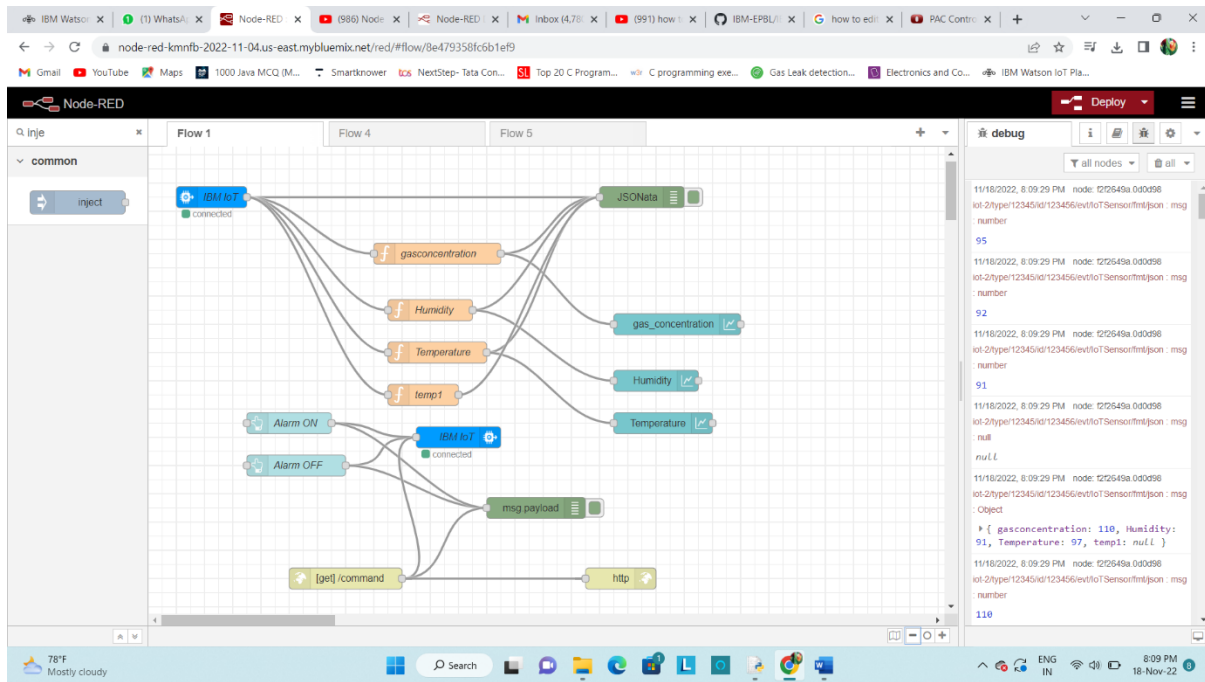


PROJECT DEVELOPMENT PHASE

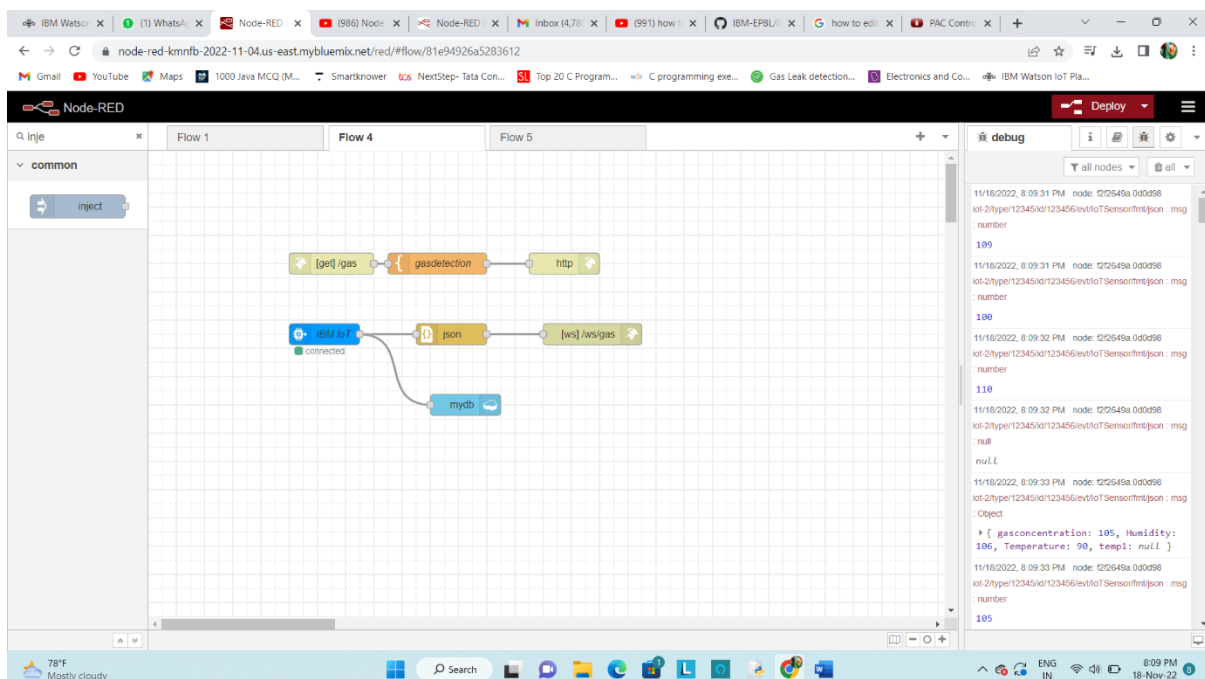
SPRINT 3

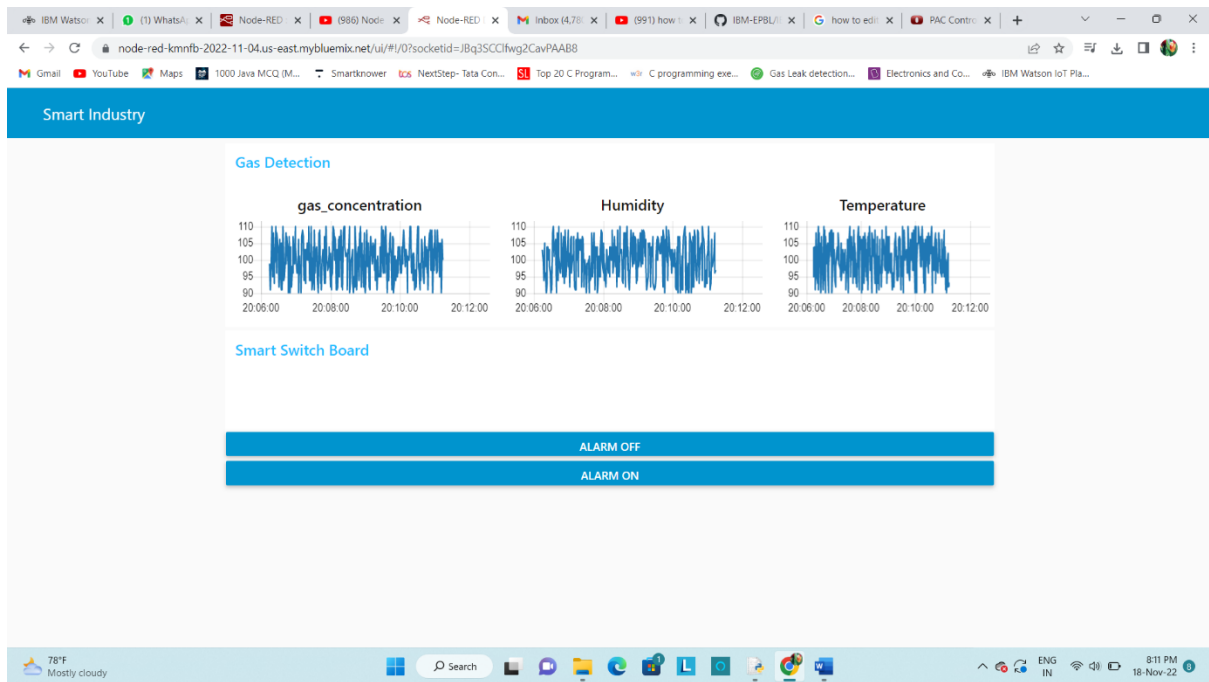
Date	18 November 2022
Team ID	PNT2022TMID13514
Project name	Gas Leakage Monitoring & Alerting System for Industries

NODE RED FLOW

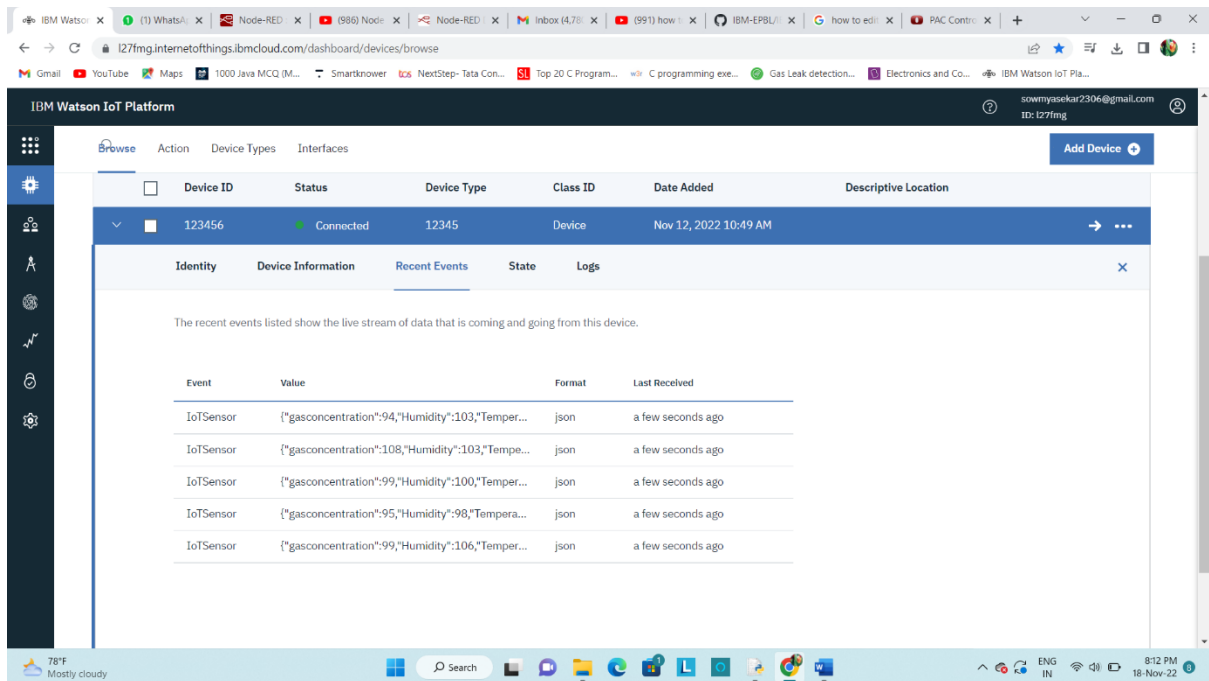


CLOUDANT CONNECTION IN NODE-RED





RECEIVE OF MESSAGE FROM WATSON:

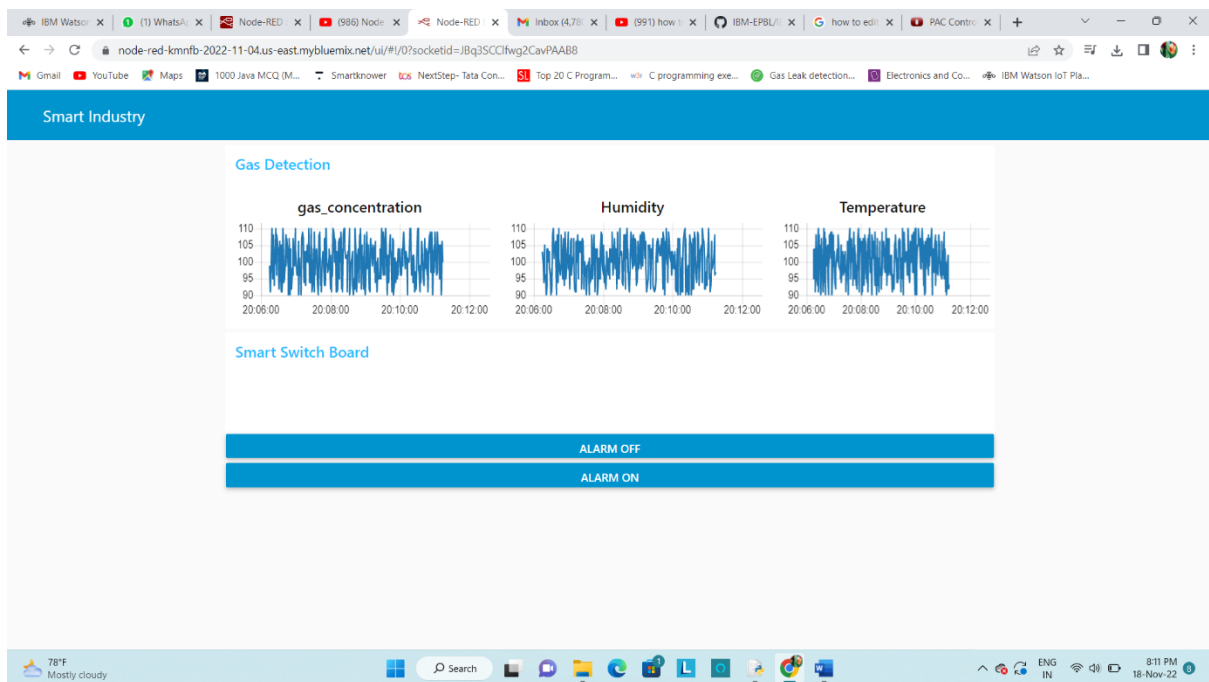


PYTHON CODE OUTPUT:

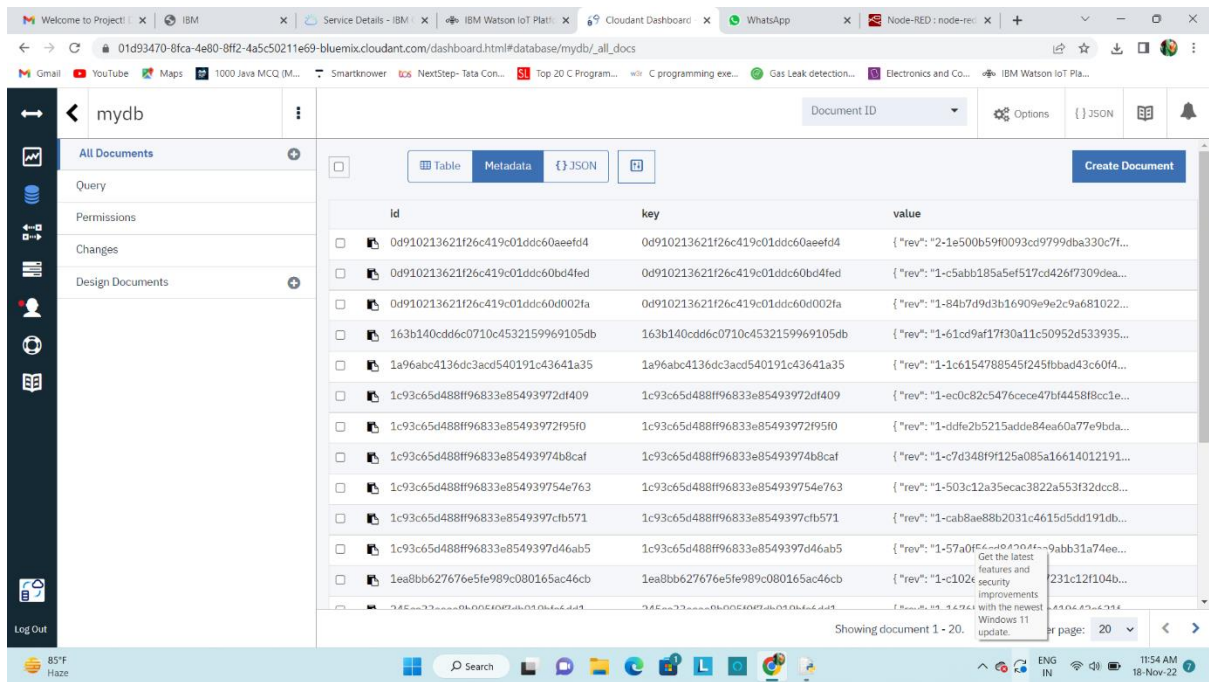
```
ibmpython.py - C:\Users\USER\Desktop\ibm\ibmpython.py (3.7.0)
File Edit Format Run Options Window Help
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
organization = "127fmg"
deviceType = "12345"
deviceId = "123456"
authMethod = "token"
authToken = "123456789"
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="lighton":
        print ("led is on")
    elif status=="lightoff":
        print ("led is off")
    else:
        print ("Please send proper command")
try:
    deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method": authMethod}
    deviceCli = ibmiotf.device.Client(deviceOptions)
except Exception as e:
    print("Caught exception connecting device %s" % str(e))
    sys.exit()
deviceCli.connect()
while True:
    gasconcentration = random.randint(90,110)
    Humidity = random.randint(90,110)
    Temperature = random.randint(90,110)
    data = {'gasconcentration': gasconcentration, 'Humidity': Humidity, 'Temperature': Temperature}
    def myOnPublishCallback():
        print(" GasConcentration = %s PPM" % gasconcentration, "to IBM Watson")
        print(" Humidity = %s%" % Humidity, "to IBM Watson")
        print(" Temperature = %s C" % Temperature, "to IBM Watson")
    success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=myOnPublishCallback)
    if not success:
        print("Not connected to IoT")
    time.sleep(10)
deviceCli.commandCallback=myCommandCallback
deviceCli.disconnect()

Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
gasconcentration = 100 PPM to IBM Watson
Humidity = 91% to IBM Watson
Temperature = 100 C to IBM Watson
===== RESTART: C:\Users\USER\Desktop\ibm\ibmpython.py =====
2022-11-16 22:09:57,891 ibmiotf.device.Client INFO Connected successfully: d:127fmg:12345:123456
GasConcentration = 110 PPM to IBM Watson
Humidity = 100% to IBM Watson
Temperature = 105 C to IBM Watson
GasConcentration = 98 PPM to IBM Watson
Humidity = 105% to IBM Watson
Temperature = 93 C to IBM Watson
GasConcentration = 107 PPM to IBM Watson
Humidity = 109% to IBM Watson
Temperature = 105 C to IBM Watson
GasConcentration = 107 PPM to IBM Watson
Humidity = 98% to IBM Watson
Temperature = 107 C to IBM Watson
GasConcentration = 103 PPM to IBM Watson
Humidity = 109% to IBM Watson
Temperature = 103 C to IBM Watson
GasConcentration = 102 PPM to IBM Watson
Humidity = 105% to IBM Watson
Temperature = 104 C to IBM Watson
GasConcentration = 107 PPM to IBM Watson
Humidity = 105% to IBM Watson
Temperature = 93 C to IBM Watson
GasConcentration = 108 PPM to IBM Watson
Humidity = 103% to IBM Watson
Temperature = 108 C to IBM Watson
GasConcentration = 100 PPM to IBM Watson
Humidity = 92% to IBM Watson
Temperature = 100 C to IBM Watson
GasConcentration = 90 PPM to IBM Watson
Humidity = 99% to IBM Watson
Temperature = 103 C to IBM Watson
GasConcentration = 107 PPM to IBM Watson
Humidity = 107% to IBM Watson
Temperature = 108 C to IBM Watson
GasConcentration = 97 PPM to IBM Watson
Humidity = 96% to IBM Watson
Temperature = 110 C to IBM Watson
GasConcentration = 92 PPM to IBM Watson
Humidity = 105% to IBM Watson
Temperature = 94 C to IBM Watson
```

WEB UI:



CLOUDANT:



PYTHON CODE:

```
import time

import sys

import ibmiotf.application

import ibmiotf.device

import random

organization = "I27fmg"

deviceType = "12345"

deviceId = "123456"

authMethod = "token"

authToken = "123456789"

def myCommandCallback(cmd):

    print("Command received: %s" % cmd.data['command'])

    status=cmd.data['command']

    if status=="alarmon":

        print ("Alarm is on")

    elif status == "alarmoff":
```

```

        print ("Alarm is off")

    else:

        print ("Please send proper command")

try:

    deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-
method": authMethod, "auth-token" : authToken}

    deviceCli = ibmiotf.device.Client(deviceOptions)

except Exception as e:

    print("Caught exception connecting device %s" % str(e))

    sys.exit()

deviceCli.connect()

while True:

    gasconcentration = random.randint(90,110)

    Humidity = random.randint(90,110)

    Temperature = random.randint(90,110)

    data = {'gasconcentration' : gasconcentration, 'Humidity' : Humidity, 'Temperature'
:Temperature}

    def myOnPublishCallback():

        print(" GasConcentration = %s PPM" % gasconcentration, "to IBM Watson")

        print(" Humidity = %s%%" % Humidity, "to IBM Watson")

        print(" Temperature = %s C" % Temperature, "to IBM Watson")

        success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,
on_publish=myOnPublishCallback)

        if not success:

            print("Not connected to IoTF")

            time.sleep(10)

deviceCli.commandCallback=myCommandCallback

```

```
deviceCli.disconnect()
```