Internship Project Report: Web Vulnerability Scanner

Abstract:
This project implements a simple **web application vulnerability scanner** that detects two common issues from the OWASP Top-10 list: **Cross-Site Scripting (XSS)** and **SQL Injection (SQLi)**. The scanner was built using Python, GitHub Actions, and optional Flask components, and is designed purely for **educational and authorized security testing**.

Introduction:
Web applications are often vulnerable to injection attacks and poor input validation. Security analysts use scanners to identify these flaws early. This project demonstrates the design of a basic scanner as part of a cybersecurity internship.

Tools & Technologies Used:
- Python 3.10
- Libraries: requests, BeautifulSoup, lxml, Flask, pytest
- GitHub Actions for CI/CD automation
- Safe Targets: OWASP Juice Shop, TestPHP Vulnweb

Methodology:
1. **Crawling:** Discover HTML forms and input fields.
2. **Payload Injection:** Insert test payloads for XSS and SQLi.
3. **Detection:** Identify reflected payloads (XSS) and SQL error messages (SQLi).
4. **Reporting:** Store results in JSON format for later review.
5. **Automation:** Execute scans through GitHub Actions with downloadable reports.

Results:
The scanner successfully identified injection points on safe test targets such as TestPHP Vulnweb. Findings are recorded in structured JSON files, which include vulnerability type, affected parameter, payload, and evidence.

Conclusion:
The project provided hands-on exposure to core web security concepts, Python programming, and DevOps workflows. While the scanner is simplified and not a replacement for professional tools, it demonstrates essential vulnerability detection logic and emphasizes **ethical hacking practices**.