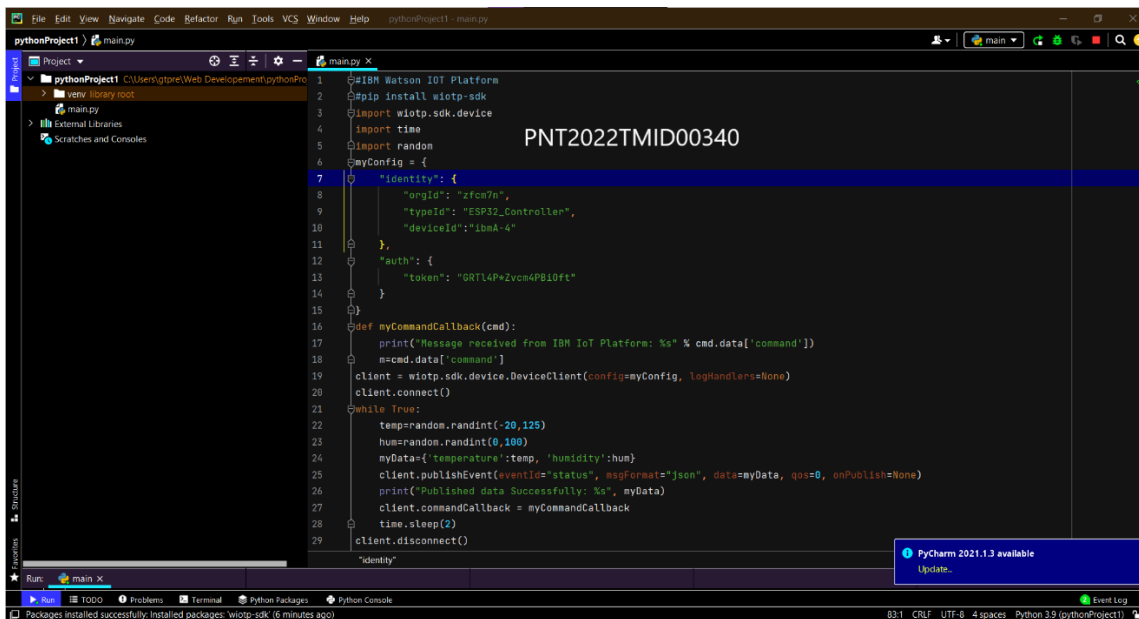


# FINAL DELIVERABLES

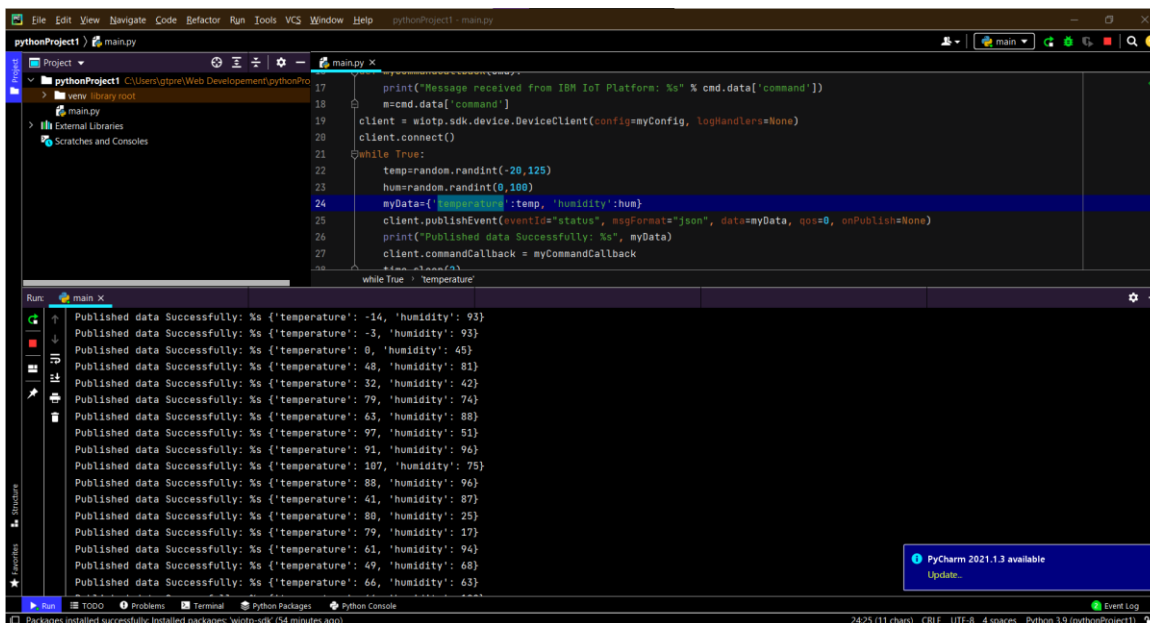
|              |   |
|--------------|---|
| Date         | 12 November 2022  |
| Team ID      | PNT2022TMID00340  |
| Project Name | Hazardous Area Monitoring for Industrial Plant powered by IoT |

## PYTHON CODE:



```
1 #IBM Watson IoT Platform
2 #pip install wiotp-sdk
3 import wiotp.sdk.device
4 import time
5 import random
6 myConfig = {
7     "identity": {
8         "orgId": "zfcw7n",
9         "typeId": "ESP32_Controller",
10        "deviceId": "lbnA-4"
11    },
12    "auth": {
13        "token": "GR114P*Zvcn4PB10ft"
14    }
15 }
16 def myCommandCallback(cmd):
17     print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
18     m=cmd.data['command']
19     client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
20     client.connect()
21 while True:
22     temp=random.randint(-20,125)
23     hum=random.randint(0,100)
24     myData={'temperature':temp, 'humidity':hum}
25     client.publishEvent(eventId='status', msgFormat='json', data=myData, qos=0, onPublish=None)
26     print("Published data Successfully: %s", myData)
27     client.commandCallback = myCommandCallback
28     time.sleep(2)
29     client.disconnect()
```

## PYTHON OUTPUT:



```
Published data Successfully: %s {'temperature': -14, 'humidity': 93}
Published data Successfully: %s {'temperature': -3, 'humidity': 93}
Published data Successfully: %s {'temperature': 0, 'humidity': 45}
Published data Successfully: %s {'temperature': 48, 'humidity': 81}
Published data Successfully: %s {'temperature': 32, 'humidity': 42}
Published data Successfully: %s {'temperature': 79, 'humidity': 74}
Published data Successfully: %s {'temperature': 63, 'humidity': 88}
Published data Successfully: %s {'temperature': 97, 'humidity': 51}
Published data Successfully: %s {'temperature': 91, 'humidity': 96}
Published data Successfully: %s {'temperature': 107, 'humidity': 75}
Published data Successfully: %s {'temperature': 88, 'humidity': 96}
Published data Successfully: %s {'temperature': 41, 'humidity': 87}
Published data Successfully: %s {'temperature': 80, 'humidity': 25}
Published data Successfully: %s {'temperature': 79, 'humidity': 17}
Published data Successfully: %s {'temperature': 61, 'humidity': 94}
Published data Successfully: %s {'temperature': 49, 'humidity': 48}
Published data Successfully: %s {'temperature': 66, 'humidity': 63}
```

## WOKIWI SIMULATION AND OUTPUT:

WOKWI PROJECT: 348135250799362644

FINAL DELIVERABLE(PNT2022TMID00340) - Hazardous Area Monitoring for Industrial Plants powered by IoT

Sketch.ino | diagram.json | libraries.txt | Library Manager

```
1 #include "DHT.h" // library for dht22
2 #include <WiFi.h> //library for wifi
3 #include <PubSubClient.h> //library for MQTT
4 #define DHTPIN 15 // what pin we're connected to
5 #define DHTTYPE DHT22 // define type of sensor DHT 11
6 #define GAS_SENSOR 34 //GAS SENSOR MQ-02
7
8 DHT dht (DHTPIN, DHTTYPE); // creating the instance by passing pin and type of dht connect
9 void callback(char* subscribtopic, byte* payload, unsigned int payloadlength);
10
11 //-----credentials of IBM Accounts-----
12 #define ORG "zfcw7n" //IBM ORGANITION ID
13 #define DEVICE_TYPE "ESP32_Controller" //Device type mentioned in ibm wa
14 #define DEVICE_ID "ibmA-4" //Device ID mentioned in ibm watson IOT Platfo
15 #define TOKEN "GRT14P*ZvcM4PBioft" //Token
16 String data3;
17 float h, t;
18 int val;
19
20 //----- Customise the above values -----
21 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // Server Name
22 char publishTopic[] = "iot-2/evt/Data/fmt/json"; // topic name and type o
23 char subscribtopic[] = "iot-2/cmd/command/fmt/String"; // cmd REPRESENT command
24 char authMethod[] = "use-token-auth"; // authentication method
25 char token[] = TOKEN;
26 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; //client id
27
28 //-----
29 WiFiClient wificlient; // creating the instance for wificlient
30 PubSubClient client(server, 1883, callback, wificlient); //calling the predefined client
31 const int DHT_PIN = 15;
32
33 void setup()
34 {
```

Simulation

00:30.843 96%

Editing DHT22

Temperature: 29.3°C

Humidity: 19.5%

Humidity:19.50

Sending payload: {"Temperature":29.30,"Humidity":19.50}

Publish ok

Temperature:29.30

Humidity:19.50

Sending payload: {"Temperature":29.30,"Humidity":19.50}

Publish ok

WOKWI PROJECT: 348135250799362644

FINAL DELIVERABLE(PNT2022TMID00340) - Hazardous Area Monitoring for Industrial Plants powered by IoT

Sketch.ino | diagram.json | libraries.txt | Library Manager

```
1 #include "DHT.h" // library for dht22
2 #include <WiFi.h> //library for wifi
3 #include <PubSubClient.h> //library for MQTT
4 #define DHTPIN 15 // what pin we're connected to
5 #define DHTTYPE DHT22 // define type of sensor DHT 11
6 #define GAS_SENSOR 34 //GAS SENSOR MQ-02
7
8 DHT dht (DHTPIN, DHTTYPE); // creating the instance by passing pin and type of dht connect
9 void callback(char* subscribtopic, byte* payload, unsigned int payloadlength);
10
11 //-----credentials of IBM Accounts-----
12 #define ORG "zfcw7n" //IBM ORGANITION ID
13 #define DEVICE_TYPE "ESP32_Controller" //Device type mentioned in ibm wa
14 #define DEVICE_ID "ibmA-4" //Device ID mentioned in ibm watson IOT Platfo
15 #define TOKEN "GRT14P*ZvcM4PBioft" //Token
16 String data3;
17 float h, t;
18 int val;
19
20 //----- Customise the above values -----
21 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // Server Name
22 char publishTopic[] = "iot-2/evt/Data/fmt/json"; // topic name and type o
23 char subscribtopic[] = "iot-2/cmd/command/fmt/String"; // cmd REPRESENT command
24 char authMethod[] = "use-token-auth"; // authentication method
25 char token[] = TOKEN;
26 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; //client id
27
28 //-----
29 WiFiClient wificlient; // creating the instance for wificlient
30 PubSubClient client(server, 1883, callback, wificlient); //calling the predefined client
31 const int DHT_PIN = 15;
32
33 void setup()
34 {
```

Simulation

01:25.786 99%

Temperature:35.30

Humidity:35.00

Sending payload: {"Temperature":35.30,"Humidity":35.00}

Publish ok

Temperature:37.80

Humidity:58.50

Sending payload: {"Temperature":37.80,"Humidity":58.50}

Publish ok

Temperature:37.80

Humidity:59.00

Sending payload: {"Temperature":37.80,"Humidity":59.00}

Publish ok

Temperature:21.70

Humidity:100.00

Sending payload: {"Temperature":21.70,"Humidity":100.00}

Publish ok

Temperature:21.70

Humidity:46.00

Sending payload: {"Temperature":21.70,"Humidity":46.00}

Publish ok

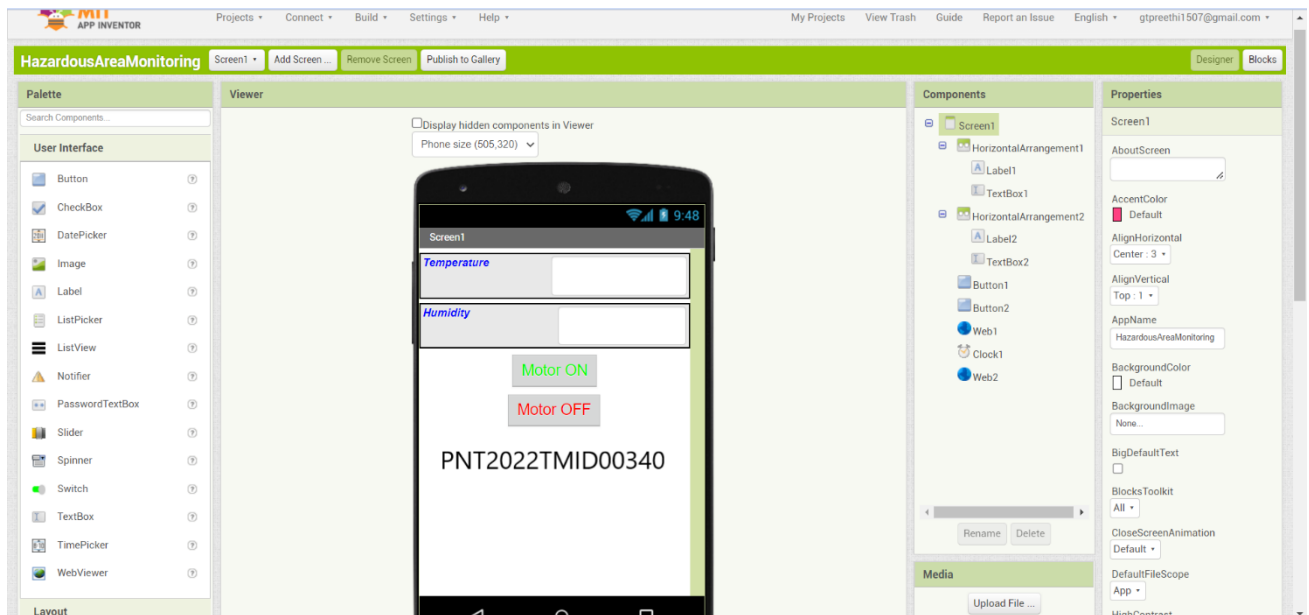
Temperature:42.00

Humidity:81.00

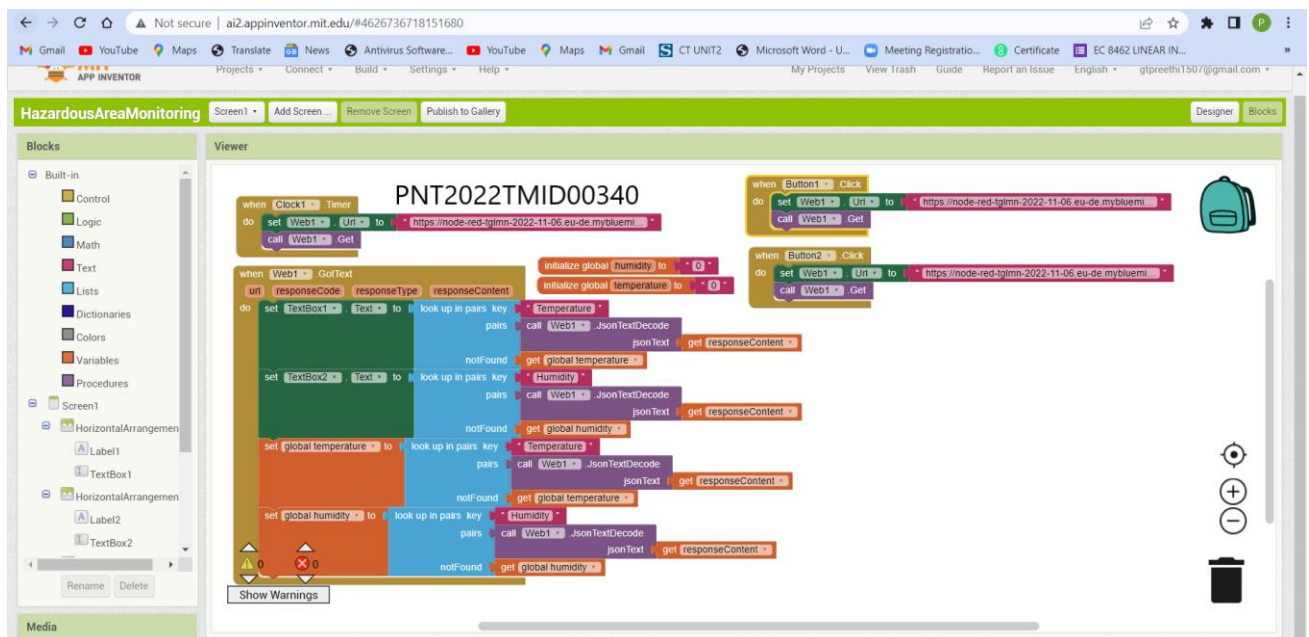
Sending payload: {"Temperature":42.00,"Humidity":81.00}

## MIT-APP INVENTOR ➔

### DESIGNER:



### BLOCKS:



## MOBILE APPLICATION:



*Temperature* 70.100000

*Humidity* 86

Motor ON

Motor OFF

*Temperature* -25.100000

*Humidity* 75

Motor ON

Motor OFF



Temperature 33

Humidity 79

Motor ON

Motor OFF

Temperature 33.300000

Humidity 79

Motor ON

Motor OFF



## IBM WATSON PLATFORM



## DEVICE EVENT LOG:

IBM Watson IoT Platform

Device: ibmA-4, Status: Connected, ESP32\_Controller, Nov 7, 2022 10:15 PM

Recent Events

The recent events listed show the live stream of data that is coming and going from this device.

| Event | Value                                   | Format | Last Received     |
|-------|---|--------|-------------------|
| Data  | {"Temperature": -25.1, "Humidity": 75}  | json   | a few seconds ago |
| Data  | {"Temperature": 33.6, "Humidity": 75}   | json   | a few seconds ago |
| Data  | {"Temperature": 10.6, "Humidity": 61.5} | json   | a few seconds ago |
| Data  | {"Temperature": -5.5, "Humidity": 50.5} | json   | a few seconds ago |
| Data  | {"Temperature": 15.7, "Humidity": 75.5} | json   | a few seconds ago |

Items per page 50 | 1-1 of 1 item

## DEVICE EVENT PAYLOAD:

IBM Watson IoT Platform

Device: ibmA-4, Status: Disconnected

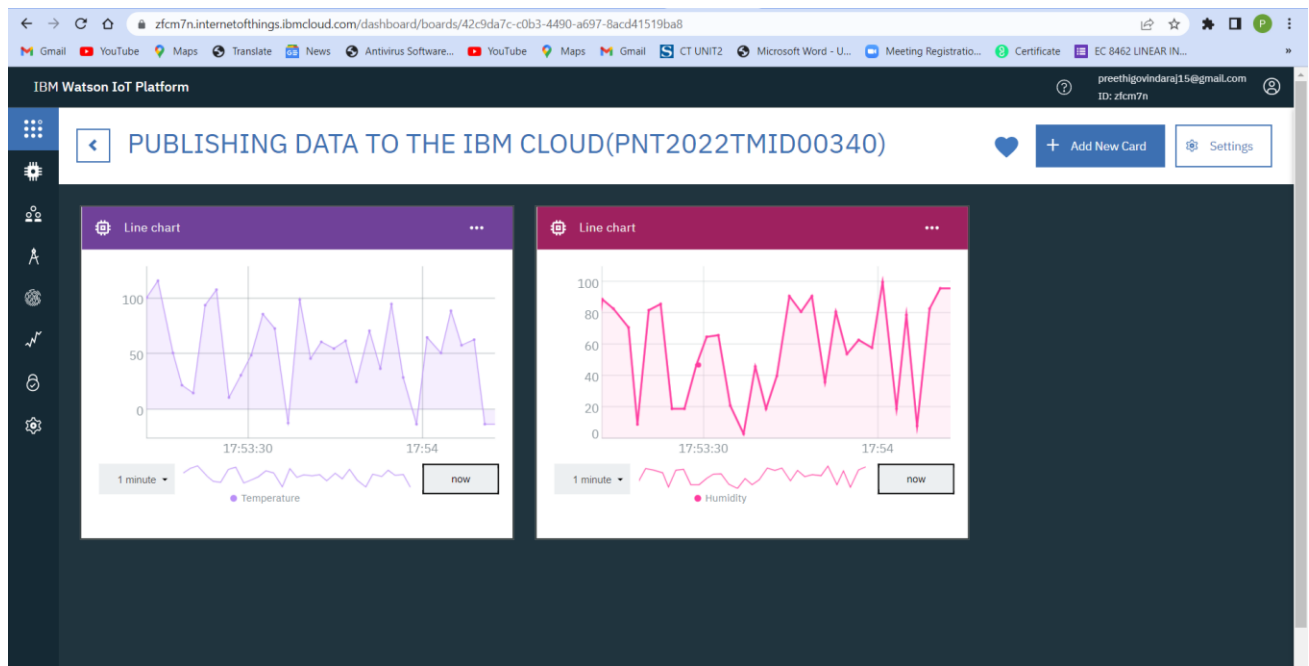
Event Payload

Event Name: Data

Time Received: Nov 12, 2022 2:33 AM

```
1 | {
2 |   "Temperature": 33.6,
3 |   "Humidity": 75
4 | }
```

## DEVICE- BOARD:



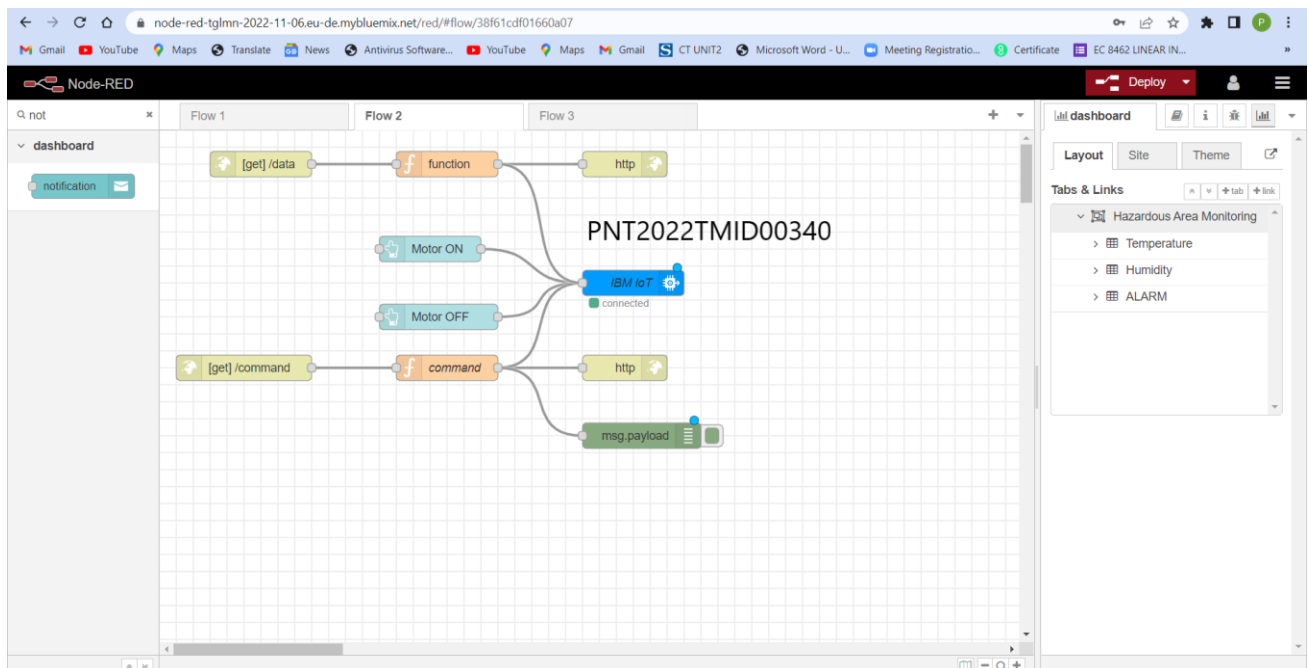
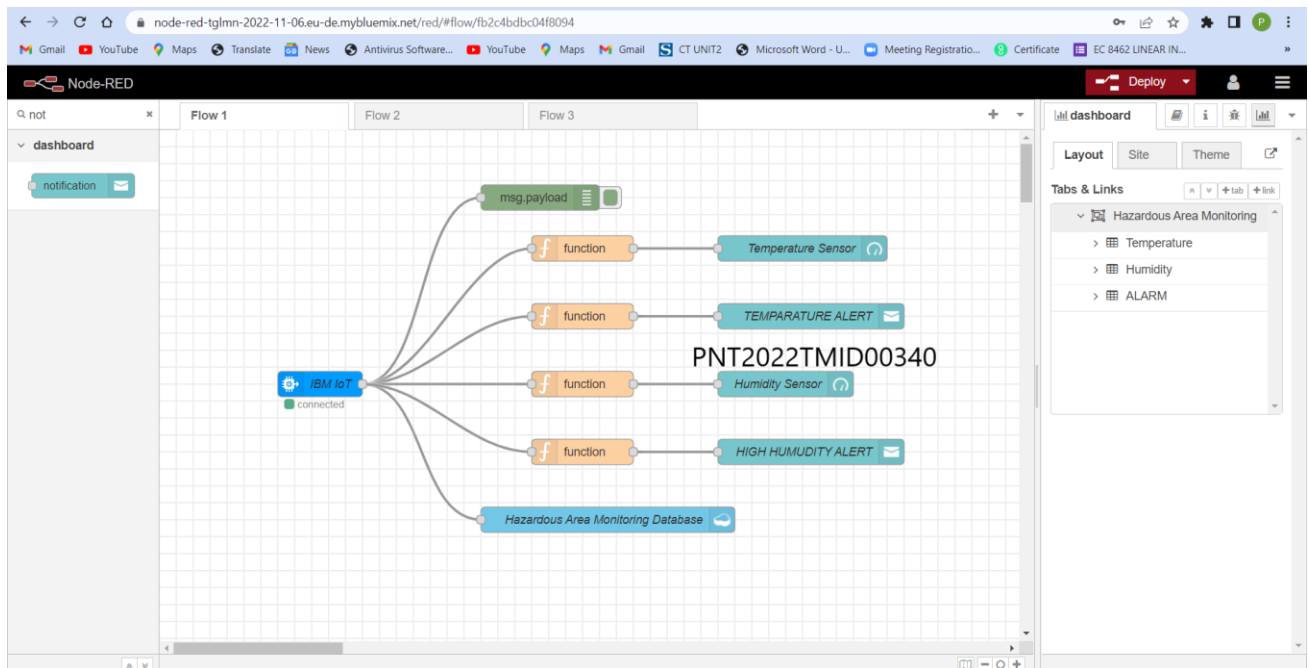
## IBM CLOUDANT DB LOG:

The screenshot shows the IBM Cloudant database interface. The database name is "noderedtglnn202...". The table view is selected, showing a table with columns: "\_id", "humidity", and "temperature". The table contains 10 rows of data. The "humidity" column values range from 6 to 87, and the "temperature" column values range from -5 to 107. The interface includes a sidebar with navigation options like "All Documents", "Query", "Permissions", "Changes", "Design Documents", and "library". There are also buttons for "Table", "Metadata", "JSON", and "Create Document".

| _id                              | humidity | temperature |
|----------------------------------|----------|-------------|
| 0096ab1244940360661f0bce73051181 | 9        | 79          |
| 0096ab1244940360661f0bce730520d0 | 68       | 122         |
| 0096ab1244940360661f0bce730d0d19 | 6        | 109         |
| 0096ab1244940360661f0bce730d1a9b | 72       | 39          |
| 0096ab1244940360661f0bce730d380a | 44       | 105         |
| 0096ab1244940360661f0bce731556d4 | 37       | 12          |
| 0096ab1244940360661f0bce73156b2b | 18       | -5          |
| 0096ab1244940360661f0bce731b7048 | 81       | 5           |
| 0096ab1244940360661f0bce731bbf33 | 25       | 90          |
| 0096ab1244940360661f0bce7320722f | 87       | 11          |
| 0096ab1244940360661f0bce73207f8c | 48       | 49          |
| 0096ab1244940360661f0bce7325d136 | 56       | 107         |

## NODE-RED APP →

### DESIGN FLOW:





WEB UI:

