Project Report: Phishing Website Detection Tool

1. Problem Statement

Phishing websites trick users into revealing sensitive information such as usernames, passwords, and banking credentials. These fraudulent websites are often hard to distinguish from legitimate ones. This project aims to build a tool that can detect phishing URLs using both **rule-based heuristics**, **machine learning models**, and **Google Safe Browsing API**, strictly for educational and awareness purposes.

2. Objectives

- Detect phishing websites based on patterns in URL structure.
- Use machine learning to classify URLs as legitimate or phishing.
- Implement rule-based filtering for instant flagging.
- Integrate Google Safe Browsing API for real-time threat lookup.
- Provide a simple, user-friendly GUI interface for detection.

3. Tools & Technologies

Tool/Library	Purpose
Python 3.x	Core programming language
pandas	Data preprocessing and handling
scikit-learn	Machine learning training and prediction
regex	Pattern matching for rule-based detection
requests	API calls to Google Safe Browsing
tkinter	GUI interface for user interaction

4. Functional Features

Rule-Based Detection

- Uses static heuristics like presence of @, -, suspicious keywords (login, secure, etc.), and long URLs.
- Flags instantly if certain conditions are met.

Machine Learning Detection

- Extracts features such as URL length, dot count, HTTPS presence, etc.
- Trains a Random Forest classifier using a labeled dataset of URLs.
- Predicts “Phishing” or “Legitimate” based on engineered features.

Google Safe Browsing API

- Queries Google's real-time database of harmful URLs.
- If a threat is detected, it overrides all other results for better accuracy.

GUI Interface

- Built using tkinter for cross-platform simplicity.
- Takes URL input from user and shows detection result.
- Color-coded results and integrated ML + API logic.

5. Output Structure

phishing_detector/

├── gui.py

├── ml_detector.py

├── rule_based_detector.py

├── dataset.csv

├── requirements.txt

└── logs/ (optional)

Sample Output:

GUI Output:

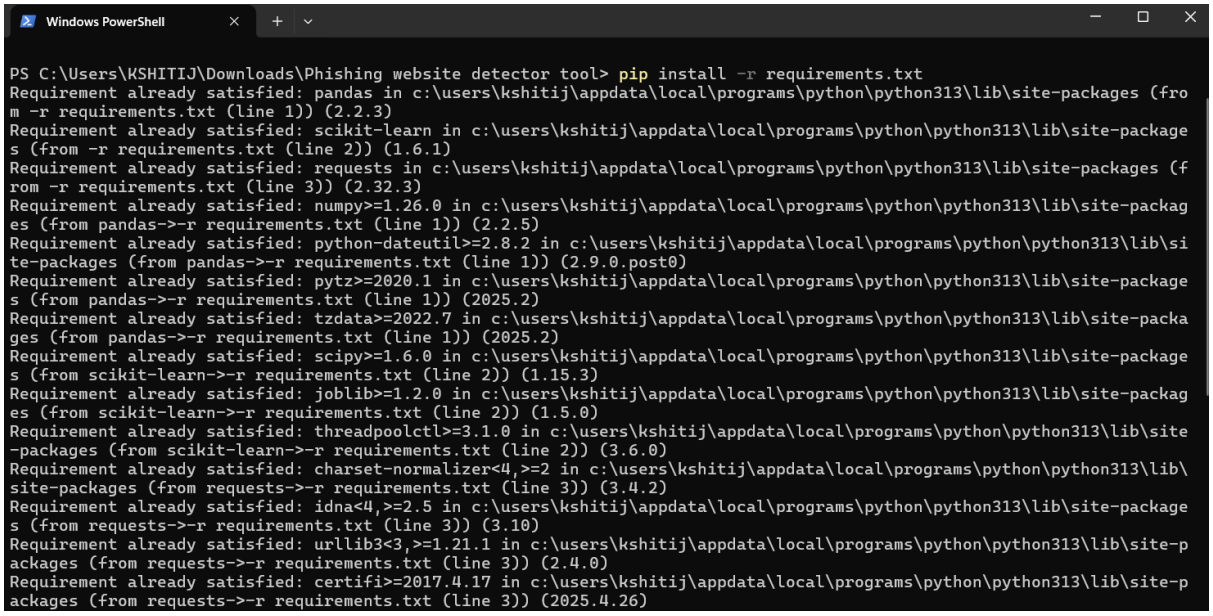
Result: Phishing (Detected via Google API)

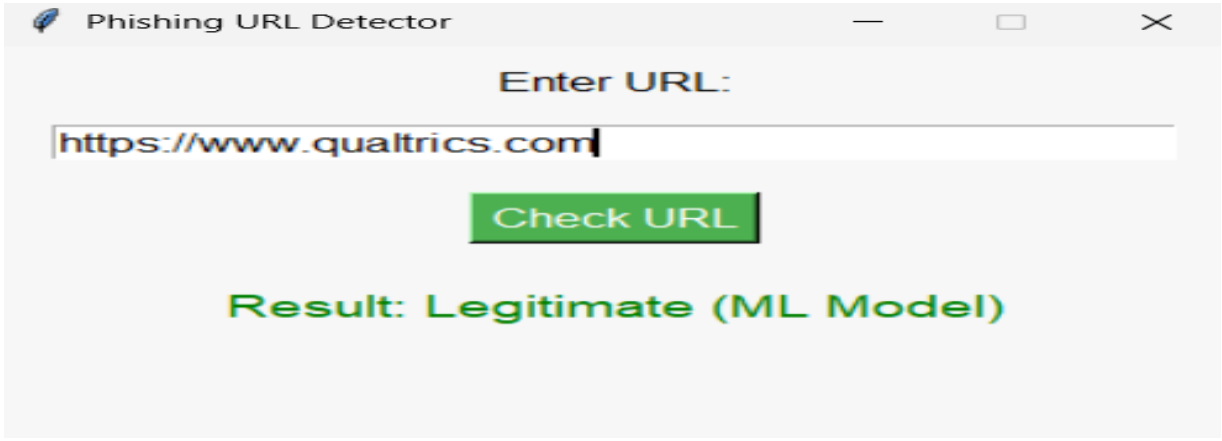
CLI Output:

Enter URL: http://*secure-login-update.com*

Result: Phishing

6. Screenshots






```
PS C:\Users\KSHITIJ\Downloads\Phishing website detector tool> python -m tkinter
PS C:\Users\KSHITIJ\Downloads\Phishing website detector tool> python gui.py
PS C:\Users\KSHITIJ\Downloads\Phishing website detector tool> python ml_detector.py
              precision    recall  f1-score   support

           0       0.99      1.00      0.99     68921
           1       0.99      0.95      0.97     21115

    accuracy                           0.99     90036
   macro avg       0.99      0.98      0.98     90036
weighted avg       0.99      0.99      0.99     90036

Enter URL to check: https://www.citehr.com/216264-leave-policy-according-ap-shop-established-act-1988-a.html
Result: Legitimate
PS C:\Users\KSHITIJ\Downloads\Phishing website detector tool> python rule_based_detector.py
Enter a URL to check: https://www.citehr.com/216264-leave-policy-according-ap-shop-established-act-1988-a.html
Result: Phishing (contains hyphen)
PS C:\Users\KSHITIJ\Downloads\Phishing website detector tool>
```

7. Learning Outcomes

- Understood how phishing URLs are structured and how attackers disguise them.
- Learned data preprocessing and feature engineering for URL classification.
- Integrated external APIs (Google Safe Browsing) with local ML models.
- Built a desktop application using tkinter for real-time interaction.

8. Future Improvements

- Add WHOIS and DNS-based features to detect newly registered domains.
- Deploy the tool as a browser plugin or web-based service.
- Train deep learning models on larger, public datasets.
- Provide reporting/logging features for phishing attempts.

9. Conclusion

This project successfully demonstrates a multi-layered approach to phishing website detection. By combining rule-based filters, machine learning classification, and Google's threat database, the tool offers a practical and educational way to understand modern phishing threats. It provides a strong foundation for further development in **cybersecurity automation**, **user protection tools**, and **ethical hacking practices**.