TECHNICAL ASSESSMENT –
DATA LAKEHOUSE DEVELOPMENT

Deadline: 23 November 2025 (EOD)

Mode: Remote submission via GitHub repository

Objective

This assignment is designed to assess your ability to design and implement a modern Data Lakehouse architecture using industry-standard tools. The task will test your skills in data engineering, orchestration, API development, and data modelling.

Scope of Work

You are required to build a complete mini-Data Lakehouse pipeline covering the following components:

1. **API Development (FastAPI)**
   I. Choose a niche or domain of your choice (e.g., e-commerce, aviation, finance, or weather).
   II. Build a REST API using FastAPI that serves structured data (JSON).
   III. The API should support:
   IV. A route for initial (full) data extraction.
   V. A route for incremental data extraction based on timestamps or IDs.

2. **Data Orchestration (Apache Airflow)**

Create an Airflow DAG that:

   I. Extracts data from the FastAPI endpoint.
   II. Handles both Day 0 full loads and incremental loads.
   III. Store the extracted raw data into MinIO (S3-compatible storage).
   IV. Ensure the DAG is modular, parameterized, and follows best practices.

3. **Data Processing (Apache Spark)**

Develop a Spark job to:

   I. Read raw data from MinIO.
   II. Perform necessary transformations (cleaning, enrichment, or standardization).
   III. Write the processed data into Apache Iceberg tables (transformed layer).
   IV. Use appropriate partitioning and schema evolution features of Iceberg.

4. **Data Modelling (Golden Layer)**
   I. Implement a dimensional model (star/snowflake schema) for your domain.

II. Build the golden layer by transforming Iceberg tables into fact and dimension tables.
III. Justify your modelling choices briefly in the documentation.

5. Containerization

Use Docker (or Docker Compose) to containerize all components:

I. FastAPI
II. Airflow
III. Spark
IV. MinIO

The setup should be runnable locally with a single command (e.g., docker compose up).

Deliverables

1. GitHub Repository containing:
    a. Source code for FastAPI, Airflow DAGs, and Spark transformations.
2. Docker setup files (Docker file, docker-compose).
3. Configuration and environment setup scripts (if any).
4. Documentation (README.md) explaining:
5. System architecture overview.
6. Setup and run instructions.
7. API endpoints and example payloads.
8. Airflow DAG logic and scheduling details.
9. Data model description (ERD or schema diagram preferred).
10. Challenges faced and learnings.
11. Share the Public Repository Link in Mail

Functional Validation

The evaluator should be able to clone the repo, run docker compose up, and see the pipeline running end-to-end.

Evaluation Criteria

Category	Weightage
Architecture C Design	25%
Code Quality C Modularity	20%
Airflow DAG Implementation	20%
Spark C Iceberg Data Transformation	20%
Documentation C Clarity	15%
Total	**100%**

Bonus Points

- Use of environment variables or config files for dynamic parameters.
- Demonstrating data validation or quality checks.
- Implementing logging or monitoring in Airflow/Spark.
- Clear dimensional model diagrams in documentation.