**WEB APPLICATION VUULNERABILITY SCANNER**

**Scope:**

* Automate the detection of common web vulnerabilities like **XSS** and **SQL** Injection**.**
* Crawls web forms, injects malicious payloads, and checks for potential vulnerabilities**.**
* Helps in identifyinglow-hanging security flawsin **web applications.**
* Designed for educationalandPentesting demo purposes**.**

**Tools & Technologies Used:**

| **Tool/Library** | **Purpose** |
| --- | --- |
| **Python 3.x** | Programming Language |
| **requests** | HTTP requests to target web applications |
| **BeautifulSoup4** | HTML parsing and form crawling |
| **re (regex module)** | Pattern matching for vulnerability detection |
| **logging** | Logging vulnerabilities to file |
| **Jupyter Notebook / PyCharm** | Development & Testing IDEs |

**Script/Code Explanation**

**Major Functions:**

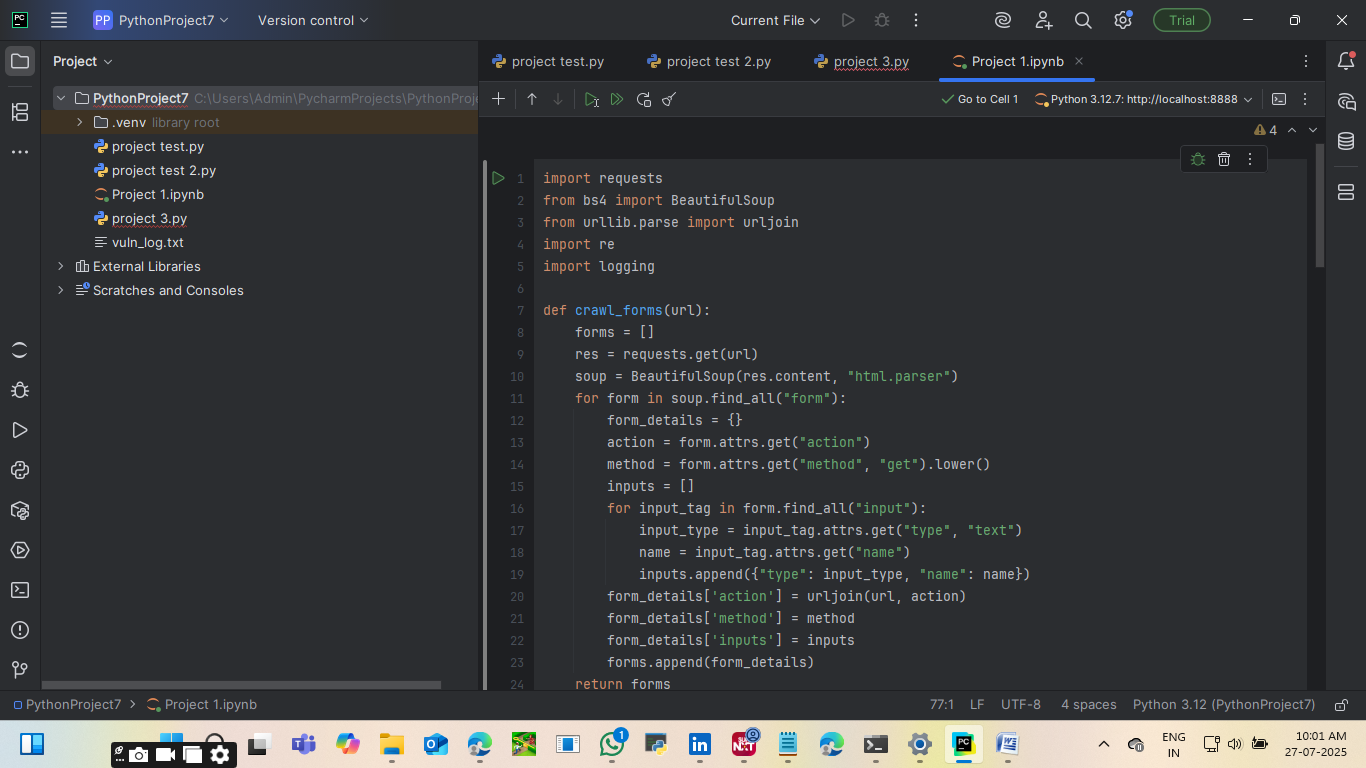
* **Crawl\_Forms(url):** Crawls the given URL and extracts all form actions, methods, and input fields.
* **Submit\_Form(form, url, payload):** Submits the form with malicious payloads (XSS, SQLi).
* **Detect\_Vulnerability(response, payload\_type):** Checks the response content for signs of vulnerability.
* **Log\_Vulnerability(url, payload, vuln\_type):** Logs found vulnerabilities into a log file.
* **Run\_Scanner(base\_url):** Orchestrates the entire scanning process for the given base URL.

### Attack Scenarios Covered:

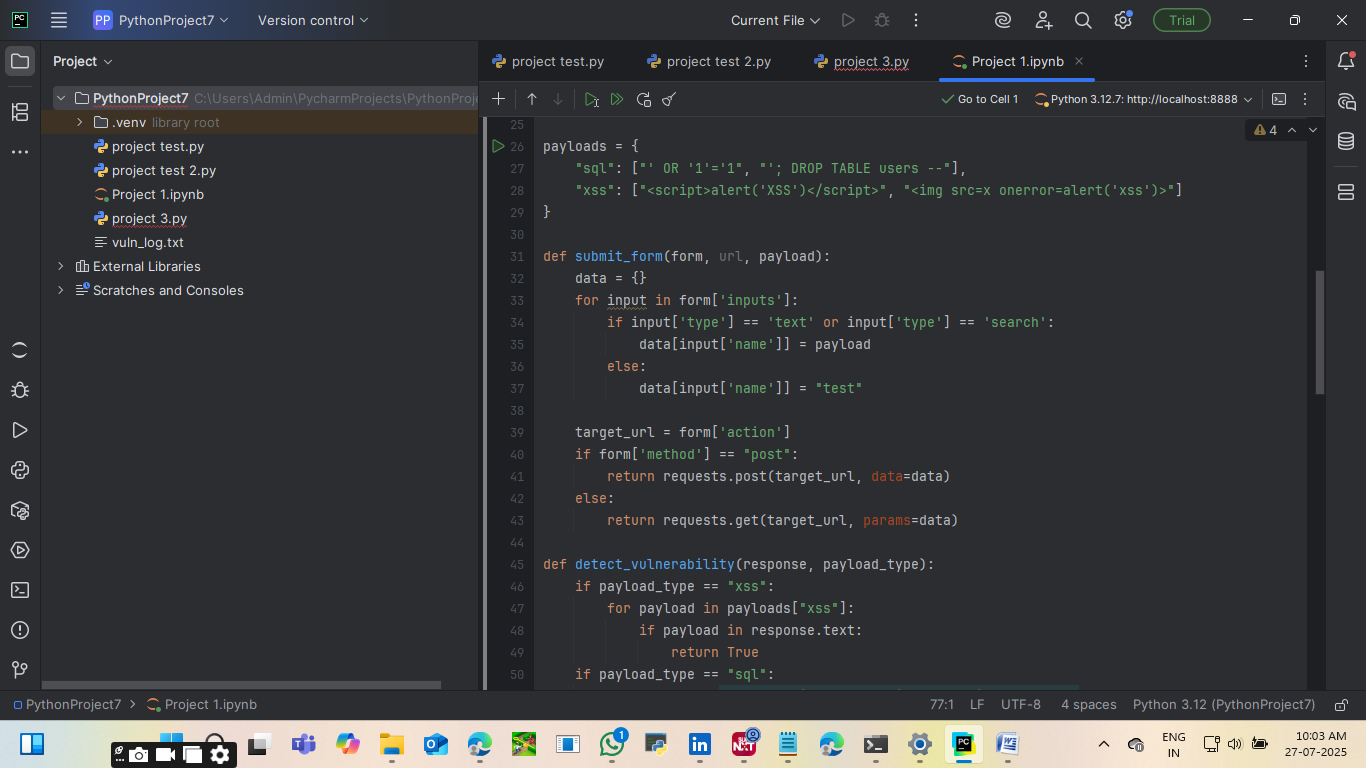
* **SQL Injection Payloads:**
  + ' OR '1'='1
  + '; DROP TABLE users --
* **Cross-Site Scripting (XSS) Payloads:**
  + <script>alert('XSS')</script>
  + <img src=x onerror=alert('xss')>

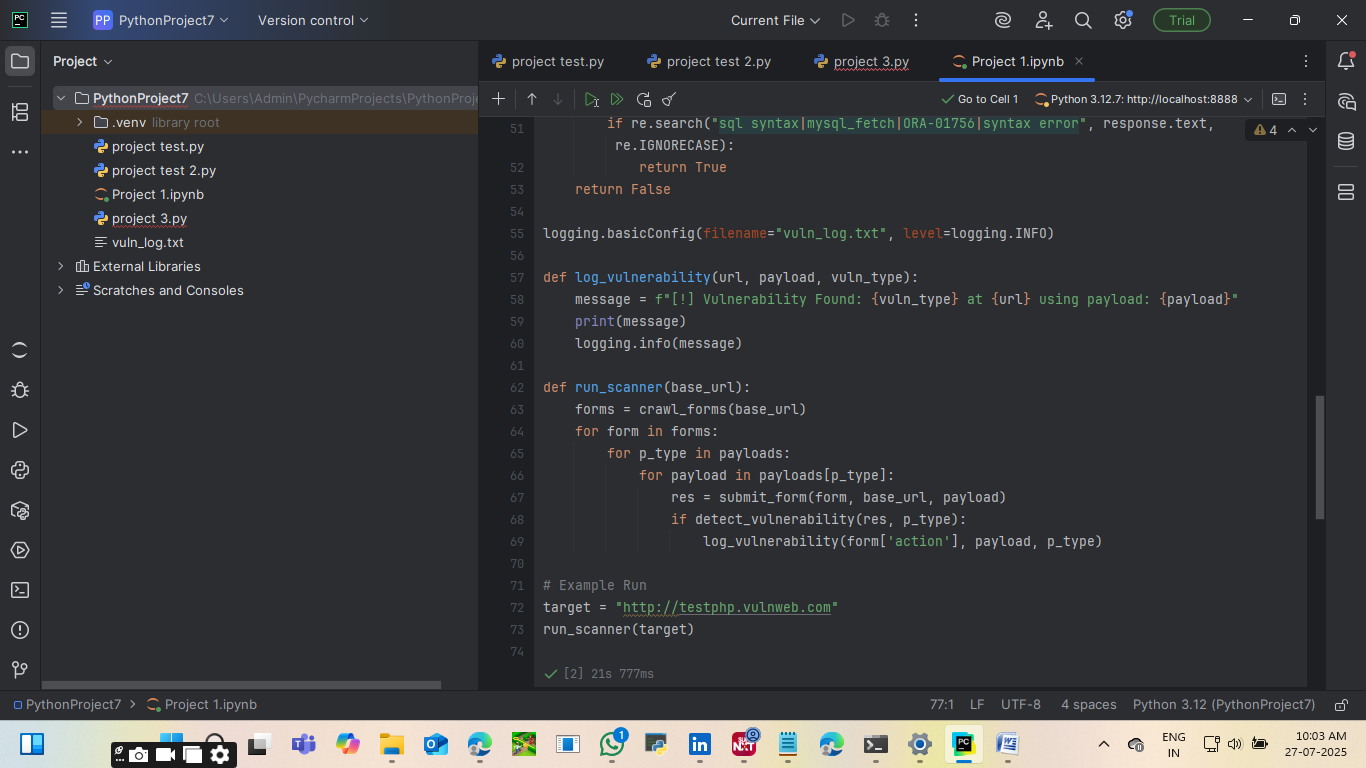
**Screenshots:**

Code Execution Screenshot (Jupyter/PyCharm),

****

vuln\_log.txt file screenshot with log entries

****

****

**Expected Output:**

The script will,

* Print vulnerable endpoints to console.
* Log vulnerabilities into a file named **vuln\_log.txt**.
* Example Output in console:

****

* [!] Vulnerability Found: xss at http://testphp.vulnweb.com/process.php using payload: <script>alert('XSS')</script>
* [!] Vulnerability Found: sql at http://testphp.vulnweb.com/process.php using payload: ' OR '1'='1

**Conclusion:**

* Successfully built a basic yet effective web scanner for detecting XSS and SQL Injection vulnerabilities.
* This project demonstrates the practical approach of form-based vulnerability scanning using Python automation.
* Can be extended further for CSRF, File Upload vulnerabilities, etc.
* Recommended to use this scanner only in safe, legal testing environments (e.g., testphp.vulnweb.com).

**Future Scope (Optional but Professional):**

* Add Authentication Handling (login forms/session-based crawling).
* Support for CSRF Detection.
* Advanced payload fuzzing (automation of large payload lists).
* Multi-threaded scanning for speed optimization.
* Result visualization via a Dashboard Interface.