TENEX.AI

## Assignment

Full-Stack Cybersecurity Application

Take Home Exercise

### Objective

Build a full-stack web application which allows users to upload log files (pick your favorite log format, if you are looking for inspiration, we suggest ZScaler Web Proxy Logs but you can choose other type of logs as well like server logs, application logs),parse them, and display the information in a human-consumable format including learnings you believe are most important for a SOC analyst such as a summarized timeline of events. The application should provide a user-friendly interface for uploading files, display the results of the analysis, and include basic authentication for security.

### Ground rules

- It is totally OK (in fact recommended) to use AI to perform the take home exercise.
- Only submit code you are comfortable thoroughly explaining at a later stage in the interview process.
- Focus on functionality over production-readiness.

### Bonus

Implement an anomaly detection feature that analyzes the uploaded log file for unusual patterns or behaviors.

- Highlight the anomalous entries in the results displayed to the user.
- Provide a brief explanation of why the entry was flagged as anomalous (e.g., "Unusual number of requests from a single IP in a short time frame").
- Include a confidence score for each anomaly detected.

### Requirements

1. Frontend:
    - Build a simple, responsive web interface where users can:
        - Log in (basic authentication).

        - Upload log files (e.g., .txt or .log files).
        - View the results of the analysis in a clear and concise format (e.g., a table or chart).
    - Use Typescript and a modern frontend framework like React, Next.js.
2. Backend:
    - Build a RESTful API to handle:
        - File uploads and storage.
        - Processing the uploaded log files and running AI-based threat detection.
    - Use a backend framework like Go, Flask (Python) or Node.js (Express).
3. AI:
    - If you are using AI/LLMs, please document clearly how and where you are using AI to perform a particular task.
4. Database:
    - If a database is needed, use one of the modern databases like PostgreSQL.
5. Deployment:
    - Provide instructions for running the application locally (e.g., using Docker or a simple setup guide).
    - Bonus: Deploy the application to a cloud platform (e.g., GCP or Vercel) and share the live link.

## Deliverables

1. A GitHub repository with:
    - The full source code for the application, please share with venkata@tenex.ai
    - A README.md file with:
        - Instructions for setting up and running the application locally.
        - A brief explanation of the AI model or approach used for anomaly detection.
    - Example log files for testing.

2. (Optional) A live demo link if deployed.

### Expected Time Commitment

This exercise is designed to take 6-8 hours to complete. Focus on building a functional prototype rather than a production-ready application.

## Submission

### Files

Upload File(s)

or drag and drop here

### Notes

Write here...

Submit Assignment