

## IOT Assignment 3

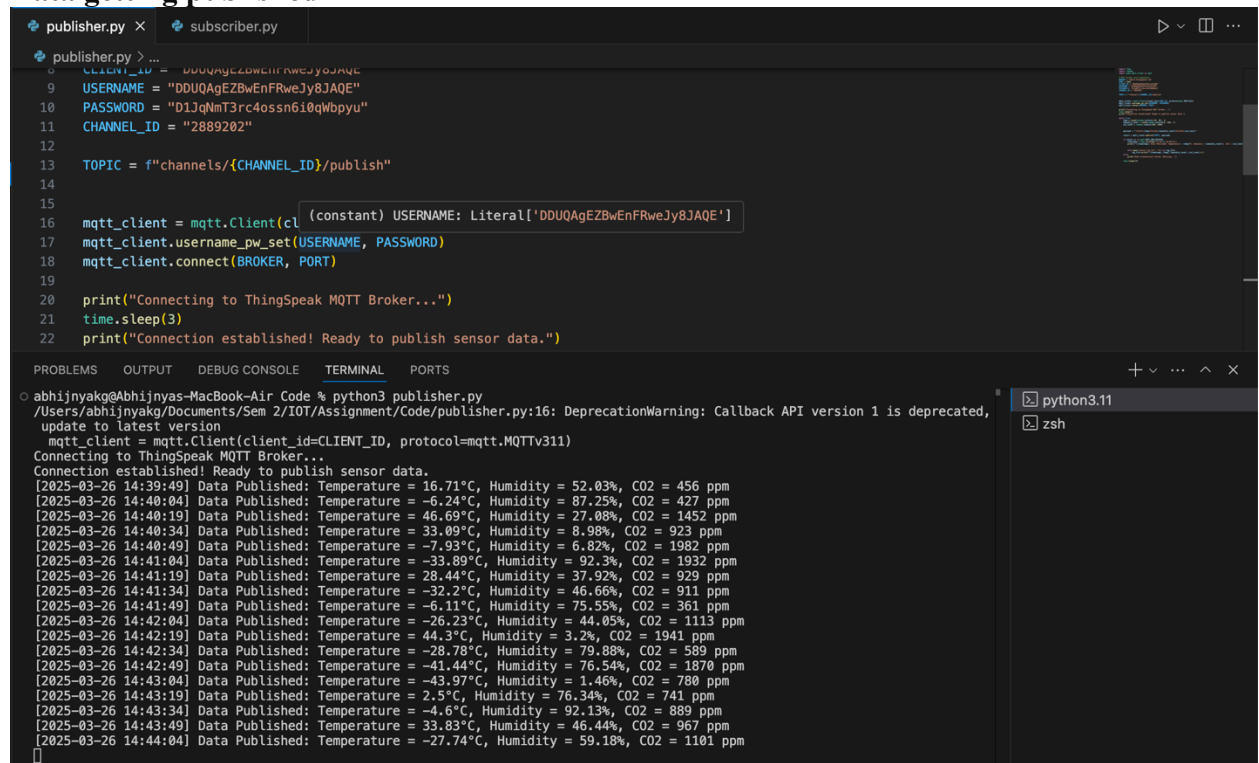
### 1. Brief explanation of the steps that you have used in developing the IOT system

In developing the IoT system, I began by simulating virtual environmental sensors to monitor temperature, humidity, and CO2 levels. I used Python to generate random values for each sensor within specified ranges: temperature (-50°C to 50°C), humidity (0% to 100%), and CO2 (300ppm to 2000ppm). To enable communication between the virtual sensors and the cloud, I implemented the MQTT protocol using the Paho MQTT client. I then connected the system to ThingSpeak, a cloud-based IoT platform, which received the sensor data every 15 seconds. I ensured that the sensor data was properly formatted and transmitted in real time to ThingSpeak by publishing the payload to the ThingSpeak MQTT channel.

Additionally, I implemented local storage by logging the sensor data into a file, allowing historical data to be accessed and analyzed offline. For querying the data, I developed a Python script that enables the user to input a sensor type (temperature, humidity, or CO2) and retrieve the latest data or filter data from the past five hours. This script reads the local log file, processes the data, and prints the relevant sensor values along with their timestamps. The cloud integration with ThingSpeak allows real-time monitoring of the sensor data from anywhere, while the efficient MQTT protocol ensures minimal bandwidth usage and fast communication. This system provides a scalable and effective solution for IoT environmental monitoring.

### 2. Screen shots of your output

#### Data getting published



```
publisher.py x subscriber.py
publisher.py > ...
1 USER_NAME = "DDUQAqEZBwEnFRweJy8JAE"
10 PASSWORD = "D1JqNmT3rc4ossn6i0qWbpyu"
11 CHANNEL_ID = "2889202"
12
13 TOPIC = f"channels/{CHANNEL_ID}/publish"
14
15
16 mqtt_client = mqtt.Client((constant) USERNAME: Literal['DDUQAqEZBwEnFRweJy8JAE'])
17 mqtt_client.username_pw_set(USERNAME, PASSWORD)
18 mqtt_client.connect(BROKER, PORT)
19
20 print("Connecting to ThingSpeak MQTT Broker...")
21 time.sleep(3)
22 print("Connection established! Ready to publish sensor data.")

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
python3 publisher.py
/Users/abhijnyakg/Documents/Sem 2/IOT/Assignment/Code/publisher.py:16: DeprecationWarning: Callback API version 1 is deprecated,
update to latest version
mqtt_client = mqtt.Client(client_id=CLIENT_ID, protocol=mqtt.MQTTv311)
Connecting to ThingSpeak MQTT Broker...
Connection established! Ready to publish sensor data.
[2025-03-26 14:39:49] Data Published: Temperature = 16.71°C, Humidity = 52.03%, CO2 = 456 ppm
[2025-03-26 14:40:04] Data Published: Temperature = -6.24°C, Humidity = 87.25%, CO2 = 427 ppm
[2025-03-26 14:40:19] Data Published: Temperature = 46.69°C, Humidity = 27.08%, CO2 = 1452 ppm
[2025-03-26 14:40:34] Data Published: Temperature = 33.09°C, Humidity = 8.98%, CO2 = 923 ppm
[2025-03-26 14:40:49] Data Published: Temperature = -7.93°C, Humidity = 6.82%, CO2 = 1982 ppm
[2025-03-26 14:41:04] Data Published: Temperature = -33.89°C, Humidity = 92.3%, CO2 = 1932 ppm
[2025-03-26 14:41:19] Data Published: Temperature = 28.44°C, Humidity = 37.92%, CO2 = 929 ppm
[2025-03-26 14:41:34] Data Published: Temperature = -32.2°C, Humidity = 46.66%, CO2 = 911 ppm
[2025-03-26 14:41:49] Data Published: Temperature = -6.11°C, Humidity = 75.55%, CO2 = 361 ppm
[2025-03-26 14:42:04] Data Published: Temperature = -26.23°C, Humidity = 44.05%, CO2 = 1113 ppm
[2025-03-26 14:42:19] Data Published: Temperature = 44.3°C, Humidity = 3.2%, CO2 = 1941 ppm
[2025-03-26 14:42:34] Data Published: Temperature = -28.78°C, Humidity = 79.88%, CO2 = 589 ppm
[2025-03-26 14:42:49] Data Published: Temperature = -41.44°C, Humidity = 76.54%, CO2 = 1870 ppm
[2025-03-26 14:43:04] Data Published: Temperature = -43.97°C, Humidity = 1.46%, CO2 = 780 ppm
[2025-03-26 14:43:19] Data Published: Temperature = 2.5°C, Humidity = 76.34%, CO2 = 741 ppm
[2025-03-26 14:43:34] Data Published: Temperature = -4.6°C, Humidity = 92.13%, CO2 = 889 ppm
[2025-03-26 14:43:49] Data Published: Temperature = 33.83°C, Humidity = 46.44%, CO2 = 967 ppm
[2025-03-26 14:44:04] Data Published: Temperature = -27.74°C, Humidity = 59.18%, CO2 = 1101 ppm
```

## Output of last 5 hour data:

### Temperature

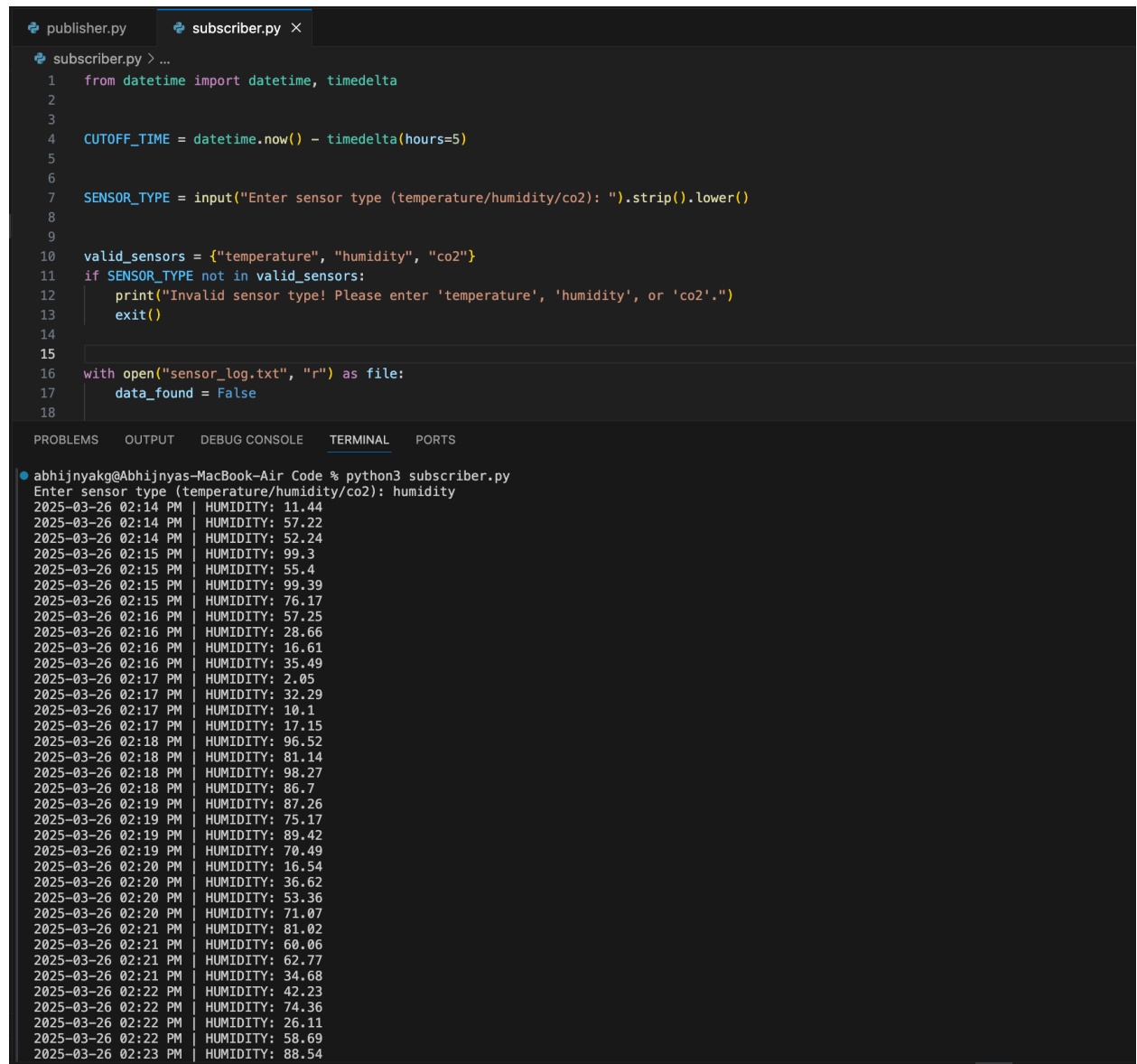
```
publisher.py subscriber.py X
subscriber.py > ...
1 from datetime import datetime, timedelta
2
3
4 CUTOFF_TIME = datetime.now() - timedelta(hours=5)
5
6
7 SENSOR_TYPE = input("Enter sensor type (temperature/humidity/co2): ").strip().lower()
8
9
10 valid_sensors = {"temperature", "humidity", "co2"}
11 if SENSOR_TYPE not in valid_sensors:
12     print("Invalid sensor type! Please enter 'temperature', 'humidity', or 'co2'.")
13     exit()
14
15
16 with open("sensor_log.txt", "r") as file:
17     data_found = False
18
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
abhiinyakg@Abhiinyas-MacBook-Air Code % python3 subscriber.py
Enter sensor type (temperature/humidity/co2): temperature
2025-03-26 02:14 PM | TEMPERATURE: -31.82
2025-03-26 02:14 PM | TEMPERATURE: -27.85
2025-03-26 02:14 PM | TEMPERATURE: 17.34
2025-03-26 02:15 PM | TEMPERATURE: 28.94
2025-03-26 02:15 PM | TEMPERATURE: -46.61
2025-03-26 02:15 PM | TEMPERATURE: -31.7
2025-03-26 02:15 PM | TEMPERATURE: 0.89
2025-03-26 02:16 PM | TEMPERATURE: -7.58
2025-03-26 02:16 PM | TEMPERATURE: 49.71
2025-03-26 02:16 PM | TEMPERATURE: 5.31
2025-03-26 02:16 PM | TEMPERATURE: 17.23
2025-03-26 02:17 PM | TEMPERATURE: -24.94
2025-03-26 02:17 PM | TEMPERATURE: -31.19
2025-03-26 02:17 PM | TEMPERATURE: 33.45
2025-03-26 02:17 PM | TEMPERATURE: -19.75
2025-03-26 02:18 PM | TEMPERATURE: -20.02
2025-03-26 02:18 PM | TEMPERATURE: 35.15
2025-03-26 02:18 PM | TEMPERATURE: 34.22
2025-03-26 02:18 PM | TEMPERATURE: -33.02
2025-03-26 02:19 PM | TEMPERATURE: -48.44
2025-03-26 02:19 PM | TEMPERATURE: 43.13
2025-03-26 02:19 PM | TEMPERATURE: 45.7
2025-03-26 02:19 PM | TEMPERATURE: 34.53
2025-03-26 02:20 PM | TEMPERATURE: 19.53
2025-03-26 02:20 PM | TEMPERATURE: -2.51
2025-03-26 02:20 PM | TEMPERATURE: -43.84
2025-03-26 02:20 PM | TEMPERATURE: -22.28
2025-03-26 02:21 PM | TEMPERATURE: -17.39
2025-03-26 02:21 PM | TEMPERATURE: -19.69
2025-03-26 02:21 PM | TEMPERATURE: 19.55
2025-03-26 02:21 PM | TEMPERATURE: 29.73
2025-03-26 02:22 PM | TEMPERATURE: -41.88
2025-03-26 02:22 PM | TEMPERATURE: -41.77
2025-03-26 02:22 PM | TEMPERATURE: -16.01
2025-03-26 02:22 PM | TEMPERATURE: 21.49
2025-03-26 02:23 PM | TEMPERATURE: 23.96
```

## Output of last 5 hour data:

### Humidity



The image shows a code editor with two tabs: 'publisher.py' and 'subscriber.py'. The 'subscriber.py' tab is active, displaying a Python script. The script imports 'datetime' and 'timedelta', sets a 'CUTOFF\_TIME' to 5 hours ago, prompts the user for a sensor type, and logs humidity data to a file named 'sensor\_log.txt'. The terminal output shows the user entering 'humidity' and a series of 25 log entries, each containing a timestamp and a humidity value.

```
subscriber.py > ...
1  from datetime import datetime, timedelta
2
3
4  CUTOFF_TIME = datetime.now() - timedelta(hours=5)
5
6
7  SENSOR_TYPE = input("Enter sensor type (temperature/humidity/co2): ").strip().lower()
8
9
10 valid_sensors = {"temperature", "humidity", "co2"}
11 if SENSOR_TYPE not in valid_sensors:
12     print("Invalid sensor type! Please enter 'temperature', 'humidity', or 'co2'.")
13     exit()
14
15
16 with open("sensor_log.txt", "r") as file:
17     data_found = False
18
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

```
abhiornyak@Abhiynyas-MacBook-Air Code % python3 subscriber.py
Enter sensor type (temperature/humidity/co2): humidity
2025-03-26 02:14 PM | HUMIDITY: 11.44
2025-03-26 02:14 PM | HUMIDITY: 57.22
2025-03-26 02:14 PM | HUMIDITY: 52.24
2025-03-26 02:15 PM | HUMIDITY: 99.3
2025-03-26 02:15 PM | HUMIDITY: 55.4
2025-03-26 02:15 PM | HUMIDITY: 99.39
2025-03-26 02:15 PM | HUMIDITY: 76.17
2025-03-26 02:16 PM | HUMIDITY: 57.25
2025-03-26 02:16 PM | HUMIDITY: 28.66
2025-03-26 02:16 PM | HUMIDITY: 16.61
2025-03-26 02:16 PM | HUMIDITY: 35.49
2025-03-26 02:17 PM | HUMIDITY: 2.05
2025-03-26 02:17 PM | HUMIDITY: 32.29
2025-03-26 02:17 PM | HUMIDITY: 10.1
2025-03-26 02:17 PM | HUMIDITY: 17.15
2025-03-26 02:18 PM | HUMIDITY: 96.52
2025-03-26 02:18 PM | HUMIDITY: 81.14
2025-03-26 02:18 PM | HUMIDITY: 98.27
2025-03-26 02:18 PM | HUMIDITY: 86.7
2025-03-26 02:19 PM | HUMIDITY: 87.26
2025-03-26 02:19 PM | HUMIDITY: 75.17
2025-03-26 02:19 PM | HUMIDITY: 89.42
2025-03-26 02:19 PM | HUMIDITY: 70.49
2025-03-26 02:20 PM | HUMIDITY: 16.54
2025-03-26 02:20 PM | HUMIDITY: 36.62
2025-03-26 02:20 PM | HUMIDITY: 53.36
2025-03-26 02:20 PM | HUMIDITY: 71.07
2025-03-26 02:21 PM | HUMIDITY: 81.02
2025-03-26 02:21 PM | HUMIDITY: 60.06
2025-03-26 02:21 PM | HUMIDITY: 62.77
2025-03-26 02:21 PM | HUMIDITY: 34.68
2025-03-26 02:22 PM | HUMIDITY: 42.23
2025-03-26 02:22 PM | HUMIDITY: 74.36
2025-03-26 02:22 PM | HUMIDITY: 26.11
2025-03-26 02:22 PM | HUMIDITY: 58.69
2025-03-26 02:23 PM | HUMIDITY: 88.54
```

## Output of last 5 hour data:

### Co2

```
publisher.py subscriber.py X
subscriber.py > ...
1 from datetime import datetime, timedelta
2
3
4 CUTOFF_TIME = datetime.now() - timedelta(hours=5)
5
6
7 SENSOR_TYPE = input("Enter sensor type (temperature/humidity/co2): ").strip().lower()
8
9
10 valid_sensors = {"temperature", "humidity", "co2"}
11 if SENSOR_TYPE not in valid_sensors:
12     print("Invalid sensor type! Please enter 'temperature', 'humidity', or 'co2'.")
13     exit()
14
15
16 with open("sensor_log.txt", "r") as file:
17     data_found = False
18
19
20 PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
● abhijnyakg@Abhijnyas-MacBook-Air Code % python3 subscriber.py
Enter sensor type (temperature/humidity/co2): co2
2025-03-26 02:14 PM | C02: 1482
2025-03-26 02:14 PM | C02: 948
2025-03-26 02:14 PM | C02: 810
2025-03-26 02:15 PM | C02: 411
2025-03-26 02:15 PM | C02: 1067
2025-03-26 02:15 PM | C02: 304
2025-03-26 02:15 PM | C02: 657
2025-03-26 02:16 PM | C02: 910
2025-03-26 02:16 PM | C02: 302
2025-03-26 02:16 PM | C02: 1654
2025-03-26 02:16 PM | C02: 1979
2025-03-26 02:17 PM | C02: 1220
2025-03-26 02:17 PM | C02: 1581
2025-03-26 02:17 PM | C02: 1993
2025-03-26 02:17 PM | C02: 699
2025-03-26 02:18 PM | C02: 1760
2025-03-26 02:18 PM | C02: 803
2025-03-26 02:18 PM | C02: 1104
2025-03-26 02:18 PM | C02: 1190
2025-03-26 02:19 PM | C02: 1169
2025-03-26 02:19 PM | C02: 1135
2025-03-26 02:19 PM | C02: 851
2025-03-26 02:19 PM | C02: 1717
2025-03-26 02:20 PM | C02: 855
2025-03-26 02:20 PM | C02: 632
2025-03-26 02:20 PM | C02: 321
2025-03-26 02:20 PM | C02: 798
2025-03-26 02:21 PM | C02: 579
2025-03-26 02:21 PM | C02: 462
2025-03-26 02:21 PM | C02: 1566
2025-03-26 02:21 PM | C02: 1731
2025-03-26 02:22 PM | C02: 942
2025-03-26 02:22 PM | C02: 471
2025-03-26 02:22 PM | C02: 905
2025-03-26 02:22 PM | C02: 957
2025-03-26 02:23 PM | C02: 1466
```

# ThingSpeak Dashboard

thingspeak.mathworks.com/channels/2889202/private\_show



ThingSpeak™

Channels ▾ Apps ▾ Devices ▾ Support ▾

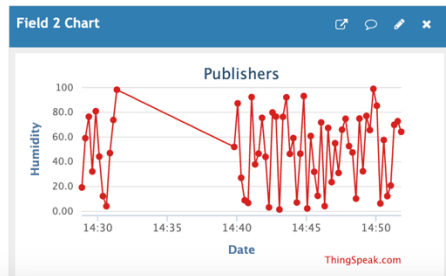
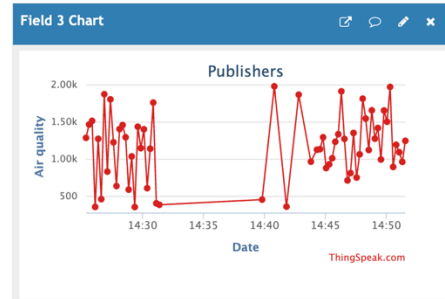
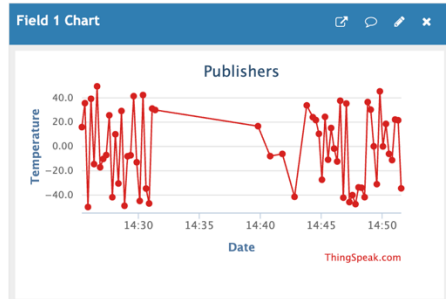
Commercial Use How to Buy

## Channel Stats

Created: 2 days ago

Last entry: less than a minute ago

Entries: 117



## MQTT Device

ThingSpeak™

Channels ▾ Apps ▾ Devices ▾ Support ▾

### Edit MQTT2

#### Device Information

Name

MQTT2

Description

Enter optional information about this device for later reference.

#### MQTT Credentials

Use these MQTT credentials to publish and subscribe to ThingSpeak channels. [Learn More](#)

Client ID

DDUQAgEZBwEnFRweJy8JAQE

Username

DDUQAgEZBwEnFRweJy8JAQE

Password

\*\*\*\*\*

#### Authorize channels to access ⓘ

-- Select a Channel --

...

Add Channel

Authorized Channel ⓘ	Allow Publish	Allow Subscribe	
Publishers (2889202)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	×

Save

Cancel

**3. Include the URL of a GitHub repository where you will push all your code and scripts that are needed to realize the assignment, along with a main README.md file.**

<https://github.com/Abhijnya002/IOT-project>

**4. Write a reflection on a specific experience that you have had when completing this assignment (incorporate your personal thoughts and opinions).**

Completing this IoT project was challenging but rewarding. I think that this assignment allowed me to exercise a wide range of skills that I've been learning, from setting up MQTT communication to creating virtual sensors for a real-world scenario. One of the most fun sections was playing around with Python to create both the publisher and subscriber scripts.

Being able to simulate environmental data with random values and then publishing them to a cloud service like ThingSpeak was a great experience, as I felt like I was creating an actual system from the ground up.

Another issue I faced was figuring out the correct setup of the MQTT client and ThingSpeak integration. I first had issues with credential configuration and channel configuration on ThingSpeak, but through troubleshooting and documentation review, I managed to fix the issues. From this activity, I gained the lesson of patience and persistence when debugging IoT systems since sometimes very small configurations have a significant impact.

I enjoyed the real-time nature of the system, in the way that I was able to witness the sensor data being posted and plotted on ThingSpeak. It was rewarding to watch my code and the IoT idea materializing. Additionally, creating the query function to retrieve the data for the last five hours was a challenging but enjoyable exercise in working with timestamps and filtering data and made me a more competent data handler. In total, I think this project taught me more about the operation of IoT systems, data collection, transmission, and storage, and MQTT application in industry for device-to-device communication. It was a fantastic learning experience, and I look forward to working on more advanced IoT applications in the future.