ISSN 2277-2685
IJESR/April-June. 2025/ Vol-15/Issue-2s/603-609
Panyam Jyoni ***et. al.***, **/ International Journal of Engineering & Science Research**

SafeSurf - Phishing Website Detection

[1]P Mounika, [2]**Panyam Jyoni, [3]T Karunya**
[1]Assistant professor, Department of CSE, Bhoj Reddy Engineering College for Women, India
[2,3]B.Tech Students, Department of CSE, Bhoj Reddy Engineering College for Women, India

ABSTRACT
SafeSurf is an advanced phishing domain detection tool developed in Python to enhance online safety by identifying potentially harmful websites. Its key feature is allowing users to preview websites without visiting them, thereby reducing the risk of exposure to phishing or malicious content. SafeSurf also assigns a trust score to each URL, providing users with an indication of its authenticity. The tool integrates with PhishTank to verify if the URL has been reported as phishing. Additionally, it provides essential domain details, such as WHOIS data and SSL certificate status, offering users a deeper understanding of the website's legitimacy. Designed with a user-friendly interface, SafeSurf ensures a seamless experience for individuals looking to protect themselves from phishing attacks.

1-INTRODUCTION

In today's digital landscape, where phishing attacks are a constant threat, SafeSurf offers a dependable, user-friendly tool for secure browsing and phishing protection. Phishing attacks—aimed at tricking users into interacting with fake websites to steal sensitive data—have become more sophisticated, making it essential for users to have tools that can help identify and avoid these risks. SafeSurf is designed to allow users to evaluate website safety without directly visiting potentially harmful sites. It features a clean, intuitive interface that enables users to navigate seamlessly and assess URL safety quickly. SafeSurf's real-time trust scoring provides a clear indication of a website's credibility, enabling users to make well-informed decisions about whether to proceed. By checking URLs against PhishTank , SafeSurf immediately identifies known phishing sites, warning users if a link is flagged as Phishing. Additionally, SafeSurf includes a secure website preview option, allowing users to view a snapshot of the website without a full visit. This feature gives users insight into a site's content, helping them spot suspicious elements and make safer browsing decisions. With its combined features, SafeSurf provides robust protection against phishing, empowering users to browse with confidence and enhancing overall online safety.

Existing System

The existing systems for phishing detection primarily rely on techniques such as blacklists, where known phishing websites are cataloged, and heuristic approaches that identify suspicious features, like excessively long URLs or unsafe connections. However, these blacklists often fail to update rapidly enough to catch new phishing sites, leaving users vulnerable.

Proposed System

The proposed system for SafeSurf is a Python-based phishing domain detection tool that ensures user safety by allowing secure website previews without direct access. The system works by analyzing URLs and cross-referencing them with a phishing database like PhishTank to detect any reported threats. It assigns a trust score based on several factors, including SSL certificate verification, domain age, and WHOIS information. Additionally, SafeSurf provides users with a snapshot preview of the website, reducing the need for direct interaction. The tool is designed with a simple, intuitive interface, making it accessible to a wide range of users.This approach aims to enhance phishing detection and improve online safety for users.

2- FUNCTIONAL REQUIREMENTS

System Architecture

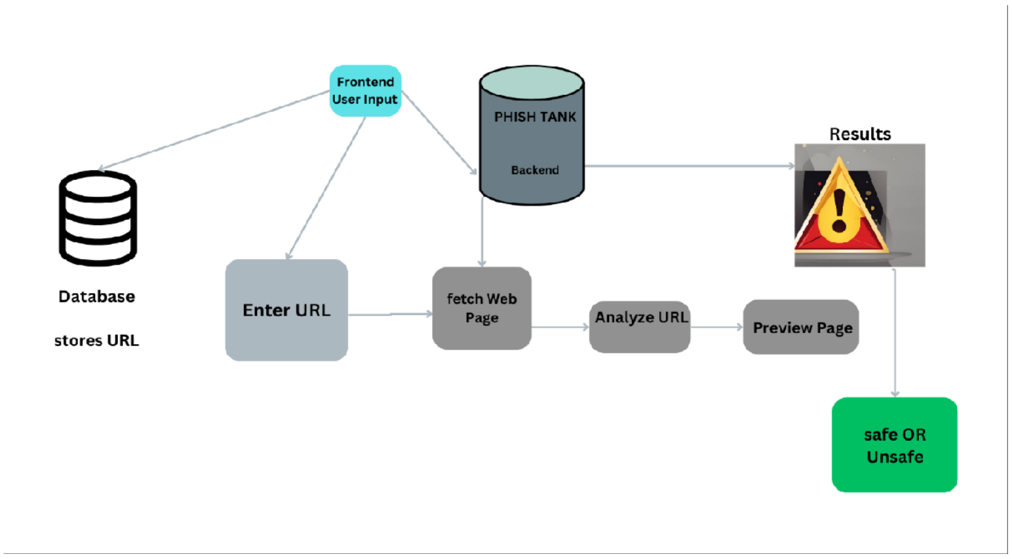


Fig-2.3.1 **System Architecture Diagram**

3-DESIGN AND IMPLEMENTATION

Methodology

The app uses Flask for web serving, SQLAlchemy for database management, BeautifulSoup for HTML parsing, and requests for making HTTP requests.

The Controller class is imported from another module, likely containing methods to process the URLs.

onetimescript and db modules are imported, which are expected to contain database initialization, and specific one-time update functions (update_db and update_json).

Database Configuration:

The app uses SQLite as a database, configured with SQLALCHEMY_DATABASE_URI.

db.init_app(app) initializes the SQLAlchemy database with the Flask app. db.create_all() creates the necessary tables within the app context if they don’t already exist.

Routes:

Home Route (/):

Supports both GET and POST requests.

If the request is a POST, it retrieves the URL from the form, processes it using the controller.main(url) method, and returns the output in index.html.

If an error occurs, it sets output to "NA". Preview Route (/preview):

This route accepts a POST request to preview an external URL.

It fetches the HTML content from the provided URL, uses BeautifulSoup to parse it, and updates the relative paths of external resources like links and images to be absolute URLs using urljoin.

The HTML is rendered in preview.html.

If an error occurs, it returns an error message with the exception.

Database Update Route (/update-db):

This route invokes onetimescript.update_db() to populate the database, likely used for initial or periodic database updates.

Prints a success message to the console and returns a 200 response if successful, otherwise, it returns a 500 response with the error.

JSON Update Route (/update-json):

Similar to /update-db, this route calls onetimescript.update_json() to update a JSON file or dataset.

It provides feedback through console output and HTTP response codes. **Running the App:**

When executed directly, the app runs in debug mode on the default Flask port.

Structural Diagram

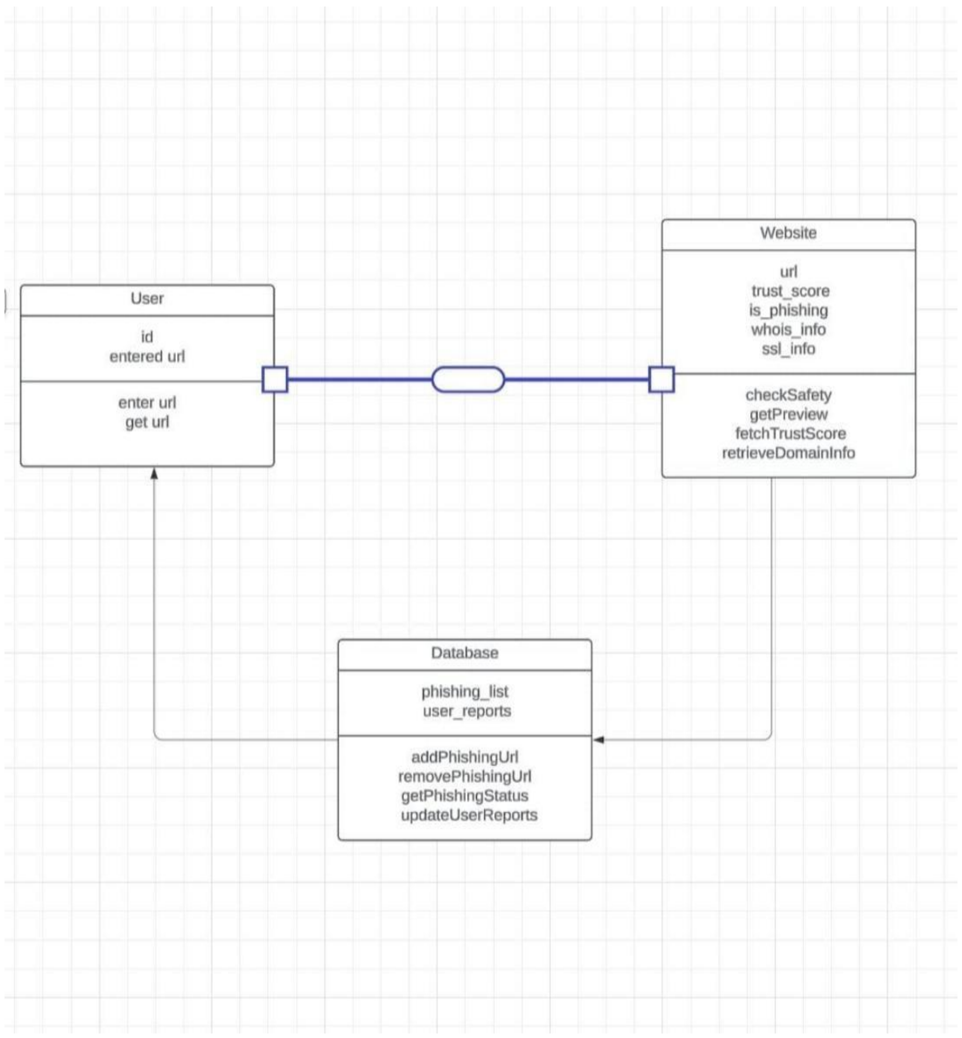


Fig 3.2.1 Class Diagram

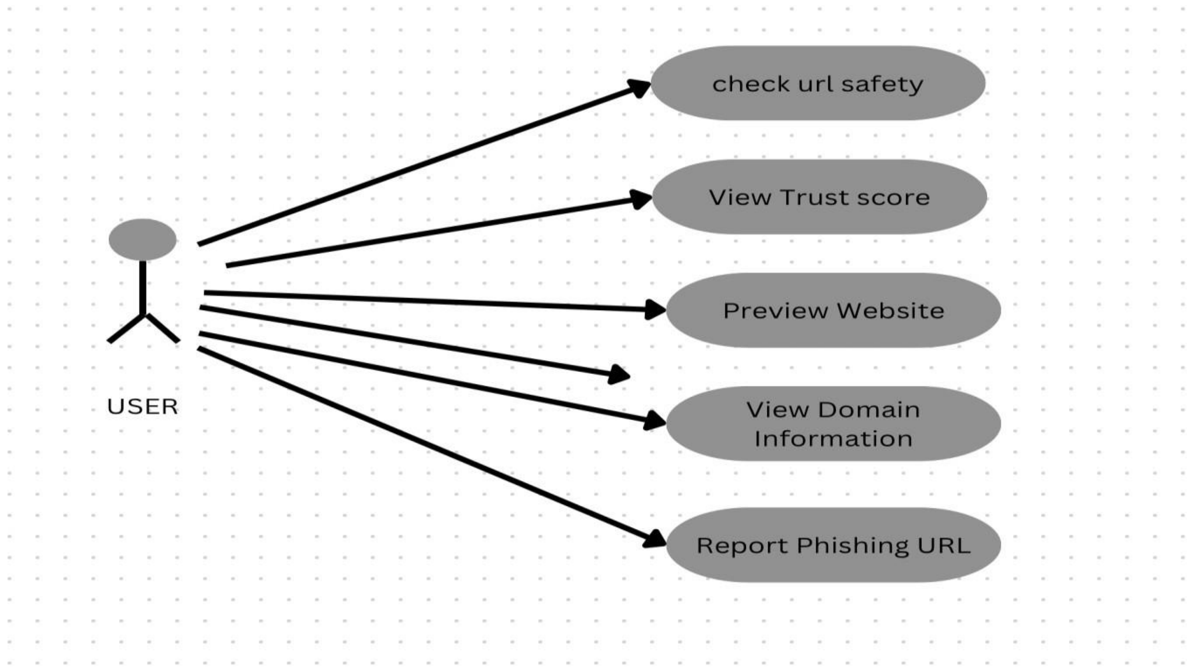


Fig 3.3.2 Usecase Diagram

4-RESULTS &DISCUSSION

Environmental Setup

Installing Visual Studio Code (VS Code)

1. Download VS Code:
   - Download Visual Studio Code from the official website: https://code.visualstudio.com/Download
2. Run Installer
   - After downloading, run the installer and follow the prompts.
3. Installation Steps:
   - Click "Next" to proceed through the setup steps.
   - Choose the installation location and click "Install".
4. Finish Setup:
   - Once installed, click "Finish" to launch VS Code.

Installing Python

1. Download Python:
   - Download Python from the official website: https://www.python.org/downloads/
2. Run Installer:
   - Check "Add Python to PATH" during installation.
   - Follow the installation prompts to install Python.
3. Verify Installation:
   - Open the terminal or command prompt and run:

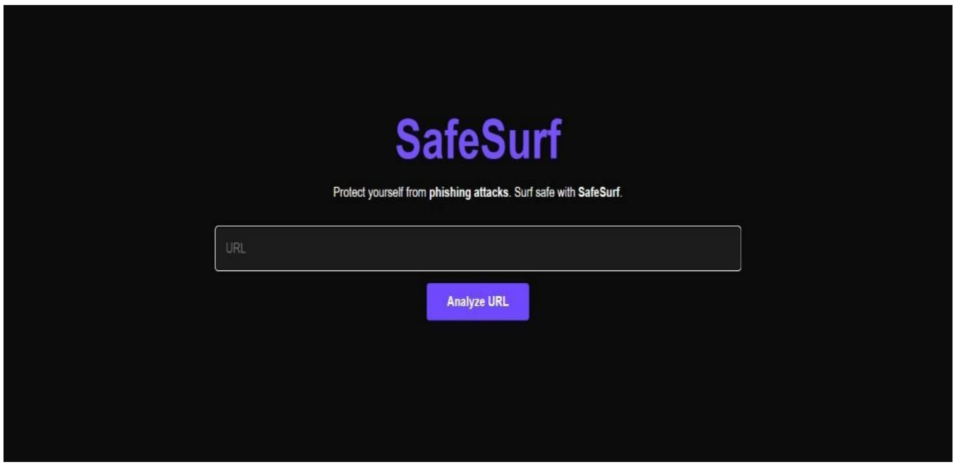
```
python --version
pip --version
```

   - This will display the installed versions of Python and pip.

Installing Dependencies for SafeSurf

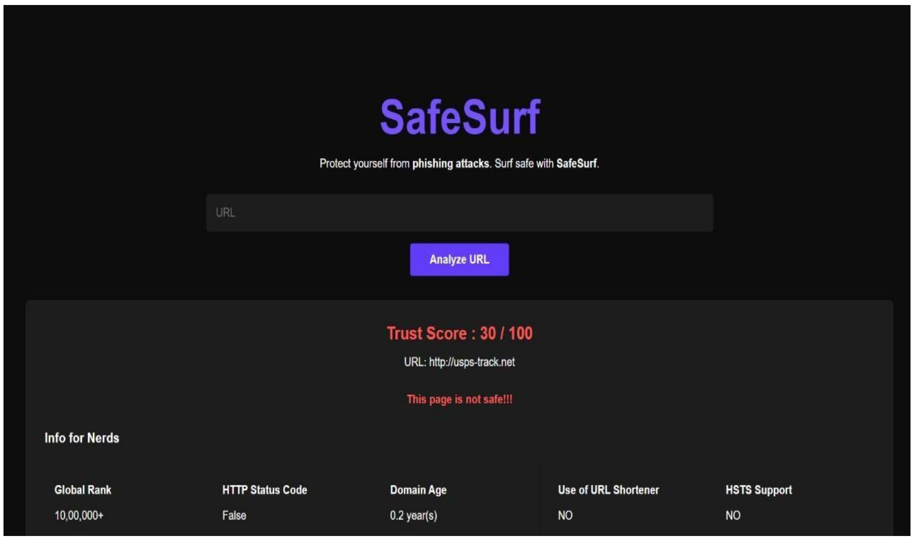
1. Install Flask and Other Required Libraries:
   - Open the terminal in VS Code and run:

RESULTS



Protect yourself from **phishing attacks**. Surf safe with **SafeSurf**.

Fig-4.3.1 Home page



Protect yourself from **phishing attacks**. Surf safe with **SafeSurf**.

Trust Score : 30 / 100

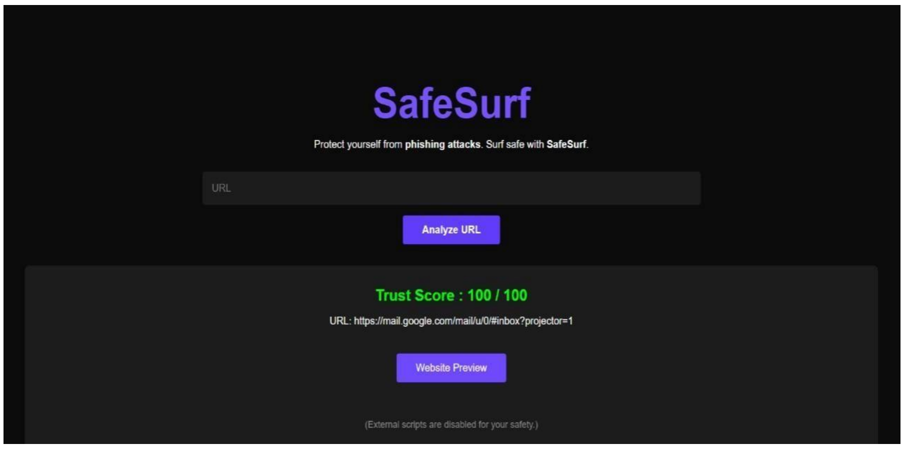
URL: http://usps-track.net

This page is not safe!!!

Info for Nerds

Global Rank	HTTP Status Code	Domain Age	Use of URL Shortener	HSTS Support
10,00,000+	False	0.2 year(s)	NO	NO

Fig-4.3.2 Unsafe url



Protect yourself from **phishing attacks**. Surf safe with **SafeSurf**.

Trust Score : 100 / 100

URL: https://mail.google.com/mail/u/0/#inbox?projector=1

(External scripts are disabled for your safety.)

Fig-4.3.3 Safe ur

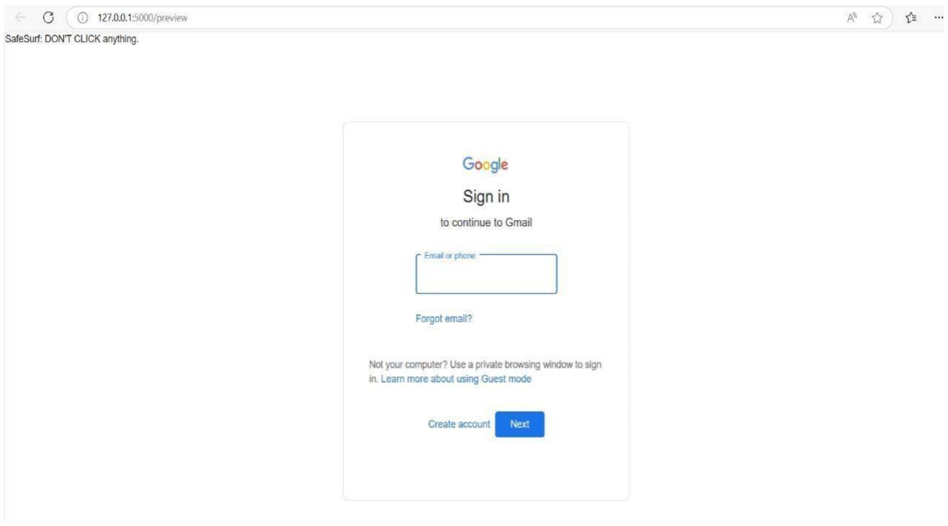

127.0.0.1:5000/preview
SafeSurf: DON'T CLICK anything.
Google
Sign in
to continue to Gmail
Email or phone
Forgot email?
Not your computer? Use a private browsing window to sign in. Learn more about using Guest mode
Create account
Next

Fig-4.3.4 Website preview

Test Cases:

SNo	**Description**	**Input**	**Output**	**Status**
1.	URL from non- trusted source	{url: 'http://unknown- source.com'}	Result: "Unsafe"	Pass
2.	Valid trusted URL	{url: 'https://google.com'}	Result: "Safe"	Pass
3.	Invalid domain format	{url: 'https://invalid_domain.com'}	Error message indicating invalid URL format	Fail
4.	Safe website preview	{url: 'https://example.com'}	Displays embedded preview of website	Pass

5-CONCLUSION

Conclusion

SafeSurf represents a significant advancement in online safety by empowering users with tools to identify phishing domains and reduce exposure to malicious content. Through its combination of URL preview functionality, trust scoring, and PhishTank integration, SafeSurf provides a reliable, accessible way for users to assess the safety of websites before interacting with them. SafeSurf adds an extra layer of transparency, allowing users to make informed decisions. With its intuitive interface, SafeSurf makes web safety approachable for users of all experience levels, underscoring its value as a practical tool in the fight against phishing attacks.

REFERENCES

- Aburrous, M., Hossain, M. A., Dahal, K., & Thabtah, F. (2010). *Intelligent phishing detection system for e-banking using fuzzy data mining*. Expert Systems with Applications, 37(12), 7913-7921.
- Patil, P., & Patil, D. (2019). Phishing website detection based on machine learning. International

Journal of Advanced Research in Computer Engineering & Technology (IJARCET), 8(2).

- Mohammad, R. M., Thabtah, F., & McCluskey, L. (2012). An assessment of features related to phishing websites using an automated technique. In Proceedings of the International Conference for Internet Technology and Secured Transactions (ICITST), 492-497