**National University of Lesotho**

WEBPAGE PHISHING DETECTOR

Proposal for a final year project

of

Lerotholi koai, 202101693

Thato gift lekhehle, 202101708

Supervisor: Mr B.Ndaba

Contents

[**1.** **Project Background** 3](#_Toc187204534)

[**2.** **Proposed solution** 4](#_Toc187204535)

[**3.** **Motivation** 4](#_Toc187204536)

[**4.** **Aims** 4](#_Toc187204537)

[**5.** **Objectives** 5](#_Toc187204538)

[**6.** **Limitations** 5](#_Toc187204539)

[**7.** **System requirements** 5](#_Toc187204540)

[**8.** **Theoretical Foundation** 6](#_Toc187204541)

[**9.** **Related Work** 6](#_Toc187204542)

[**10.** **High-level / conceptual system design** 7](#_Toc187204543)

[**11.** **Methodology and Software Tools** 8](#_Toc187204544)

[**12.** **Tools and technologies to be used** 9](#_Toc187204545)

[**13.** **Project schedule** 10](#_Toc187204546)

[**15.** **References** 11](#_Toc187204547)

# **Project Background**

**Problem Definition**

Now a days, users are increasingly reliant on the internet for a wide range of services, including banking, shopping, communication, and social networking. However, this increased online presence has also opened the door for various cybercrimes, with phishing attacks being among the most damaging. Phishing is a type of social engineering attack where malicious actors create deceptive websites that mimic legitimate ones, tricking users into revealing sensitive personal information such as passwords, credit card numbers, and other credentials.

Despite advancements in cybersecurity, phishing attacks continue to evolve by being more complex, making it harder for average users to differentiate between real and fake websites. Traditional defense mechanisms such as URL blacklists or basic browser security alerts are often insufficient because:

* Phishing sites can appear extremely similar to legitimate websites, often using slight variations in URLs.
* These sites are often short-lived, making blacklist updates slow and ineffective.
* Users may not always be knowledgeable enough to identify phishing attempts

The core problem, therefore, is the lack of a reliable, real-time, and efficient system to detect phishing websites, leaving users vulnerable to data theft and financial losses. This project seeks to address this problem by developing a webpage phishing detector system that uses techniques, including machine learning and URL/content analysis, to accurately and quickly identify phishing attempts and alert users before any damage can be done.

**Background**

The concept of phishing is not new. Since the early days of the internet, malicious actors have been coming up tactics to deceive users into revealing sensitive information. Originally, phishing attacks were limited to simple email scams, where users would receive deceptive emails asking them to click on a malicious link. However, over time, these attacks have evolved in both scale and sophistication, now including phishing websites that convincingly replicate legitimate websites. These websites often look identical to authentic ones, with fraudulent login forms or payment pages that collect and send user information to attackers.

There have been several existing solutions to detect phishing websites, including:

URL-based detection: This method focuses on identifying suspicious patterns in URLs, such as misspelled domains or suspicious characters. However, criminals often bypass this by using URL-shortening services or domains that look almost identical to the original.

Blacklist-based systems: These maintain lists of known phishing sites, and websites are compared against these lists. Although widely used, blacklist systems are often not very effective, as phishing sites are usually short-lived and can disappear before they are added to the list.

Browser Security Features: Web browsers (e.g., Chrome, Firefox) have built-in security features that warn users of potentially dangerous sites. However, these warnings are not always reliable, and some phishing sites still slip through.

# **Proposed solution**

To address the problem of phishing attacks and provide users with a reliable way to detect fraudulent websites in real time, this project proposes the development of a browser extension-based Webpage Phishing Detector System. The system will utilize a combination of advanced techniques such as:

* URL pattern analysis; it will analyze the URL of the website for signs of phishing, based on established patterns used by attackers to create deceptive URLs. It will check for: Suspicious domain structures, misspelled or altered domain names and unusual characters: The presence of symbols, numbers, or other irregularities in the URL.
* Machine learning algorithms to identify phishing websites: a model will be trained to classify websites as either phishing or legitimate. This model will be trained on a labeled dataset of both legitimate and phishing websites.
* Blacklist-Based Detection: To improve the accuracy of phishing detection, the system will maintain and constantly update a blacklist of known phishing websites. This blacklist will be regularly updated from various cybersecurity sources, allowing the system to quickly detect any website already recognized as malicious.

# **Motivation**

The increasing reliance on the internet for daily activities such as banking, shopping, and communication has heightened the need for robust cybersecurity measures. As phishing attacks have become one of the most common forms of cybercrime, we became deeply interested in how these malicious tactics operate and how they can be effectively mitigated. The ability to protect individuals and organizations from phishing attacks is critical in today’s digital landscape, and this sparked our interest in exploring cybersecurity, particularly the detection of phishing websites.

Phishing attacks target unsuspecting users by mimicking trusted entities, leading to the compromise of sensitive information like passwords, credit card details, and other personal data. After witnessing the growing sophistication of these attacks, we felt a strong desire to understand how phishing detection systems could help prevent such breaches. This personal interest in improving online safety motivated us to undertake this project as part of our final year in university.

# **Aims**

* **Real-Time Detection:**

Develop a robust browser extension to detect phishing websites and alert users to threats.

* **Enhanced Protection:**

Utilize advanced techniques like URL analysis and machine learning for accurate classification.

* **Reliable Detection:**

Always update database with lists of known phishing sites to block malicious websites.

* **Cybersecurity Awareness:**

Provide clear, user-friendly alerts to promote safe online behavior.

# **Objectives**

* **URL Pattern Analysis:**

Design algorithms to identify phishing indicators, including suspicious domain structures and misspellings.

* **Machine Learning Detection:**

Train a model on phishing and legitimate URL datasets for accurate classification.

Continuously refine the model so that it can adapt to evolving phishing techniques.

* **Known Phishing Site Detection:**

Maintain and update a list of malicious sites obtained from users who happen to be victims of phishing.

# **Limitations**

* **Accuracy of Detection**:

The system aims for high accuracy but false positives or negatives may occur, especially with sophisticated phishing sites.

* **Machine Learning Dependency**:

The effectiveness of the machine learning model depends on the quality and quantity of training data.

* **Real-Time Performance**:

Analyzing URLs and checking them against a database might slightly delay page loading times.

# **System requirements**

**Functional Requirements**

* The system should analyze URLs for patterns associated with phishing, including suspicious domain structures, misspellings, and unusual characters and also the system should use a machine learning model to classify websites as either phishing or legitimate.
* The system should alert the user in real-time if a website is suspected of being a phishing site by sending out notifications; which includes a warning message, and an option to proceed or navigate away from the suspicious site.
* The system should allow users to report websites they suspect to be phishing sites, contributing to the blacklist or database of potential threats.

**Non-functional requirements:**

* **Performance:**

The phishing detection process should be fast enough to provide real-time alerts without noticeable delays in website loading times.

* **Scalability:**

The system should be able to handle a large volume of URL checks and blacklist updates as the number of known phishing sites grows.

* **Usability:**

The browser extension should have a simple, user-friendly design, making it easy for non-technical users to understand and use the phishing detection features.

* **Reliability:**

It should be consistently available whenever users are browsing to ensure continuous protection.

# **Theoretical Foundation**

**1. Cybersecurity Principles**

Phishing Detection and Prevention:

Phishing is a form of social engineering that manipulates individuals into revealing sensitive information. Theories related to human behavior and manipulation promote phishing attacks.

Phishing involves exploiting weaknesses in human behavior and system vulnerabilities, such as through deceptive URLs, email spoofing, and fake websites. The concept of attack surfaces and their minimization is fundamental in cybersecurity, which ties into the system's ability to detect and mitigate phishing attempts.

**2. Machine Learning Principles**

Supervised Learning:

Classification Theory: Phishing detection often falls under the category of binary classification problems. The goal is to classify URLs as either “legitimate” or “phishing.” Supervised learning, where a model is trained on labeled examples, is commonly used for this purpose.

# **Related Work**

There are already existing extensions such as:

* **Netcraft:**

Netcraft has an anti-phishing community which is effectively a giant neighborhood watch scheme, empowering the most alert and most expert members to defend everyone within the community. As soon as the first recipients of a phishing mail report it, they can block it for all users of the extension providing an additional level of protection from Phishing. Netcraft processes reports of fraudulent URLs from a diverse variety of sources and proactively searches for new fraudulent sites.

* **PIXM:**

PIXM uses artificial intelligence to secure user’s browser and stop phishing in real-time. This technology visually analyzes websites to instantly determine if a page is using an unauthorized domain/URL to impersonate a legitimate company. When malicious pages are identified they are blocked, so that one does not inadvertently submit their data to a phishing website.

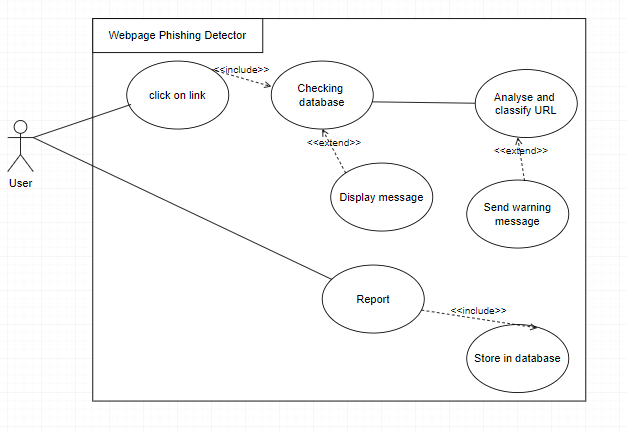
* **PhishDetect:**

PhishDetect is a browser extension designed to protect user from phishing attacks. Using the menu, the user is able to request a link or the content of a visited page to be analyzed by PhishDetect service and determine if it's suspicious.

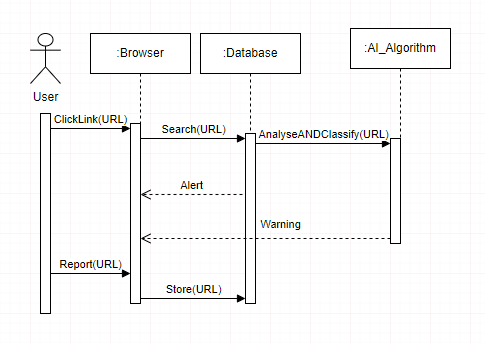
* Our project aims to improve on what already exists. It will have all of the above features that is; it will analyze URLs, have a database that will store already reported phishing website URLs and use Machine Learning algorithms to classify the phishing websites from legitimate ones.

# **High-level / conceptual system design**

* **Use case diagram**



* **Sequence diagram**



# **Methodology and Software Tools**

We will be using Scrum programming which is an agile methodology because; Scrum allows for quick adjustments to changes in requirements or project scope, enabling us to respond effectively to feedback, by breaking the project into smaller, manageable sprints, we will concentrate on delivering functional components of the project more frequently also, scrum emphasizes continuous testing and feedback, which helps identify and resolve errors as soon as they come up, resulting in a polished final product.

Using Scrum as a methodology offers numerous benefits, particularly for projects requiring flexibility, collaboration, and iterative progress. Here are the key advantages:

**Flexibility and Adaptability**

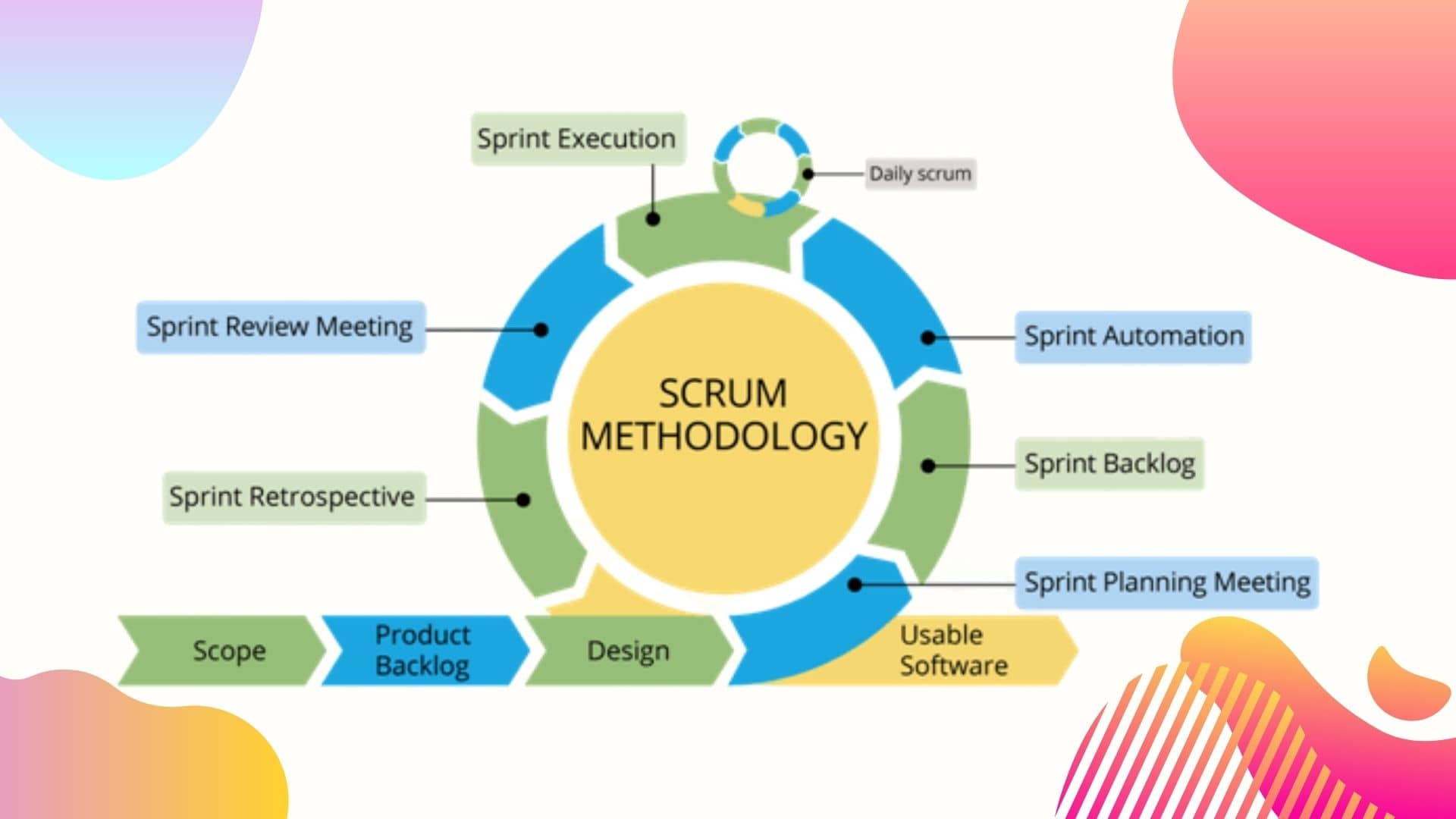
* Scrum enables teams to respond quickly to changing requirements or priorities, ensuring the final product aligns with user needs and business goals.

**Enhanced Collaboration**

* Daily stand-ups, sprint reviews, and retrospectives promote open communication and collaboration among team members, fostering a shared understanding of project goals.

**Continuous Improvement**

* Regular feedback through sprint reviews and retrospectives helps identify areas for improvement, ensuring the process evolves with each sprint



# **Tools and technologies to be used**

**Development tools to be used:**

**Python:**

Popular programming language for machine learning.

**Scikit-learn:**

Frameworks for building and training Machine Learning models.

**Jupyter Notebook:**

For experimenting and prototyping machine learning algorithms.

**MySQL:**

Structured relational databases to store reported URLs.

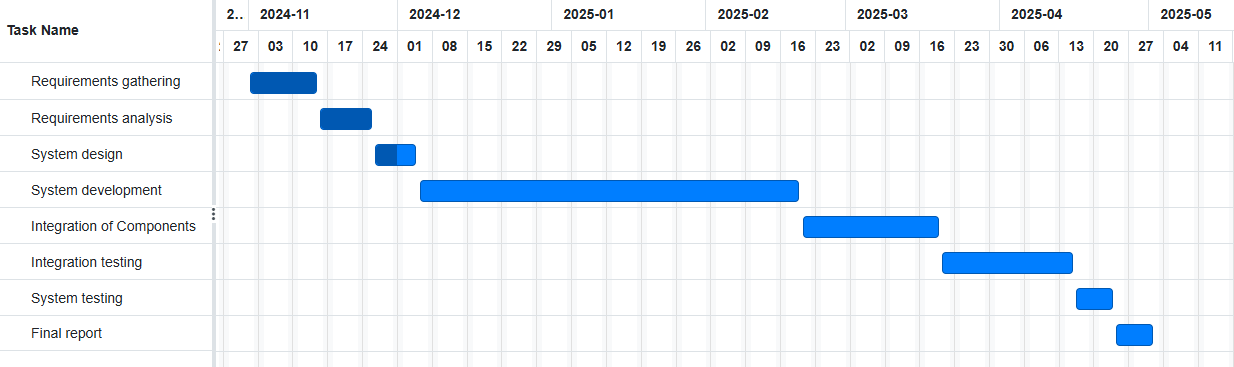
**JavaScript:**

Core languages for developing browser extensions and Chrome Extension API for building browser extensions and integrating with browser environment.

**Pandas/Numpy:**

For data manipulation and preprocessing

# **Project schedule**



1. The proposal outlines a final-year project to develop a Webpage Phishing Detector as a browser extension. The system combines URL pattern analysis, machine learning, and a blacklist-based approach to detect phishing websites in real time and alert users. By addressing limitations of existing solutions like slow blacklist updates and insufficient user knowledge, it aims to provide accurate, reliable, and user-friendly protection against phishing attacks

# **References**

[1] Kaspersky, 2023. Phishing explained. Available at: https://www.kaspersky.com/resource-center/definitions/phishing [Accessed 11 September 2024].

[2] Scikit-learn, 2024. Scikit-learn Documentation. Available at: https://scikit-learn.org/stable/ [Accessed 14 September 2024].

[3] Kaggle, 2024. Kaggle. Available at: https://www.kaggle.com/ [Accessed 11 September 2024].

[4] PhishTank, 2024. PhishTank. Available at: https://www.phishtank.com/ [Accessed 11 September 2024].

[5] Google, 2024. Chrome Developers - Extensions. Available at: https://developer.chrome.com/docs/extensions/ [Accessed 20 September 2024].

And many more