Data Elicit: Technical Round

Welcome to the technical coding round!

You are given two scenario-based problem statements that reflect real-world challenges in SIEM, Data Engineering, and Cybersecurity - the core areas we work in.

- **Problem Title 1: Daily Temperature Analysis** (Difficulty Level: Easy)
- **Problem Title 2: Brute Force Login Detector** (Difficulty Level: Hard)

You are **free to choose any one** to begin with.

If you complete one successfully, you may proceed to attempt the second.

Use of Generative AI tools (e.g., ChatGPT, GitHub Copilot, Gemini, etc.) is NOT allowed.

You are allowed to search the internet and refer to documentation/tutorials (e.g., StackOverflow, blogs, etc.)

Total Time Limit: 2 hours

Submit your code files in this **Google Form**

Problem Title 1: Daily Temperature Analysis

Problem Scenario:

You've joined a Smart City startup developing **HeatSense Analyzer** — a system that processes temperature readings from **IoT sensors** deployed across the city. Each sensor reports an **hourly temperature reading** tagged with a **timestamp**.

Your job is to implement the **Daily Analyzer Module**, which will:

1. Aggregate temperature readings **by day**
2. Compute statistics:
    - Minimum temperature
    - Maximum temperature
    - Average temperature (rounded to 2 decimal places)
3. Trigger a heat alert if the **daily maximum temperature** exceeds a given threshold (e.g. `40°C`)

Input:

You are given a CSV file named `temperature_data.csv` with the following columns:

```
timestamp,temperature
2025-07-20T08:00:00,41.2
2025-07-20T11:00:00,40.5
2025-07-20T13:00:00,41.8
2025-07-20T16:00:00,40.1
2025-07-21T07:00:00,29.7
2025-07-21T10:00:00,31.4
2025-07-21T14:00:00,33.2
2025-07-22T09:00:00,40.9
2025-07-22T12:00:00,42.1
2025-07-22T15:00:00,41.6
2025-07-22T18:00:00,40.3
2025-07-23T08:00:00,30.6
2025-07-23T11:00:00,32.7
2025-07-23T13:00:00,33.5
2025-07-24T07:00:00,41.4
2025-07-24T09:00:00,42.3
2025-07-24T12:00:00,40.2
2025-07-24T15:00:00,41.1
```

- `timestamp`: ISO 8601 format (e.g., `2025-07-20T08:00:00`)
- `temperature`: A float representing degrees Celsius

Output:

For each **unique date**, output the following fields:

```
date,min_temperature,max_temperature,avg_temperature,aler
```

Where:

- `date`: in `YYYY-MM-DD` format
- `min_temperature`, `max_temperature`, `avg_temperature`: floats rounded to 2 decimal places
- `alert`: `"YES"` if max > 40.0°C, else `"NO"`

Expected Output:

```
date,min_temperature,max_temperature,avg_temperature,alert
2025-07-20,40.1,41.8,40.9,YES
2025-07-21,29.7,33.2,31.43,NO
2025-07-22,40.3,42.1,41.23,YES
2025-07-23,30.6,33.5,32.27,NO
2025-07-24,40.2,42.3,41.25,YES
```

Problem Title 2: **Brute Force Login Detector**

Problem Scenario:

You work for a cybersecurity company that monitors authentication logs from multiple servers. Your task is to detect potential brute-force login attempts by analyzing the login logs.

> A brute-force login attempt is defined as **more than 10 failed login attempts within a 5-minute sliding window** from the **same IP address and user**.

Such patterns usually indicate a brute force attack where an attacker is trying multiple passwords for the same account in a short time.

Once Detected:

- You **trigger an alert**
- You **assign a severity** based on the number of failures within that 5-minute window:
  - **Low Severity**: More than 10 attempts
  - **High Severity**: More than 10 attempts + Successful attempt
- Severity should be updated from LOW to HIGH when successful attempt occurs once brute force is detected and a single alert should be triggered.

Input:

Create a JSON file `auth_logs.json`, which contains logs with the following format:

- `timestamp` - ISO 8601 format
- `ip` - IP address of the user
- `user` - Username
- `status` - "FAIL" or "SUCCESS"

```
[
  {"timestamp": "2025-07-25T10:01:35", "username": "david", "ip": "10.9.1.1", "status": "SUCCESS"},
  {"timestamp": "2025-07-25T10:01:35", "username": "carol", "ip": "172.154.1.1", "status": "FAIL"},
  {"timestamp": "2025-07-25T10:24:30", "username": "bob", "ip": "10.0.0.1", "status": "FAIL"},
  {"timestamp": "2025-07-25T10:01:00", "username": "alice", "ip": "192.168.1.1", "status": "FAIL"},
  {"timestamp": "2025-07-25T10:02:30", "username": "alice", "ip": "192.168.1.1", "status": "FAIL"},
  {"timestamp": "2025-07-25T10:23:00", "username": "bob", "ip": "10.0.0.1", "status": "FAIL"},
  {"timestamp": "2025-07-25T10:04:30", "username": "alice", "ip": "192.168.1.1", "status": "FAIL"},
  {"timestamp": "2025-07-25T10:00:00", "username": "alice", "ip": "192.168.1.1", "status": "FAIL"},
  {"timestamp": "2025-07-25T10:20:45", "username": "carol", "ip": "172.154.1.1", "status": "SUCCESS"},
  {"timestamp": "2025-07-25T10:23:30", "username": "bob", "ip": "10.0.0.1", "status": "FAIL"},
  {"timestamp": "2025-07-25T10:03:00", "username": "alice", "ip": "192.168.1.1", "status": "FAIL"},
  {"timestamp": "2025-07-25T10:22:30", "username": "bob", "ip": "10.0.0.1", "status": "FAIL"},
  {"timestamp": "2025-07-25T10:04:45", "username": "alice", "ip": "192.168.1.1", "status": "SUCCESS"},
  {"timestamp": "2025-07-25T10:22:00", "username": "bob", "ip": "10.0.0.1", "status": "FAIL"},
  {"timestamp": "2025-07-25T10:20:00", "username": "bob", "ip": "10.0.0.1", "status": "FAIL"},
  {"timestamp": "2025-07-25T10:02:00", "username": "alice", "ip": "192.168.1.1", "status": "FAIL"},
  {"timestamp": "2025-07-25T10:01:45", "username": "carol", "ip": "172.154.1.1",
"status": "SUCCESS"},
  {"timestamp": "2025-07-25T10:21:00", "username": "bob", "ip": "10.0.0.1", "status": "FAIL"},
  {"timestamp": "2025-07-25T10:25:00", "username": "bob", "ip": "10.0.0.1", "status": "FAIL"},
  {"timestamp": "2025-07-25T10:23:45", "username": "carol", "ip": "172.154.1.1", "status": "SUCCESS"},
  {"timestamp": "2025-07-25T10:00:30", "username": "alice", "ip": "192.168.1.1", "status": "FAIL"},
  {"timestamp": "2025-07-25T10:23:25", "username": "carol", "ip": "172.154.1.1", "status": "FAIL"},
  {"timestamp": "2025-07-25T10:04:40", "username": "alice", "ip": "192.168.1.1", "status": "FAIL"},
  {"timestamp": "2025-07-25T10:40:00", "username": "bob", "ip": "10.0.0.1", "status": "SUCCESS"},
  {"timestamp": "2025-07-25T10:04:00", "username": "alice", "ip": "192.168.1.1", "status": "FAIL"},
  {"timestamp": "2025-07-25T10:03:30", "username": "alice", "ip": "192.168.1.1", "status": "FAIL"},
  {"timestamp": "2025-07-25T10:24:00", "username": "bob", "ip": "10.0.0.1", "status": "FAIL"},
  {"timestamp": "2025-07-25T10:20:30", "username": "bob", "ip": "10.0.0.1", "status": "FAIL"},
  {"timestamp": "2025-07-25T10:21:30", "username": "bob", "ip": "10.0.0.1", "status": "FAIL"},
  {"timestamp": "2025-07-25T10:01:30", "username": "alice", "ip": "192.168.1.1", "status": "FAIL"}
]
```

Expected Output:

Display alerts of suspicious login attempts in JSON

```
{"timestamp": "2025-07-25T10:20:00", "username": "bob", "ip": "10.0.0.1", "failed_attempts": 11, "alert": "Brute-force detected", "severity": "LOW"}
{"timestamp": "2025-07-25T10:00:00", "username": "alice", "ip": "192.168.1.1", "failed_attempts": 11, "alert": "Brute-force detected", "severity": "HIGH"}
```