System Requirements Specification

for

AutoPen

**Version 6.1 Approved**

**Prepared by Michael Allen**

**Caleb Hall**

**Calla Robison**

**Myles Scott**

**Joshua Buscher**

**4/14/2024**

**Table of Contents**

[**Revision History**](#_heading=h.6ltov612afoh) **iii**

[**1. Introduction 1**](#_heading=h.1fob9te)

[**1.1 Purpose 1**](#_heading=h.3znysh7)

[**1.2 Document Conventions 1**](#_heading=)

[**1.3 Intended Audience and Reading Suggestions 1**](#_heading=)

[**1.4 Product Scope 1**](#_heading=)

[**1.5 References 2**](#_heading=)

[**2. Overall Description**](#_heading=h.dxquq0idb247) **2**

[**2.1 Product Perspective 2**](#_heading=h.2s8eyo1)

[**2.2 Product Functions 3**](#_heading=)

[**2.3 User Classes and Characteristics 3**](#_heading=)

[**2.4 Operating Environment 6**](#_heading=)

[**2.5 Design and Implementation Constraints 6**](#_heading=)

[**2.6 User Documentation 6**](#_heading=)

[**2.7 Assumptions and Dependencies**](#_heading=) **6**

[**3. External Interface Requirements 7**](#_heading=h.tfoo3e983jbg)

[**3.1 User Interfaces 7**](#_heading=h.2jxsxqh)

[**3.2 Hardware Interfaces 1**](#_heading=)**1**

[**3.3 Software Interfaces 1**](#_heading=)**1**

[**3.4 Communications Interfaces 1**](#_heading=)**2**

[**4. System Features 1**](#_heading=h.nzsk60xmpkla)**3**

[**4.1 System Feature: Automated Penetration Testing 1**](#_heading=)**3**

[**4.2 System Feature: User Profile Management 1**](#_heading=)**4**

[**5. Other Nonfunctional Requirements 1**](#_heading=h.ambpd1yr81kx)**4**

[**5.1 Performance Requirements 1**](#_heading=)**4**

[**5.2 Safety Requirements 1**](#_heading=)**5**

[**5.3 Security Requirements 1**](#_heading=)**5**

[**5.4 Software Quality Attributes 1**](#_heading=)**5**

[**5.5 Business Rules 1**](#_heading=)**6**

[**6. Other Requirements 1**](#_heading=h.xlqospw3fpjr)**6**

[**7. Appendix A: Glossary 1**](#_heading=h.gxd7q0225i42)**8**

[**8. Appendix B: Analysis Models 1**](#_heading=h.cz18rx2b6vcu)**8**

[**9. Appendix C: To Be Determined List**](#_heading=h.r2uwbttfh8a) **19**

# 

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Michael Allen | 9/21/23 | Rewording Section 1.4 and 2 | 1.0 |
| Michael Allen | 9/26/23 | Section 4 and 5 | 1.1 |
| Michael Allen | 9/28/23 | Section 3 and 6 | 1.2 |
| Caleb Hall | 9/29/23 | Overall Editing and formatting | 1.3 |
| Myles Scott | 10/26/23 | Changing title from 1.0 to 2.0, adding use case diagram | 2.0 |
| Caleb Hall | 10/31/23 | Editing 1.5, 3.3 | 2.1 |
| Michael Allen | 10/31/23 | Editing Title Page, Table of Contents | 2.2 |
| Joshua Buscher | 11/21/23 | Editing Section 2.5, 2.7 | 3.1 |
| Caleb Hall | 11/21/23 | System diagram, operating environment, editing | 3.2 |
| Joshua Buscher | 2/5/24 | Removing all usage of words “AI”/”Artificial Intelligence” | 4.0 |
| Caleb Hall | 2/5/24 | Editing, applying change in scope | 4.1 |
| Joshua Buscher | 3/2/24 | Altering Reasons for Changes, editing requirements, | 5.0 |
| Calla Robison | 3/3/24 | Requirements 3.1, 3.2, 3.3, 3.4 | 5.1 |
| Michael Allen | 3/3/24 | Formatting the document | 5.2 |
| Michael Allen and Caleb Hall | 4/14/24 | Worked on implementing missed comments from Version 1 into Version 3 and new comments from Version 2.   Worked on Section 1 Introduction, modifying 1.0. Also Worked on Section 2 Detailing Requirements and Formatting Document. Updated use case diagram and description | 6.0 |
| Calla Robison | 4/14/24 | Worked on Section 3 to include appropriate figures for the visual requirements and edited some requirements to be more technical and actionable.  Made comments on what figures are still being worked on | 6.1 |

# Introduction

The AutoPen Project is a web-based tool created to revolutionize penetration testing in the fast-changing field of cybersecurity. AutoPen integrates the high-quality tool Burp Suite with a web-hosting service, guaranteeing a versatile testing procedure. This system is designed to facilitate autonomous testing. It includes a user-friendly dashboard for easy monitoring and analysis of results. The AutoPen platform performs assessments as needed, providing a cost-effective, efficient, and comprehensive solution for improving cybersecurity resilience in different companies.

Section 1 is an overview of the document's purpose and intended audience. Section 1 uses subsections to accomplish this. Subsection 1.1 covers the purpose of the document. Subsection 1.2 continues with the formatting conventions of the document. Next, Subsection 1.3 details the audience of the product and which further sections are of interest to them. Subsection 1.4 describes the scope of the product. Finally, Subsection 1.5 lists the references of the product.

## Purpose

The purpose of this Software Requirements Specification (SRS) document is to provide a detailed description and comprehensive outline of the AutoPen Project. This document is intended to serve as a foundational guide for the development and implementation of AutoPen, a web-based tool that leverages the capabilities of Burp Suite for advanced, automated penetration testing. The SRS outlines the functional and non-functional requirements, system behavior, user interfaces, and interactions between the system components. It aims to ensure that all project stakeholders, including developers, testers, and end-users, have a clear understanding of the system's objectives, capabilities, and constraints. This document also establishes the framework for ongoing project management and development activities.

## Document Conventions

1. The document will be in Times New Roman size 12 font.
2. The requirements have their own authority.

## Intended Audience and Reading Suggestions

This document is specifically intended for those engaged in the development, implementation, and upkeep of the AutoPen Project. The system's features and capabilities are intended for the primary audience of web developers and penetration testers. The user should prioritize Section 4, which provides a comprehensive overview of the capabilities of the system, and Section 5, which covers important aspects such as system performance, security, and quality specifications. Project managers and IT professionals should thoroughly examine the entire document, paying close attention to Section 2, in order to understand the overall product description, and Section 3, to understand the system specifications. This will facilitate the seamless integration of AutoPen into their IT infrastructure. Stakeholders, such as investors and senior management, may find Section 1 and Section 6 particularly relevant. These sections discuss the purpose of the document and include additional appendices that offer an executive summary of the project's scope and potential business impact. The SRS also fulfills educational objectives for cybersecurity students and researchers, who may find the entire document valuable for comprehending the practical implementation of theoretical concepts.

## Product Scope

The AutoPen project aims to create a web-based automated penetration testing system that utilizes the Burp Suite API for its core testing procedures. In scope functionality is a web platform that enables users to conduct vulnerability assessments and penetration tests using a user-friendly interface. The objective is to offer comprehensive reporting capabilities and immediate feedback mechanisms to facilitate quick decision-making and remediation procedures.

The AutoPen project does not include any functionalities related to Artificial Intelligence for conducting penetration testing or making autonomous decisions based on test results. The project will not create its own virtual machines for web browsing; instead, it will incorporate existing technologies. Additionally, the AutoPen will not possess hardware testing functionalities or provide assistance for testing non-web applications. The program will utilize the Burp Suite API and will not involve the development of testing algorithms beyond this product.

## References

* The URL of the web application is [https://www.autopentest.net](about:blank)/
* Jira site for agile sprint planning can be found at: <https://autopentest.atlassian.net/jira/software/projects/PEN/boards/1>
* Documentation for Burp Suite, the APT used to run penetration testing, can be found at: <https://portswigger.net/burp/documentation>
* Github for the project can be found at: <https://github.com/Caleb-Hall-1015/AutoPen>

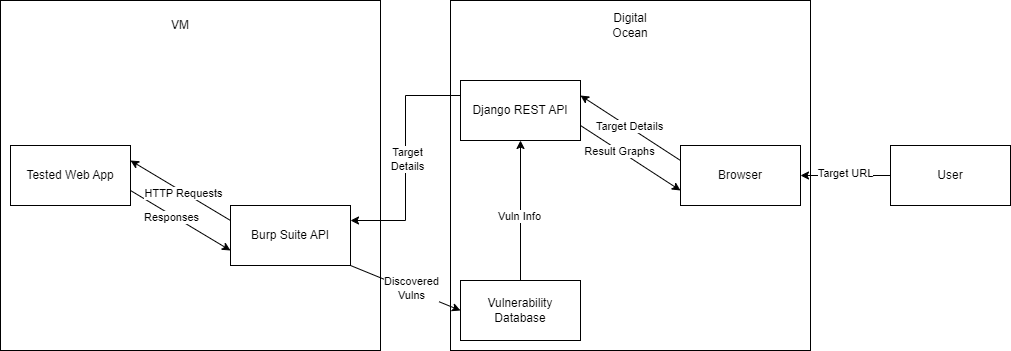
# Overall Description

Section 2 contains information regarding the product use and description. Subsection 2.1 covers the perspective of the product. The product function is located in Subsection 2.2. Subsection 2.3 contains the user information and classes. The operating environment is discussed in Subsection 2.4. Subsection 2.5 details the design and implementation constraints. In Subsection 2.6 there is information on how to find a product tutorial. The assumptions and dependencies are listed in Subsection 2.7.

## Product Perspective

The integration of automated penetration testing represents a new paradigm shift in the field of cybersecurity. By replacing manual systems and innovating with advanced technologies, organizations can enhance their security posture, reduce vulnerabilities, and stay ahead of potential cyber threats. The increased efficiency, accuracy, and scalability offered by automated penetration testing make it an indispensable tool in today's digital landscape. Embracing automation in penetration testing is not only a step towards better security but also a strategic investment in the long-term success and resilience of an organization's IT infrastructure.

.



**Figure 1: System Diagram**

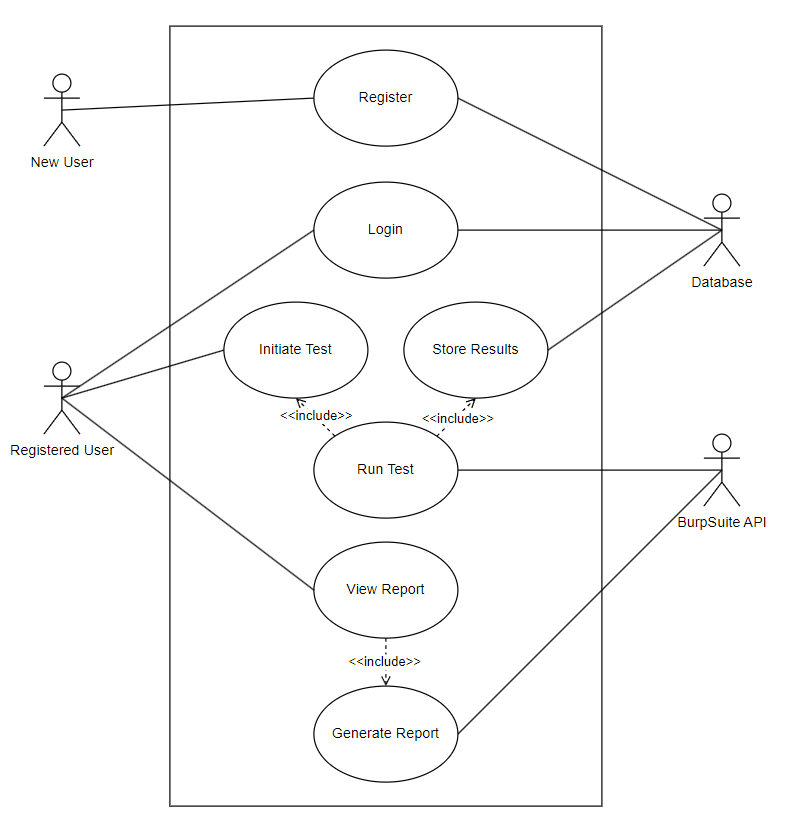
Figure 1 presents the system architecture, illustrating the flow of data and interactions between different components. At the core is a web application tested through a Virtual Machine (VM), which communicates with Burp Suite API to handle HTTP requests and responses, and to identify vulnerabilities. The discovered vulnerabilities are logged into a Vulnerability Database. Additionally, a Django RESTful API is employed to manage target details and vulnerability information, which can be accessed via a browser interface hosted on SiteGround. This browser interface allows the user to input a target URL and visualize result graphs and target details.

## Product Functions

1. The product must allow users to test a website for vulnerabilities
2. The product must be a functional website, allowing any user to visit using the internet

## User Classes and Characteristics

AutoPen has two types of users, a new user and a registered user. Additionally, there are two systems involved in the use cases: the database and the Burp Suite API.



**Figure 2: Use Case Diagram**

Figure 2 outlines the use case flow for AutoPen. It differentiates between registered and unregistered users, showing that registered users can proceed to user login, while new users must undergo user registration. Once authenticated, users can initiate a test or view previous reports. On the other end, the database stores user login records and the scan results. Once a user attempts to view a report, the API will generate the report.

## New User

New user is a user who has never used AutoPen before. They are any web developer, penetration tester, or individual interested in the service of AutoPen.

## Registered User

A registered user is one who has created an account with AutoPen. They are to provide a URL to scan and use the AutoPen system. During the scan, this user is to monitor progress and take manual control to cancel the scan in case of system failure.

## Database

The database is the system where the user records and scan results are stored. This system is to authenticate user login attempts and store and retrieve scan results.

## Burp Suite API

The API is responsible for conducting the vulnerability scans. Once given a URL, this system will perform a security assessment of the site and provide a report containing any vulnerabilities found within.

## Use Case & Relationships

**User Registration:** This use case allows unregistered users to create an account by providing necessary details such as username, email, and password. The system then validates this information and stores it, enabling the user to access the system with their new credentials. No prerequisites exist for this action, and upon completion, the user's data is stored in the system.

**User Login:** Registered users can log into the system by entering their credentials, which the system verifies. If the credentials are correct, the system grants access, establishing a user session. This process requires that a user account already exists and results in the user being logged in.

**Initiate Penetration Test:** Authenticated users can start a penetration test by specifying a target URL and the depth of the test. The system then processes this request and begins the test, requiring the user to be logged in prior to this action.

**Run Test:** Once a user requests a scan to begin, the Burp Suite API will test the site and collect information.

**Store Results:** At the conclusion of the testing cycle, the API will send its findings to the database.

**View Report:** A registered user can request to view the report of a finished test.

**Generate Test Report:** Upon requesting a detailed report from the system, the report will generate. This report includes findings on vulnerabilities, their severity, and suggested remediations. The generation of this report depends on the prior completion of a penetration test

## Operating Environment

To effectively utilize the required system, the user must have the following operating environment:

1. Any form of computer that can run a web page
   1. Windows, Apple, or Linux Desktop/Laptop
   2. Android or Apple Mobile Smartphone
2. Stable Internet Connection
   1. At least 1 Mb/s
3. Up-To-Date Website/Web Browser
   1. Latest version of Chrome, Firefox, Safari, or equivalent browser

## Design and Implementation Constraints

The design and implementation of AutoPen must consider the potential legal and ethical constraints associated with the device. By addressing these concerns, such as intellectual property rights, privacy, consumer protection, plagiarism, and human interaction, AutoPen can be developed and used responsibly. It is crucial to prioritize legal compliance and ethical considerations to ensure the successful integration and acceptance of AutoPen in various industries and domains.

Time is another constraint, as the full software development lifecycle must fall within a single academic year. From project vision to final deliverable, this project can take no longer than 10 calendar months, limiting the scope and depth of the project.

A third constraint is budgetary. Limited funds dictate that all tools used are either provided by the university or are available for a generally low price. As such, the project is unable to offer the full gamut of cybersecurity tools, rather it relies on the professional version of Burp Suite and the basic tier of Siteground.

Similar to financial constraints are technical constraints. The tier of Siteground is low, and as such the project has limited resources. This forces the Burp Suite API to be hosted on a system other than the Siteground cloud server as the scan requires more RAM and CPU cores than the cloud server provides.

The final constraint is skill, as the team is made up of intermediately skilled software engineers. As such, development of in-house penetration testing tools and advanced algorithms is not feasible.

## User Documentation

An instruction panel and video tutorial will be implemented into our website. The AutoPen website aims to enhance user experience, provide clear guidance, and support users throughout their penetration testing journey. This addition will ensure that users can effectively utilize the platform's capabilities and maximize the benefits of automated penetration testing.

## Assumptions and Dependencies

## The system shall adhere to General Data Protection Regulation (GDPR) and similar regional data protection laws.

## The system shall use a hosting platform that has integrated or allows the import of virtual machines.

## The system must use a hosting platform that has the ability to host back-end algorithm tasks.

## The system shall use Django RESTful API to implement back-end functionality.

# External Interface Requirements

Section 3 covers the External interface requirements associated with the product. The format of Section 3 is as follows: Section 3.1 covers the user interface requirements, Section 3.2 covers the hardware interface requirements, Section 3.3 covers the software interface requirements, and Section 3.4 covers the communications interface requirements.

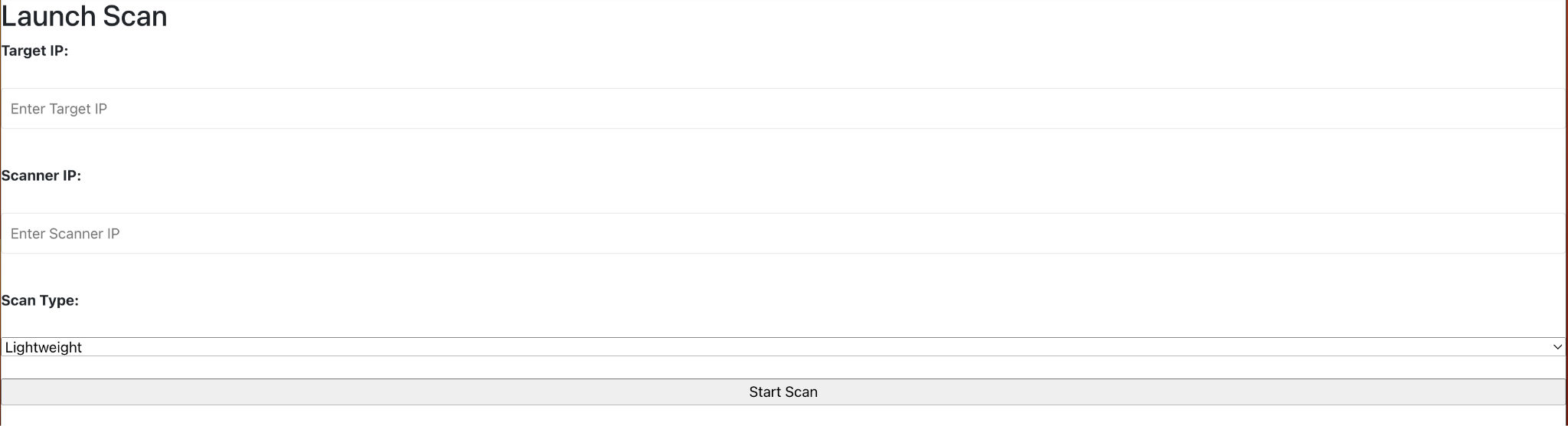
Section 3 outlines the external interface requirements associated with the AutoPen product. This section is structured to cover various aspects of system interfacing, ensuring comprehensive functionality and interoperability. Specifically, Subsection 3.1 details the user interface requirements, Subsection 3.2 addresses the hardware interface requirements, Subsection 3.3 discusses the software interface requirements, and Subsection 3.4 explores the communications interface requirements. Each subsection is designed to provide clear, concise specifications for each interface category to support effective system integration and operation.

## User Interfaces

## Main Interface for Test Execution

## Input Fields for Target:

## The interface shall include specifically labeled input fields for entering IP addresses or domain names required for test execution, as shown in figure 3.



**Figure 3: Start Scan Input Fields**

Figure 3 shows the web page users will interact with to start a scan. There are two text fields and one drop down accompanying a button to start the scan.

* + - * 1. Input fields shall accept alphanumeric characters
        2. Input fields shall support copy-paste actions.

## Start Test Button:

## The interface shall include a "Start Scan" button as shown in figure 3

## The “Start Test” button shall begin the penetration test.

## The “Start Test” button shall be positioned centrally on the page for easy access, as shown in figure 3.

## Results Display Area:

## The interface shall include a dedicated area for displaying test results, organizing information into categories such as vulnerabilities found, severity levels, and suggested actions.

## Test results shall be colorized based on severity (red for high, yellow for medium, green for low).

## GUI Standards

The system will adhere to modern web application design principles, ensuring ease of use and intuitive navigation in the graphical user interface (GUI).

## Consistent Design:

## The GUI shall maintain a uniform font type and size across all pages, using Arial, 14-point for text and 16-point for headings

## Navigation buttons such as "Home," "About Us," and "FAQ" shall be located at the top of every page, consistently designed and colored, as shown in Figure 6.

## Intuitive Navigation:

## Navigation through the GUI shall be facilitated by a fixed navigation bar containing links to all major sections of the application, as shown in Figure 6.

**Figure 6: Fixed Navigation Bar**

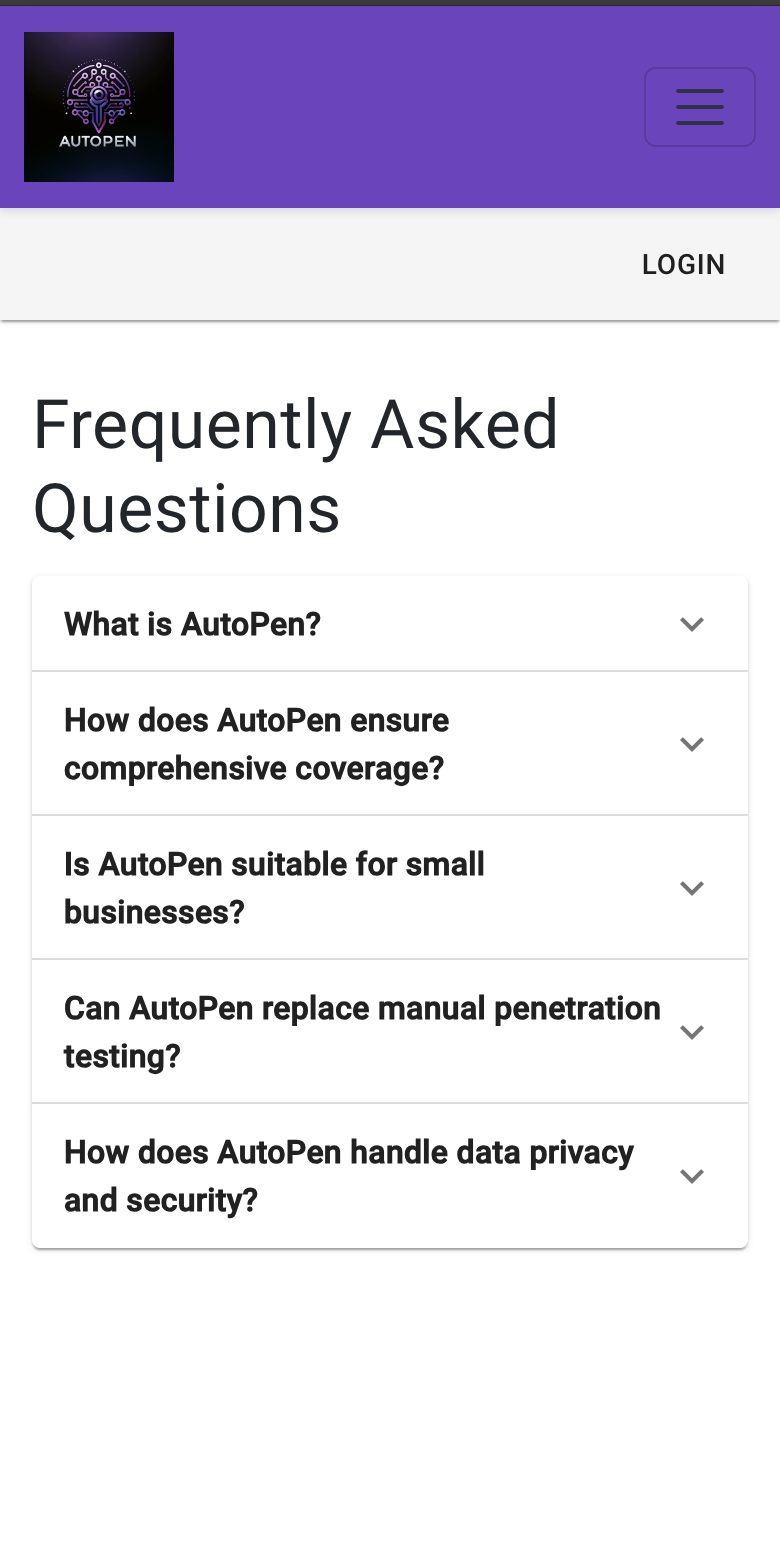
Figure 6 shows the navigation bar present on every page. The bar allows for users to switch to the Home, About Us, Contact Us, Pricing, and FAQ pages

## Responsive Layout:

## The layout shall adjust to fit the screen sizes of devices ranging from mobile phones to desktop monitors without losing functionality or aesthetic theme, as shown in Figures 7, 8, and 9.

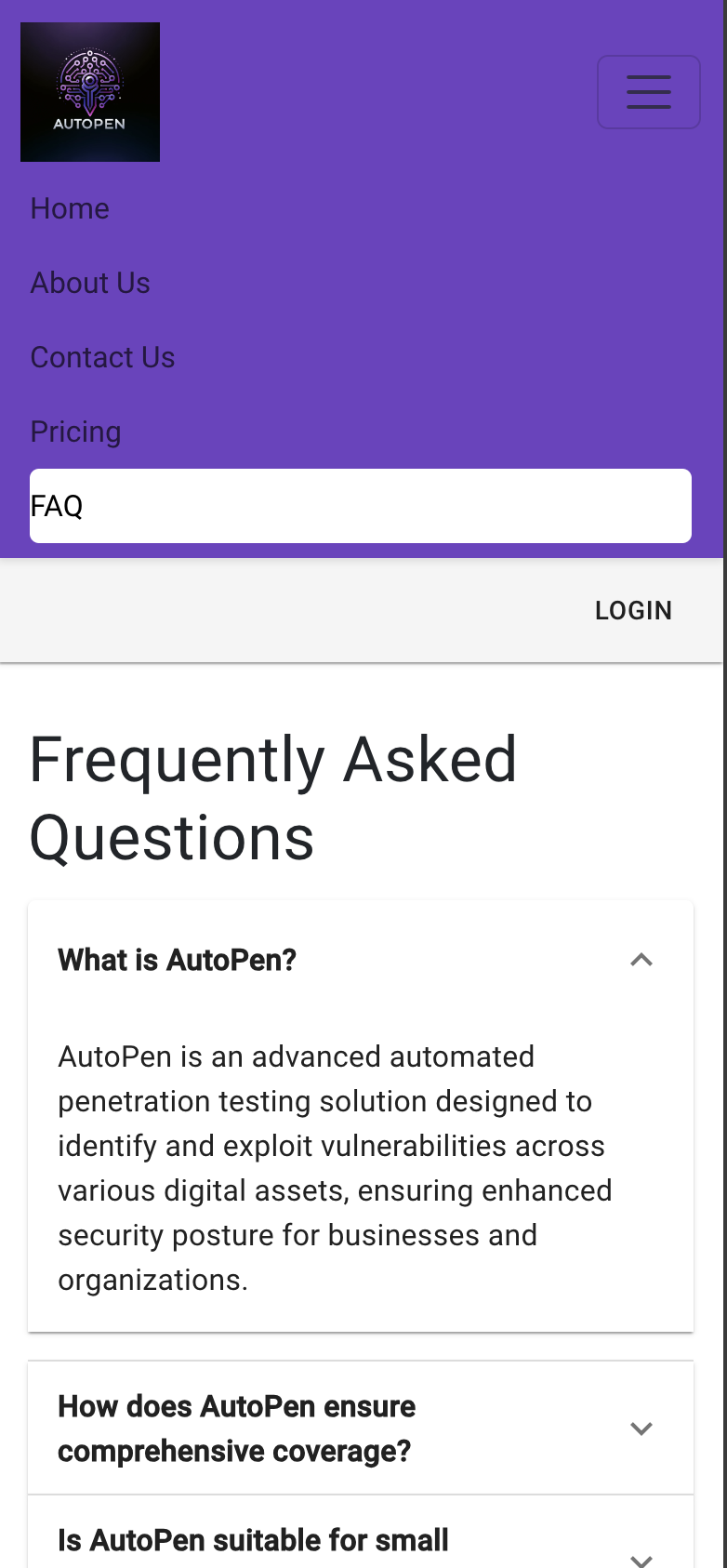
**Figure 7: FAQ Page on Full Desktop Screen**

Figure 7 shows the desktop version of the FAQ page. It shows all details and information from the page.



**Figure 8: FAQ Page on Mobile Size Screen**

Figure 8 shows a mobile version of the FAQ page. There is a closed navigation bar on the top of the page. When compared to the desktop version, there is no loss of functionality or visuals.



**Figure 9: FAQ Page on Mobile Sized Screen with Extended Tabs**

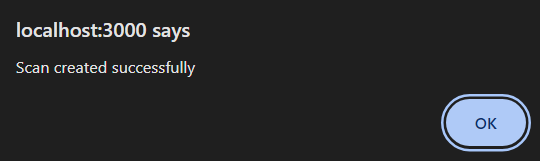
Figure 9 shows the FAQ page as it would appear on a smartphone. Here, the navigation bar is extended.

## User-Friendly Controls:

## All interactive GUI elements such as buttons, dropdowns, and sliders shall respond to user interactions with visual feedback such as highlighting or animation, as shown in Figure 6

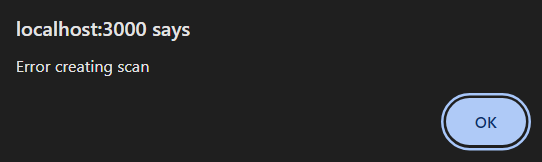
## Feedback Mechanisms:

## Upon completion of actions within the GUI, users shall receive feedback through messages like "Action Successful" or "Error Detected," displayed in a temporary, dismissible pop-up, as shown in Figures 10 and 11.



**Figure 10: Success Message**

Figure 10 is a picture of the message shown on the web page after a scan is successfully started.



**Figure 11: Error Message**

Figure 11 shows the pop up message given to users upon an error when configuring the scan.

## Error Messages

## Error messages shall be displayed as pop-up banners at the top of the screen, as shown in Figure 11.

## UI Components

The system will incorporate several critical components with user interfaces, each designed to enhance user interaction and system usability. Here are the specific requirements for each:

## Settings:

## The system shall allow users to modify settings related to system behavior, notifications, and personalization.

## Changes made in the settings interface shall persist across user sessions.

## Results Page:

## The system shall present a results page with scan result information on completed processes, tests, or relevant activities.

## Users shall be able to filter and sort results based on various parameters such as date, severity, and type.

## User Profile:

## The system shall feature a user profile interface accessible from the main menu as shown in Figure 12.

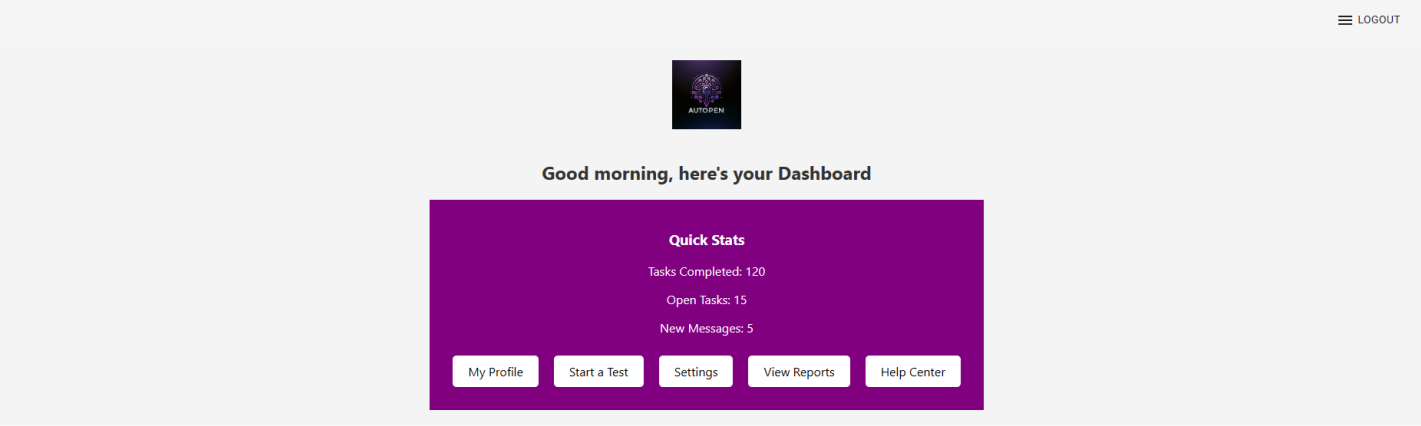
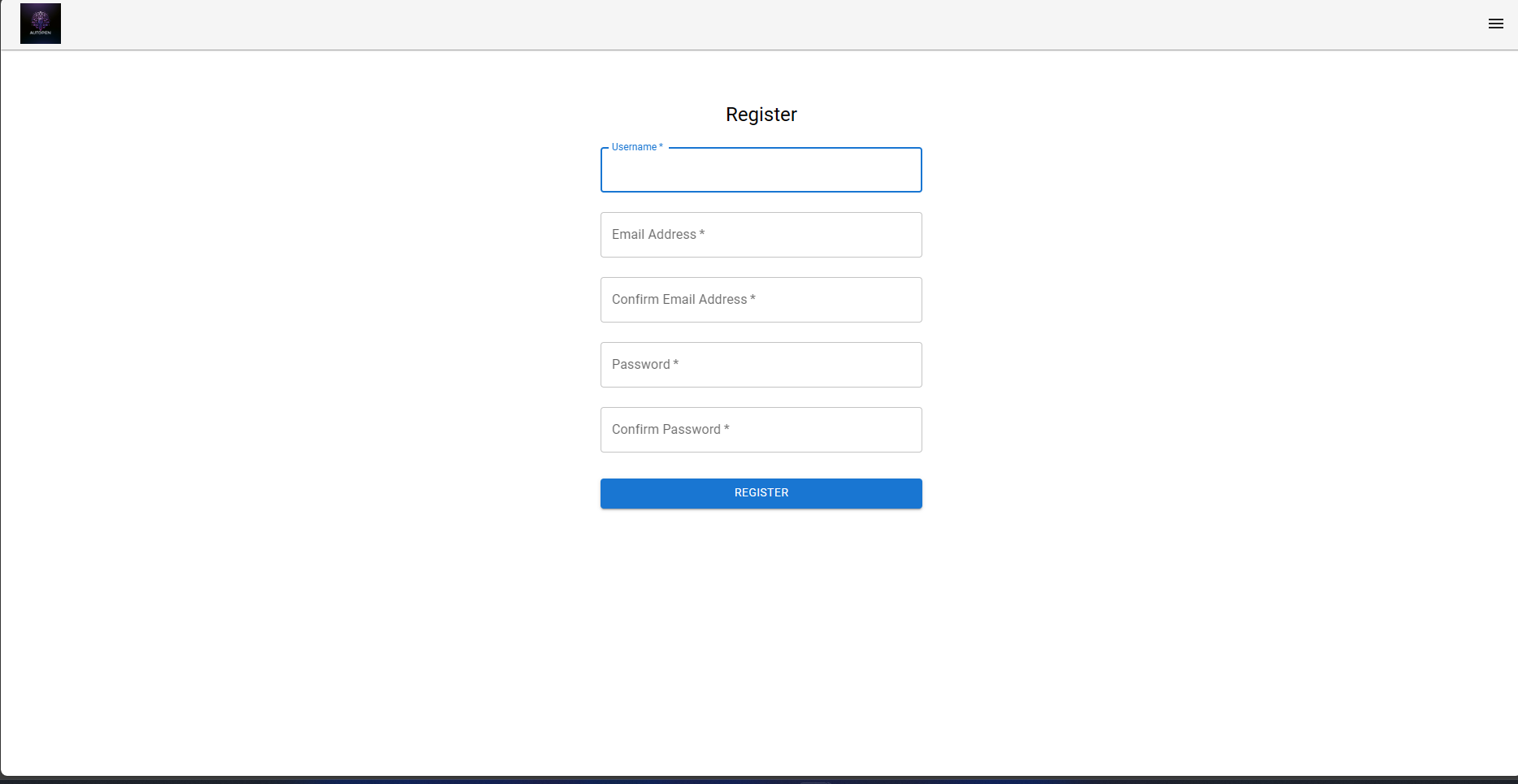
**Figure 12: User Profile and Dashboard Accessible from Main Screen**

Figure 12 shows the user dashboard after the user is logged in. This allows users to navigate the web page through the use of buttons. In addition, metrics are provided to the user to show the state of their work.

## The system shall allow users to view and edit their profile information, including but not limited to name, contact details, and password.

## Changes to the user profile shall be reflected throughout the system within 24 hours.



**Figure 13: Example User Interface Screen**

Figure 13 shows an example user interface for a registration page. The form fields prompt the user for an email address, with a note indicating that a valid email address is required, a username, and a password, along with a password confirmation field. The password field includes instructions specifying the password criteria such as a minimum length and restrictions on similarity to other personal information.

## Hardware Interfaces

## The application shall be hosted on Siteground's cloud server.

## The system shall allocate a minimum of 20 GB storage to the web server.

## Software Interfaces

## Connected Software:

## The system shall utilize a PostgreSQL database hosted on Siteground

## The system shall operate within Sitegrounds' web hosting environment.

## The system shall be compatible with a Kali Linux VM.

## The backend of the system shall be developed using the Python Django Web Framework.

## The frontend of the system shall be developed using the React Web Framework.

## Shared Data Mechanism:

### The system shall leverage of data storage and retrieval processes through Django’s RESTful API backend and Axios HTTP request service frontend

## The system shall integrate with the Burp Suite API to facilitate vulnerability and penetration testing services.

## Communications Interfaces

## Functions:

## The system shall support communication functions for email notifications, web browser alerts, and server-to-server communications.

## Data Formatting:

## The system shall accept incoming data including user credentials, target specifications, and user configurations

* + - 1. The system shall generate outgoing data in the form of test results, progress updates, and error messages

## The system shall convert the data received from the burpsuite API into a JSON format

* + - 1. The system shall use the JSON formatted data for data manipulation and visualization

## Communication Standards:

## The system shall employ Secure HTTP (HTTPS) for all web communications.

## Security/Encryption:

## All communications within the system shall be encrypted using TLS 1.3 or as supported by Sitegrounds to ensure data security and integrity.

## Data Transfer Rates:

## The system shall be optimized for broadband connections, with a minimum recommended speed of 5Mbps to facilitate efficient data transfer.

## Services & Communications:

## The system shall employ a Django RESTful API for seamless communication between the frontend and backend components

## Comprehensive API documentation shall be made available in a separate document for reference and integration purposes

## The interface shall allow users to access the website across multiple devices, including PCs, laptops, tablets, and mobile phones.

# System Features

Section 4 is about the system feature of the product, describing what they do, their stimuli and requirements. The formatting for Section 4 is as follows: 4.1 is about automated penetration testing, and Section 4.2 is about user profile management.

## System Feature: Test Configuration

## Description and Priority

Enables users to configure penetration tests by specifying target systems and setting parameters such as test depth and intensity, ensuring tailored security assessments.

Priority: High

Benefits: 9 (Essential for customizable and effective security assessments)

Penalty: 4 (Inadequate configuration options may lead to suboptimal testing)

Cost: 7 (Complex development due to the need for flexible test parameters)

Risk: 8 (Risks of misconfiguration leading to inaccurate testing)

Overall Score: 7

## Stimulus/Response Sequences

1. Configuration: User specifies target systems and test parameters.
2. Validation: System confirms settings and prepares for testing.
3. Initiation: User starts the penetration test based on configured settings.

## Functional Requirements

## The system shall allow users to specify target systems and set parameters for depth and intensity of tests.

## The system shall validate configurations to ensure they meet security testing protocols.

## The system shall provide real-time feedback during penetration tests to inform users of progress.

## The system shall adjust testing parameters dynamically based on system feedback and user input.

## System Feature: Vulnerability Detection

## Description and Priority

Automatically identifies and categorizes vulnerabilities within targeted systems, crucial for effective security management.

Priority: High

Benefits: 9 (Critical for detecting and addressing vulnerabilities swiftly)

Penalty: 4 (Failure in detection could compromise entire security posture)

Cost: 6 (Requires ongoing updates to detection algorithms)

Risk: 7 (Potential for overlooking or misclassifying vulnerabilities)

Overall Score: 7.5

## Stimulus/Response Sequences

1. Detection: System scans the configured targets for vulnerabilities.
2. Categorization: Identified vulnerabilities are classified by type and severity.
3. Notification: System alerts users to the detection results.

## Functional Requirements

## The system shall detect vulnerabilities using the most current security protocols.

## The system shall categorize detected vulnerabilities by severity and type.

## The systems hall alert users immediately upon detection of vulnerabilities.

## The system shall recommend actionable steps based on the category and severity of vulnerabilities.

## System Feature: Automated Reporting

## Description and Priority

Generates detailed reports on the vulnerabilities found during tests, offering insights into potential impacts and recommended remedial actions.

Priority: High

Benefits: 8 (Provides essential documentation for compliance and remediation)

Penalty: 5 (Lack of comprehensive reporting could affect remedial processes)

Cost: 5 (Integration with existing systems and data synthesis)

Risk: 6 (Inaccuracies in reporting could lead to inadequate response measures)

Overall Score: 7

## Stimulus/Response Sequences

1. Report Generation: Following a test, the system compiles data into a comprehensive report.
2. Distribution: Reports are made available to users for action.

## Functional Requirements

## The system shall compile test data into user-friendly formats for review.

## The system shall analyze test results to identify critical vulnerabilities and impacts.

## The system shall generate detailed reports that outline findings, impacts, and recommended actions.

## The system shall ensure reports are accessible immediately post-testing to facilitate quick response.

## System Feature: Profile Creation

## Description and Priority

Allows users to create new profiles by entering personal information, such as name, email address, and password. This feature supports the secure and efficient setup of new user accounts.

Priority: Medium

Benefits: 8 (Facilitates user onboarding and personalization)

Penalty: 5 (Inadequate profile creation features may deter new users)

Cost: 3 (Standard feature across platforms)

Risk: 6 (Potential risks include data integrity and privacy concerns)

Overall Score: 7

## Stimulus/Response Sequences

1. Input: User provides required information via a registration form.
2. Validation: System checks the completeness and format of the input data.
3. Creation: On successful validation, a new user profile is created.
4. Confirmation: User receives a confirmation of successful profile creation.

## Functional Requirements

## System shall allow input of name, email, and password for new profiles.

## System shall validate email format and check for existing accounts.

## System shall enforce password complexity requirements.

## System shall send a verification email to confirm the user's email address.

## System Feature: Profile Editing

## Description and Priority

Enables users to update or modify their profile details, such as name and contact information, ensuring their personal and professional data remains current.

Priority: Medium

Benefits: 7 (Enhances user satisfaction and data accuracy)

Penalty: 4 (Outdated or incorrect user information could impact service delivery)

Cost: 2 (Relatively simple to implement)

Risk: 5 (Risks associated with unauthorized access and data breaches)

Overall Score: 6.5

## Stimulus/Response Sequences

1. Access: User logs into their profile and accesses the edit section.
2. Modification: User makes changes to their profile details.
3. Verification: System validates the updated information.
4. Update: Changes are saved and the user is notified of the update success.

## Functional Requirements

## System shall provide an interface for users to edit profile details.

## System shall verify changes against security standards before updating.

## System shall log all changes for audit purposes.

## System shall confirm with users post-update via email or in-app notification.

## System Feature: Profile Deletion

## Description and Priority

Allows users to delete their profiles, requiring authentication to ensure that deletion requests are legitimate and secure.

Priority: Medium

Benefits: 7 (Supports user autonomy and data privacy)

Penalty: 5 (Improper handling could lead to data persistence concerns)

Cost: 3 (Standard feature but critical for compliance)

Risk: 6 (Risks include unauthorized deletions and data loss)

Overall Score: 6.5

## Stimulus/Response Sequences

1. Initiation: User requests profile deletion.
2. Authentication: System re-authenticates the user to confirm deletion rights.
3. Deletion: Upon successful authentication, the profile is permanently deleted.
4. Confirmation: User receives confirmation of deletion.

## Functional Requirements

* + - 1. System shall require user re-authentication before profile deletion.
      2. System shall permanently delete all user data upon confirmed deletion.
      3. System shall log deletion requests and actions for compliance and audit.
      4. System shall inform the user of successful deletion via email confirmation.

## System Feature: Password Management

## Description and Priority

Facilitates the management of user passwords, including changing passwords and recovering forgotten passwords, enhancing security and user access control.

Priority: High

Benefits: 9 (Critical for maintaining account security)

Penalty: 6 (Inadequate password management could lead to unauthorized access)

Cost: 4 (Requires robust backend support for security features)

Risk: 7 (High risk if password management is compromised)

Overall Score: 8

## Stimulus/Response Sequences

1. Request: User initiates a password change or recovery.
2. Authentication: System verifies user identity for password operations.
3. Reset/Change: User sets a new password following authentication.
4. Confirmation: System confirms the password update.

## Functional Requirements

## System shall provide secure mechanisms for password reset and change.

## System shall enforce password complexity and expiration policies.

## System shall offer password recovery options through secure methods.

## System must notify the user of password changes via email or SMS.

# Other Nonfunctional Requirements

Section 5 is about covering all nonfunctional requirements that were not mentioned in earlier sections. The formatting of Section 5 is as follows: 5.1 is about requirements relating to performance, 5.2 is about requirements related to safety, 5.3 is about requirements related to security, 5.4 is about general software quality attributes, and 5.5 is about the business rules related to the product.

## Performance Requirements

## User input processing shall complete within 2 seconds under normal operating conditions.

## The system shall update progress on operations in real-time with intervals not exceeding 3 seconds.

## The system must maintain operational performance for up to 100 simultaneous users conducting penetration tests.

## Webpages under the system domain shall load within 3 seconds using standard broadband internet access.

## Safety Requirements

## The system shall strictly adhere to the target specified by the user.

## The system shall provide clear disclaimers and guidance to users on ethical use.

## Security Requirements

## The software shall encrypt all user data using industry-standard protocols for password hashing.

## The system shall employ unique session identifiers to mitigate session hijacking.

## The software shall protect user sessions against cookie theft.

## The software shall host all algorithms on a separate server, isolated from the primary website, with direct access to these models by external entities being strictly prohibited.

## The software shall conduct bi-weekly security audits.

## All data communication within the system shall employ TLS 1.2 or higher for encryption.

## The software shall only allow registered and authorized users to view respective penetration test reports.

## The system shall log all unauthorized access attempts and alert administrators.

## System functionalities shall vary based on user role, with distinct permissions for 'Normal User' and 'Admin'.

## Access to sensitive data shall be restricted based on the user's role: 'Normal User' can only access their own data, while 'Admin' can access all user data.

## Software Quality Attributes

The following requirements are basic goals that the systems should attempt to follow. These are guidelines intended to direct the development and maintenance of the system. These attributes set the standard for the system’s performance, maintainability, and user interface design.

## The software should be able to accommodate changes in penetration testing methodologies or emerging threats with minimal modifications.

## The system should be accessible 99.5% of the time.

## The penetration testing results shall have an accuracy rate of at least 95%.

## The system shall be able to integrate new tools or functionalities without major architectural changes.

## The system shall be able to seamlessly interact with common third-party platforms or services, if needed, such as vulnerability databases.

## The system shall be web based, with any auxiliary tools or scripts working across different OS platforms.

## The system shall not crash during a penetration test, handling errors gracefully.

## The system shall be designed in a way such that they can be used in different contexts or projects.

## The system shall handle unexpected inputs or situations without failing

## The system shall have provisions to be tested easily, both for individual units and end-to-end functionality.

## The user interface shall be intuitive with a preference towards ease of use, even if it comes with a slight learning curve.

## Business Rules

## Only admins or creators of the pentesting software shall be allowed to alter system mechanisms such as data available to the user as a result of the scan, access to the overall color and font theme of the application, and access to the BurpSuite API separately from the web application.

# Other Requirements

Section 6 is about covering all miscellaneous requirements not covered earlier. The formatting for Section 6 is as follows, 6.1 is about requirements relating to reuse objectives, 6.2 is about requirements related to the database, 6.3 is about internationalization requirements, 6.4 is about legal requirements, and 6.5 is about requirements related to accessibility.

## Reuse Objectives

* + 1. The system shall use a RESTful API for intersystem communication with third-party integrations.

## Database Requirements

## The database must support scalability to handle up to 10,000 user records and 400,000 penetration test results.

## The database shall perform daily incremental backups and weekly full backups to an offsite storage server.

## The database shall have strict role-based access to ensure that only authorized personnel can view or modify the database.

## The system shall use a third-party vulnerability database to retrieve vulnerability metrics.

## Internationalization Requirements

## The system should support multiple languages, starting with English, Spanish, and French.

## The system should be able to handle and convert multiple currencies if there is a payment system in place.

## The system shall have user profiles and scheduling features with accurate user time zones

## Legal Requirements

## The system shall adhere to international data protection regulations like GDPR (General Data Protection Regulation) for European users and CCPA (California Consumer Privacy Act) for California residents.

## The system shall have features to prevent misuse, with users needing necessary permissions to conduct penetration tests on the target.

## The system shall properly license, and follow the terms and conditions of all third party tools, libraries and services.

## Accessibility Requirements

## The website shall be designed following W3C Web Content Accessibility Guidelines.

## The website shall have adequate contrast ratios and readable fonts.

## The website shall be navigable using just the keyboard.

# Appendix A: Glossary

Penetration Testing (Pentest): The practice of testing a computer system, network, or web application to find vulnerabilities that attackers could exploit.

SRS: Software Requirements Specification. A document that describes the features, behaviors, and attributes of a software system.

TLS: Transport Layer Security. A protocol ensuring privacy and data security between two communicating applications.

RESTful API: Representational State Transfer. A set of rules that developers follow when they create their API, allowing for interaction between systems using HTTP.

# Appendix B: Analysis Models

**Stakeholders:**

1. Users (Registered and Unregistered)
2. Developers and IT Administrators
3. External vulnerability databases
4. Payment gateway providers
5. Hosting service providers

**System Boundaries:**

1. The system interacts with external entities, including external vulnerability databases, payment gateways, and cloud storage services.
2. Users interact with the system via a web-based interface, which includes data entry screens and data display screens.
3. The system operates within a cloud-based environment, including cloud servers for hosting and data storage.

**Non-Functional Requirements:**

1. Security and compliance with legal constraints, including access control, encryption, and adherence to relevant laws and regulations.
2. Scalability to handle increased user load and data storage needs.
3. Availability to ensure that the system is accessible and operational.
4. Performance to provide efficient penetration testing and report generation.
5. Usability with a user-friendly web interface.
6. Reliability for accurate and consistent penetration testing results.
7. Maintainability for ongoing updates, patches, and system enhancements.

# Appendix C: To Be Determined List

1. Database Selection: Specific database technology to be used hasn't been finalized.
2. Third-Party Integration: Deciding on whether to integrate third-party vulnerability databases directly.
3. Pricing Model: How users will be charged for the service is TBD.
4. Notification System: How and when users receive notifications about test results or system updates.
5. User Data Retention Policy: Duration and conditions under which user data and reports will be stored.

(These are placeholders and might change based on the actual pending decisions in a real-world project.)