Week 1 – Environment Setup and Testing

✅Objective

Set up a basic MQTT environment using:

- Mosquitto (MQTT broker)
- Python (`paho-mqtt`) to publish messages
- Node-RED to visualize message flow

✔️Environment Setup

Mosquitto Broker

- Installed and added to PATH
- Verified `mosquitto -v` command runs correctly

Python Setup

Installed the `paho-mqtt` library:

```
pip install paho-mqtt
```

Test Script (`mqtt_test.py`)

```
import paho.mqtt.client as mqtt
import time

broker = "localhost"
topic = "lab/test"

client = mqtt.Client()
client.connect(broker)

try:
    while True:
        client.publish(topic, "Dean's control lab test message")
        time.sleep(2)
except KeyboardInterrupt:
    client.disconnect()
```

🛄Node-RED Test

Flow Description

- Used MQTT In node subscribed to topic: `lab/test`
- Connected to a Debug node to display messages in real-time
- Successfully received and displayed messages from Python script

To Export Flow:

In Node-RED:

- Go to top-right menu → Export → Clipboard
- Save JSON as `node_red_test.json`

🧠Notes

- **Mosquitto** is the **MQTT broker** (middleware). It runs locally on `localhost:1883` and handles message routing between publishers and subscribers.
- The **Python script** (using `paho-mqtt`) is the **publisher**, sending the message `"Dean's control lab test message"` to the topic `lab/test`.
- **Node-RED** is the **subscriber**, listening to the topic `lab/test` on `localhost`. The MQTT In node receives messages and passes them to a Debug node for display.
- `mqtt.Client()` creates the MQTT client instance
- `client.connect()` connects to the local Mosquitto broker
- Messages are published to the topic every 2 seconds and appear in Node-RED's debug window

---

⚠️Troubleshooting

- "mosquitto" not recognized → Ensure Mosquitto is installed and added to system PATH
- Red line in Vim → Fixed by customizing `CursorLine` highlight in `.vimrc`
- `.vimrc` also adjusted to support Python coding (indentation, color scheme, etc.)
