**System Test Plan**

**For**

**AutoPen**

***Caleb Hall***

***Michael Allen***

***Calla Robison***

***Myles Scott***

***Joshua Buscher***

**Revision history**

| Version / Author | Date |
| --- | --- |
| 1.0 / Joshua Buscher | 10/16/2023 |
| 1.1 / Michael Allen | 10/17/2023 |
| 1.2 / Caleb Hall | 10/18/2023 |
| 2.0 / Michael Allen | 11/14/2023 |
| 2.1 / Michael Allen | 12/5/2023 |
| 2.2 / Michael Allen | 12/6/2023 |
| 2.3 / Caleb Hall | 12/6/2023 |
| 3.0 / Joshua Buscher | 1/25/2024 |
| 3.1 / Joshua Buscher, Michael Allen, Caleb Hall | 2/25/2024 |
| 4.0 / Joshua Buscher | 4/2/2024 |
| 4.1 / Joshua Buscher | 4/11/2024 |
| 4.2 / Michael Allen, Caleb Hall | 4/14/2024 |
| 4.3 / Michael Allen, Caleb Hall | 4/19/2024 |

Table of Contents

[1. Introduction 1](#_heading=h.gjdgxs)

[1.1 Purpose 2](#_heading=h.30j0zll)

[1.2 Objectives 2](#_heading=h.1fob9te)

[2. Functional Scope 2](#_heading=h.3znysh7)

[3. Overall Strategy and Approach 2](#_heading=h.2et92p0)

[3.1 Testing Strategy 2](#_heading=h.tyjcwt)

[3.1.1 Functionalities Testing 3](#_heading=h.siuokrgineci)

[3.1.2 UI Testing 3](#_heading=h.e68coc46u0ru)

[3.1.3 Workflow Testing 3](#_heading=h.i8slitmqaazf)

[3.1.4 Access Testing 3](#_heading=h.4quyvylbge6f)

[3.2 System Testing Entrance Criteria 3](#_heading=h.3dy6vkm)

[3.3 Testing Types 4](#_heading=h.1t3h5sf)

[3.4 Suspension Criteria and Resumption Requirements 12](#_heading=h.4d34og8)

[3.4.1 Suspension Criteria 12](#_heading=h.194ijb7y5raq)

[3.4.2 Resumption Requirements 13](#_heading=h.esb6ch8nxr47)

[4. Execution Plan 13](#_heading=h.2s8eyo1)

[4.1 Execution Plan 13](#_heading=h.17dp8vu)

[5. Traceability Matrix & Defect Tracking 28](#_heading=h.3rdcrjn)

[5.1 Traceability Matrix 28](#_heading=h.26in1rg)

[5.2 Defect Severity Definitions 32](#_heading=h.lnxbz9)

[6. Environment 33](#_heading=h.35nkun2)

[6.1 Environment 33](#_heading=h.1ksv4uv)

[7. Assumptions 34](#_heading=h.2jxsxqh)

[8. Risks and Contingencies 34](#_heading=h.z337ya)

# Introduction

## Purpose

This document is a test plan for AutoPen System Testing, produced by the System Testing team. It describes the testing strategy and approach to testing the team will use to verify that the application meets the established requirements of the business prior to release.

## Objectives

* Meets the requirements, specifications and the Business rules.
* Supports the intended business functions and achieves the required standards.
* Satisfies the Entrance Criteria for User Acceptance Testing.

# Functional Scope

The modules in the scope of testing for AutoPen System Testing are mentioned in the document attached in the following path:

1. The Software Design Document:  
   <https://docs.google.com/document/d/1bsdYF6pc5wq42goeOdWyzdGO5NMxhz6m/edit?usp=sharing&ouid=107851779739044104235&rtpof=true&sd=true>
2. The System Requirements Specification Document:  
   <https://docs.google.com/document/d/1--zXq_zFsEa_CdH05QXsuDOT1ZxZHEgw/edit?usp=sharing&ouid=107851779739044104235&rtpof=true&sd=true>
3. Section 3.1 of this document

# Overall Strategy and Approach

## Testing Strategy

The AutoPen System Testing will include testing of all functionalities that are in scope as detailed in Section 2, Functional Scope. This section will discuss the testing of new functionalities, modified functionalities, screen level validations, workflows, functionality access, and testing of internal & external interfaces. The objective is to ensure a comprehensive assessment of the AutoPen system by meticulously detailing both functional and non-functional areas that require evaluation. This section will describe the methodologies used in subsequent subsections to test different types of functionalities.

## Functionalities Testing

**Test Objective:** Test user login, registration, scan start, and scan retrieval

**Technique:** Mimic user interaction with the system. Input test data into data fields. Attempt to start scans and view test results.

**Completion Criteria:** All test cases pass. When given correct information, the system shall execute the expected function. When given incorrect information, the system shall not execute.

## UI Testing

**Test Objective:** Perform screen level validations to ensure usability and consistency, and errors in design implementation

**Technique:** Navigate through all web pages, noting errors in design, usability, and consistency. Test different orientations and screen sizes and resolutions. Check links for functionality. Using a subject with no prior AutoPen exposure, attempt to complete normal user tasks and note ease of navigation and clarity of information provided.

**Completion Criteria:** Inconsistency within visuals are resolved. The site is viewable and no information is cut off across all tested screen sizes. Hyperlinks are clickable and deliver the user to the correct endpoint. New user was able to navigate the web pages and complete assigned tasks within one hour and did not require outside assistance.

## Workflow Testing

**Test Objective:** Examine the workflow process for accuracy and efficiency

**Technique:** After a sprint completes, review all work completed.

**Completion Criteria:** Tasks are accomplished within the assigned sprint, sprint reports (ie burndown) are smooth and linear with a y=-x slope.

## Access Testing

**Test Objective:** Ensure that different roles have appropriate and restricted access to functionalities as per their permissions

**Technique:** As a user, attempt to access the scan results of another user. As an administrator, repeat the process

**Completion Criteria:** Users are able to access their own results, but no results of another user. Administrators are able to view all data.

## System Testing Entrance Criteria

For system testing to commence, certain prerequisites need to be fulfilled. The prerequisites can be divided as follows: URL Access, User Status and Test Status.

**Criteria for Testing Readiness:**

* URL Access: The test environment should be accessible at autopentest.com.
* User Status: A user or developer must be logged into the system.
* Test Status: AutoPen should be in a “test-ready” state, implying that the application is stable, all primary features are implemented, and it is ready to undergo the testing process using BurpSuite for comprehensive scanning and penetration testing activities.

## 

## Testing Types

### Usability Testing

**Objective**: The objective is to assess the user-friendliness and usability of the "AutoPen" web interface. Specifically, user interface attributes, and its respective content, will be tested for accuracy and usability.

**Details**: This testing focuses on the usability of the AutoPen web interface, ensuring the user has a smooth and logical experience, with special emphasis on ensuring a coherent navigation flow. The elements being checked for this form of testing includes access shortcuts, consistent navigation patterns, and legibility of text

System Requirement Specification 3.1.1.1.1 The interface shall include specifically labeled input fields for entering IP addresses or domain names required for test execution.

System Requirement Specification 3.1.1.1.2 Input fields shall accept alphanumeric characters.

System Requirement Specification 3.1.1.1.3 Input fields shall support copy-paste actions.

System Requirement Specification 3.1.1.2.1 The interface shall include a "Start Scan" button.

System Requirement Specification 3.1.1.2.2 The “Start Test” button shall begin the penetration test.

System Requirement Specification 3.1.1.2.3 The “Start Test” button shall be positioned centrally on the page.

System Requirement Specification 3.1.1.3.1 The interface shall include a dedicated area for displaying test results, organizing information into categories such as vulnerabilities found, severity levels, and suggested actions.

System Requirement Specification 3.1.2.1.1 Navigation buttons such as "Home," "About Us," and "FAQ" shall be consistently designed and colored.

System Requirement Specification 3.1.2.2.1 Navigation through the GUI shall be facilitated by a fixed navigation bar containing links to all major sections of the application.

System Requirement Specification 3.1.2.3.1 The layout shall adjust to fit the screen sizes of devices ranging from mobile phones to desktop monitors

without losing functionality or aesthetic theme.

System Requirement Specification 3.1.2.4.1 All interactive GUI elements such as buttons, dropdowns, and sliders shall respond to user interactions with visual feedback such as highlighting or animation.

System Requirement Specification 3.1.2.5.1 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.

System Requirement Specification 3.1.2.6.1 Error messages shall be displayed as pop-up banners at the top of the screen.

System Requirement Specification 3.1.3.1.1 The system shall allow users to modify settings related to system behavior, notifications, and personalization.

System Requirement Specification 3.1.3.1.2 Changes made in the settings interface shall persist across user sessions.

System Requirement Specification 3.1.3.2.1 The system shall present a results page with scan result information on completed processes.

System Requirement Specification 3.1.3.3.1 The system shall feature a user profile interface accessible from the main menu.

System Requirement Specification 3.1.3.3.2 The system shall allow users to view and edit their profile information, including but not limited to name, contact details, and password.

System Requirement Specification 3.1.3.3.3 Changes to the user profile shall be reflected throughout the system within 24 hours.

System Requirement Specification 3.2.1 The application shall be hosted on SiteGround's cloud server.

System Requirement Specification 3.2.2 The system shall allocate a minimum of 20 GB storage to the web server.

System Requirement Specification 3.3.1.1 The system shall utilize a MySQL database hosted on SiteGround

System Requirement Specification 3.3.1.2 The system shall operate within SiteGrounds' web hosting environment.

System Requirement Specification 3.3.1.3 The system shall be compatible with a Kali Linux VM.

System Requirement Specification 3.3.1.4 The backend of the system shall be developed using the Python Django Web Framework.

System Requirement Specification 3.3.1.5 The frontend of the system shall be developed using the React Web Framework.

System Requirement Specification 3.3.2.1 The system shall leverage data storage and retrieval processes through Django’s RESTful API backend and Axios HTTP request service frontend

System Requirement Specification 3.3.2.2 The system shall integrate with the Burp Suite API.

System Requirement Specification 3.4.1.1 The system shall support communication functions for email notifications, web browser alerts, and server-to-server communications.

System Requirement Specification 3.4.2.1 The system shall accept incoming data including user credentials, target specifications, and user configurations.

System Requirement Specification 3.4.2.2 The system shall generate outgoing data in the form of test results and error messages.

System Requirement Specification 3.4.2.3 The system shall convert the data received from the Burp Suite API into a JSON format.

System Requirement Specification 3.4.2.4 The system shall use the JSON formatted data for data manipulation and visualization.

System Requirement Specification 3.4.3.1 The system shall employ Secure HTTP (HTTPS) for all web communications.

System Requirement Specification 3.4.4.1 All communications within the system shall be encrypted using TLS 1.3.

System Requirement Specification 3.4.5.1 The system shall load web pages within 3 seconds of a connection attempt for Internet connections with a minimum speed of 5Mbps.

System Requirement Specification 3.4.6.1 The system shall employ a Django RESTful API for communication between the frontend and backend components.

System Requirement Specification 3.4.6.2 The interface shall allow users to access the website across multiple devices, including PCs, laptops, tablets, and mobile phones.

### Functional Testing

**Objective:** Validate that "AutoPen" fulfills all functional requirements and operates in accordance with predefined business rules.

**Details:** This testing focuses on individual functionalities to ensure they align with the stipulated business and functional requirements. All business rules or conditions are verified to ensure they are correctly implemented in the system. Resolutions to issues, feedback, and change requests documented during the project lifecycle will also be subject to functional testing.

System Requirement Specification 4.1.3.1 The system shall allow users to specify the target URL.

System Requirement Specification 4.1.3.2 The system shall allow users to set the scanner IP address.

System Requirement Specification 4.1.3.3 The system shall allow users to set vulnerability scan intensity.

System Requirement Specification 4.1.3.4 The system shall validate configurations to ensure they meet security testing protocols.

System Requirement Specification 4.2.3.1 The system shall detect vulnerabilities using the Burp Suite vulnerability scanner.

System Requirement Specification 4.2.3.2 The system shall categorize detected vulnerabilities by severity.

System Requirement Specification 4.2.3.3 The system shall recommend remediation steps based on the vulnerability type.

System Requirement Specification 4.3.3.1 The system shall compile test results into a table.

System Requirement Specification 4.3.3.2 The system shall analyze test results to identify critical vulnerabilities and impacts.

System Requirement Specification 4.3.3.3 The system shall generate reports that outline findings, impacts, and recommended actions.

System Requirement Specification 4.3.3.4 The system shall ensure reports are accessible within 1 hour following the completion of the test.

System Requirement Specification 4.4.3.1 The system shall allow users to input a name during profile creation.

System Requirement Specification 4.4.3.2 The system shall allow users to input an email during profile creation.

System Requirement Specification 4.4.3.3 The system shall allow users to input a password during profile creation.

System Requirement Specification 4.4.3.4 The system shall validate the input email follows traditional email format.

System Requirement Specification 4.4.3.5 The system shall check for existing accounts using the input email.

System Requirement Specification 4.4.3.6 The system shall enforce password complexity requirements.

System Requirement Specification 4.5.3.1 The system shall provide an interface for users to edit profile details.

System Requirement Specification 4.5.3.2 The system shall verify changes against security standards before updating.

System Requirement Specification 4.5.3.3 The system shall confirm with users post-update via email or in-app notification.

System Requirement Specification 4.6.3.1 System shall require user re-authentication before profile deletion.

System Requirement Specification 4.6.3.2 System shall permanently delete all user data upon confirmed deletion.

System Requirement Specification 4.6.3.3 System shall log deletion requests and actions for compliance and audit.

System Requirement Specification 4.6.3.4 System shall inform the user of successful deletion via email confirmation.

System Requirement Specification 4.7.3.1 System shall provide secure mechanisms for password reset and change.

System Requirement Specification 4.7.3.2 System shall enforce password complexity and expiration policies.

System Requirement Specification 5.1.1 User input processing shall complete within 2 seconds under normal operating conditions, being less than 50% server CPU utilization.

System Requirement Specification 5.1.2 The system must maintain operational performance for up to 100 simultaneous users conducting penetration tests.

System Requirement Specification 5.1.3 Webpages under the system domain shall load within 3 seconds when internet access is faster than 5 Mbps.

System Requirement Specification 5.2.1 The system shall only test the target specified by the user.

System Requirement Specification 5.2.2 The system shall provide disclaimers to users on ethical use.

System Requirement Specification 5.3.1 The software shall encrypt all user data using the Bcrypt protocol for password hashing.

System Requirement Specification 5.3.2 The system shall employ unique session identifiers to mitigate session hijacking.

System Requirement Specification 5.3.3 The system shall employ encryption to mitigate cookie theft.

System Requirement Specification 5.3.4 The system shall detect and prevent session hijacking attempts.

System Requirement Specification 5.3.5 The system shall conduct bi-weekly security audits.

System Requirement Specification 5.3.6 The software shall only allow registered and authorized users to view respective penetration test reports.

System Requirement Specification 5.3.7 The system shall log all unauthorized access attempts.

System Requirement Specification 5.3.8 User role 'Normal User' access shall be restricted to only their own data.

System Requirement Specification 5.3.9 User role 'Admin' shall have access to all user data.

System Requirement Specification 5.4.1 The software should be able to accommodate changes in penetration testing methodologies.

System Requirement Specification 5.4.2 The software should be able to accommodate emerging threats with minimal modifications.

System Requirement Specification 5.4.3 The system should be accessible 99.5% of the time.

System Requirement Specification 5.4.4 The system shall be able to integrate new tools or functionalities using RESTful APIs.

System Requirement Specification 5.4.5 The system shall be web based, with any auxiliary tools or scripts working across different OS platforms.

System Requirement Specification 5.4.6 The system shall not crash during a penetration test, handling errors gracefully.

System Requirement Specification 5.4.7 The system shall handle unexpected inputs or situations without failing.

## Suspension Criteria and Resumption Requirements

This section will specify the criteria that will be used to suspend all or a portion of the testing activities on the items associated with this test plan.

## Suspension Criteria

**Criteria:**

* Any significant malfunction or incident that renders further testing of "AutoPen" impossible or counterproductive will lead to a suspension of testing activities.
* Alterations made to the software, hardware, or database requires suspension of testing.

**Responsibility:**

* The Testing Manager has the discretion to decide if the entire test plan should be rerun or only specific sections of it, especially after significant modifications.

## Resumption Requirements

### Testing activities will resume only after the problematic functionality, which initially caused the halt, has undergone successful retesting.

### A comprehensive review and verification process will be established to ensure the cause for suspension has been rectified.

* Testing will resume for halts due to alterations of the software, hardware, or database after the changes are complete and applied to the testing environment.

# Execution Plan

## Execution Plan

The execution plan outlines the procedure for carrying out this project’s test cases. It ensures coverage of all requirements and is adaptable to accommodate any necessary changes during the testing phase. All test cases for the release are logically sequenced based on their interdependencies and the overall testing strategy. This plan is designed to confirm that AutoPen meets its specified requirements for functionality, database integrity, and performance.

The following is the test plan for AutoPen’s system:

* + 1. **UI Testing** - Refer to 3.1.2 for objectives and details.
    2. **Functionalities Testing -** Refer to section 3.1.1 for objectives and details.
    3. **Workflow Testing** - Refer to section 3.1.3 for objectives and details.
    4. **Access Testing** - Refer to section 3.1.4 for objectives and details.

| **Requirement (From SRS)** | **Test Case ID** | **Input** | **Output** | **Pass/Fail** |
| --- | --- | --- | --- | --- |
| 3.1.1.1.1 The interface shall include specifically labeled input fields for entering IP addresses or domain names required for test execution. | 1.1 | User types in the desired IP or domain target for penetration testing. | The system displays input fields labeled appropriately and accepts the input without errors. | Pass |
| 3.1.1.1.2 Input fields shall accept alphanumeric characters. | 2.1 | User types alphanumeric characters into the website fields | The input fields accept and display the alphanumeric characters correctly. | Pass |
| 3.1.1.1.3 Input fields shall support copy-paste actions. | 3.1 | User copy-pastes into the input field. | The system allows the action and the pasted value appears correctly in the input field. | Pass |
| 3.1.1.2.1 The interface shall include a "Start Scan" button. | 4.1 | User views the interface. | A "Start Scan" button is visible. | Pass |
| 3.1.1.2.2 The “Start Test” button shall begin the penetration test. | 5.1 | User clicks the "Start Test" button. | The system initiates a penetration test. | Pass |
| 3.1.1.2.3 The “Start Test” button shall be positioned centrally on the page. | 6.1 | User views the interface. | The "Start Test" button is centrally positioned on the page. | Pass |
| 3.1.1.3.1 The interface shall include a dedicated area for displaying test results, organizing information into categories such as vulnerabilities found, severity levels, and suggested actions. | 7.1 | User visits the results page. | The results page displays finished test and pertinent information. | Pass |
| 3.1.2.1.1 Navigation buttons such as "Home," "About Us," and "FAQ" shall be consistently designed and colored. | 8.1 | User navigates through various pages of the GUI. | Navigation buttons are visible at the top of every page, consistently designed and colored. | Pass |
| 3.1.2.2.1 Navigation through the GUI shall be facilitated by a fixed navigation bar containing links to all major sections of the application. | 9.1 | User navigates through various sections of the application. | A fixed navigation bar with links to all major sections is present and functions correctly. | Pass |
| 3.1.2.3.1 The layout shall adjust to fit the screen sizes of devices ranging from mobile phones to desktop monitors without losing functionality or aesthetic theme. | 10.1 | User accesses the GUI from different devices (mobile phone, tablet, desktop). | The GUI adjusts to fit different screen sizes without losing functionality or aesthetic appeal. | Pass |
| 3.1.2.4.1 All interactive GUI elements such as buttons, dropdowns, and sliders shall respond to user interactions with visual feedback such as highlighting or animation. | 11.1 | User interacts with buttons, dropdowns, and sliders in the GUI. | Interactive elements respond with visual feedback like highlighting or animation. | Pass |
| 3.1.2.5.1 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. | 12.1 | User completes an action within the GUI. | A temporary, dismissible pop-up displays a message indicating "Action Successful" or "Error Detected.” | Pass |
| 3.1.2.6.1 Error messages shall be displayed as pop-up banners at the top of the screen. | 13.1 | An error occurs during user interaction with the GUI. | Error messages appear as pop-up banners at the top of the screen. | Pass |
| 3.1.3.1.1 The system shall allow users to modify settings related to system behavior, notifications, and personalization. | 14.1 | User accesses and modifies various settings in the settings interface. | The system allows modifications and shows updated settings in real-time. | Pass |
| 3.1.3.1.2 Changes made in the settings interface shall persist across user sessions. | 15.1 | User logs out and logs back in after making changes to settings. | Changes persist across sessions, with the user finding settings as they were left. | Pass |
| 3.1.3.2.1 The system shall present a results page with scan result information on completed processes. | 16.1 | User completes a scan. | A results page is displayed showing detailed information about the completed scan. | Pass |
| 3.1.3.3.1 The system shall feature a user profile interface accessible from the main menu. | 17.1 | User selects the user profile option from the main menu. | The user profile interface is accessible. | Pass |
| 3.1.3.3.2 The system shall allow users to view and edit their profile information, including but not limited to name, contact details, and password. | 18.1 | User accesses and edits their profile information. | The system allows editing and displays the updated profile information correctly. | Pass |
| 3.1.3.3.3 Changes to the user profile shall be reflected throughout the system within 24 hours. | 19.1 | User changes their profile settings. | The changes are reflected throughout the system within 24 hours. | Pass |
| 3.2.1 The application shall be hosted on SiteGround's cloud server. | 20.1 | Admin configures deployment settings on SiteGround. | The application is successfully hosted and running on SiteGround's cloud server. | Pass |
| 3.2.2 The system shall allocate a minimum of 20 GB storage to the web server. | 21.1 | Admin sets up or verifies storage allocation during system setup. | The web server shows a minimum allocation of 20 GB storage. | Pass |
| 3.3.1.1 The system shall utilize a MySQL database hosted on SiteGround. | 22.1 | Admin sets up the database on SiteGround. | A MySQL database is active and correctly functioning on SiteGround. | Pass |
| 3.3.1.2 The system shall operate within SiteGrounds' web hosting environment. | 23.1 | Admin deploys and configures the system on SiteGround. | The system is operational and meets performance benchmarks within SiteGround's hosting environment. | Pass |
| 3.3.1.3 The system shall be compatible with a Kali Linux VM. | 24.1 | User tests system functionality on a Kali Linux virtual machine. | The system functions correctly and integrates with Kali Linux VM. | Pass |
| 3.3.1.4 The backend of the system shall be developed using the Python Django Web Framework. | 25.1 | Developer codes and deploys backend services using Django. | Backend functionality is verified to be running on Django with all endpoints responding as expected. | Pass |
| 3.3.1.5 The frontend of the system shall be developed using the React Web Framework. | 26.1 | Developer creates and deploys frontend components using React. | The frontend is operational, and all React components are rendering correctly. | Pass |
| 3.3.2.1 The system shall leverage data storage and retrieval processes through Django’s RESTful API backend and Axios HTTP request service frontend | 27.1 | Developer sets up RESTful API using Django and connects with frontend via Axios. | Data storage and retrieval are functioning correctly through the setup, with the frontend successfully sending requests and receiving responses via Axios. | Pass |
| 3.3.2.2 The system shall integrate with the Burp Suite API. | 28.1 | Setup API integration with Burp Suite and execute a penetration test. | The system successfully communicates with the Burp Suite API and conducts penetration testing, receiving and processing data as expected. | Pass |
| 3.4.1.1 The system shall support communication functions for email notifications, web browser alerts, and server-to-server communications. | 29.1 | Test email, web alert, and server communication functions. | Email notifications are sent, browser alerts are displayed, and server-to-server communications are successful without any issues. | Pass |
| 3.4.2.1 The system shall accept incoming data including user credentials, target specifications, and user configurations. | 30.1 | Input data including user credentials and target specifications into the system. | The system correctly accepts and stores the incoming data as expected without errors. | Pass |
| 3.4.2.2 The system shall generate outgoing data in the form of test results and error messages. | 31.1 | Request information from the system. | The system produces and displays test results and error messages accurately. | Pass |
| 3.4.2.3 The system shall convert the data received from the Burp Suite API into a JSON format. | 32.1 | Receive data from Burp Suite API. | The data from Burp Suite is successfully converted to JSON format and validated. | Pass |
| 3.4.2.4 The system shall use the JSON formatted data for data manipulation and visualization. | 33.1 | Use the JSON data for various functions within the system. | The JSON data is communicated in the system's interface. | Pass |
| 3.4.3.1 The system shall employ Secure HTTP (HTTPS) for all web communications. | 34.1 | Check all web communications for HTTPS protocol use. | All system communications over the web are secured with HTTPS. | Pass |
| 3.4.4.1 All communications within the system shall be encrypted using TLS 1.3. | 35.1 | Verify encryption protocols during data transmission. | Communications are encrypted using TLS 1.3. | Pass |
| 3.4.5.1 The system shall load web pages within 3 seconds of a connection attempt for Internet connections with a minimum speed of 5Mbps. | 36.1 | Test system performance on various broadband speeds starting from 5Mbps. | The system loads pages at and above the minimum broadband speed of 5Mbps. | Pass |
| 3.4.6.1 The system shall employ a Django RESTful API for communication between the frontend and backend components. | 37.1 | Test the integration and communication via the Django RESTful API. | The Django RESTful API facilitates communication between frontend and backend, handling requests and responses without issues. | Pass |
| 3.4.6.2 The interface shall allow users to access the website across multiple devices, including PCs, laptops, tablets, and mobile phones. | 38.1 | Access the website from various devices. | The website is accessible and fully functional on PCs, laptops, tablets, and mobile phones without any compatibility issues. | Pass |
| 4.1.3.1 The system shall allow users to specify the target URL. | 39.1 | User inputs target URL to be scanned. | The system accepts and displays the specified URL. | Pass |
| 4.1.3.2 The system shall allow users to set the scanner IP address. | 40.1 | User inputs scanner IP. | The system accepts and displays the input IP. | Pass |
| 4.1.3.3 The system shall allow users to set vulnerability scan intensity. | 41.1 | User selects the scan intensity. | The system displays the selected intensity. | Pass |
| 4.1.3.4 The system shall validate configurations to ensure they meet security testing protocols. | 42.1 | User selects to start the scan. | System performs input validation on the user’s input. | Pass |
| 4.2.3.1 The system shall detect vulnerabilities using the Burp Suite vulnerability scanner. | 43.1 | User selects to start the scan. | System transfers scan settings to Burp Suite and requests a new scan. | Pass |
| 4.2.3.2 The system shall categorize detected vulnerabilities by severity. | 44.1 | User views scan results. | Detected vulnerabilities are accurately categorized by their severity. | Pass |
| 4.2.3.3 The system shall recommend remediation steps based on the vulnerability type. | 45.1 | User views scan results. | The system provides remediation steps. | Pass |
| 4.3.3.1 The system shall compile test results into a table. | 46.1 | User views scan results. | Test data is compiled into a table. | Pass |
| 4.3.3.2 The system shall analyze test results to identify critical vulnerabilities and impacts. | 47.1 | Review results from completed tests. | The system analyzes and identifies critical vulnerabilities and their impacts accurately. | Pass |
| 4.3.3.3 The system shall generate detailed reports that outline findings, impacts, and recommended actions. | 48.1 | Generate a report post-testing. | Detailed reports are created outlining findings, impacts, and recommended actions, with clear and actionable information. | Pass |
| 4.3.3.4 The system shall ensure reports are accessible within 1 hour following the completion of the test. | 49.1 | Check report availability in 5 minute intervals after finishing a scan. | Reports are readily accessible and available within one hour following testing. | Pass |
| 4.4.3.1 The system shall allow users to input a name during profile creation. | 50.1 | Input name while creating a new profile. | The system accepts input during profile creation. | Pass |
| 4.4.3.2 The system shall allow users to input an email during profile creation. | 51.1 | Enter an email address to create a new account. | The system accepts input during profile creation. | Pass |
| 4.4.3.3 The system shall allow users to input a password during profile creation. | 52.1 | User creates a password when registering. | The system accepts input during profile creation. | Pass |
| 4.4.3.4 The system shall validate the input email follows traditional email format. | 53.1 | User confirms profile creation. | The system checks and validates the email format. | Pass |
| 4.4.3.5 The system shall check for existing accounts using the input email. | 54.1 | User confirms profile creation. | The system verifies if the email already exists in the database. | Pass |
| 4.4.3.6 The system shall enforce password complexity requirements. | 55.1 | User confirms profile creation. | The system checks the password for complexity requirements and accepts it if it meets the criteria | Pass |
| 4.5.3.1 System shall provide an interface for users to edit profile details. | 56.1 | User accesses the profile edit interface. | The interface for editing profile details allows changes to be made. | Pass |
| 4.5.3.2 System shall verify changes against security standards before updating. | 57.1 | User submits changes to their profile details. | The system verifies the changes against established security standards before updating the profile. | Pass |
| 4.5.3.3 System shall confirm with users post-update via email or in-app notification. | 58.1 | User updates their profile details. | User receives a confirmation via email or an in-app notification confirming the update. | Pass |
| 4.6.3.1 System shall require user re-authentication before profile deletion. | 59.1 | User attempts to delete their profile. | System requires user to re-authenticate before the profile can be deleted. | Pass |
| 4.6.3.2 System shall permanently delete all user data upon confirmed deletion. | 60.1 | User confirms deletion of their profile after re-authentication. | All user data is permanently deleted from the system. | Pass |
| 4.6.3.3 System shall log deletion requests and actions for compliance and audit. | 61.1 | User requests profile deletion. | System logs all details of the deletion request and action. | Pass |
| 4.6.3.4 System shall inform the user of successful deletion via email confirmation. | 62.1 | Profile is deleted. | User receives an email confirmation of the successful deletion of their profile. | Pass |
| 4.7.3.1 System shall provide secure mechanisms for password reset and change. | 63.1 | User requests a password reset or change. | System provides an encrypted process for password reset/change. | Pass |
| 4.7.3.2 System shall enforce password complexity and expiration policies. | 64.1 | User sets or changes a password. | System enforces complexity requirements and expiration policies for passwords, enhancing security. | Pass |
| 5.1.1 User input processing shall complete within 2 seconds under normal operating conditions, being less than 50% server CPU utilization.. | 65.1 | User inputs data into the system. | Input processing is completed within 2 seconds. | Pass |
| 5.1.2 The system must maintain operational performance for up to 100 simultaneous users conducting penetration tests. | 66.1 | Simultaneously test applications, beginning with 75 at once and increasing in intervals of 5 applications until 125 are being tested at once. | Scans progress as usual when at least 100 tests are being performed at a time. | Pass |
| 5.2.1 The system shall only test the target specified by the user. | 67.1 | User specifies a target for a system operation. | System adheres strictly to the specified target without deviation. | Pass |
| 5.2.2 The system shall provide disclaimers to users on ethical use. | 68.1 | User accesses the system. | System displays clear disclaimers and ethical use guidelines to users, ensuring they are well-informed. | Pass |
| 5.3.1 The software shall encrypt all user data using the Bcrypt protocol for password hashing. | 69.1 | User data is processed and stored. | All user data is encrypted using Bcrypt for password hashing. | Pass |
| 5.3.2 The system shall employ unique session identifiers to mitigate session hijacking. | 70.1 | User logs into the system. | System assigns a unique session identifier to each session to prevent hijacking. | Pass |
| 5.3.3 The system shall employ encryption to mitigate cookie theft. | 71.1 | User navigates within the system. | Cookies used in the system are secured and protected against theft through encryption | Pass |
| 5.3.4 The system shall detect and prevent session hijacking attempts. | 72.1 | Review system architecture and access protocols. | Algorithms are hosted on a dedicated server that is isolated from the main server and external access is restricted. | Pass |
| 5.3.5 The software shall conduct bi-weekly security audits. | 73.1 | Conduct security audits bi-weekly. | Regular bi-weekly security audits are conducted and documented. | Pass |
| 5.3.6 The software shall only allow registered and authorized users to view respective penetration test reports. | 74.1 | Attempt to access penetration test reports. | Only registered and authorized users can access the penetration test reports. | Pass |
| 5.3.7 The system shall log all unauthorized access attempts. | 75.1 | Simulate unauthorized access attempts. | All such attempts are logged and alerts are sent to administrators. | Pass |
| 5.3.8 User role 'Normal User' access shall be restricted to only their own data. | 76.1 | Check system functionalities across different user roles. | Functionalities are differentiated based on user roles; 'Normal Users' and 'Admins' have distinct permissions that are enforced. | Pass |
| 5.3.9 User role 'Admin' shall have access to all user data. | 77.1 | Access attempts by different user roles. | Access to sensitive data is properly restricted according to user roles, with controls in place to ensure compliance. | Pass |
| 5.4.1 The software should be able to accommodate changes in penetration testing methodologies. | 78.1 | Implement changes in testing methodologies. | The software adapts to new methodologies or threats with minimal need for modifications. | Pass |
| 5.4.2 The software should be able to accommodate emerging threats with minimal modifications. | 79.1 | New vulnerability information is released. | System includes testing for the new vulnerabilities. | Pass |
| 5.4.3 The system should be accessible 99.5% of the time. | 80.1 | Monitor system uptime over an extended period. | System availability is measured at 99.5% uptime. | Pass |
| 5.4.4 The system shall be able to integrate new tools or functionalities using RESTful APIs. | 81.1 | Integrate a new tool or functionality. | Integration is performed by including new RESTful API calls. | Pass |
| 5.4.5 The system shall be web based. | 82.1 | User attempts to access the system. | System is hosted on a web app. | Pass |
| 5.4.6 The system shall not crash during a penetration test, handling errors gracefully. | 83.1 | Attempt to start a penetration test. | Test completes without crashing. | Pass |
| 5.4.7 The system shall handle unexpected inputs or situations without failing. | 84.1 | Provide malicious inputs. | The system accepts input without crashing. | Pass |

# Traceability Matrix & Defect Tracking

## Traceability Matrix

The following table details the severity of each test case using the ratings described in Section 5.2, as well as listing the person responsible for ensuring the system meets that requirement.

| **Test Case ID** | **Severity** | **Responsible** |
| --- | --- | --- |
| 1.1 | Medium | Myles Scott |
| 2.1 | Medium | Calla Robinson |
| 3.1 | Low | Calla Robinson |
| 4.1 | Medium | Myles Scott |
| 5.1 | Medium | Caleb Hall |
| 6.1 | Low | Myles Scott |
| 7.1 | Medium | Myles Scott |
| 8.1 | Low | Calla Robinson |
| 9.1 | Medium | Myles Scott |
| 10.1 | Medium | Calla Robinson |
| 11.1 | Low | Myles Scott |
| 12.1 | Low | Caleb Hall |
| 13.1 | Medium | Caleb Hall |
| 14.1 | Medium | Myles Scott |
| 15.1 | Medium | Calla Robinson |
| 16.1 | Medium | Caleb Hall |
| 17.1 | Medium | Myles Scott |
| 18.1 | Medium | Myles Scott |
| 19.1 | Medium | Myles Scott |
| 20.1 | Medium | Michael Allen |
| 21.1 | Medium | Michael Allen |
| 22.1 | Medium | Myles Scott |
| 23.1 | Medium | Michael Allen |
| 24.1 | Critical | Caleb Hall |
| 25.1 | Critical | Caleb Hall |
| 26.1 | Critical | Calla Robinson |
| 27.1 | Medium | Caleb Hall |
| 28.1 | Critical | Caleb Hall |
| 29.1 | Medium | Michael Allen |
| 30.1 | Critical | Caleb Hall |
| 31.1 | Critical | Caleb Hall |
| 32.1 | Medium | Caleb Hall |
| 33.1 | Medium | Calla Robinson |
| 34.1 | Medium | Michael Allen |
| 35.1 | Medium | Michael Allen |
| 36.1 | Low | Michael Allen |
| 37.1 | Medium | Caleb Hall |
| 38.1 | Critical | Calla Robinson |
| 39.1 | Critical | Caleb Hall |
| 40.1 | Critical | Caleb Hall |
| 41.1 | Medium | Caleb Hall |
| 42.1 | Medium | Michael Allen |
| 43.1 | Low | Caleb Hall |
| 44.1 | Medium | Caleb Hall |
| 45.1 | Medium | Caleb Hall |
| 46.1 | Medium | Caleb Hall |
| 47.1 | Medium | Caleb Hall |
| 48.1 | Medium | Caleb Hall |
| 49.1 | Low | Michael Allen |
| 50.1 | Medium | Myles Scott |
| 51.1 | Medium | Myles Scott |
| 52.1 | Medium | Myles Scott |
| 53.1 | Medium | Calla Robinson |
| 54.1 | Low | Myles Scott |
| 55.1 | Medium | Calla Robinson |
| 56.1 | Medium | Myles Scott |
| 57.1 | Medium | Michael Allen |
| 58.1 | Low | Michael Allen |
| 59.1 | Medium | Myles Scott |
| 60.1 | Medium | Myles Scott |
| 61.1 | Low | Calla Robinson |
| 62.1 | Low | Michael Allen |
| 63.1 | Medium | Myles Scott |
| 64.1 | Low | Calla Robinson |
| 65.1 | Low | Michael Allen |
| 66.1 | Medium | Caleb Hall |
| 67.1 | Medium | Caleb Hall |
| 68.1 | Low | Myles Scott |
| 69.1 | Medium | Michael Allen |
| 70.1 | Medium | Michael Allen |
| 71.1 | Medium | Michael Allen |
| 72.1 | Medium | Michael Allen |
| 73.1 | Low | Michael Allen |
| 74.1 | Medium | Caleb Hall |
| 75.1 | Low | Calla Robinson |
| 76.1 | Medium | Calla Robinson |
| 77.1 | Medium | Calla Robinson |
| 78.1 | Medium | Caleb Hall |
| 79.1 | Medium | Caleb Hall |
| 80.1 | Critical | Michael Allen |
| 81.1 | Medium | Caleb Hall |
| 82.1 | Medium | Michael Allen |
| 83.1 | Critical | Caleb Hall |
| 84.1 | Critical | Caleb Hall |

## Defect Severity Definitions

The below table defines the severity of test cases. Each description provides the level of effort to fix the defect, as well as the affect the defect has on system operation. Severity is split into three categories, being: Critical, Medium, and Low.

| **Critical** | The defect causes a catastrophic or severe error that results in major problems and the functionality rendered is unavailable to the user. A manual procedure cannot be either implemented or a high effort is required to remedy the defect. Examples of a critical defect are as follows:   * System abends * Data cannot flow through a business function/lifecycle * Data is corrupted or cannot post to the database |
| --- | --- |
| **Medium** | The defect does not seriously impair system function and can be categorized as a medium defect. A manual procedure requiring medium effort can be implemented to remedy the defect. Examples of a medium defect are as follows:   * Form navigation is incorrect * Field labels are not consistent with global terminology |
| **Low** | The defect is cosmetic or has little to no impact on system functionality. A manual procedure requiring low effort can be implemented to remedy the defect. Examples of a low defect are as follows:   * Repositioning of fields on screens * Text font on reports is incorrect |

# Environment

## Environment

The testing environment details the systems used to test AutoPen. The testing environment is split into two categories: frontend testing and backend testing. Each refers to the system used to test the corresponding component of AutoPen. That is, the frontend testing details the system used when creating and testing the frontend of AutoPen. Backend testing refers to the system used to test Burp Suite and APIs. To ensure consistent testing, the tester’s system should meet the following minimum specifications and have the following software downloaded.

Backend testing system:

* HP laptop with an Intel® Core™ i5-1035G1 processor
* 12 gigabytes of RAM

Backend testing software:

* Kali Linux operating system
* Burp Suite Professional version >2023.10.2.3
* DVWA

Frontend testing system:

* MSI laptop with Intel® Core™ i5-10500H
* 32 gigabytes of RAM
* RTX 3060 GPU

Frontend testing software:

* Windows 11 operating system
* React version >18.0.0
* Python version >3.12.0
* Django version >5.0.4
* Visual Studio Community 2022

# Assumptions

AutoPen contains two main assumptions regarding the testing process.

* Assume that there is a possibility of not being able to do all testing on the required schedule, and attempt to prioritize testing actions.
* Assume that input to the system is not malicious.

# Risks and Contingencies

Below are the risks regarding testing AutoPen. For each possible risk, the impact to testing is listed along with the steps taken to remediate.

| ID | Risk | Description | Impact | Contingency plan |
| --- | --- | --- | --- | --- |
| 1 | Environment Instability | Test environments may not mimic production perfectly or may be unstable, leading to false positives or negatives. | High | Backup files on external device, reinstall software |
| 2 | Resource Unavailability | Key testing resources, both human and technical, might become unavailable during crucial testing phases. | High | Contact support |
| 3 | Data Integrity Issues | Test data might not be comprehensive or accurate, potentially leading to skewed results. | High | Restore from recent backup, review test data |
| 4 | Inadequate Coverage | Some critical scenarios or functionalities might be missed out during testing. | Medium | Remediate any findings brought to the team's attention |
| 5 | Software Dependencies | There might be external systems or applications that AutoPen depends on, and any instability in those can affect the testing. | Low | Find new library or create custom scripts |