Capstone Project-II

Report

On

Cyber Safe -

A Phishing Detection and Web Safety Extension

Submitted By

Drashti Shah	202202626010086
Priyanshu Khambalkar	202302626020140
Kahan Velani	202302626020155
Dhruv Parmar	202302626020156

Under Guidance of

DR. Tejas Bhatt

Assistant Professor

Faculty of Engineering and Technology
GLS University

Academic Year
2024-2025

CERTIFICATE

This is to certify that the project report entitled **Cyber Safe - A Phishing Detection and Web Safety Extension** has been satisfactorily carried out by the following students

Drashti Shah	202202626010086
Priyanshu Khambalkar	202302626020140
Kahan Velani	202302626020155
Dhruv Parmar	202302626020156

under my guidance in the fulfilment of the course Capstone Project-II (2601606) work during the academic year 2024-2025.

Dr. Tejas Bhatt
Internal Guide
FET, CS&E
GLS University

Dr. Madhuri Chopade
Capstone Project-II Coordinator
FET, CS&E
GLS University

Acknowledgement

We would like to take this opportunity to sincerely thank all those who have supported and guided us throughout the successful completion of our project titled Cyber Safe – A Browser Extension for Phishing Detection.

We are deeply grateful to our project guide, Dr. Tejas Bhatt, for their continuous support, timely feedback, and expert guidance. Their encouragement and suggestions played a key role in helping us understand the core concepts and implement them effectively during each phase of the project.

We also extend our gratitude to the Faculty of Engineering and Technology, GLS University, and all the faculty members for providing us with a strong academic foundation and the resources required for the successful execution of this project.

We would like to thank the internal review panel and mentors for their valuable comments, which helped us identify key areas for improvement and fine-tune our solution.

We appreciate the institutional support provided through access to labs, internet facilities, and relevant tools and platforms necessary for development and testing.

Lastly, we acknowledge the collaborative effort of our team members. Each individual contributed with full dedication—from initial research and planning to development, documentation, and presentation—making this project a truly team-driven achievement.

Sincerely,

Drashti Shah
Priyanshu Khambalkar
Kahan Velani
Dhruv Parmar

Abstract

In today's digital era, phishing attacks have become one of the most common and dangerous cybersecurity threats faced by internet users. These attacks often trick users into sharing sensitive information such as login credentials, banking details, or personal data by mimicking legitimate websites. With the increasing number of such attacks, there is a growing need for effective and accessible solutions to protect users in real time while browsing the internet.

Our project, Cyber Safe, is a browser extension designed to help users identify and avoid phishing websites before any harm is done. The extension uses the Google Safe Browsing API, which maintains an up-to-date database of malicious URLs. When a user visits a website, the extension automatically scans the current URL and compares it against this database to detect any potential threats. In addition, it checks the SSL/TLS certificate of the website to ensure it is secure and legitimate.

If the website is found to be suspicious or unsafe, the extension displays an instant warning pop-up to alert the user. This alert also provides basic tips to help the user make safe browsing decisions. The interface is designed to be lightweight, user-friendly, and informative without disrupting the browsing experience.

Cyber Safe aims to enhance user awareness, prevent accidental data exposure, and contribute to safer internet usage. Future enhancements include mobile compatibility, user feedback reporting, detailed analytics for threat monitoring, and possible machine learning integration for improved detection accuracy.

This project demonstrates how simple, practical tools can be used to create meaningful impact in the field of cybersecurity by protecting users from real-world threats in real time.

Content

List of Figure

List of Table

Chapter 1: Introduction

1.1 Project Detail

Project Title: Cyber Safe – A Phishing Detection and Web Safety Extension

In today's digital world, where internet usage is a part of our daily routine, online threats like phishing attacks have become increasingly common. Phishing is a technique attackers use to trick users into providing sensitive information such as usernames, passwords, or credit card details by pretending to be a trustworthy source.

Our project, *Cyber Safe*, is a browser extension designed to help users stay safe while browsing the web. The extension automatically checks if a website is secure by using the Google Safe Browsing API, which maintains a constantly updated list of unsafe websites. If the user visits a site that is found to be suspicious or dangerous, the extension immediately alerts them with a warning message.

In addition to URL checking, the extension also verifies a website's SSL/TLS certificate, which helps confirm whether a website is authentic and secure. This provides an extra layer of protection, especially on websites that require personal or payment information.

We developed this project with the aim of making online browsing safer for everyone—whether they are students, professionals, or casual internet users. The extension is lightweight, easy to install, and requires minimal user interaction, making it convenient for daily use.

The extension is built with simplicity and usability in mind. It features a user-friendly interface with a pop-up that summarizes the website's safety status and offers practical tips for staying secure online. By minimizing the need for technical knowledge, *Cyber Safe* empowers all internet users—from students and professionals to elderly users and casual browsers—to protect themselves effortlessly.

Cyber Safe not only detects threats but also helps educate users by providing useful tips on identifying phishing sites. With cybercrimes on the rise, this tool empowers users with real-time protection and awareness, promoting safer internet habits.

1.2 Project Purpose

The purpose of the *Cyber Safe* browser extension project is to enhance online safety by preventing users from unknowingly accessing harmful or phishing websites. This tool is created with the intention of addressing real-world cybersecurity issues in a simple and accessible way. The key purposes of this project are:

1. Protect Users from Phishing Attacks:
   - To prevent users from falling victim to phishing attempts that aim to steal personal data, such as passwords, banking information, or login credentials.
2. Promote Safe Browsing Habits:

- To create awareness among users by providing real-time feedback and warnings about unsafe websites, helping them make better decisions while browsing.

3. Provide Real-Time Threat Detection:
   - To offer a quick and automatic analysis of any website the user visits, using trusted sources like the Google Safe Browsing API, and instantly warn users about potential threats.
4. Verify Website Legitimacy:
   - To help users identify whether a website is genuine or not by analysing the website's SSL/TLS certificate for legitimacy.
5. Eliminate the Need for Technical Knowledge:
   - To offer non-technical users an easy-to-use security tool that doesn't require them to understand complicated cybersecurity principles or take manual actions to detect threats.
6. Lightweight & User-Friendly:
   - To build a browser extension that runs smoothly in the background, uses minimal system resources, and provides security without interrupting the browsing experience.

1.3 Scope

The scope of the Cyber Safe project is to design and implement a browser extension that enhances online safety by detecting phishing websites in real time and alerting users before they fall victim. Below is a detailed breakdown of the scope:

- Real-Time URL Monitoring: Continuously checks the active tab's URL to detect phishing or unsafe links
- Google Safe Browsing API Integration: Uses Google's trusted API to verify URLs against a live database of known threats.
- SSL/TLS Certificate Validation: Inspects website security certificates to confirm legitimacy and encryption standards.
- Instant User Alerts: Displays warning pop-ups when suspicious or dangerous websites are detected.
- User-Friendly Interface: Designed to be lightweight, easy to use, and accessible for all users—technical or not.
- Cross-Browser Compatibility: Built using the Web Extension API model to work across Chrome, Firefox, and more.
- Desktop Support (Current Focus): Targeted at desktop browsers for now; mobile version planned in future updates.
- Educational Guidance: Provides users with tips and security advice to promote safe browsing habits.
- Future-Ready Architecture: Structured to allow future integration of machine learning, user reporting, and analytics.

1.4 Objective

The main objectives of the *Cyber Safe* browser extension project are:

1. Detect Phishing Websites in Real-Time:

- To identify malicious or deceptive websites instantly using the Google Safe Browsing API.

2. Validate Website SSL/TLS Certificates:
   - To analyse SSL certificate details and ensure the legitimacy and safety of websites.
3. Alert Users Proactively:
   - To notify users through a simple and clear pop-up whenever a threat is detected.
4. Create a User-Friendly Tool:
   - To design an intuitive interface that is easy to use and understand for all types of users.
5. Promote Safer Browsing Habits:
   - To help users become more aware of online threats by providing prevention tips along with alerts.
6. Ensure Cross-Browser Compatibility:
   - To build the extension so it functions seamlessly across different browsers using Web Extension standards.

1.5 Literature Review

Phishing has become one of the most common and dangerous threats in the field of cybersecurity. It involves tricking users into revealing sensitive information by mimicking trusted websites. Several research studies and cybersecurity organizations have highlighted the steady growth of phishing attacks in recent years.

According to the Anti-Phishing Working Group (APWG), phishing attacks have significantly increased over the last decade, reaching an all-time high in 2023 with over 4.7 million reported incidents globally. The rapid evolution of phishing techniques, such as fake login pages and spoofed URLs, makes it difficult for users to distinguish between real and fake websites.

Traditional detection methods, like blacklists and static filters, are often unable to detect newly created or zero-day phishing websites. In response, newer approaches such as API-based detection, SSL certificate validation, and real-time scanning have been introduced in recent tools and academic projects. APIs like Google Safe Browsing provide up-to-date information about known unsafe sites, making them an effective solution for real-time protection.

However, many of these solutions are either too complex for the average user or not integrated into a user-friendly browser environment. This highlights the need for lightweight and easily accessible tools that can provide instant alerts without requiring technical knowledge.

Figure 1.1 below shows the steady rise in phishing attacks from 2017 to 2023, emphasizing the growing need for solutions like Cyber Safe that provide proactive, real-time phishing detection and improve user awareness.

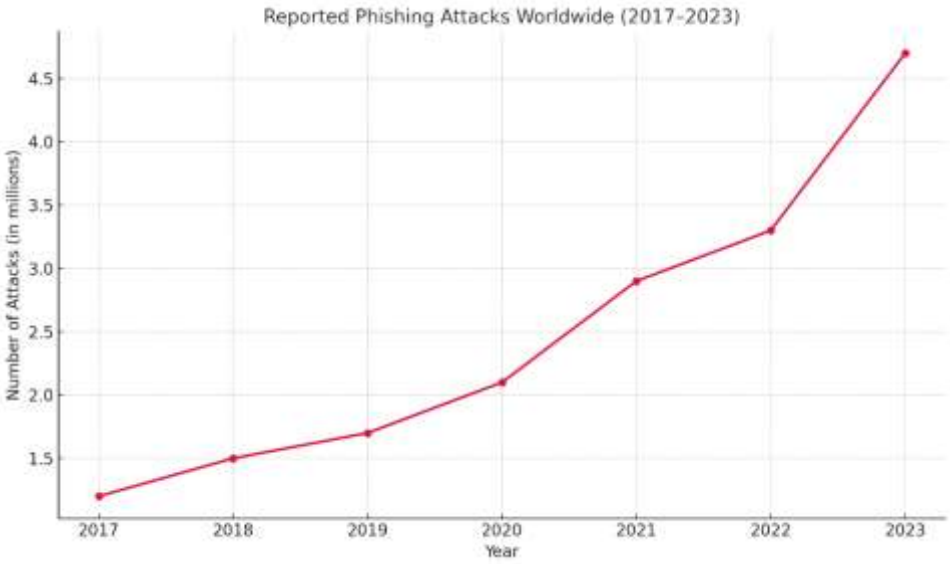

Figure 1.1 Rise in Phishing Attacks (2017–2023)

Chapter 2: About the system

2.1 Software Requirement Specification

2.1.1 Technical Requirement

Component	Details
Programming Language	Java script ,HTML ,CSS
API Used	Google Safe Browsing API, SSL/TLS Certificate Check
Browser Support	Chrome, Fire Fox, Edge, Brave, Opera
Development Tool	Visual Studio Code
Testing Tools	Browser Developer Tool

Table 2.1 Technical Requirement

2.1.2 Non-Technical Requirement

1. Internet Connectivity:
   A stable internet connection is required for the extension to communicate with external APIs and detect threats in real time.
2. Browser Compatibility:
   The extension should be used on supported browsers like Google Chrome or Mozilla Firefox to ensure full functionality.
3. Basic User Knowledge:
   Users should know how to install and use browser extensions. No advanced technical knowledge is required.
4. Accessibility**:**
   The system must be easy to understand and usable by non-technical users, with intuitive design and clear alert messages.
5. Device Requirements:
   Any desktop or laptop with a modern web browser can run the extension. No special system configurations are needed.
6. User Awareness:
   Users must understand that the extension scans URLs for security but does not store personal data or browsing history.

2.2 Project Planning

The development of the Cyber Safe browser extension was strategically planned and executed over two academic semesters to ensure a structured and efficient workflow. The planning process was divided into multiple stages, starting with requirement analysis and system design during Semester 5, followed by implementation, testing, and final deployment in Semester 6. Each phase was carefully scheduled to meet specific goals and deadlines, allowing for proper research, development, evaluation, and improvement. This systematic approach ensured that

the project progressed smoothly, met functional expectations, and aligned with academic objectives.

Semester 5: Documentation
Focus: Research, Requirement Analysis, Design, and Report Writing

Activity	Time Line
Problem Identification	Week 1-Week 2
Requirement Gathering	Week 3-Week 4
Literature Review	Week 5-Week 6
System Design (Diagram)	Week 7-Week 8
Writing Project Report Draft	Week 9-Week 12

Table 2.2 Documentation Time Line

Semester 6: Implementation
Focus: Development, Testing, and Final Submission

Activity	Time Line
Extension Development	Week 1-Week 4
API Integration & SSL check	Week 5-Week 6
Interface Design & Alert System	Week 7-Week 8
Testing & Debugging	Week 9-Week 10
Feedback & Improvement	Week 11
Final Report & Project Submission	Week 12

Table 2.3 Implementation Time Line

Chapter 3: Analysis of System

3.1 Use-case Diagram

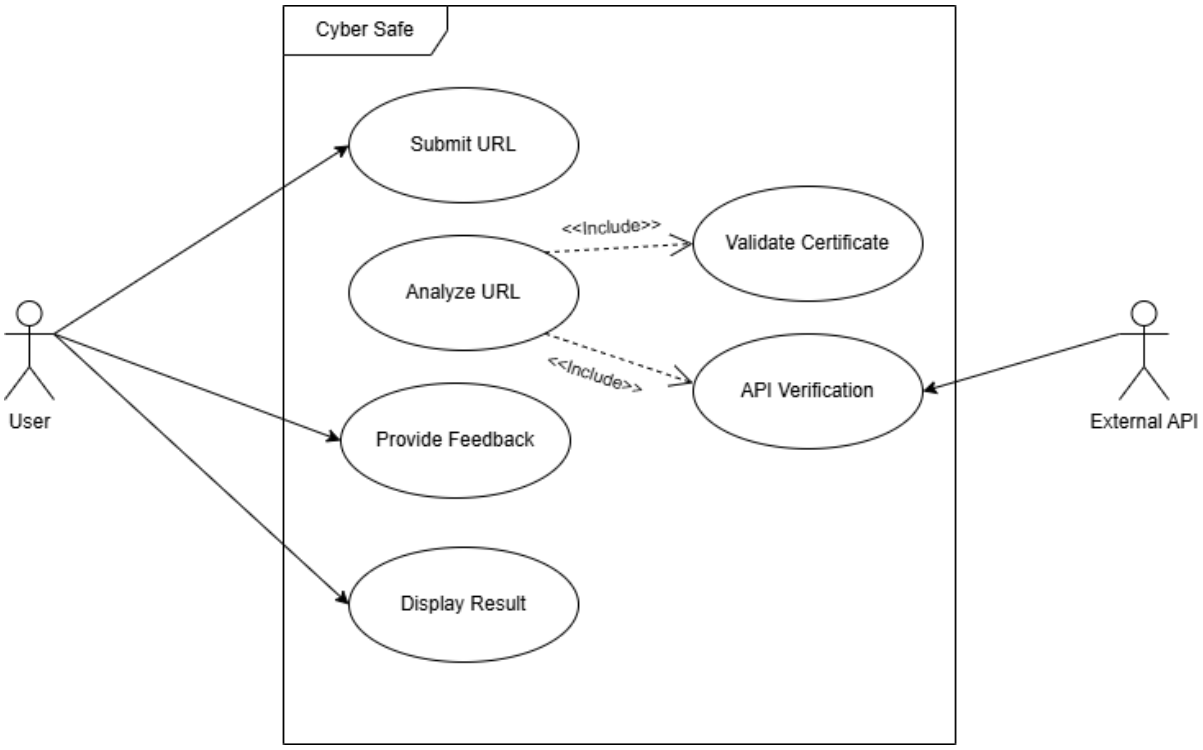

Figure 3.1 Use-case Diagram

3.2 Sequence Diagram

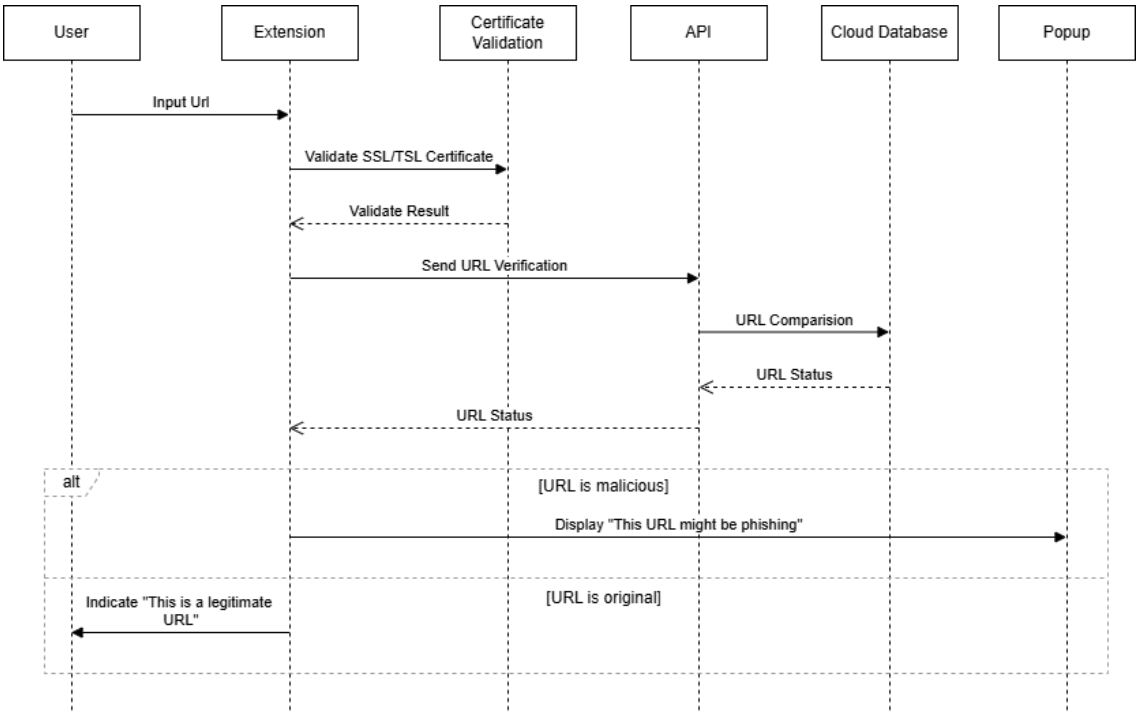

Figure 3.2 Sequence Diagram

3.3 Activity Diagram

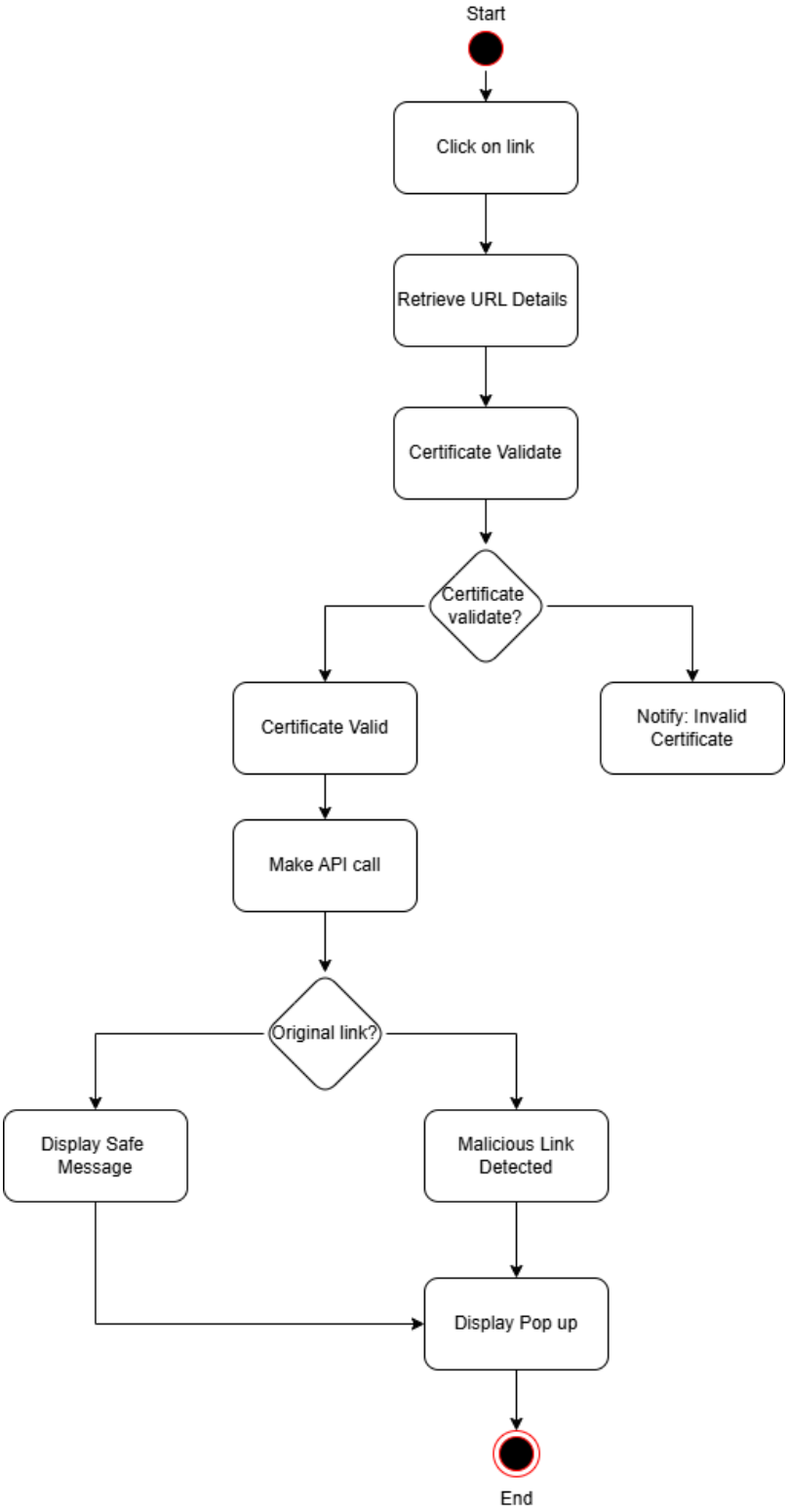

Figure 3.3 Activity Diagram

3.4 Data Flow Diagram

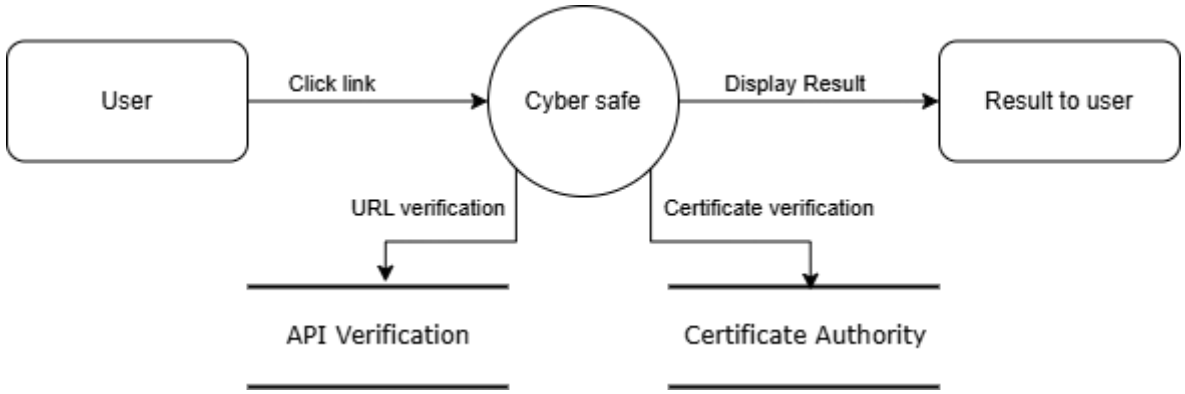

Figure 3.4 DFD Lv-0 Diagram

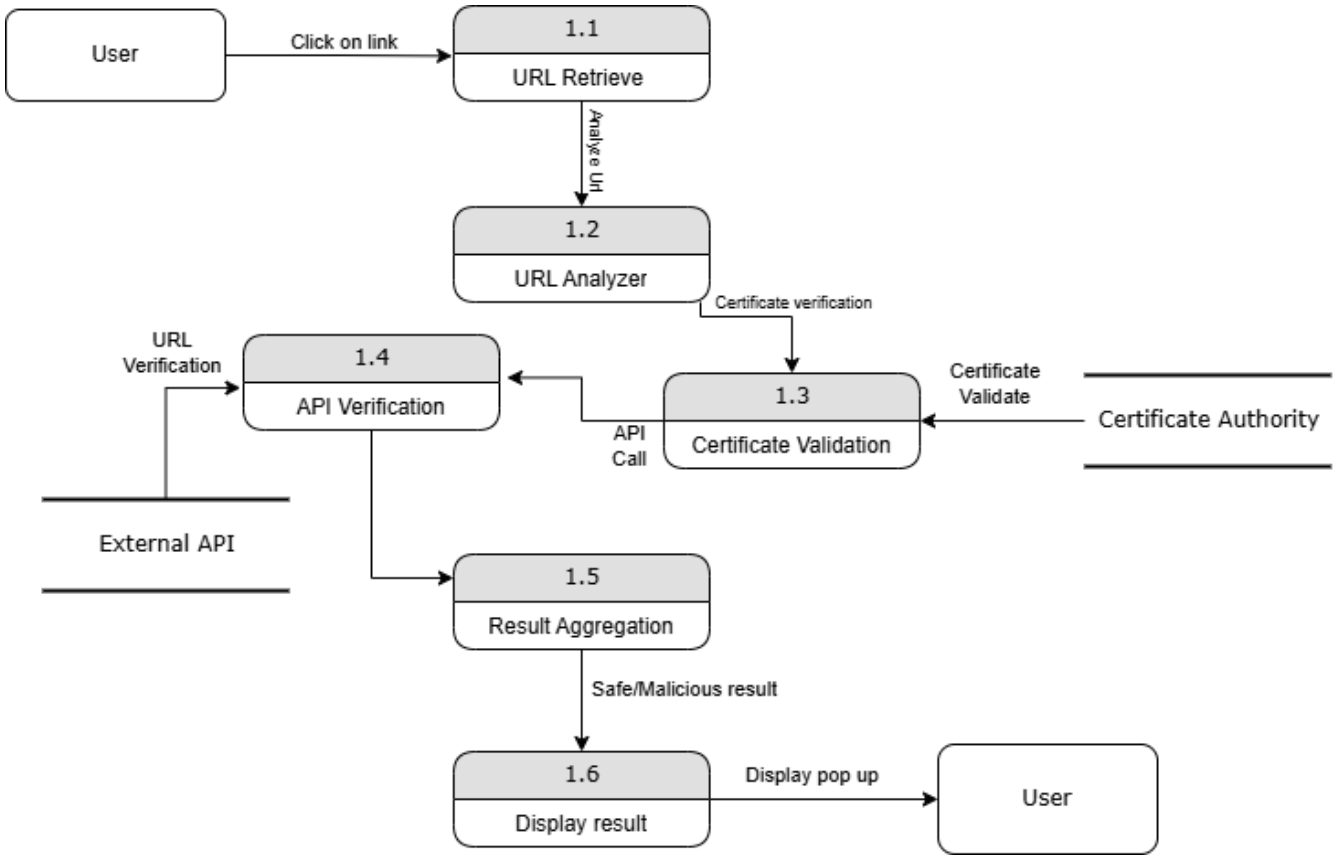

Figure 3.5 DFD Lv-1 Diagram

3.5 State Diagram

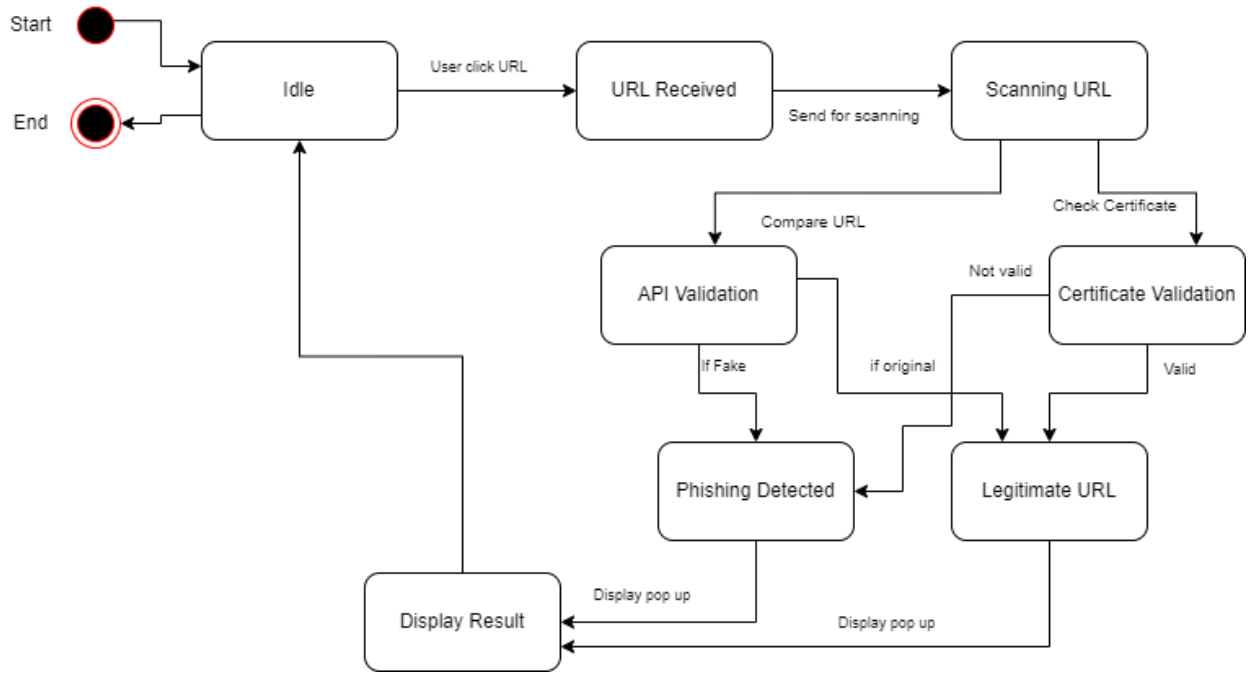

Figure 3.6 State Diagram

3.6 Class Diagram

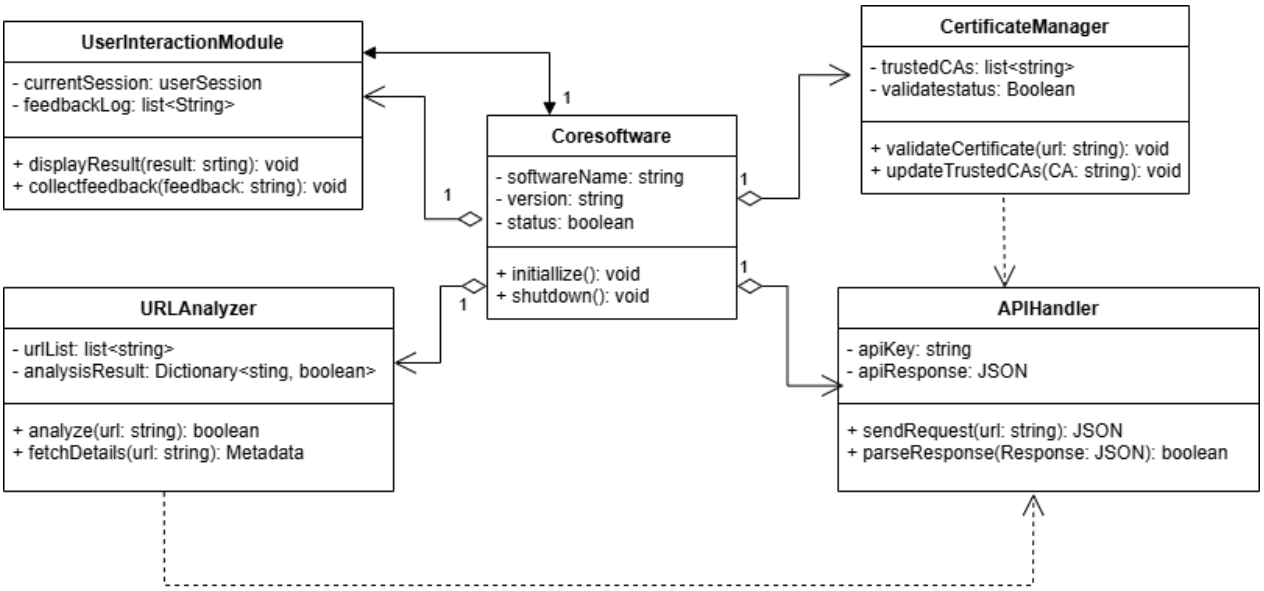

Figure 3.7 Class Diagram

Chapter 4: Design

4.1 System Flow Diagram

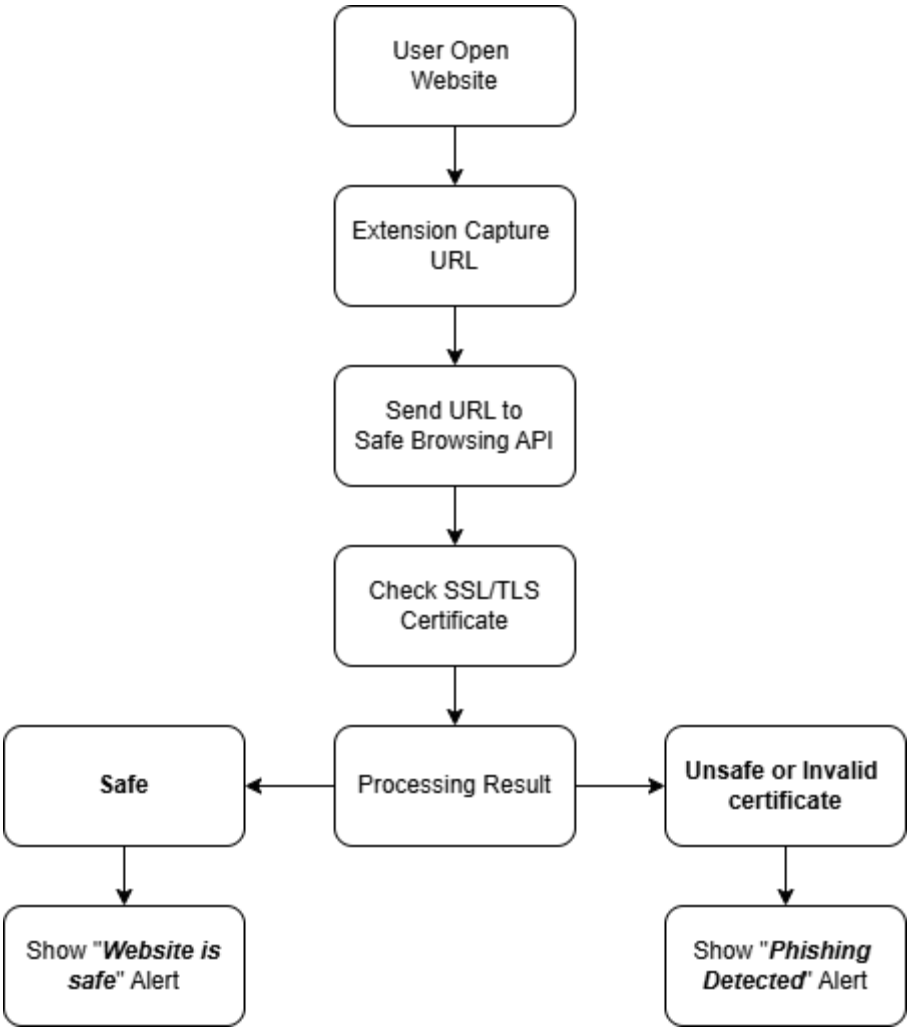

Figure 4.1 System Flow

- When a user opens a website, the browser extension captures the current URL and sends it to the Google Safe Browsing API. The API checks the URL against its database of known phishing and malicious sites. Simultaneously, the extension verifies the SSL/TLS certificate of the site. Based on the results, an appropriate alert or safe confirmation is displayed to the user.

4.2 Data Dictionary

Field Name	Description	Data Type	Source
url	The full URL of the current active tab being analyzed	String	Browser tab
checkedUrl	The URL checked via the Safe Browsing API	String	background.js (API Response)
isPhishing	Result of Google Safe Browsing API (true if URL is flagged)	Boolean	background.js (API Response)
issuer	Certificate issuer name	String	CertSpotter API
validFrom	Start date of certificate validity	String (Date)	CertSpotter API
validUntil	End date of certificate validity	String (Date)	CertSpotter API
subject	Subject details of the SSL certificate	String	CertSpotter API
signatureAlgorithm	Algorithm used to sign the certificate	String	CertSpotter API
alertMessage	Message shown in the UI (e.g., "Phishing Detected")	String	popup.js
certificateDetails	Combined certificate fields shown to the user	HTML String	popup.js
legitimateUrls	A predefined list of known safe/whitelisted URLs	Array[String]	popup.js
scanTimestamp	Timestamp when the URL was checked	Date/Time	System-generated (optional)

Table 4.1 Data dictionary

4.3 Relationship of table (From Data Base System)

Table Name	Primary Key	Fields
url_logs	log_id	url, isPhishing, scanTimestamp, certificateStatus
certificate_logs	cert_id	url, issuer, validFrom, validUntil, subject, signatureAlgorithm
user_alerts	alert_id	url, alertMessage, dateShown

Table 4.2 Relationship of table

4.4 User Interface

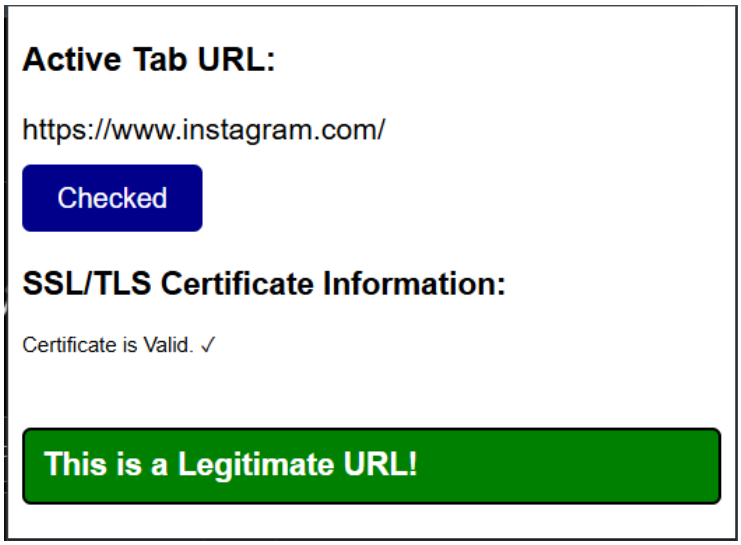

Figure 4.2 Cyber Safe interface when website is marked safe.

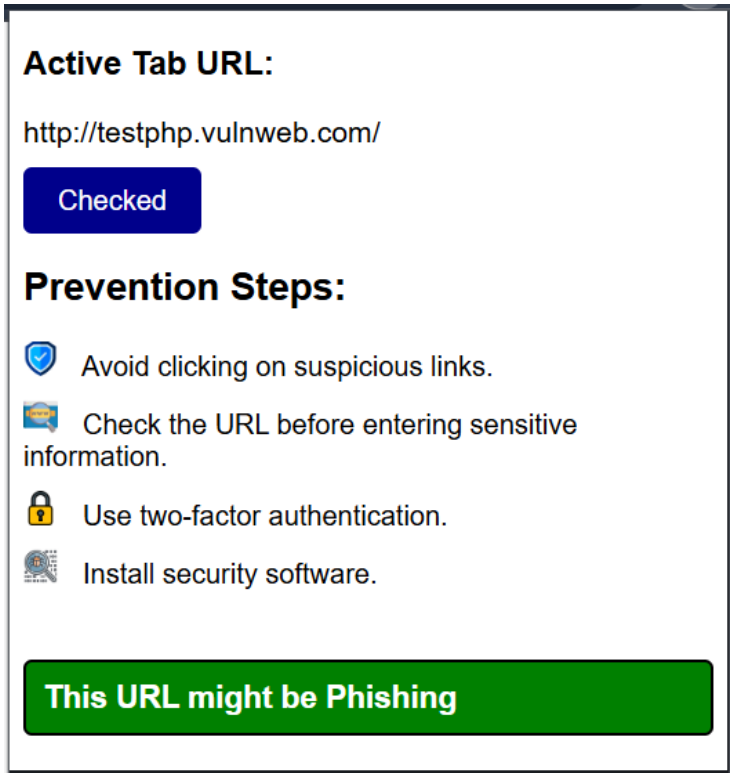

Figure 4.3 Cyber Safe popup showing a phishing warning message.

- When the extension detects a phishing website, a warning message is displayed in the popup along with advice such as "Avoid clicking on suspicious links." This helps users take immediate action without needing technical knowledge.

Chapter 5: Implementation

5.1 Implementation Environment

- The Cyber Safe browser extension was developed using a set of modern web technologies and tools. The development environment was chosen to ensure compatibility, efficiency, and ease of testing.
- Software Environment:
  - Programming Languages: JavaScript, HTML, CSS
  - Code Editor: Visual Studio Code
  - Browser Compatibility: Google Chrome, Mozilla Firefox (Web Extension compliant)
  - API Services:
    - Google Safe Browsing API (for phishing detection)
    - CertSpotter API (for SSL/TLS certificate verification)
  - Testing Tools: Postman, Chrome Developer Tools
- System Requirement:
  - Operating System: Windows 10 / 11 or Ubuntu (for testing)
  - RAM: Minimum 4GB
  - Internet: Required for real-time API calls and updates

5.2 Security Features

Security is a central aspect of the Cyber Safe project. The extension is designed with features that help detect and alert users about unsafe websites in real-time.

➢ Implemented Security Features:

- Phishing URL Detection:
  Uses the Google Safe Browsing API to check if the active tab's URL is reported as a phishing or malware site.
- SSL/TLS Certificate Verification:
  Verifies the legitimacy and trustworthiness of a website's SSL certificate using CertSpotter, ensuring encrypted and secure connections.
- Instant Pop-up Alerts:
  Displays clear and immediate visual warnings if a threat is detected, helping users avoid unsafe interactions.
- Whitelist Mechanism:
  Maintains a list of known legitimate URLs to avoid false positives and allow smooth browsing on trusted sites.
- No Data Storage:
  The extension does not store any personal user data, URLs, or browsing history, ensuring user privacy.

5.3 Coding Standard

- To ensure clarity, maintainability, and quality in the development of the Cyber Safe browser extension, the team followed a consistent set of coding standards across all source files. These standards helped streamline development and make the codebase easier to understand and extend.
  1. File Organization
     - Code was organized into separate files based on their functionality:
       - background.js: Handles core logic, API requests, and background event listeners.
       - manifest.json: Contains metadata, permissions, scripts, and configuration settings for the extension.
       - popup.js: Manages UI rendering, response handling, and dynamic updates.
       - popup.html:Defines the layout and structure of the extension's popup UI.
       - popup.css:Applies styling and visual formatting to the popup interface.
  2. Naming Convention
     - Variable and function names use camelCase for readability (e.g., checkUrlSafety(), getCertificateDetails()).
     - Constants are written in UPPERCASE (e.g., API_KEY).
     - Filenames are descriptive and relevant to their functionality.
  3. Commenting and Documentation
     - Each function includes a brief comment describing its purpose.
     - Inline comments were added where logic might not be immediately obvious.
     - API calls and event listeners are documented to explain data flow.
  4. Error Handling
     - try-catch blocks are used for API calls to catch failures gracefully.
     - Conditional checks are applied to prevent undefined or broken responses from affecting the user experience.

Chapter 6: Project Screen Shot

6.1 Prototype with result / Screen shot

6.1.1 Extension Interface – Safe Website

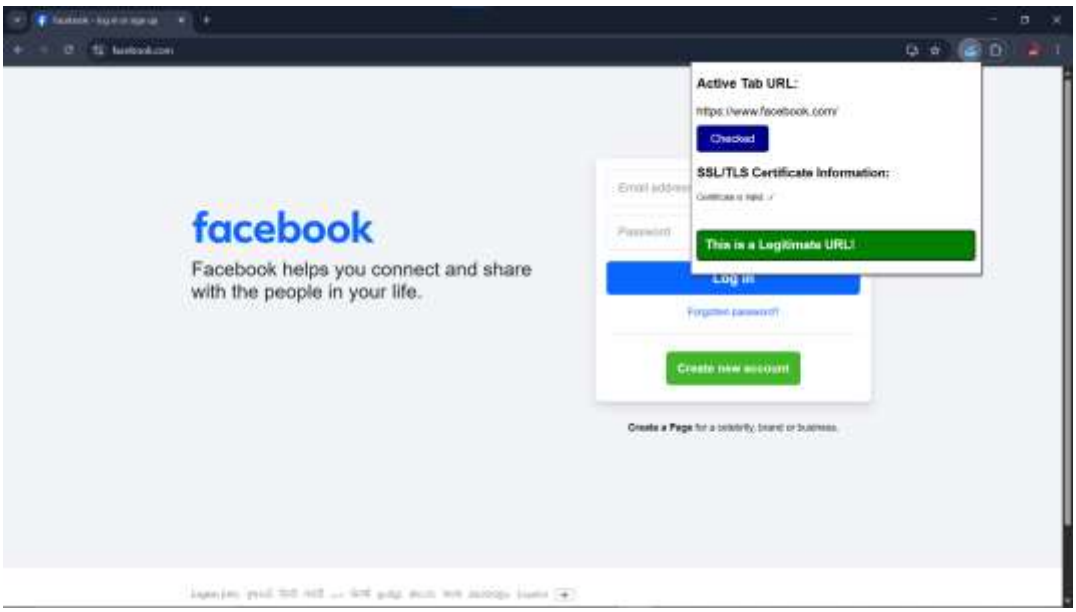

Figure 6.1 Extension popup when a safe website is accessed.

- When the user visits a legitimate and secure website, the extension displays a green-coloured status in the popup with the message "This Is a Legitimate URL." It also confirms that the SSL/TLS certificate is valid. This reassures the user that they are browsing securely.

6.1.2 Extension Interface – Phishing Detected

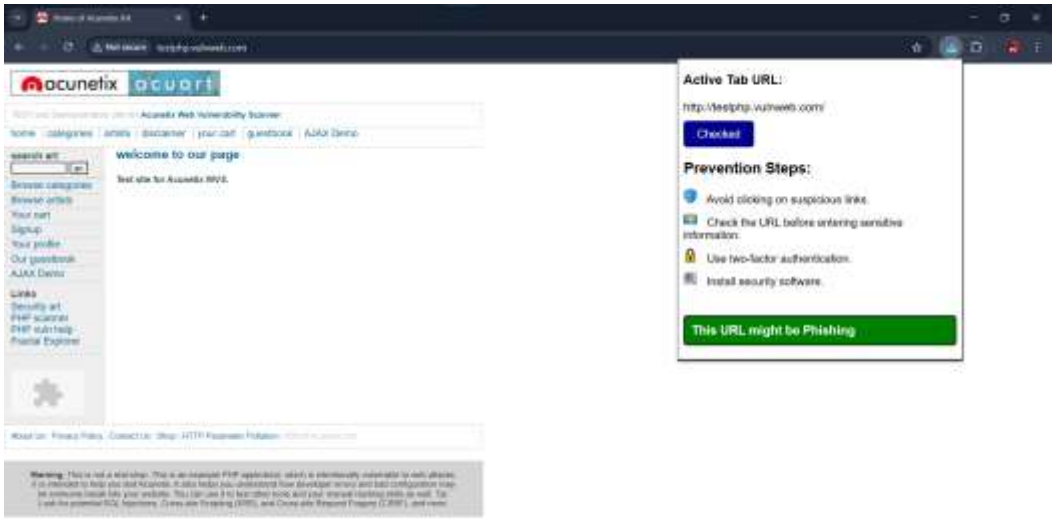

Figure 6.2 Alert pop-up when a phishing or unsafe website is detected.

- If the extension identifies a phishing threat using the Google Safe Browsing API or detects an invalid SSL certificate, it displays a red warning popup with the message "Phishing Detected – Proceed with Caution." This prevents users from accidentally sharing sensitive information.

Chapter 7: Conclusion & Future Work

7.1 Conclusion

The *Cyber Safe* project was developed to address the growing need for real-time protection against phishing attacks. With the increasing use of online platforms for personal, educational, and financial activities, users are more exposed than ever to deceptive websites designed to steal sensitive information.

Our browser extension offers a simple yet effective solution by using the Google Safe Browsing API to detect harmful websites and alert users instantly. It also verifies the website's SSL/TLS certificate to help users confirm if the site is secure and trustworthy. The extension is designed to be lightweight, user-friendly, and accessible to both technical and non-technical users.

Throughout this project, we gained a deeper understanding of web security, browser extension development, API integration, and user interface design. The successful implementation of *Cyber Safe* demonstrates how a small tool can make a significant difference in improving internet safety and user awareness.

This project not only helped us apply technical concepts in a real-world scenario but also contributed to the larger goal of promoting safe browsing practices. While the current version achieves its core purpose, there is room for further improvement and expansion in the future.

7.2 Future work

- While the current version of *Cyber Safe* is functional and effective, there is potential for future enhancement and scalability. Some proposed improvements include:
  1. Machine Learning Integration
     - Incorporate AI models to detect phishing attempts based on visual patterns, URL structure, and site behaviour—allowing for more adaptive and intelligent detection.
  2. Mobile Browser Support
     - Extend the extension's compatibility to mobile browsers, ensuring security on smartphones and tablets where phishing risks are also increasing.
  3. User Reporting Feature
     - Allow users to manually report suspicious websites. This feedback can be stored for analysis or used to improve detection rates.
  4. Admin Dashboard & Analytics
     - Build a backend system where developers/admins can monitor phishing trends, track usage statistics, and review user reports for continuous improvement.

Reference

Google Developers. (n.d.). *Safe Browsing APIs (v4)*. Retrieved from: https://developers.google.com/safe-browsing

APWG. (2023). *Phishing Activity Trends Report*. Anti-Phishing Working Group. Retrieved from: https://apwg.org/trendsreports/

Mozilla Developer Network (MDN). (n.d.). *WebExtensions API Documentation*. Retrieved from: https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions

Zhang, Y., Hong, J. I., & Cranor, L. F. (2007). *CANTINA: A Content-Based Approach to Detecting Phishing Websites*. Proceedings of the 16th International Conference on World Wide Web (WWW '07). DOI: 10.1145/1242572.1242659

Verma, R., & Dyer, K. P. (2015). *On the Character of Phishing URLs: Accurate and Robust Statistical Learning Classifiers*. Proceedings of CODASPY 2015. DOI: 10.1145/2699026.2699100

Kumaraguru, P., Rhee, Y., Acquisti, A., Cranor, L. F., Hong, J., & Nunge, E. (2007). *Protecting people from phishing: The design and evaluation of an embedded training email system*. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.

Open Web Application Security Project (OWASP). (n.d.). *Phishing*. Retrieved from: https://owasp.org/www-community/attacks/Phishing

Symantec. (2019). *Internet Security Threat Report*. Volume 24. Retrieved from: https://www.symantec.com/security-center

CertSpotter by SSLMate. (n.d.). *Certificate Transparency Log Monitoring API*. Retrieved from: https://sslmate.com/certspotter/

Statista. (2023). *Number of phishing sites detected worldwide from 2015 to 2023*. Retrieved from: https://www.statista.com

W3C. (n.d.). *HTML & CSS Standards and Best Practices*. Retrieved from: https://www.w3.org/

Wichers, D. (2017). *OWASP Top 10 - 2017: The Ten Most Critical Web Application Security Risks*. OWASP Foundation. Retrieved from: https://owasp.org/www-project-top-ten/