Take Home Assessment

Background

Singapore's MRT and bus system occasionally (actually a lot these days) experiences breakdowns, causing disruptions for commuters. The Land Transport Authority (LTA) wants a system to track, analyze, and provide insights on breakdowns to improve reliability.

Your goal is to build a prototype analytics system that is realistic, practical, and testable.

Requirements

1. Data Ingestion

Accept sample breakdown data with these fields:
- Line/Route (MRT or bus)
- Station/Stop
- Timestamp
- Duration of disruption
- Incident type (mechanical, power, signal, etc.)

Implement a simple API endpoint to submit data:
- Endpoint: /incidents
- Method: POST
- Request Body (JSON): Candidates should define the structure themselves.
- Response (JSON): Candidates should define the response themselves.

2. Database & Schema

Design a schema that supports:
- Querying by line, station, and time range
- Aggregating metrics (e.g., breakdown counts, MTBF)
- Filtering and joining for all APIs

Minimum tables/entities:
- lines / routes
- stations
- incidents

Guidelines:
- Index frequently queried fields (timestamp, station_id, line_id) for performance.
- Schema should allow easy extension for future analytics or operational features.

3. Analytics Endpoints

/analytics/top-breakdowns — Returns top 5 lines or stations with the most breakdowns.
/analytics/mean-time-between-failures — Returns MTBF per line/route.

/analytics/recent-disruptions — Returns recent breakdowns, optionally filtered by line/station and limited in number.

Candidates should design the request and response formats themselves.

4. Optional Visualization

A simple dashboard to display trends or recent breakdowns (bonus, not required).

Technical Requirements

Component	Expectation
Backend	Any framework (Node.js/NestJS, Python/ FastAPI, etc.) with working endpoints
Database	Dockerized Postgres/MySQL with schema initialization (init.sql)
Containerization	Provide docker-compose.yml to run backend + DB easily
Documentation	README.md with setup instructions, API examples, and brief architecture explanation

Deliverables

Candidates should submit a folder structured like this:

```
transport-reliability-analytics/
├── README.md
├── docker-compose.yml
├── backend/
│   ├── Dockerfile
│   ├── src/
│   └── ...
├── database/
│   ├── init.sql
│   └── Dockerfile
└── docs/
    ├── architecture-diagram.png
    └── design.md
```

Include clear instructions on how to run the system and how to test the APIs.