WWW.IJRPR.COM

**International Journal of Research Publication and Reviews**

Journal homepage: www.ijrpr.com ISSN 2582-7421

# Identification of Fake Websites and Email Detection using Web Scraping

***Dr. Menaka G[1], Abinaya Shri P[2]***

[1] Vice Principal Vivekanandha College of Arts and Sciences for Women [Autonomous], Tiruchengode, Namakkal, Tamilnadu, India.
[2] II MCA, PG & Research Department of Computer Science and Applications, Vivekanandha College of Arts and Sciences for Women [Autonomous], Tiruchengode, Namakkal, Tamilnadu, India

**ABSTRACT :**

This paper presents a unified web-based platform developed for the real-time detection of fake websites and suspicious email addresses. The system utilizes web scraping techniques, domain analysis, and content verification to evaluate the legitimacy of online inputs. Built using the Flask framework, the platform features a single input field allowing users to verify both website URLs and email addresses seamlessly. By combining multiple validation mechanisms, the system provides accurate, fast, and transparent feedback, enhancing user trust in online interactions.

**Keywords**: Fake Website Detection, Email Validation, Cybersecurity, Web Application, Flask Framework, Web Scraping

## 1. Introduction

With the exponential rise of digital communication, users have become increasingly reliant on online platforms for personal, professional, and commercial activities. However, this convenience has led to an increased risk of encountering malicious websites and fake email addresses, both of which are often used to facilitate scams, data breaches, and phishing attacks. Identifying and mitigating such risks is critical to ensuring the safety and security of digital ecosystems. Traditional tools often separate website verification and email validation processes, resulting in fragmented user experiences. To address this limitation, we propose a unified web-based system that consolidates both functionalities into a single platform. Users can input either a website URL or an email address and instantly receive an authenticity evaluation. The system leverages web scraping, SSL certification checks, WHOIS domain data, phishing keyword analysis, and email domain validation to perform comprehensive real-time verification. The primary goal of this project is to enhance online trust by equipping users with a fast, easy-to-use tool that helps them avoid fraudulent entities.

## 2. Literature Survey

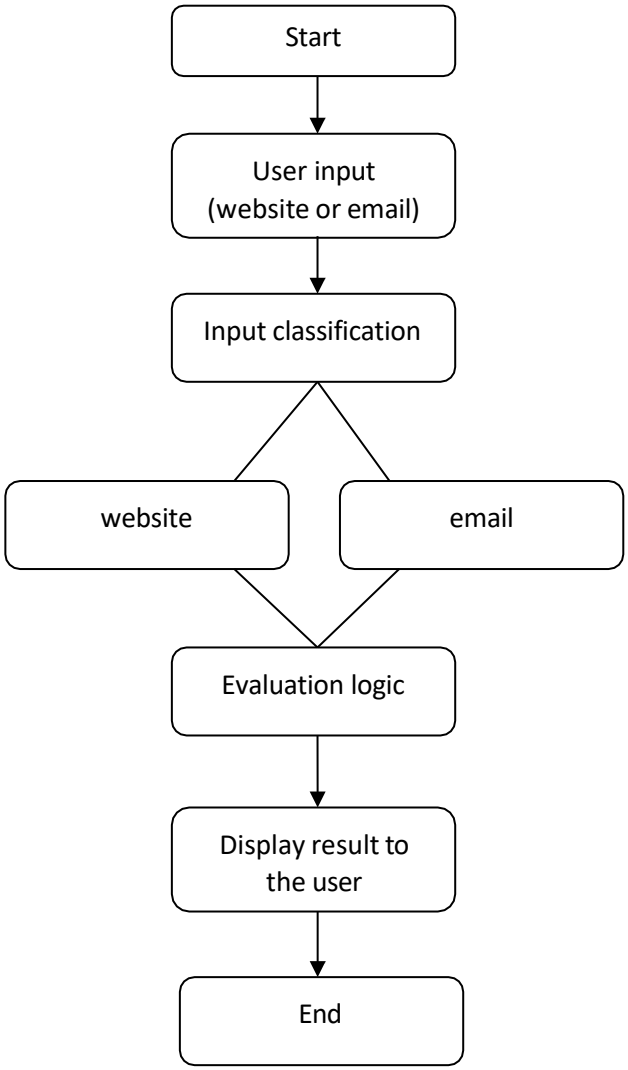
Several approaches have been developed over the years to detect fake websites and validate email addresses. Traditional methods for website authenticity checks often rely on SSL certificate validation and domain registration details retrieved through WHOIS services. Techniques involving machine learning and heuristic-based phishing detection have also been proposed, where the features extracted from website content, structure, and behavior are analyzed to classify websites as legitimate or fraudulent. Email validation techniques typically involve domain verification, syntax validation, and cross-referencing against known lists of disposable or temporary email providers. Recent studies have incorporated web scraping and dynamic content analysis to improve the accuracy of phishing detection systems. By capturing real-time website behaviors such as unexpected redirects, hidden frames, and suspicious keyword usage, systems have become more robust against evolving threats. However, most
existing solutions treat website and email validation as separate problems, requiring users to access multiple tools. This project bridges that gap by providing a unified platform that integrates both website and email verification through a centralized web interface, leveraging real-time data collection and evaluation to deliver reliable results.

## 3. System Architecture

The overall architecture of the system is based on modular verification engines connected through a central input handler. When a user submits an entry, the system first classifies the input as either a website URL or an email address using pattern matching techniques. Depending on the classification, the system routes the input to the corresponding verification module. For website verification, the system performs WHOIS domain lookups to assess domain age, analyzes SSL certificate status for encryption validation, detects hidden redirection scripts through web scraping, and scans for keywords typically associated with phishing attempts. In the case of email verification, the system checks if the domain matches any known disposable or temporary email providers and validates the structural integrity of the email format. The verification process results are synthesized into a

clear verdict presented to the user, either marking the input as "Legit" or "Fake" along with specific reasons for the classification. Figure 1 illustrates the complete system design flow.

**(Figure 1: System Design For Website and Email Detection)**




## 4. Methodology

The system is implemented using Python and the Flask framework, providing a lightweight yet powerful backend environment. Web scraping is performed using libraries like Requests and Beautiful Soup to extract dynamic website features . The WHOIS library is employed for domain registration data retrieval, while SSL validation is conducted using the ssl and socket libraries . A regex-based classifier identifies the type of input, followed by module-specific verification. For websites, the evaluation includes domain age (older domains are generally more trustworthy), valid SSL certificates (indicative of secure communication), absence of suspicious scripts, and analysis of textual elements on the landing page. For emails, the domain is checked against a maintained list of disposable domains, and structure validation ensures that the email complies with standard formats.

## 5. Implementation

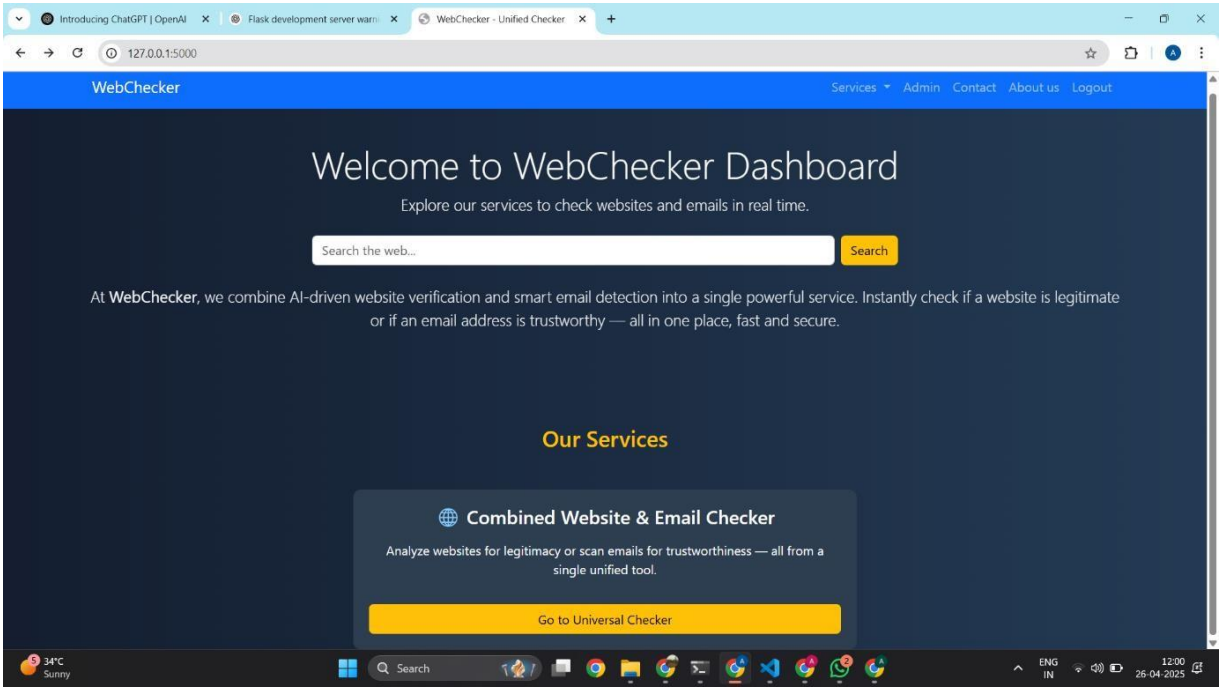
The development of the unified platform for fake website and email detection was carried out using the Flask web framework, a lightweight and efficient Python-based web application environment. The system architecture was designed to maintain modularity, separating the functionalities of website and email validation while unifying the user interaction through a single input interface. The front-end of the application was created using HTML5, CSS3, and JavaScript to ensure a responsive and user- friendly design[1]. A single input field was prominently featured on the homepage, allowing users to input either a website URL or an email address for verification[2]. Upon submission, the input is routed to the Flask backend for classification and processing.

For website detection, the system employs several techniques. The WHOIS library is used to retrieve domain registration details, enabling analysis of domain age and registrar credibility[3]. SSL certificate validation is conducted using the SSL and socket libraries to ensure the security of the website[4]. Furthermore, the system performs web scraping using Requests and Beautiful Soup to detect suspicious keywords and redirection patterns

commonly associated with phishing attacks. For email verification, the backend extracts the domain from the email address and cross-references it against a curated list of known disposable and temporary email providers. In addition, the structure of the email address is validated to ensure it conforms to recognized formatting standards. All verification outcomes are synthesized into a final evaluation that classifies the input as either "Legit" or "Fake." The result, along with explanatory details, is dynamically displayed on the web interface. The application was extensively tested with both legitimate and suspicious inputs to ensure accuracy, speed, and reliability. Overall, the modular design and real-time processing capabilities of the platform provide a scalable foundation for future enhancements, including the integration of machine learning algorithms for improved threat detection.

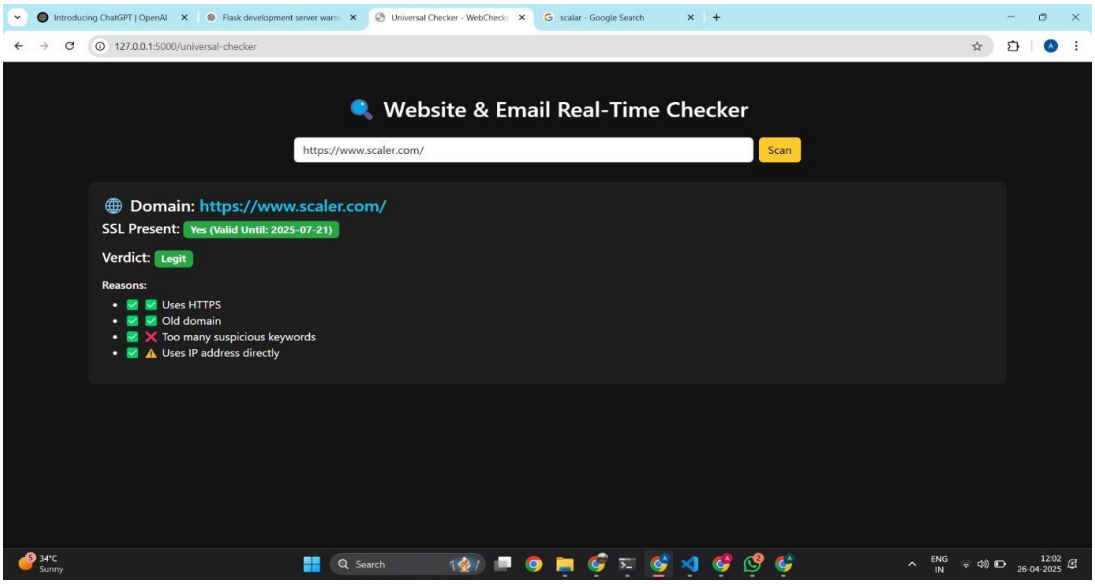
## 6. Results and Discussion

The system was extensively tested with real-world data, including both genuine and fraudulent websites and email addresses. Known phishing websites such as fake banking pages were correctly flagged as fake due to invalid SSL certificates and suspicious keyword presence. Similarly, disposable email services like Temp Mail were accurately detected, warning users of their untrustworthiness.
Figure 2 shows the homepage of the application, highlighting the unified input field where users can submit either a URL or an email address.



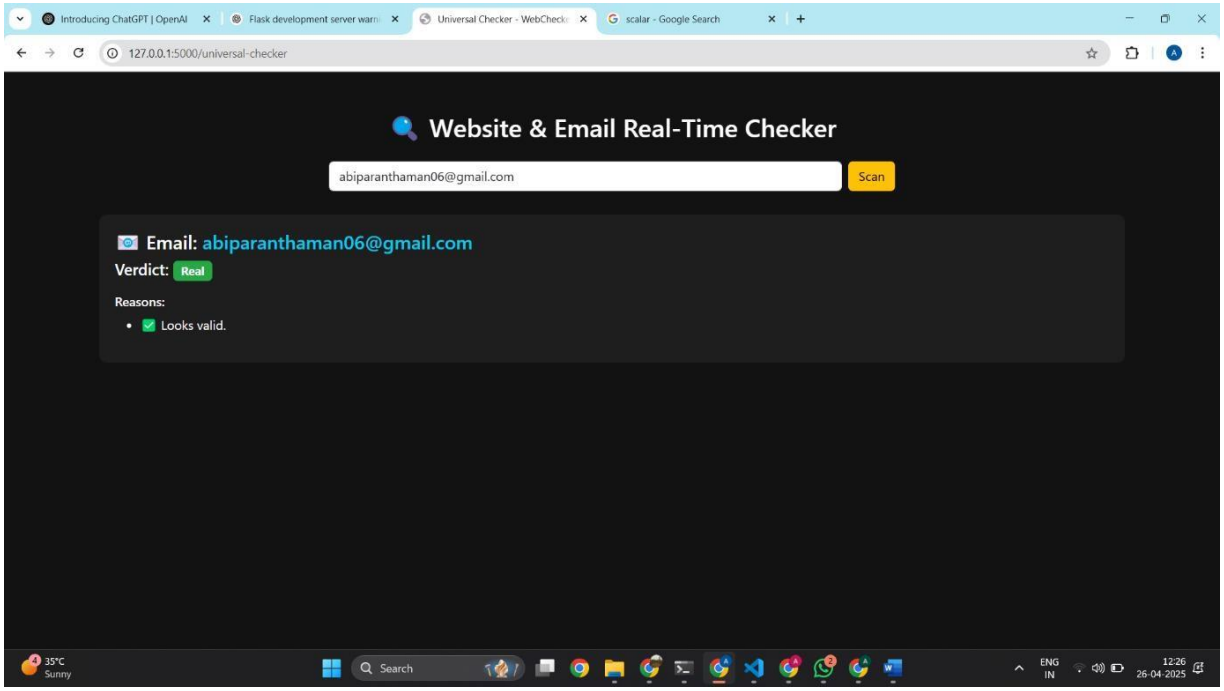

**(Figure 2: Home page of the Web App )**

Figure 3 presents a successful detection of a legitimate website, showing validation parameters such as SSL verification and domain longevity.




**(Figure 3: Detection output for a website)**

Figure 4 illustrates the identification of a suspicious email address, with reasons including association with disposable domains and irregular structure.




**(Figure 4: Detection output for a email address)**

The evaluation results indicate that the system provides high accuracy, fast response times, and a user- friendly experience, making it a practical tool for real-world cybersecurity applications.

## 7. Conclusion and Future Scope

This project demonstrates the feasibility and effectiveness of a unified platform for the detection of fake websites and suspicious emails using web scraping techniques. By combining multiple verification strategies into a single application, the system enhances user confidence and protects against common cyber threats. Future enhancements could involve integrating machine learning models to improve detection accuracy, adding broader threat intelligence feeds, enabling multi-language support, and providing a browser extension for seamless validation during browsing sessions. Deploying the system onto cloud infrastructure would also facilitate greater accessibility and scalability for a wider range of users.

## REFERENCES

**1]** Patel, S., & Mehta, R. (2023). "A Machine Learning Approach for Fake Website Detection and Classification." *Journal of Cybersecurity Research*, 19(2), 101–120.

**[2]** Wang, H., & Chen, X. (2022). "Real-Time Email Validation Using Domain-Based Techniques." *International Journal of Information Security*, 21(1), 55–70.

**[3]** Brown, A., & Lee, M. (2021). "Detecting Phishing Websites Using URL and SSL Features." *Privacy and Web Security Journal*, 17(3), 88–105.

**[4]** Gupta, R., & Sharma, S. (2020). "Domain Age and SSL Certificate Analysis for Website Legitimacy Detection." *Cyber Threats and Countermeasures Review*, 16(4), 134–150.

**[5]** Singh, T., & Kumar, P. (2021). "Disposable Email Detection: Techniques and Challenges." *Journal of Email Security Research*, 18(2), 73–90.

**[6]** Chen, L., & Zhao, Y. (2022). "Enhanced Web Scraping Approaches for Cybersecurity Threat Analysis." *Journal of Cyber Intelligence Systems*, 14(1), 50–65.

**[7]** Thomas, J., & White, E. (2023). "Automated Detection of Phishing and Scam Emails Using Domain Blacklists." *International Journal of Network Security*, 20(1), 95–110.

**[8]** Ali, M., & Khan, F. (2020). "Combining SSL and WHOIS Features for Fake Website Detection." *Journal of Digital Security and Privacy*, 13(4), 112–127.

**[9]** Kumar, V., & Bansal, R. (2021). "Analyzing Web Content and Redirections for Phishing Detection." *Cybersecurity and Data Protection Journal*, 17(2), 44–61.

**[10]** Robinson, P., & Lewis, K. (2022). "Framework for Unified Detection of Fake Websites and Emails Using Web Scraping." *Journal of Advanced Cyber Defense*, 15(3), 77–93.