

Project Development Phase
Sprint – 2

Date	7 November 2022
Team ID	PNT2022TMID11546
Project Name	Hazardous Area Monitoring for Industrial Plant powered by IoT
Maximum Marks	2

Task:

Obtain the sensed data and visualize it in Node-RED.

Steps:

1. IBM IoT node is used to gather sensor data and the necessary API key is provided to establish connection.
2. Using functions namely Temperature, the data is obtained independently and displayed in dashboard.
3. Dashboard Nodes are used to display the sensed data to the user in a portal.

Codes for functions:

Temperature - `msg.payload = msg.payload.temp`
 `return msg;`

Output:

1. Data from Node-RED

The screenshot displays the Node-RED web interface in a browser. The top navigation bar includes tabs for 'Verify your identity', 'IBM', 'Node-RED : node-red', 'IBM Cloudant docs', 'Node-RED Dashboard', and 'MIT App Inventor'. The browser address bar shows the URL: `node-red-avsut-2022-11-14.au-syd.mybluemix.net/red/#flow/907ddfc4a4e67ced`.

The main workspace, titled 'Flow 1', contains the following nodes and connections:

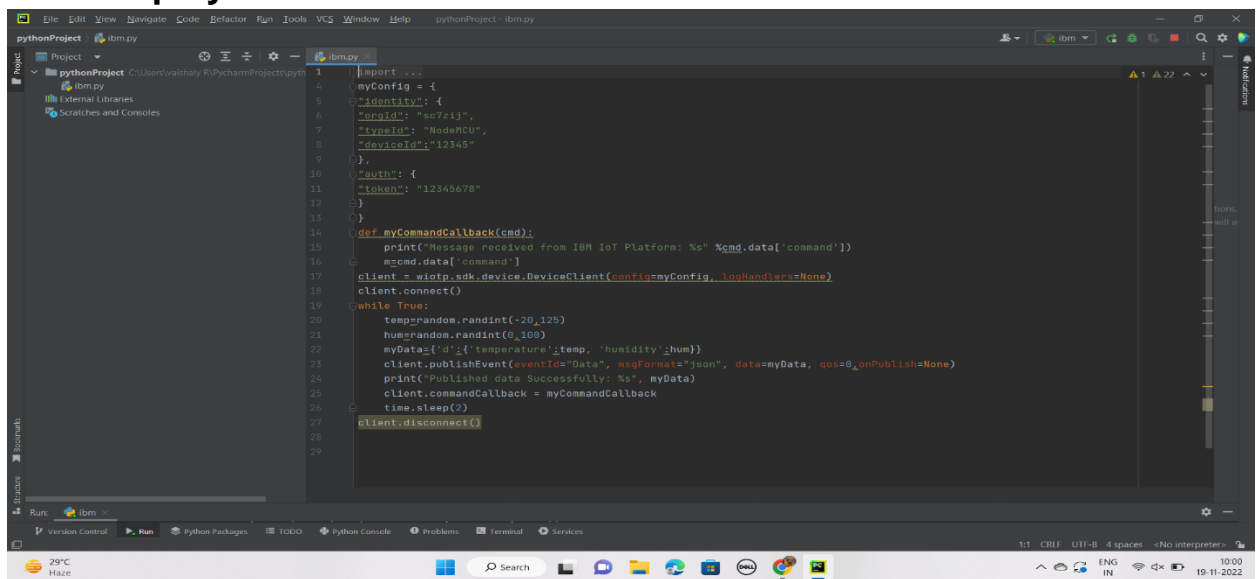
- IBM IoT** node (connected) is connected to two **function** nodes: `temperature` and `humidity`.
- The `temperature` function node is connected to a **Temp** output node.
- The `humidity` function node is connected to a **Humi** output node.
- A **[get] /data** node is connected to a **function** node, which is then connected to an **http** output node.

The left sidebar shows a list of available nodes: button, dropdown, switch, slider, numeric, text input, date picker, colour picker, form, text, gauge, chart, audio out, notification, ui control, and template.

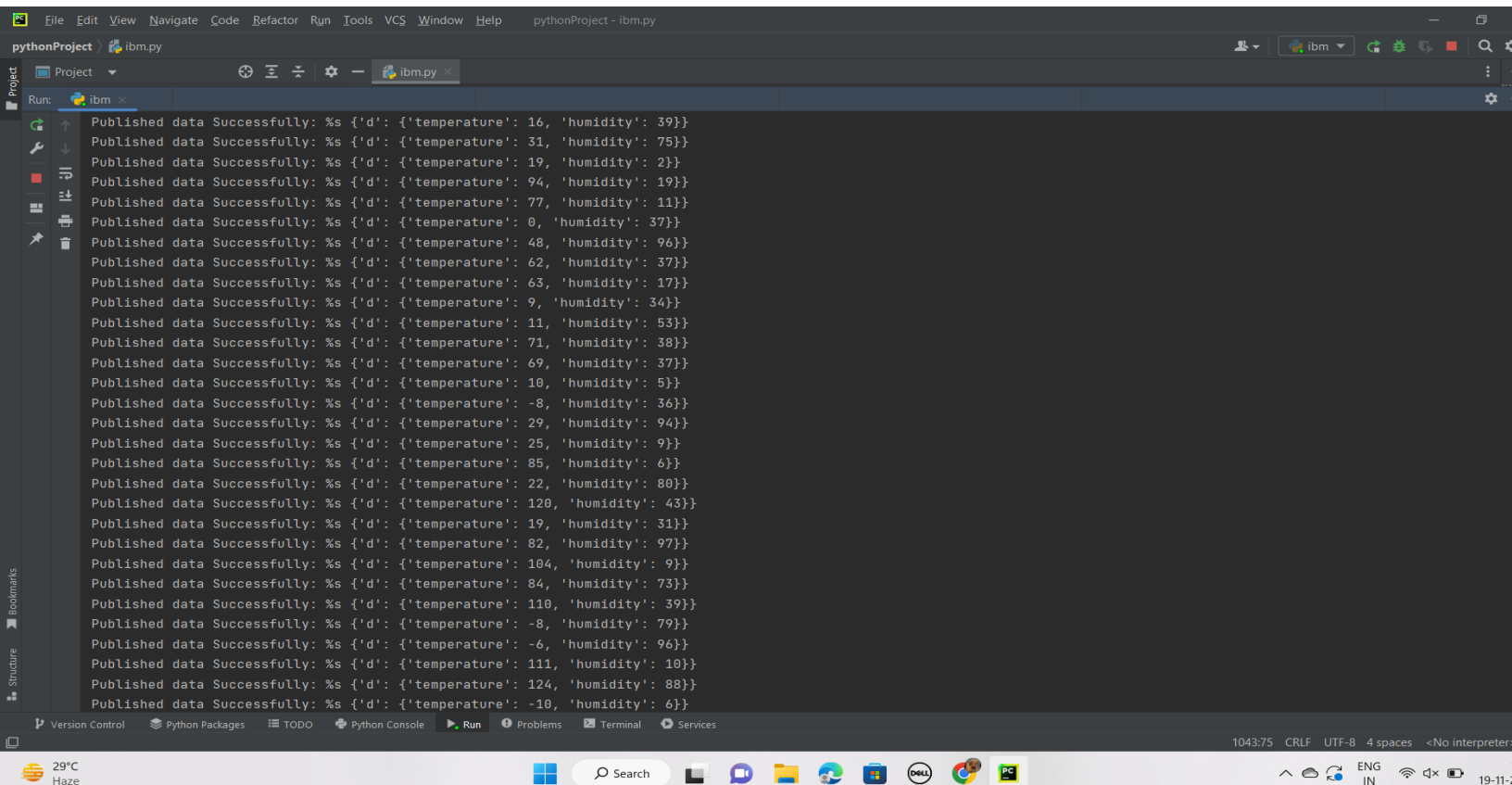
The right sidebar features a **debug** console showing a series of messages. Each message includes a timestamp (11/19/2022, 9:54:10 AM), a device ID (c27387f7365e64b4), and a payload (number). The messages are numbered 13 through 14.

The bottom status bar indicates a temperature of 29°C, a 'Haze' weather condition, and the system time is 10:02 on 19-11-2022.

2. Data displayed in dashboard.



```
1 import ...
2
3
4 myConfig = {
5     "identity": {
6         "orgId": "sc7z1j",
7         "typeId": "NodeMCU",
8         "deviceId": "12345"
9     },
10    "auth": {
11        "token": "12345678"
12    }
13 }
14
15 def myCommandCallback(cmd):
16     print("Message received from IBM IoT Platform: %s" %cmd.data['command'])
17     mcmd.data['command']
18
19 client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandler=None)
20 client.connect()
21
22 while True:
23     temp=random.randint(-20,125)
24     hum=random.randint(0,100)
25     myData={ 'd': {'temperature':temp, 'humidity':hum}}
26     client.publishEvent(eventId="Data", msgFormat="json", data=myData, qos=0,onPublish=None)
27     print("Published data Successfully: %s" %myData)
28     client.commandCallback = myCommandCallback
29     time.sleep(2)
30
31 client.disconnect()
```



```
Published data Successfully: %s {'d': {'temperature': 16, 'humidity': 39}}
Published data Successfully: %s {'d': {'temperature': 31, 'humidity': 75}}
Published data Successfully: %s {'d': {'temperature': 19, 'humidity': 2}}
Published data Successfully: %s {'d': {'temperature': 94, 'humidity': 19}}
Published data Successfully: %s {'d': {'temperature': 77, 'humidity': 11}}
Published data Successfully: %s {'d': {'temperature': 0, 'humidity': 37}}
Published data Successfully: %s {'d': {'temperature': 48, 'humidity': 96}}
Published data Successfully: %s {'d': {'temperature': 62, 'humidity': 37}}
Published data Successfully: %s {'d': {'temperature': 63, 'humidity': 17}}
Published data Successfully: %s {'d': {'temperature': 9, 'humidity': 34}}
Published data Successfully: %s {'d': {'temperature': 11, 'humidity': 53}}
Published data Successfully: %s {'d': {'temperature': 71, 'humidity': 38}}
Published data Successfully: %s {'d': {'temperature': 69, 'humidity': 37}}
Published data Successfully: %s {'d': {'temperature': 10, 'humidity': 5}}
Published data Successfully: %s {'d': {'temperature': -8, 'humidity': 36}}
Published data Successfully: %s {'d': {'temperature': 29, 'humidity': 94}}
Published data Successfully: %s {'d': {'temperature': 25, 'humidity': 9}}
Published data Successfully: %s {'d': {'temperature': 85, 'humidity': 6}}
Published data Successfully: %s {'d': {'temperature': 22, 'humidity': 80}}
Published data Successfully: %s {'d': {'temperature': 120, 'humidity': 43}}
Published data Successfully: %s {'d': {'temperature': 19, 'humidity': 31}}
Published data Successfully: %s {'d': {'temperature': 82, 'humidity': 97}}
Published data Successfully: %s {'d': {'temperature': 104, 'humidity': 9}}
Published data Successfully: %s {'d': {'temperature': 84, 'humidity': 73}}
Published data Successfully: %s {'d': {'temperature': 110, 'humidity': 39}}
Published data Successfully: %s {'d': {'temperature': -8, 'humidity': 79}}
Published data Successfully: %s {'d': {'temperature': -6, 'humidity': 96}}
Published data Successfully: %s {'d': {'temperature': 111, 'humidity': 10}}
Published data Successfully: %s {'d': {'temperature': 124, 'humidity': 88}}
Published data Successfully: %s {'d': {'temperature': -10, 'humidity': 6}}
```

3. Temperature

