

Project development phase

Sprint - III

| | |
|--------------|--|
| Date | 11 November 2022 |
| Team ID | PNT2022TMID19483 |
| Project Name | Project - Industry-specific intelligent fire management system |

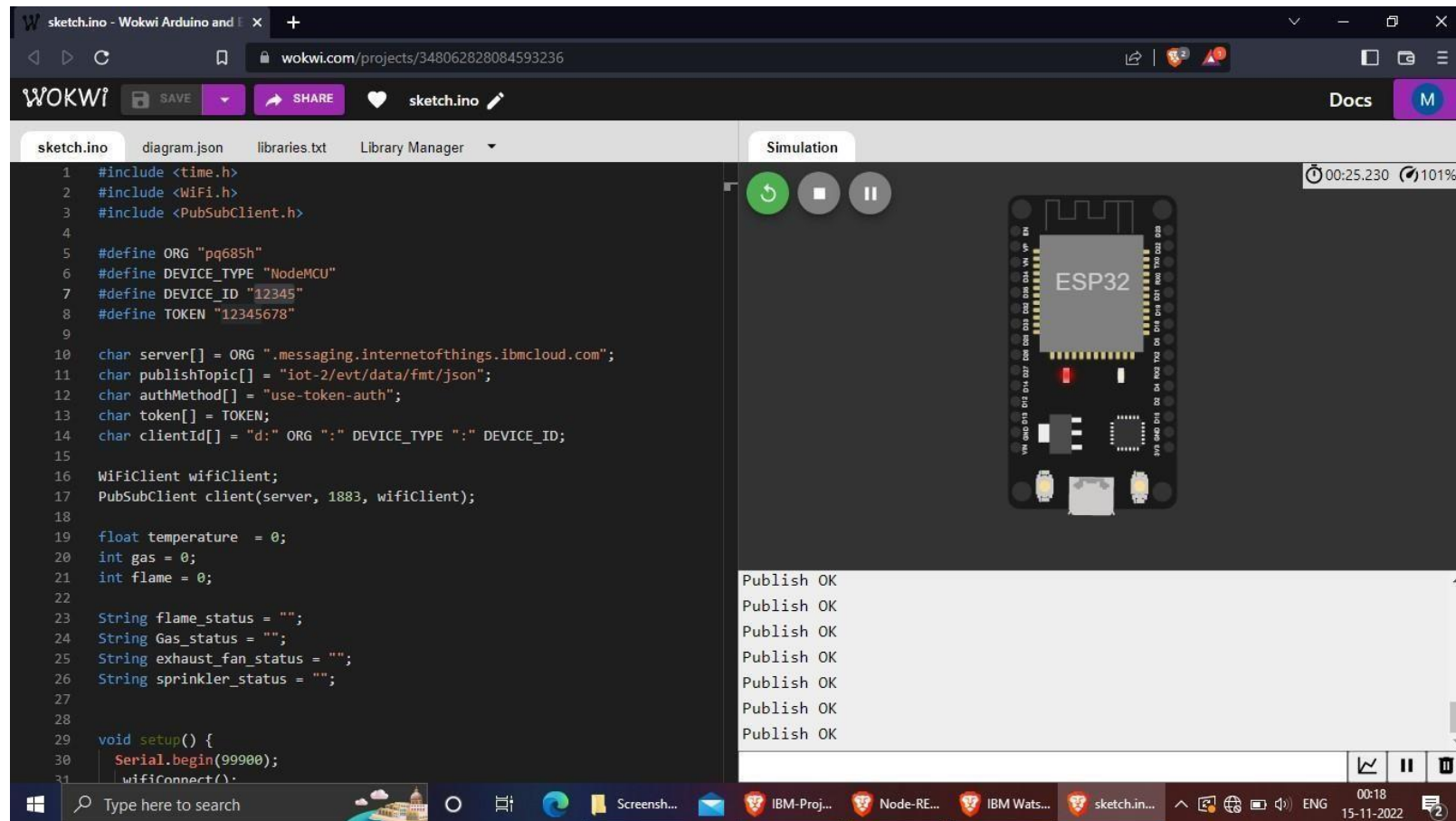
LINK: <https://wokwi.com/projects/348062828084593236>

NODE-RED DASHBOARD UILINK:

<https://node-red-iwivz-2022-11-13.eu-gb.mybluemix.net/ui/#!/0?socketid=RNNTsORzKbrlp-UqAAAu>

OUTPUT:

WOKWI SIMULATOR



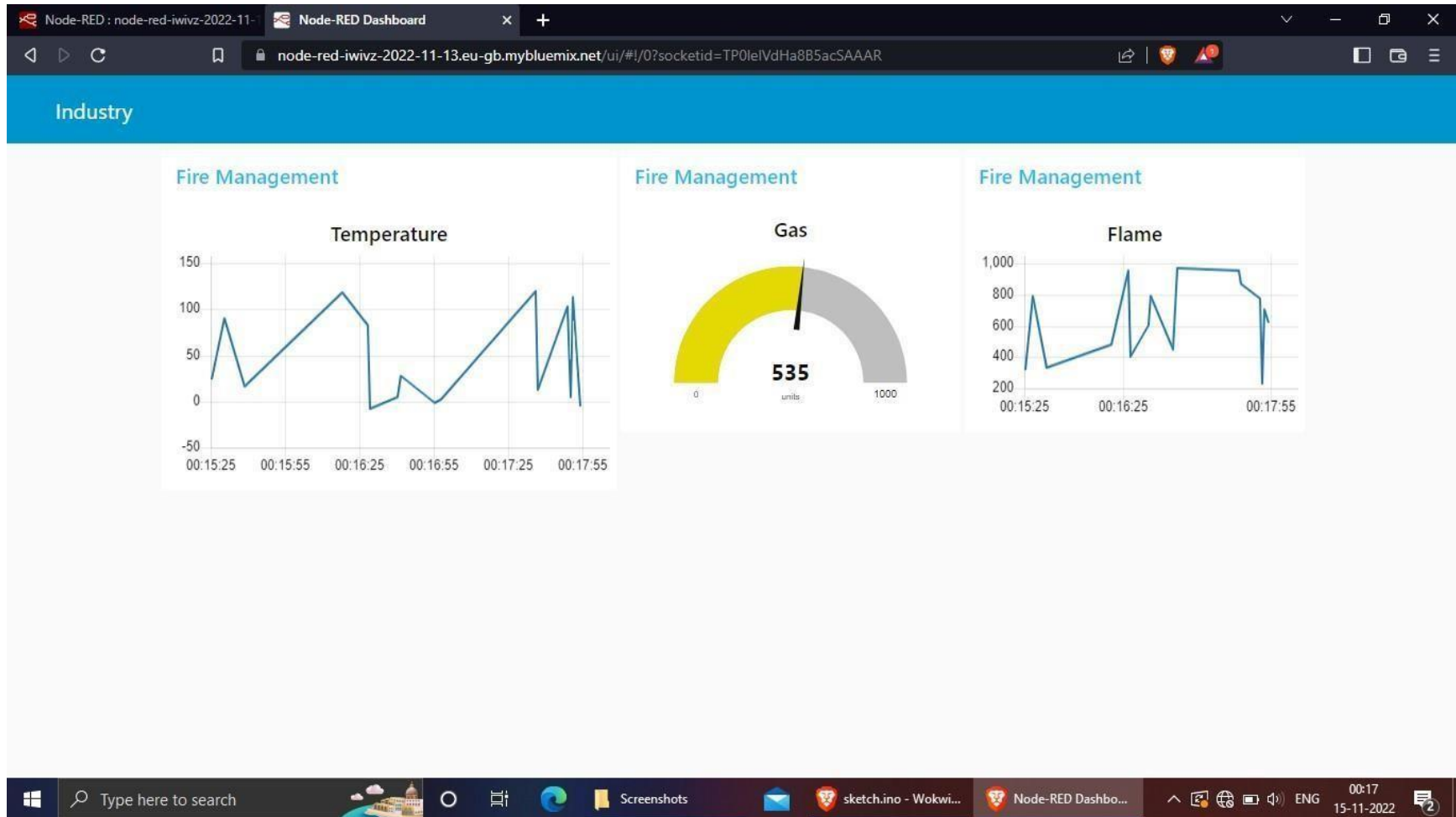
TRANSFERRING DATA FROM IBM WATSON INTO NODE-RED

The screenshot displays the Node-RED web interface in a browser window. The address bar shows the URL: `node-red-iwivz-2022-11-13.eu-gb.mybluemix.net/red/#`. The interface includes a left sidebar with a 'filter nodes' search bar and a list of available nodes such as 'switch', 'numeric', 'text input', 'colour picker', 'date picker', 'slider', 'text', 'gauge', 'form', 'notification', 'audio out', 'ui control', 'chart', and 'template'. The main workspace, titled 'Flow 1', contains a flow starting with an 'IBM IoT' node (labeled 'connected'). This node is connected to a 'msg.payload' node. The output of 'msg.payload' is then distributed to several nodes: 'temp', 'Gas', 'Flame', 'Fire Status', 'Sprinkler Status', 'Gas Status', and 'Exhaust Fan Status'. These nodes are further connected to output nodes: 'Temperature' (with a line graph icon), 'Gas' (with a gauge icon), and 'Flame' (with a line graph icon). The right sidebar features a 'debug' tab showing a log of messages. The log entries include timestamps, node IDs, and payloads, such as:

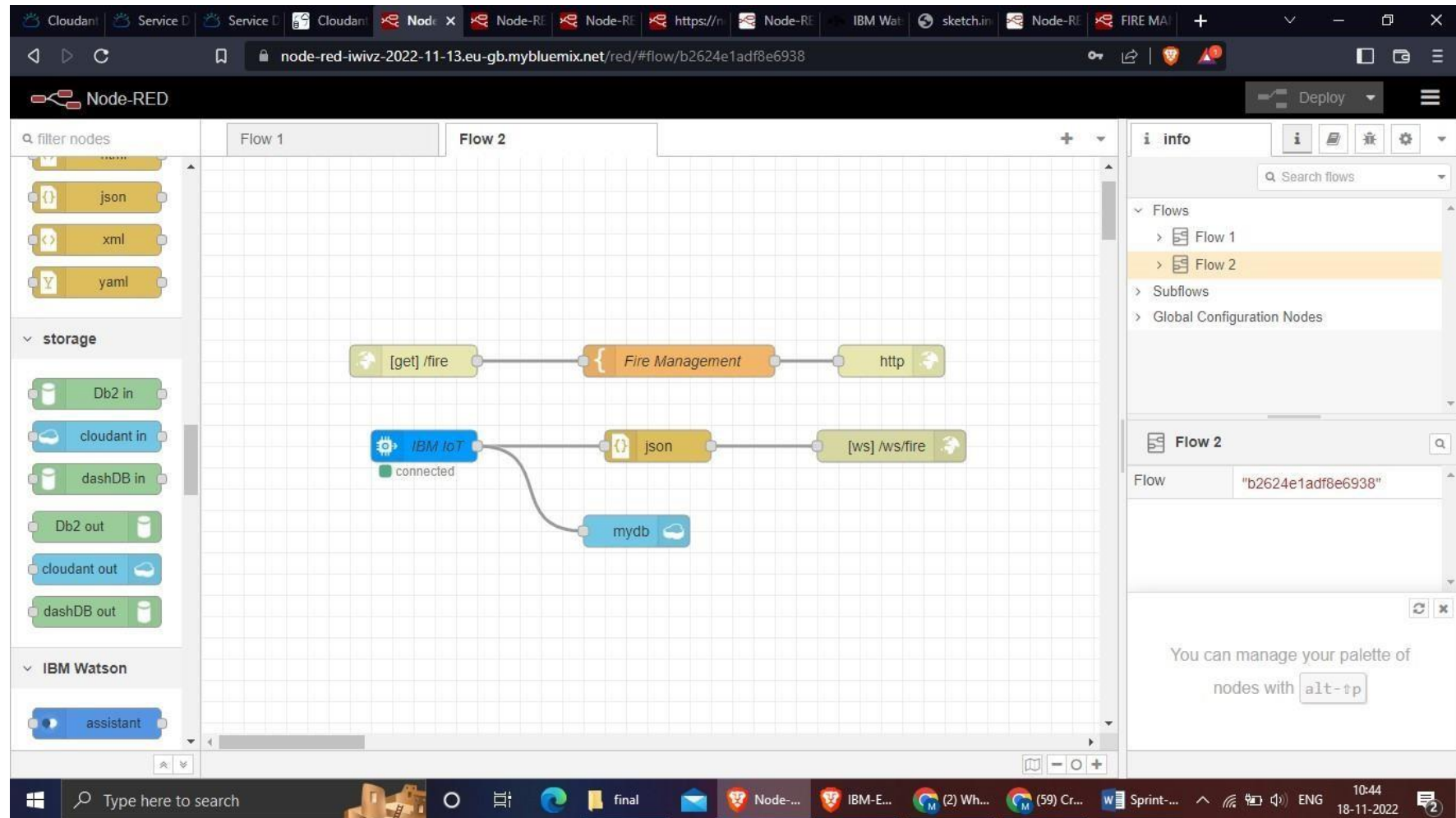
- 11/15/2022, 12:15:39 AM node: f2f2649a.0d0d98
iot-2/type/NodeMCU/id/12345/evt/data/fmt/json :
msg.payload : number
670
- 11/15/2022, 12:15:40 AM node: f2f2649a.0d0d98
iot-2/type/NodeMCU/id/12345/evt/data/fmt/json :
msg.payload : string[7]
"No Fire"
- 11/15/2022, 12:15:41 AM node: f2f2649a.0d0d98
iot-2/type/NodeMCU/id/12345/evt/data/fmt/json :
msg.payload : string[11]
"Not Working"
- 11/15/2022, 12:15:42 AM node: f2f2649a.0d0d98
iot-2/type/NodeMCU/id/12345/evt/data/fmt/json :
msg.payload : string[23]
"Gas Leakage is Detected"
- 11/15/2022, 12:15:43 AM node: f2f2649a.0d0d98
iot-2/type/NodeMCU/id/12345/evt/data/fmt/json :
msg.payload : string[7]
"Working"

The bottom of the image shows the Windows taskbar with the search bar and several open applications, including 'sketch.ino - Wokwi...' and 'Node-RED : node-r...'.

NODE DASHBOARD

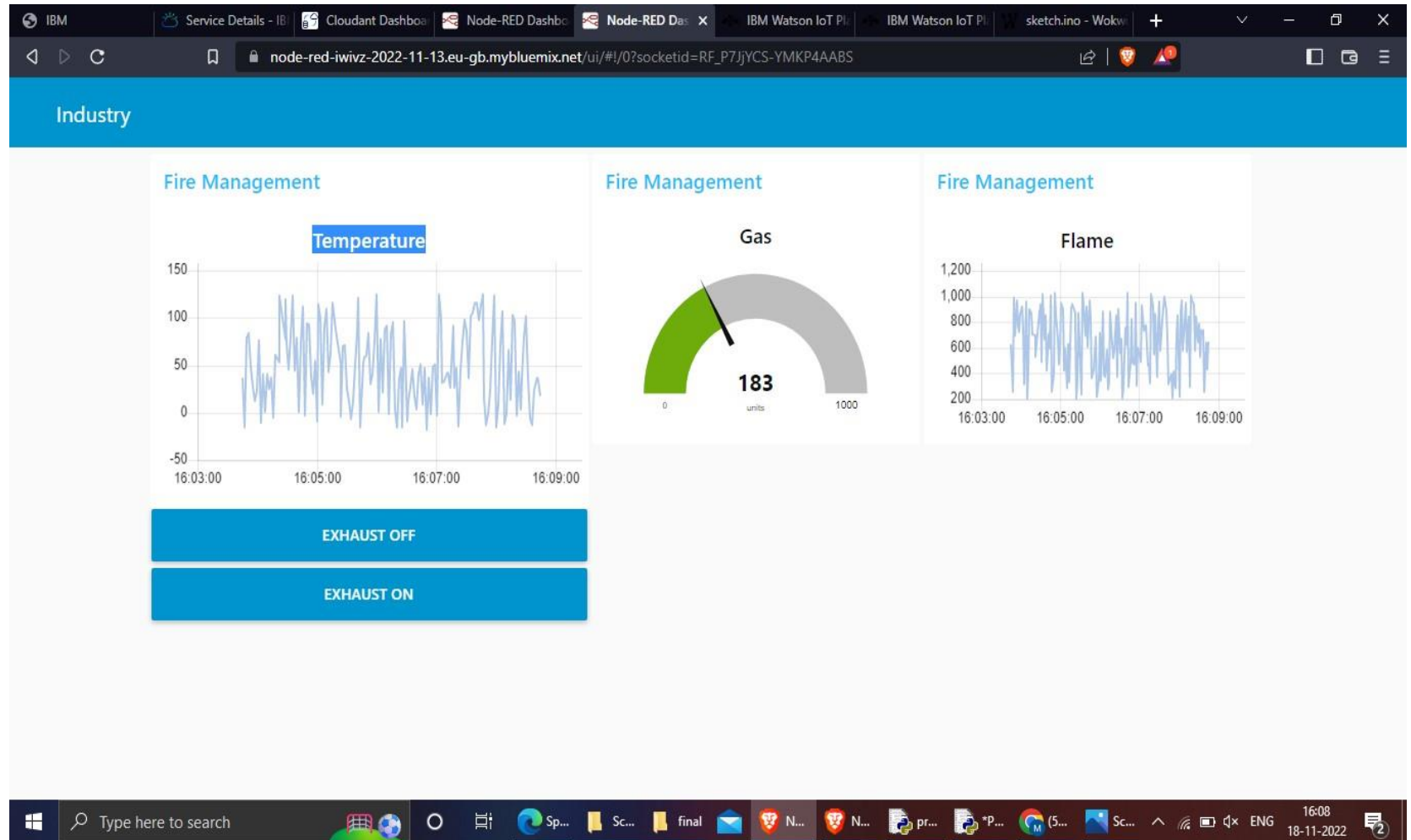


TRANSFERRING DATA FROM NODE-RED INTO WEB UI

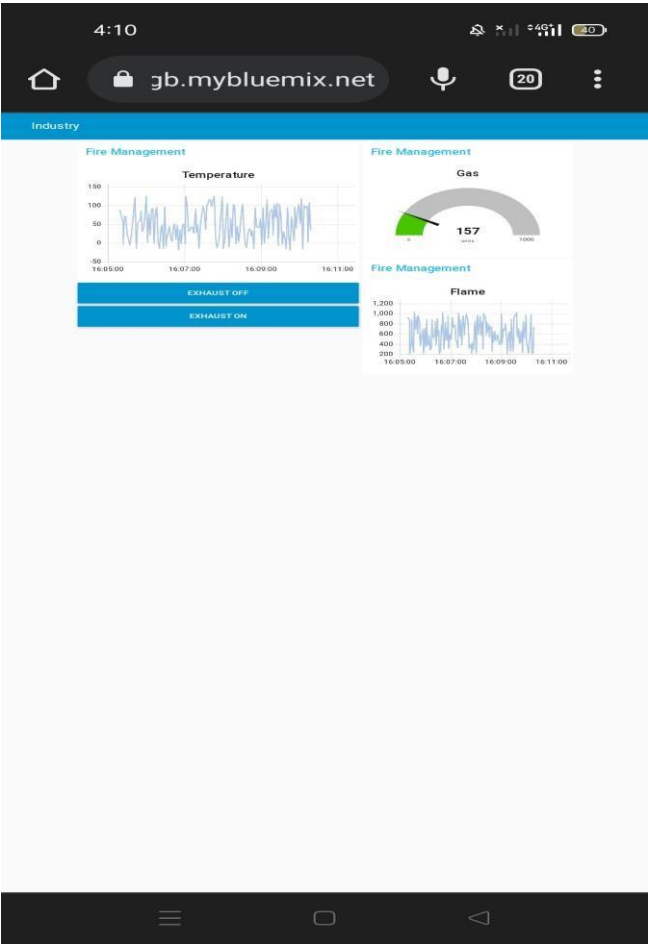


WEB UI

DESKTOP VIEW :



MOBILE VIEW



CLOUDANT:

The screenshot displays the Cloudant dashboard interface. The browser's address bar shows the URL: `ed75f248-e5f6-411c-a1c8-e466095c7806-bluemix.cloudant.com/dashboard.html#database/mydb/_all_docs`. The left sidebar contains navigation options: All Documents (selected), Query, Permissions, Changes, and Design Documents. The main content area shows a table of documents with columns for id, key, and value. The table lists 7 documents, each with a unique ID and a JSON value containing a revision number. At the bottom, a status bar indicates 'Showing document 1 - 7' and 'Documents per page: 20'.

| id | key | value |
|-----------------------------------|-----------------------------------|---|
| 72618fddd3cea28f4100edcd12261a00 | 72618fddd3cea28f4100edcd12261a00 | {"rev": "1-1f66cc2ed0f1fb639600ac4..."} |
| a3100f15787de63c24667f680b004a... | a3100f15787de63c24667f680b004a... | {"rev": "1-4d68bc125889aa3e06f35e..."} |
| adbffbf82906f810019caade1ac78ef8 | adbffbf82906f810019caade1ac78ef8 | {"rev": "1-d5052516c8737348608cf..."} |
| adbffbf82906f810019caade1ac79142 | adbffbf82906f810019caade1ac79142 | {"rev": "1-51f558253a3af2601b2a54..."} |
| ccbe7e2ea854f5f7dfa3f6f4470381f9 | ccbe7e2ea854f5f7dfa3f6f4470381f9 | {"rev": "1-2cc18a1cd7959e8e6c80d..."} |
| ccbe7e2ea854f5f7dfa3f6f4470954a1 | ccbe7e2ea854f5f7dfa3f6f4470954a1 | {"rev": "1-85fb75d7377f5a48a1babd..."} |
| cde24cff3370e8cc9e63467145c375f1 | cde24cff3370e8cc9e63467145c375f1 | {"rev": "1-401973158e2bd03b0f6b2..."} |

CloudantServiceServiceCloudNode-RENode-RENode-REhttps://Node-REIBM WatSketch.inNode-REFIRE MA

ed75f248-e5f6-411c-a1c8-e466095c7806-bluemix.cloudant.com/dashboard.html#database/mydb/72618fddd3cea28f4100edcd12261a00

mydb > 72618fddd3cea28f4100edcd12261a00

Save ChangesCancelUpload AttachmentClone DocumentDelete

123456789101112131415161718

```
"_id": "72618fddd3cea28f4100edcd12261a00",
"_rev": "1-1f66cc2ed0f1fb639600ac461672d675",
"topic": "iot-2/type/NodeMCU/id/12345/evt/data/fmt/json",
"payload": {
  "gas": 960,
  "temperature": 67,
  "flame": 767,
  "fire_status": "Fire is Detected",
  "sprinkler_status": "Working",
  "Gas_status": "Gas Leakage is Detected",
  "exhaust_fan_status": "Working"
},
"deviceId": "12345",
"deviceType": "NodeMCU",
"eventType": "data",
"format": "json"
```

Log Out

Type here to searchfinalCloud...IBM-E... (2) Wh... (59) Cr... Sprint...ENG10:4618-11-2022

CODE:

```
#include <time.h>
#include <WiFi.h>
#include <PubSubClient.h>
```

```
#define ORG "pq685h"
#define DEVICE_TYPE "NodeMCU"
#define DEVICE_ID "12345"
#define TOKEN "12345678"
```

```
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/data/fmt/json";
authMethod[] = "use-token-auth";
token[] = TOKEN; char clientId[] = "d:" ORG ":" DEVICE_TYPE
":" DEVICE_ID;
```

char

```
WiFiClient wifiClient;
PubSubClient client(server, 1883, wifiClient);
```

```
float temperature = 0; int gas =  
0; int flame  
= 0;  
String flame_status = "";  
String Gas_status = "";  
String exhaust_fan_status = "";  
String sprinkler_status = "";
```

```
void setup() { Serial.begin(99900);  
wifiConnect(); mqttConnect();  
}
```

```
void loop() {
```

```
    srand(time(0));
```

```
    //initial variables and random generated data
```

```
    temperature = random(-20,125);          gas =  
random(0,1000);    int flamereading = random(200,1024);  
flame = map(flamereading,200,1024,0,2);
```

```
    //set a flame status    switch
(flame) {    case 0:    flame_status
= "No Fire";
    break;    case 1:
flame_status = "Fire is Detected";
break;
}
```

```
    //send the sprinkler status
```

```
    if(flame==1){    sprinkler_status
= "Working";    }    else{
sprinkler_status =
"Not Working";

}
```

```
    //toggle the fan according to gas reading
```

```
    if(gas > 100){
        Gas_status = "Gas Leakage is Detected";    exhaust_fan_status =
        "Working";
    }    else{
```

```
    Gas_status = "No Gas Leakage is Detected";    exhaust_fan_status =  
"Not Working";  
}
```

//json format for IBM Watson

```
String payload = "{";    payload+="\"gas\":";    payload+=gas;  
    payload+=",";  
payload+="\"temperature\":";  
payload+=(int)temperature;    payload+=",";  
payload+="\"flame\":";    payload+=flamereading;    payload+=",";  
payload+="\"fire_status\":"+"\""+flame_status+"\",";  
payload+="\"sprinkler_status\":"+"\""+sprinkler_status+"\",";  
payload+="\"Gas_status\":"+"\""+Gas_status+"\",";  
    payload+="\"exhaust_fan_status\":"+"\""+exhaust_fan_status+"\"}";
```

```
if(client.publish(publishTopic, (char*) payload.c_str()))  
{  
    Serial.println("Publish OK");  
} else{  
    Serial.println("Publish failed");  
}  
delay(1000);  
if
```

```
(!client.loop())  
  {  
    mqttConnect();  
  }  
}
```

```
void wifiConnect()  
{  
  Serial.print("Connecting to ");  
  Serial.print("Wifi");  
  WiFi.begin("Wokwi-GUEST", "", 6);  
  while (WiFi.status() != WL_CONNECTED)  
  {  
    delay(500);    Serial.print(".");  
  }  
  Serial.print("WiFi connected, IP address: ");  
  Serial.println(WiFi.localIP());  
}
```

```
}  
void mqttConnect()  
{  
  if (!client.connected())  
  {
```

```
    Serial.print("Reconnecting MQTT client to ");  
    Serial.println(server);    while  
    (!client.connect(clientId, authMethod, token))  
    {  
        Serial.print(".");    delay(500);  
    }  
  
    Serial.println();  
}  
}
```