Project 1 : SentinelShield – Advanced Intrusion Detection & Web Protection System

- **Project Overview :**

SentinelShield is a lightweight Intrusion Detection and Web Protection System developed as a practical cybersecurity project. The system simulates the behavior of a basic Web Application Firewall (WAF) by inspecting incoming HTTP requests, detecting malicious patterns, logging security events, and generating alerts.

This project helps in understanding how real-world security systems monitor, analyze, and respond to web-based attacks.

- **Features :**

- Detects SQL Injection attempts
- Detects Cross-Site Scripting (XSS) attacks
- Logs all malicious activities
- Generates real-time alerts
- Simple and lightweight implementation

- **Technologies Used :**

- Operating System: Windows / Kali Linux
- Programming Language: Python
- Framework: Flask
- Tools: VS Code, Terminal/PowerShell, Web Browser

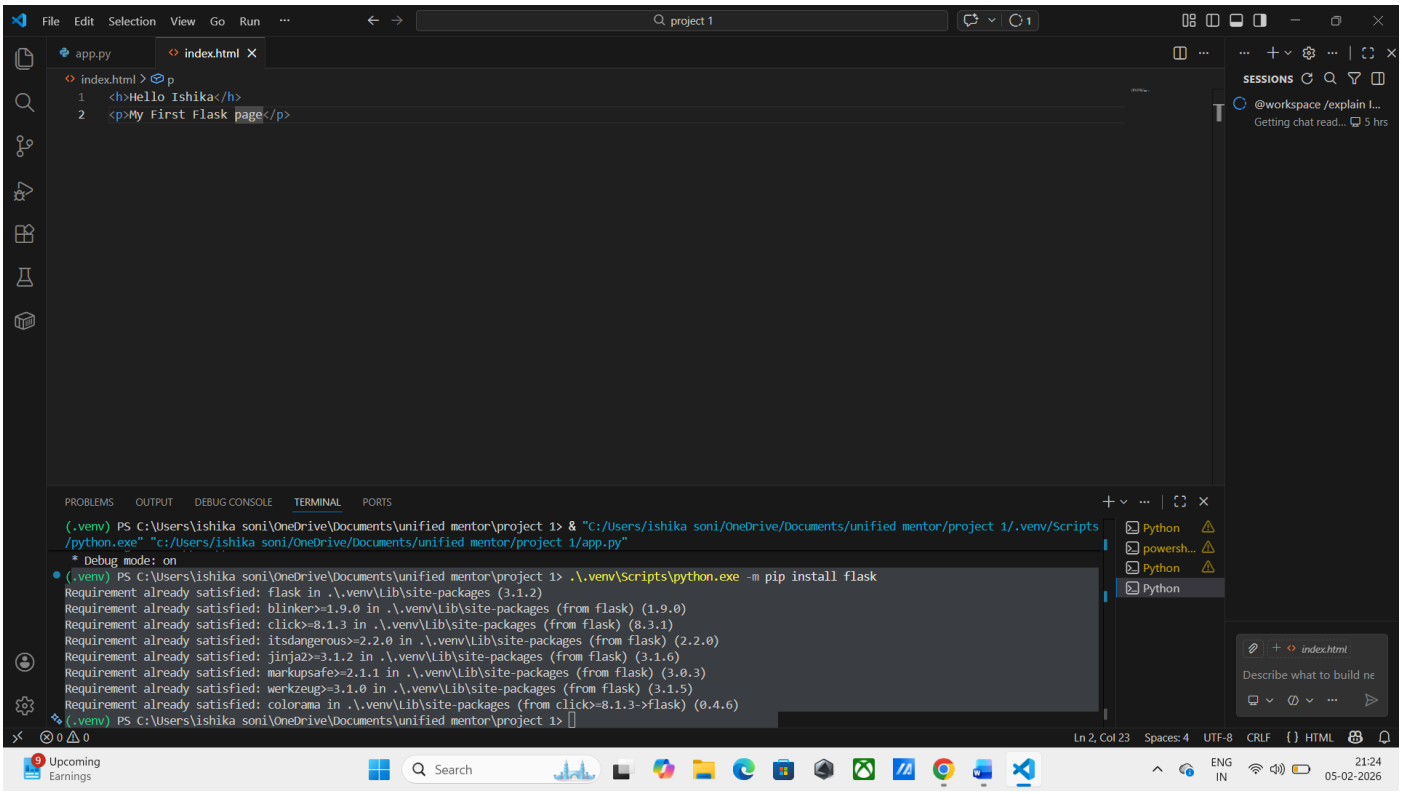
- **Project Structure**

```
project 1/
├── app.py
├── logs/
│   └── alerts.log
└── README.md
```

- **How to Run the Project :**
  - **Step 1: Install Dependencies**

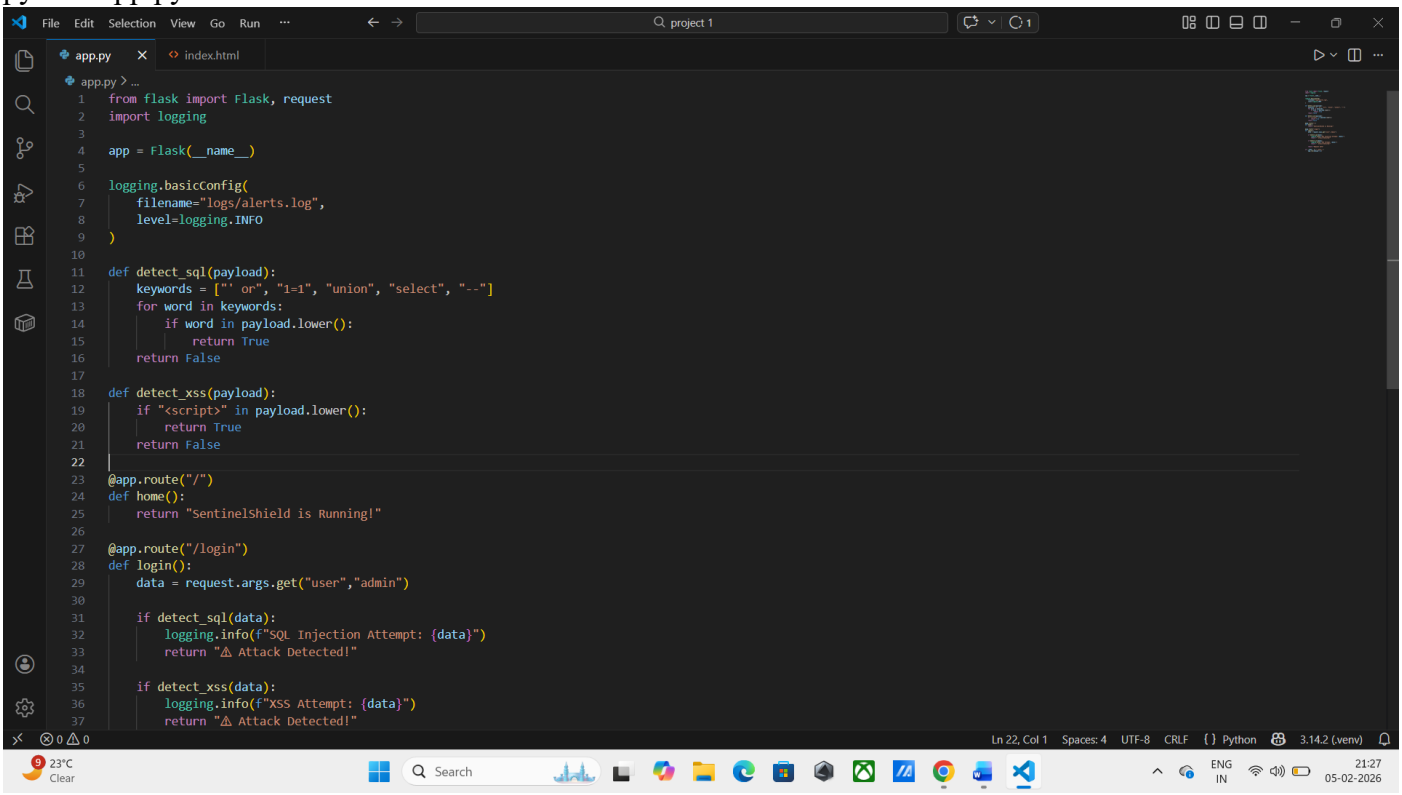
pip install flask or

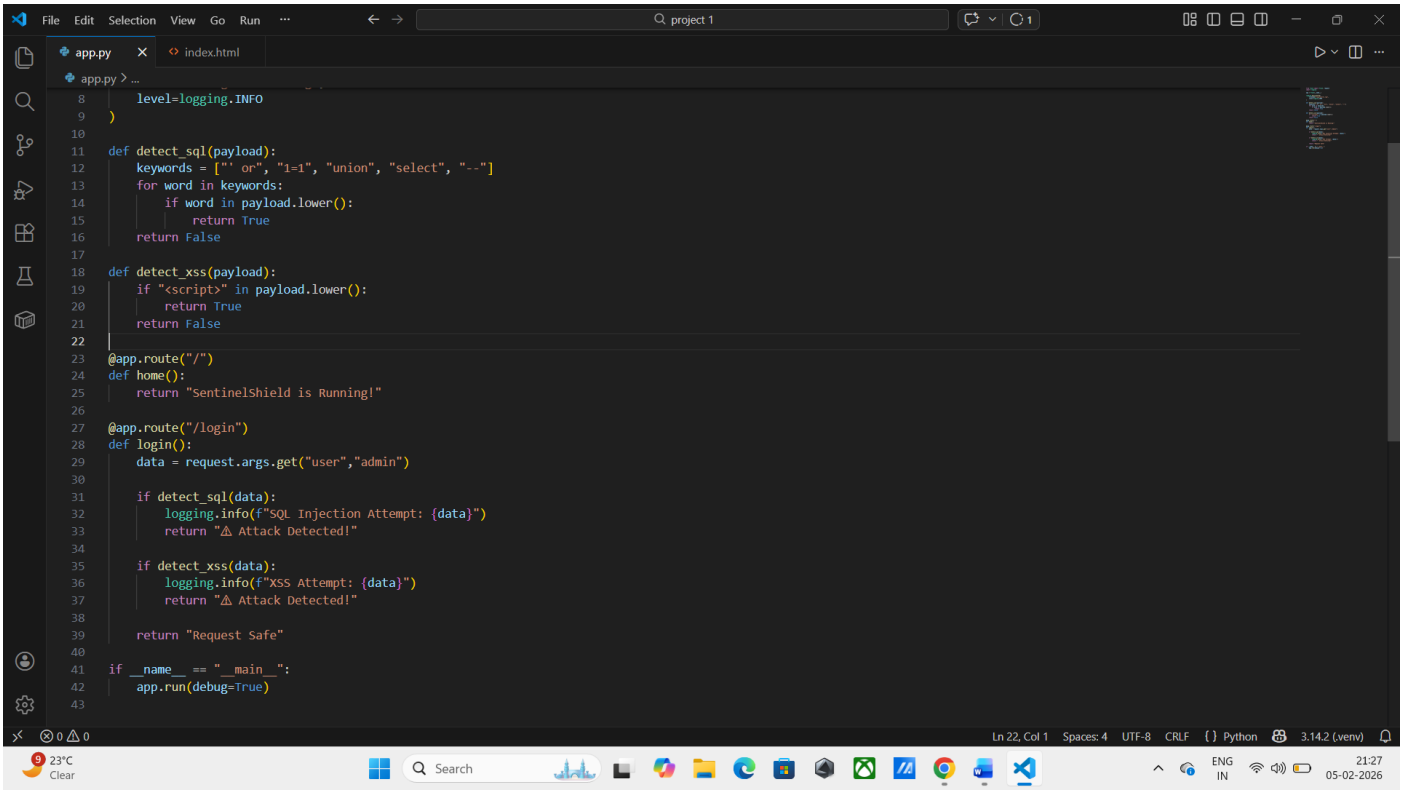
.\.venv\Scripts\python.exe -m pip install flask



  - **Step 2: Run the Application**

python app.py

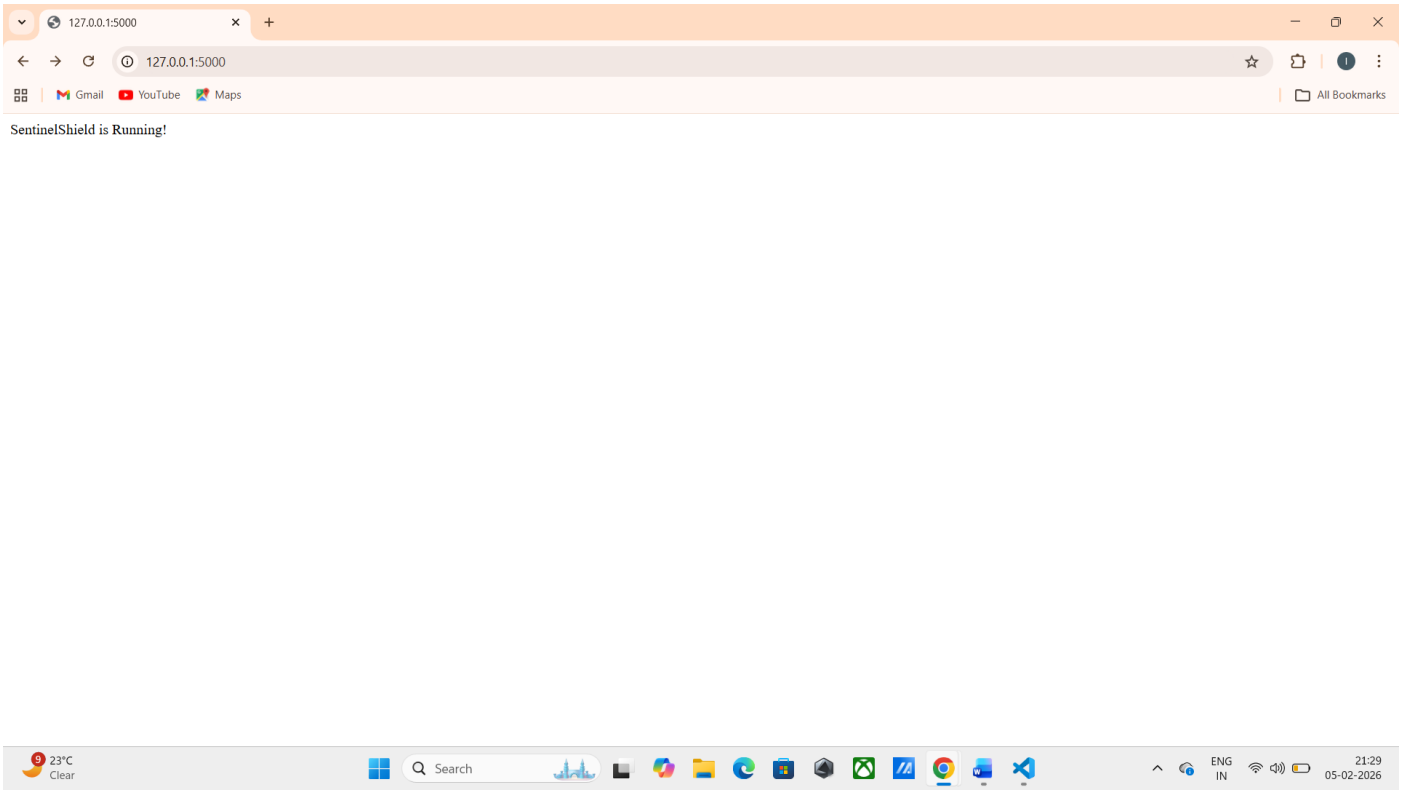




```
    level=logging.INFO
)

def detect_sql(payload):
    keywords = ["' or", "1=1", "union", "select", "--"]
    for word in keywords:
        if word in payload.lower():
            return True
    return False

def detect_xss(payload):
    if "<script>" in payload.lower():
        return True
    return False

@app.route("/")
def home():
    return "SentinelShield is Running!"

@app.route("/login")
def login():
    data = request.args.get("user","admin")

    if detect_sql(data):
        logging.info(f"SQL Injection Attempt: {data}")
        return "⚠ Attack Detected!"

    if detect_xss(data):
        logging.info(f"XSS Attempt: {data}")
        return "⚠ Attack Detected!"

    return "Request Safe"

if __name__ == "__main__":
    app.run(debug=True)
```

- **Step 3: Open in Browser**

http://127.0.0.1:5000

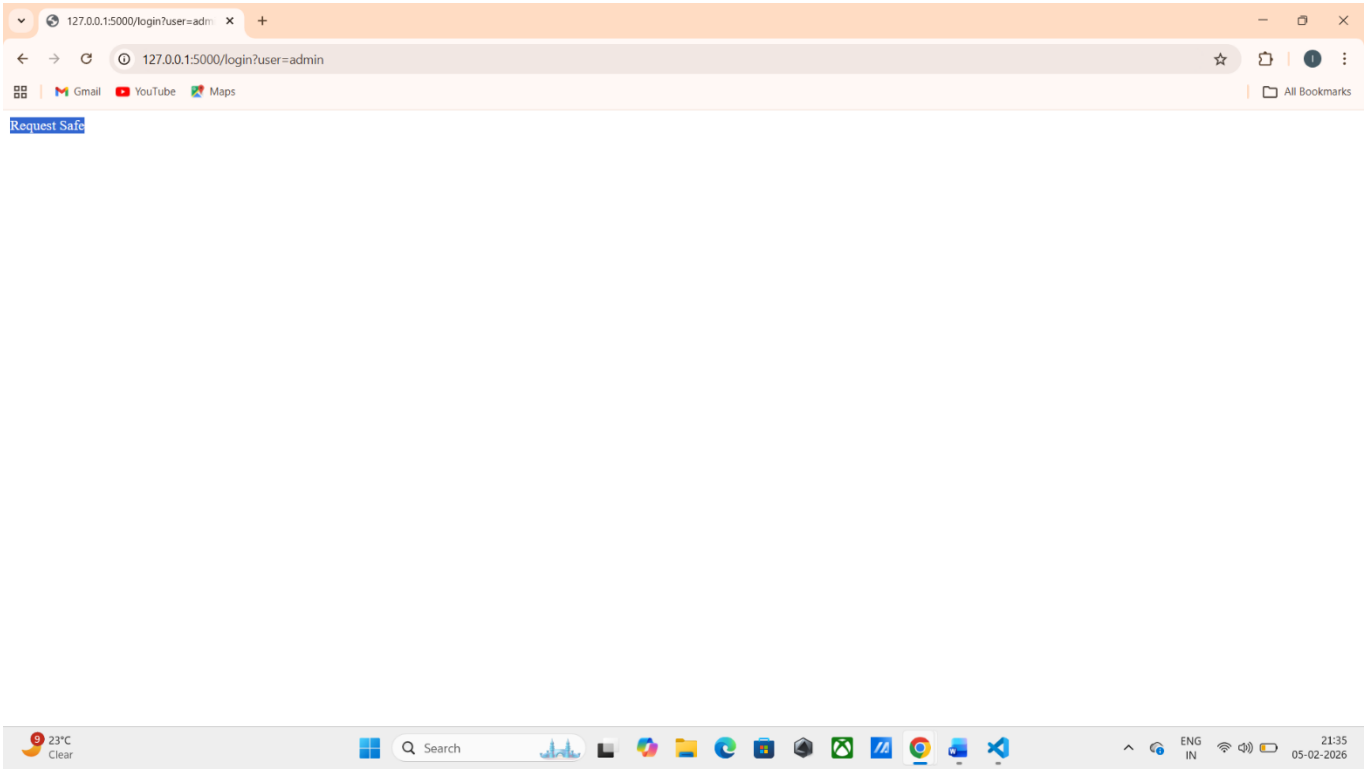


- **Testing the System :**
  - **Normal Request**

http://127.0.0.1:5000/login?user=admin

  - **Output:**

Request Safe

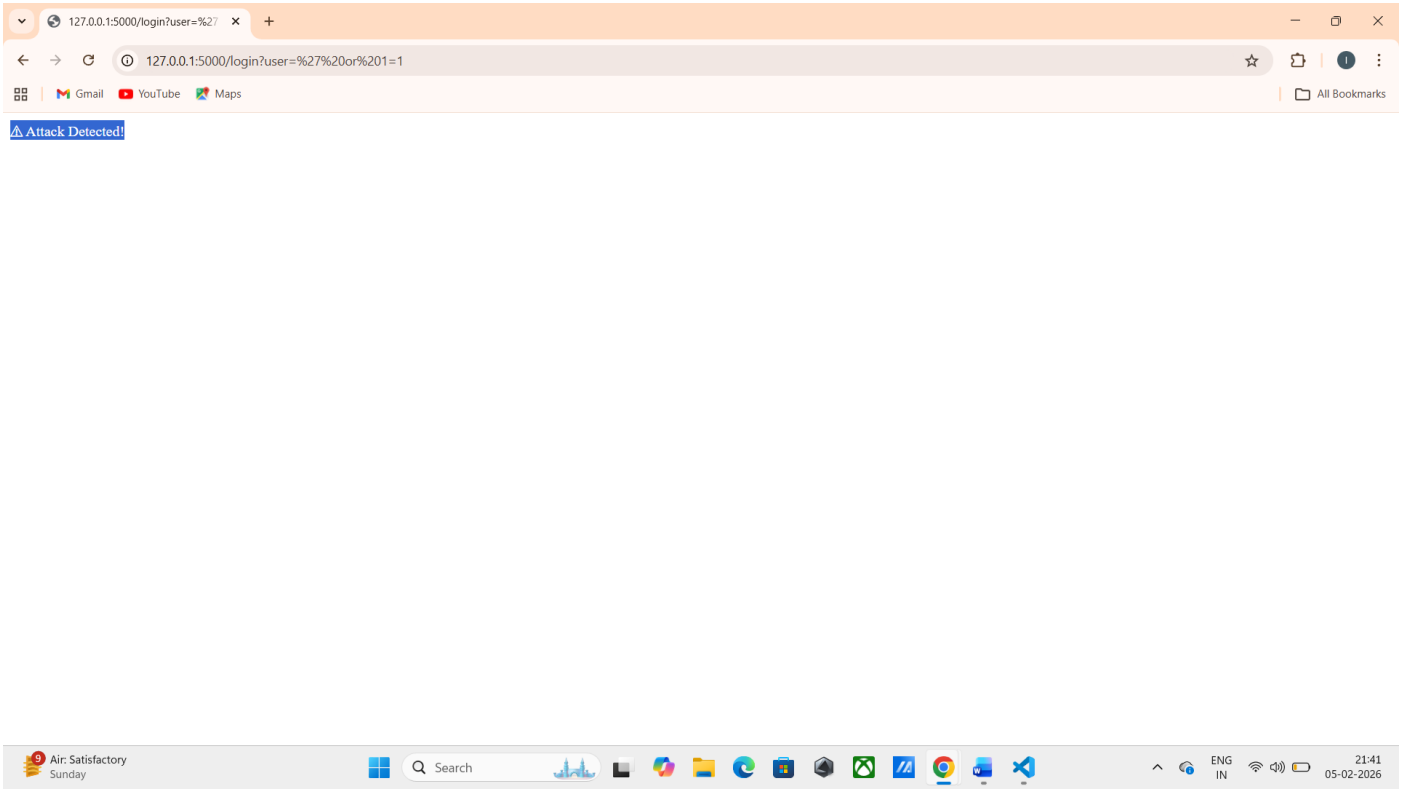


  - **SQL Injection Test**

http://127.0.0.1:5000/login?user=' or 1=1

- **Output:**

Attack Detected!

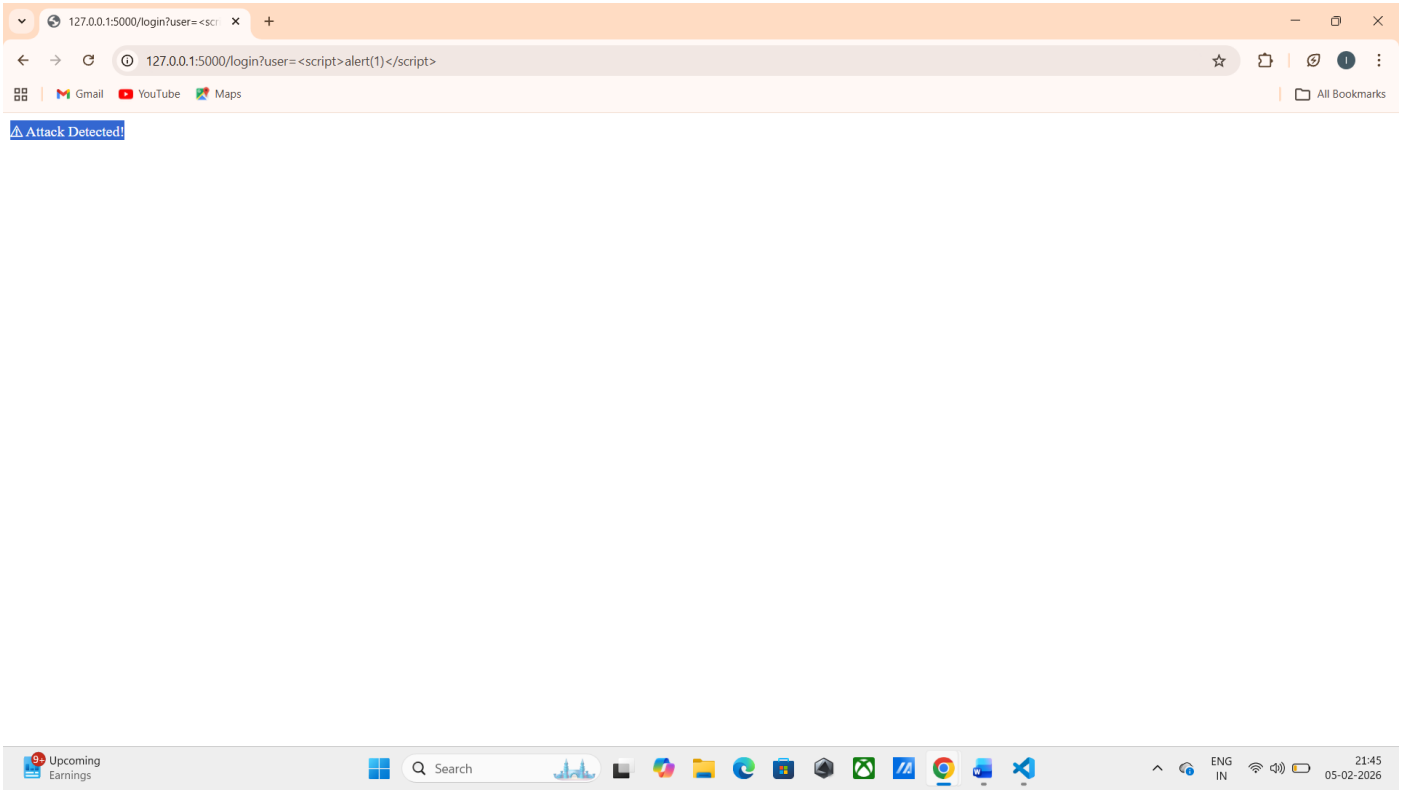


- **XSS Test**

http://127.0.0.1:5000/login?user=<script>alert(1)</script>

- Output:

Attack Detected!



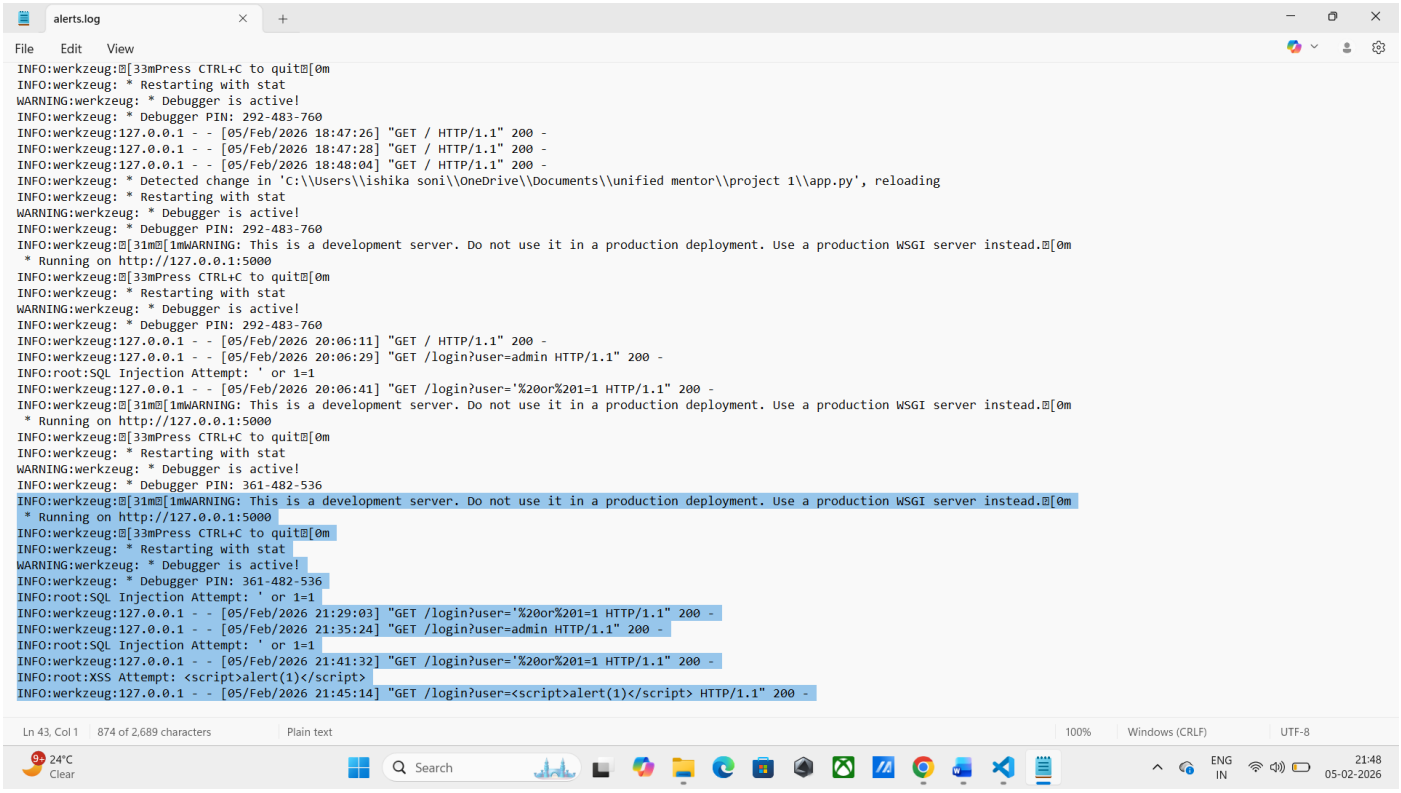
- **Log Output**

  All detected attacks are stored in:

  - logs/alerts.log
- **Example:**

  SQL Injection Attempt: ' or 1=1

  XSS Attempt: <script>alert(1)</script>



- **Learning Outcomes :**

  - Understanding of web application security
  - Practical knowledge of intrusion detection
  - Hands-on experience with Flask framework
  - Basic implementation of logging and monitoring
  - Real-world simulation of IDS/WAF concepts

- **Future Enhancements :**

  - IP-based blocking system
  - Brute force attack detection
  - Web dashboard for monitoring
  - Email or SMS alert system
  - Integration with SIEM tools