Project development phase

Sprint - III

Date	11 November 2022
Team ID	PNT2022TMID13566
Project Name	Project - Industry-specific intelligent fire management system

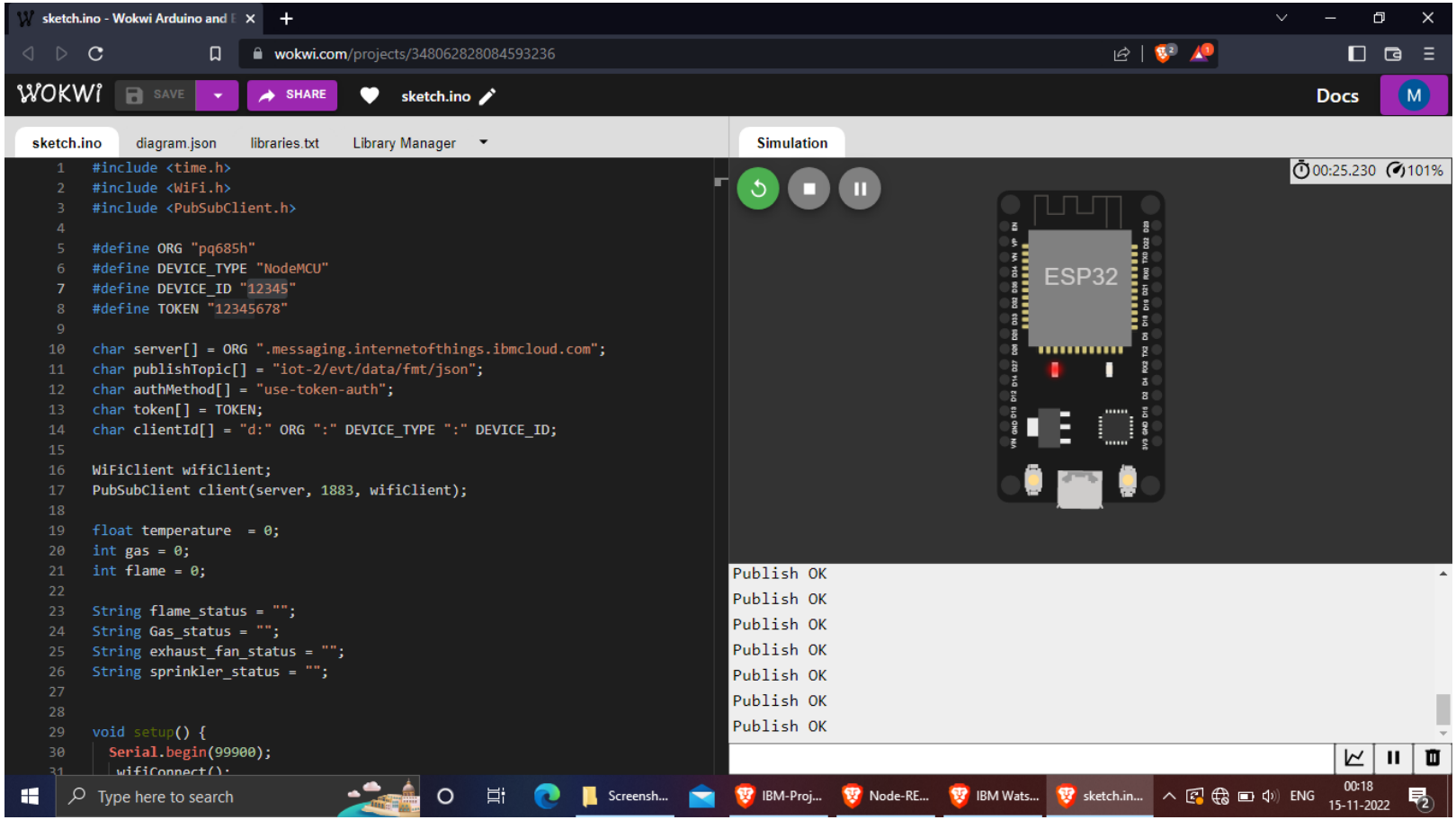
LINK: https://wokwi.com/projects/348062828084593236

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

WEB UI LINK : https://node-red-dashboard059.eu-gb.mybluemix.net/fire

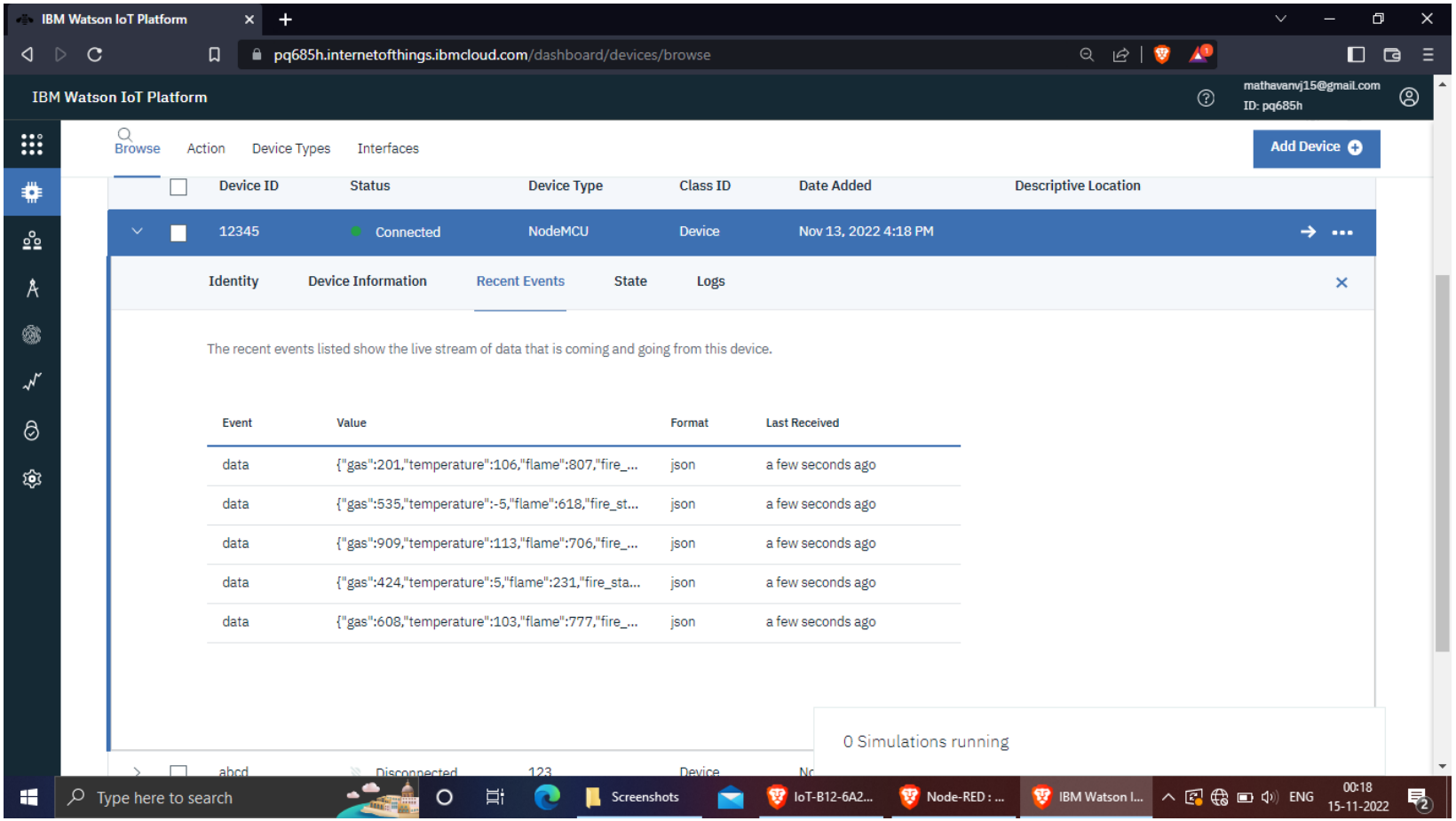
OUTPUT:

WOKWI SIMULATOR

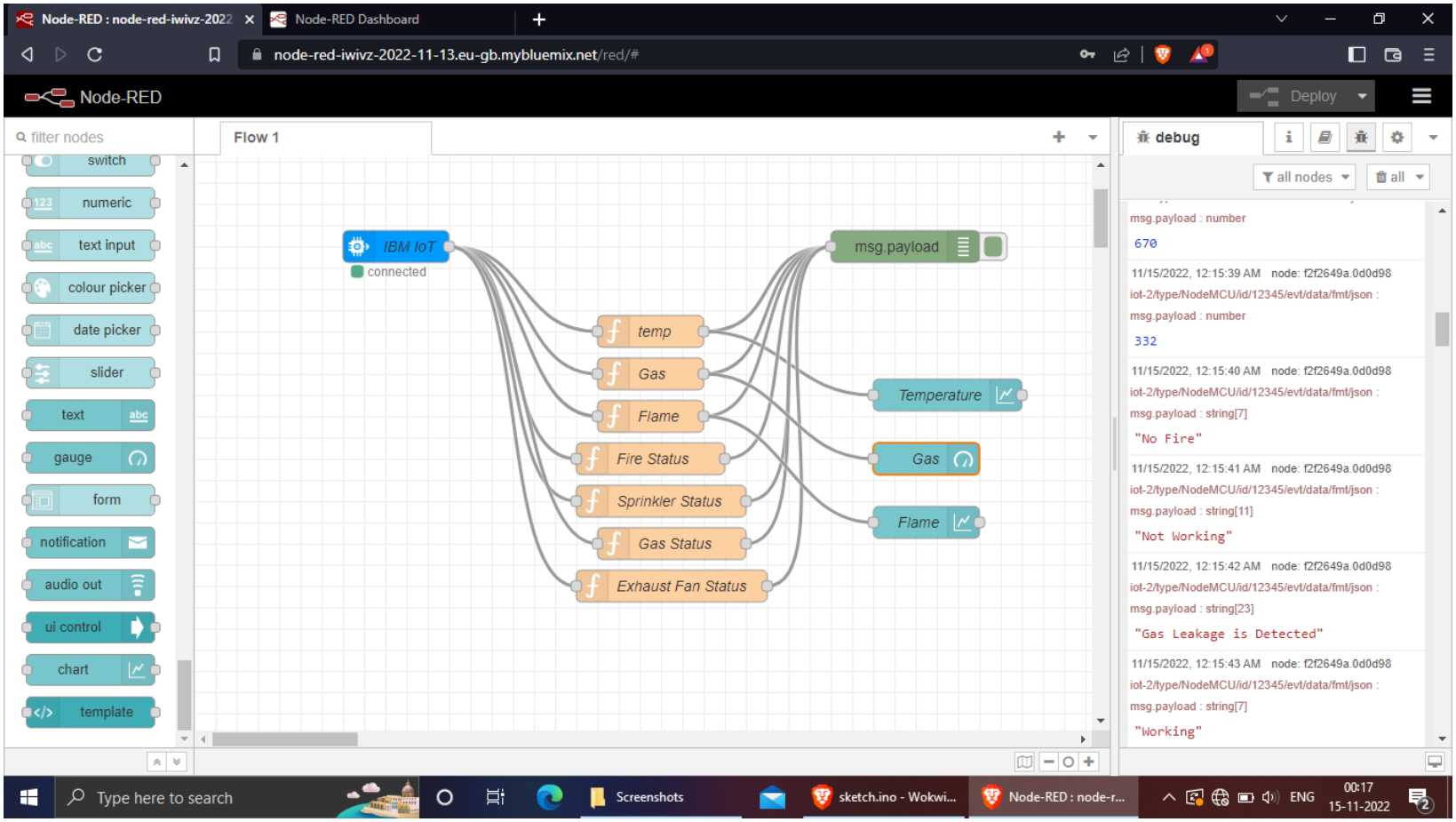



```
#include <time.h>
#include <WiFi.h>
#include <PubSubClient.h>

#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";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;

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();
```

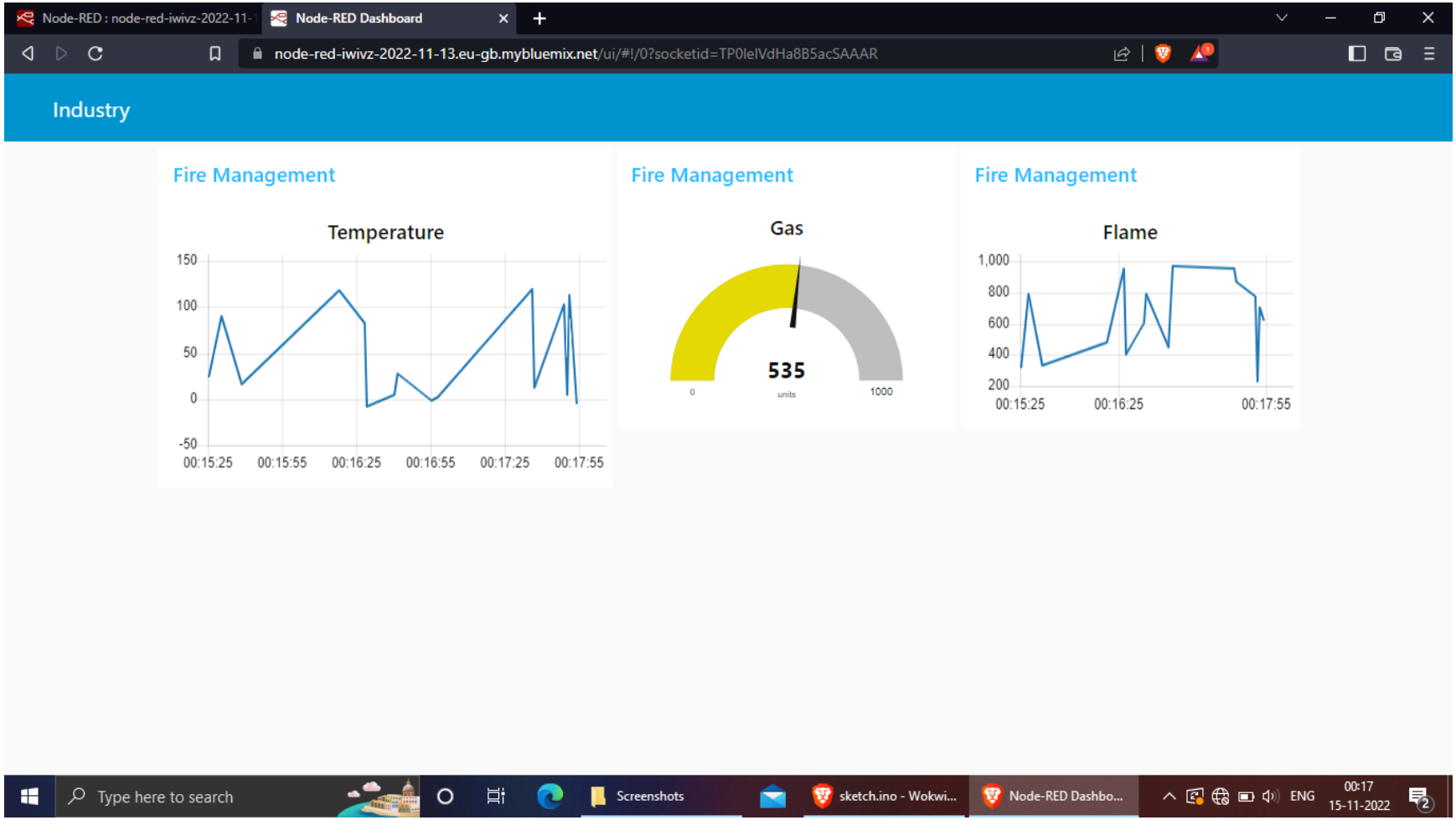
IBM WATSON OUTPUT



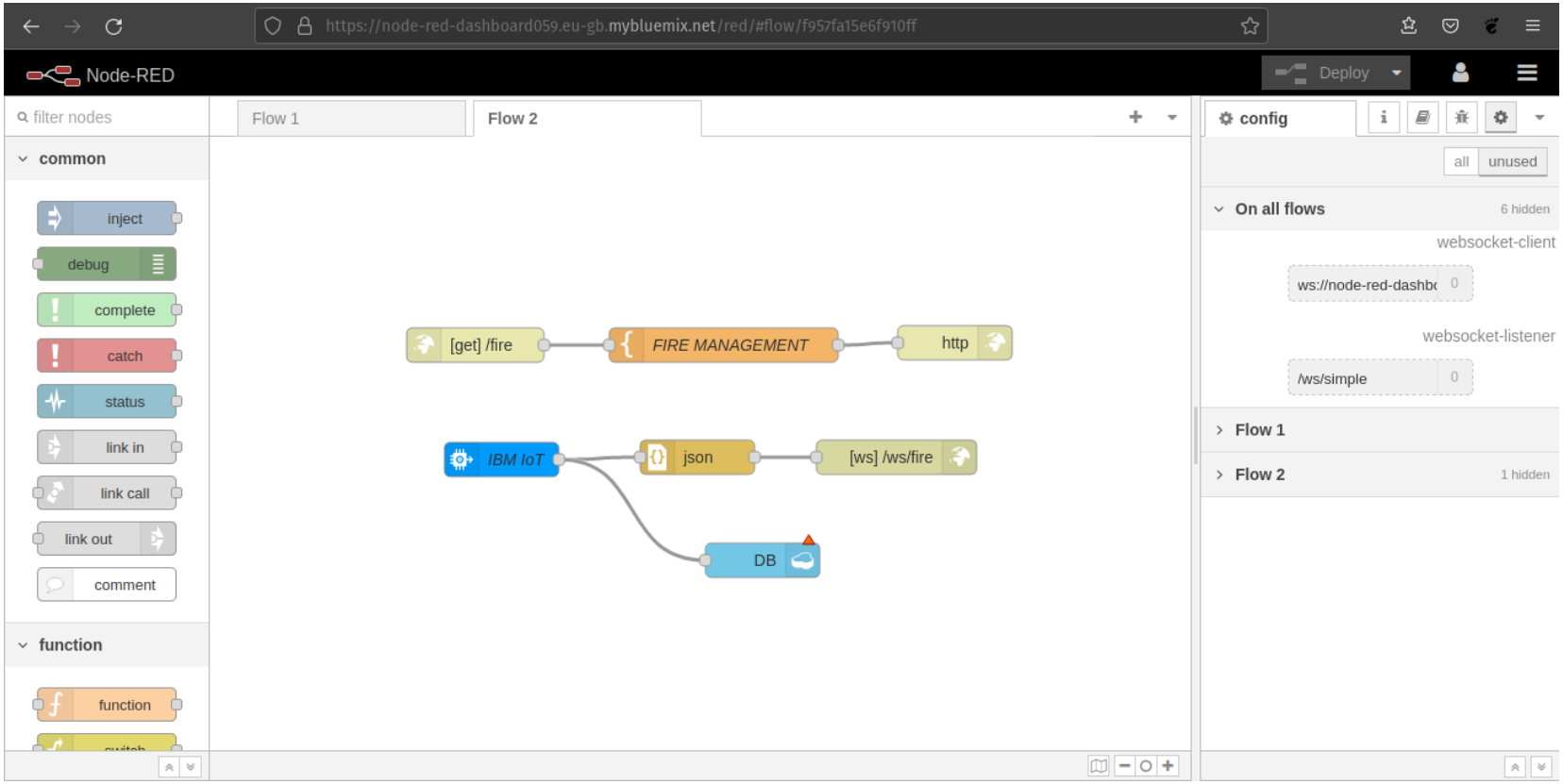
TRANSFERRING DATA FROM IBM WATSON INTO NODE-RED



NODE DASHBOARD



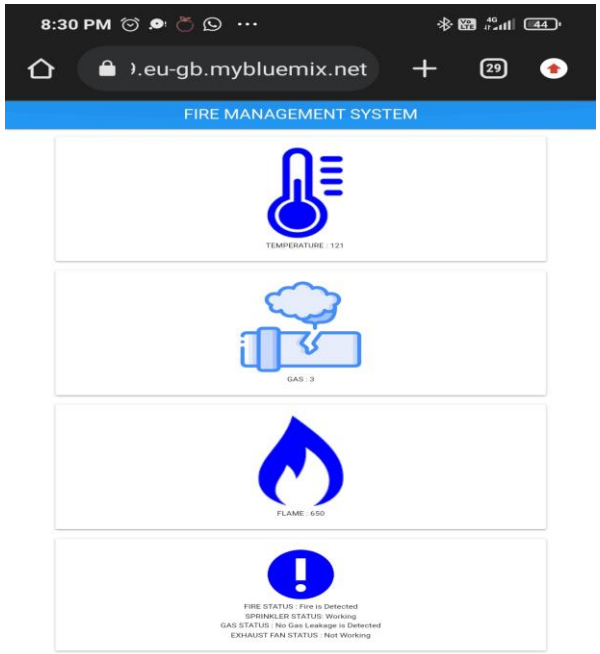
TRANSFERRING DATA FROM NODE-RED INTO WEB UI



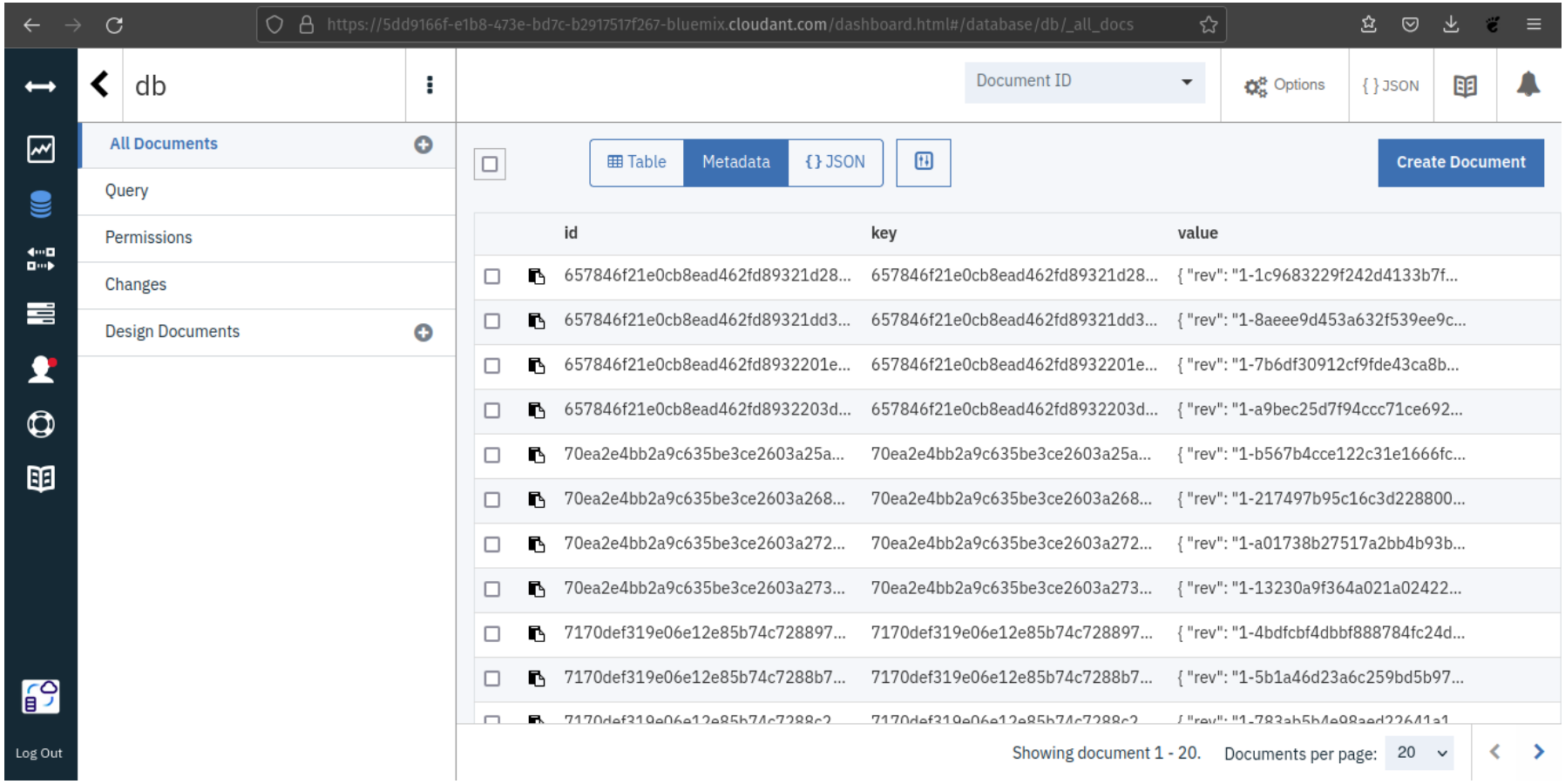
WEB UI

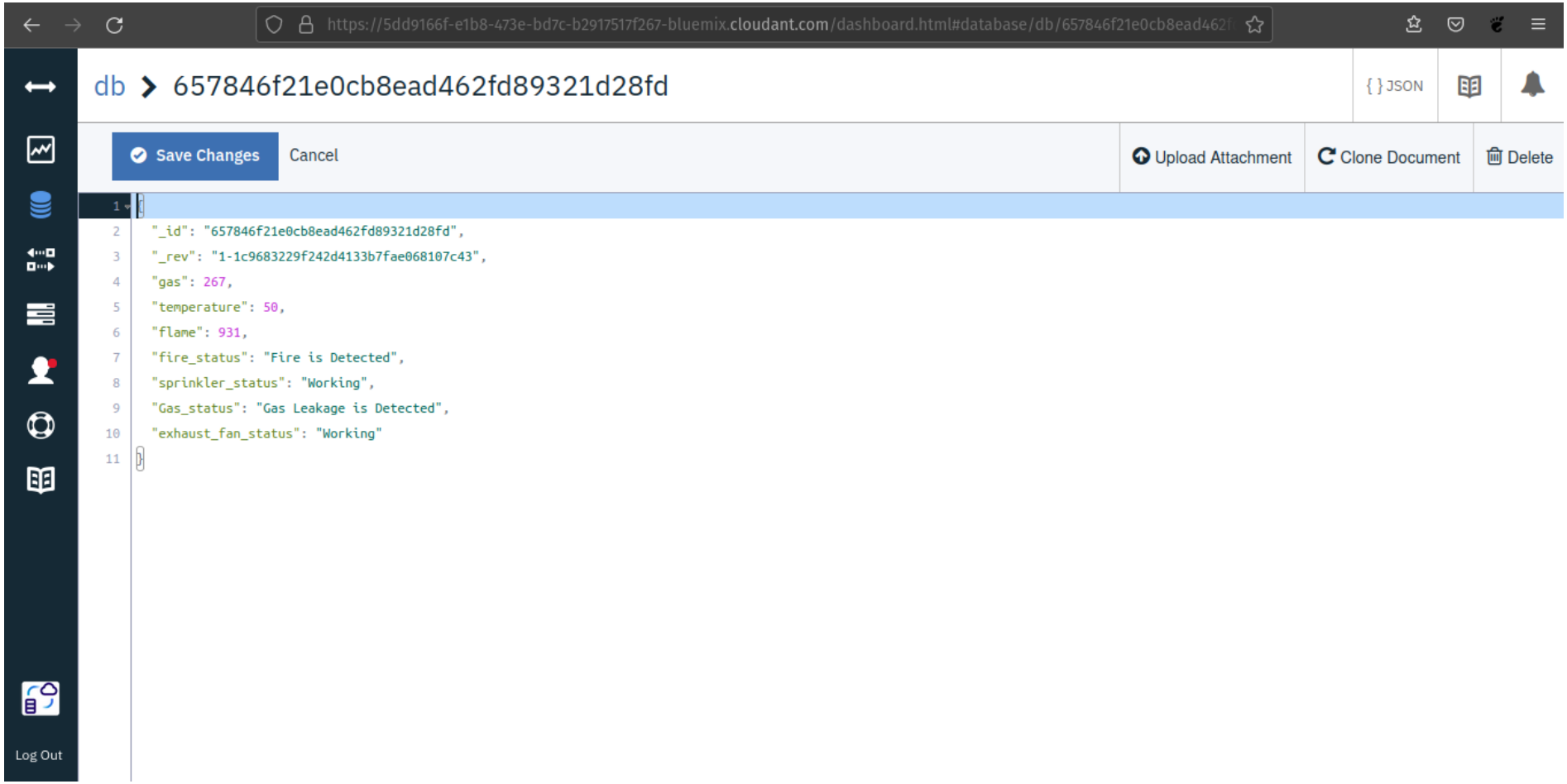
DESKTOP VIEW

MOBILE VIEW


8:30 PM
).eu-gb.mybluemix.net
29
FIRE MANAGEMENT SYSTEM
TEMPERATURE : 121
GAS : 3
FLAME : 650
FIRE STATUS : Fire is Detected
SPRINKLER STATUS: Working
GAS STATUS : No Gas Leakage is Detected
EXHAUST FAN STATUS : Not Working

CLOUDANT:


https://5dd9166f-e1b8-473e-bd7c-b2917517f267-bluemix.cloudant.com/dashboard.html#/database/db/_all_docs
db
All Documents
Query
Permissions
Changes
Design Documents
Log Out
Document ID
Options
{ } JSON
Table
Metadata
{ } JSON
Create Document
id
key
value
657846f21e0cb8ead462fd89321d28...
657846f21e0cb8ead462fd89321d28...
{ "rev": "1-1c9683229f242d4133b7f...
657846f21e0cb8ead462fd89321dd3...
657846f21e0cb8ead462fd89321dd3...
{ "rev": "1-8aeee9d453a632f539ee9c...
657846f21e0cb8ead462fd8932201e...
657846f21e0cb8ead462fd8932201e...
{ "rev": "1-7b6df30912cf9fde43ca8b...
657846f21e0cb8ead462fd8932203d...
657846f21e0cb8ead462fd8932203d...
{ "rev": "1-a9bec25d7f94ccc71ce692...
70ea2e4bb2a9c635be3ce2603a25a...
70ea2e4bb2a9c635be3ce2603a25a...
{ "rev": "1-b567b4cce122c31e1666fc...
70ea2e4bb2a9c635be3ce2603a268...
70ea2e4bb2a9c635be3ce2603a268...
{ "rev": "1-217497b95c16c3d228800...
70ea2e4bb2a9c635be3ce2603a272...
70ea2e4bb2a9c635be3ce2603a272...
{ "rev": "1-a01738b27517a2bb4b93b...
70ea2e4bb2a9c635be3ce2603a273...
70ea2e4bb2a9c635be3ce2603a273...
{ "rev": "1-13230a9f364a021a02422...
7170def319e06e12e85b74c728897...
7170def319e06e12e85b74c728897...
{ "rev": "1-4bdfcbf4dbbf888784fc24d...
7170def319e06e12e85b74c7288b7...
7170def319e06e12e85b74c7288b7...
{ "rev": "1-5b1a46d23a6c259bd5b97...
Showing document 1 - 20.
Documents per page:
20


https://5dd9166f-e1b8-473e-bd7c-b2917517f267-bluemix.cloudant.com/dashboard.html#database/db/657846f21e0cb8ead462f
db
657846f21e0cb8ead462fd89321d28fd
{ } JSON
Save Changes
Cancel
Upload Attachment
Clone Document
Delete
{
"_id": "657846f21e0cb8ead462fd89321d28fd",
"_rev": "1-1c9683229f242d4133b7fae068107c43",
"gas": 267,
"temperature": 50,
"flame": 931,
"fire_status": "Fire is Detected",
"sprinkler_status": "Working",
"Gas_status": "Gas Leakage is Detected",
"exhaust_fan_status": "Working"
}
Log Out

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";
char authMethod[] = "use-token-auth";                    char
token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;

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();
  }
}
```