**Internship Project Report**

**Title: Web Application Vulnerability Scanner**

# Introduction

In today’s digital world, web applications are highly vulnerable to cyberattacks such as SQL Injection (SQLi), Cross-Site Scripting (XSS), and Cross-Site Request Forgery (CSRF). The objective of this project was to build a Python-based Web Vulnerability Scanner with a modern web interface to detect such vulnerabilities. This scanner allows security researchers and developers to analyze websites for common flaws, view results in a clean dashboard, and improve the overall security posture of applications.

# Abstract

The project focuses on creating a lightweight yet powerful vulnerability scanner using Python. It crawls input fields of a target website, injects test payloads, and analyzes responses to identify potential security issues. The scanner provides clear results with vulnerability evidence, making it easier for developers to take corrective action. A Flask-based web interface was developed to make scanning interactive, user-friendly, and professional.

# Tools Used

* Python – Core programming language
* Flask – Web framework for UI
* Requests & Beautiful Soup – For crawling and input analysis
* Regex (Pattern Matching) – For vulnerability detection- Bootstrap 5 – For modern and responsive frontend design

# Steps Involved in Building the Project

1. Setup Environment – Installed Python, Flask, and required dependencies.
2. Crawler Development – Implemented input field detection using Requests & Beautiful Soup.
3. Payload Injection – Created test payloads for XSS & SQLi.
4. Response Analysis – Used regex and string-matching techniques to identify vulnerabilities.
5. Web Interface – Designed a Flask-based UI with Bootstrap for user interaction.
6. Result Logging – Displayed findings with severity messages inside the web interface.

# Conclusion

The Web Vulnerability Scanner successfully detects basic vulnerabilities such as XSS and SQLi in web applications. It provides a modern and user-friendly dashboard, enabling even beginners to test web security easily. In the future, the project can be extended with advanced modules like CSRF detection, scan history storage, and export-to-PDF/CSV reporting.