Group 2 Solar Energy Farm Project

Team Lead: **Joseph Bello**

Team Members:

- Adekola Adeboye
- Oluwaseun Afolabi

1. Introduction

The purpose of this project is to design and implement a data pipeline that processes, stores, and visualizes solar energy generation and household consumption data. The system will help monitor solar panel performance, household usage, battery storage, and overall efficiency of the solar farm. The project covers the following: Data Engineering (ETL pipeline), Database Design (PostgreSQL schema), and Visualization (Grafana dashboard).

2. Project Objectives

1. Collect and process solar energy farm production and household consumption data.
2. Transform raw logs into meaningful metrics such as Efficiency (%), Battery State of Charge (SOC), and Peak energy usage hours.
3. Store processed data in a PostgreSQL time-series database.
4. Create interactive dashboards in Grafana for monitoring and analysis.

3. Data Sources

- Solar Panel Output Logs – records of energy generated (kWh).
- Household Consumption Logs – records of energy consumed (kWh).
- Battery Storage Logs – charge and discharge cycles (SOC %).
- Timestamps – every log is time-series data.

4. ETL Pipeline Design

Extract:
- Data collected from solar panels, smart meters, and battery systems in CSV/JSON format.
Transform:
- Cleaning, timestamp formatting, and metric calculations (Efficiency %, Net Energy, Battery SOC, Peak Hours).
Load:

- Transformed data stored in PostgreSQL database.

5. PostgreSQL Schema

```
CREATE TABLE solar_data (
id SERIAL PRIMARY KEY,
timestamp TIMESTAMP NOT NULL,
generation_kwh FLOAT,
consumption_kwh FLOAT,
battery_soc FLOAT,
efficiency_percent FLOAT,
net_energy FLOAT
);

CREATE INDEX idx_timestamp ON solar_data(timestamp);
```

6. Visualization (Grafana Dashboard)

1. Solar Generation vs Consumption Chart – line graph.
2. Efficiency % Over Time – trend monitoring.
3. Battery SOC Monitoring – charge/discharge cycles.
4. Peak Usage Hours – heatmap visualization.

7. Expected Benefits

- Real-time monitoring of solar energy production and usage.
- Identification of inefficiencies.
- Improved decision-making.
- Scalable for IoT integration and predictive analytics.

8. Tools & Technologies

- Programming: Python (ETL pipeline, Pandas, SQLAlchemy).
- Database: PostgreSQL (time-series schema).
- Visualization: Grafana.
- Data Processing: Pandas, Apache Airflow (optional).

9. Conclusion

This project provides a complete data engineering workflow for managing solar farm energy data. By implementing an ETL pipeline, structured database storage, and a Grafana dashboard, the solar farm can monitor energy flow efficiently and make data-driven decisions.