**SRI SIDDARTHA INSTITUTE OF TECHNOLOGY, MARLUR**

**TUMAKURU-572105**

**(A Constitute College of Sri Siddartha Academy of Higher Education)**

# Mini-Project(CS5MP1) Report On:

# “Phishing Website Analyser”

submitted in partial fulfillment of the requirement for the completion of V semester of

**BACHELOR OF ENGINEERING**

**in**

**Computer Science**

**Submitted by**

| | |
|---|---|
| Chirag P | (21CS022) |
| Maaz Pasha | (21CS050) |

Under the guidance of:

**Mrs.K Noor Fathima**

Assistant Professor

Department of CSE

SSIT, Tumakuru-03

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**2023-24**

**SRI SIDDHARTHA INSTITUTE OF TECHNOLOGY,**
**TUMAKURU-572105**
**DEPARTMENT OF COMPUTER SCIENCE& ENGINEERING**

# CERTIFICATE

Certified that the mini project work entitled "Phishing Website Analyser" is a bonafide work being carried out by Chirag P (21CS022), Maaz Pasha (21CS050) in partial fulfillment for the completion of V Semester of Bachelor of Engineering in Department of Computer Science & Engineering from Sri Siddhartha Institute of Technology,A Constitute College of Sri Siddartha Academy of Higher Education during the academic year 2023-24. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report deposited in the department library. The mini Project report has been approved as it satisfies the academic requirements in respect of mini project work prescribed for the Bachelor of Engineering degree.

Signature of the Guide
**Mrs.K Noor Fathima**
Assistant Professor, Dept of CSE

Signature of the HOD
**DR.RENUKALATHA S**
Professor & Head, Dept of CSE

# ACKNOWLDEGEMENT

**Chirag P (21CS022)** and **Maaz Pasha (21CS050)** of fifth semester, Department of Computer Science and Engineering of Sri Siddhartha Institute of Technology, Tumakuru, hereby declare that this Mini project titled, “Phishing Website Analyser”, has been carried out by us under the supervision of K Noor Fathima Assistant Professor of CSE, Department of Computer Science and Engineering, Sri Siddhartha Institute of Technology, Tumakuru in partial fulfilment of the requirement for the completion of V semester in Computer Science and Engineering.

| | |
|---|---|
| Chirag P | (21CS022) |
| Maaz Pasha | (21CS050) |

# Contents

# Abstract

This project introduces a robust solution, the Phishing Website Analyzer, designed to enhance cybersecurity through automated detection of potential phishing websites. With a primary focus on simplicity and effectiveness, this Python-based tool offers a streamlined approach to evaluate the safety of online platforms. The Phishing Website Analyzer employs a range of techniques, including URL structure analysis, domain age verification, SSL certificate validation, and content scrutiny. By leveraging these methods, the tool assesses websites for characteristics commonly associated with phishing attempts. The analysis is designed to be rapid and resource-efficient, allowing users to quickly ascertain the potential threat level of a given website. At its core, PhishingGuard boasts a user-friendly interface, catering to a diverse audience ranging from cybersecurity professionals to everyday internet users. Operating in real-time, the tool delivers immediate feedback and seamlessly integrates into popular web browsers, enhancing the user experience. Beyond its cybersecurity utility, this project serves as an educational resource, promoting awareness about the evolving landscape of online threats.

# Chapter 1

# Introduction

In an era dominated by digital connectivity, the internet serves as an integral part of our daily lives, facilitating communication, commerce, and information exchange. However, with the increasing reliance on online platforms comes a growing threat—phishing attacks. Phishing, a deceptive practice wherein malicious actors attempt to trick individuals into divulging sensitive information, poses a significant risk to the security and privacy of users. As part of our commitment to fortifying digital defenses, this project introduces the "Phishing Website Analyzer," a Python-based tool meticulously crafted to automate the identification of potential phishing websites. Phishing attacks often exploit the trust users place in seemingly legitimate websites, making it imperative to develop tools that can swiftly and accurately assess the authenticity of online platforms.

## 1.1 Problem Statement

The ubiquity of online activities has ushered in an era where the internet serves as the backbone for communication, commerce, and information dissemination. However, this digital landscape is marred by a pervasive threat—phishing attacks. Cybercriminals employ increasingly sophisticated tactics, creating deceptive websites that mimic legitimate platforms to exploit unsuspecting users and compromise sensitive information. Traditional methods of identifying phishing websites rely heavily on manual inspection, making them laborious, time-intensive, and often lag behind the rapidly evolving strategies employed by malicious actors. Consequently, there exists a critical gap in our cybersecurity defenses, demanding an automated solution that can swiftly and accurately analyze websites in real-time, empowering users to make informed decisions about the legitimacy and safety of online platforms. The challenge at hand is to develop a robust Phishing Website Analyzer, grounded in Python, that transcends the limitations of current approaches. This tool must seamlessly integrate advanced analysis techniques, including but not limited to URL structure examination, domain age verification, SSL certificate validation, and content scrutiny. By doing so, it aims to provide a comprehensive and user-friendly solution that not only identifies potential phishing threats but does so with the agility required in today's dynamic cyber landscape.

This project seeks to bridge the gap between cybersecurity needs and available tools, fostering a proactive defense mechanism against the rising tide of phishing attacks. Through the creation of an automated and intuitive Phishing Website Analyzer, we aim to empower users to navigate the digital realm with confidence, secure in the knowledge that they possess a tool capable of swiftly discerning between authentic and potentially malicious online entities.

## 1.2 Aim and Objectives

The aim of this project is to develop a Phishing Website Analyzer, a Python-based tool designed to automate the detection of potential phishing websites. The primary goal is to enhance cybersecurity by providing users with a user-friendly and efficient means of evaluating the legitimacy of online platforms, ultimately empowering them to navigate the digital landscape with confidence and resilience. The onjectives of the project includes are:

**1.Design and Develop the Phishing Website Analyzer:** Create a robust and scalable architecture for the Phishing Website Analyzer. Implement algorithms for URL structure analysis, domain age verification, SSL certificate validation, and content scrutiny.

**2.Real-time Analysis and User-Friendly Interface:**Enable the tool to perform real-time analysis for immediate feedback. Design an intuitive and user-friendly interface accessible to individuals with varying levels of technical expertise.

**3.Integration with Popular Web Browsers:** Seamlessly integrate the Phishing Website Analyzer with popular web browsers to enhance user accessibility. Ensure compatibility and smooth operation across different browser environments.

**4.Comprehensive Testing and Validation:** Conduct rigorous testing to validate the accuracy and efficiency of the analyzer. Test the tool against a diverse dataset of known phishing and legitimate websites to ensure its reliability.

**5.Open-Source Collaboration and Community Engagement:**Publish the tool as an open-source project to encourage collaboration and contributions from the cybersecurity community. Foster community engagement to gather feedback, address issues, and implement improvements.

**6.Educational Outreach and Awareness:**Develop educational resources accompanying the tool to raise awareness about phishing threats. Provide documentation and tutorials to help users understand the significance of cybersecurity and the role of the Phishing Website Analyzer.

**7.Continuous Improvement and Adaptation:**Establish mechanisms for continuous improvement based on emerging phishing tactics. Implement updates to the analyzer to adapt to evolving cybersecurity challenges.

**8.Evaluation of Impact:**Evaluate the impact of the Phishing Website Analyzer by measuring its effectiveness in mitigating phishing threats. Gather user feedback and assess the tool's contribution to creating a safer online environment.

By achieving these objectives, the project aims to deliver a valuable tool that contributes to the proactive defense against phishing attacks, promotes cybersecurity awareness, and fosters a safer online environment.

# Chapter 2

# Literature Survey

**1.“Machine Learning Approaches for Phishing Detection: A Comparative Analysis”**

**Authors:Ankit Kumar Jain, B B Gupta**

This study compares machine learning techniques for phishing detection, evaluating their accuracy and efficiency. It provides insights into the advantages and limitations of different machine learning models.

**2.“Enhancing Security Measures in Phishing Website Analyzers”**

**Authors: Amir Herzberg, Ahmed Jbara**

The Focusing on security aspects, this paper examines encryption methods, access controls, and compliance measures in phishing website analyzers. It proposes strategies to enhance overall security.

**3.“User Experience in Phishing Website Analysis”**

**Authors:John A Clark,Jeremy L Jacob**

Investigating user satisfaction and experience in the context of phishing website analysis tools, this research explores the impact on both analysts and end-users. It delves into usability, accessibility, and the overall improvement in service quality.

**4.“Scalability and Resource Management in Phishing Detection Systems”**

**Authors:Fabrizio Carcillo a, Andrea Dal Pozzolo**

This study discusses the scalability aspects of phishing detection systems, examining how these tools efficiently manage resources, adapt to varying demands, and optimize performance. It provides insights into ensuring the scalability of a Phishing Website Analyzer.

**5.“Challenges and Opportunities in Phishing Analysis Tools Implementation”**

**Authors: Patel R, Ankit Kumar Jain**

Addressing implementation challenges, this paper identifies common hurdles in deploying and using phishing analysis tools. It suggests strategies to overcome these obstacles, providing valuable insights for the successful implementation of a Phishing Website Analyzer.

# Chapter 3

# Requirements

## 3.1 System Hardware

**1.Server Requirements:** The system shall require a server infrastructure with adequate computational resources to handle the processing demands of website analysis.
Minimum Recommended Configuration: Processor: Dual-core or higher, RAM: 8 GB or higher, Storage: 100 GB SSD or higher
**2.Client-Side Requirements:** The client-side, where the user interacts with the system, shall have moderate hardware specifications to ensure a smooth user experience.
Minimum Recommended Configuration:Processor: Core i3 or equivalent, RAM: 4 GB, Storage: 128 GB HDD or higher
**3.Network Requirements:**The system shall require a stable and high-speed internet connection for real-time analysis and updates. Minimum Recommended Network Speed is 10 Mbps or higher.The system shall be compatible with major web browsers, including Chrome, Firefox, Safari, and Edge.
**4.Testing Environment:**The testing environment should replicate the hardware configurations of the production environment, including both server and client specifications. This ensures that testing accurately simulates the conditions under which the Phishing Website Analyzer will operate.

## 3.2 Software Requirements:

**1.Programming Languages:**
Python environment, Python 3.x should be installed on the system.Also Html which is a markup language and Css which is a style sheet language are required.

**2.Required Python Packages and IDE:**Install the necessary Python packages using the following command: pip install flask beautifulsoup4 requests,pip install requests,pip install Flask and use a Python-friendly IDE, such as PyCharm or Visual Studio Code, for development and debugging.

**3.Web Server:**A web server is required to host the Flask application. Flask's built-in development server can be used for testing purposes. For produThe BeautifulSoup library (beautifulsoup4) is used for HTML parsing. Ensure it is installed using the command mentioned above.ction deployment, consider using a production-ready web server such as Nginx or Apache with a WSGI server (e.g., Gunicorn).

**4.HTML Parser:**The BeautifulSoup library (beautifulsoup4) is used for HTML parsing. Ensure it is installed using the command mentioned above.

**5.Networking and Browser Compatibility:** Ensure that the system has internet connectivity, as the code makes requests to external websites for HTML content retrieval.The application is designed to be accessed through a web browser. Ensure that the system has a compatible web browser installed.

**6.Operating System:** The code is platform-independent and can run on various operating systems such as Windows, macOS, or Linux.

**7.Version Control (Optional):** Git: If you plan to use version control, you'll need Git to be installed

**8.Cloud Platform (Optional for Deployment):**

Heroku, AWS, or another cloud platform if you plan to deploy your application online.

**9Dependency Management (Optional):**

Consider using a virtual environment for managing project dependencies. You can use venv or virtualenv in Python.

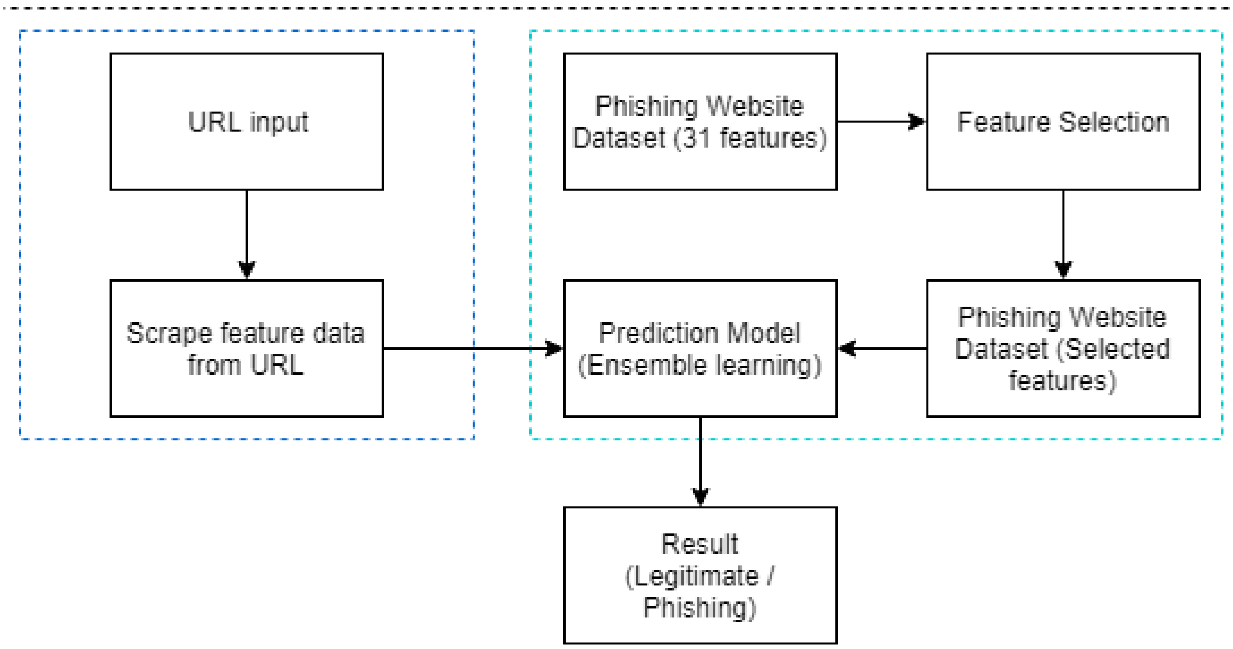
**10.Containerization (Optional):**

Docker: If you want to containerize your application for easy deployment, consider using Docker.

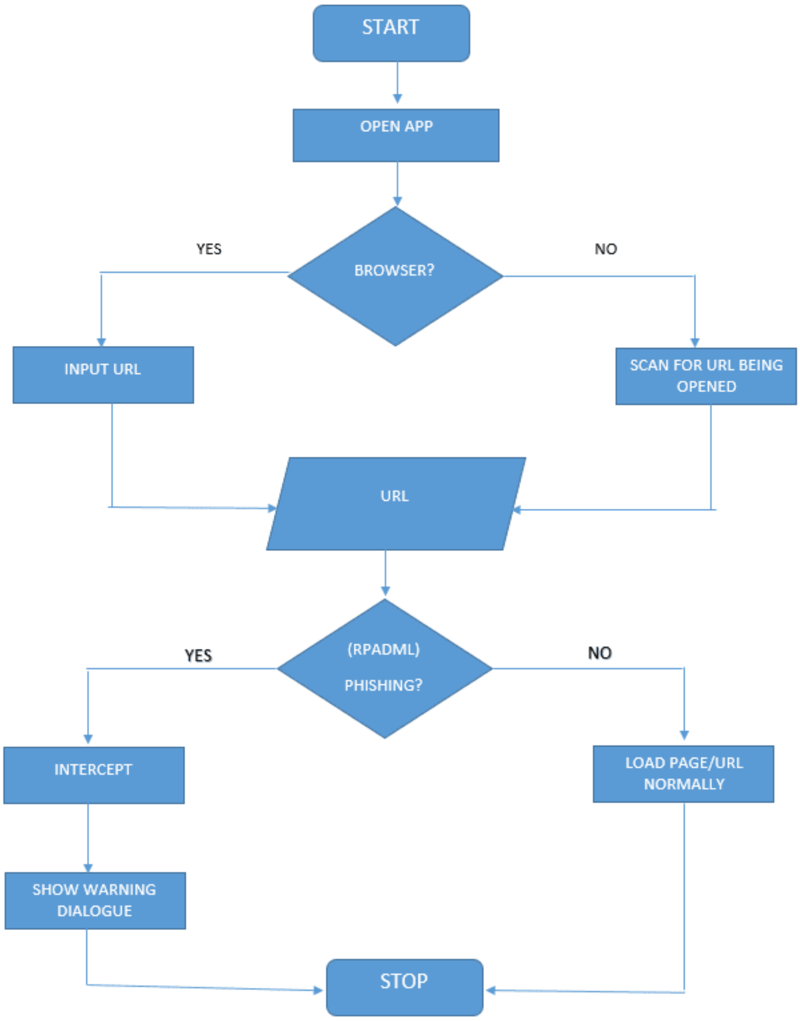
# Chapter 4

# Design

## 4.1 Data flow Diagram:


URL input
Scrape feature data from URL
Phishing Website Dataset (31 features)
Feature Selection
Phishing Website Dataset (Selected features)
Prediction Model (Ensemble learning)
Result (Legitimate / Phishing)

## 4.2 Activity diagram for Detection of phishing website:



# Conclusion

In conclusion, the Phishing Website Analyzer project represents a significant step towards enhancing online security by providing users with a reliable tool to identify potential phishing threats.Through the seamless integration of user interface, URL validation, HTML content retrieval, parsing, and phishing analysis modules, this project offers a comprehensive solution to safeguard users from malicious online activities. As the cybersecurity landscape continues to evolve, the marriage of traditional phishing analysis techniques with machine learning augments the system's proactive defense against emerging threats. By continuously learning and adapting, the Phishing Website Analyzer with machine learning capabilities exemplifies a resilient and forward-thinking approach to safeguarding users in the complex and dynamic online environment.In summary, the Phishing Website Analyzer project is not just a tool; it's a proactive measure towards cultivating a culture of cybersecurity awareness.

# Bibliography

[1] Beautiful Soup Documentation:Richardson, L. (2021). Beautiful Soup Documentation. Retrieved from https://www.crummy.com/software/BeautifulSoup/bs4/doc/

[2] Requests Documentation:Kenneth Reitz. (2021). Requests Documentation. Retrieved from **https://docs.python-requests.org/en/latest/**

[3] Flask Documentation: Pallets Projects. (2021). Flask Documentation. Retrieved from https://flask.palletsprojects.com/

[4] HTML5 (for general understanding of HTML): W3C. (2014). HTML5 - A vocabulary and associated APIs for HTML and XHTML. Retrieved from **https://www.w3.org/TR/html52/**

[5] CSS (for general understanding of CSS): W3C. (2018). Cascading Style Sheets (CSS) - The Official Definition. Retrieved from **https://www.w3.org/Style/CSS/**

[6] Python Official Documentation: Python Software Foundation. (2021). Python Documentation. Retrieved from **https://docs.python.org/3/**