Software Engineer Take-Home Assignment

Overview

You're building a simple heartbeat monitoring system.

A service is expected to send a heartbeat every fixed interval. If it misses three heartbeats in a row, your system should trigger an alert.

Language & Tools

You may use any one of the following:

- Python
- JavaScript / Typescript (Node.js)

We will be running your code locally. Make sure your submission is:

- Easy to set up
- Includes any dependencies (e.g., requirements.txt or package.json)
- Includes instructions for how to:
    - Run your main solution
    - Run your test cases

Important Notes

- Poor or missing documentation will result in point deductions
- We value:
    - Code readability
    - Simplicity
    - Correct handling of edge cases

What You're Given

You'll receive:

- Inside the shared **Google Drive folder** (Link: Software Engineer Take-Home Assignment), you'll find a JSON file containing **heartbeat events**:
  `{ "service": "email", "timestamp": "2025-08-04T10:00:00Z" }`
- Two global parameters:
    - `expected_interval_seconds: e.g., 60`
    - `allowed_misses: e.g., 3`

Some events may be malformed (e.g., missing fields, bad timestamp formats). Your program should gracefully skip them without crashing.

Your Task

Implement a program that:

1. Sorts the events per service, chronologically
2. Tracks expected heartbeat times based on the given interval
3. Triggers an alert if a service misses 3 expected heartbeats in a row
4. Returns output like:

```
[{ "service": "email", "alert_at": "2025-08-04T10:06:00Z" }]
```

Required Test Cases

Please include the below test cases:

- A working alert case
- A “near-miss” case (only 2 missed → no alert)
- Unordered input (checks whether your solution can handle heartbeat events that don’t arrive in chronological order)
- At least 1 malformed event (missing fields or invalid timestamp)

You may use:

- unittest, pytest, Jest, assert statements - anything standard and runnable

---

What to Submit

- main.py / main.ts / main.js
- Your test file(s)
- README.md with clear run instructions
- requirements.txt or package.json if needed