

**Web Enumeration Tool**

Content Talbes

1. Introduction 7

2. Aim 7

3. Objectives 7

4. Problem Statement 8

5. Features and Functionalities 8

I. Web Technology Identification: 9

II. Subdomain Enumeration: 9

III. WebArchive URL Retrieval: 10

IV. Hidden Directory Discovery: 11

V. CVE Scan: 11

VI. Whois Number Lookup: 11

VII. Real-time Progress and Reporting: 11

VIII. User-Friendly Interface: 12

IX. Socket.IO Integration: 12

6. Scope of the Project 13

7. Development Methodology 14

8. Agile Methodology and Waterfall Methodology 14

9. Tools and Technology 14

I. Scanning and Validation Tools: 15

10. Design 15

11. Architecture 15

II. Class Diagram 15

12. Runtime Architecture 16

13. Coding and Implementation 21

I. Setting Up the Web Application: 21

II. Scan Function (run\_command): 22

III. Creating Web Pages: 22

IV. Real-Time Scanning: 23

V. Running Scan Commands: 23

 Different commands are executed to perform scans (e.g., checking technology, finding subdomains) using external tools. 23

VI. Running the Web Application: 23

 The web application is started so users can access it through a web browser. 23

VII. index.html (HTML Template) 24

VIII. Styling: 24

IX. ASCII Art 24

 ASCII art for visual appeal. 24

X. User Input Section: 25

 Users can enter a target domain and initiate a scan. 25

XI. Progress and Navigation: 25

 Sections for displaying scan progress and navigation links. 25

XII. Result Containers: 25

 Div elements for displaying scan results for various categories. 25

XIII. JavaScript and Socket.IO: 26

 JavaScript code includes Socket.IO events for real-time updates based on scan results and status. 26

14. Testing 27

15. Version Control 30

Conclusion 30

16. Reference 33

# Introduction

Web enumeration is an important step in ethical hacking and cyber security. It involves gathering information about a web application or website that is the target to find security risks and vulnerabilities. The goal of this project is to make a web enumeration tool that automates different enumeration tasks. This will make it easier for cybersecurity professionals and ethical hackers to do thorough assessments of target URLs. This can also help people who aren't proficient with technology learn more about their website.

# Aim

The goal of this project is to make a web enumeration tool that makes it easier and faster to gather information about a specific website or web application. The main goal of the tool is to help security professionals find any vulnerabilities, weaknesses, or security risks that might be linked to the target URL. By automating the process of web enumeration, the tool aims to improve the speed, accuracy, and completeness of security assessments. This will help improve the security of web-based systems. By cutting down on the time needed for enumeration, we can find bugs faster and make our website safer.

# Objectives

The main goal of the tool is to make a powerful and easy-to-use web enumeration tool that automates the process of getting information about web applications or websites that are being targeted. The tool uses various enumeration techniques quickly and well, reducing the amount of work that cybersecurity professionals have to do by hand. The tool tries to find out exactly what programming languages, frameworks, and content management systems are being used in the target URL. The tool helps with vulnerability assessment and risk reduction by looking for known exploits and holes in these technologies. It will also use effective fuzzing techniques to find hidden or unprotected resources, like directories and files, that could be security risks.

Additionally, the tool will perform DNS enumeration to retrieve DNS records and port scanning to

identify open ports and associated services, providing insights into the network configuration.

Retrieving WHOIS records and SSL certificate information further enhances the understanding of domain ownership and security measures in place. To facilitate easy analysis and reporting, the tool will generate a comprehensive PDF report summarizing the collected information.

# Problem Statement

The manual process of web enumeration for security assessments is time-consuming, labor- intensive, and prone to human errors, hindering the efficiency and accuracy of identifying potential vulnerabilities and security risks in web applications. Security professionals often face challenges in detecting the underlying technologies, searching for known exploits, conducting comprehensive fuzzing, and retrieving essential information like DNS records and SSL certificates. Furthermore, the lack of an automated and standardized approach for web enumeration leads to inconsistencies in results and difficulties in generating comprehensive reports.

Our goal is to develop a comprehensive and automated web enumeration tool that surpasses manual enumeration processes. Our tool should ensure accuracy, generate detailed reports, and help detect vulnerabilities while enhancing web application security.

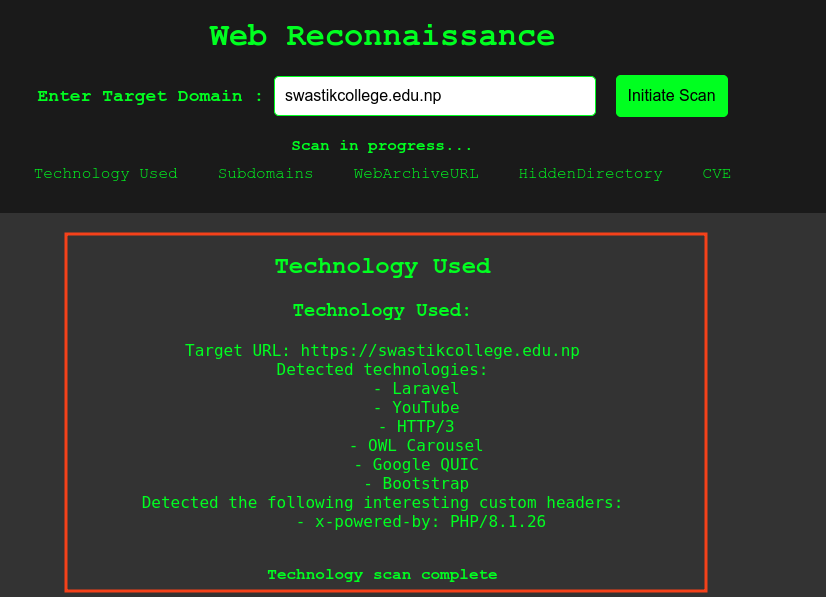
# Features and Functionalities

Features are distinctive characteristics or capabilities of software or tools that provide specific functionalities to users. Functionalities refer to the tasks and operations that the tool can perform based on its features. The web enumeration tool will offer the following features and functionalities:

## Web Technology Identification:

Identify and report the technologies used by the target web application (e.g., web server,

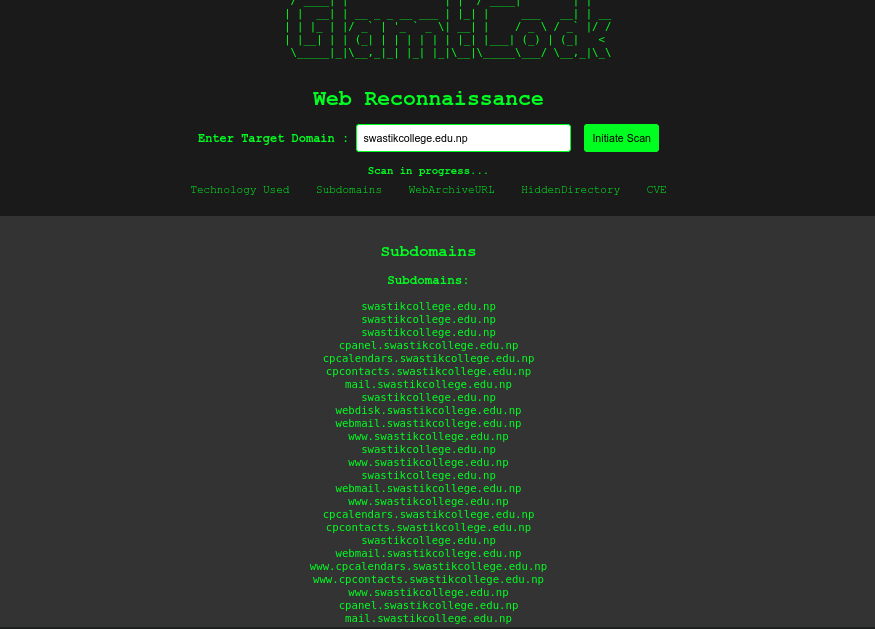
CMS, programming languages, etc.). Provide a detailed breakdown of the technologies discovered.



*Fig: Technology Information*

## Subdomain Enumeration:

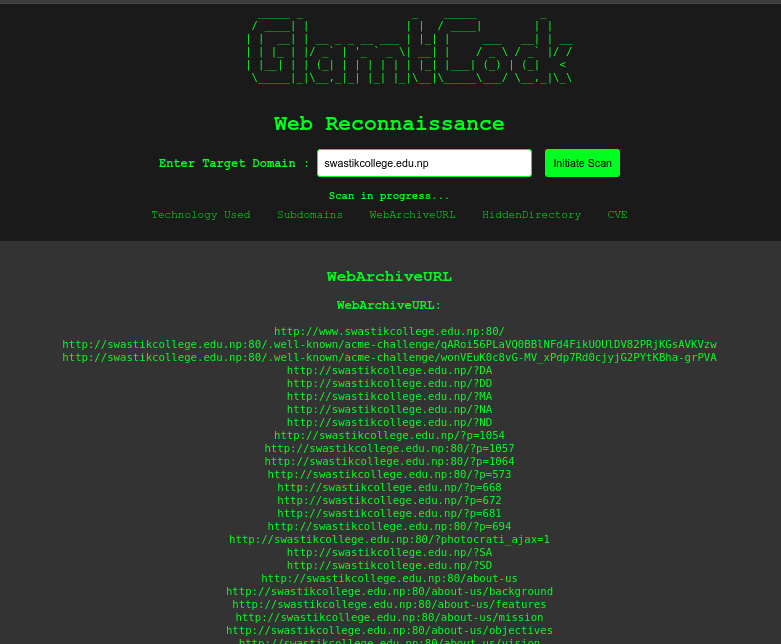
Enumerate subdomains associated with the target domain. Display a list of discovered subdomains and their associated information.



*Fig: List of Subdomain*

## WebArchive URL Retrieval:

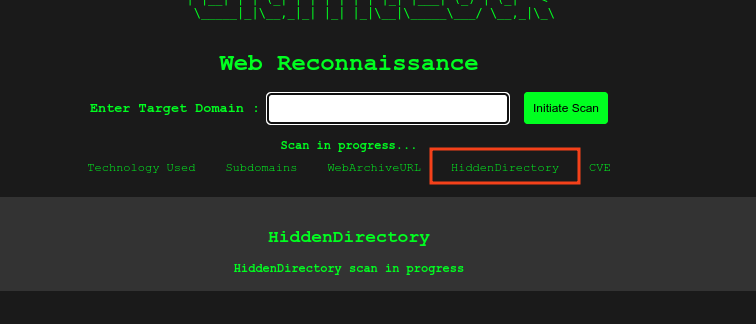
Retrieve and present URLs from the WebArchive for the target domain.Offer insights into historical web pages associated with the domain.



*Fig: WebArchive Url*

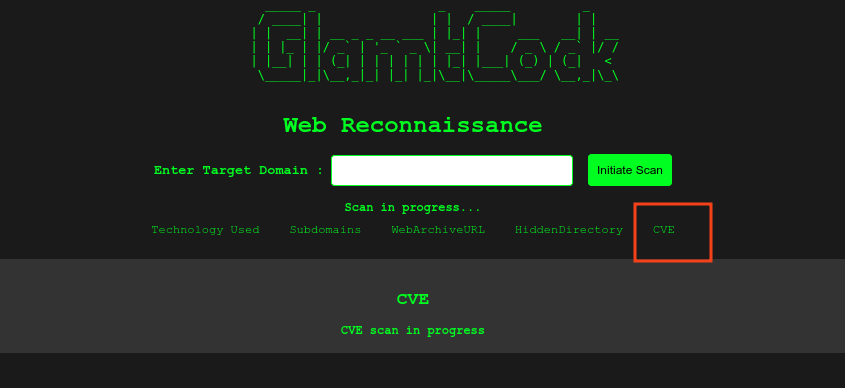
## Hidden Directory Discovery:

Discover hidden directories on the target web server. Provide a list of potentially sensitive ornon-public directories.



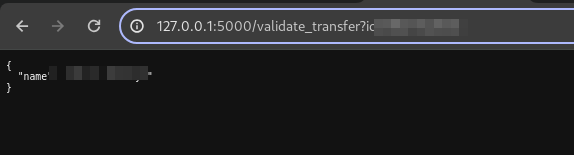
## CVE Scan:

It will search for known exploits and vulnerabilities associated with the identifie technologies by accessing relevant databases and repositories.



## Whois Number Lookup:

Implement a feature to perform a Whois lookup for a given phone number. Display relevant information about the phone number's registration. (Paid)

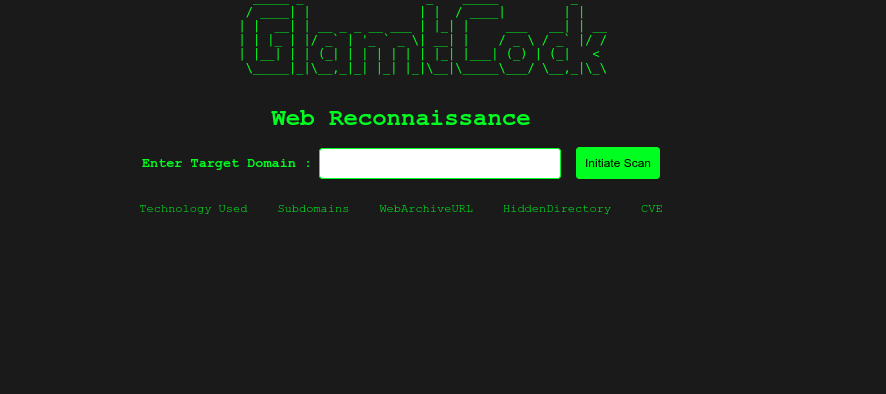


## Real-time Progress and Reporting:

Provide a real-time progress update for each scanning category. Display detailed results for each category, including success or failure status.

## User-Friendly Interface:

Develop a user-friendly web interface for initiating scans and viewing results.



Ensure ease of use and accessibility for users with minimal technical knowledge.

## Socket.IO Integration:

Implement Socket.IO for real-time communication between the server and the client.

Enable instant updates and progress tracking for users during scans.

## Requirements

**To use the Web Reconnaissance Tool, make sure you have the following:**

* Python Installed: Ensure Python is installed on your computer.
* Required Libraries: Install necessary Python libraries using the command: pip install Flask Flask-SocketIO requests.

**External Tools: Install additional tools by following these steps:**

Install **subfinder** for subdomain enumeration.

Install **Dirb** for hidden directory scanning.

Install **Nuclei** for CVE scanning.

Install **webtech**for technology detection.

Install **Waybackurls** for WebArchive URL retrieval.

Run the Tool: Launch the tool by running python app.py in your terminal.

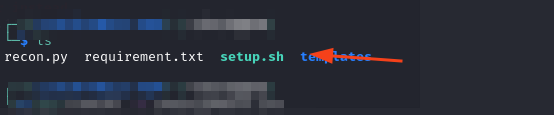
Access Through Browser: Open a web browser and go to the tool's address (usually http://127.0.0.1:5000).

Enter Target Domain: Input the domain you want to assess and start the scan.

Real-time Updates: View live updates on subdomains, technologies, WebArchive URLs, hidden directories, and CVE scan results.

## Run the Tool:

**Execute the setup script using: bash setup.sh.**

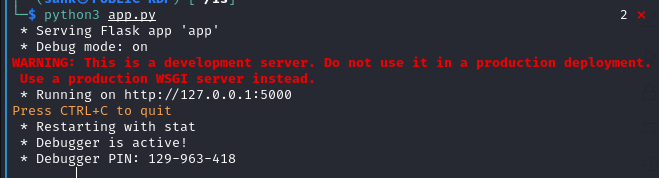


**Install Dependencies:**

Ensure required dependencies are installed by running: *pip install -r requirements.txt.*

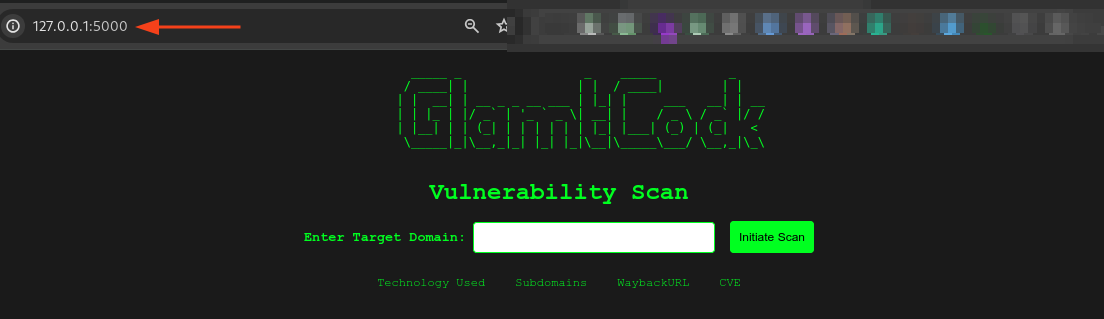


Start the tool by running *python3 app.py* in your terminal.



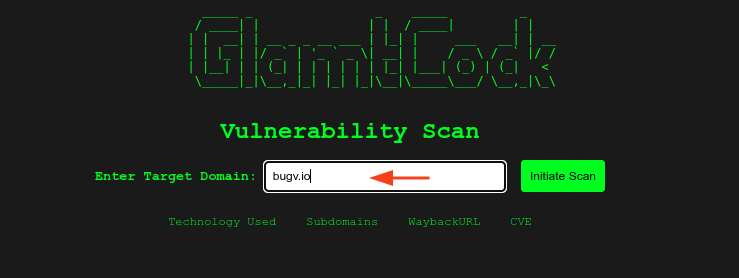
**Access Through Browser:**

Open a web browser and go to the tool's address (usually <http://127.0.0.1:5000).>



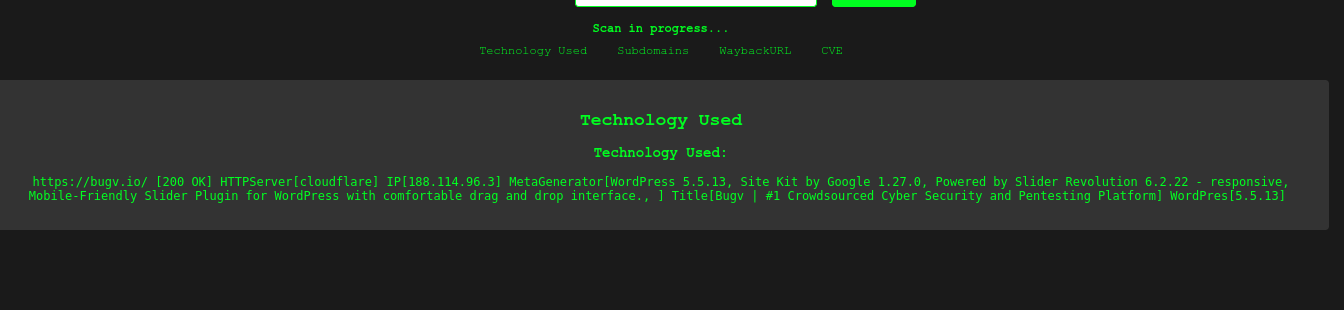
**Enter Target Domain:**

Input the domain you want to assess and kick off the scan



**Real-time Updates:**

Witness live updates on subdomains, technologies, WebArchive URLs, hidden directories, and CVE scan results.



# Scope of the Project

The scope of the web enumeration tool is to provide a comprehensive and efficient solution for conducting web application security assessments. The tool aims to assist security professionals in identifying potential vulnerabilities, misconfigurations, and security weaknesses within web applications. It will offer a range of features and functionalities, including technology detection, exploit search, directory, subdomain, file fuzzing, DNS enumeration, port scanning, WHOIS record retrieval, and url WebArchive UR.

The tool is designed to work with various types of URLs, accommodating both HTTP and HTTPS protocols, making it versatile for different web applications. Leveraging webtech and relevant APIs, it will accurately detect the technologies used in the target web application, allowing for targeted exploit

searches based on the identified technologies. Systematic fuzzing for directories, subdomains, and files will help uncover hidden resources and potential entry points for attackers.

# Development Methodology

The development methodology for the web enumeration tool will follow an iterative and incremental approach, combining elements of Agile and Waterfall methodologies. The goal is to ensure efficient development, frequent feedback, and continuous improvement while maintaining a structured and well-documented process.

# Agile Methodology and Waterfall Methodology

Agile methodology focuses on iterative and incremental development, collaboration, and responsiveness to changing requirements. Key aspects of Agile that have been applied to the web enumeration tool include Iterative Development and Continuous Improvement.

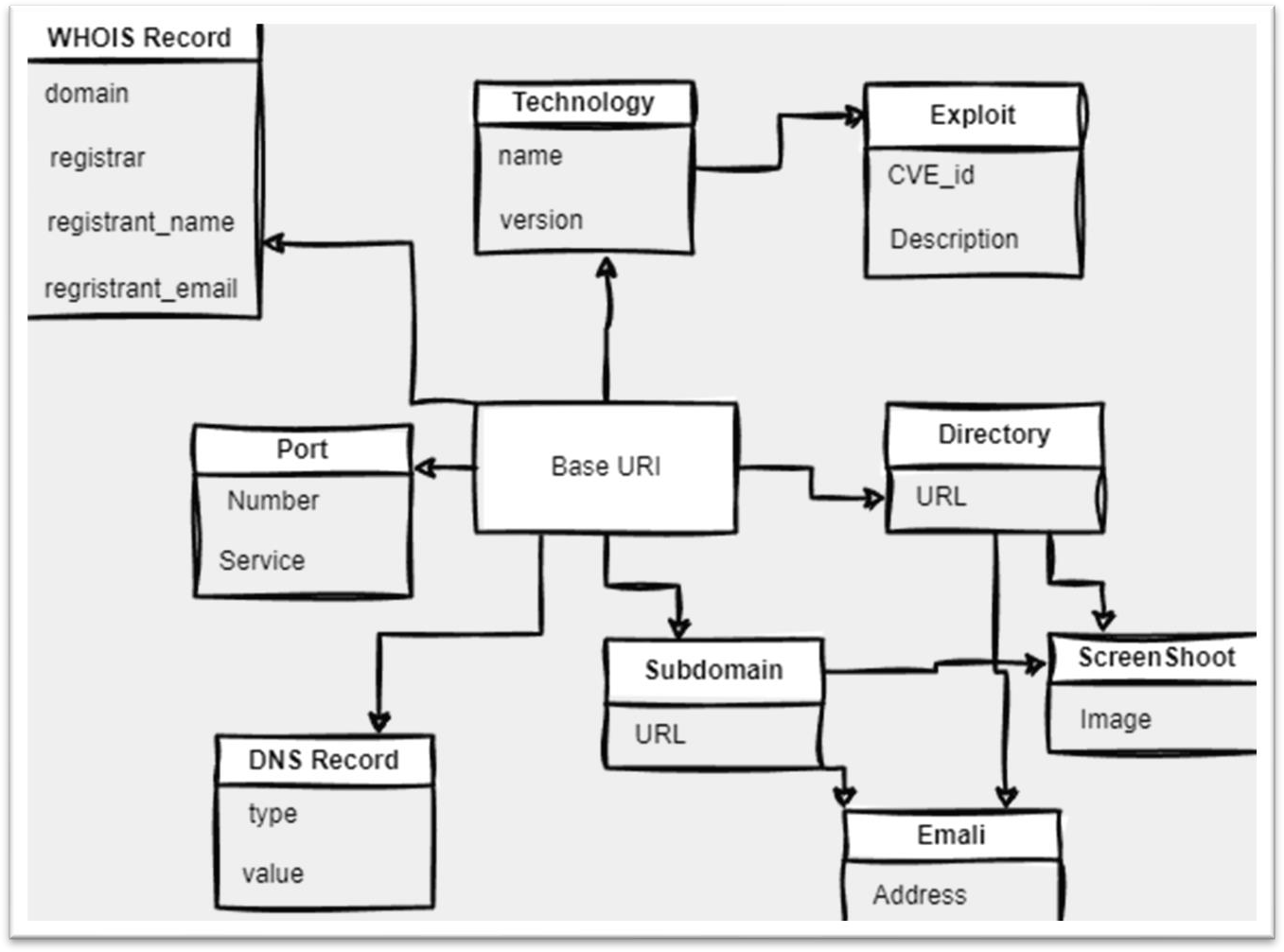
Waterfall methodology is a linear and structured approach to development, where each phase is completed before moving to the next. For the web enumeration tool, certain aspects of the Waterfall methodology have been incorporated to ensure a well-planned and documented development process. i.e., Requirements Gathering, Design and Planning, and testing.

# Tools and Technology

The web enumeration tool is developed using Python as the primary programming language, complemented by essential libraries and modules such as Requests, Builtwith, Selenium, dns.resolver, Whois, SSL, and ReportLab. Version control is managed using Git, while Visual Studio code is used as IDEs. These tools and technologies ensure an efficient and effective development process, resulting in a reliable and valuable web enumeration tool.

# Architecture

## Class Diagram

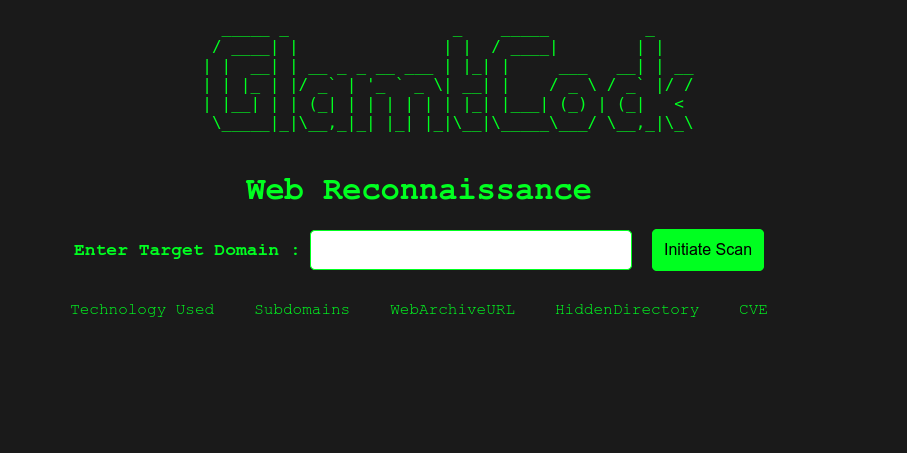


*Figure 1: Class Diagram of Web enum*

The Class Diagram for the web enumeration tool represents the classes, attributes, and methods involved in the tool's implementation.

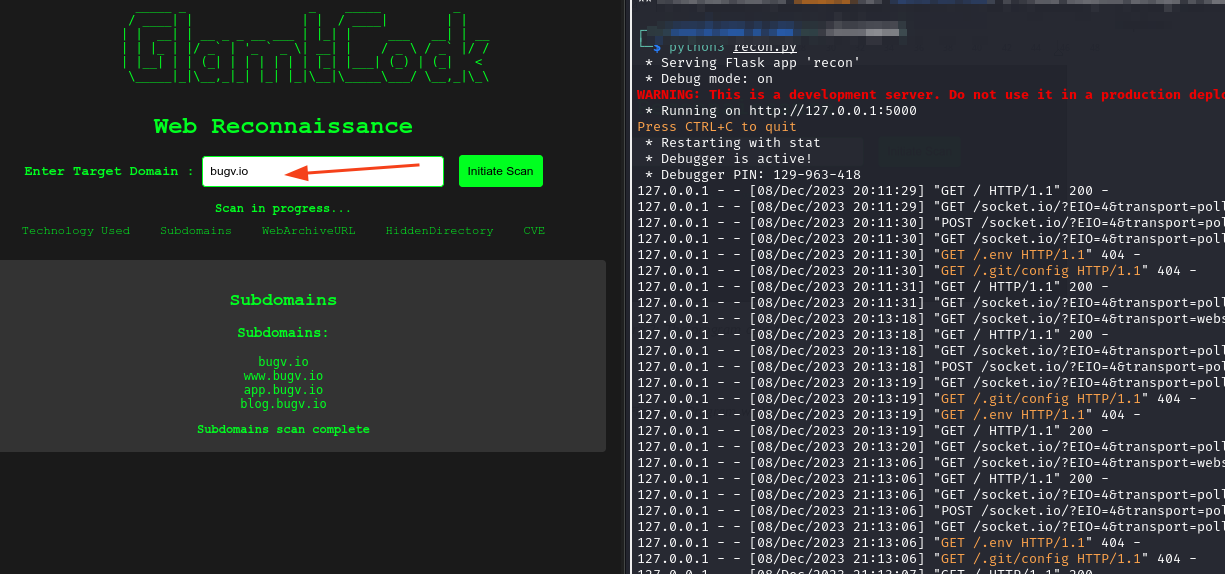
# Runtime Architecture

The Dynamic Architecture of the web enumeration tool focuses on the runtime behavior and interactions between various components during the execution of the tool. It describes how different modules and functionalities collaborate to achieve the intended functionality of the tool.

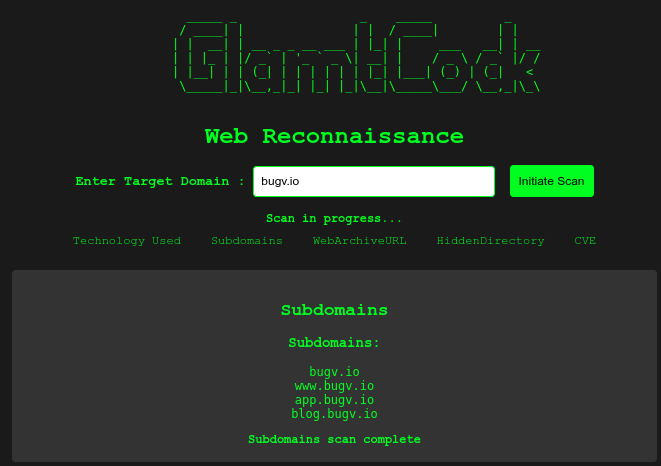


*Figure 2: The program asks for a target.*

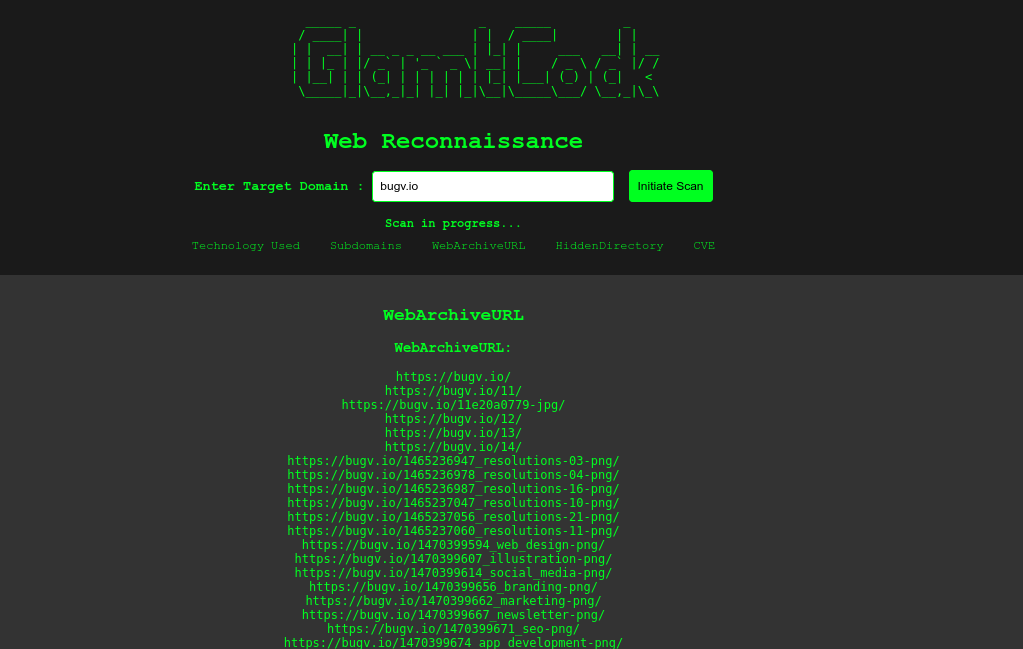
***Note: Before Testing On a web First Please Take a Permission From Site Owner***



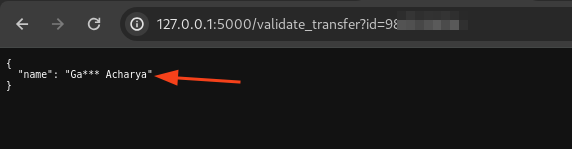
*Figure 3: Program while running.*



*Figure 4: Program while running searching for subdomain.*



* **We Added a New feature to check the Unknow phone number**

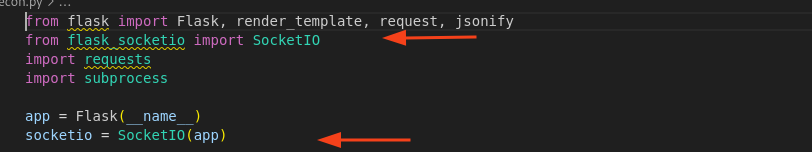


# Coding and Implementation

## Setting Up the Web Application:

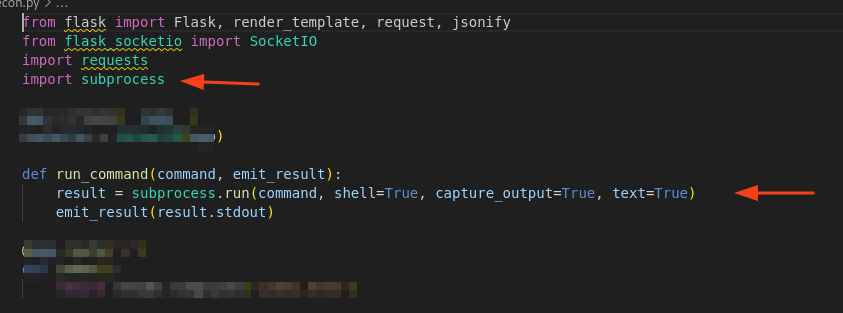
### We're using a Python framework called Flask to create a web application.

### This web app can do things in real-time thanks to another tool called Socket.IO.



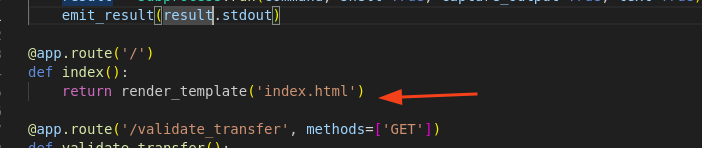
## Scan Function (run\_command):

* We have a function (run\_command) that can run commands in the computer's shell and capture the outpu



*Fig: Scan Function*

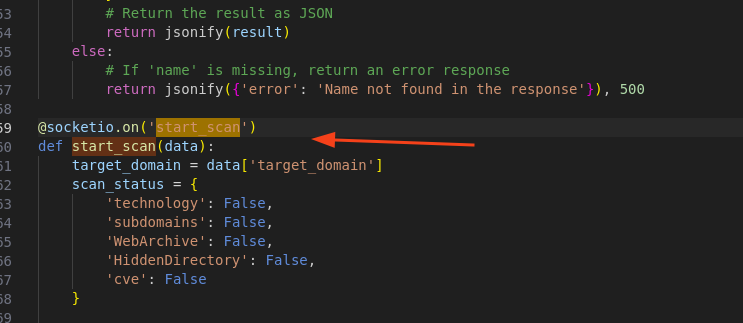
## Creating Web Pages:

* We've set up a basic web page (index.html) where users can interact with the tool

*Fig: Creating Web Pages*

## Real-Time Scanning:

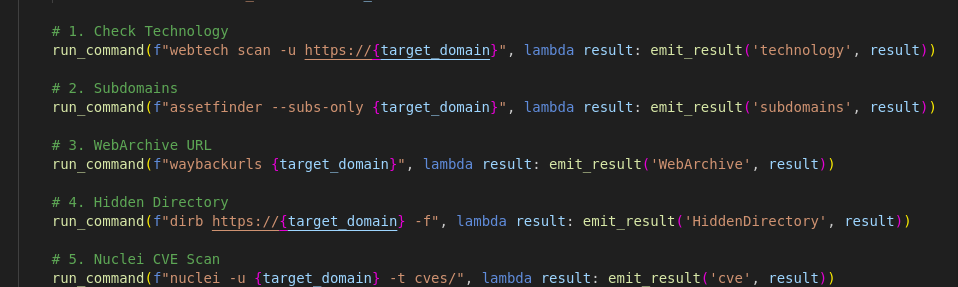
* Users can initiate different scans on a target domain, like checking what technologies it uses, finding subdomains, and more.
* Results are sent to the user's web page in real-time as each scan completes.



*Fig:Real-Time Scanning*

## Running Scan Commands:

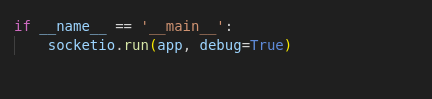
## Different commands are executed to perform scans (e.g., checking technology, finding subdomains) using external tools.



*Fig: Running Scan Commands*

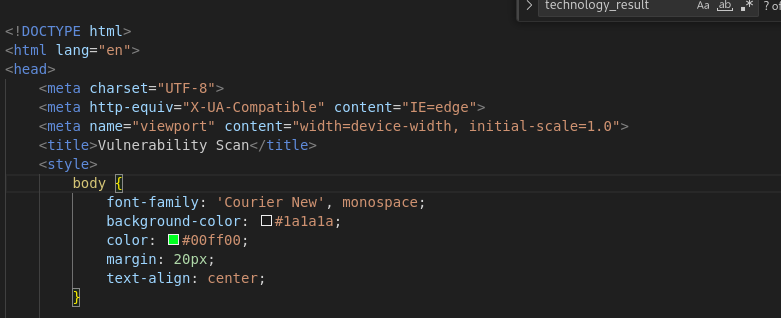
## Running the Web Application:

## The web application is started so users can access it through a web browser.



## index.html (HTML Template)

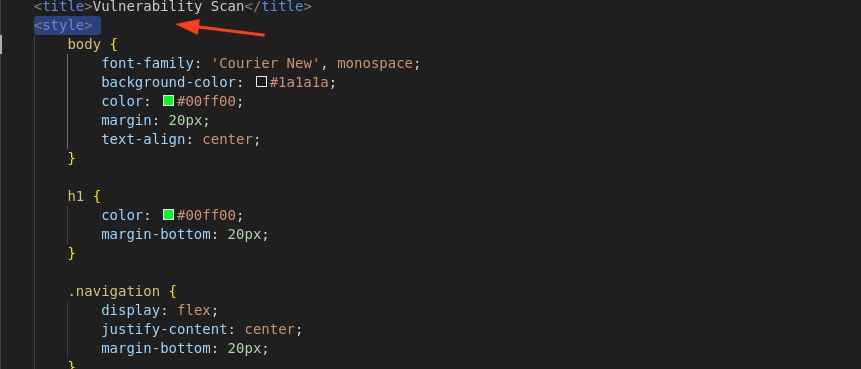
* Document Structure: The HTML document includes metadata and a body for content.



*Fig: Index.html file*

## Styling:

* CSS styles are defined for the page.



*Fig: Styling*

## ASCII Art

## ASCII art for visual appeal.



*Fig: ASCII Art*

## User Input Section:

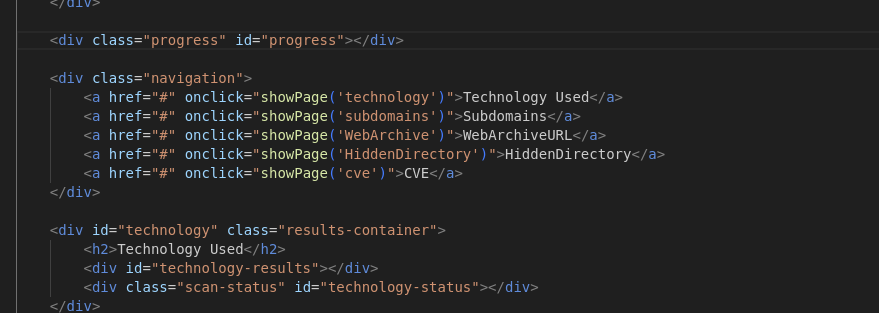
## Users can enter a target domain and initiate a scan.



*Fig: User Input Section*

## Progress and Navigation:

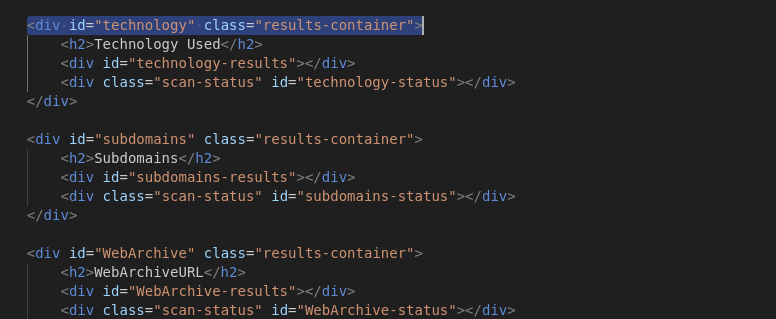
## Sections for displaying scan progress and navigation links.



*Fig: Progress and Navigation*

## Result Containers:

## Div elements for displaying scan results for various categories.



*Fig:Result Containers*

## JavaScript and Socket.IO:

## JavaScript code includes Socket.IO events for real-time updates based on scan results and status.

# Version Control

Version control is a system that manages changes to files and directories over time. It allows multiple users to collaborate on a project, track changes, and keep a history of the development process. For the version control and commit history of the web enumeration tool can visit this <add your Github> .

GitHub Link:

# Conclusion

In conclusion, our web enumeration tool is a powerful and versatile asset for gathering vital information about a target website or domain. Through the implementation of various key functionalities, we can effectively perform DNS enumeration, port scanning, WHOIS lookups, SSL certificate analysis, technology detection, fuzzing of subdomains, directories, and files, as well as email

harvesting. The tool compiles all the collected data into a well-structured PDF report, presenting DNS records, port scan results, technologies used, discovered subdomains, directories, files, and more.

Security professionals can utilize this comprehensive report to assess potential risks, identify vulnerabilities, and enhance the target's overall security. However, it is crucial to use this tool responsibly and ethically, ensuring it is solely employed for legitimate security testing and not for any malicious purposes.

# Reference

*Welcome to Python.org*. (2023, August 2). Python.org; Python.org. <https://www.python.org/>

*Automating The Web Application Reconnaissance Process -All In One Recon*. (n.d.). Retrieved August 3, 2023, from <https://ijcrt.org/papers/IJCRT2205171.pdf>

Reichinger, Kurt & Baumgartner, Robert. (2023). SEMNUM RDF Vocabulary Specification v. 01.

Reichinger, K., & Baumgartner, R. (n.d.). *INTRODUCING ENUM TO THE SEMANTIC WEB*.

Retrieved August 3, 2023, from <https://www.dbai.tuwien.ac.at/proj/semnum/documents/494-048.pdf>

*Webdiscover Automate The Web Enumeration Process And Search For Exploits*. (2022, January 2). GeeksforGeeks; GeeksforGeeks. [https://www.geeksforgeeks.org/webdiscover-](https://www.geeksforgeeks.org/webdiscover-automate-the-web-enumeration-process-and-search-for-exploits/) [automate-the-web-enumeration-process-and-search-for-exploits/](https://www.geeksforgeeks.org/webdiscover-automate-the-web-enumeration-process-and-search-for-exploits/)

*OWASP Automated Threat Handbook Web Applications*. (n.d.). [https://owasp.org/www-pdf-](https://owasp.org/www-pdf-archive/Automated-threat-handbook.pdf) [archive/Automated-threat-handbook.pdf](https://owasp.org/www-pdf-archive/Automated-threat-handbook.pdf)

Singh, N., Bajpai, P., & Ranjan, R. (2022). *Web Application Penetration Automation Testing Tool - WAPATT*. *8*. <https://ijirt.org/master/publishedpaper/IJIRT154789_PAPER.pdf>

Oliveau, A. (n.d.). *OFFENSIVE ENUMERATION*. Retrieved August 3, 2023, from <https://competitivecyber.club/past-talks/MasonCC-S20-Offensive-Enumeration.pdf>

AUTOMATION IN MANUAL PENETRATION TESTING USING BASH SHELL SCRIPT.

(2023). *International Research Journal of Modernization in Engineering Technology and Science*. <https://doi.org/10.56726/irjmets40392>