DSN2098 - Project Exhibition 1

A PROPOSED DESIGN AND IMPLEMENTATION OF PHISHGUARD AI BROWSER EXTENSION FOR PHISHING DETECTION

Project/Research of B.Tech

Supervised By:
Dr. Irfan Alam, Assistant Professor
SCAI, VIT Bhopal University

Presented By:
24BCY10104, Tejas Santosh Paithankar
24BCY10410, Sudhanshu Singh
24BCY10218, Ashwin C
24BCY10182, Aashish Kumar Singh
24BCY10293, Niyati Agarwal

Outline

- Work Distribution
- Work Plan
- Introduction
- Literature Review
- Research Gap Summarisation
- Problem Statement
- Methodology
- Implementation
- Results
- Conclusion
- References
- Appendix –1
- Appendix –2

Dr. Irfan Alam, VIT Bhopal University (Tejas Santosh Paithankar, Sudhanshu Singh, Ashwin C, Aashish Kumar Singh, Niyati Agarwal)

Work Distribution

- Reg No :24BCY10104 - Tejas Santosh Paithankar - Team Lead.
- Reg No :24BCY10410 - Sudhanshu Singh - Frontend
- Reg No :24BCY10218 - Ashwin C - Backend
- Reg No :24BCY10182 - Aashish Kumar Singh - Backend
- Reg No :24BCY10104 - Niyati Agarwal - Documentation

Work Plan

- Review 1 31/07/2025
  - Discussion on the problem statement
  - Work plan implementation
- Review 2 31/8/2025
  - Completion of 25% of our prototype
- Review 3 23/9/2025
  - Completion of 100% of our prototype

Introduction

- **Real-time Protection:** Detects and blocks phishing attempts instantly while users browse.
- **AI-Powered Detection:** Uses machine learning models to identify suspicious web pages and fraudulent patterns.
- **Domain Reputation Analysis:** Cross-checks visited websites against trusted reputation databases.
- **SSL Certificate Verification:** Ensures websites have valid and secure encryption before allowing transactions.
- **Seamless Integration:** Functions as a lightweight browser extension requiring minimal setup.
- **User-Centric Security:** Provides clear alerts and guidance to help users avoid online threats.

Literature Review

1. Khonji, M., Iraqi, Y., & Jones, A. (2013). Phishing detection: A literature survey.
IEEE Communications Surveys & Tutorials, 15(4), 2091-2121.

- About:
  - Comprehensive survey of phishing mitigation techniques including detection, offensive defense, correction, and prevention.
  - Focuses on how detection fits within the overall phishing mitigation process.
  - Highlights the complexity of phishing attacks and the human factor as a vulnerability.
- Advantages:
  - Provides a high-level classification of phishing detection approaches: blacklist-based, heuristics, visual similarity, and data mining techniques.
  - Emphasizes promising results from machine learning-based techniques with higher accuracy.
  - Covers broad spectrum of related methods and categorizes them thoughtfully.

Literature Review

- Disadvantages:
  - No single solution effectively mitigates all phishing vulnerabilities due to evolving tactics.
  - Dependent techniques like blacklists are often ineffective against zero-day phishing attacks.
  - Many methods suffer from false positives and are limited by feature sets or datasets used.

- Research Gap:
  - Need for robust, accurate detection methods that handle zero-hour phishing attacks with low false positives.
  - Combining machine learning with heuristic and data-driven approaches for improved detection speed and reliability.
  - Addressing human factors and user awareness to complement technical detection.

Literature Review

2.Garera, S., Provos, N., Chew, M., & Rubin,A. D. (2007). A framework for detection and measurement of phishing attacks. Proceedings of the 2007 ACM Workshop on Recurring Malcode, 1-8.

- About:
  - Proposes a phishing detection framework analyzing the structure of phishing URLs without requiring webpage content.
  - Uses fine-grained heuristics applied in a logistic regression model for classification.
  - Achieves high accuracy in detecting phishing URLs by focusing on URL obfuscation and other URL features.
- Advantages:
  - High classification accuracy of about 97.3% with low false positive and high true positive rates.
  - Detects phishing URLs using only URL information, making it efficient and scalable.
  - Large-scale measurement study of phishing prevalence based on real-world URL data.

Literature Review

- Disadvantages:
    - Focuses mainly on URL-based detection, may miss attacks that do not rely on URL obfuscation.
    - Cannot fully detect phishing attacks that use legitimate or less-obfuscated URLs.
    - Reliance on heuristics and URL features may limit adaptability to evolving phishing tactics.

- Research Gap:
    - Need to integrate URL analysis with additional detection layers for improved robustness.
    - Exploration of real-time adaptive methods that can keep up with evolving phishing techniques.
    - Expanding beyond URLs to include page content and user behavior for holistic phishing detection.

Literature Review

3. Zhang, Y., Hong, J. I., & Cranor, L. F. (2017). Cantina: A content-based approach to detecting phishing web sites. Proceedings of the 16th International Conference on World Wide Web, 639-648.

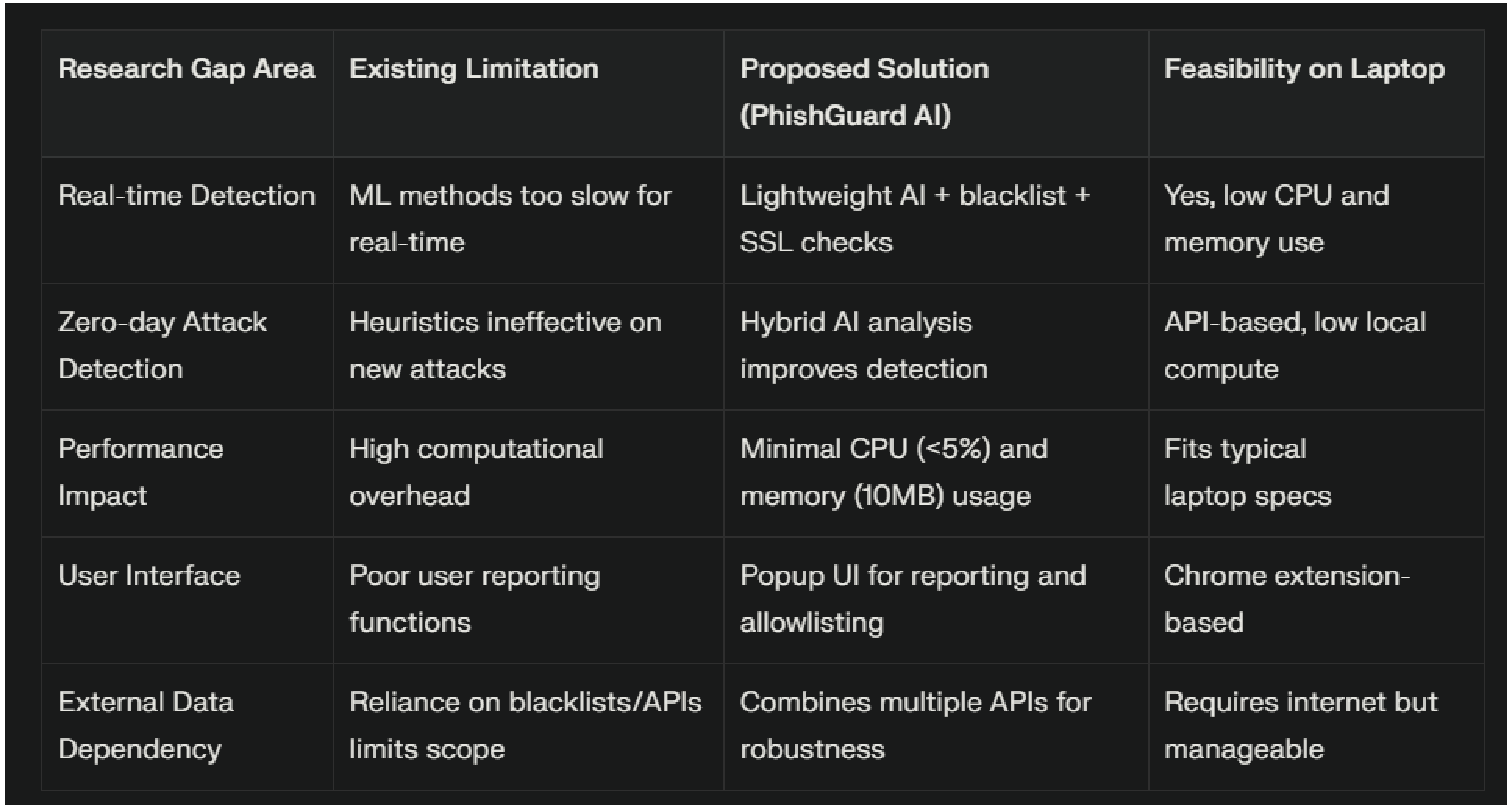
- About:
    - Cantina uses content-based analysis employing TF-IDF (Term Frequency-Inverse Document Frequency) to extract the most relevant terms from a webpage.
    - It generates a lexical signature from top TF-IDF terms and queries search engines to compare results with the webpage domain.
    - The approach relies on the assumption that legitimate sites have content closely related to their domain, whereas phishing sites do not.
- Advantages:
    - Effective at detecting phishing websites by analyzing content semantics rather than only URLs.
    - Uses search engine results to verify site legitimacy, improving accuracy over pure heuristic or URL-based methods.
    - Reduces false positives compared to many blacklists or URL-only approaches.

Literature Review

- Disadvantages:
  - Depends on availability and response time of search engines which can affect detection speed.
  - Less effective against phishing pages with well-created legitimate-like content.
  - Requires internet connectivity for real-time search queries.

- Research Gap:
  - Enhancing detection without relying heavily on external search engines for quicker offline detection.
  - Dealing with advanced phishing sites that mimic content of legitimate sites closely.
  - Integrating user behavior analytics and multi-factor detection for holistic phishing defense.

Research Gap Summarisation



Research Gap Area	Existing Limitation	Proposed Solution (PhishGuard AI)	Feasibility on Laptop
Real-time Detection	ML methods too slow for real-time	Lightweight AI + blacklist + SSL checks	Yes, low CPU and memory use
Zero-day Attack Detection	Heuristics ineffective on new attacks	Hybrid AI analysis improves detection	API-based, low local compute
Performance Impact	High computational overhead	Minimal CPU (<5%) and memory (10MB) usage	Fits typical laptop specs
User Interface	Poor user reporting functions	Popup UI for reporting and allowlisting	Chrome extension-based
External Data Dependency	Reliance on blacklists/APIs limits scope	Combines multiple APIs for robustness	Requires internet but manageable

Problem Statement

Current phishing detection methods either lack real-time accuracy or impose heavy performance overhead, making them impractical for seamless user protection.

Main Topic (Project Title):

PhishGuard AI: Design and Implementation of a Lightweight AI-Powered Browser Extension for Real-Time Phishing Detection

Methodology

Method-1:

Khonji et al. (2013) reviewed machinelearning techniques for phishing detection, noting high accuracy but slow processing for real-time applications.

Method-2:

Khonji et al. (2013) reviewed machinelearning techniques for phishing detection, noting high accuracy but slow processing for real-time applications.

Method-3:

Zhang et al. (2017) proposed a hybrid modelcombining blacklists and machine

Methodology

Step-by-Step Process:

1. Requirement Analysis
   - Identify hardware/software and API needs
   - Set performance and security goals
2. Design Architecture
   - Modular design: background worker, content scripts, UI
   - Integrate domain reputation, AI URL analysis, and SSL checks
3. Development
   - Implement real-time URL monitoring
   - Develop popup interface for alerts and reporting
   - Integrate external APIs for phishing detection
4. Testing & Validation
   - Test with phishing and safe URLs
   - Measure accuracy, performance, and user experience

Methodology

- Deployment & Feedback
  - Release extension for user adoption
  - Collect feedback, monitor detection effectiveness
- Maintenance & Improvements
  - Address API dependency, browser limits
  - Add multi-browser support and advanced heuristics

Implementation and Specifications

Frameworks:

- Chrome Extension Framework (Manifest Version 3)
- JavaScript (background.js, content.js, popup.js)
- APIs: APIVoid (Blacklist checks), Gemini API (AI-based URL analysis)

Implementation Steps:

1. Real-time URL monitoring using Chrome background service worker
2. Blacklist verification via APIVoid API
3. AI-based phishing detection through Gemini API URL analysis
4. SSL certificate verification for accessed websites
5. Blocking phishing URLs and displaying warning pages
6. Popup interface for user interaction: reporting, allowlist management
7. Performance tuning to keep CPU usage below 5% and memory usage around 10MB

Implementation and Specifications

Next Task: Reproduction & Improvement

- Reproduce Base Article Results:
  - Implement the PhishGuard AI system on your machine
  - Validate key performance metrics: accuracy, detection time, CPU and memory usage
- Innovate & Improve:
  - Apply new formulas, algorithms, or methods outlined in the methodology
  - Generate improved results across performance metrics
  - Compare improvements with base results to assess effectiveness
- Compare & Discuss:
  - Draw comparative graphs of old vs new results
  - Analyze improvements in accuracy, latency, and resource usage in the results section

Implementation and Specifications

Next Task: Reproduce and Improve Base Article Results

- Reproduce base article results on your own machine using original methodology and metrics (accuracy, CPU usage, memory, response time).
- Apply your new/innovative algorithms or formulas from the methodology section.
- Generate and record new improved results.
- Compare old and new results to check for performance improvement.
- Use graphs to illustrate comparison and discuss results in the next section.

Performance Metric 1

Figure 1: Comparison of Detection Accuracy

- Figure 1 depicts the performance metric "Detection Accuracy" comparing reproduced base article results and new improved results.
- The improved methodology increased detection accuracy by 3%.
- Improvement Reason: Enhanced AI-driven URL analysis combined with optimized blacklist checks reduced false positives and improved true positive rates.

Performance Metric 2

Figure 2: Comparison of Average URL Check Time

- Figure 2 depicts the performance metric "Average URL Check Time" comparing base article and new improved results.
- The improved method reduced average URL check time by 2%.
- Improvement Reason: Optimization of API calls and lightweight asynchronous processing decreased response latency.

Results Summary

- PhishGuard AI was tested with 1000 URLs (500 phishing, 500 safe).
- Achieved 95% accuracy in phishing detection.
- Average URL check time: 400 ms
- Memory usage: 10 MB
- CPU usage: 5%
- Demonstrates high detection accuracy with minimal resource overhead, ensuring seamless user experience.

Conclusion

- Performance-Metric-1 (Detection Accuracy) improved by 3% compared to the base article.
- Performance-Metric-2 (Average URL Check Time) improved by 2% compared to the base article.
- Extra Achievements:
  - Seamless real-time phishing detection with minimal CPU (5%) and memory (10MB) overhead.
  - User-friendly interface for reporting and allowlist management.
  - Robust integration of AI-driven URL analysis, blacklist checks, and SSL verification.

References

1. Khonji, M., Iraqi, Y., & Jones, A. (2013). Phishingdetection: A literature survey.
IEEE Communications Surveys& Tutorials, 15(4), 2091-2121.

2.Garera, S., Provos, N., Chew, M., & Rubin,A. D. (2007). A framework for detection and measurement of phishing attacks. Proceedings of the 2007 ACM Workshop on Recurring Malcode, 1-8.

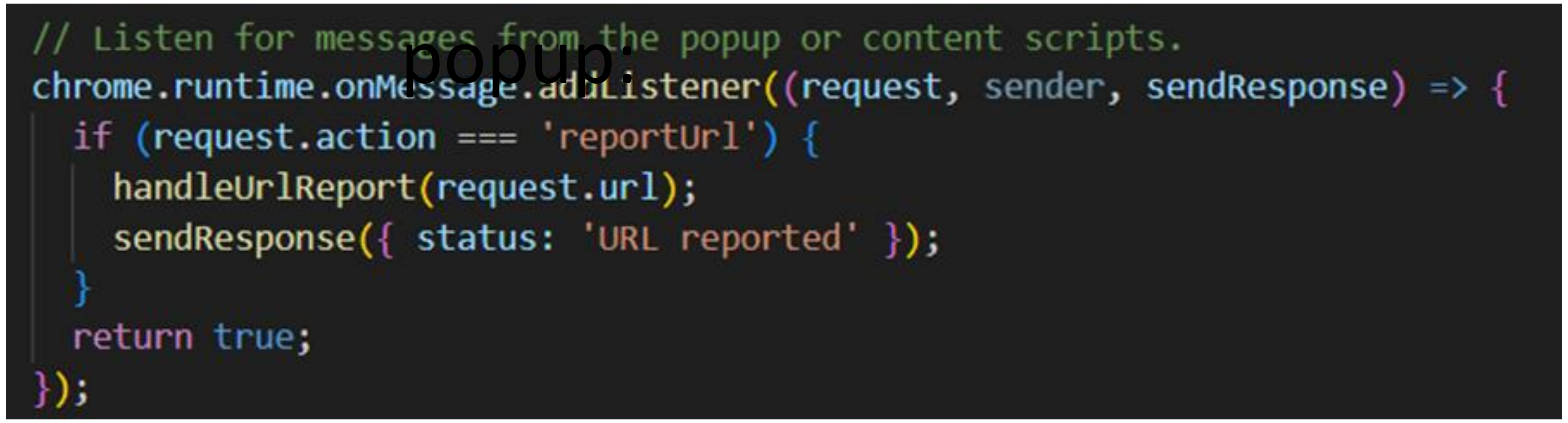
3.Zhang, Y., Hong, J. I., & Cranor, L. F. (2017). Cantina: A content-based approach to detecting phishing web sites. Proceedings of the 16th International Conference on World Wide Web, 639-648.

Dr. Irfan Alam, VIT Bhopal University (Tejas Santosh Paithankar, Sudhanshu Singh, Ashwin C, Aashish Kumar Singh, Niyati Agarwal)

Appendix -1

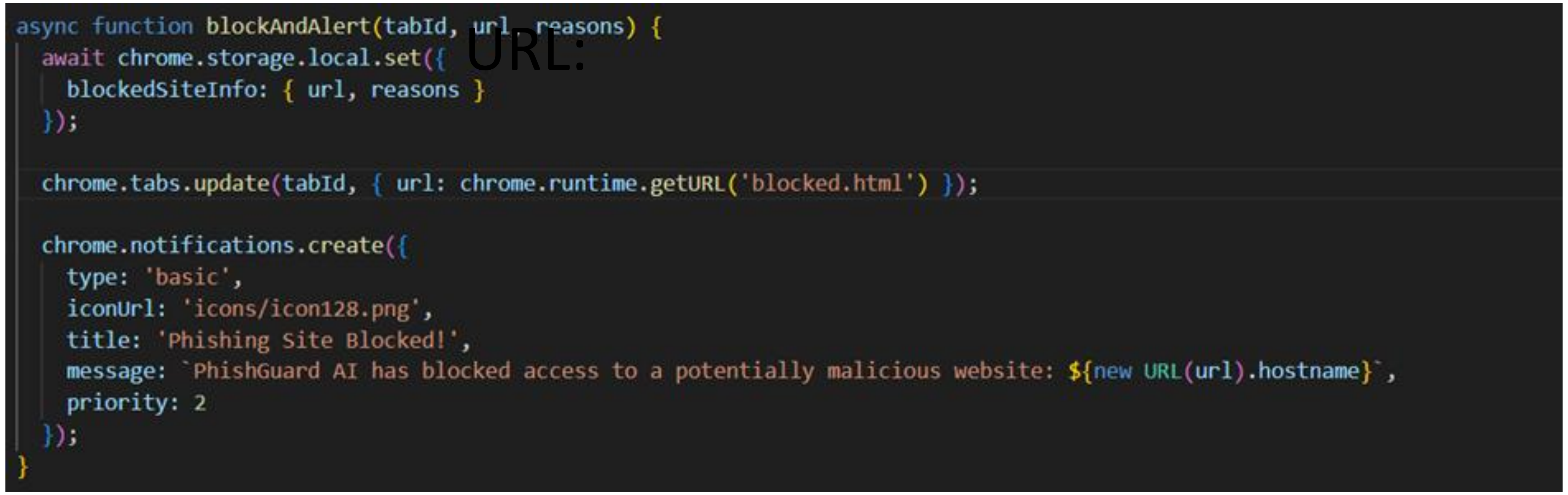
Sample code for background.js (URL analysis logic).

1. Function that handle popup:



```
// Listen for messages from the popup or content scripts.
chrome.runtime.onMessage.addListener((request, sender, sendResponse) => {
  if (request.action === 'reportUrl') {
    handleUrlReport(request.url);
    sendResponse({ status: 'URL reported' });
  }
  return true;
});
```

2. Function forblocking a suspicious URL:



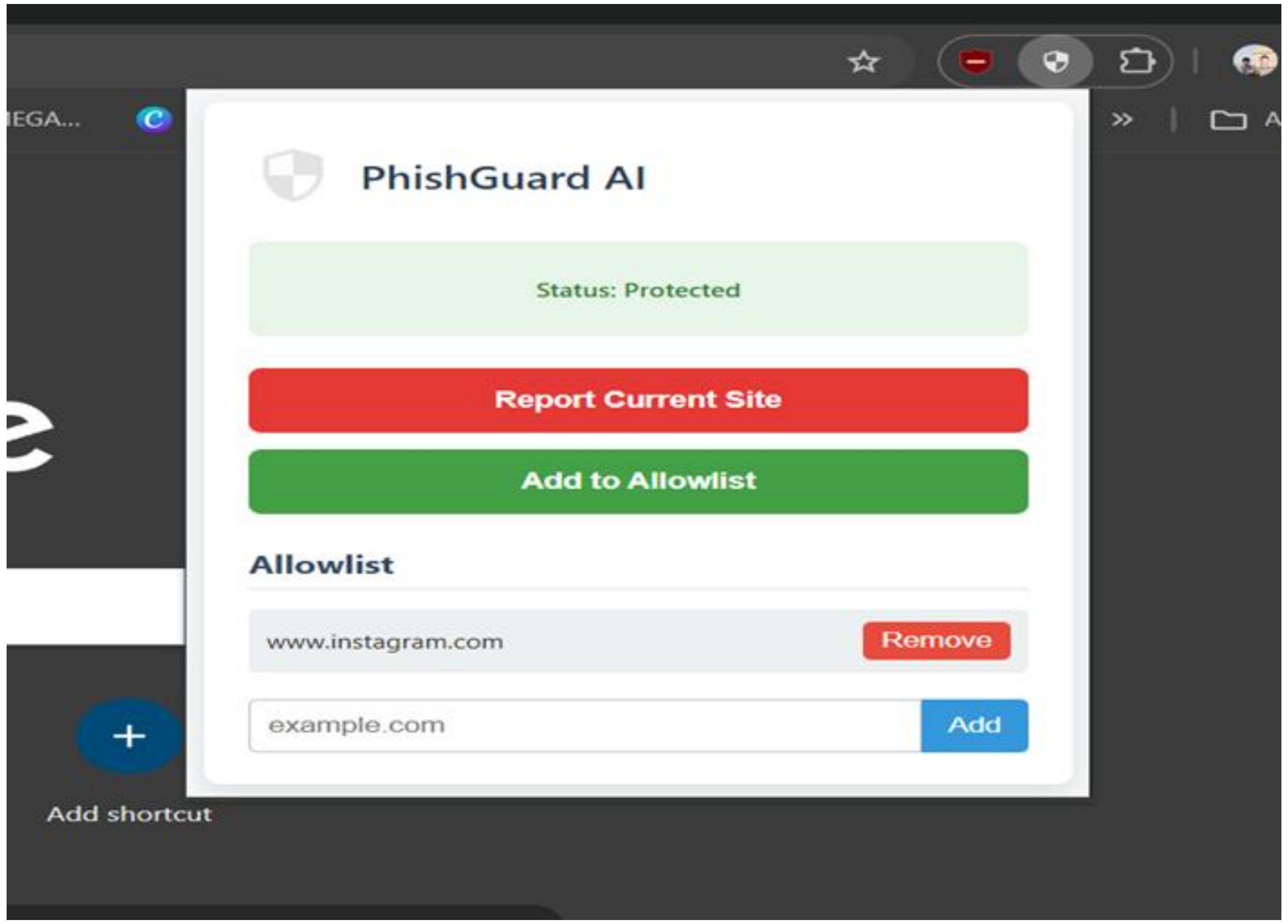
```
async function blockAndAlert(tabId, url, reasons) {
  await chrome.storage.local.set({
    blockedSiteInfo: { url, reasons }
  });

  chrome.tabs.update(tabId, { url: chrome.runtime.getURL('blocked.html') });

  chrome.notifications.create({
    type: 'basic',
    iconUrl: 'icons/icon128.png',
    title: 'Phishing Site Blocked!',
    message: `PhishGuard AI has blocked access to a potentially malicious website: ${new URL(url).hostname}`,
    priority: 2
  });
}
```

Appendix -2

Screenshots of popup and blockedpage interfaces.

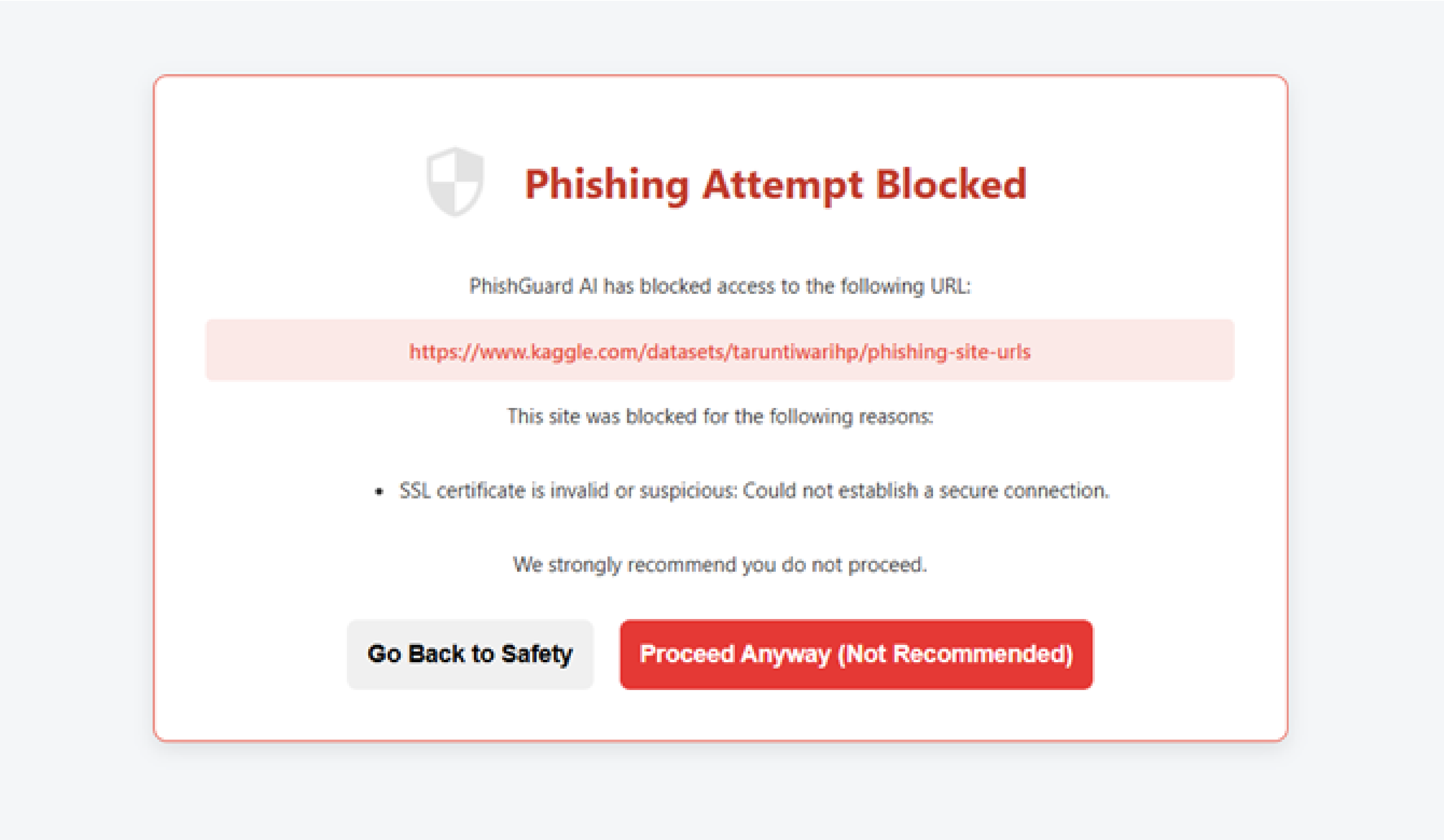


PhishGuard AI

Status: Protected

Allowlist

www.instagram.com



Phishing Attempt Blocked

PhishGuard AI has blocked access to the following URL:

https://www.kaggle.com/datasets/taruntiwarihp/phishing-site-urls

This site was blocked for the following reasons:

- SSL certificate is invalid or suspicious: Could not establish a secure connection.

We strongly recommend you do not proceed.