**Development Process:**

I started by creating a new ThingSpeak channel with three fields: CO₂, temperature, and humidity. To securely publish data via MQTT, I then set up an MQTT device in the "MQTT Devices" section of my ThingSpeak account, which provided the necessary Client ID, Username, and Password.

In my Python script, I used the random library to generate simulated sensor readings within realistic ranges (temperature: -50 to 50°C, humidity: 0 to 100%, CO₂: 300 to 2000 ppm). These values were sent every 15 seconds to ThingSpeak's MQTT broker using the paho-mqtt library, with a unique station ID for identification.

Additionally, I wrote a separate script to retrieve sensor data and log all readings into sensor\_log.txt for offline access. The system was validated by monitoring live graphs on ThingSpeak to ensure successful transmission and display of MQTT messages.

**Screenshots:**

MQTT Device credentials:

A screenshot of a video chat

AI-generated content may be incorrect.

Channel Details:

A screenshot of a computer

AI-generated content may be incorrect.

Graph for all the fields:

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Output Saved in a text file:

A screenshot of a computer

AI-generated content may be incorrect.

**Github Link:**

<https://github.com/vinuthBabu01/mqtt_simulation>

**Outcomes:**

While I was already familiar with ThingSpeak and MQTT, this assignment gave me the opportunity to apply that knowledge in a more structured and practical way. The real challenge wasn’t establishing the MQTT connection itself but ensuring that the data was correctly formatted, assigned to the right fields, and displayed in real-time on the ThingSpeak dashboard. I also had to carefully verify that the MQTT credentials were accurate and that the client ID and topic structure met ThingSpeak’s requirements.

An entirely new aspect for me was implementing a local logging system and developing a script to extract the last five hours of data from the log file. This required a deeper understanding of time-based data filtering and how to efficiently process historical data for analysis. Additionally, I improved my proficiency with GitHub and Git, especially in handling merge conflicts and integrating my local project with an existing remote repository.

Overall, this assignment not only reinforced my technical skills but also helped me refine how I organize and present my work, which is just as essential as writing effective code.