A Midterm Progress Report

on

Cyber-Wise (A Vulnerability Scanner)

Submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

Computer Science and Engineering

SUBMITTED BY

Aditya Sankhyan	Shashank Sharma	Armaandeep Singh
CRN:2115004	CRN:2115165	CRN:2115021
URN:2104058	URN:2104187	URN:2104074

UNDER THE GUIDANCE OF

Prof. Amandeep Kaur Sohal

(April-2025)

Department of Computer Science and Engineering

GURU NANAK DEV ENGINEERING COLLEGE, LUDHIANA

INDEX

INTRODUCTION

In today's digital age, small businesses are increasingly vulnerable to cyber threats that can compromise sensitive data, damage their reputation, and result in financial losses. Many small business owners lack the technical expertise and resources to implement robust cybersecurity measures, leaving their websites exposed to risks such as data breaches, hacking, and fraud. As cyberattacks become more sophisticated, the need for accessible, affordable, and effective security solutions has never been more critical.

Cyberwise aims to address this gap by offering a comprehensive platform that combines essential cybersecurity features with user-friendly educational resources. This platform is designed to help small business owners secure their websites, protect sensitive data, and reduce the risk of cyberattacks, all while providing valuable cybersecurity education. Cyberwise empowers businesses to take control of their online security and build trust with their customers by offering simple, scalable security tools and educational materials.

The project focuses on three core objectives: developing and implementing fundamental security features for websites, creating a modular and scalable cybersecurity framework, and offering an accessible web-based scanner to identify vulnerabilities. Ultimately, Cyberwise strives to make cybersecurity both approachable and actionable for small business owners, ensuring they can operate confidently in an increasingly interconnected digital world.

Objectives

The objectives of Cyber-Wise are:

1. To Develop and Implement Core Security Modules for Foundational Functionality.
2. To Integrate and Thoroughly Test a Modular and Scalable Cybersecurity Framework.
3. To develop an accessible, user-friendly Web Cyber Scanner to help small owners safeguard sensitive data and build customer trust.

To accomplish our cybersecurity web application project, we will follow three key steps, aligned with the objectives outlined earlier:

1. Developing and Implementing Core Security Modules

(Objective: Develop and Implement Core Security Modules for Foundational Functionality)

The first step is to build the essential security features that will form the backbone of our cybersecurity platform. These include:

- IP Scanner: Identifies and analyzes IP addresses to detect potential threats.
- URL Scanner: Checks websites for malicious content, phishing risks, and blacklisting status.
- Encryption Module: Implements secure encryption techniques to protect sensitive user data.
- Vulnerability Scanner: Scans web applications for common security flaws and provides mitigation guidance.

By implementing these modules, we establish a strong foundational security framework, enabling users to effectively assess and protect their systems.

2. Integrating and Testing a Modular, Scalable Cybersecurity Framework

(Objective: Integrate and Thoroughly Test a Modular and Scalable Cybersecurity Framework)

To ensure that our cybersecurity platform is efficient, adaptable, and future-proof, we will:

- Develop a Modular System Architecture, allowing seamless integration of additional security tools in the future.
- Optimize System Scalability, ensuring it can efficiently handle increasing user requests and data loads.
- Integrate External Threat Intelligence Feeds to provide real-time cybersecurity insights.
- Perform Comprehensive Testing, including:
    - Unit Testing: Verifying individual security features.
    - Integration Testing: Ensuring all modules work cohesively.
    - Penetration Testing: Identifying and mitigating vulnerabilities.
    - Load Testing: Ensuring stability under high traffic conditions.

By following this step, we ensure that our platform is robust, scalable, and capable of addressing evolving cybersecurity challenges.

3. Creating an Accessible, User-Friendly Cyber Scanner for Small Businesses

(Objective: Develop an Accessible, User-Friendly Web Cyber Scanner to Help Small Business Owners Safeguard Sensitive Data and Build Customer Trust)

To make cybersecurity tools accessible and easy to use, we will:

- Design a Web Interface, allowing users to perform security scans effortlessly.
- Provide Actionable Security Insights, guiding users on how to mitigate identified risks.
- Publish Cybersecurity Blogs and Updates, educating users on best practices and emerging cyber threats.

By focusing on usability and accessibility, we empower small business owners with effective, easy-to-use cybersecurity tools, helping them safeguard their sensitive data and build customer trust.

SYSTEM REQUIREMENTS

Software Requirements:

Backend Requirements

- Programming Language: Node.js (Express)
- Web Server: Local Host, Vercel
- Security Libraries: OpenSSL (for encryption), OWASP ZAP (for vulnerability scanning), and Virustotal (for malicious scanning)

Frontend Requirements

- Frameworks: React.js, Next.js, Javascript
- CSS Frameworks: Tailwind CSS, Next UI
- JavaScript Libraries: Axios (for API requests), Chart.js (for data visualization)

Hardware Requirements:

Server-Side (Hosting Environment)

- Processor: Recommended 8-Core+ for high-performance scanning
- RAM: 8GB (Recommended) for handling concurrent scan requests
- Storage: 500GB SSD (Recommended) for storing scan results and logs
- Network: 1Gbps network speed (recommended) for efficient data retrieval and processing

Client-side (User Devices)

- Processor: Any modern multi-core CPU (Intel i5/i7 or AMD Ryzen 5/7).
- RAM: 4GB (Minimum), 8GB+ (Recommended) for smooth application performance.
- Browser Support: Google Chrome, Mozilla Firefox, Microsoft Edge (latest versions).
- Internet Connection: Minimum 10 Mbps for seamless interaction with the web interface.

SOFTWARE REQUIREMENT ANALYSIS

Problem Statement

In today's digital era, cybercrime is increasing at an alarming rate, posing significant threats to businesses of all sizes. While large enterprises invest heavily in cybersecurity infrastructure, small businesses often lack the resources, expertise, and awareness to implement robust security measures. As a result, they become easy targets for cyberattacks, leading to data breaches, financial losses, and reputational damage.

To address this issue, we are developing a web-based cybersecurity application that provides essential security tools such as IP scanning, URL scanning, encryption, and vulnerability assessment. This platform is designed to be user-friendly, accessible, and cost-effective, enabling small businesses to identify potential threats, secure their sensitive data, and enhance their cybersecurity posture without requiring extensive technical knowledge.

By bridging the cybersecurity gap for small businesses, our application aims to promote awareness, strengthen digital security, and build trust in the online ecosystem.

Models and Functionalities

Our cybersecurity web application is designed to provide essential security tools for small businesses to protect their digital assets. It includes multiple security modules that help users detect, analyze, and prevent cyber threats effectively. Below is an overview of the core modules and their functionalities:

1. IP Scanning Module

Functionality:

- Allows users to scan and analyze IP addresses to detect potential threats.
- Identifies blacklisted or suspicious IPs using external threat intelligence databases.
- Provides a detailed report on IP reputation and potential risks.

Use Case: Small businesses can use this feature to verify whether an IP address is safe before allowing access to their systems.

2. URL Scanning Module

Functionality:

- Analyzes URLs to detect phishing sites, malware, and blacklisted domains.
- Uses domain reputation checks and API integrations with cybersecurity databases.
- Provides a risk assessment score for the scanned URL.

Use Case: Businesses can check suspicious links in emails to avoid phishing attacks.

3. Encryption Module

Functionality:

- Provides a secure encryption and decryption tool for sensitive data.
- Supports AES and RSA encryption algorithms to ensure high security.
- Allows users to encrypt messages, files, or confidential data before sharing.

Use Case: Small businesses can secure confidential information before storing or sending it online.

4. Vulnerability Scanning Module

Functionality:

- Scans web applications for common security vulnerabilities such as:
    - SQL Injection
    - Cross-Site Scripting (XSS)
    - Weak Passwords and Authentication Issues
- Provides detailed reports and mitigation suggestions.

Use Case: Small businesses can identify weaknesses in their websites and fix them before hackers exploit them.

5. Security Blogs & Awareness Module

Functionality:

- Provides regular updates and blogs on cybersecurity threats, best practices, and prevention methods.
- Covers topics like ransomware protection, password security, and phishing prevention.
- Helps small businesses stay informed about the latest cyber threats.

Use Case: Users can access educational resources to improve their cybersecurity awareness.

SOFTWARE DESIGN

Overview

The software design of our cybersecurity web application follows a modular and scalable approach, ensuring that each feature functions independently while integrating seamlessly with external security libraries. The design focuses on efficiency, security, and ease of use, making it accessible for small businesses.

1. Architecture Overview

Our web application follows the client-server architecture, where:

- The front end interacts with users.
- The backend processes scan requests and communicate with external security libraries.
- There is no dedicated database, as we rely on security APIs and open-source threat intelligence sources for scanning.

2. Implemented Modules

IP Scanning Module(refer to figure 4.1)

- Accepts an IP address as input from the user.
- Sends a request to third-party security APIs (e.g., VirusTotal, AbuseIPDB) to check for threats.
- Displays a detailed security report based on API responses.

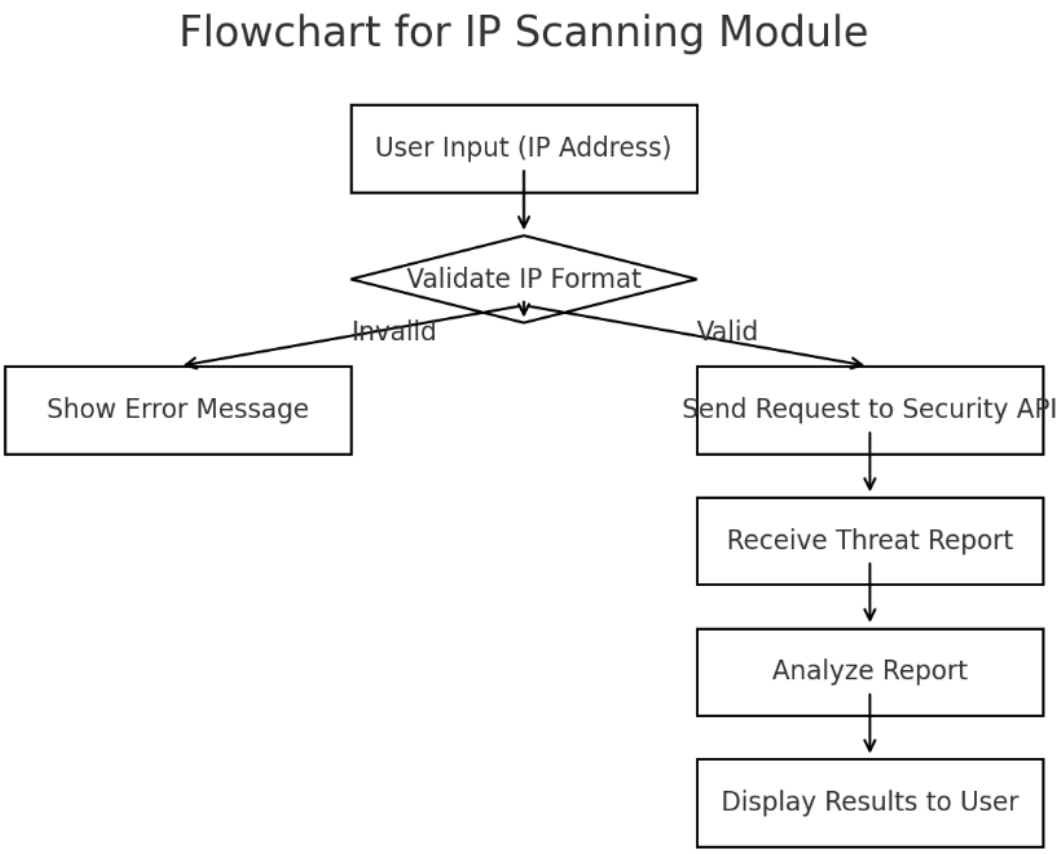

Flowchart for IP Scanning Module
User Input (IP Address)
Validate IP Format
Invalid
Valid
Show Error Message
Send Request to Security API
Receive Threat Report
Analyze Report
Display Results to User

Figure 4.1

Encryption & Decryption Module(refer to figure 4.2)

- Provides AES encryption to secure user-provided data.
- Ensures encrypted data is properly encoded and decoded without information loss.
- Allows users to encrypt and decrypt messages/files without requiring advanced cryptographic knowledge.

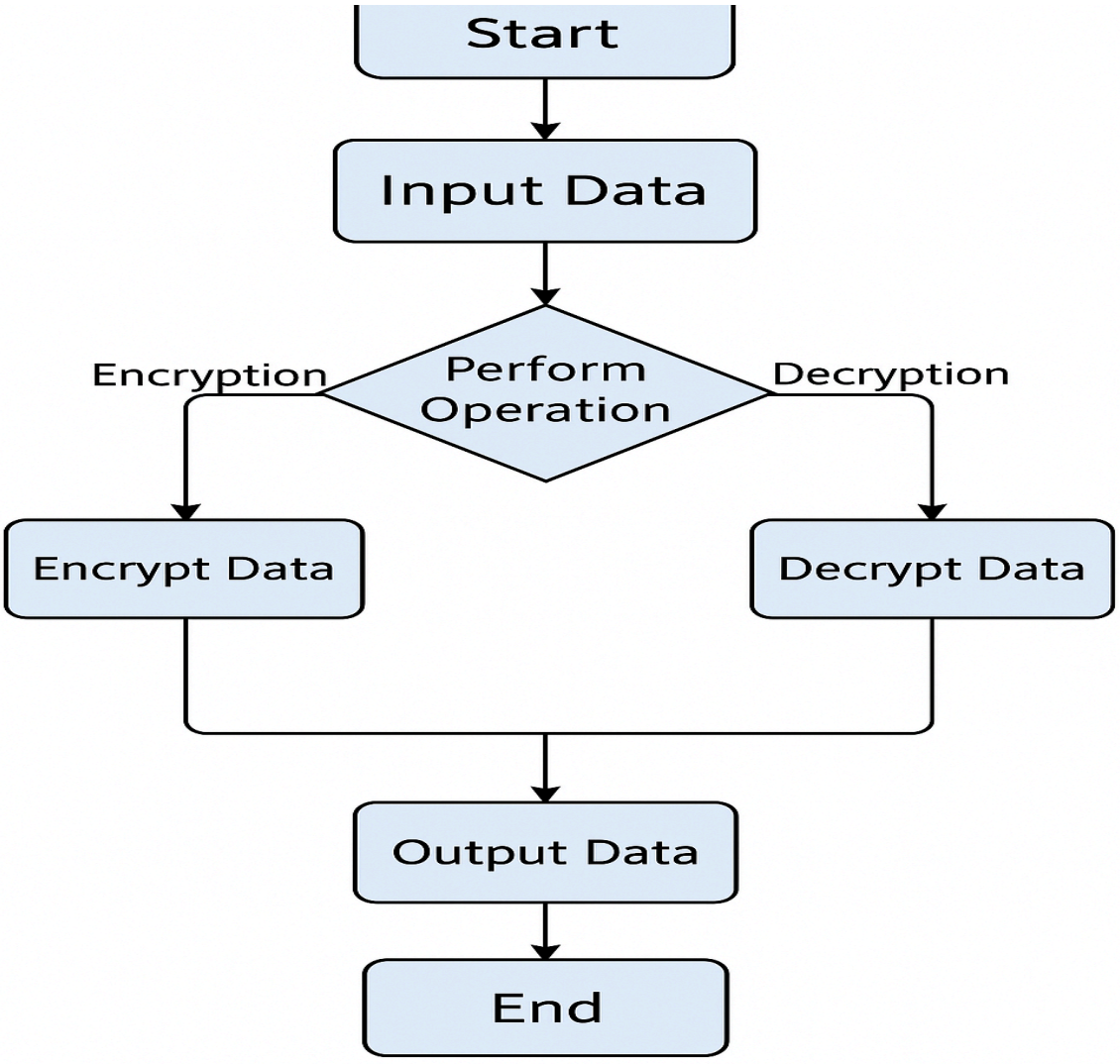


Figure 4.2

User Interface:

The primary goal of a user interface is to enable effective and efficient communication between the user and the system. This involves providing clear feedback, intuitive navigation, and accessible controls that allow users to perform tasks or access information easily (refer to figure 4.3).

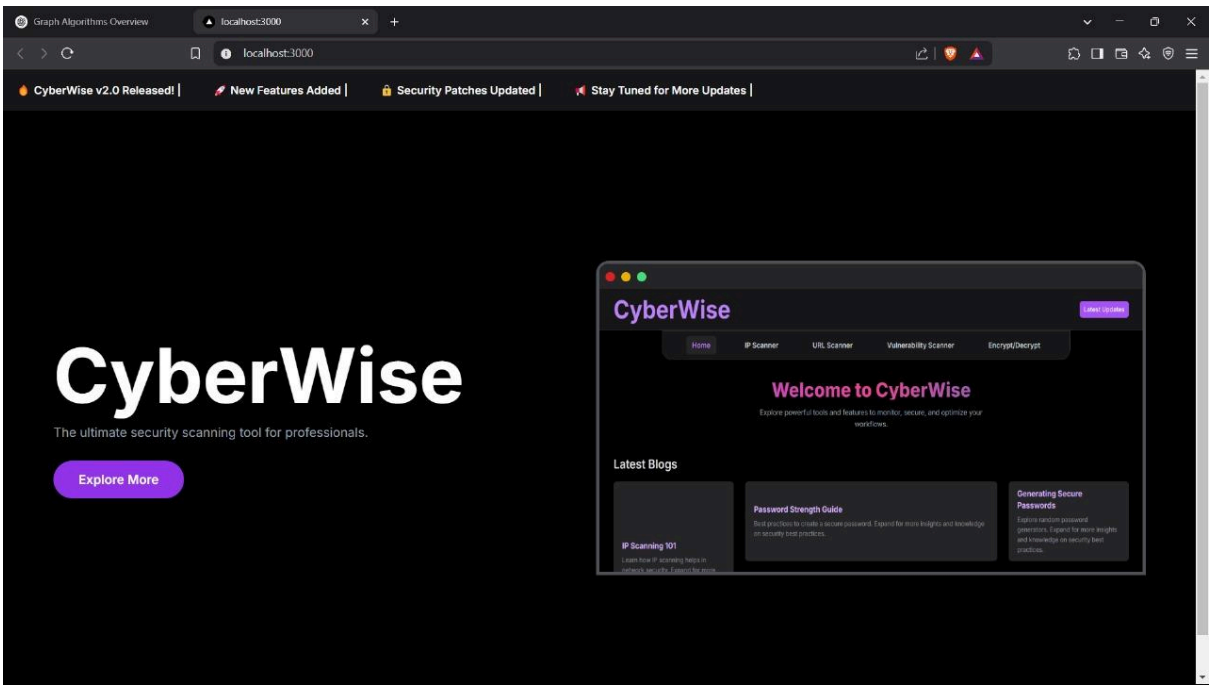


Figure 4.3

Testing Model

The testing model for our project includes the following key elements:

a) Functional Testing

- Verifies that each module (IP Scanner, URL Scanner, Encryption, Vulnerability Scanner, Blog Updates) works as intended (refer to figure 5.1).
- Ensures that security scans provide accurate and reliable results.

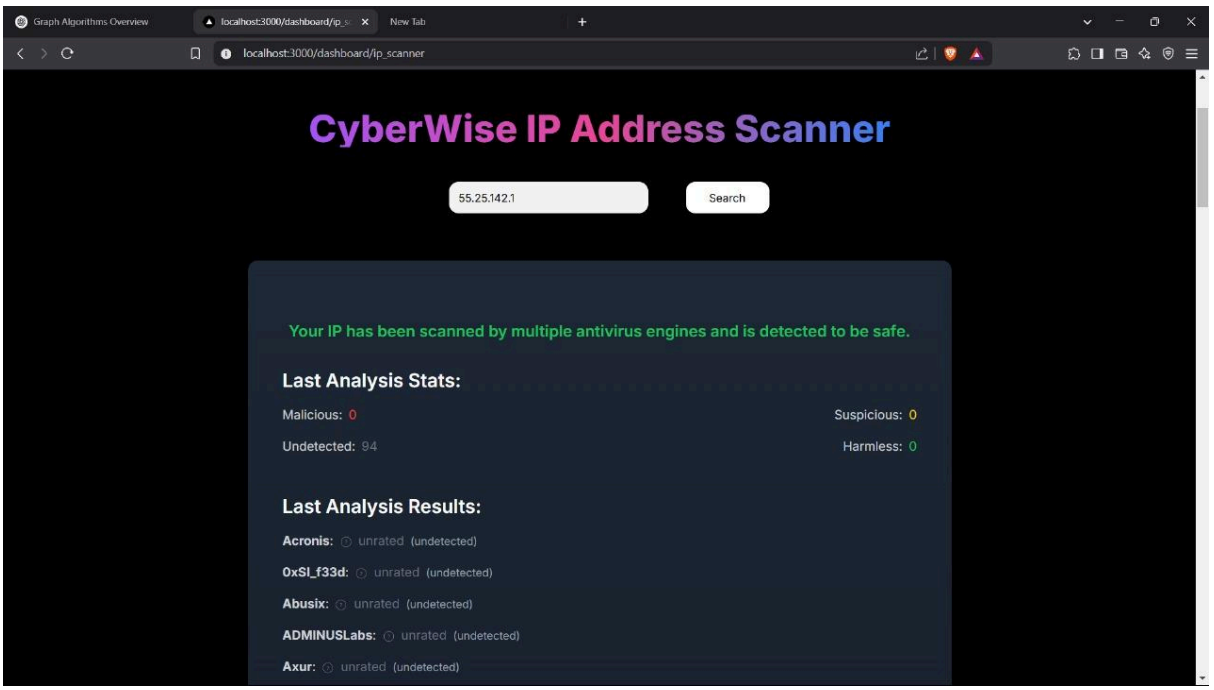


Figure 5.1

b) Performance Testing

- Measures the response time for different scan requests(refer to Figures 5.2 & 5.3).
- Evaluates system behavior under varying loads (multiple simultaneous users).

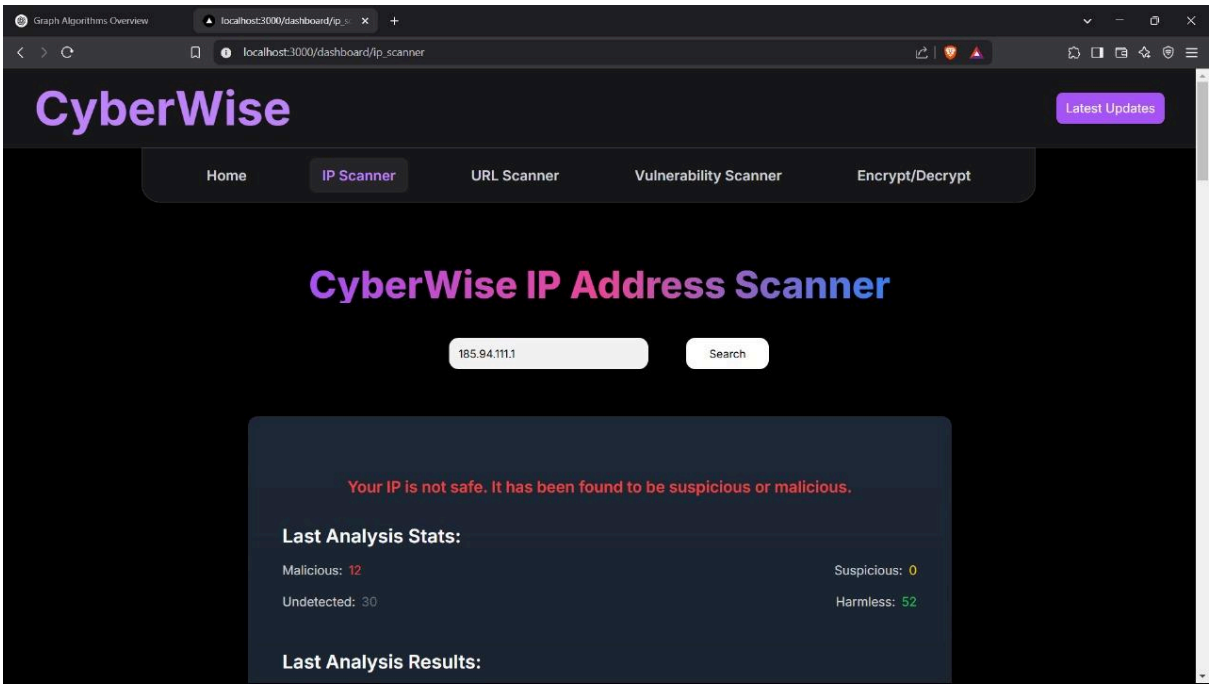


Figure 5.2

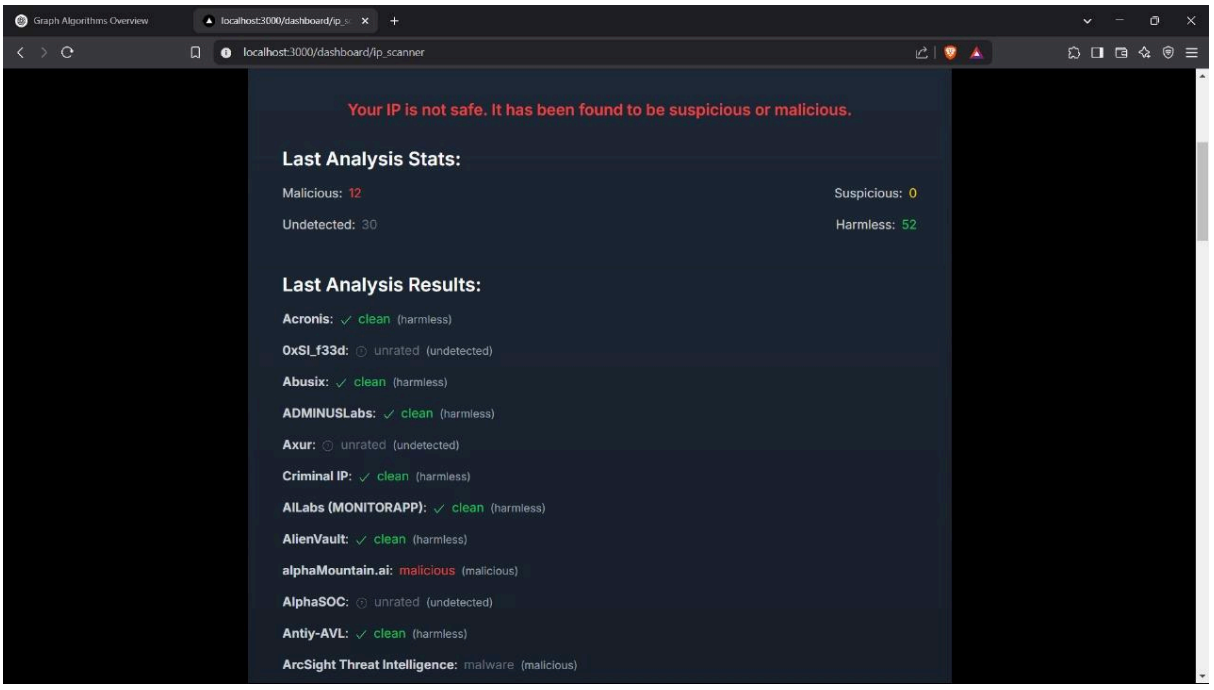


Figure 5.3

c) Security Testing

- Identifies vulnerabilities within the web application itself.
- Ensures data encryption mechanisms function correctly(refer to figure 5.4).
- Tests against SQL Injection, Cross-Site Scripting (XSS), and CSRF attacks.

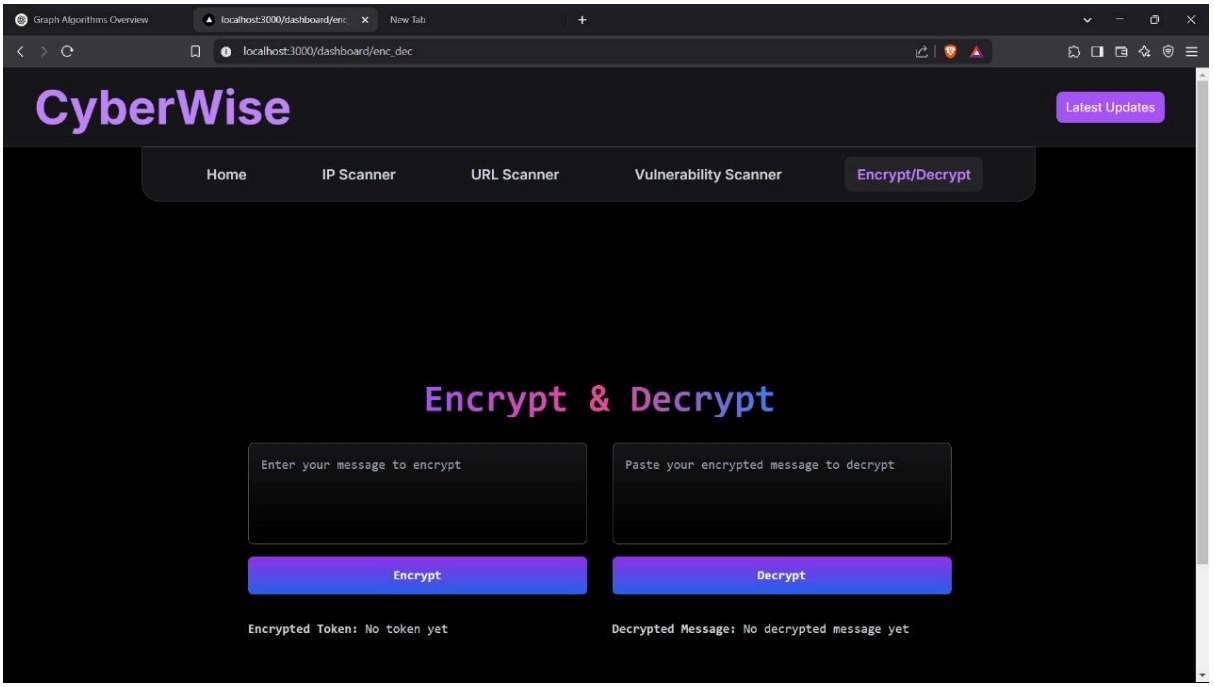


Figure 5.4

d) Usability Testing

- Evaluates the user interface for ease of use(refer to Figure 5.5).
- Check if security reports and recommendations are understandable for non-technical users.

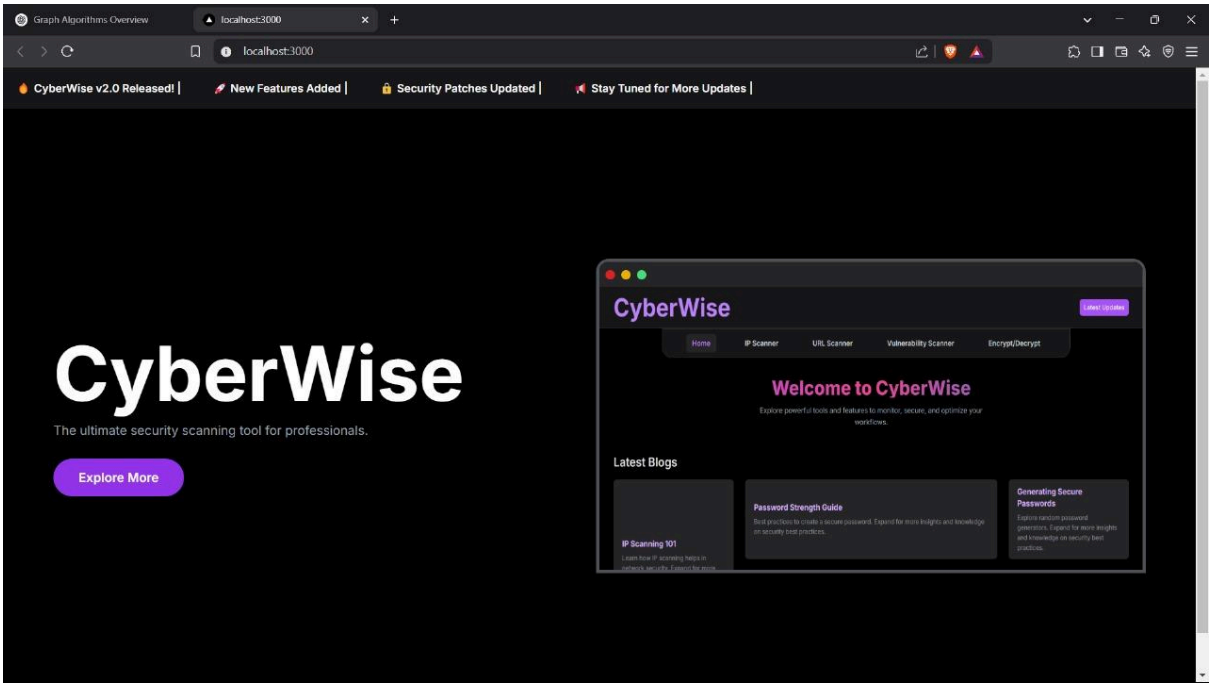


Figure 5.5

PERFORMANCE OF THE PROJECT DEVELOPMENT

The performance of our cybersecurity web application has been evaluated based on its functional modules, responsiveness, and accuracy of operations while running on a local server environment.

1. Current Functional Status

Successfully Implemented Features

- IP Scanning Module
    - The IP scanning functionality runs smoothly on the local server.
    - It provides accurate results by checking for blacklisted or suspicious IP addresses.
    - The response time is optimal, even with multiple scan requests.

- Encryption & Decryption Module
    - The encryption module is functioning correctly, allowing users to securely encrypt and decrypt data.
    - The system supports AES and RSA encryption algorithms, ensuring data protection.

Features Under Development

- URL Scanning Module
    - The module is currently in progress and being integrated with malicious URL detection databases.
    - Initial tests indicate the need for faster lookup mechanisms to improve response time.

- Vulnerability Scanning Module
    - The module is under development and undergoing testing for detecting common vulnerabilities such as SQL Injection and XSS.
    - Efforts are focused on refining the scanning algorithms for accurate detection.

2. Performance Observations

- Server Stability:
    - The web application runs smoothly on localhost without major crashes or downtime.
    - Backend processes handle requests efficiently, especially for IP scanning and encryption.

- Processing Speed:
    - IP scans are completed within 2-5 seconds, depending on the network response.
    - Encryption & decryption operations execute almost instantly.
    - URL and vulnerability scanning need further optimization to reduce processing delays.

- Resource Utilization:
    - CPU & Memory Usage remains stable, with no significant spikes observed during testing.
    - The system maintains performance even under multiple concurrent user requests.

OUTPUT SCREEN

This section describes the commands working inside the file - server

Server screen:

When the server is activated, it gives us details about all the connections established and the website is now running successfully (refer to Figure 7.1).

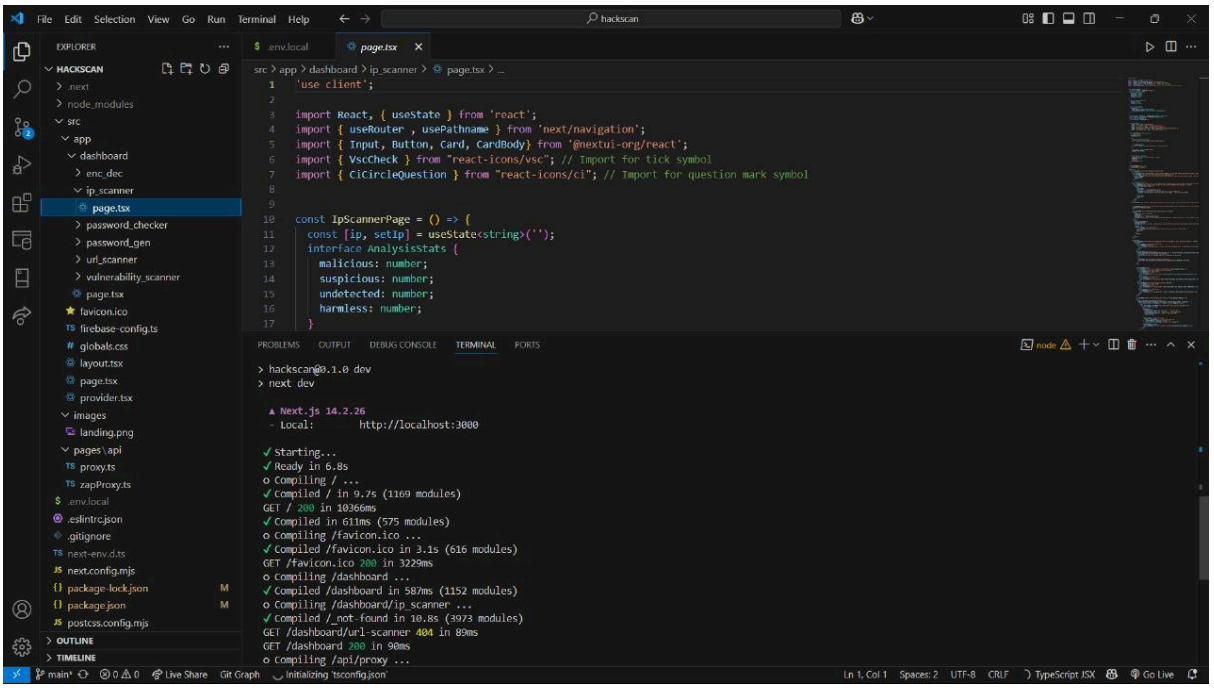


Figure 7.1

IP scan output:

The IP is scanning and responding properly(refer to figure 7.2 & 7.3).

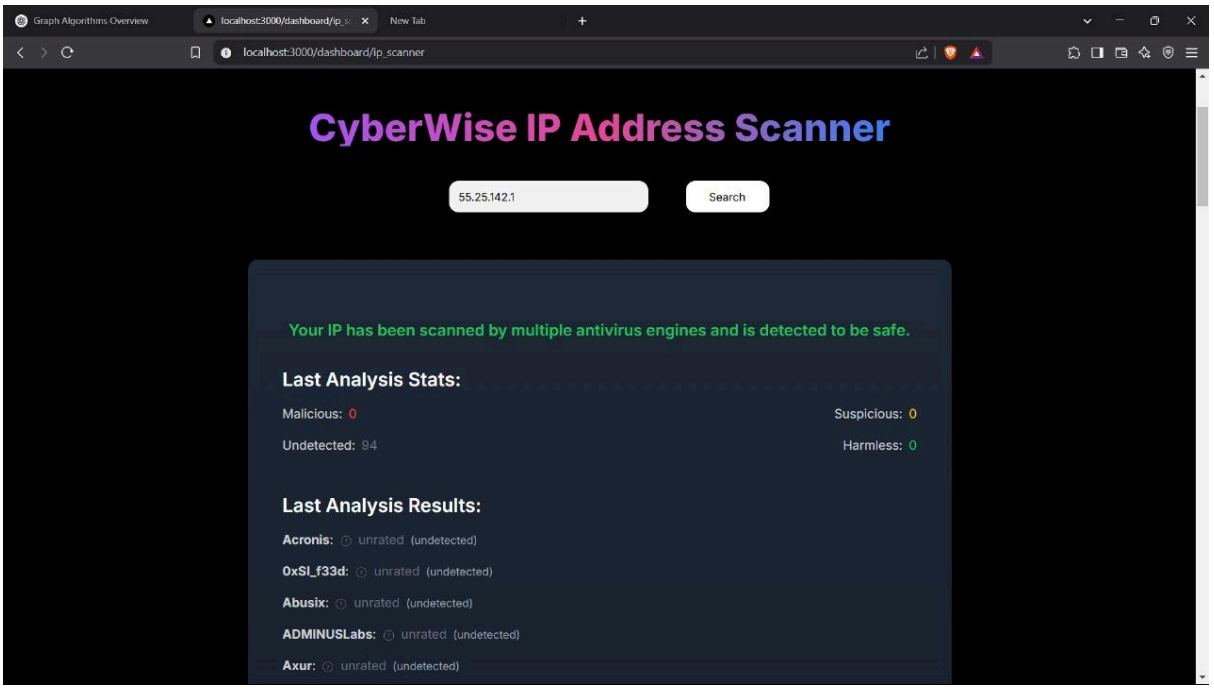
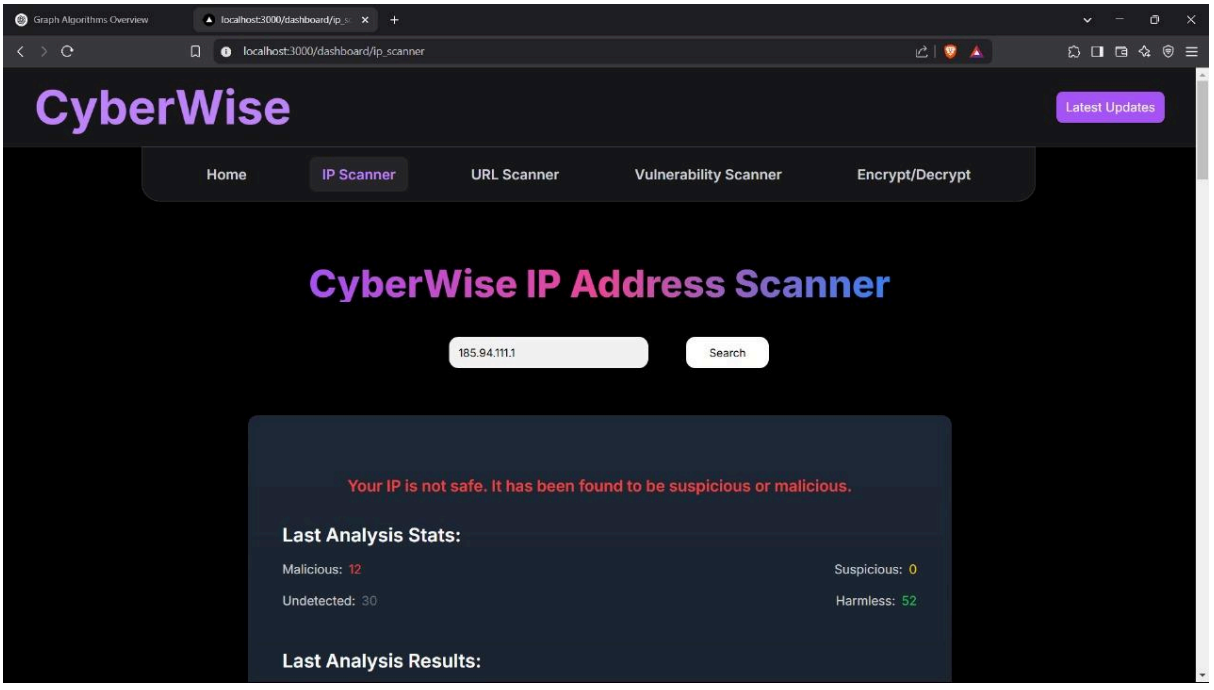


Figure 7.2(Non-malicious ip)



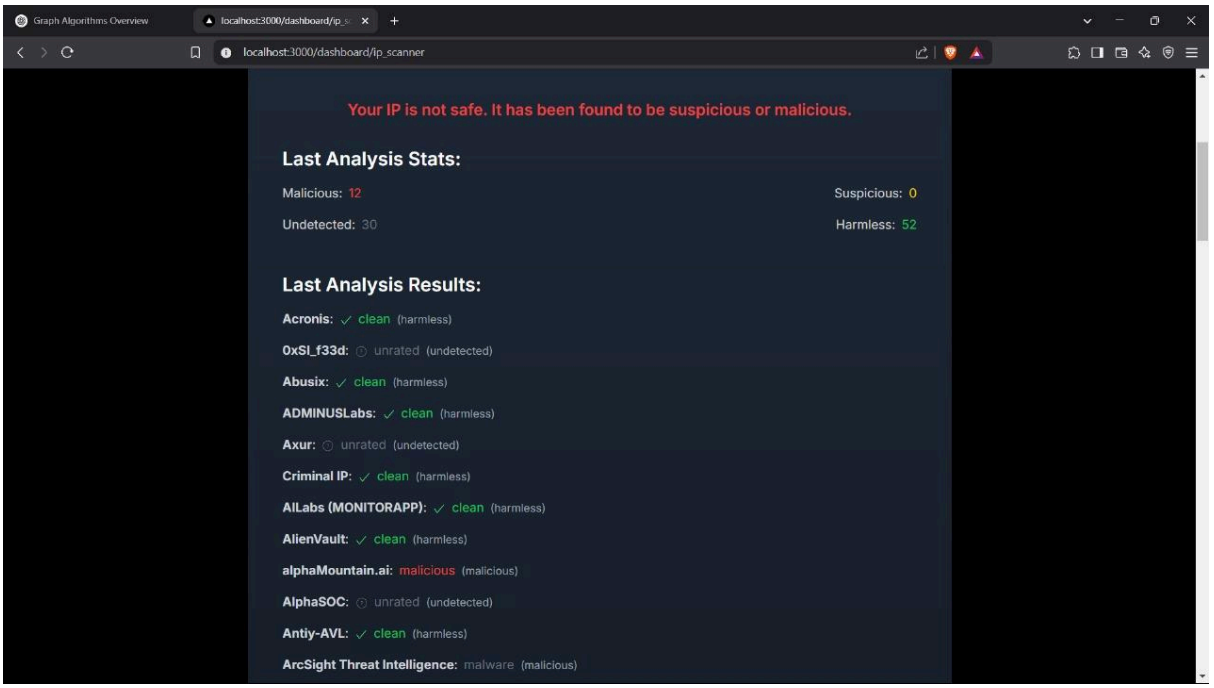


Figure 7.3(malicious ip)

Web screen output:

This is our web screen (refer to figure 7.4).

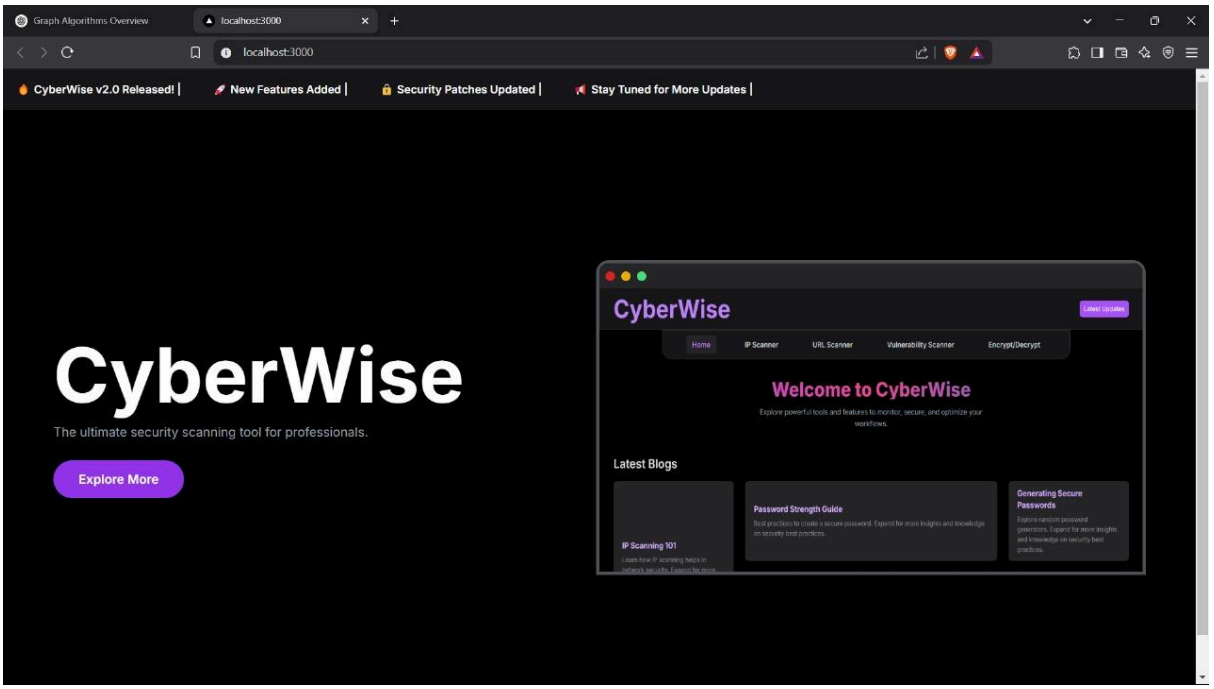


Figure 7.4

REFERENCES

- S. Gupta and R. Sandhu, “Vulnerability Scanners – A Proactive Approach to Assess Web Application Security,” *ResearchGate*, Mar. 2014. [Online]. Available: https://www.researchgate.net/publication/261182006_Vulnerability_Scanners-A_Proactive_Approach_To_Assess_Web_Application_Security. [Accessed: 31-Mar-2025].
- Smith and B. Johnson, “Cybersecurity Challenges and Solutions for Small Businesses,” *ResearchGate*, May 2023. [Online]. Available: https://www.researchgate.net/publication/380360965_Cybersecurity_Challenges_and_Solutions_for_Small_Businesses. [Accessed: 22-July-2024].
- S. B, S. NRK, T. J, and S. S, “A Comparative Analysis of Vulnerability Management Tools: Evaluating Nessus, Acunetix, and Nikto for Risk Based Security Solutions,” *arXiv*, Nov. 2024. [Online]. Available: https://arxiv.org/abs/2411.19123. [Accessed: 01-Apr-2024].