CS 331 -COMPUTER NETWORKS

Prof. SAMEER G.KULKARNI

SMART GUARD

(Centralized IoT Environmental Monitoring System)

Chepuri Venkata Naga Thrisha (23110079)
Katta Revathi (23110159)
Pappala Sai Keerthana (23110229)
Thoutam Dhruthika (23110340)

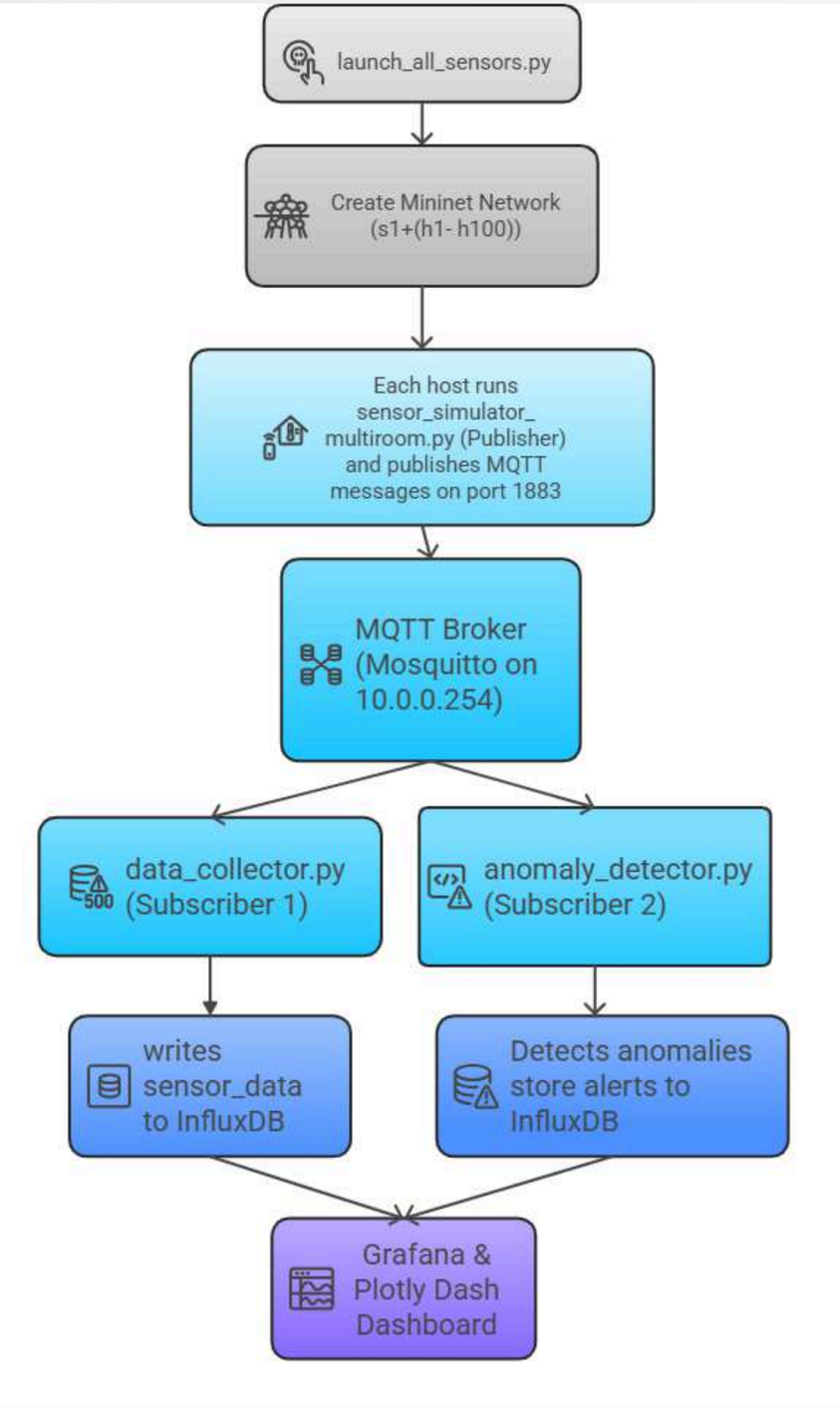
PROBLEM STATEMENT

Manual monitoring of classroom and lab environments is time-consuming and lacks real-time updates. Issues like high CO_2 levels may go unnoticed, and managing multiple rooms is challenging due to the scattered and uncentralized nature of data.

This project addresses these challenges by developing a centralized monitoring system that efficiently collects and visualizes data from multiple simulated sensors using Mininet, providing real-time insights and efficiently managing multiple spaces from a single interface.

OBJECTIVES

1. Develop a centralized system to collect real-time temperature, humidity, light, and CO_2 data from classrooms and labs.
2. Simulate a scalable 100-node sensor network using Mininet to represent virtual IoT clients.
3. Implement asynchronous data transmission between sensors and the server using the MQTT protocol.
4. Store collected sensor data efficiently in an InfluxDB time-series database for historical analysis.
5. Implement threshold-based alerting to detect environmental hazards like overheating or poor air quality.
6. Establish automated pipelines to continuously collect temperature, humidity, CO_2, and light intensity data
7. Build comprehensive Grafana dashboards featuring heatmaps, graphs, and alert panels for intuitive data exploration.



SYSTEM ARCHITECTURE

Simulation Layer:

- **Implemented using Mininet**
- **Each host acts as a simulated IoT sensor node**
- **Generates data (Temperature, Humidity, Light, Air Quality)**
- **Publish readings via MQTT topics (e.g., sensors/temperature/CR101/h1)**

Communication Layer:

- **Mosquitto MQTT Broker routes messages between publishers and subscribers**
- **Enables lightweight, real-time data transfer**

Processing Layer:

- **Data Collector Service: Stores sensor data into InfluxDB (time-series DB)**
- **Anomaly Detector Service: Detects abnormal readings and logs alerts**

Visualization Layer:

- **Grafana Dashboard displays real-time and historical trends**
- **Also accessible via a web browser for centralized room monitoring**

WHAT IS MQTT?

MQTT (Message Queuing Telemetry Transport)

- A lightweight publish–subscribe protocol designed for IoT and sensor networks.
- Works through a broker that routes messages between publishers (senders) and subscribers (receivers).
- Operates over TCP/IP and uses minimal bandwidth.
- Ideal for real-time, low-power, and high-latency environments

WHY IS IT USED?

- Lightweight and efficient for large numbers of IoT sensors.
- Reliable data delivery with low network overhead.
- Supports topic-based filtering for flexible data routing.
- Easy to integrate with Python, InfluxDB, and Grafana.

Alternative Protocols (Not Used):

- HTTP/REST: Heavyweight; adds overhead due to frequent connection setup.
- CoAP (Constrained Application Protocol): Lightweight but limited broker support and a less mature ecosystem.
- AMQP (Advanced Message Queuing Protocol): More complex, designed for enterprise messaging rather than IoT.
- WebSockets: Real-time, but lacks built-in topic-based message management.

METHODOLOGY

1. Network Interfaces and IPs in VM
2. Network Setup Script (setup_network.sh)
3. Network Simulation using Mininet (launch_all_sensors.py)
4. Sensor Data Simulation (sensors/sensor_simulator_multiroom.py) and MQTT Communication
5. Data Collection (server/data_collector.py) and Storage in InfluxDB
6. **Anomaly Detection (**server/anomaly_detector.py**)**
7. Visualization with Grafana Dashboard
8. REST API Backend (web_dashboard.py)
9. Custom Web Dashboard Frontend (app.py)

2 **Network Setup Script (setup_network.sh):**

- We configure IP forwarding and NAT using a Bash script to enable Mininet hosts (in the 10.0.0.0/8 network) to access external networks.
- This setup routes traffic through the host machine's IP, enabling full internet connectivity for simulated nodes.

3 **Network Simulation using Mininet (launch_all_sensors.py):**

- Sets up a virtual network with 1 switch (s1) and 100 hosts (h1–h100), each host representing a unique room and IoT sensor.
- Each host runs a sensor script that sends real-time environmental data to the MQTT Broker (10.0.0.254:1883).

4 **Sensor Data Simulation (sensors/sensor_simulator_multiroom.py) and MQTT Communication:**

- Each Mininet host simulates temperature, humidity, light, and CO_2 readings every 5 seconds with realistic randomness.
- Publishes values via plain MQTT (no auth/TLS) to Mosquitto broker at 10.0.0.254:1883.
- Data sent to topic: sensors/<metric>/<room>/<sensor_id> using paho-mqtt library.

5 **Data Collection (server/data_collector.py) and Storage in InfluxDB:**

- Subscribes to all sensor topics on MQTT broker (10.0.0.254:1883) using paho-mqtt.
- Parses JSON data to extract room, sensor ID, metric, and timestamp.
- Stores readings in InfluxDB for fast ingestion and easy historical analytics, enabling later filtering and visualization.

6

Anomaly Detection (server/anomaly_detector.py):

- Subscribes to MQTT sensor data stream.
- Checks readings against thresholds (e.g., CO_2 > 1000 ppm, light < 300 lux).
- On anomaly: prints real-time alerts and logs them to InfluxDB for later analysis.

8

Backend – REST API (web_dashboard.py):

- Flask-based service exposing sensor and alert data from InfluxDB (port 8086).
- Endpoints for registering sensors, listing rooms, getting latest/historic readings, and thresholds/alerts.
- Runs on port 5000, serving HTTP data for web dashboards and other applications.

7

Visualization with Grafana Dashboard:

- Visualization with Grafana Dashboard
- Connects to InfluxDB to display real-time graphs, heatmaps, and alert panels.
- Visualizes temperature, humidity, light, and CO_2 across rooms.
- Accessible via browser at http://192.168.56.103:3000.
- Supports centralized, data-driven monitoring and decision-making.

9

Frontend – Custom Web Dashboard (app.py):

- Built with Dash (Plotly) for interactive, real-time visualization.
- Fetches sensor and alert data from Flask API (port 5000).
- Displays metrics, heatmaps, alerts; supports room filtering and history.
- Runs on port 8050, accessible in any web browser.

OBJECTIVES ACHIEVED

The terminal output below shows real-time sensor data being collected and logged via MQTT, confirming successful data ingestion by the collector for dashboard analysis.

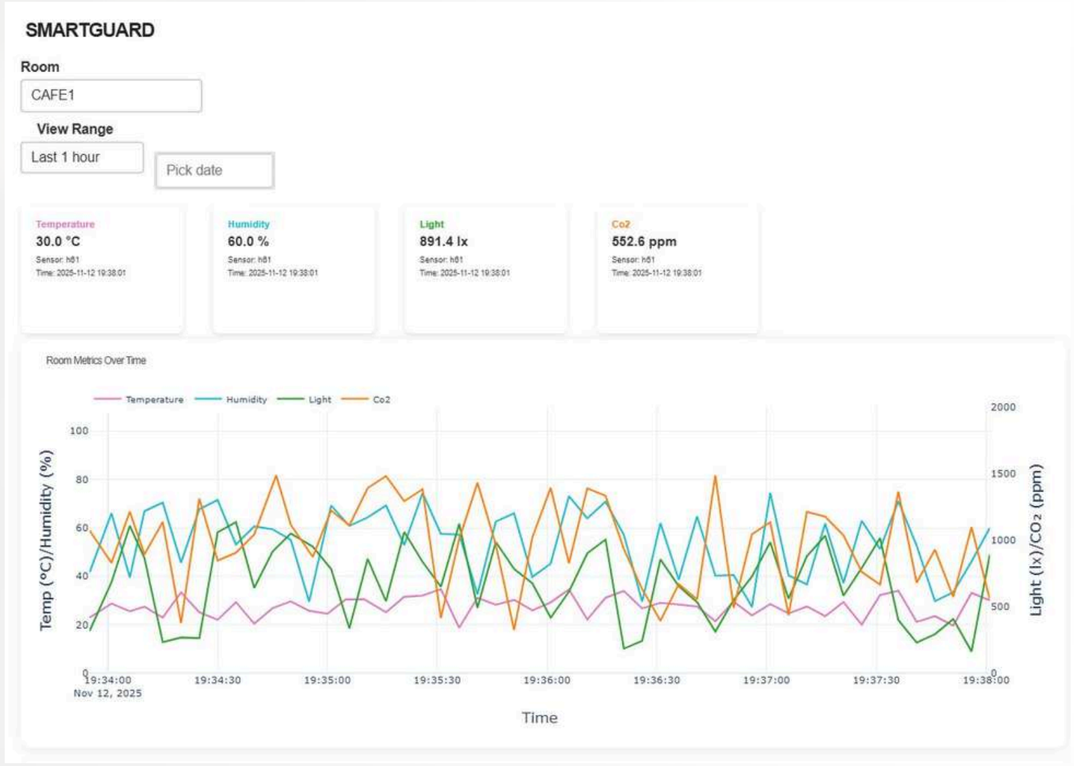
```
mininet@mininet-vm:~/environmental_monitoring$ source venv39/bin/activate
etector.py
(venv39) mininet@mininet-vm:~/environmental_monitoring$ python server/anomaly_detector.py
/home/mininet/environmental_monitoring/server/anomaly_detector.py:81: DeprecationWarning: Callback API version 1 is deprecated, update to latest version
  client = mqtt.Client()
SSL module imported successfully
Anomaly Detector started...
======================================================================
ANOMALY DETECTOR - Connected to MQTT Broker
======================================================================
[ALERT] ↑ HIGH | TEMPERATURE | Room: CR305    | Sensor: h85         | Value:    31.2
[ALERT] ↓ LOW  | LIGHT       | Room: CAFE3    | Sensor: h63         | Value:   174.7
[ALERT] ↓ LOW  | LIGHT       | Room: LIB2     | Sensor: h42         | Value:   224.0
[ALERT] ↓ LOW  | LIGHT       | Room: HALL1    | Sensor: h31         | Value:   257.0
[ALERT] ↓ LOW  | LIGHT       | Room: ROOM3    | Sensor: h93         | Value:   230.7
[ALERT] ↑ HIGH | CO2         | Room: ROOM3    | Sensor: h93         | Value:  1238.8
[ALERT] ↑ HIGH | CO2         | Room: CR307    | Sensor: h87         | Value:  1001.6
[ALERT] ↑ HIGH | TEMPERATURE | Room: CR207    | Sensor: h17         | Value:    34.0
[ALERT] ↑ HIGH | TEMPERATURE | Room: ROOM6    | Sensor: h96         | Value:    32.0
[ALERT] ↓ LOW  | HUMIDITY    | Room: CR306    | Sensor: h86         | Value:    28.1
[ALERT] ↓ LOW  | HUMIDITY    | Room: ROOM8    | Sensor: h98         | Value:    26.7
[ALERT] ↓ LOW  | LIGHT       | Room: CR306    | Sensor: h86         | Value:   119.3
[ALERT] ↑ HIGH | CO2         | Room: ROOM10   | Sensor: h100        | Value:  1369.8
[ALERT] ↑ HIGH | CO2         | Room: ROOM9    | Sensor: h99         | Value:  1144.9
[ALERT] ↑ HIGH | CO2         | Room: ROOM6    | Sensor: h96         | Value:  1133.4
[ALERT] ↑ HIGH | CO2         | Room: ROOM5    | Sensor: h95         | Value:  1221.4
[ALERT] ↓ LOW  | LIGHT       | Room: CR203    | Sensor: h13         | Value:   276.4
[ALERT] ↑ HIGH | CO2         | Room: CR203    | Sensor: h13         | Value:  1383.4
[ALERT] ↑ HIGH | CO2         | Room: ROOM4    | Sensor: h94         | Value:  1256.2
[ALERT] ↑ HIGH | TEMPERATURE | Room: LAB1     | Sensor: h21         | Value:    30.2
[ALERT] ↓ LOW  | LIGHT       | Room: LAB1     | Sensor: h21         | Value:   118.9
[ALERT] ↑ HIGH | CO2         | Room: CAFE5    | Sensor: h65         | Value:  1219.4
[ALERT] ↑ HIGH | TEMPERATURE | Room: LAB17    | Sensor: h77         | Value:    30.6
[ALERT] ↑ HIGH | TEMPERATURE | Room: CR205    | Sensor: h15         | Value:    32.9
[ALERT] ↑ HIGH | CO2         | Room: CR107    | Sensor: h7          | Value:  1485.7
[ALERT] ↑ HIGH | CO2         | Room: LAB17    | Sensor: h77         | Value:  1153.4
[ALERT] ↑ HIGH | CO2         | Room: CR205    | Sensor: h15         | Value:  1209.3
[ALERT] ↑ HIGH | CO2         | Room: LAB14    | Sensor: h74         | Value:  1151.3
[ALERT] ↑ HIGH | TEMPERATURE | Room: CAFE4    | Sensor: h64         | Value:    32.8
[ALERT] ↓ LOW  | HUMIDITY    | Room: CAFE4    | Sensor: h64         | Value:    28.7
[ALERT] ↓ LOW  | LIGHT       | Room: CAFE4    | Sensor: h64         | Value:   175.1
```

The terminal output below confirms that the anomaly detector is issuing real-time alerts for abnormal environmental values across monitored rooms.

```
mininet@mininet-vm:~/environmental_monitoring$ source venv39/bin/activate
ector.py
(venv39) mininet@mininet-vm:~/environmental_monitoring$ python server/data_collector.py
SSL module imported successfully
/home/mininet/environmental_monitoring/server/data_collector.py:51: DeprecationWarning: Callback API version 1 is deprecated, update to latest version
  client = mqtt.Client()
======================================================================
DATA COLLECTOR - Connected to MQTT Broker
======================================================================
[STORED] 2025-11-12T20:27:55Z | Room: CR207    | Sensor: h17         | TEMPERATURE | Value:    25.4
[STORED] 2025-11-12T20:27:55Z | Room: CR301    | Sensor: h81         | TEMPERATURE | Value:    19.1
[STORED] 2025-11-12T20:27:55Z | Room: ROOM10   | Sensor: h100        | TEMPERATURE | Value:    28.8
[STORED] 2025-11-12T20:27:55Z | Room: CR207    | Sensor: h17         | HUMIDITY    | Value:    39.0
[STORED] 2025-11-12T20:27:55Z | Room: CAFE3    | Sensor: h63         | TEMPERATURE | Value:    30.7
[STORED] 2025-11-12T20:27:55Z | Room: CR301    | Sensor: h81         | HUMIDITY    | Value:    60.2
[STORED] 2025-11-12T20:27:55Z | Room: ROOM10   | Sensor: h100        | HUMIDITY    | Value:    61.2
[STORED] 2025-11-12T20:27:55Z | Room: CR207    | Sensor: h17         | LIGHT       | Value:   875.4
[STORED] 2025-11-12T20:27:55Z | Room: CAFE3    | Sensor: h63         | HUMIDITY    | Value:    62.2
[STORED] 2025-11-12T20:27:55Z | Room: CR301    | Sensor: h81         | LIGHT       | Value:  1068.1
[STORED] 2025-11-12T20:27:55Z | Room: ROOM10   | Sensor: h100        | LIGHT       | Value:   462.6
[STORED] 2025-11-12T20:27:55Z | Room: CR207    | Sensor: h17         | CO2         | Value:  1118.2
[STORED] 2025-11-12T20:27:55Z | Room: CAFE3    | Sensor: h63         | LIGHT       | Value:   236.5
[STORED] 2025-11-12T20:27:55Z | Room: CR301    | Sensor: h81         | CO2         | Value:   758.8
[STORED] 2025-11-12T20:27:55Z | Room: ROOM10   | Sensor: h100        | CO2         | Value:   508.0
[STORED] 2025-11-12T20:27:55Z | Room: CAFE3    | Sensor: h63         | CO2         | Value:   513.8
[STORED] 2025-11-12T20:27:55Z | Room: CAFE8    | Sensor: h68         | TEMPERATURE | Value:    23.7
[STORED] 2025-11-12T20:27:55Z | Room: LIB2     | Sensor: h42         | TEMPERATURE | Value:    26.0
[STORED] 2025-11-12T20:27:55Z | Room: ROOM3    | Sensor: h93         | TEMPERATURE | Value:    28.1
[STORED] 2025-11-12T20:27:55Z | Room: CR307    | Sensor: h87         | TEMPERATURE | Value:    33.1
[STORED] 2025-11-12T20:27:55Z | Room: CR306    | Sensor: h86         | TEMPERATURE | Value:    29.6
[STORED] 2025-11-12T20:27:55Z | Room: ROOM9    | Sensor: h99         | TEMPERATURE | Value:    28.6
[STORED] 2025-11-12T20:27:55Z | Room: ROOM2    | Sensor: h92         | TEMPERATURE | Value:    34.8
[STORED] 2025-11-12T20:27:55Z | Room: CAFE8    | Sensor: h68         | HUMIDITY    | Value:    59.2
[STORED] 2025-11-12T20:27:55Z | Room: LIB2     | Sensor: h42         | HUMIDITY    | Value:    64.7
[STORED] 2025-11-12T20:27:55Z | Room: ROOM3    | Sensor: h93         | HUMIDITY    | Value:    68.6
[STORED] 2025-11-12T20:27:55Z | Room: CR307    | Sensor: h87         | HUMIDITY    | Value:    40.3
[STORED] 2025-11-12T20:27:55Z | Room: CR306    | Sensor: h86         | HUMIDITY    | Value:    39.6
[STORED] 2025-11-12T20:27:55Z | Room: ROOM9    | Sensor: h99         | HUMIDITY    | Value:    51.9
[STORED] 2025-11-12T20:27:55Z | Room: ROOM2    | Sensor: h92         | HUMIDITY    | Value:    25.3
[STORED] 2025-11-12T20:27:55Z | Room: CAFE8    | Sensor: h68         | LIGHT       | Value:   363.3
[STORED] 2025-11-12T20:27:55Z | Room: LIB2     | Sensor: h42         | LIGHT       | Value:   154.7
```

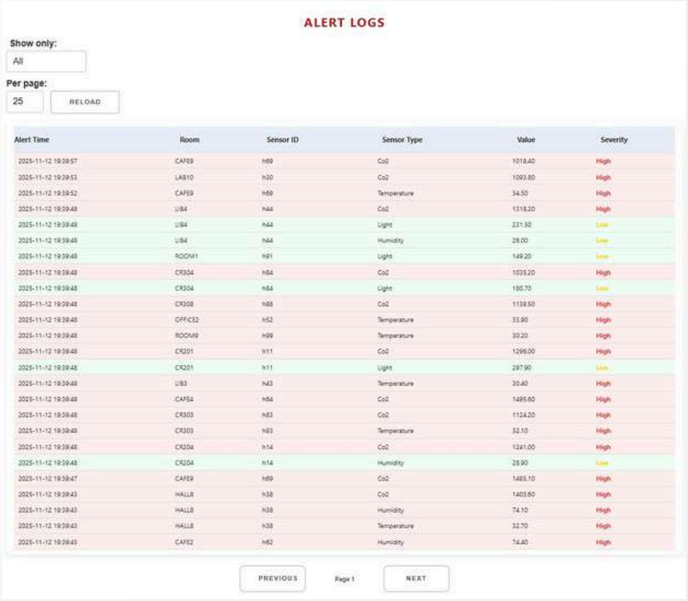
OBJECTIVES ACHIEVED

The plotly dashboard below displays real-time and historical environmental data (temperature, humidity, light, and CO_2) for a selected room, allowing users to monitor and analyze room conditions over a chosen time range.



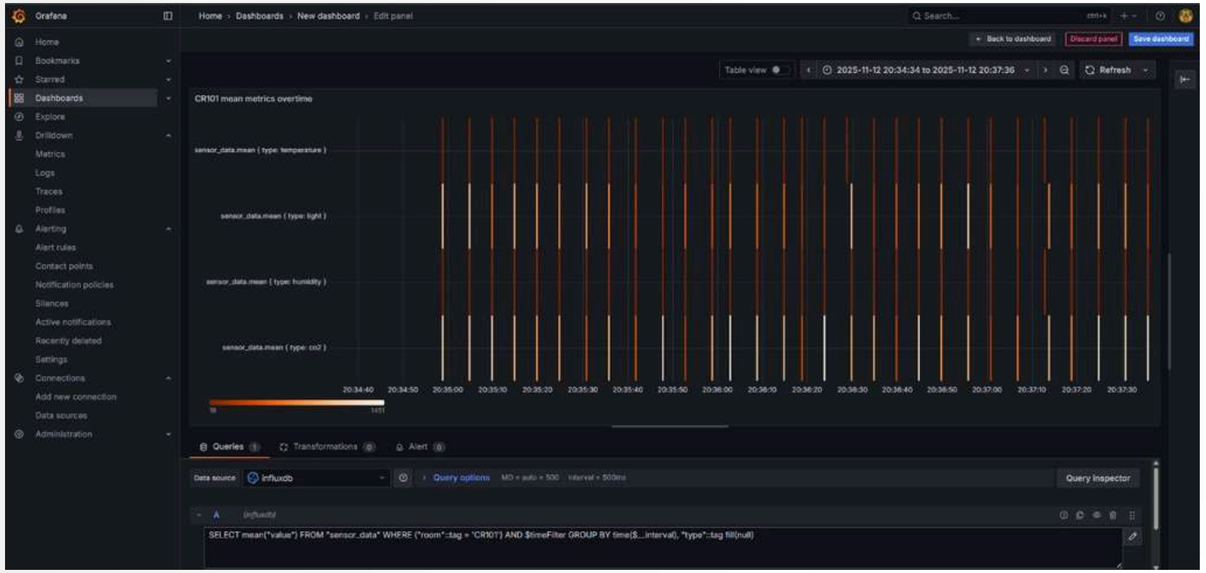
The plotly dashboard below visualizes past trends of individual environmental metrics temperature, humidity, light intensity, and CO_2 levels over time, helping to monitor and analyze variations in room conditions.

The plotly dashboard beside displays alert logs generated by sensor readings, showing the time, room, sensor type, value, and severity (high or low) to help identify abnormal environmental conditions in real-time.

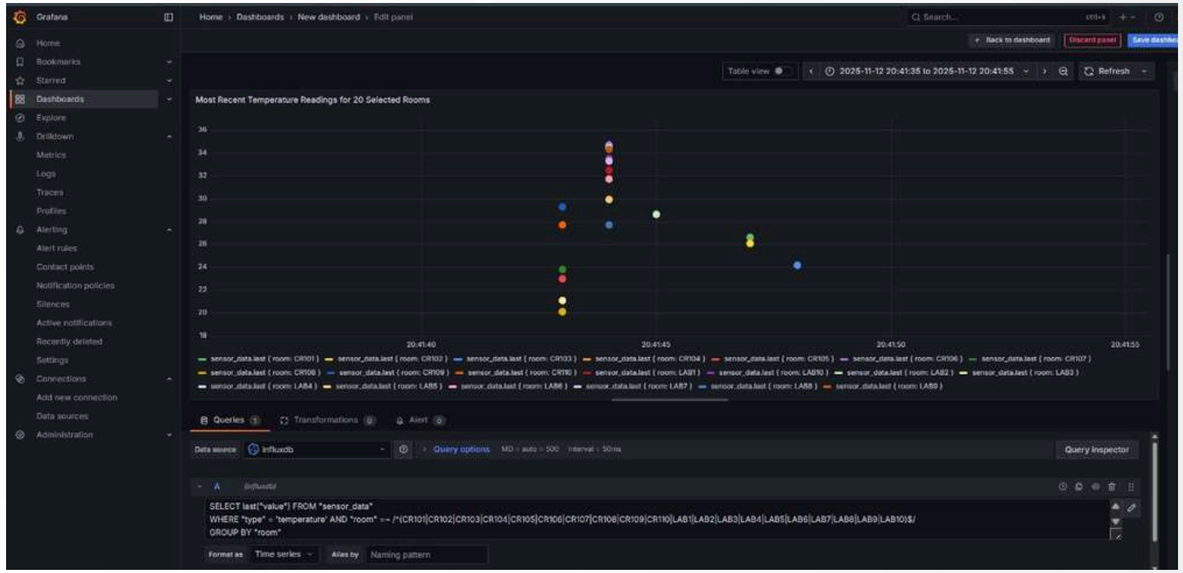


OBJECTIVES ACHIEVED

The Grafana panel below displays the most recent temperature readings from 20 selected rooms, allowing for a quick comparison of environmental conditions across multiple locations using data retrieved from InfluxDB.



The Grafana dashboard below visualizes real-time mean sensor metrics (temperature, humidity, light, and CO_2) for room CR101 over time, utilizing data queried from InfluxDB to monitor environmental conditions dynamically.



FUTURE PROSPECTS

1. **Real Sensor Deployment:** Replace simulated data with real-time IoT sensor readings (ESP32, DHT22, BH1750, MQ-135) sent via MQTT.

2. **Machine Learning Integration:** Use collected data to predict temperature trends and detect anomalies for proactive monitoring.

3. **Enhanced Security:** Utilize secure MQTT with TLS encryption and user authentication for secure data access and control.

4. **User Authentication System:** Implement role-based login (e.g., OAuth 2.0) to provide personalized and secure access to the dashboard.

REFERENCES

[1]Lantz, B., Heller, B., and McKeown, N. "A network in a laptop: Rapid prototyping for Software-Defined Networks." *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks (HotNets)*, Monterey, CA, USA, October 2010. *(The foundational paper for Mininet, which you used for network simulation.)*

[2]Banks, A., and Gupta, R. "MQTT Version 3.1.1." OASIS Standard. October 2014. [Online]. Available: http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html *(The official specification for the MQTT protocol you implemented.)*

[3]S. H. L., T. K. L., and C. F. L. "Design and Implementation of an IoT-Based Environmental Monitoring System with MQTT, InfluxDB, and Grafana," *2021 International Conference on Computer and Information Sciences (ICCOINS)*, 2021, pp. 1-6. *(A paper with an almost identical technology stack, ideal for citing as related work.)*

[4]InfluxData. "InfluxDB Technical Documentation." [Online]. Available: https://docs.influxdata.com/influxdb/ *(The official documentation for the time-series database you used.)*

[5]Grafana Labs. "Grafana Documentation." [Online]. Available: https://grafana.com/docs/grafana/latest/ *(The official documentation for the Grafana visualization platform.)*

[6]*"Dash Documentation & User Guide | Plotly," dash.plotly.com. https://dash.plotly.com*

[7]*Flask, "Welcome to Flask — Flask Documentation (1.1.x)," flask.palletsprojects.com. https://flask.palletsprojects.com/*

[8]*L. Gupta, "What is REST – Learn to create timeless REST APIs," Restfulapi.net, Jun. 13, 2019. https://restfulapi.net/*

THANK YOU