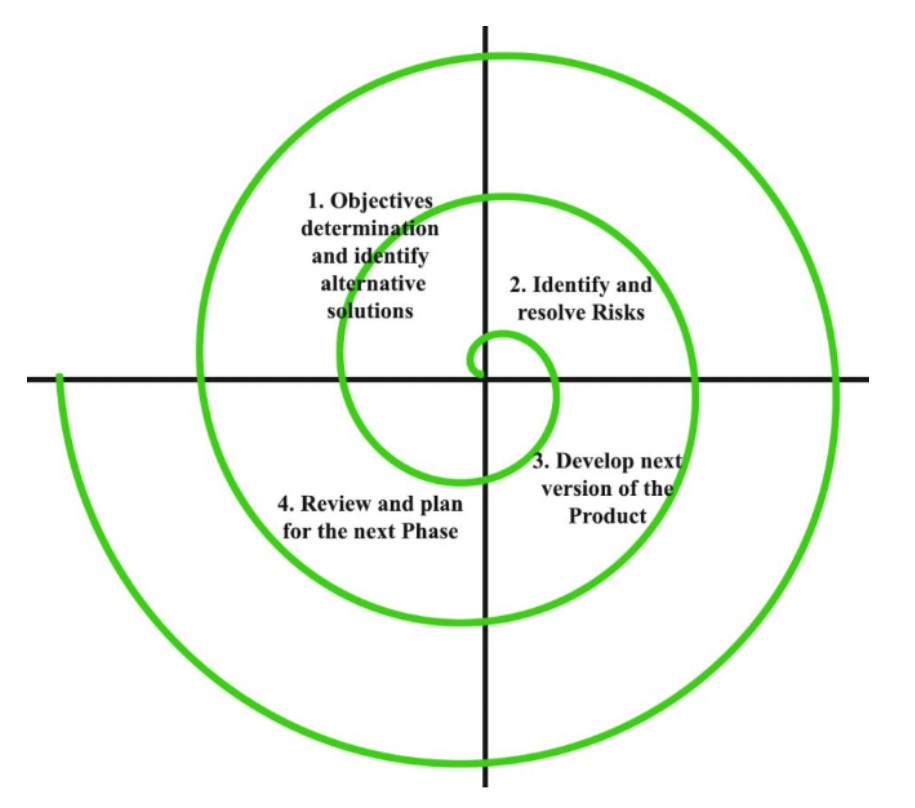
**figure 1: Waterfall model**

The waterfall mythology is sequential and inflexible. Each stage has its way to create predictability and simplicity for project management. However, the rigidity in their system is a huge challenge from projects such as mine, which require flexibility to be constantly reefing and improving certain functionalities, although not being required flexibility toward new threats.

The most salient disadvantages that come with using the waterfall methodology for this project include:

* Inflexibility: regarding revesting or modifying a given stage when required without restarting the whole process, and thus may undermine rapid iteration or enhancement, based on testing or feedback (Boehm, 1986).
* Delayed testing: The testing takes place only post the build, which adds further delays to the identification of issues and consequent implementation for necessary corrections or improvements.
* Poor response to feedback: The water fall method is not favorable for poor reactions to feedback since it does not allow continuous feedback in between the development phases, hence it very suitable for projects whose outputs benefits from iterative refinement bases on user interaction and evaluation such as this project.

**Spiral methodology**



**figure 2: Spiral methodology**

The other approach that I could have adopted is the spiral methodology, which includes repetitive refinement or rather, systematic risk assessments at every stage, considering the approach is designed to take care of complex projects (Ben-Zahia & Jaluta, 2014). The method, however, introduces complexities that are likely not in line with the requirements:

* Complex documentation and management: The spiral model needs elaborate documentation and management oversight to keep track of the risk analysis and mitigation at each cycle which diverts resources from the core development (Boehm, 1986).
* Slower progression: The focus on risk in each iteration could slow down the development cycle, as a considerable amount of time is being spent on the risk assessment rather than the development of functionalities which in my case would be my scanner's ability to scan for vulnerabilities such as SQL, XXS and click jacking.

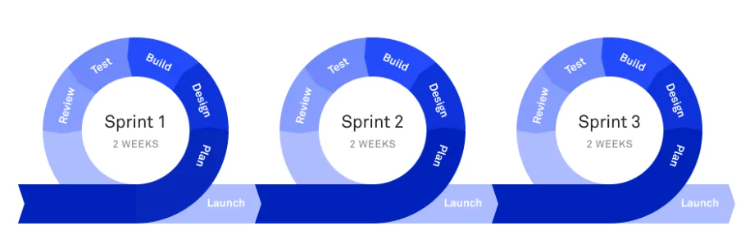
**4.1.2 Scrum as chosen methodology**

Both The waterfall and spiral methodologies will, in the long run, discourage the project with an approach that is systematic yet flexible to develop a predefined set of functionalities without the requirement to adapt to emerging threats. The project methodology in this kind of environment should therefore facilitate quick turns and revisions within a set framework of security features. In this case, the Scrum methadology is best suited as it enables a flexible yet focused approach. The iterative nature of Scrum brings significant benefits to large projects, allowing for continuous improvements of the scanner's capabilities within the defined scope of SQL injection, XSS, and Clickjacking, free from the constraints imposed by the linear and risk-heavy approaches of the Waterfall and Spiral model. This will ensure that every sprint directly results in performance improvements of the scanner in these areas, backed by user feedback and rigorous testing. The structured, agile methodology ensures the scanner is effective against well-known threats and fulfills the project's aim and objectives, which is to provide a high-quality, reliable security tool. The Scrum framework is continuous since there are continuous feedback loops with the view of ensuring the product performs to the best standards to satisfy the User. In conclusion the adoption of Scrum, as opposed to other methodologies, has ensured a deliberate focus on developing a specialized tool capable of being adapted to tailored security needs, without the necessity to adapt to new, emerging threats. This decision ensures that the development process is effectively conducted in alignment with the project's goals.

This is also supported by research that explains the agility of the Scrum framework in quick iterations and being adaptable, thus enabling it to keep up in continuous development and effective incorporation of feedback from users (Azanha et al., 2017)..Further, Scrum has also proven to improve team dynamics and project management outcomes significantly, making it a huge advantage in projects that need a strong responsive development process (Permana, 2015).This body of research undergirds the strategic choice to use Scrum in the development of a security focused software tool.

**4.2 SCRUM Application**

In this Section I will be adopting the Scrum methodology not because of its capacity to support me to address newly emerging threats, but to continually refine and improve the detection capability within the defied operational scope for the development of my web application security scanner, targeting specific, well- understood vulnerabilities such as SQL injection, XXS, and clickjacking. This adaption of Scrum tailors it precisely to the projects needs for reliability and user-centric design.

**figure 3: SCRUM Development Methodology**

The development of the web application vulnerability scanner will be organized into sprint cycles, each aimed at achieving specific goals that collectively enhance the tools capabilities. Here’s a look at the proposed sprint plan:

**4.2.1 Sprint 1: initial Scanner Setup and basic scanning Engine**

Duration: 2 weeks

Purpose: Define a functional base for the scanner that can detect primary vulnerabilities

Tasks:

* Pull together requirements for basic scanning features.
* Design a simple and clean user interface for URL input and results display.
* Develop the initial scanning engine using best practices in security software.
* Provide first sets of test cases for SQL injections, XXS, clickjacking, and others.

Testing plan:

* Automated unit tests: Develop automated unit tests to ensure each module functions correctly
* Integration Tests: Initiate integration tests to check the compatibility of all scanner components

Deliverables:

* A functional scanning engine capable of basic vulnerability scanning.
* A user interface prototype for inputting scans and viewing results.

Contribution to Project objectives:

* Provides the core technology upon which additional features will be built.
* Establishes the usability benchmark for further interface improvements.
* Review and feedback: organize a review session with a small group of potential users to evaluate the scanning engines performance and interface usability

**4.2.3 Sprint 2: The Human-Friendly, Advanced Detection**

Duration: 3 weeks

Objective: Enhance the scanners capabilities to detect complex vulnerabilities with high accuracy.

Tasks:

* Research and integrate more sophisticated analysis algorithms
* Optmize the backend to support new algorithms
* Update automated tests to include advanced vulnerability scenarios.
* Collect and integrate initial user feedback from sprint 1.

Testing plan:

* Performance testing: Tests the enhanced scanners performance under various load conditions.
* Security testing: Conduct through security assessments to ensure effective detection algorithm's

Deliverables:

* An updated scanning engine with advanced detection capabilities
* Performance and security test reports

**4.2.4 Sprint 3: UI/UX Refinement**

Duration: 4 weeks

**Objectives**: redesign the user interface for better aesthetic appeal and user interaction.

Tasks:

* Overhaul the UI/UC to improve usability and navigation.
* Develop interactive elements that guide users through the scanning process.
* Conduct detailed user experience tests.
* Implement feedback to refine the interface.

Testing plan:

* User Acceptance testing (UAT): Engage test users to evaluate the new interface and gather qualitative feedback.
* Usability Testing: Assess ease of navigation and clarify of scanning process instructions.

Deliverables:

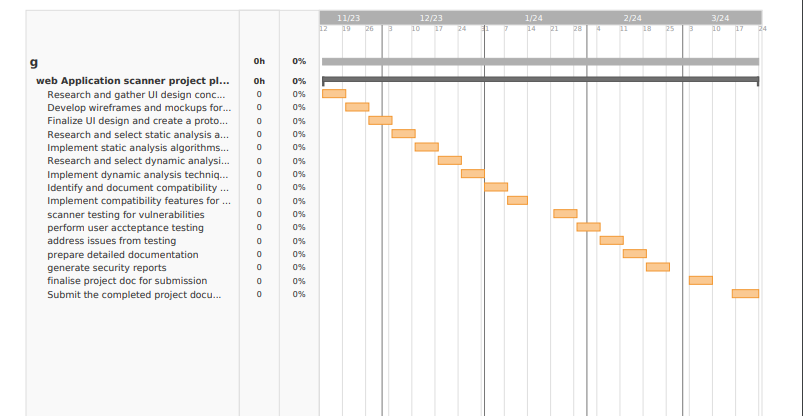
* A completely redesigned user interface.
* Comprehensive UX test reports and feedback implementation details

This structured approach ensures that each feature is not only developed but also thoroughly tested for functionality, performance, security, and user satisfaction. The final product meets both technical and user-centric goals, establishing the scanner as reliable and effective tool in the cybersecurity landscape. This integration of development testing ensures the scanner is robust, user-friendly, and capable of effectively protecting against SQL injections, XSS, and clickjacking.

**4.3 Project management**

**4.3.1 project Time scales**

The Gantt chart (see Figure 6) shows the time scale for the development of this project in relational to the functional requirements that were listed in the previous chapters. Each one of these requirements are allocated a time frame where they will be accomplished.

  
**figure 4: Gantt Chart (note: redo to match functional requirements)**

**4.3.2 Risk analysis**

The table below highlights the potential risks that could be encountered during and after the development phases of my Artefact. A thorough risk analysis was conducted as part of the creation of my web application vulnerability scanner to address any issues that might compromise the project’s success. This study which is presented in (see table 1) and ( see table 2) evaluates the prospective effects of those risks and suggests specific preventive actions. Each danger is evaluated as high, medium, or low depending on how serious it is and how often it is to occur. This systematic approach allows for targeted mitigation and effective use of available resources. Among the notable concerns are the potential for erroneous vulnerability identification, which might compromise the scanners dependability, and the chance of system overload at times of high usage, which could potentially cause performance issues. To keep the scanner effective against emerging cybersecurity threats, thorough testing procedures and frequent updates to the vulnerability database are planned. Furthermore, strategies for cloud scalability and application code optimization are suggested to avoid system overloads. Every risk that has been identified is connected to certain project requirements, guaranteeing that preventative actions can be taken and incorporated into the project's workflow. In addition, a risk matrix helps the project management to be quick and make well-informed decisions by visualizing the risk levels by comparing the possible impact with the likelihood of occurrence. Finally, the risk tables reinforce the promise to create a dependable, effective, and user-friendly web application vulnerability scanner, protecting it against a variety of dangers, by offering a strategic framework for anticipating and resolving possible project roadblocks.

**Table 1 : Risk Analysis.**



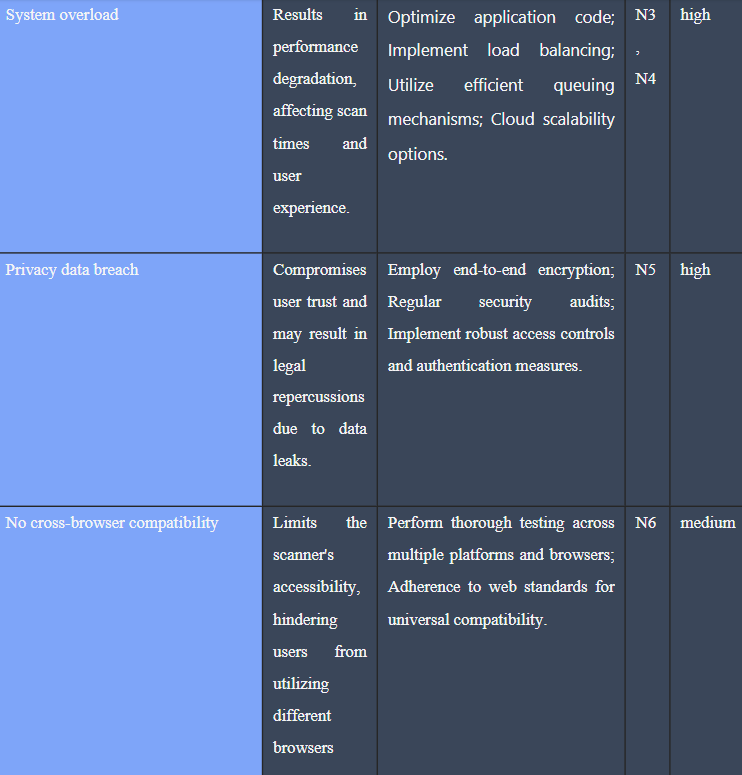
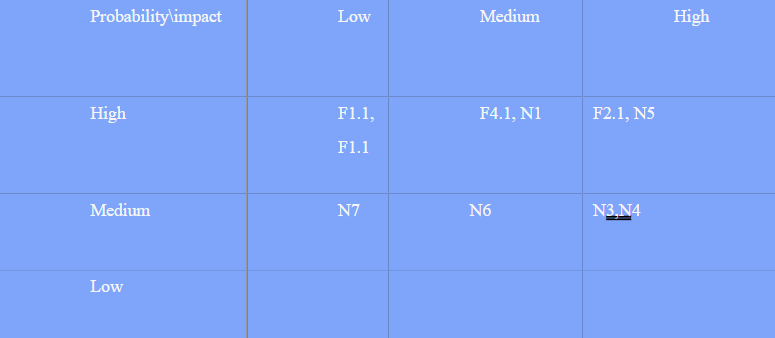




Table 2: Risk matrix

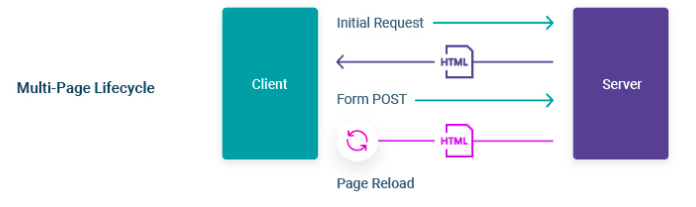


**4.4 Initial designs**

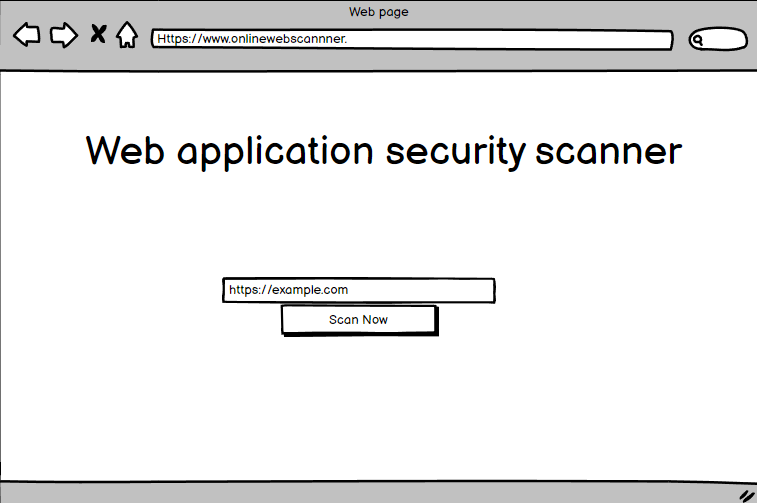
The reporting system, scanning engine, and user interface make up the three primary parts of the web application vulnerability scanners architecture. These designs are anticipated to change in response to input and modifications in the cybersecurity environment, in keeping with the Agile and Scrum approaches that have been used for this project. During the initial design phase of the web application scanner, working as a solo developer, I utilized the Scrum methodology described earlier and followed an iterative process that aligns with the given methodology. This design has a definite commitment to agility and constant evolution. It started with creating wireframes with the initial blueprints of the required elements of the user interface, which included the input field for the URL and the area displaying scan results. The use of the wireframes will be crucial in conceptualizing the flow and functionality of the application early on

**4.4.1 front-end design**

The front-end design prioritizes usability and simplicity with the goal of being simple enough for people of all skill levels to understand. To make sure the interface is accessible and responsive on a range of devices and screen sizes, it uses HTML and CSS. My web application scanner employs a multi-page application approach, which is a traditional model where each data or action is presented on a new page. Research by (Paternò, Santoro, & Spano, 2010) highlights the effectiveness of multipage applications in maintaining clear navigation and organization, especially when dealing with complex functionalities like those in the security scanner. This suits applications such as mine that need clear separation between their functional segments, allowing each part to focus on a specific task without interference or causing any confusion to the user. The scanning interface and the results display are distinct, focusing respectively on user input and results presentation. This separation helps users distinguish different stages of their interaction with the scanner. (see figure 4 below.)

Figure 4 : multi-page application mode

**4.4.2 initial front-end screen designs prototypes**

Figure 5: main screen wireframe

A user-centric and simple design concept is embodied in the web application security scanners wireframe. Research by (yee, 2002) emphasizes the importance of a user-cantered design in security systems, where clear and minimalistic interfaces support in making security decisions more intuitive and less error prone. It displays the first interface that consumers see when they try to use the scanner. This wireframe(figure,5) shows how the front-end design of the application was first conceptualized. Because it is purposefully minimalistic, the user's attention is drawn to the work at hand without being diverted by extraneous items or intricate navigation. In security-focused systems where accuracy and clarity are crucial, the designs simplicity is meant to improve user experience by reducing cognitive load and potential operational errors. The choice to designate a specific page for the scanner's functionality aligns with the multi-Page application paradigm, which calls for distinct user journeys for each stage. Therefore, the wireframe serves as a tangible representation of the project's dedication to usability, effectiveness, and simple interaction, serving as the cornerstone around which additional design improvements and functional integrations can be methodically applied.

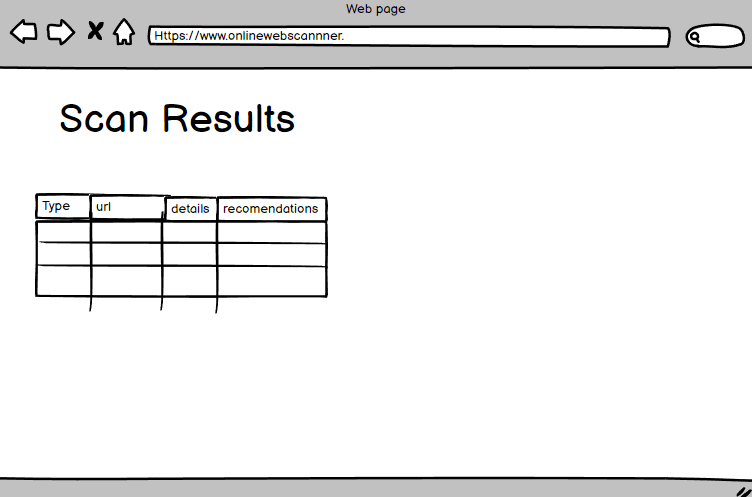


Figure 6 : scan results page wireframe

A User-friendly results interface is outlined in the web application security scanners “scan Results” wireframe. It has a simple layout with a header and an organized table with collum's labelled “Type” “URL”, “details” and “recommendations.” Users may quickly find vulnerabilities, comprehend their significance, and implement suggested mitigation measures due to this design's easy navigation and quick interpretation capabilities. This design facilitates quick scanning by users to identify and understand the security threats.( Starke and Baber ,2018) found that different user interface designs significantly affect how users engage with and process information underscoring the value of a clear and organized interface such as this one enable efficient decision making and data interpretation of security tasks.by aligning the wireframe with these established usability principles, the scanner ensures that users can effectively and efficiently perform interactions with the system, ultimately enhancing the security posture by enabling more precise and actionable insights on identified potential vulnerabilities.

***4.4.3 Personas***

In designing the web application scanner, a deep understanding of the end-users' perspectives is cruciality ensure the product development aligns with the user's needs, experiences and goals, creating detailed personas helps in visualizing the diverse spectrum of users who will benefit from the scanner. Here are two personas that exemplify the intended user base:

**Cybersecurity expert:**

“Alex is a 32-year-old Cybersecurity Analyst working with one of the top technology organizations. He brings a decades worth of experience, having watched the cybersecurity landscape and understanding the subtleties that come with securing digital assets. He has deep knowledge in security tools and best practices but always, with great caution, looks for fresh decisions, which would provide a competitive advantage.Every day he needs to perform a detailed security assessment, manage a small team of junior analysts, and design security strategies as per customer needs. Among these, alex is especially looking out for a web application scanner that can bring depth, flexibility, and integration with his present systems. He appreciates tools that can reveal system weaknesses and also advise on how one may better prepare systems from perceived threats

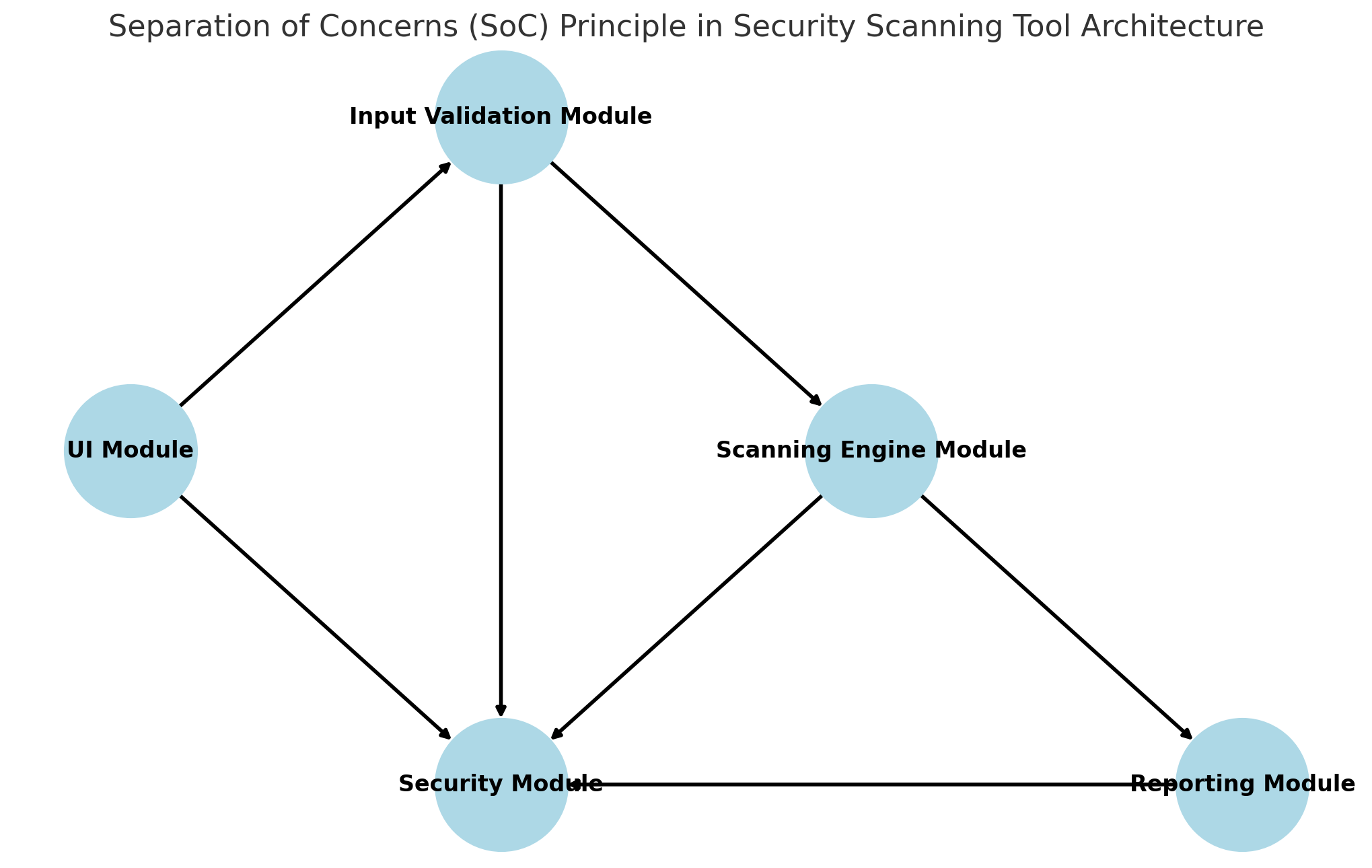
**small Business owner:**

**“45** year old Sophia for the past 5 years has been able to operate her online retail store successfully. However, she is not very knowledgeable about cybersecurity, despite her basic knowledge of its significance in our world today. She wants security solutions that are simple to implement and don't need her to become an expert in the area. The security of her client's personal information and the integrity of her website are her top priorities. Sophia uses simplicity and functionality to guide her interactions with technology; she requires a scanner that can provide clear corrective options and a straightforward explanation of the hazards. Her objective is to keep on top of the technical issues while maintaining a trustworthy and safe online experience for her clients.”

Both personas underline the aspect of a user-based design that is flexible in design so that that it caters to users like Alex, with deeper analytical needs, and at the same time, it should be simple enough for users like Sophia to derive actionable insights. This dual approach makes sure that the scanner does not alienate any end of the user spectrum but rather provide a scalable solution that is as effective in an environment filled with high-stakes as it is in a small online storefront. This is attainable through continuous feedback from personas representing real-world users, therefore changing the scanner into a tool that educates and empowers its users reflects the goal in making the digital space a safer place for everybody

**4.4.4 Back-end design**

The back-end architecture of the web application security scanner is well designed and represents the separation of concerns principle(SOC), facilitating modularity and keeping the maintainability of the codebase. It divides the scanner into different modules, working on a definite section of features. This ensures an effective iterative development of the scan er bases on SCRUM methodology.(see figure 7 below).

Figure 7 : SOC Principle diagram

**UI module:**

The user interface module wills serve as the face of the scanner. To dynamically provide HTML templates, it will take advantage of flasks\_render template method in the back end. When HTTP GET and Post requests are used by web application users to interact, this module will react to them by serving the relevant HTML page. Inputs from users , like the URL will also be captured and sent to the input validation module for processing.

**Input validation module:**

The scanner will be heavily dependant on this module, which meticulously compares every piece of user input to a predetermined set of patterns using pythons re module to search for patterns that could lead to a SQL injection ,XXS, or other injection attacks by looking through the URLs and other input data. "Designing and Implementing a Database for Thesis Data Management by Using the Python Flask Framework," by (Suraya and Sholeh ,2021) discusses flasks ability for managing data-intensive applications, which aligns with the structured, secure handling of data required in the input validation module of the scanner.

**Scanning engine module:**

The main component of the security scanner is the scanning engine module. It makes use of the Beautiful Soup library for efficient HTML response aprsing and the requests library to send HTTP requests to target URLs.

**Reporting module:**

This module takes over after the scan to arrange the information gathered. This uses the jsonifiy function in Flask to package the raw data into a JSON format that is suitable for reading. It gives the user important information that motivates them to take action by summarizing scan results and displaying the type of vulnerability found and its possible impact. To ensure that the user is fully informed about the remediation procedures to be done, the same also suggest appropriate remedial activities. “A survey on server-side approaches to securing web applications” by (Li & Xue , 2014) establish the importance of server-side security, including data handling security that can be addressed through proper serialization methods such as JSON. The result output in JSON format enhances data integrity of the scan result information and is , therefore a reliable format to present scan results.

**Security module:**

The security module employs flask session management, which includes safeguards against CSRF and other session based threats, to securely manage user sessions. This means that in order to guarantee the safety of data in transit from the start is securely interacts with the frontend.

Overall, the backend architecture is the personification of the culmination of meticulous planning and strategic implementation of Flask and pythons security-oriented features. As each module would only address a particular aspect of the applications operation, this design choice would, in fact, make Maintenace and upgrades easier and improve the scanners overall security and performance. I have iteratively designed and refined these modules with the Scrum monadology.

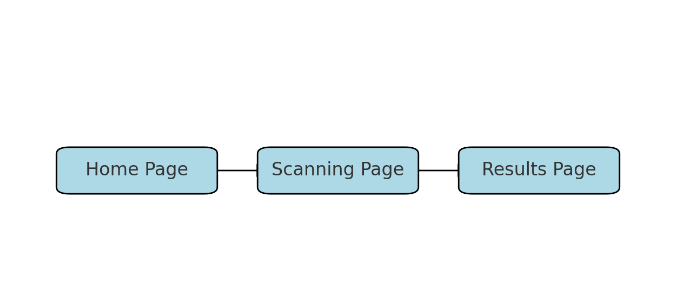
**Chapter 5**

# Implementation

**5.1 introduction to implementation**

By providing some background information between the theoretical design and actual software functionality, this chapter will go further into the practical implementation of the web application security scanner. This implies a scanner architecture that has been carefully designed to host a user-friendly interface, being implemented with a robust backend that will support the scanning process. New implementation details are Flask as a micro web framework, which eases the routing and responses flow. It also includes Python scripts for scanning logic, hence showing flexibility and capabilities to manipulate data powerfully. In the following sections in this chapter, we will dive into details of code, application structure, and implementation that make the security scanner effective and novel.

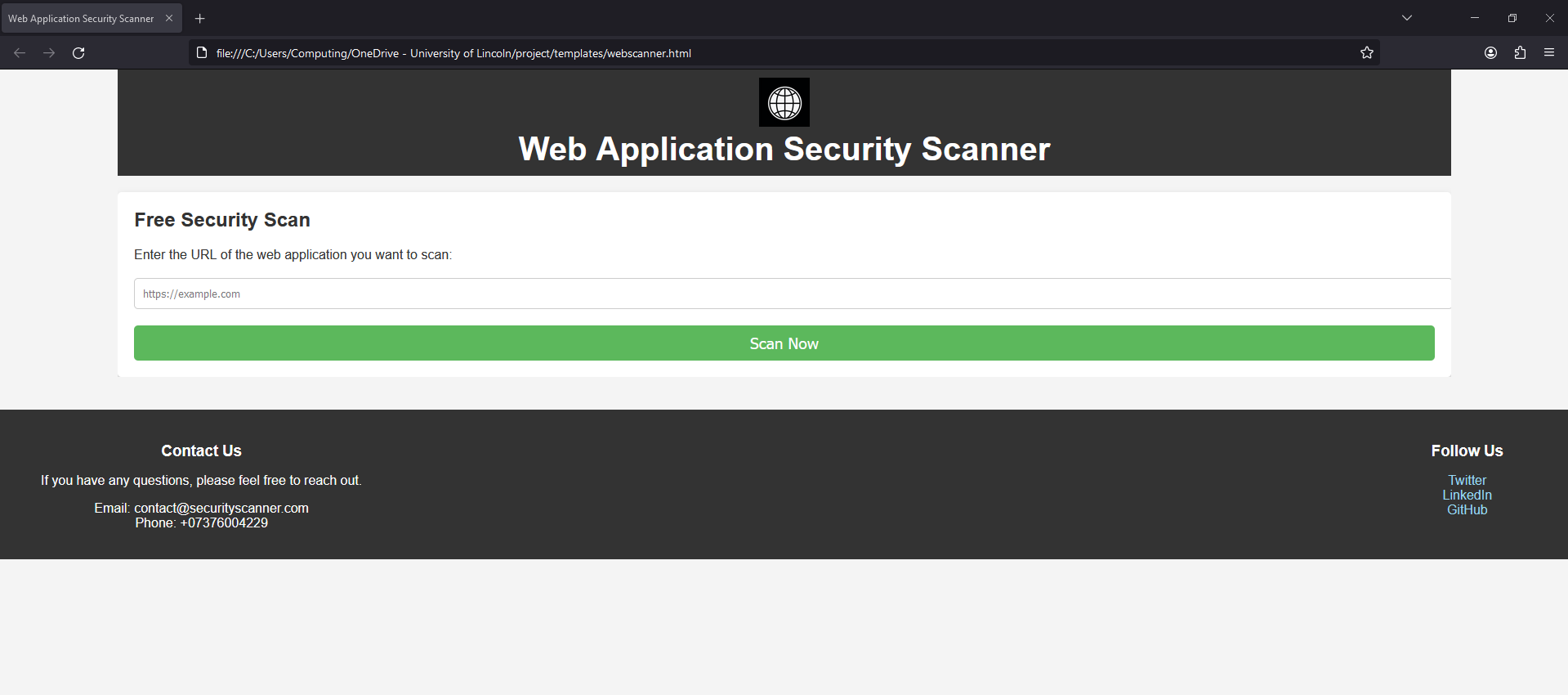
**5.2 front end implementation overview**

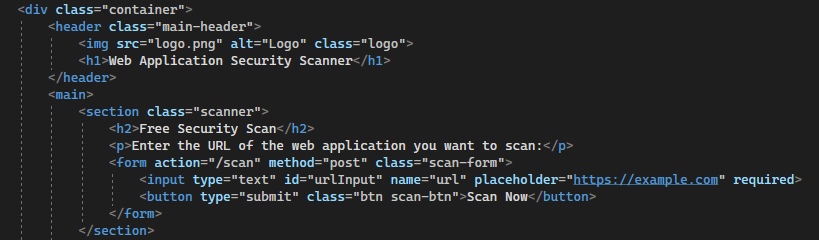
Figure 8: State diagram

Illustrates the transitions between Home page, Scanning Page, and Results Page.

**5.2.1 home state**

The Home State represents the first interaction of the user with the web application security scanner. This interface is intentionally simple and direct to make sure there are no diversions, and the user gets to know instantly what steps need to be able to proceed with the scanning. I have designed it with intention: a clear and concise copy, whitespace, familiar iconography to ensure that it's one of the easiest-to-use tools in my digital arsenal. I will now being going through how this was implemented. (See figure 10. Below)

Figure 9: Home page

Figure 10:

The frontend is carefully designed, especially the homepage, for it to reflect a user-focused experience in their implementation. The header is the very important part being the entry that frames the application. The scanner section is the interactive heartbeat of the homepage. this part is designed with the sole-focus goal that users must click a button to start scanning. The call-to-action is very clear in a brief, informing and captivating the user with the promises of a “free scan”.

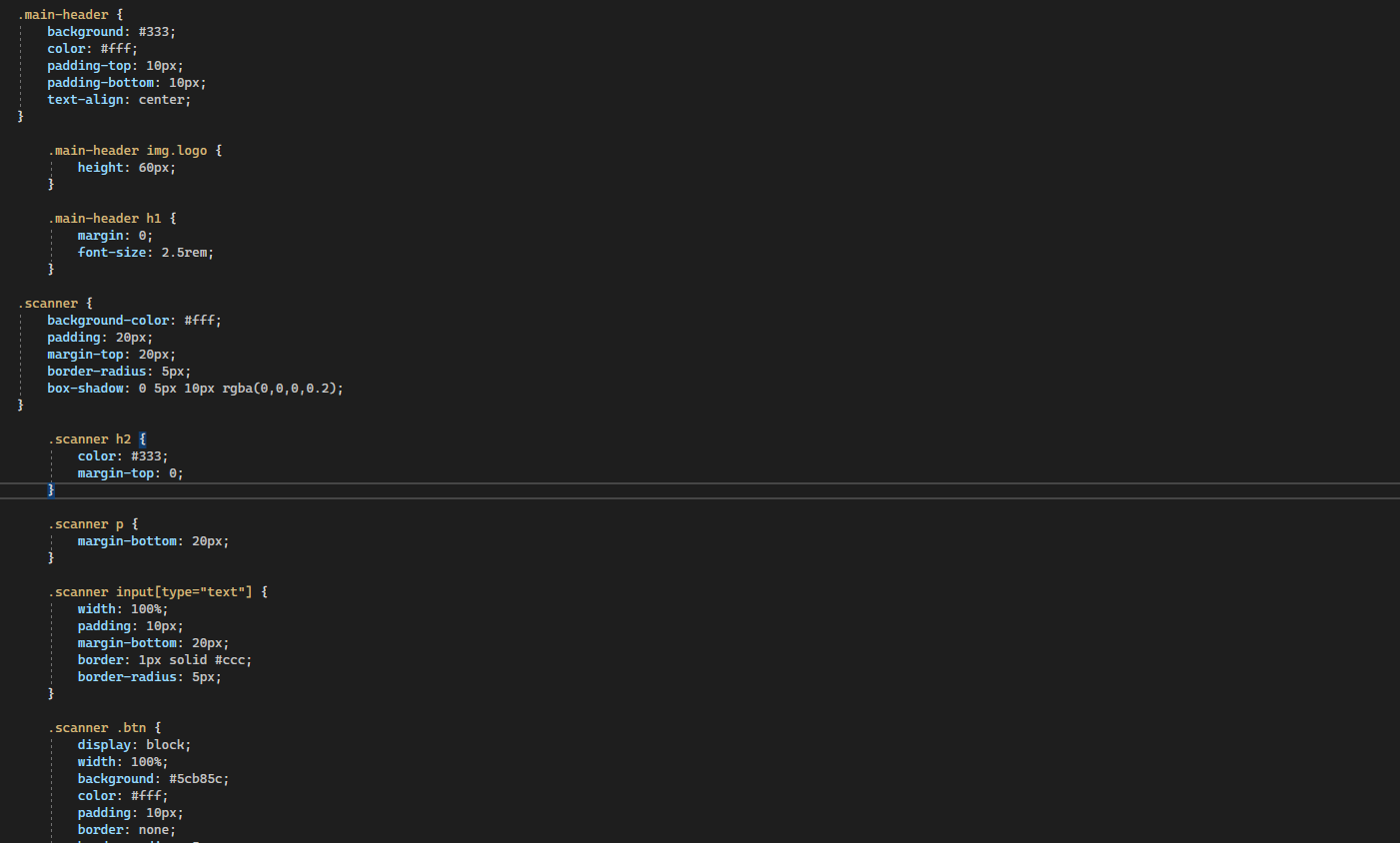
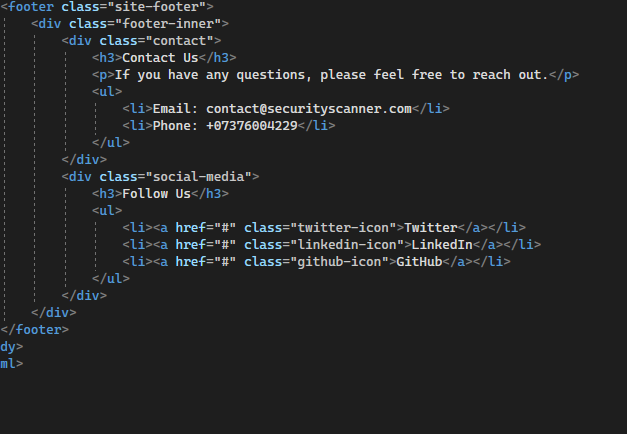


Figure: 11

The CSS for the main header is used here to create a visual distinction from the rest of the web page, which is an important aspect for the branding. The scanners styling defines a contained area for the user input, which is both inviting and visulaly seprated from the rest of the page thanks to the box-shadow effect. The green button indicates a clear to call action with its distinctive colour, which not only stands on the page but is also commonly associated with go or start.

Figure 12:

The footer is divided into contact information and social media links, providing additional resources for the users to engage with. In summary, the front-end implementation often home screen uses clear, simple outputs that reduce cognitive load on the user as they seamlessly taken from introduction to action. It demonstrates an interface designed with the depth of understanding of the user flow to assure the user is never more than a step away from carrying out the main function of the application: scanning for vulnerabilities.

**5.2.2scanning state**

The web applications scanning state is a crucial stage in ehich the system does the security analysis. Upon initiation, this mode shows the user a static, informative screen with an animation to show that the scan is running. This gives the user peace of mind knowing the system is actively operating in the background.During this stage, the frontend displays the updates that the applications backend skilfully retrieves and update data in real-time. The scanning state, which bridges the gap between the users command and the final results, highlights the process efficiency and is designed with simplicity in mind. Its a fusion of front-end design and back end featured that support delivering a smooth user experience without requiring user input while scanning. The program seamlessly moves to the results stage after the required data has been prepared and processed, concluding the users experience in a clear and accurate manner.(see figure 10) and (figure 14) below to see the implementation of this state.

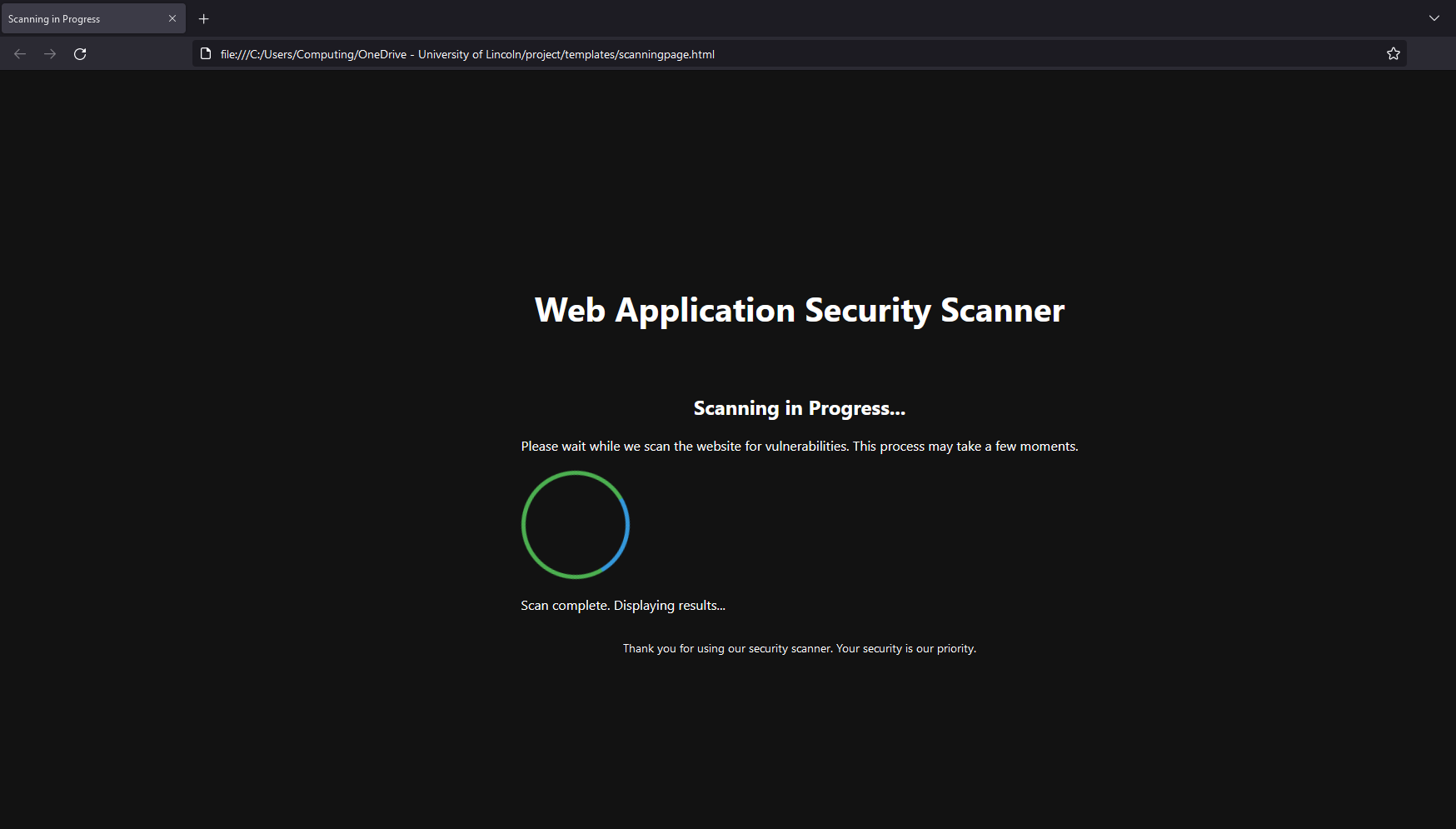
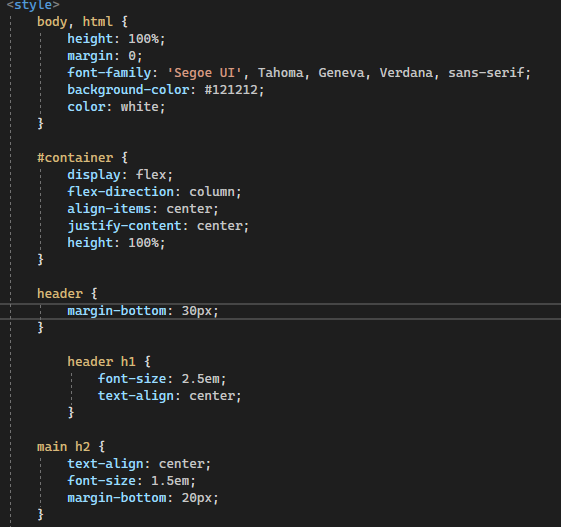


Figure 13: scanning page

Figure 14: layout and styling

The user interface semantic HTML5 elements to structure it in a logically organizer manner. A tag is surrounded by the brand identity: a tittle with the most highlighted font that serves as an immediate communicator to the user regarding the purpose of the application. The <main> section is purposed for the main interaction- to start a scan.

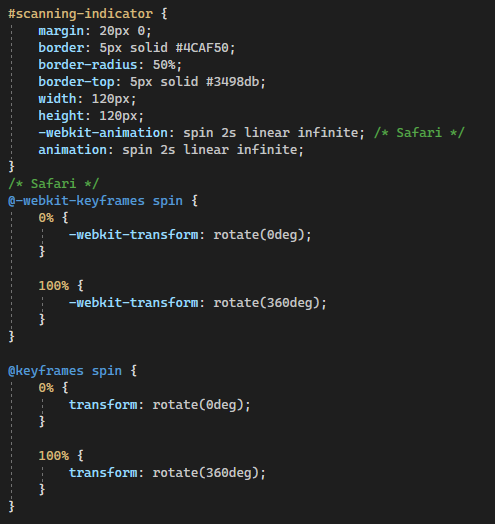
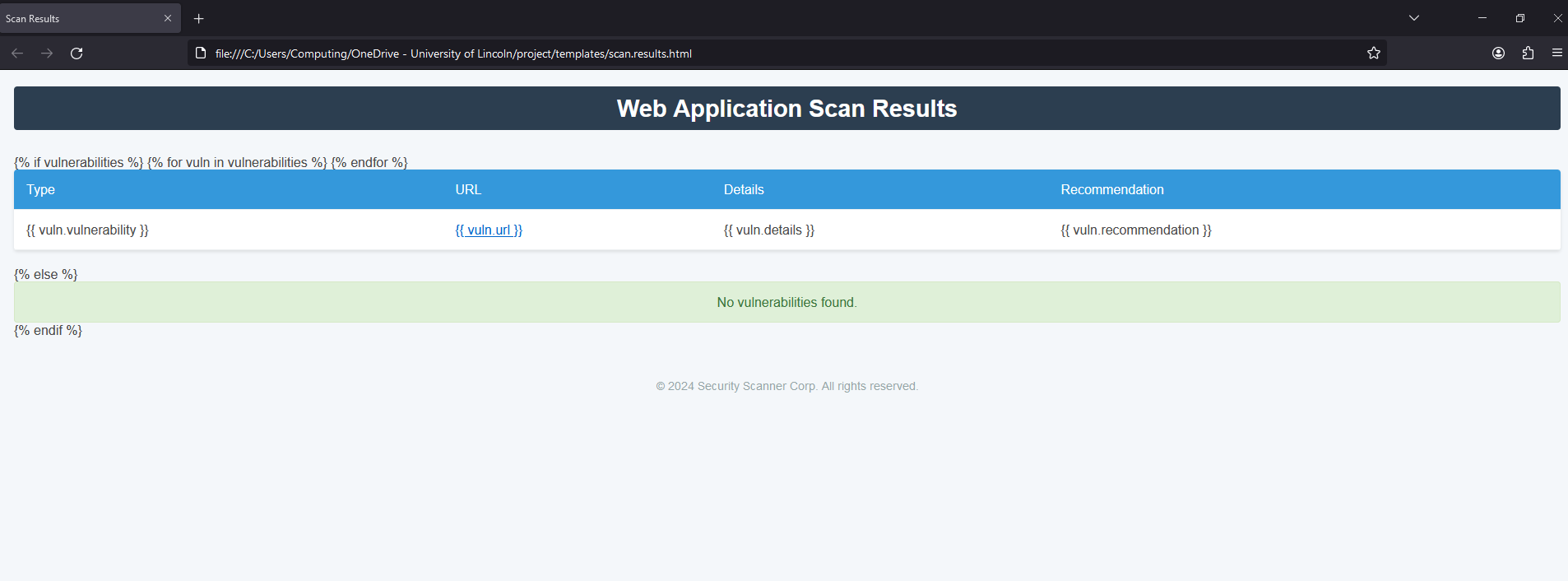


Figure 15: scanning indicator animation

In the process of scanning. The scanning indicator, with its CSS-based animation, gives a universal signifier of progress. It visually reminds users of the ongoing scanning process, and that the system is responding to. The CSS styling choices reflect the contrast and hierarchy of the palter. The colours are designed with a dark background (#121212) and light text(white) to give a very high contrast in visibility and reading.

**5.2.3 Results state**

The Results State is a very crucial part of the Web Application Security Scanner, with the purpose to present the user with the general summary of identified vulnerabilities resulting from the process of scanning. As shown in (Figure 15) below, this state is displayed to the user upon completion of scanning. This is the state where the user is presented with a summarized view of the findings in a table. The table is partitioned into columns stating the type of vulnerability, URL of which the vulnerability was found, details of the finding, and recommended remediation actions. User-oriented feature, the Results State, lists the vulnerability type in a clear and neat list that makes it easy for the user to see the kind of risk. The column URL gives the exact position of this vulnerability within the application, hence an ability to navigate for further analysis or fixing. The column "Details" contains the brief explanation of the Vulnerability, making it indispensable to comprehend the Potential Impact but without inundating the user with technical complexities. Ultimately, in the "Recommendations" column, specific advice will be provided on a case-by-case Vulnerability, giving users clear indications on how to act on the problems found. The design made effective use of familiar web conventions, ranging from table views to clear typographical hierarchies, making the results readable by users of varied levels of expertise. The interface avoided visual clutter and drew the user's eye to important information and next actions.

Figure 15: Results page

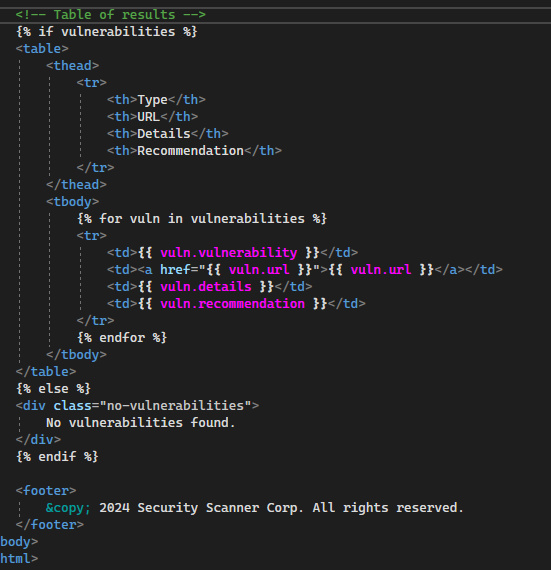


Figure 16: snippet of results rendering logic

Shown above is an HTML snippet presenting the logic for the Results State. This forms part of the template system used in the rendition of scan findings to the user. The code lies within a conditional structure that determines if any vulnerabilities have been discovered. If the scan finds any vulnerabilities, then the following loop will iterate through all the detected vulnerabilities and populate a table, which has categories like 'Type', 'URL', 'Details', and 'Recommendation'. Not really random fields; they are very carefully chosen to convey the most necessary information to the user in a tidy and effective way. The "Type" column immediately tells the category to which each of the vulnerabilities belongs, providing important information for quick triage. The URL column is particularly useful since it points out the vulnerability and gives a direct link in order to ease the user in navigation around should he want to investigate the issues further. The URL column is particularly useful since it points out the vulnerability and gives a direct link in order to ease the user in navigation around should he want to investigate the issues further. The 'Details' column is kept intentionally very short not to overwhelm the user with the information but contains enough data for understanding the meaning of the vulnerability. Finally, in the "Recommendation" column, the recommendation for mitigation of vulnerabilities completes the interface, noting the intention for the application to not only identify but also help in the repair of security issues.

**5.3 Back-end**

**Chapter 6**

# Results & Discussion

**Chapter 7**

# Conclusion

# References

Azanha, A. *et al.* (2017) “Agile project management with Scrum: A case study of a Brazilian pharmaceutical company IT project,” *International journal of managing projects in business*, 10(1), pp. 121–142. doi: 10.1108/ijmpb-06-2016-0054.

Azodolmolky, S., Wieder, P. and Yahyapour, R. (2013) “Cloud computing networking: challenges and opportunities for innovations,” *IEEE communications magazine*, 51(7), pp. 54–62. doi: 10.1109/mcom.2013.6553678.

Boehm, B. (1986) “A spiral model of software development and enhancement,” *Software engineering notes*, 11(4), pp. 14–24. doi: 10.1145/12944.12948.

Cavusoglu, H., Mishra, B. and Raghunathan, S. (2004) “The effect of internet security breach announcements on market value: Capital market reactions for breached firms and internet security developers,” *International journal of electronic commerce*, 9(1), pp. 70–104. doi: 10.1080/10864415.2004.11044320.

Chen, H. *et al.* (2020) “An automatic vulnerability scanner for web applications,” in *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. IEEE, pp. 1519–1524.

Curphey, M. and Arawo, R. (July-Aug 2006) “Web application security assessment tools,” *IEEE security & privacy*, 4(4), pp. 32–41. doi: 10.1109/msp.2006.108.

*Detecting and defeating SQL injection attacks* (no date) *Ijiee.org*. Available at: <https://www.ijiee.org/show-7-14-1.html> (Accessed: April 10, 2024).

Dwivedi, A. (2021) “Vulnerability scanning technology on web applications,” *International journal for research in applied science and engineering technology*, 9(VI), pp. 991–995. doi: 10.22214/ijraset.2021.35135.

Espadas, J. *et al.* (2013) “A tenant-based resource allocation model for scaling Software-as-a-Service applications over cloud computing infrastructures,” *Future generations computer systems: FGCS*, 29(1), pp. 273–286. doi: 10.1016/j.future.2011.10.013.

Fong, E. and Okun, V. (2007) “Web application scanners: Definitions and functions,” in *2007 40th Annual Hawaii International Conference on System Sciences (HICSS’07)*. IEEE, pp. 280b–280b.

van Ginkel, N. *et al.* (2019) “A server-side JavaScript security architecture for secure integration of third-party libraries,” *Security and communication networks*, 2019, pp. 1–21. doi: 10.1155/2019/9629034.

Halani, K. R. and Jhajharia, K. (2022) “A quantitative study of waterfall and agile methodologies with the perspective of project management,” in *Contemporary Challenges for Agile Project Management*. Hershey, PA: IGI Global, pp. 111–133.

Hema, V. *et al.* (2020) “Scrum: An effective software development agile tool,” *IOP conference series. Materials science and engineering*, 981(2), p. 022060. doi: 10.1088/1757-899x/981/2/022060.

Huang, Y.-W. *et al.* (2004) “Securing web application code by static analysis and runtime protection,” in *Proceedings of the 13th international conference on World Wide Web*. New York, NY, USA: ACM.

*Inderscience Publishers - linking academia, business and industry through research* (no date) *Inderscience.com*. Available at: <https://www.inderscience.com/offers.php?id=82865> (Accessed: April 11, 2024).

Jensen, M. *et al.* (2009) “On technical security issues in cloud computing,” in *2009 IEEE International Conference on Cloud Computing*. IEEE, pp. 109–116.

Kearney, P. (2005) “Message level security for web services,” *Information security technical report*, 10(1), pp. 41–50. doi: 10.1016/j.istr.2004.11.003.

Khan, A. U. *et al.* (2012) “Security risks and their management in cloud computing,” in *4th IEEE International Conference on Cloud Computing Technology and Science Proceedings*. IEEE, pp. 121–128.

Koch, F. *et al.* (2016) “Optimising resource costs of cloud computing for education,” *Future generations computer systems: FGCS*, 55, pp. 473–479. doi: 10.1016/j.future.2015.03.013.

Li, L. *et al.* (2016) “Cyber security awareness and its impact on employee’s behavior,” in *Lecture Notes in Business Information Processing*. Cham: Springer International Publishing, pp. 103–111.

Maimon, D. *et al.* (2017) “Re-thinking online offenders’ SKRAM: Individual traits and situational motivations as additional risk factors for predicting cyber attacks,” in *2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech)*. IEEE, pp. 232–238.

Ojamaa, A. and Düüna, K. (2012) “Assessing the security of Node.js platform,” in *2012 International Conference for Internet Technology and Secured Transactions*. IEEE, pp. 348–355.

Permana, P. A. G. (no date) *Scrum method implementation in a software development project management*, *Thesai.org*. Available at: <https://thesai.org/Downloads/Volume6No9/Paper_27-Scrum_Method_Implementation_in_a_Software_Development_Project_Management.pdf> (Accessed: April 23, 2024).

Raban, Y. and Hauptman, A. (2018) “Foresight of cyber security threat drivers and affecting technologies,” *foresight*, 20(4), pp. 353–363. doi: 10.1108/fs-02-2018-0020.

Singh, P. *et al.* (no date) *Detection of SQL injection and XSS vulnerability in web application*, *Ijeas.org*. Available at: <https://www.ijeas.org/download_data/IJEAS0203008.pdf> (Accessed: April 23, 2024).

Subih, M. A. *et al.* (no date) *Comparison of agile method and scrum method with software quality affecting factors*, *Thesai.org*. Available at: <https://thesai.org/Downloads/Volume10No5/Paper_69-Comparison_of_Agile_Method_and_Scrum_Method.pdf> (Accessed: April 23, 2024).

Wyld, D. C. (2010) “Risk in the clouds?: Security issues facing government use of cloud computing,” in *Innovations in Computing Sciences and Software Engineering*. Dordrecht: Springer Netherlands, pp. 7–12.

Zhang, D. and Xu, J. (2005) “Securing instance-level interactions in web services,” in *Proceedings Autonomous Decentralized Systems, 2005. ISADS 2005*. IEEE, pp. 443–450.

Zhao, R. and Lv, S. (2007) “Neural-network based test cases generation using genetic algorithm,” in *13th Pacific Rim International Symposium on Dependable Computing (PRDC 2007)*. IEEE, pp. 97–100.

(No date a) *Indjst.org*. Available at: <https://indjst.org/articles/an-analysis-on-efficient-resource-allocation-mechanisms-in-cloud-computing> (Accessed: April 11, 2024).

(No date b) *Wjaets.com*. Available at: <https://wjaets.com/content/review-security-challenges-web-based-systems> (Accessed: April 12, 2024).

(No date c) *Researchgate.net*. Available at: <https://www.researchgate.net/publication/4322871_Testing_and_Comparing_Web_Vulnerability_Scanning_Tools_for_SQL_Injection_and_XSS_Attacks> (Accessed: April 23, 2024).

(No date d) *Thesai.org*. Available at: <https://thesai.org/Publications/ViewPaper?Volume=10&Issue=5&Code=IJACSA&SerialNo=69> (Accessed: April 23, 2024).

(No date e) *Ssrn.com*. Available at: <https://papers.ssrn.com/sol3/Data_Integrity_Notice.cfm?abid=3610759> (Accessed: April 23, 2024).

Amarala, O., Abualhaijaa, S., Sabetzadehb,a, M. and Briand, L. (2021). *A Model-based Conceptualization of Requirements for Compliance Checking of Data Processing against GDPR | IEEE Conference Publication | IEEE Xplore*. [online] ieeexplore.ieee.org. Available at: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9582337> [Accessed 31 Dec. 2023].

Barney, N. (2023). *What Is PCI DSS (Payment Card Industry Data Security Standard)? - Definition from WhatIs.com*. [online] SearchSecurity. Available at: <https://www.techtarget.com/searchsecurity/definition/PCI-DSS-Payment-Card-Industry-Data-Security-Standard>.

Belay, T. (2021). *Web Security Vulnerability Analysis of Ethiopian Government Offices*. [online] pp.2–7. Available at: <https://www.dpublication.com/wp-content/uploads/2021/10/10-7020.pdf> [Accessed 31 Dec. 2023].

Elbachir, N., Moussaid, E. and Toumanari, A. (2014). Web Application Attacks Detection: A Survey and Classification. *International Journal of Computer Applications*, [online] 103(12), pp.975–8887. Available at: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=05e26eb9e4650b852b3fb57ffdd46f501b71ae5d>.

Esposito, D., Rennhard, M., Ruf, L. and Wagner, A. (2018). *Exploiting the potential of web application vulnerability scanning*. [online] digitalcollection.zhaw.ch. doi:https://doi.org/10.21256/zhaw-3927.

Gov.UK (2018). *Data Protection Act*. [online] Gov.uk. Available at: <https://www.gov.uk/data-protection> [Accessed 31 Dec. 2023].

Hofman, D., Duranti, L. and How, E. (2017). Trust in the Balance: Data Protection Laws as Tools for Privacy and Security in the Cloud. *Algorithms*, 10(2), p.47. doi:https://doi.org/10.3390/a10020047.

Li, X. and Xue, Y. (2018). *A Survey on Web Application Security*. [online] pp.1–2. Available at: <https://www.isis.vanderbilt.edu/sites/isis.vanderbilt.edu/files/bibcite_files/main_0_0.pdf>.

Saeedi, S., Ghazisaeedi, M. and Rezayi, S. (2021). Applying Game-Based Approaches for Physical Rehabilitation of Poststroke Patients: A Systematic Review. *Journal of Healthcare Engineering*, [online] 2021, p.e9928509. doi:https://doi.org/10.1155/2021/9928509.

Yuan Yuan Pan (2019). *Interactive Application Security Testing | IEEE Conference Publication | IEEE Xplore*. [online] ieeexplore.ieee.org. Available at:<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8901378>.

Yang, J., Tan, L., Peyton, J. and A Duer, K. (2019). *Towards Better Utilizing Static Application Security Testing*. [online] IEEE Xplore. doi:https://doi.org/10.1109/ICSE-SEIP.2019.00014.

Setiawan, H., Erlangga, L.E. and Baskoro, I. (2020). *Vulnerability Analysis Using The Interactive Application Security Testing (IAST) Approach For Government X Website Applications*. [online] IEEE Xplore. doi:https://doi.org/10.1109/ICOIACT50329.2020.9332116.

‌

Corrado, E. and Moulaison, H. (n.d.). *Getting Started with Cloud Computing: A LITA Guide*. [online] Available at: <https://core.ac.uk/download/pdf/286572644.pdf#page=47>.

Patil, Sonali. “A Review of the OW ASP Top 10 Web Application Security Risks and Best Practices for Mitigating These Risks | IEEE Conference Publication | IEEE Xplore.” *Ieeexplore.ieee.org*, 2023, ieeexplore.ieee.org/document/10392030. Accessed 30 Jan. 2024.

Li, X., & Xue, Y. (2014). A survey on server-side approaches to securing web applications. *ACM Computing Surveys*, *46*(4), 1–29. <https://doi.org/10.1145/2541315>

Paternò, F., Santoro, C., & Spano, L. D. (n.d.). *User task-based development of multi-device service-oriented applications*. Cnr.It. Retrieved April 27, 2024, from <https://giove.isti.cnr.it/attachments/publications/2010-A2-043.pdf>

Paternó, F., Santoro, C., & Spano, L. D. (2010). User task-based development of multi-device service-oriented applications. *Proceedings of the International Conference on Advanced Visual Interfaces*.

Starke, S. D., & Baber, C. (2018). The effect of four user interface concepts on visual scan pattern similarity and information foraging in a complex decision making task. *Applied Ergonomics*, *70*, 6–17. <https://doi.org/10.1016/j.apergo.2018.01.010>

Suraya, S., & Sholeh, M. (2021). Designing and implementing a database for thesis data management by using the Python flask framework. *International Journal of Engineering, Science and Information Technology*, *2*(1), 9–14. <https://doi.org/10.52088/ijesty.v2i1.197>

Yee, K.-P. (n.d.). *User Interaction Design for Secure Systems*. Zesty.Ca. Retrieved April 27, 2024, from <http://zesty.ca/pubs/csd-02-1184.pdf>

Yee, K.-P. (2002). User Interaction Design for Secure Systems. In *Information and Communications Security* (pp. 278–290). Springer Berlin Heidelberg.