IoT Environmental Monitoring Using MQTT, SQL, MongoDB, and Neo4j

Name: Kahathota Liyanage Yasindu Janiya Weerasinghe\ **Matriculation Number:** 557466\ **Course:** Database B\ **Professor:** Armando Ruggeri

---

1. Project Objective

The objective of this project is to simulate a basic Internet of Things (IoT) system that collects environmental data such as temperature, humidity, and air quality using Python. These data points are transmitted via MQTT and stored across three different types of databases:

- **Relational** (MySQL)
- **Document** (MongoDB)
- **Graph** (Neo4j)

The system demonstrates real-time data collection, processing, and storage using industry-relevant tools and technologies.

---

2. Tools and Technologies Used

- **Python 3.13**: Main programming language
- **Paho-MQTT**: Python MQTT client
- **MySQL**: SQL database for temperature data
- **MongoDB**: NoSQL database for humidity data
- **Neo4j**: Graph database for air quality data
- **Docker & Docker Compose**: Container management for MongoDB and Neo4j
- **VS Code**: Development environment
- **GitHub**: Source code hosting and version control

---

3. System Architecture

```
+-------------------+       MQTT       +-------------------+
| Temperature       | --------------> |                   |
| Humidity          |                |     Python        |
| Air Quality       | <-------------- |   Subscriber      |
+-------------------+     Publish     +-------------------+
                                          |     |     |
                                    v        v        v
                                  MySQL   Mongo   Neo4j
```

- **Sensor Publisher (Python)**: Simulates sensors and publishes messages to MQTT topics.
- **Subscriber (Python)**: Listens to MQTT topics and routes data.
- **Database Layer**:
- MySQL stores temperature with timestamps and sensor IDs
- MongoDB stores humidity as documents
- Neo4j stores air quality as nodes with relationships possible

---

4. How the System Works

1. The Python script `sensor_publisher.py` generates random sensor data (temperature, humidity, air quality) with sensor IDs and timestamps.
2. It publishes the data using MQTT protocol to the local Mosquitto broker.
3. The `sensor_subscriber.py` script subscribes to MQTT topics.
4. Based on the topic, the data is stored:
5. Temperature → MySQL with timestamp and CSV log
6. Humidity → MongoDB
7. Air Quality → Neo4j
8. Docker is used to deploy MongoDB and Neo4j locally for testing.

---

5. Result and Conclusion

The system worked successfully. All three types of environmental data were published and stored in real-time using a modular and scalable architecture. The integration of multiple databases via MQTT and Python showcases practical knowledge in database interoperability, IoT data streams, and containerization.

This project strengthened understanding of database types, real-time communication, and Docker-based deployments.

GitHub Repository:\ https://github.com/Yasindu01/iot-environmental-monitoring