**Web Application Vulnerability Scanner Report**

**Introduction:-**

This project aims to develop a basic Web Application Vulnerability Scanner capable of detecting common security flaws like Cross-Site Scripting (XSS) and SQL Injection (SQLi) in web applications. The scanner automates crawling, input injection, and response analysis to identify vulnerabilities.

**Abstract:-**

Web applications are frequently targeted by attackers exploiting vulnerabilities such as XSS and SQLi. This project builds a lightweight vulnerability scanner using Python and Flask, integrating crawling, payload injection, and pattern matching to detect common issues. The scanner features a user-friendly web interface for managing scans and viewing detailed reports.

**Tools Used:-**

Python: Core language for developing the scanner and web interface.

Requests: For HTTP requests during crawling and injection.

BeautifulSoup: HTML parsing and form extraction.

Flask: Web framework to build the UI and serve scan results.

Regex/Pattern Matching: For detecting vulnerability indicators in HTTP responses.

**Steps Involved in Building the Project:-**

Setup: Initialize Python environment and install required libraries.

Crawler Implementation: Use Requests and BeautifulSoup to recursively crawl the target domain for URLs and HTML forms.

Payload Injection: Automate injection of predefined payloads (e.g., <script>alert(1)</script> for XSS) into form fields and URL parameters.

Response Analysis: Parse HTTP responses to detect reflections of payloads or error messages indicating vulnerabilities using regex and string matching.

Web UI: Create a Flask-based interface allowing users to input target URLs, start scans, and view vulnerability reports with evidence and severity.

Reporting: Aggregate findings and display them clearly on the web interface for easy interpretation.

**Conclusion:-**

The project demonstrates the creation of a functional vulnerability scanner that automates the detection of common web application flaws. Although simplified, this tool provides a foundation for understanding vulnerability assessment automation. Future enhancements could include support for additional vulnerability types, better crawling heuristics, authentication handling, and exportable reports.