Final Group Project Report

Team

Yergazy Adil 22B22B1519
Kairov Danial 23B031324
Babayev Yerassyl 23B031232

1. Goal

Implement a complete pipeline for frequently updating real-world data: pseudo-streaming ingestion (API to Kafka), hourly batch cleaning + storage (Kafka to SQLite), and daily analytics (SQLite to summary table).

2. Data Source (API) and Justification

API: OpenAQ v3 (air-quality measurements).

- Real values from sensors (e.g., PM2.5).
- It was in the Allowed API categories.
- Structured JSON, stable/documented REST API.
- New/updated measurements typically appear hourly or more (depends on sensor).

Endpoints used:

- GET /v3/locations/{location_id}/sensors
- GET /v3/locations/{location_id}/latest

3. Architecture & Airflow DAGs

Flow: OpenAQ API -> Job 1 -> Kafka (raw_events) -> Job 2 -> SQLite (events) -> Job 3 -> SQLite (daily_summary).

Job	Schedule	What it does
DAG 1	*/10 * * * *	Poll API every POLL_SECONDS (~60s) and send raw JSON to Kafka (RUN_SECONDS=540 per run).
DAG 2	@hourly	Read new Kafka messages,

		clean with pandas, append to SQLite table events.
DAG 3	@daily	Aggregate events per day and write to SQLite table daily_summary (count/avg/min/max).

4. Kafka Topic Schema

Topic: raw_events (default; env KAFKA_TOPIC). Each message is one measurement with fields:

fetched_at_utc (text), location_id (int), sensor_id (int), parameter (text), unit (text), parameter_display (text), datetime_utc (text), datetime_local (text), value (real), latitude (real), longitude (real).

5. Cleaning Rules (Kafka -> SQLite)

- Convert value to numeric; convert datetime_utc to UTC datetime.
- Drop rows with missing sensor_id/parameter/unit/datetime_utc/value.
- Filter invalid measurements: value >= 0.
- Drop duplicates by (sensor_id, parameter, datetime_utc).
- Add ingested_at_utc = current UTC time (when storing).

6. SQLite Schema

Database: SQLite file (default data/app.db; env SQLITE_PATH).

Table: events (cleaned measurements)

Columns: id (PK), ingested_at_utc, fetched_at_utc, location_id, sensor_id, parameter, unit, parameter_display, datetime_utc, datetime_local, value, latitude, longitude.

Table: daily_summary (daily aggregates)

Columns: id (PK), created_at_utc, date_utc, parameter, unit, measurements_count, value_avg, value_min, value_max.

7. Daily analytics logic

Group by (date_utc, parameter, unit) and compute: measurements_count, value_avg, value_min, value_max. Before inserting, delete existing daily_summary rows for the same date_utc to keep reruns consistent.

8. Screenshots