Assessment for Data Engineer - Health Sensing

GOAL

Design and implement a small data pipeline to process raw event data into a cleaned and analytics-friendly dataset.

BACKGROUND

You’re working on a product that tracks user interactions (e.g., clicks, purchases) on a website. The website sends raw event logs to your backend, and your job is to transform this raw data into a structured format that downstream teams (e.g., analytics, data science) can use.

Input Data: raw event logs in JSON format, the **raw_events.json** in the zip.

Example:

```
[
  {
    "user_id": "abc123",
    "timestamp": "2025-03-01T10:15:30+00:00",
    "event_type": "click",
    "metadata": {
      "screen": "home",
      "button_id": "signup"
    }
  },
  {
    "user_id": "xyz789",
    "timestamp": "2025-03-01T10:16:30+00:00",
    "event_type": "purchase",
    "metadata": {
      "screen": "checkout",
      "amount": "49.99",
      "currency": "USD"
    }
  }
]
```

YOUR TASKS

Part 1: Data Modeling

- Define a schema for a structured version of this data.
- Consider data normalization and cleanup (e.g. how do we store metadata?)

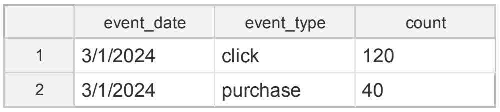
Part 2: Build a Simple Pipeline

- Write code (in Python preferred, with pandas or PySpark) that:
    - Reads the raw JSON files.
    - Validates required fields (`user_id`, `timestamp`, `event_type`).
    - Handles malformed events, log and skip them
    - Writes the transformed data to disk (structured format in CSV or Parquet).

Part 3: Simple Aggregation

- Compute the total number of events per event type per day.
- Find the total number of active users.
- Find the most active app user.

Note that output for each aggregation should be a summary table like below. Write the summary table for **each** aggregation to disk in CSV or Parquet format.



	event_date	event_type	count
1	3/1/2024	click	120
2	3/1/2024	purchase	40

DELIVERABLES

- Your code in a `.zip` or GitHub repo (include instructions to run).
- A short README (1–2 paragraphs) explaining:
  - Your approach.
  - Any assumptions made.
  - How to run the pipeline and view the output.
- Bonus: Add basic unit tests

NOTES

- This assessment should take about 1 hour to complete. Please spend no more than 2 hours working on it.
- Please work on this assessment on your own. You should not solicit help or advice from others, including artificial intelligence.