

# CIS600 Internet of Things: Application Development

## Development Steps:

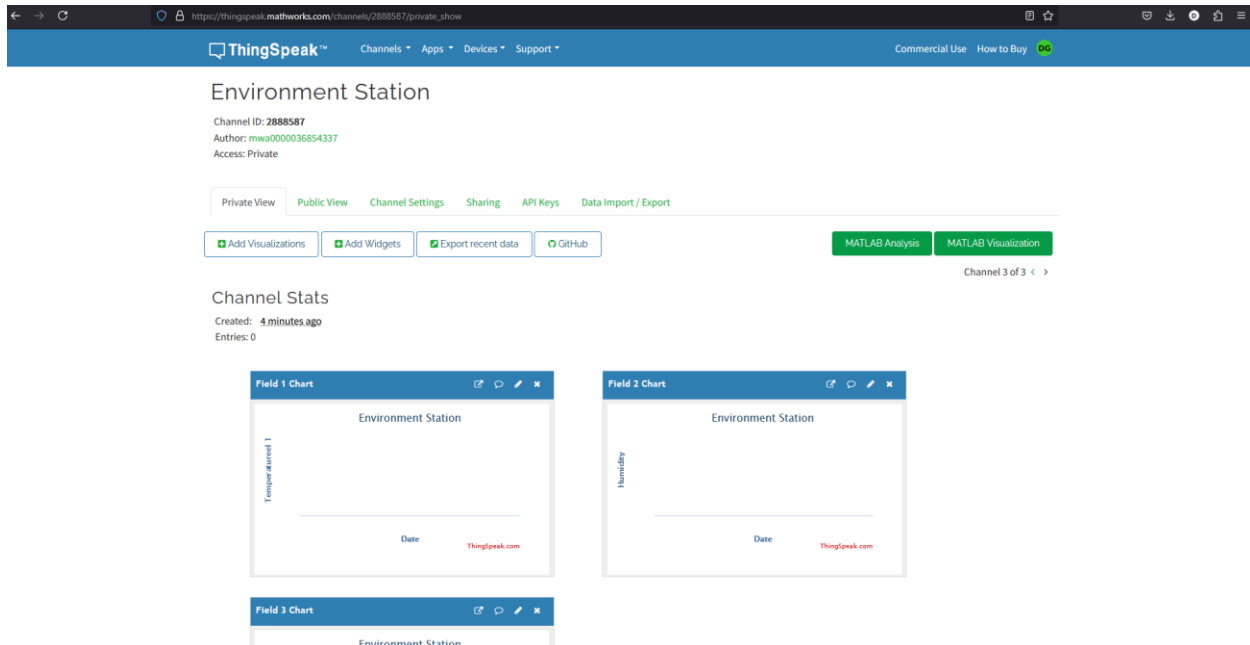
First, I made a new ThingSpeak channel with three fields: CO<sub>2</sub>, temperature, and humidity. In order to publish data securely via MQTT, I then created an MQTT device under the "MQTT Devices" section of my ThingSpeak account, which provided me with the Client ID, Username, and Password.

I generated simulated sensor values within reasonable ranges (temperature: -50 to 50°C, humidity: 0 to 100%, and CO<sub>2</sub>: 300 to 2000 ppm) in my Python script by using the random library. I then published sensor readings every 15 seconds to ThingSpeak's MQTT broker using the paho-mqtt library. The virtual station includes a unique station ID to identify itself when sending data.

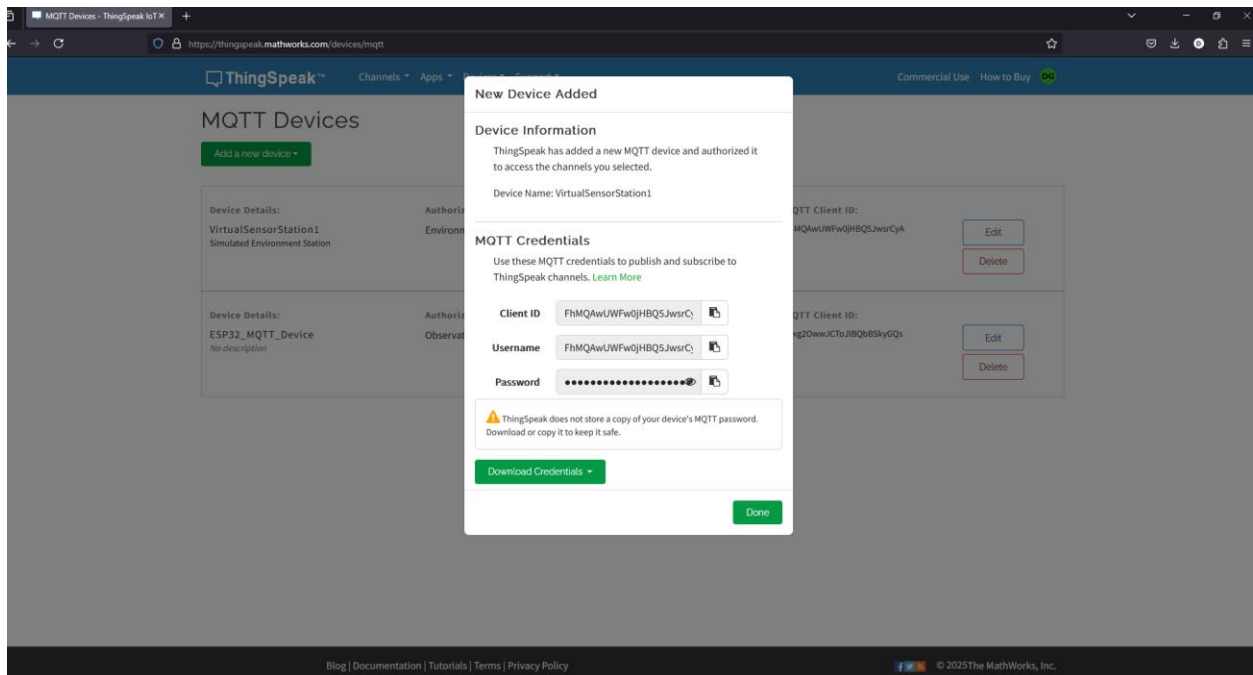
In addition, I created a secondary script to extract data from the previous five hours for a particular sensor type and locally recorded all data to a sensor\_log.txt file for offline access. By keeping an eye on live graphs on ThingSpeak, the system was tested to make sure all MQTT messages were successfully received and displayed.

## Screenshots of Output

Channel Status:

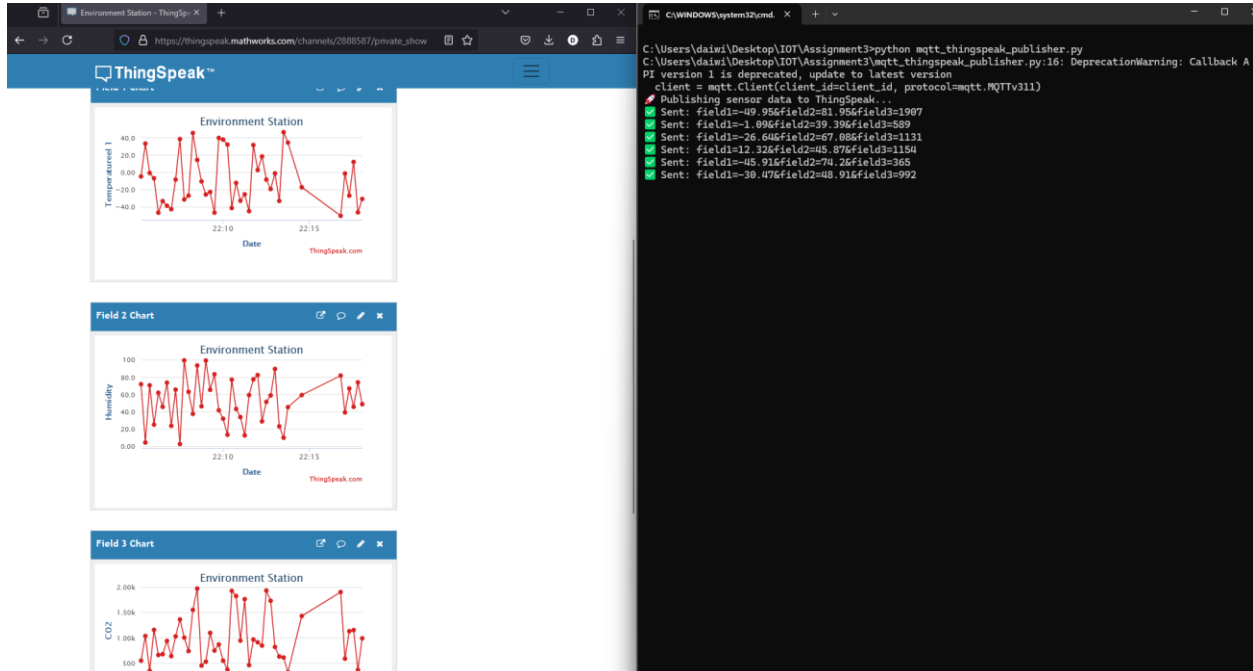


## MQTT Device:



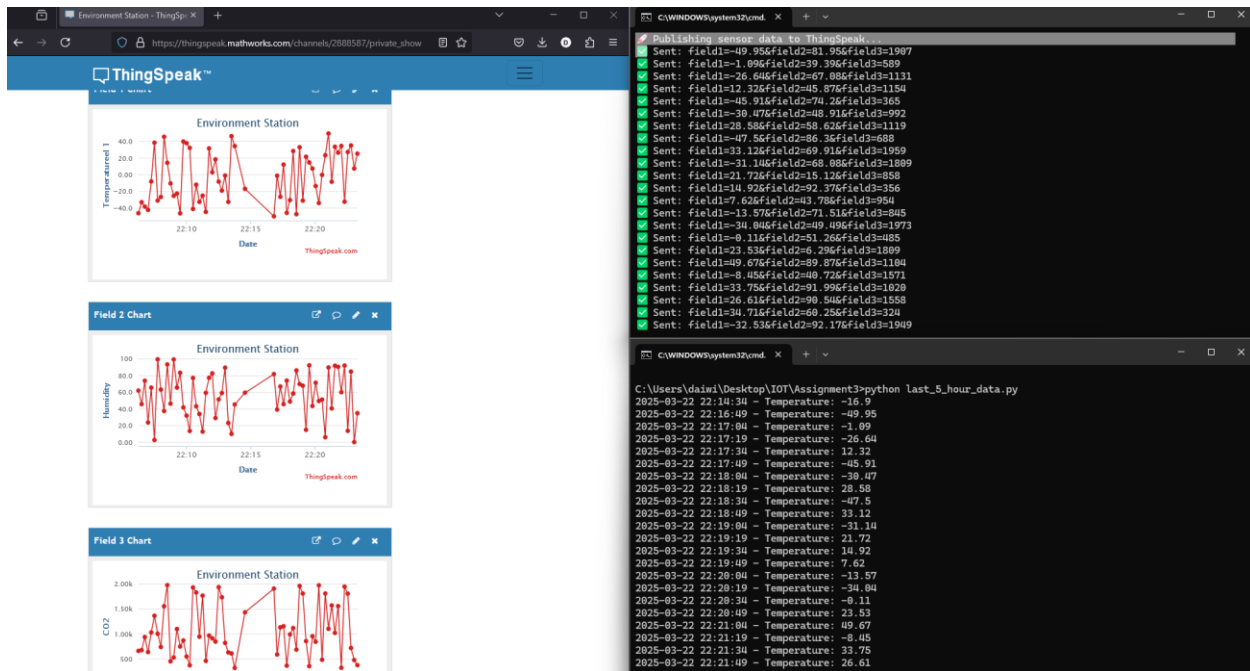
MQTT Device credentials (Client ID, Username)

## ThingSpeak dashboard graphs:

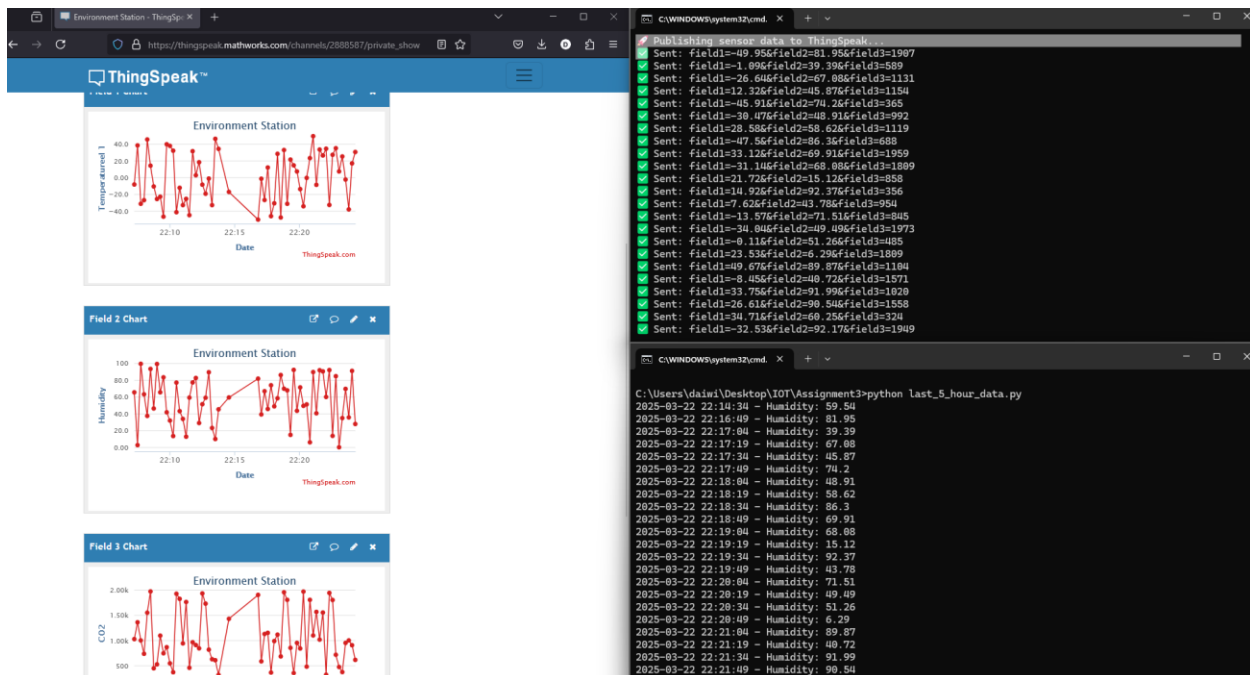


ThingSpeak Channel Status with real-time sensor updates

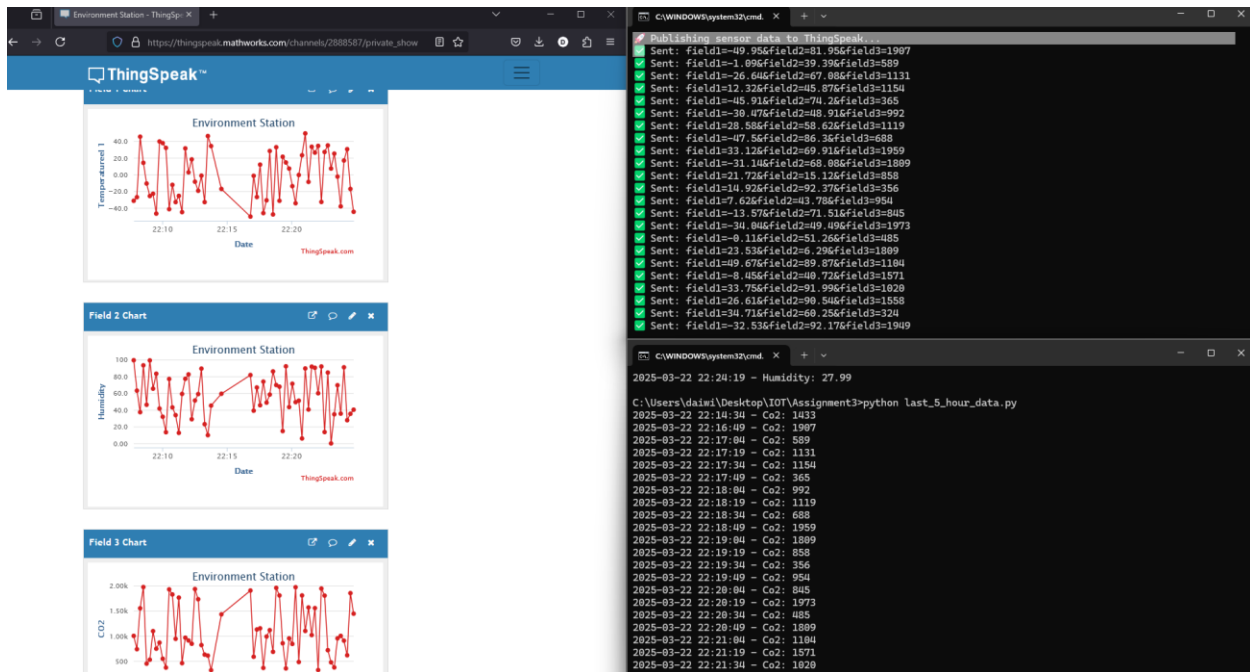
Output of last 5 hour data:



Output from last\_5\_hour\_data.py script (Temperature)



Output from last\_5\_hour\_data.py script (Humidity)



Output from last\_5\_hour\_data.py script (CO2)

## GitHub Repository

<https://github.com/Daivik-Gangappa/iot-mqtt-simulation>

## Reflection

Despite my prior experience with ThingSpeak and MQTT, this assignment allowed me to apply that knowledge in a more comprehensive and structured manner. The primary challenge was not establishing the MQTT connection itself, but rather making sure that the data was consistently formatted, mapped to the appropriate fields, and displayed in real-time on the ThingSpeak dashboard; I also had to make sure that the MQTT credentials matched precisely and that the client ID and topic structure complied with ThingSpeak's specifications.

Developing the local logging system and creating a script to retrieve the most recent five hours' worth of data from the log file were two things I had never done before. This part made me think more deeply about time-based filtering and how to process historical data effectively for analysis. Additionally, I had the opportunity to strengthen my knowledge of GitHub and Git, particularly in the areas of resolving merge conflicts and connecting my local project to an already-existing remote repository.

All things considered, this assignment strengthened my prior knowledge while providing me with an opportunity to arrange and display my work in a more polished manner, which is equally as crucial as creating functional code.