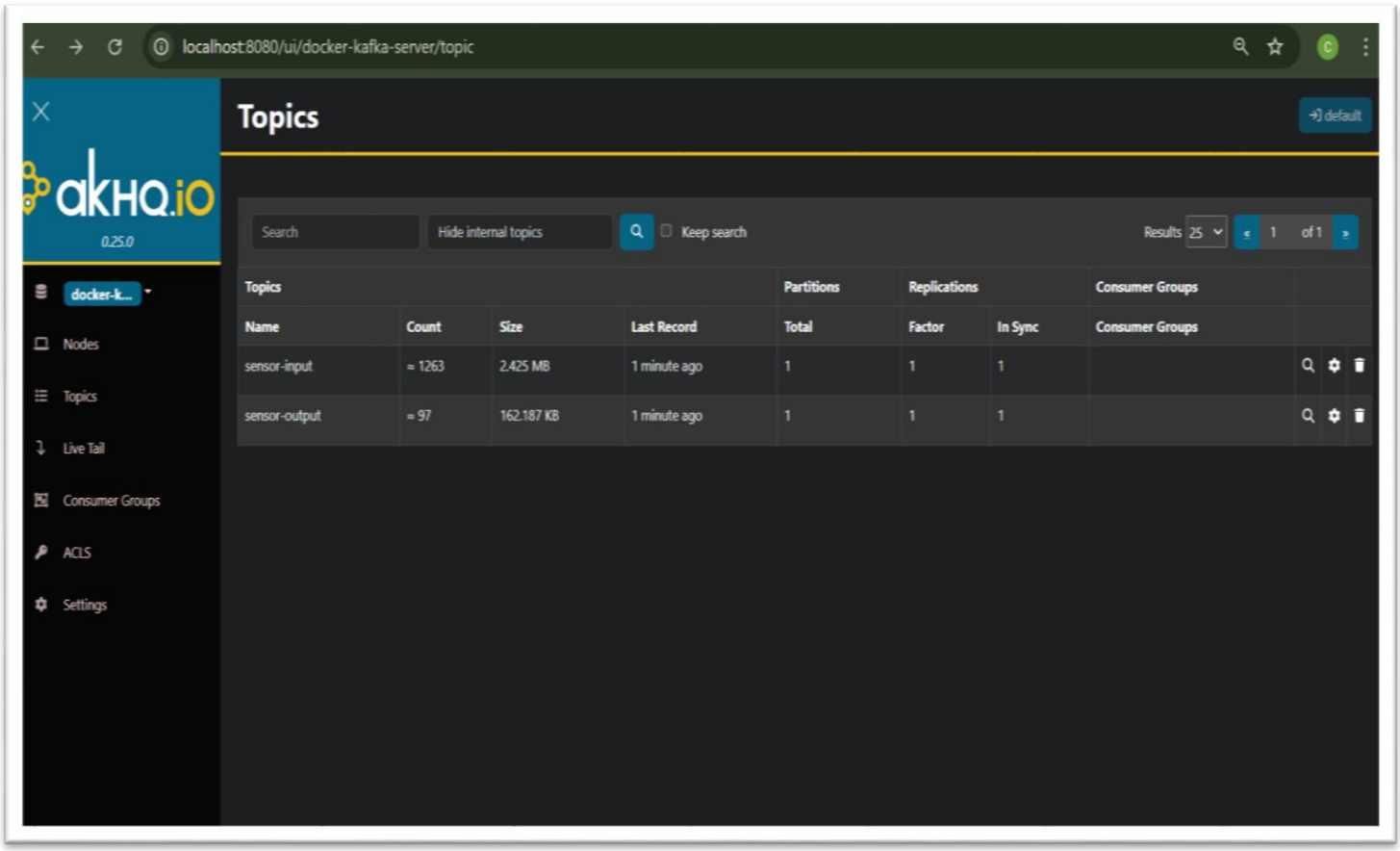
Real-Time Sensor Data Processing

Overview:

In this project, I developed a real-time data processing pipeline to handle sensor data streamed via Kafka. Each message includes a sensor ID, a value, and a timestamp in milliseconds. The goal was to compute the 1-minute average value for each sensor in real time. Using PySpark Structured Streaming, I read data from the Kafka input topic, applied window-based aggregation, and published the results to an output Kafka topic. Additionally, I persisted the results to MongoDB for further analysis. This solution demonstrates scalable, fault-tolerant stream processing suitable for real-world IoT applications.

Visual Output Overview:

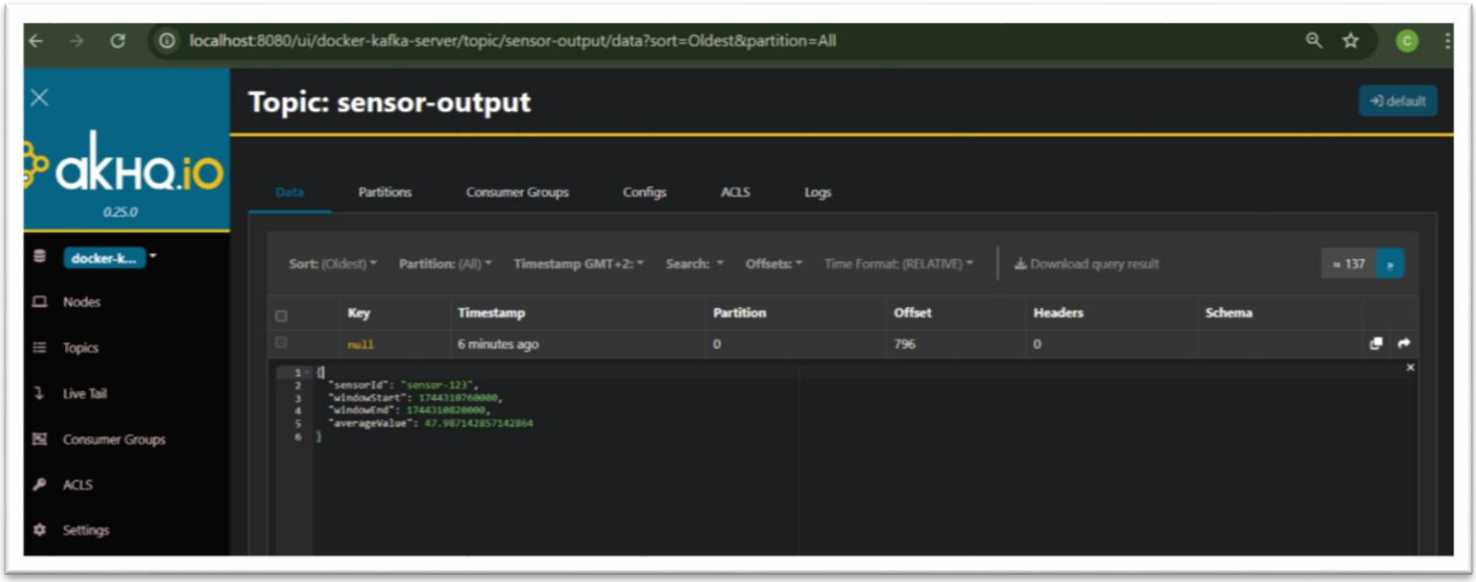
1. **The sensor input data was read, processed according to the defined aggregation logic, and the results were sent to the sensor output.**



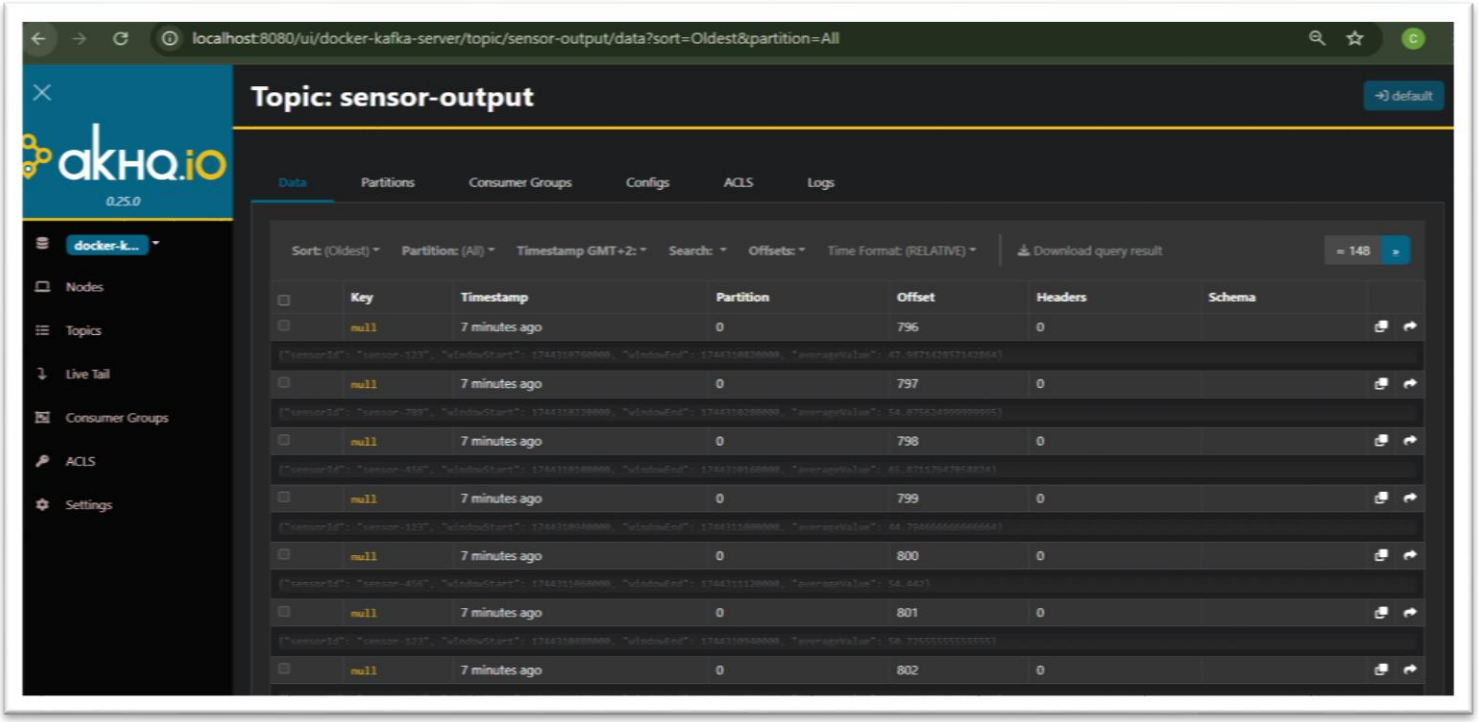
2. **The sensor-output topic contains the final processed data from the sensors. Each message is in JSON format and shows the average value of a sensor within a specific 1-minute time window.**

   Description of Fields:

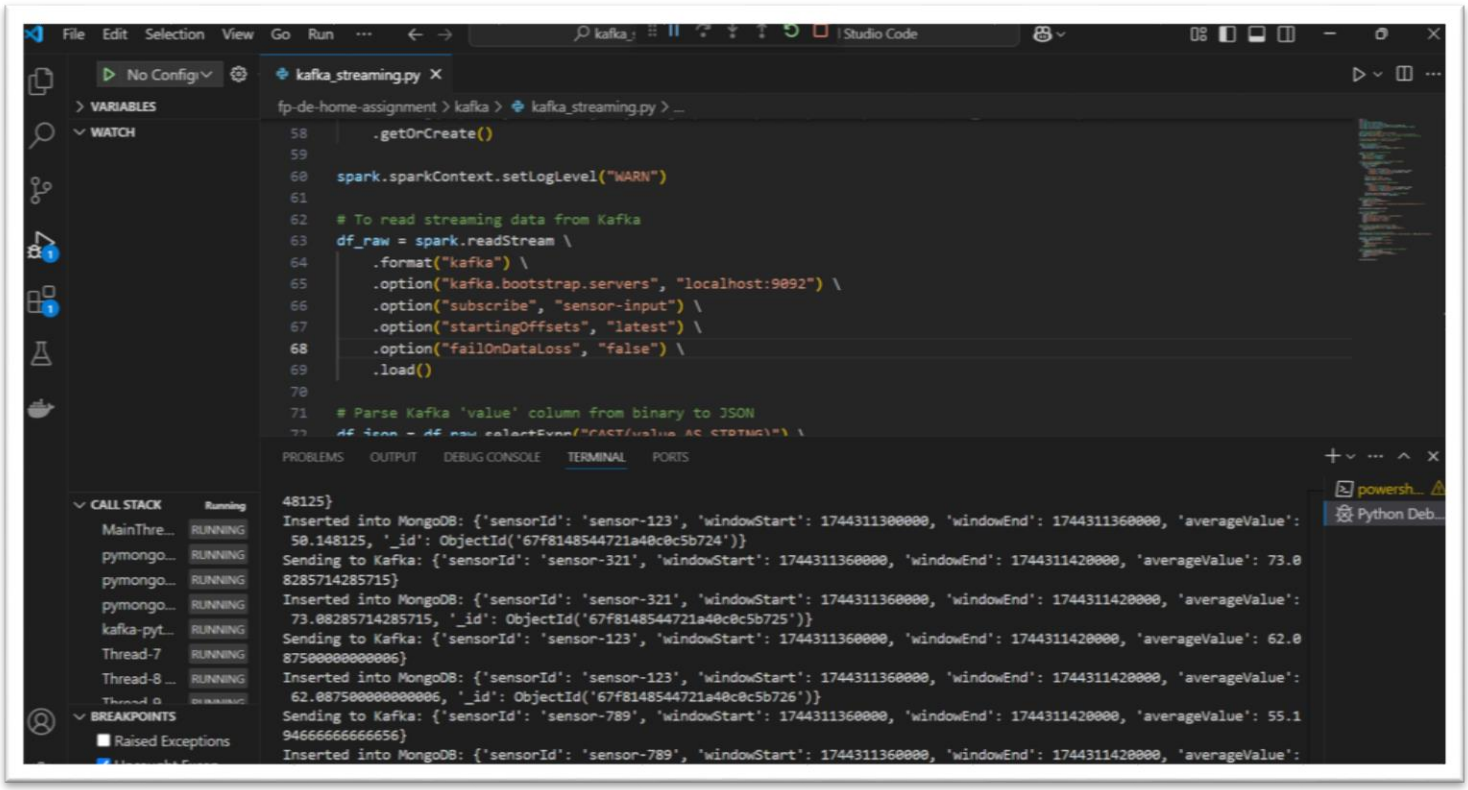
- **sensorId**: The ID of the sensor.
- **windowStart**: Start time of the 1-minute window.
- **windowEnd**: End time of the 1-minute window.
- **averageValue**: The average of all readings from that sensor during the time window.



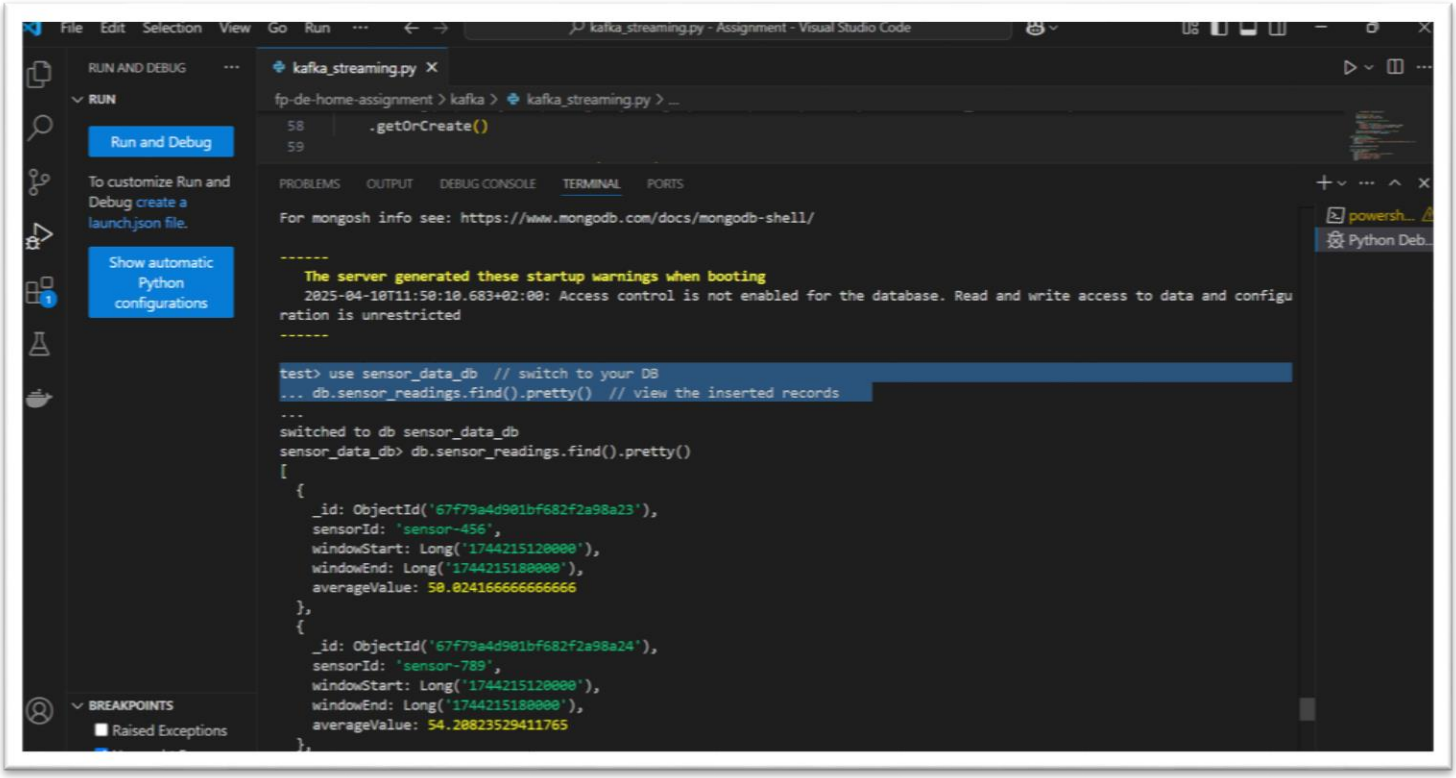
3. **This screenshot shows a list of JSON messages published to the sensor-output topic. Each message represents the average sensor reading within a specific 1-minute time window.**
   Multiple such messages are generated in real time, one for each sensor and time window



4. **This screenshot displays a list of JSON messages printed in the Visual Studio Code terminal. This output confirms that the data is being processed correctly and that the results are ready to be sent to the sensor-output Kafka topic and stored in a database (MongoDB).**



5. **This screenshot shows a successful MongoDB connection established within Visual Studio Code. The data from the sensor-output Kafka topic has been written to the MongoDB database named sensor_data_db**



6. **This screenshot shows the MongoDB Compass UI connected to the database sensor_data_db, specifically viewing the sensor_readings collection.**