

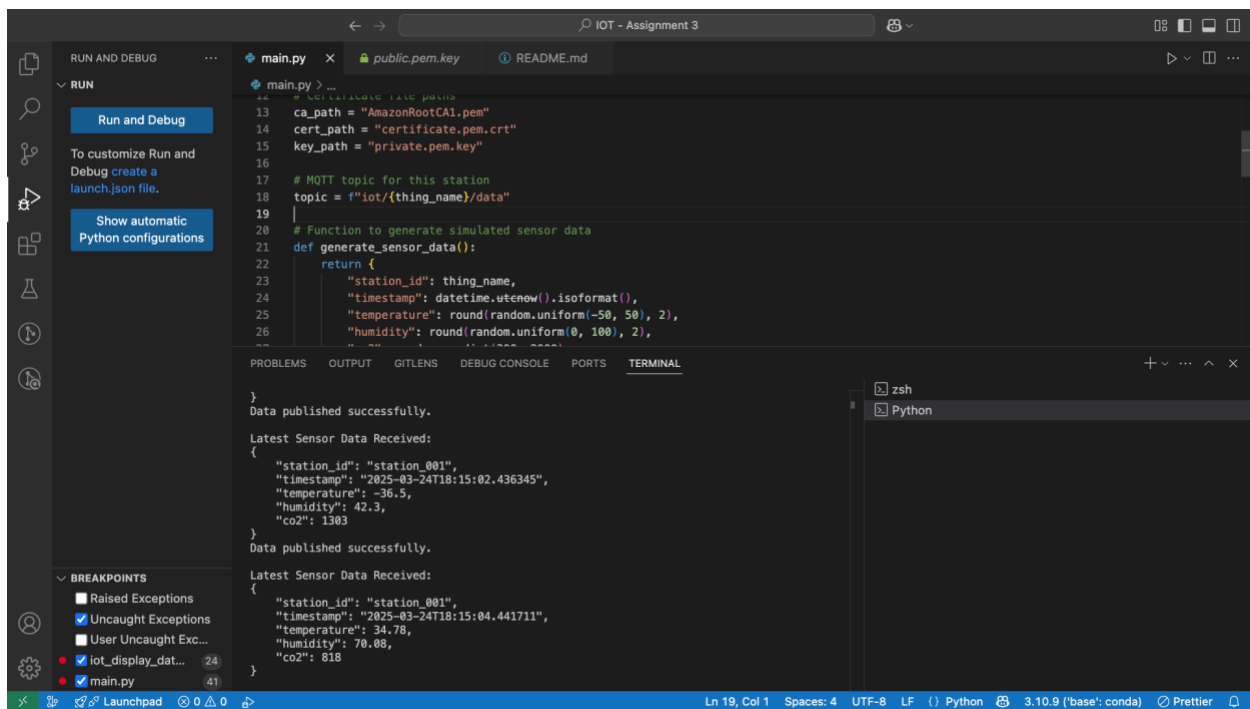
CIS600 – IoT Assignment 3 Report

1. [0.5 point] Brief explanation of the steps that you have used in developing the IOT system

I developed a Python-based IoT system that simulates an environmental station using virtual sensors for temperature, humidity, and CO2. The system connects securely to AWS IoT Core using MQTT and publishes sensor data every 2 seconds. It also subscribes to the same topic to receive data. All received messages are stored in a local JSON file, and users can view sensor readings from the last 5 hours by filtering data based on timestamp and sensor type.

2. [1 point] Screenshots of the output

I've included the following screenshots as part of this submission:



The screenshot displays a VS Code editor window titled "IoT - Assignment 3". The editor shows a Python file named `main.py` with the following code:

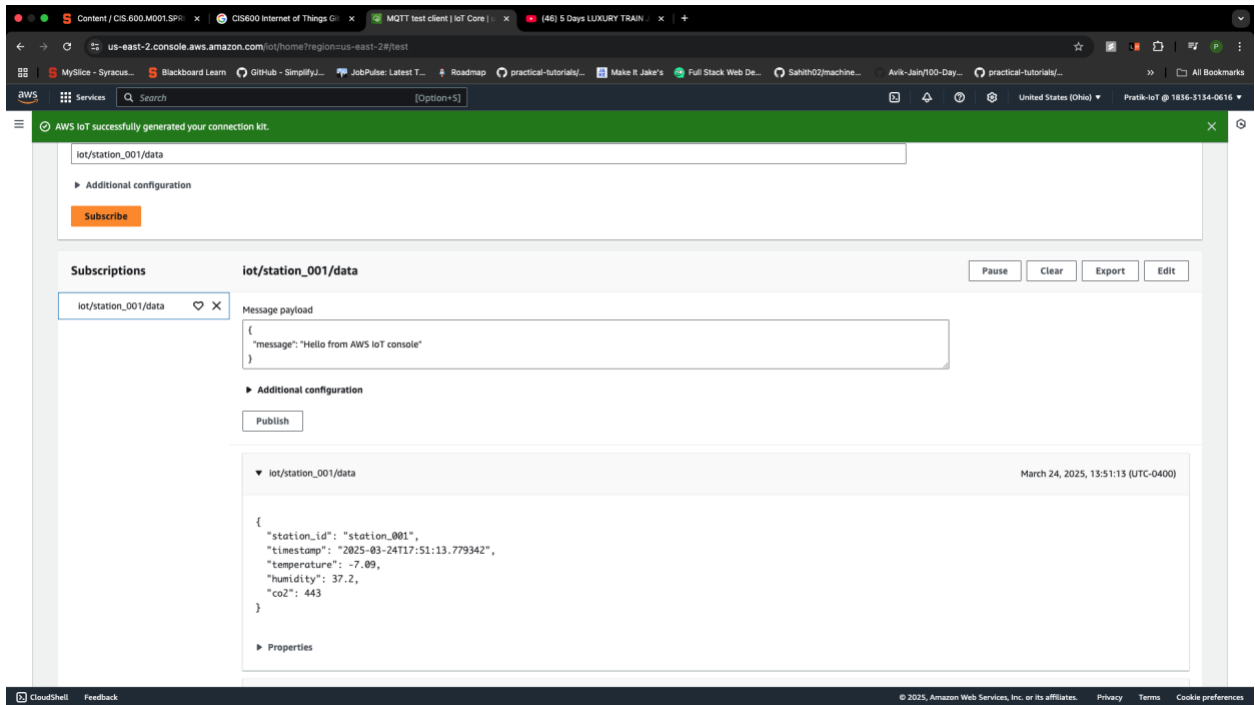
```
13 ca_path = "AmazonRootCA1.pem"
14 cert_path = "certificate.pem.crt"
15 key_path = "private.pem.key"
16
17 # MQTT topic for this station
18 topic = f"iot/{thing_name}/data"
19
20 # Function to generate simulated sensor data
21 def generate_sensor_data():
22     return {
23         "station_id": thing_name,
24         "timestamp": datetime.utcnow().isoformat(),
25         "temperature": round(random.uniform(-50, 50), 2),
26         "humidity": round(random.uniform(0, 100), 2),
27         "co2": round(random.uniform(100, 1000), 2)
28     }
```

The terminal output shows the following messages:

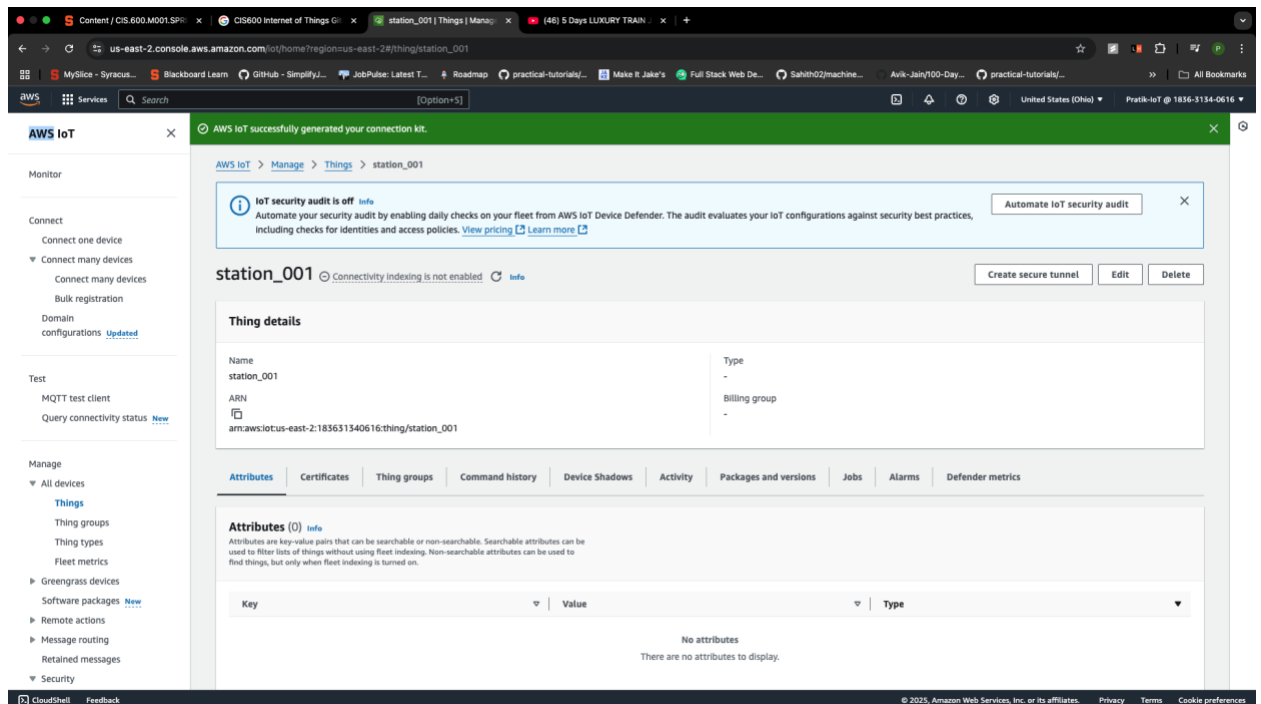
```
Data published successfully.
Latest Sensor Data Received:
{
  "station_id": "station_001",
  "timestamp": "2025-03-24T18:15:02.436345",
  "temperature": -36.5,
  "humidity": 42.3,
  "co2": 1303
}
Data published successfully.
Latest Sensor Data Received:
{
  "station_id": "station_001",
  "timestamp": "2025-03-24T18:15:04.441711",
  "temperature": 34.78,
  "humidity": 70.08,
  "co2": 818
}
```

- The Python script running and generating sensor data

CIS600 – IoT Assignment 3 Report

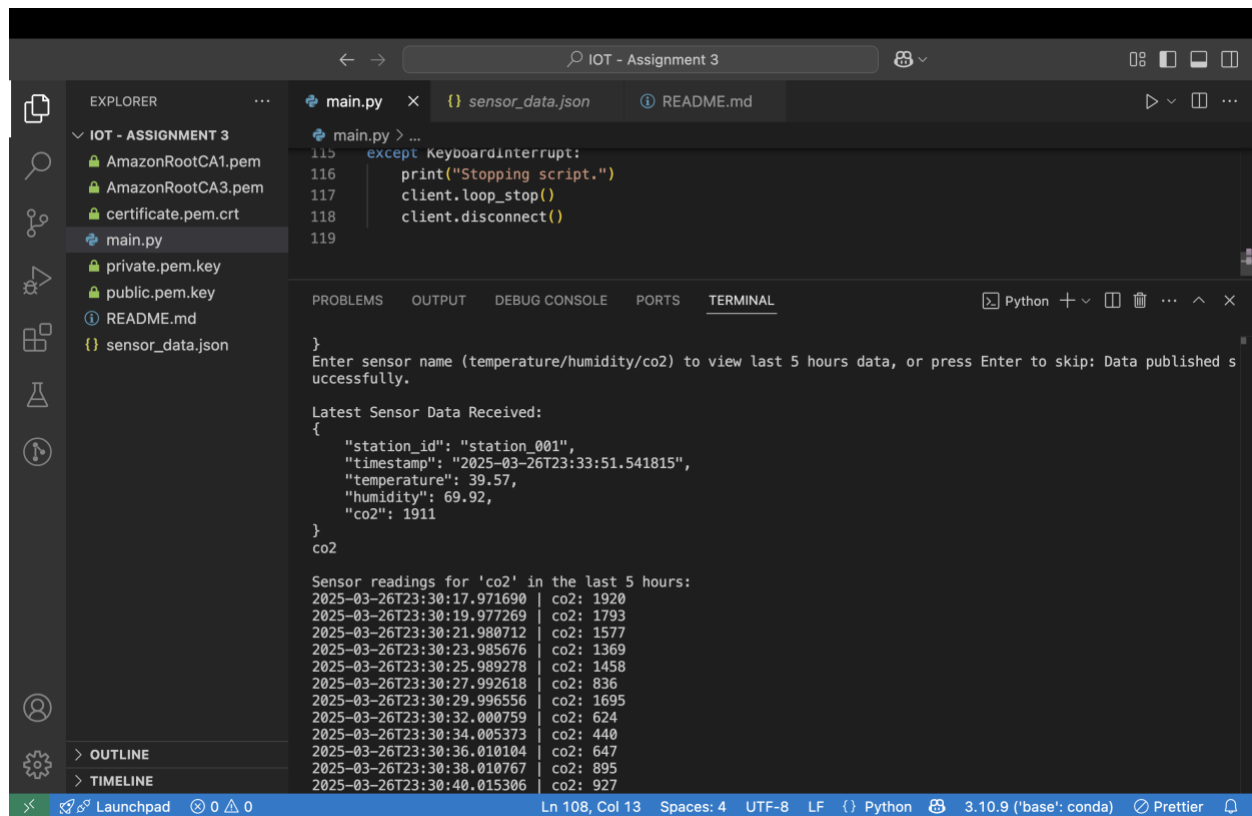


- The MQTT Test Client in AWS IoT Core showing the incoming messages



- AWS IoT console showing the configured Thing and certificates

CIS600 – IoT Assignment 3 Report



The screenshot shows a VS Code editor window titled "IOT - Assignment 3". The Explorer sidebar on the left shows a project structure for "IOT - ASSIGNMENT 3" with files: AmazonRootCA1.pem, AmazonRootCA3.pem, certificate.pem.crt, main.py, private.pem.key, public.pem.key, README.md, and sensor_data.json. The main editor displays the code in main.py, which includes a KeyboardInterrupt exception handler. The TERMINAL panel at the bottom shows the following output:

```
}
Enter sensor name (temperature/humidity/co2) to view last 5 hours data, or press Enter to skip: Data published s
uccessfully.

Latest Sensor Data Received:
{
  "station_id": "station_001",
  "timestamp": "2025-03-26T23:33:51.541815",
  "temperature": 39.57,
  "humidity": 69.92,
  "co2": 1911
}
co2

Sensor readings for 'co2' in the last 5 hours:
2025-03-26T23:30:17.971690 | co2: 1920
2025-03-26T23:30:19.977269 | co2: 1793
2025-03-26T23:30:21.980712 | co2: 1577
2025-03-26T23:30:23.985676 | co2: 1369
2025-03-26T23:30:25.989278 | co2: 1458
2025-03-26T23:30:27.992618 | co2: 836
2025-03-26T23:30:29.996556 | co2: 1695
2025-03-26T23:30:32.000759 | co2: 624
2025-03-26T23:30:34.005373 | co2: 440
2025-03-26T23:30:36.010104 | co2: 647
2025-03-26T23:30:38.010767 | co2: 895
2025-03-26T23:30:40.015306 | co2: 927
```

- Data for last 5 hours

3. [1 point] GitHub repository

All the code and necessary setup instructions are available on my GitHub repository:
<https://github.com/pratiksatpute7/IOT---Assignment-3>

4. [1 point] Reflection

Working on this assignment helped me understand how cloud-based IoT systems function in practice. Setting up secure communication with AWS IoT Core was a bit challenging at first, especially configuring certificates and policies correctly. However, once I got the MQTT connection working, it was exciting to see real-time data flowing between the virtual sensor and the cloud. Implementing local data storage and filtering also gave me insights into managing time-series data. Overall, this project strengthened my skills in MQTT communication, Python scripting, and cloud integration. It was a rewarding experience to see everything working together smoothly.