AI-Powered XSS Detector – Final Report

Name: Phani Vemula , R.Nagaraju

Roll No : 160123737065 , 160123737303

Date: 30/08/2025

GitHub Repository: [link](https://github.com/Rodopollunagaraju/XSS_Detector_Ai)

# Project Overview

This project was developed as part of the Cyber Security . The objective was to build an AI-powered XSS (Cross-Site Scripting) detector that identifies whether user input is safe or potentially malicious.  
  
The system integrates machine learning with rule-based checks to detect common XSS payloads (e.g., <script>, onerror=, javascript:). It simulates real-world scenarios where user input needs to be validated in comments, forms, and search fields to prevent injection attacks.

# Technologies & Tools Used

**- Programming Language:** Python  
**- Frontend Framework:** Streamlit  
- **Machine Learning:** Scikit-learn (Logistic Regression)  
- **Text Processing:** TfidfVectorizer (for feature extraction)  
- **Dataset:** Labeled dataset of Safe (0) and Malicious (1) inputs from XSS payload collections  
- **Version Control:** Git & GitHub

# System Architecture

1. User enters text in the Streamlit web interface.  
2. The text is vectorized using TF-IDF to extract key features.  
3. The trained Logistic Regression classifier predicts whether the input is malicious.  
4. The result is displayed as either:  
 - Safe  
 - Potential XSS attack  
5. A probability score (0–1) indicates the confidence of detection.

# Security Features

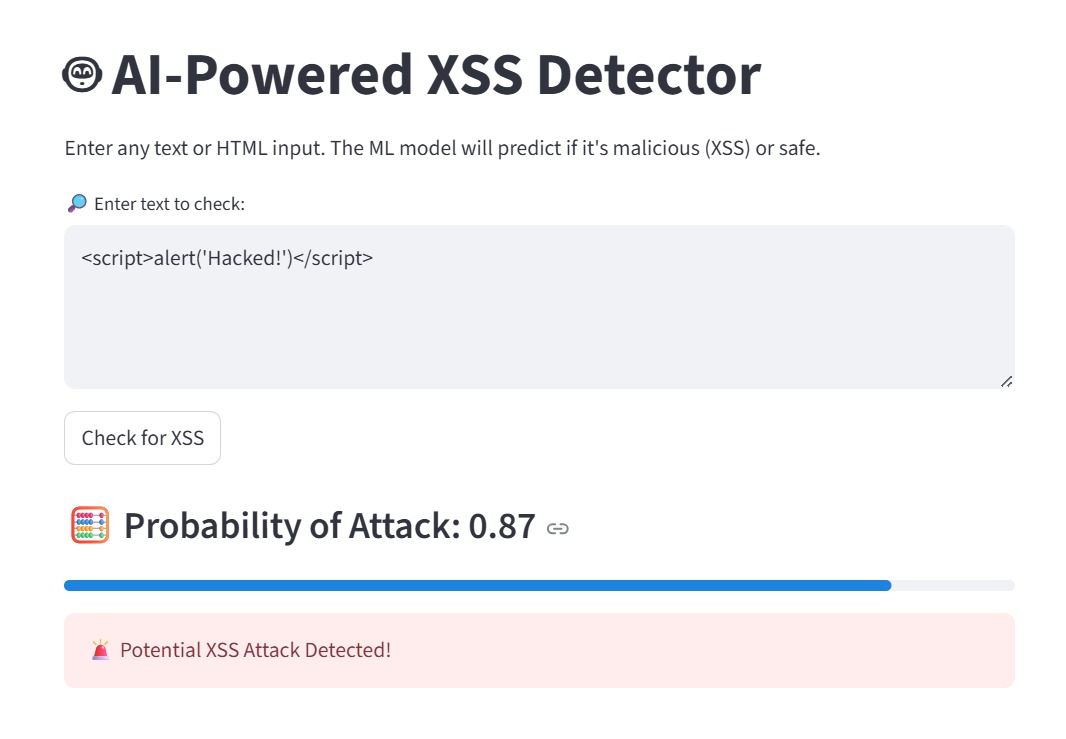
- Regex Filters: Initial detection of common payloads like <script>, onerror, javascript:.  
- ML Classifier: AI-based detection for more subtle and obfuscated payloads.  
- Real-time Detection: User input is scanned instantly before being processed or stored.  
- Probability Output: Users can see the model’s confidence level, improving transparency.

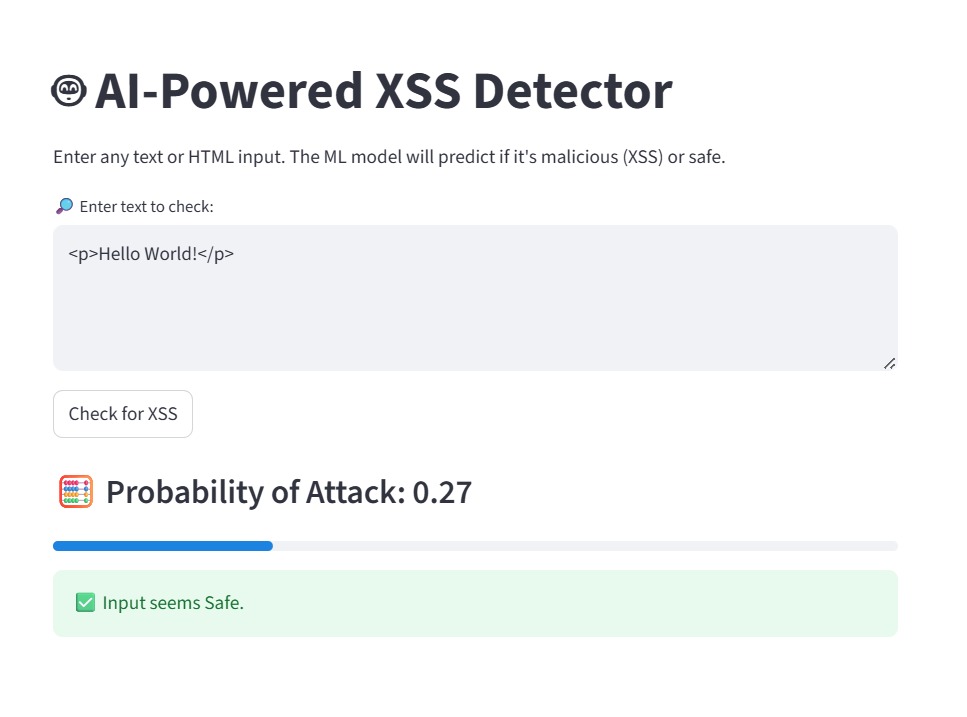
# Folder Structure

AI\_XSS\_Detector/  
│─ Main.py # Main Streamlit application  
│── XSS\_dataset.csv # Dataset used for training   
│── requirements.txt # Dependencies  
│── README.md # Project documentation

# Screenshots

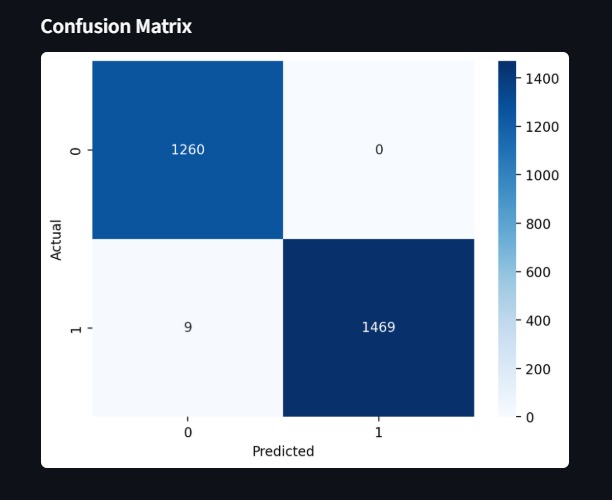
1. Input Interface – Streamlit textbox for entering comments or form input.

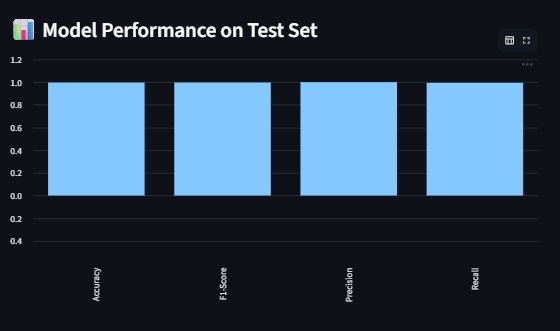
  
2. Prediction Result – Green “Safe” or Red “Potential XSS Attack” status message.

  
3. Probability Bar – Progress bar showing model confidence.

# Testing & Results

- Verified that malicious inputs such as <script>alert(1)</script> and onerror=alert(1) were correctly flagged as attacks.  
- Safe inputs such as “Hello, how are you?” were classified correctly as Safe.  
- The trained model achieved good accuracy on test data, showing reliable separation between benign and malicious samples.  
  
Result: The system successfully detects both obvious and subtle XSS payloads with high accuracy.





# Deliverables

- GitHub Repository with full project code & documentation  
- Walkthrough video (demo of Safe vs. Malicious inputs)  
- Security Report (this document)

# Learning Outcomes

- Gained practical understanding of XSS attack vectors  
- Hands-on experience with ML model training for cybersecurity tasks  
- Skills in text preprocessing and TF-IDF feature extraction  
- Built a real-time security scanner using Streamlit  
- Strengthened experience in Git & GitHub for project collaboration

# Conclusion

This project demonstrates a practical AI-powered solution to detect Cross-Site Scripting (XSS) attacks in real time. By integrating regex rules, machine learning, and a user-friendly interface, it provides a solid foundation for input validation and web application security.  
  
The system can be expanded in the future by:  
- Using deep learning models (e.g., LSTM, BERT) for advanced detection.  
- Integrating into web application firewalls (WAFs).  
- Supporting detection of other injection-based attacks like SQL Injection.