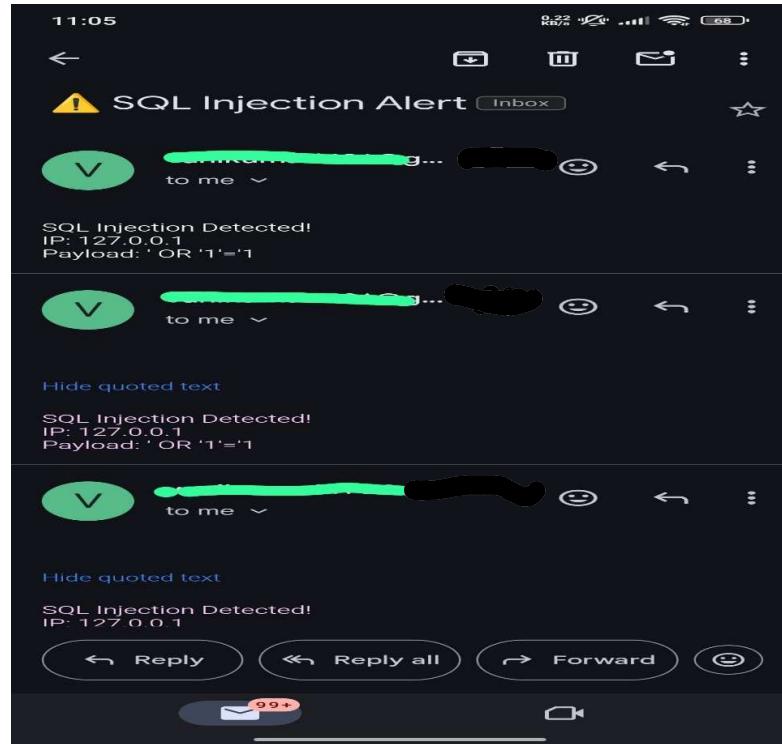
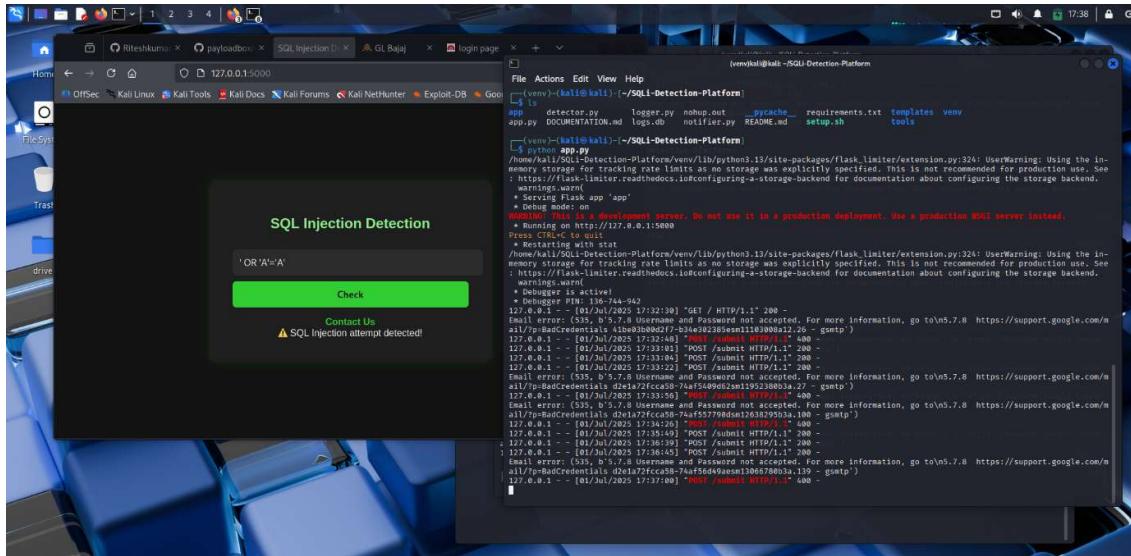


Sample Output:

SQLi Detection Platform – Secure Web Input Monitoring Module

This output highlights the results of a controlled test of the SQL Injection Detection Platform, performed in a Kali Linux environment. The platform is designed to monitor user input in real time, detect potential SQL injection (SQLi) attempts instantly, and alert both the user and administrators via a secure Gmail notification system.



Detection Event Overview

During each test session, the platform performs the following actions:

-  **Pattern Detection:**

The system scans for well-known SQL injection patterns like:

' OR '1'='1, UNION SELECT, --, and DROP TABLE.

It uses regular expressions to catch suspicious input before it can reach the backend.

-  **On-Screen Alert:**

As soon as an injection attempt is detected, the user sees a clear warning message:

 SQL Injection attempt detected!

-  **Backend Protection:**

The backend (powered by Flask) automatically blocks the malicious input by returning an HTTP 400 Bad Request, ensuring the attack does not proceed.

Real-Time Gmail Alerts

To keep the admin informed, the platform sends an instant alert email through Gmail. Each alert contains:

- The injection payload detected
- The timestamp of the attempt
- The source IP address (e.g., 127.0.0.1 for local tests)
- Basic HTTP request information

All emails are securely sent over TLS using Gmail's SMTP server, and are automatically scanned for security. This ensures alerts are reliable and confidential.

Terminal Logging & Evidence

Each detection event is logged in real time in the Flask server console. For example:

[01/Jul/2025 17:32:41] "POST /submit HTTP/1.1" 400 -

These logs serve as direct evidence of the SQLi detection and prevention in action.

System Metadata Summary

Every session also captures system-level details to assist with documentation and analysis:

- Username (if applicable)
- Timestamp of the event

- Public and private IPs
 - OS: Kali Linux (2025.2)
 - Tools: Python 3.13, Flask
 - Test URL: <http://127.0.0.1:5000>
-