

## Assignment 2: Virtual IoT Fleet & Digital Twin Dashboard

Name: Ibrahim

Use Case: Industrial Boiler

Simulation Tool: Python (paho-mqtt)

Visualization Tool: Streamlit

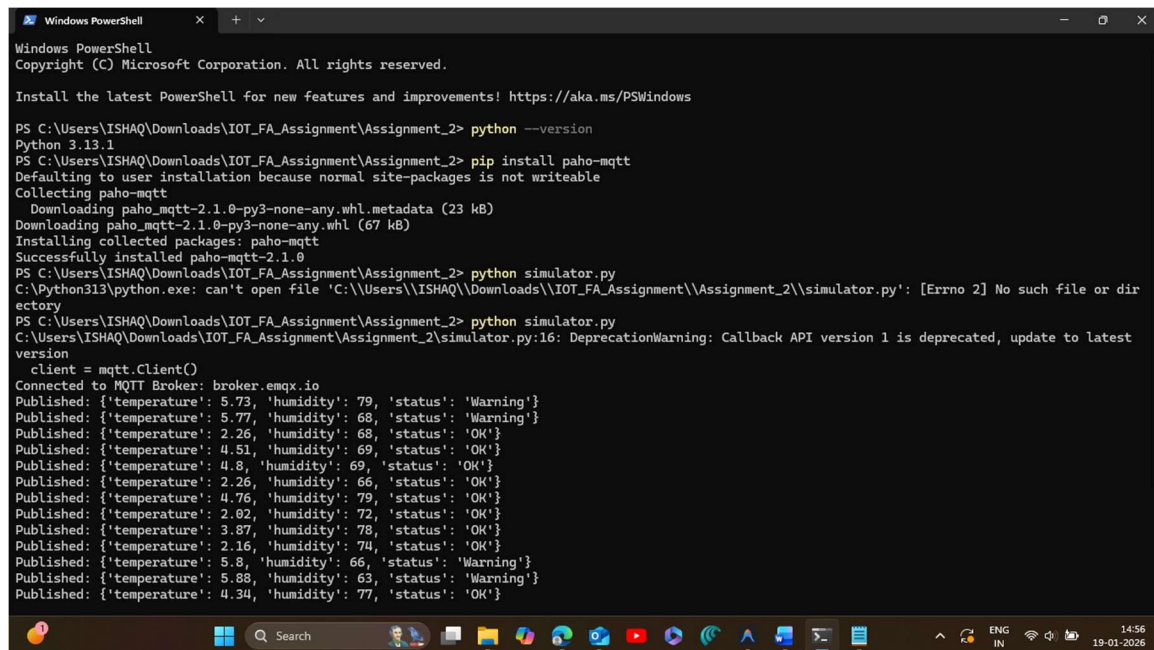
MQTT Broker: broker.emqx.io

### 1. Objective

The objective of this assignment is to simulate an IoT device using software, transmit sensor data through a cloud broker, and visualize the data using a digital twin dashboard. This approach mimics real-world industrial IoT systems that are tested before physical deployment.

### 2. Simulation Strategy

A Python-based IoT simulator was used to generate synthetic sensor data. The simulator publishes telemetry data such as temperature, humidity, and system status in JSON format to a public MQTT broker. This method allows testing of IoT logic without using physical hardware.



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\ISHAQ\Downloads\IOT_FA_Assignment\Assignment_2> python --version
Python 3.13.1
PS C:\Users\ISHAQ\Downloads\IOT_FA_Assignment\Assignment_2> pip install paho-mqtt
Defaulting to user installation because normal site-packages is not writeable
Collecting paho-mqtt
  Downloading paho_mqtt-2.1.0-py3-none-any.whl.metadata (23 kB)
  Downloading paho_mqtt-2.1.0-py3-none-any.whl (67 kB)
Installing collected packages: paho-mqtt
Successfully installed paho-mqtt-2.1.0
PS C:\Users\ISHAQ\Downloads\IOT_FA_Assignment\Assignment_2> python simulator.py
C:\Python313\python.exe: can't open file 'C:\Users\ISHAQ\Downloads\IOT_FA_Assignment\Assignment_2\simulator.py': [Errno 2] No such file or directory
PS C:\Users\ISHAQ\Downloads\IOT_FA_Assignment\Assignment_2> python simulator.py
C:\Users\ISHAQ\Downloads\IOT_FA_Assignment\Assignment_2\simulator.py:16: DeprecationWarning: Callback API version 1 is deprecated, update to latest version
  client = mqtt.Client()
Connected to MQTT Broker: broker.emqx.io
Published: {'temperature': 5.73, 'humidity': 79, 'status': 'Warning'}
Published: {'temperature': 5.77, 'humidity': 68, 'status': 'Warning'}
Published: {'temperature': 2.26, 'humidity': 68, 'status': 'OK'}
Published: {'temperature': 4.51, 'humidity': 69, 'status': 'OK'}
Published: {'temperature': 4.8, 'humidity': 69, 'status': 'OK'}
Published: {'temperature': 2.26, 'humidity': 66, 'status': 'OK'}
Published: {'temperature': 4.76, 'humidity': 79, 'status': 'OK'}
Published: {'temperature': 2.02, 'humidity': 72, 'status': 'OK'}
Published: {'temperature': 3.87, 'humidity': 78, 'status': 'OK'}
Published: {'temperature': 2.16, 'humidity': 74, 'status': 'OK'}
Published: {'temperature': 5.8, 'humidity': 66, 'status': 'Warning'}
Published: {'temperature': 5.88, 'humidity': 63, 'status': 'Warning'}
Published: {'temperature': 4.34, 'humidity': 77, 'status': 'OK'}
```

### 3. Data Generation & Transmission

The simulator generates the following telemetry data:

- Temperature (Float, °C)

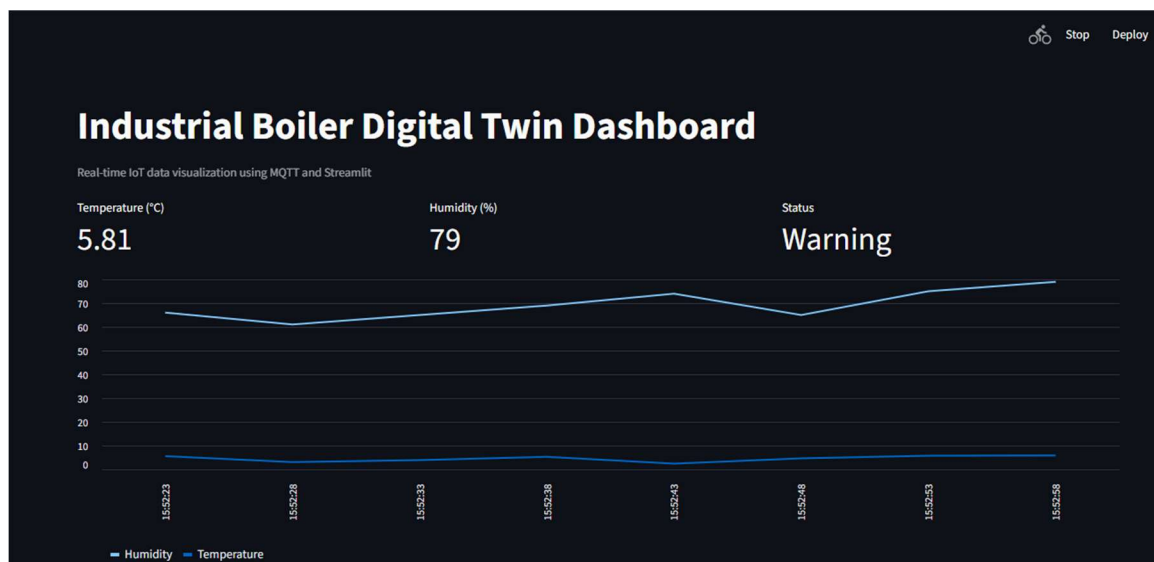
- Humidity (Integer, %)
- Status (String: OK, Warning, Critical)

Data is transmitted every 5 seconds to the MQTT broker using the publish-subscribe model.

```
Windows PowerShell
Published: {'temperature': 5.77, 'humidity': 68, 'status': 'Warning'}
Published: {'temperature': 2.26, 'humidity': 68, 'status': 'OK'}
Published: {'temperature': 4.51, 'humidity': 69, 'status': 'OK'}
Published: {'temperature': 4.8, 'humidity': 69, 'status': 'OK'}
Published: {'temperature': 2.26, 'humidity': 66, 'status': 'OK'}
Published: {'temperature': 4.76, 'humidity': 79, 'status': 'OK'}
Published: {'temperature': 2.02, 'humidity': 72, 'status': 'OK'}
Published: {'temperature': 3.87, 'humidity': 78, 'status': 'OK'}
Published: {'temperature': 2.16, 'humidity': 74, 'status': 'OK'}
Published: {'temperature': 5.8, 'humidity': 66, 'status': 'Warning'}
Published: {'temperature': 5.88, 'humidity': 63, 'status': 'Warning'}
Published: {'temperature': 4.34, 'humidity': 77, 'status': 'OK'}
Published: {'temperature': 2.22, 'humidity': 68, 'status': 'OK'}
Published: {'temperature': 5.21, 'humidity': 78, 'status': 'Warning'}
Published: {'temperature': 5.4, 'humidity': 78, 'status': 'Warning'}
Published: {'temperature': 3.74, 'humidity': 61, 'status': 'OK'}
Published: {'temperature': 3.59, 'humidity': 76, 'status': 'OK'}
Published: {'temperature': 4.14, 'humidity': 60, 'status': 'OK'}
Published: {'temperature': 3.25, 'humidity': 68, 'status': 'OK'}
Published: {'temperature': 2.01, 'humidity': 70, 'status': 'OK'}
Published: {'temperature': 5.04, 'humidity': 61, 'status': 'Warning'}
Published: {'temperature': 3.4, 'humidity': 64, 'status': 'OK'}
Published: {'temperature': 5.03, 'humidity': 74, 'status': 'Warning'}
Published: {'temperature': 2.13, 'humidity': 61, 'status': 'OK'}
Published: {'temperature': 4.67, 'humidity': 65, 'status': 'OK'}
Published: {'temperature': 3.59, 'humidity': 76, 'status': 'OK'}
Published: {'temperature': 3.12, 'humidity': 78, 'status': 'OK'}
Published: {'temperature': 2.57, 'humidity': 74, 'status': 'OK'}
Published: {'temperature': 2.11, 'humidity': 74, 'status': 'OK'}
Published: {'temperature': 4.03, 'humidity': 77, 'status': 'OK'}
Published: {'temperature': 4.99, 'humidity': 78, 'status': 'OK'}
Published: {'temperature': 3.58, 'humidity': 80, 'status': 'OK'}
Published: {'temperature': 5.56, 'humidity': 80, 'status': 'Warning'}
Published: {'temperature': 2.94, 'humidity': 78, 'status': 'OK'}
```

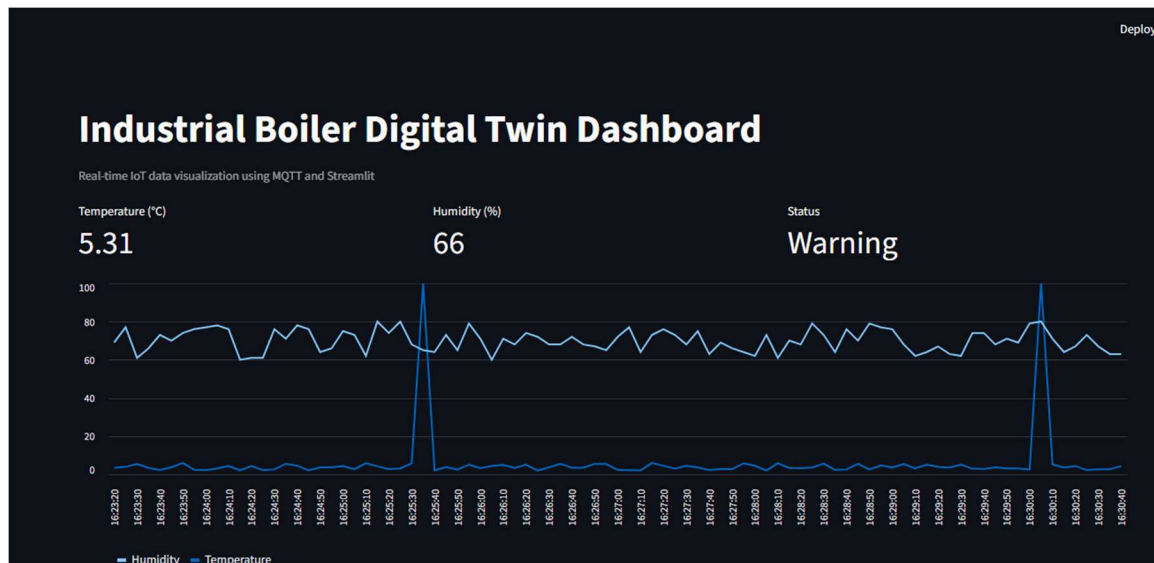
#### 4. Digital Twin Dashboard

A Streamlit-based digital twin dashboard was developed to visualize real-time IoT data. The dashboard displays current temperature, humidity, system status, and a historical line chart showing data trends over time.



## 5. Stress Test & Anomaly Handling

To evaluate system behavior under abnormal conditions, a stress test was performed by artificially increasing the temperature to 100°C. The dashboard immediately reflected this change by displaying a sharp spike in the temperature graph and updating the system status to a critical condition. This confirms that the system correctly handles anomaly scenarios.



## 6. Conclusion

This assignment demonstrates how virtual IoT devices and digital twin dashboards can be used to simulate, monitor, and analyze industrial systems efficiently without physical hardware. The approach improves testing reliability and reduces deployment risks.