Enhanced Web Application and Server Security Using Secure Packet Evaluation and Logging System Using OpenAI GPT-4 (NETWORK ANOMALY DETECTION)

INTERNSHIP PROJECT REPORT

by

Raghuttama Padakandla
Aviral Avesh
TEAM:- 9.3

SmartInternz
November, 2023

TABLE OF CONTENTS

CHAPTER NO. TITLE PAGE NO.

Introduction

Robust web application and server security solutions are important in light of the growing frequency and sophistication of cyber-attacks. 'Enhanced Web Application and Server Security Using Secure Packet Evaluation and Logging System Using OpenAI GPT-4,' our project, tries to address this urgent concern by introducing a state-of-the-art method to strengthen online application security.

Our research analyses incoming data using OpenAI GPT-4 and a Flask-based server, with an emphasis on finding fraudulent data packets. This introduction aims to set the project's context while highlighting its importance in the current cybersecurity environment.

We go into great depth about our project in this report, including application features, high availability concerns, components and technologies utilized, and user stories. Readers will have a comprehensive knowledge of our novel method and how it may enhance online application security by the end of this article.

Abstract

This project implements a Flask-based server with integrated middleware, introducing an innovative method to strengthen online application security. To evaluate the possible maliciousness of incoming packets, the system uses a two-step validation procedure that makes use of OpenAI's GPT-4 capabilities.

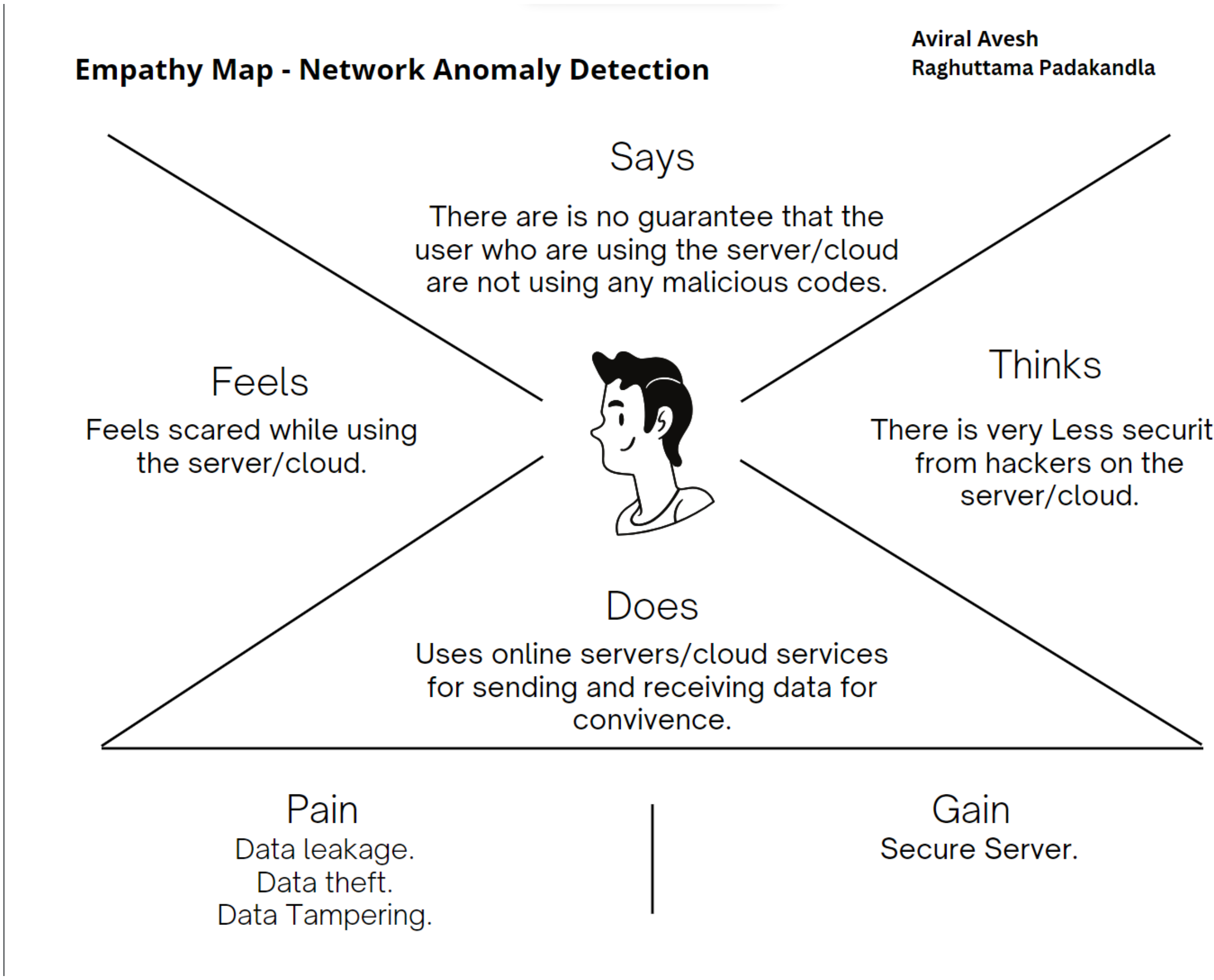
In the first stage, incoming packets are received by the Flask server and sent to OpenAI GPT-4 for analysis using middleware. In order to produce a score on a scale of 1 to 10, the system consults GPT-4 to determine the possibility of malicious intent. Packets are marked as possibly malicious if they score more than a set threshold (for example, 6).

The system then logs pertinent data, such as the packet content, IP address, and GPT-4 assessment score, into a safe database. This thorough logging system makes post-analysis easier and enables the detection of trends or new dangers.

Only the IP address is recorded for non-malicious packets in order to protect privacy while still revealing potential attack vectors. This two-tiered logging strategy provides a compromise between security and user privacy while enabling a sophisticated analysis of online traffic.

The suggested solution improves the overall security posture of online applications by utilizing OpenAI GPT-4 to provide proactive threat detection and response tactics. The system's modular structure enables scalability and flexibility in response to changing cybersecurity
problems.

Empathy Map



Empathy Map - Network Anomaly Detection

Aviral Avesh
Raghuttama Padakandla

Says

There are is no guarantee that the user who are using the server/cloud are not using any malicious codes.

Feels

Feels scared while using the server/cloud.

Thinks

There is very Less security from hackers on the server/cloud.

Does

Uses online servers/cloud services for sending and receiving data for convivence.

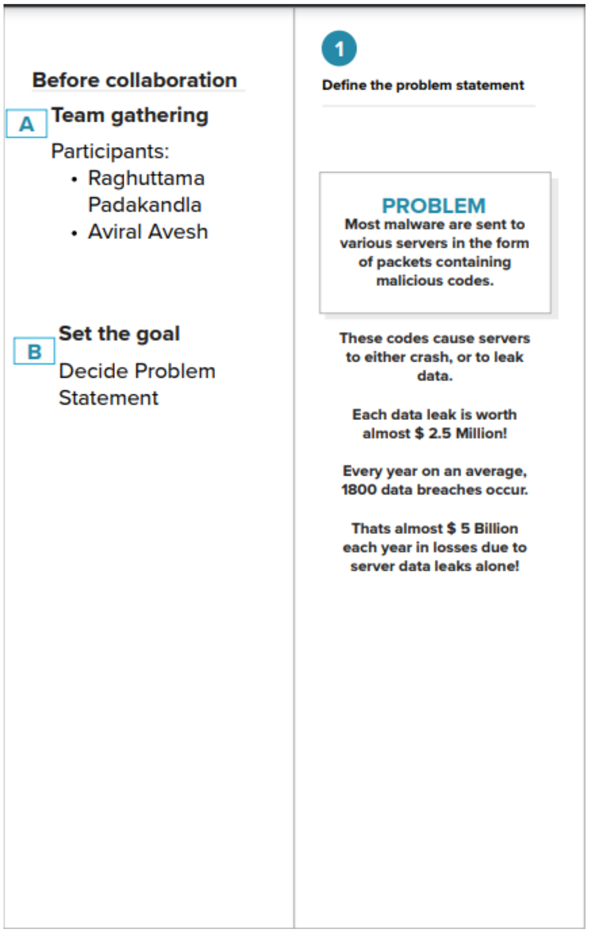
Pain

Data leakage.
Data theft.
Data Tampering.

Gain

Secure Server.

Brainstorming Report



Before collaboration

A Team gathering

Participants:

- Raghuttama Padakandla
- Aviral Avesh

B Set the goal

Decide Problem Statement

1 Define the problem statement

PROBLEM

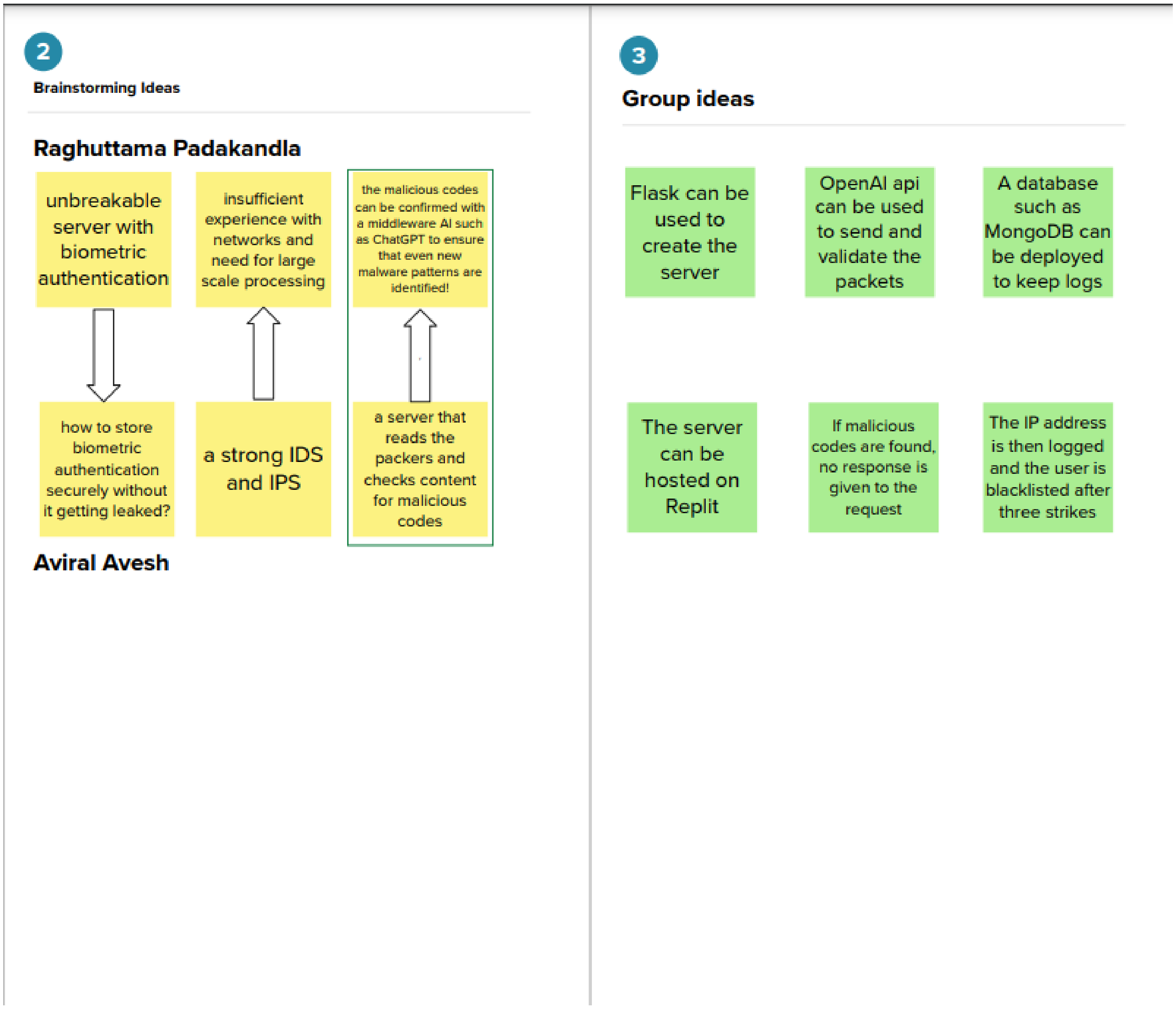
Most malware are sent to various servers in the form of packets containing malicious codes.

These codes cause servers to either crash, or to leak data.

Each data leak is worth almost $ 2.5 Million!

Every year on an average, 1800 data breaches occur.

Thats almost $ 5 Billion each year in losses due to server data leaks alone!



2

Brainstorming Ideas

Raghuttama Padakandla

unbreakable server with biometric authentication

how to store biometric authentication securely without it getting leaked?

insufficient experience with networks and need for large scale processing

a strong IDS and IPS

the malicious codes can be confirmed with a middleware AI such as ChatGPT to ensure that even new malware patterns are identified!

a server that reads the packers and checks content for malicious codes

Aviral Avesh

3

Group ideas

Flask can be used to create the server

OpenAI api can be used to send and validate the packets

A database such as MongoDB can be deployed to keep logs

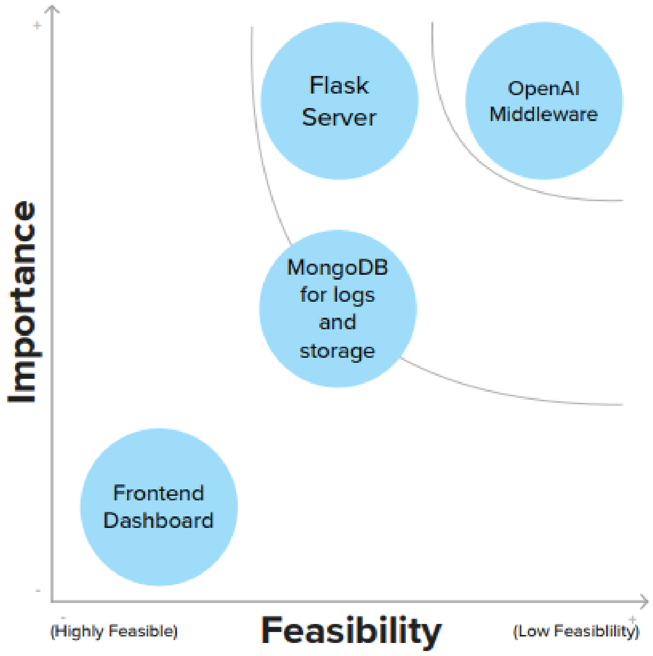
The server can be hosted on Replit

If malicious codes are found, no response is given to the request

The IP address is then logged and the user is blacklisted after three strikes

4

Prioritize



Technology Stack

Technical Architecture:

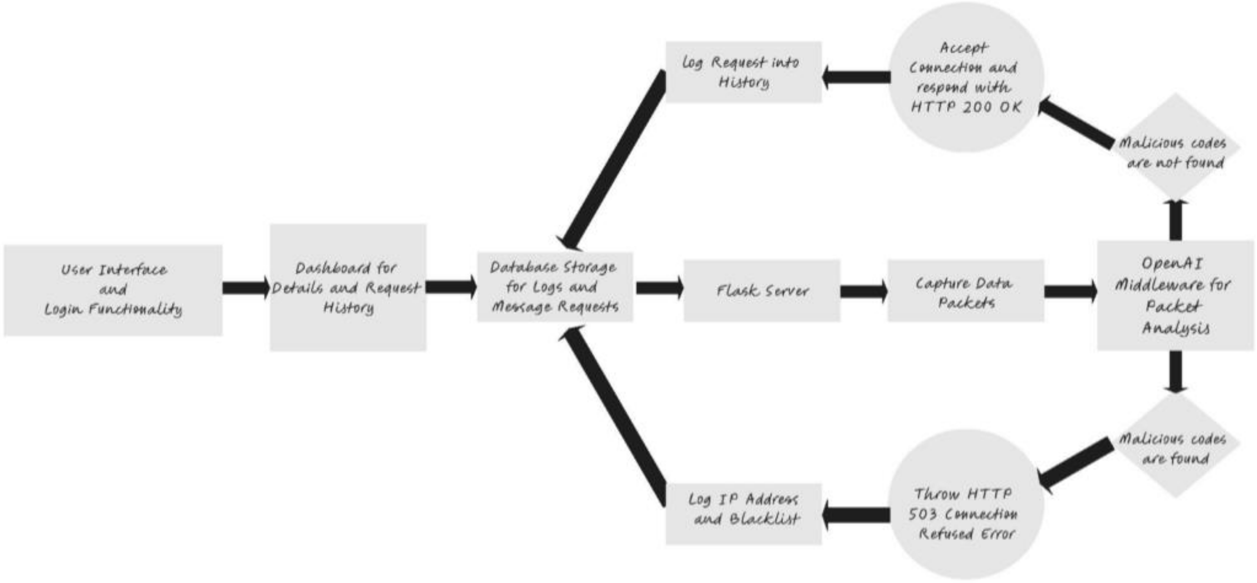

User Interface and Login Functionality
Dashboard for Details and Request History
Database Storage for Logs and Message Requests
Flask Server
Capture Data Packets
OpenAI Middleware for Packet Analysis
Malicious codes are not found
Accept Connection and respond with HTTP 200 OK
Log Request into History
Malicious codes are found
Throw HTTP 503 Connection Refused Error
Log IP Address and Blacklist

Table-1 : Components & Technologies:

Sr. No	Component	Description	Technology
1.	User Interface	Accepts user input for packet submission. Displays feedback and results from the backend.	HTML, CSS, JavaScript, JAVA
2.	Communication with Backend	Receives HHTP responses and displays the results to the user.	HHTP POST request.
3.	Backend	The Data Packet is extracted into a string (through JSON format)	Flask Server
4.	Backend	The server sends the extracted string to OpenAI to analyze for maliciousness.	API Call
5.	Cloud Database	Saves the user log in information.	MongoDB
6.	File Storage	All files and logs are stored on the cloud.	Replit Cloud Storage and MongoDB Database
7.	Open AI	Purpose of OpenAI for processing the packets of malicious code.	GPT-4
8.	Infrastructure (Server / Cloud)	Cloud Server used for hosting the entire application.	REPLIT

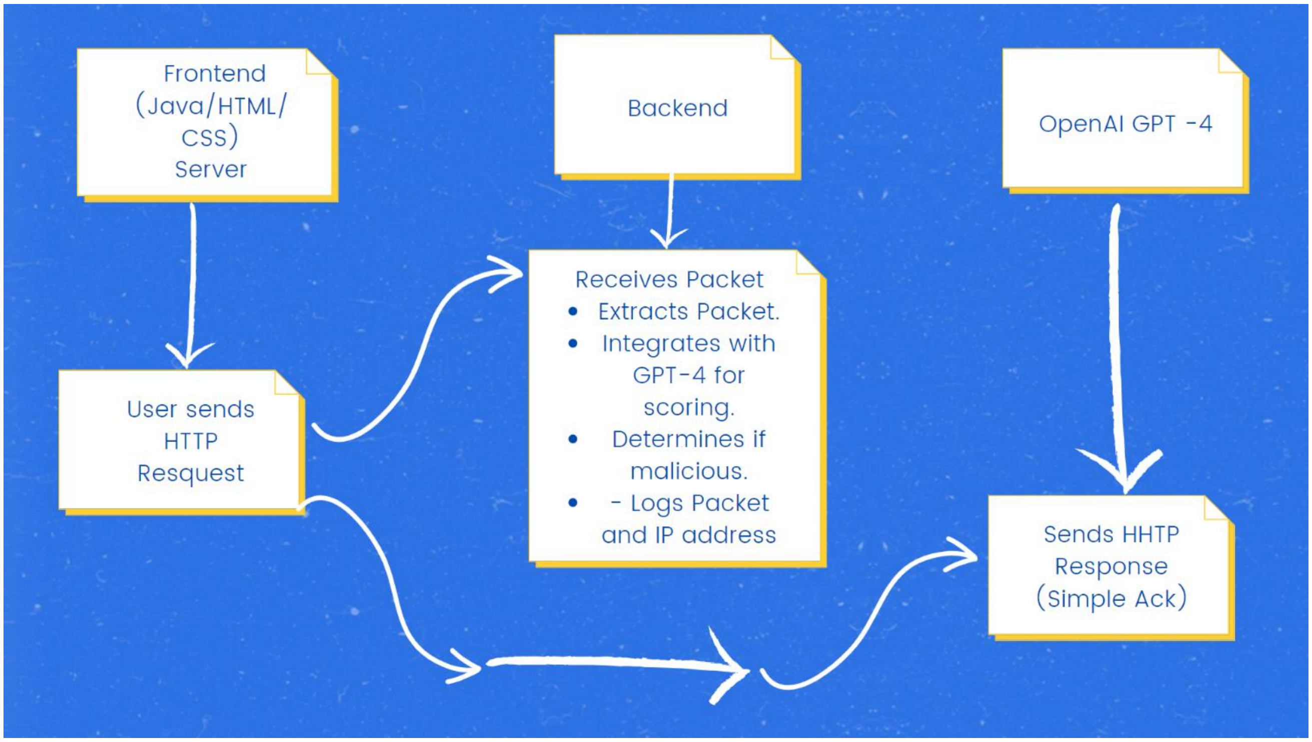
Table-2: Application Characteristics:

Sr. No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Chat GPT-4, Replit, MongoDB	
2.	Security Implementations	All server-side processing. If malicious data in packet is found, then HTTP request is denied.	HTTP and Server Side Processing
3.	Scalable Architecture	Multi–tier	HTML/CSS/JavaScript/JAVA/Python/OpenAI-GPT-4/MongoDB/Replit
4.	Availability	Utilizing cloud services, implementing disaster recovery practices, and distributing components across multiple servers or regions contribute to a highly available architecture for the security dashboard application. These considerations collectively enhance the system's resilience to fluctuations in traffic and potential failures, providing a reliable and available service to users.	Java, HTML/CSS/JavaScript, MongoDB, Replit, GPT-4/ Python
5.	Performance	Can handle up to 100 requests per sec, and continuous spam HTTP requests is auto blocked by Replit	Replit

Data Flow Diagram & User Stories

Data Flow Diagrams:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



Frontend (Java/HTML/ CSS) Server

Backend

OpenAI GPT -4

User sends HTTP Resquest

Receives Packet

- Extracts Packet.
- Integrates with GPT-4 for scoring.
- Determines if malicious.
- - Logs Packet and IP address

Sends HHTP Response (Simple Ack)

User Stories

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
User	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	Medium	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	Medium	Sprint-1
		USN-3	As a user, I can register for the application through Gmail	I can register & access the dashboard with my Gmail account	Medium	Sprint-1
	Login	USN-4	As a user, I can log into the application by entering email & password		High	Sprint-1
	Dashboard	USN-5	As a user, I cannot see the results from GPT-4 for security to ensure NO JAILBREAKS.	If non-malicious, HTTP 200 response. If malicious, HTTP 403 response.	Very High	Sprint-1

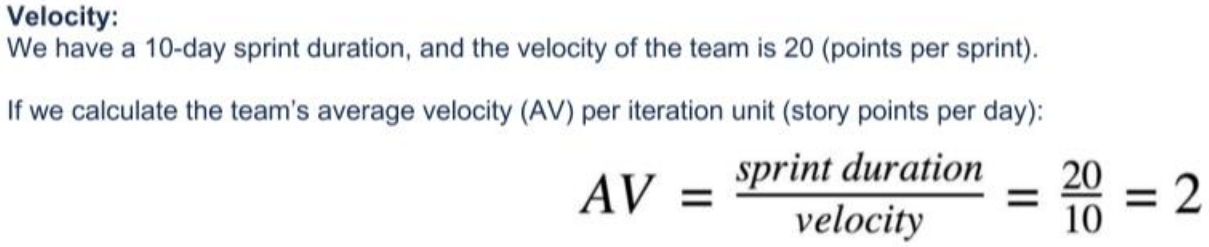
Project Planning Template

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	Medium	Aviral
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	Medium	Aviral
Sprint-1		USN-3	As a user, I can register for the application through Gmail	I can register & access the dashboard with my Gmail account	Medium	Aviral
Sprint-1	Login	USN-4	As a user, I can log into the application by entering email & password		High	
Sprint-1	Dashboard	USN-5	As a user, I cannot see the results from GPT-4 for security to ensure NO JAILBREAKS.	If non-malicious, HTTP 200 response. If malicious, HTTP 403 response.	Very High	Raghu
	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	Medium	Raghu

Project Tracker, Velocity & Burndown Chart: (4 Marks)

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	10 Days	1 Nov 2023	10 Nov 2023	20	29 Oct 2022



Velocity:

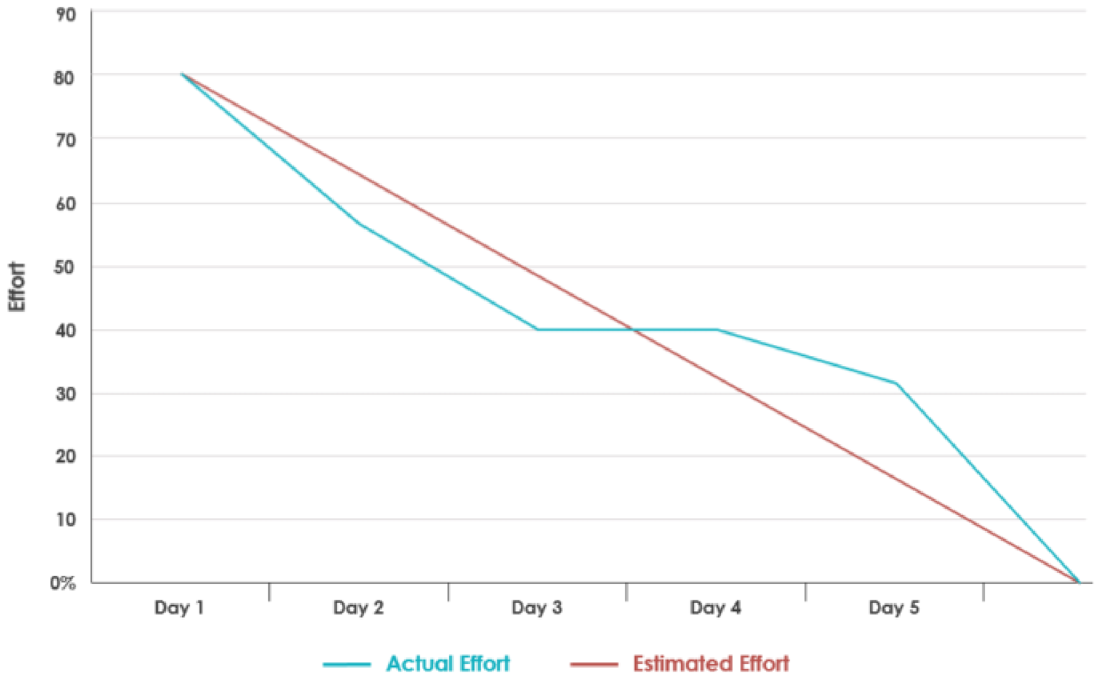
We have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint).

If we calculate the team's average velocity (AV) per iteration unit (story points per day):

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

Burndown Chart:

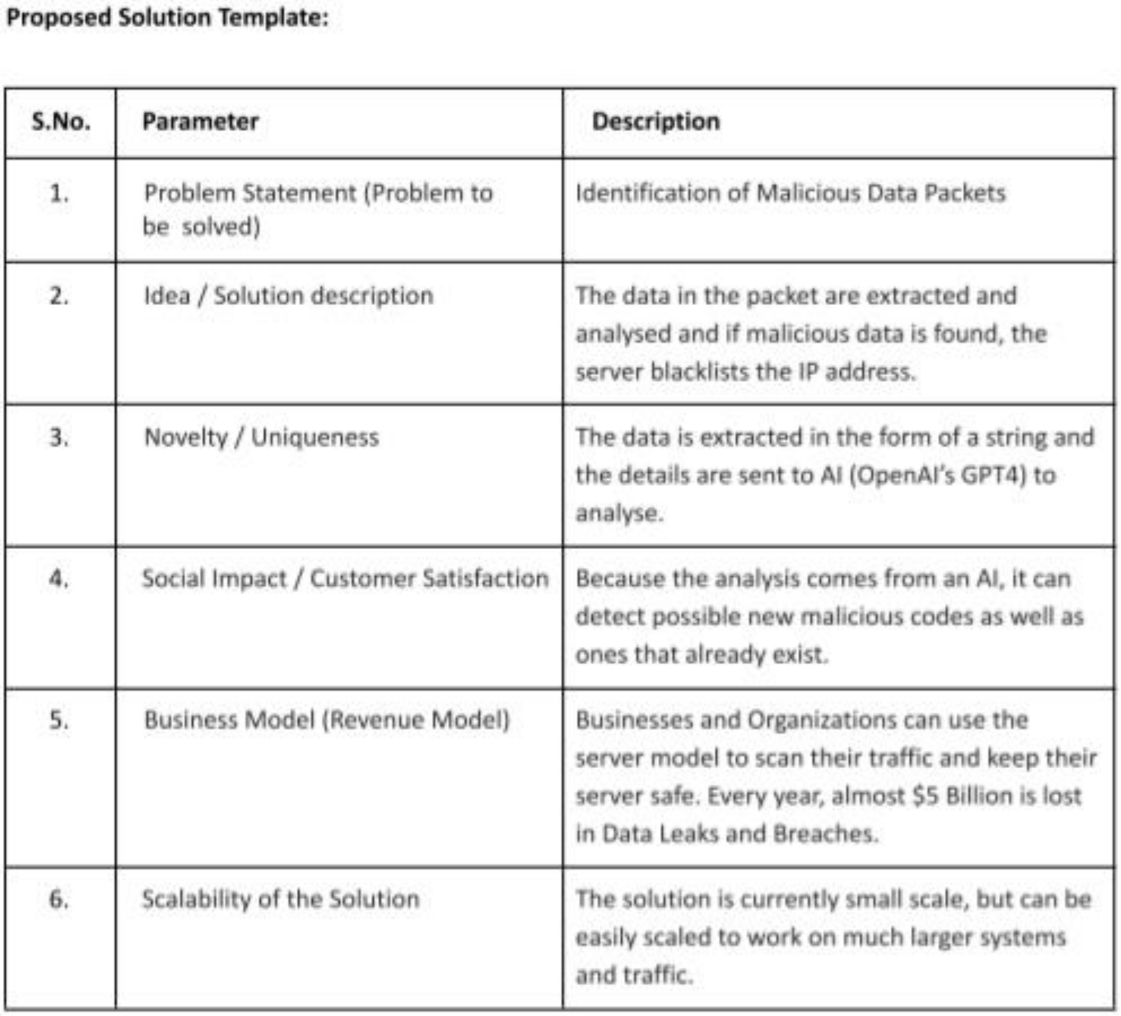
A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.



Based on development during Sprint 0.
Tasks tackled in sprint Zero:

- Main Flask Server Deployment
- Frontend Example for Dashboard
- OpenAI API calling
- HTTP Requests and Responses

Proposed Solution Template

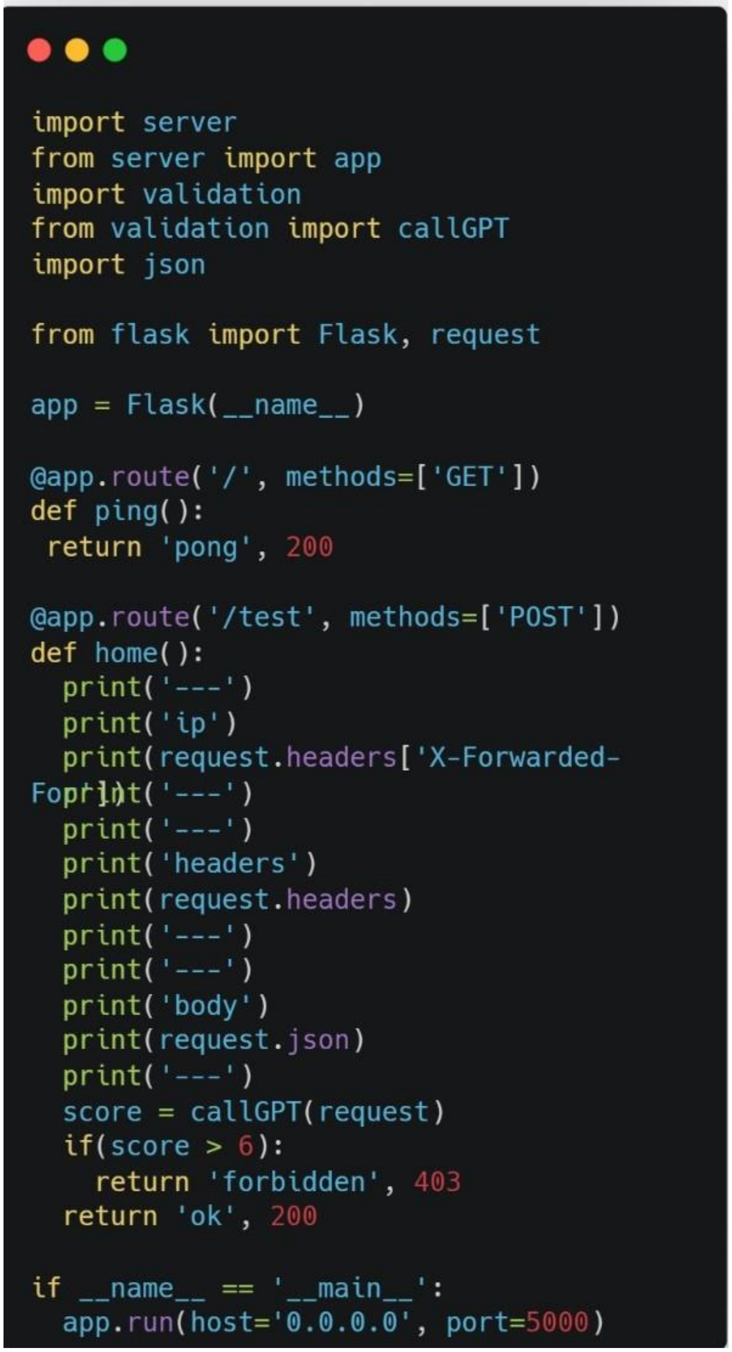


Proposed Solution Template:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Identification of Malicious Data Packets
2.	Idea / Solution description	The data in the packet are extracted and analysed and if malicious data is found, the server blacklists the IP address.
3.	Novelty / Uniqueness	The data is extracted in the form of a string and the details are sent to AI (OpenAI's GPT4) to analyse.
4.	Social Impact / Customer Satisfaction	Because the analysis comes from an AI, it can detect possible new malicious codes as well as ones that already exist.
5.	Business Model (Revenue Model)	Businesses and Organizations can use the server model to scan their traffic and keep their server safe. Every year, almost $5 Billion is lost in Data Leaks and Breaches.
6.	Scalability of the Solution	The solution is currently small scale, but can be easily scaled to work on much larger systems and traffic.

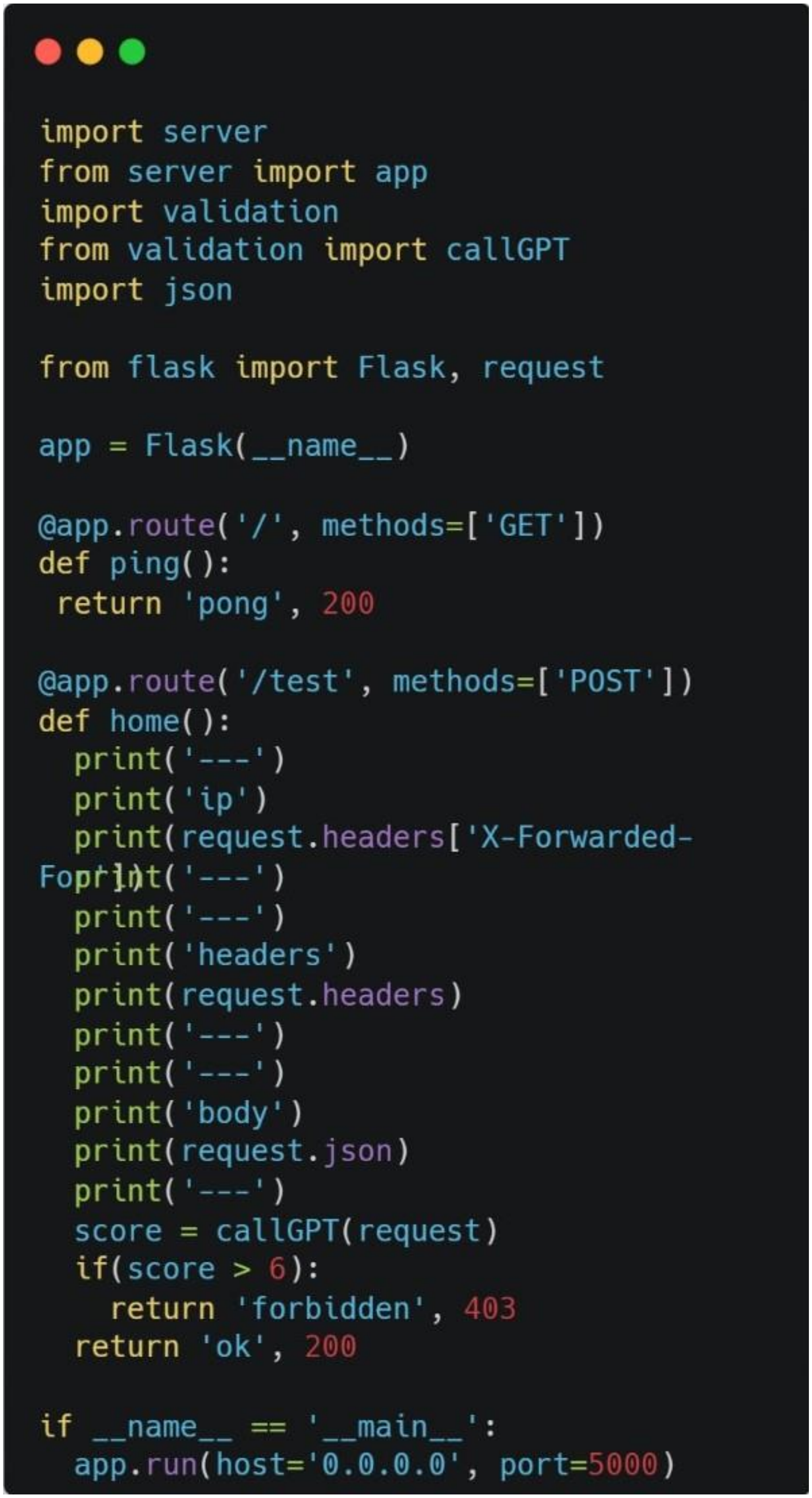
Code Screen Shorts

Asking for response from GPT-4



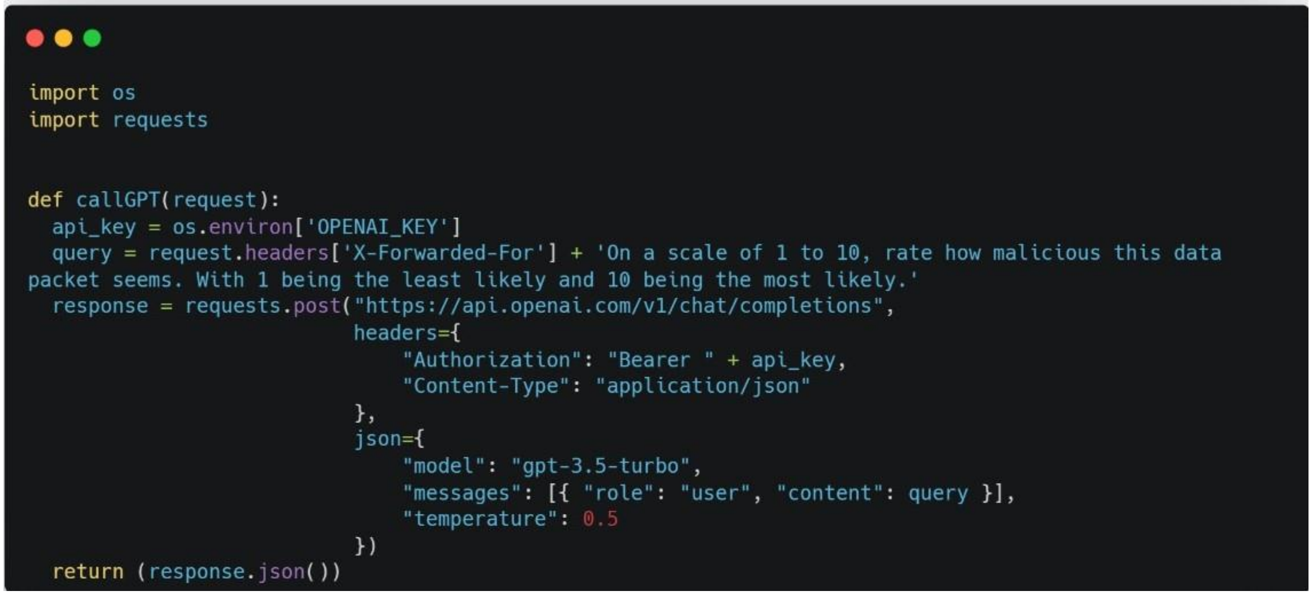
```
import server
from server import app
import validation
from validation import callGPT
import json

from flask import Flask, request

app = Flask(__name__)

@app.route('/', methods=['GET'])
def ping():
 return 'pong', 200

@app.route('/test', methods=['POST'])
def home():
  print('---')
  print('ip')
  print(request.headers['X-Forwarded-
For'])
  print('---')
  print('---')
  print('headers')
  print(request.headers)
  print('---')
  print('---')
  print('body')
  print(request.json)
  print('---')
  score = callGPT(request)
  if(score > 6):
    return 'forbidden', 403
  return 'ok', 200

if __name__ == '__main__':
  app.run(host='0.0.0.0', port=5000)
```

Flask Server



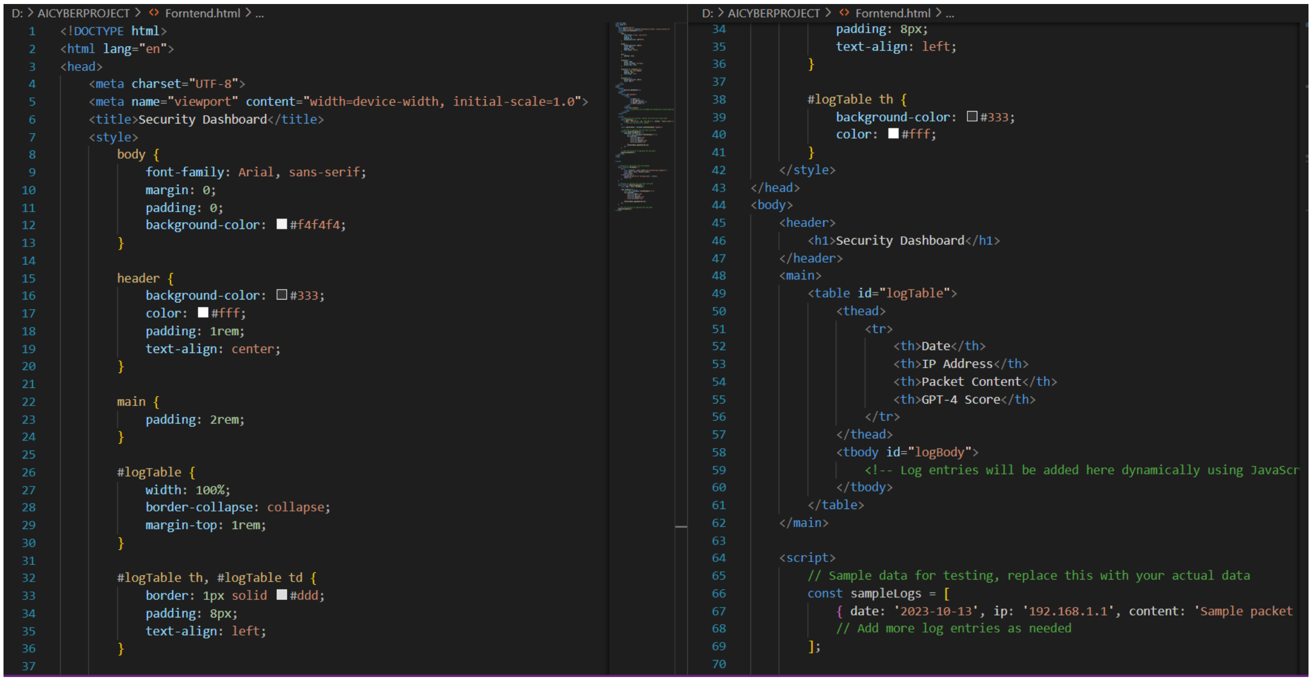
```
import server
from server import app
import validation
from validation import callGPT
import json

from flask import Flask, request

app = Flask(__name__)

@app.route('/', methods=['GET'])
def ping():
 return 'pong', 200

@app.route('/test', methods=['POST'])
def home():
  print('---')
  print('ip')
  print(request.headers['X-Forwarded-
For'])
  print('---')
  print('---')
  print('headers')
  print(request.headers)
  print('---')
  print('---')
  print('body')
  print(request.json)
  print('---')
  score = callGPT(request)
  if(score > 6):
    return 'forbidden', 403
  return 'ok', 200

if __name__ == '__main__':
  app.run(host='0.0.0.0', port=5000)
```

API Call



```
import os
import requests


def callGPT(request):
  api_key = os.environ['OPENAI_KEY']
  query = request.headers['X-Forwarded-For'] + 'On a scale of 1 to 10, rate how malicious this data
packet seems. With 1 being the least likely and 10 being the most likely.'
  response = requests.post("https://api.openai.com/v1/chat/completions",
                           headers={
                               "Authorization": "Bearer " + api_key,
                               "Content-Type": "application/json"
                           },
                           json={
                               "model": "gpt-3.5-turbo",
                               "messages": [{ "role": "user", "content": query }],
                               "temperature": 0.5
                           })
  return (response.json())
```

Dashboard

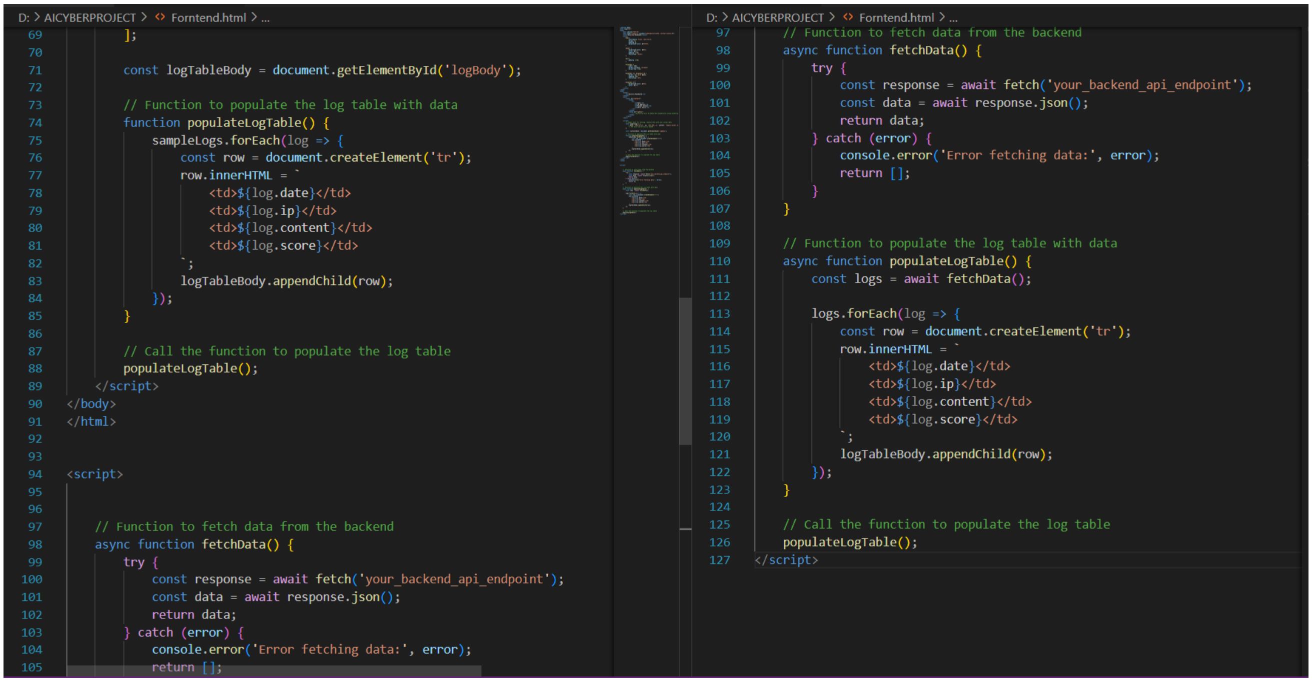


D: > AICYBERPROJECT > Forntend.html > ...

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Security Dashboard</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
            background-color: ■#f4f4f4;
        }

        header {
            background-color: □#333;
            color: ■#fff;
            padding: 1rem;
            text-align: center;
        }

        main {
            padding: 2rem;
        }

        #logTable {
            width: 100%;
            border-collapse: collapse;
            margin-top: 1rem;
        }

        #logTable th, #logTable td {
            border: 1px solid ■#ddd;
            padding: 8px;
            text-align: left;
        }

```

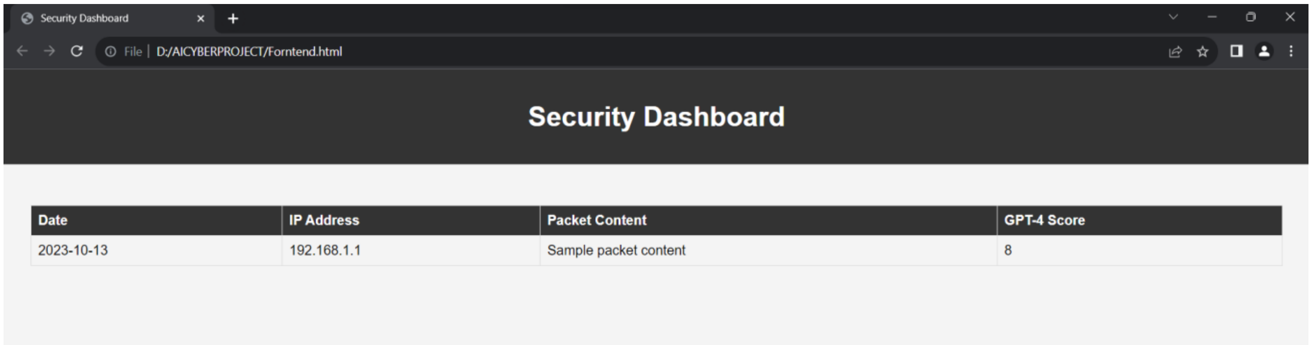
D: > AICYBERPROJECT > Forntend.html > ...

```
            padding: 8px;
            text-align: left;
        }

        #logTable th {
            background-color: □#333;
            color: ■#fff;
        }
    </style>
</head>
<body>
    <header>
        <h1>Security Dashboard</h1>
    </header>
    <main>
        <table id="logTable">
            <thead>
                <tr>
                    <th>Date</th>
                    <th>IP Address</th>
                    <th>Packet Content</th>
                    <th>GPT-4 Score</th>
                </tr>
            </thead>
            <tbody id="logBody">
                <!-- Log entries will be added here dynamically using JavaScr
            </tbody>
        </table>
    </main>

    <script>
        // Sample data for testing, replace this with your actual data
        const sampleLogs = [
            { date: '2023-10-13', ip: '192.168.1.1', content: 'Sample packet
            // Add more log entries as needed
        ];

```


D: > AICYBERPROJECT > Forntend.html > ...
D: > AICYBERPROJECT > Forntend.html > ...

```
        ];

        const logTableBody = document.getElementById('logBody');

        // Function to populate the log table with data
        function populateLogTable() {
            sampleLogs.forEach(log => {
                const row = document.createElement('tr');
                row.innerHTML = `
                    <td>${log.date}</td>
                    <td>${log.ip}</td>
                    <td>${log.content}</td>
                    <td>${log.score}</td>
                `;
                logTableBody.appendChild(row);
            });
        }

        // Call the function to populate the log table
        populateLogTable();
    </script>
</body>
</html>


<script>


    // Function to fetch data from the backend
    async function fetchData() {
        try {
            const response = await fetch('your_backend_api_endpoint');
            const data = await response.json();
            return data;
        } catch (error) {
            console.error('Error fetching data:', error);
            return [];
```

```
    // Function to fetch data from the backend
    async function fetchData() {
        try {
            const response = await fetch('your_backend_api_endpoint');
            const data = await response.json();
            return data;
        } catch (error) {
            console.error('Error fetching data:', error);
            return [];
        }
    }

    // Function to populate the log table with data
    async function populateLogTable() {
        const logs = await fetchData();

        logs.forEach(log => {
            const row = document.createElement('tr');
            row.innerHTML = `
                <td>${log.date}</td>
                <td>${log.ip}</td>
                <td>${log.content}</td>
                <td>${log.score}</td>
            `;
            logTableBody.appendChild(row);
        });
    }

    // Call the function to populate the log table
    populateLogTable();
</script>
```



Security Dashboard

Date	IP Address	Packet Content	GPT-4 Score
2023-10-13	192.168.1.1	Sample packet content	8

Advantages

1. Proactive Threat Detection: - Advantage: Using OpenAI GPT-4 enables proactive threat detection, which identifies potentially dangerous codes before they have a chance to do damage.

2. Improved Security Position: - Advantage: By adding another line of defense against online attacks, the initiative helps to strengthen web applications' overall security posture.

3. Advantage: - Scalability: Because of its scalable architecture, the solution can easily accommodate bigger systems and fluctuating traffic volumes without sacrificing its functionality.

4. AI-Powered Analysis: - Advantage: Using artificial intelligence (GPT-4) for analysis raises the bar and makes it possible for the system to develop and recognize new threats.

5. Privacy Protection for Users: - Advantage: The two-tiered logging technique, which logs just IP addresses of benign packets, strikes a compromise between security and user privacy.

6. Automated Spam Request Blocking: - Advantage: Performance and security are enhanced by the system's capacity to automatically block persistent spam HTTP requests.

7. All-inclusive Logging System: - Advantage: Post-analysis is facilitated by the comprehensive logging system, which makes it simpler to identify patterns, evaluate possible risks, and carry out in-depth investigations.

8. Simple User Authentication and Registration: - Advantage: Users benefit from flexibility and ease in the user registration procedure, which includes choices for email and Gmail registration.

9. Prospect for a Business Model: - Advantage: By offering a security solution that solves the significant financial losses connected to data breaches and leaks, the initiative presents a workable business model.

10. Open-Source Frameworks: - Advantage: By using open-source frameworks, the project may be completed more affordably, community contributions are welcomed, and continual progress is promoted.

Disadvantages

1. GPT-4 Dependency: - Disadvantage: Because of the project's dependence on OpenAI GPT-4, in the event that the AI service is unavailable, there might be a single point of failure.

2. The Price of AI Services: - Disadvantage: High operating expenses might arise from using advanced AI services, particularly if the project becomes very popular and needs a lot of processing power.

3. Complexity for Small-Scale Implementation: - Disadvantage: Adoption by smaller enterprises may be hampered by the solution's perceived excessive complexity for lesser-scale deployments.

4. Potential False Positives: - Disadvantage: There is a chance that the analysis will produce false positives, which might result in the banning of harmless traffic and possible interference for authorized users.

5. Initial Learning Curve: - Disadvantage: There can be a learning curve for users as they become used to comprehending and using the unique two-step validation process.

6. Restricted Data Extraction Scope: - Disadvantage: When compared to more sophisticated extraction techniques, the depth and richness of data may be restricted when obtaining data in the form of a string.

7. Email Verification Procedure: - The disadvantage: Although the email confirmation method improves security, it might cause delays or annoyance to users throughout the registration process.

8. Dependence on Cloud Services: - Disadvantage: Reliance on cloud services entails possible risks associated with disruptions in service, security lapses, or modifications to the policy of the cloud service provider.

9. Cloud Storage Privacy Concerns: - Disadvantage: While keeping logs on cloud services preserves user privacy, it may also present issues with data security and privacy law compliance.

10. Ongoing Inspection and Upkeep: - The disadvantage: To guarantee that the project's threat detection is successful, constant monitoring and maintenance are needed.

Conclusion

To sum up, we have effectively tackled the issues related to web application security with our project, "Enhanced Web Application and Server Security Using Secure Packet Evaluation and Logging System Using OpenAI GPT-4." Proactive threat identification and overall system resilience have advanced significantly with the use of OpenAI GPT-4 in an innovative two-step validation approach.

We have succeeded in achieving scalability, putting in place a thorough logging system, and introducing a solution that respects user privacy while simultaneously enhancing security. The research has made a significant contribution to the world of cybersecurity by demonstrating the ability of AI-driven analysis to detect risks that are both current and emerging.

There are chances for more improvements as we move forward, such enhancing threat detection precision, investigating new AI models, and broadening the project's scope to handle changing cybersecurity issues. Future initiatives will surely be guided by the lessons acquired from this project, which emphasize the value of adaptation and constant development in the dynamic field of web security.

We sincerely thank the project team for their invaluable contributions to this initiative's success. We are enthusiastic about the potential influence our effort can have on improving web application security and contributing to the larger area of cybersecurity given the successful outcomes and lessons learnt.

Future Scopes

1. Incorporation of Advanced AI Models: - Examine how threat analysis could benefit from the incorporation of more sophisticated or specialized AI models. In order to improve the precision and comprehensiveness of harmful code detection, think about implementing more recent iterations of language models or other AI technologies.

2. Fine-Tuning for Specific Industries: - Adjust the AI models for hazards unique to a certain industry in order to customize the solution to that industry or sector. Different industries could have different cybersecurity problems, therefore tailoring the solution can make it work better in a range of settings.

3. Real-time Threat Intelligence Feed: - To ensure that the system is updated with the most recent threat data, put in place a real-time threat intelligence feed. This may entail utilizing the data to improve the system's threat detection capabilities through integration with external threat information sources.

4. Dynamic Threshold Adjustment: - Put in place mechanisms for dynamic threshold adjustment that can change in response to evolving danger environments and traffic patterns. This may entail the use of machine learning algorithms that automatically modify detection criteria in response to past performance and emerging patterns.

5. Behavioral Analysis for Anomaly Detection: Examine how behavioral analysis methods may be applied to anomaly detection. Through an examination of users' and apps' typical behavior, the system is better equipped to spot odd patterns that could point to malicious activity.

6. User Behavior Analytics: - Incorporate this information to improve comprehension of typical user behaviour. This can provide a more sophisticated approach to security by assisting in the differentiation of possibly harmful behaviour from normal user behaviours.
7. Comprehensive Reporting and Analytics: - Provide customers an in-depth understanding of their web traffic, threat landscape, and system performance by

developing capabilities for comprehensive reporting and analytics. As a result, users may be more equipped to make judgements and fortify their security measures.

8. Multi-Cloud and Hybrid Cloud Support: - Expand the compatibility of the solution to accommodate environments with multiple clouds and hybrid clouds. This guarantees that companies with a variety of cloud infrastructures may incorporate the security solution into their current configurations without difficulty.

9. Collaboration with Threat Intelligence Platforms: - Exchange information and insights by working together with reputable threat intelligence platforms. The project's capabilities can be improved by this partnership by utilizing a larger network of threat intelligence sources.

10. Blockchain Technology Incorporation: - Examine how blockchain technology may be used to provide safe, unchangeable logging. Blockchain can give the logging system an additional degree of integrity, guaranteeing the reliability of the data that is saved.

****