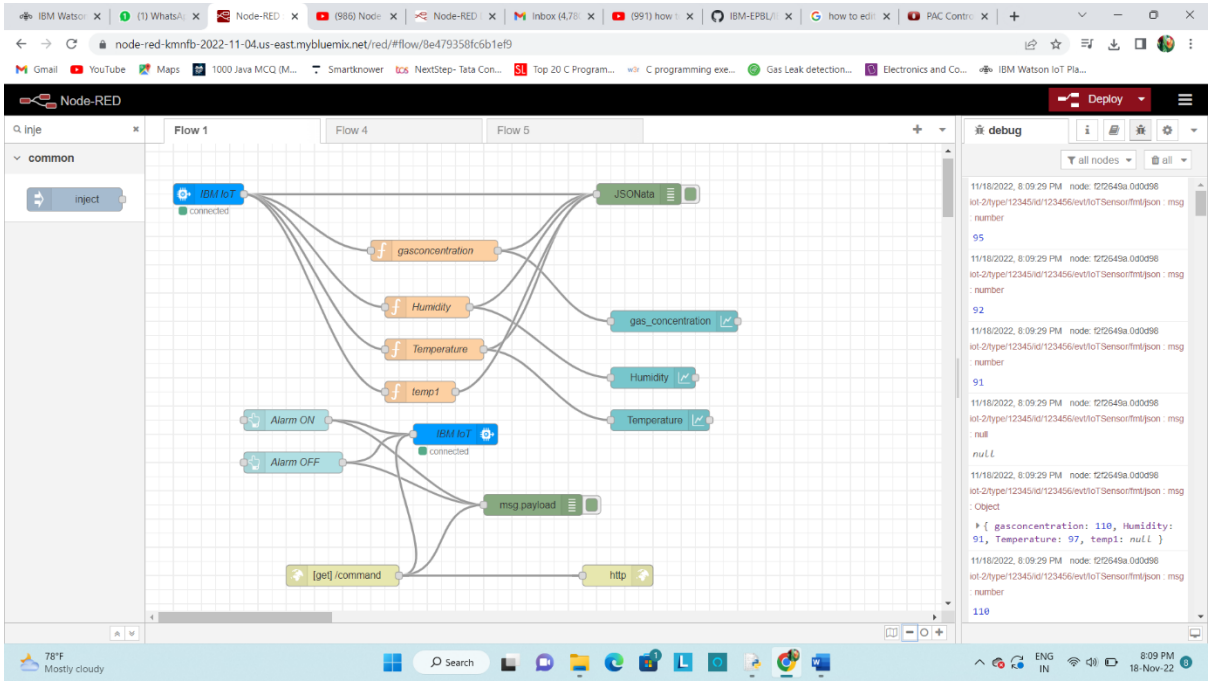
PROJECT DEVELOPMENT PHASE

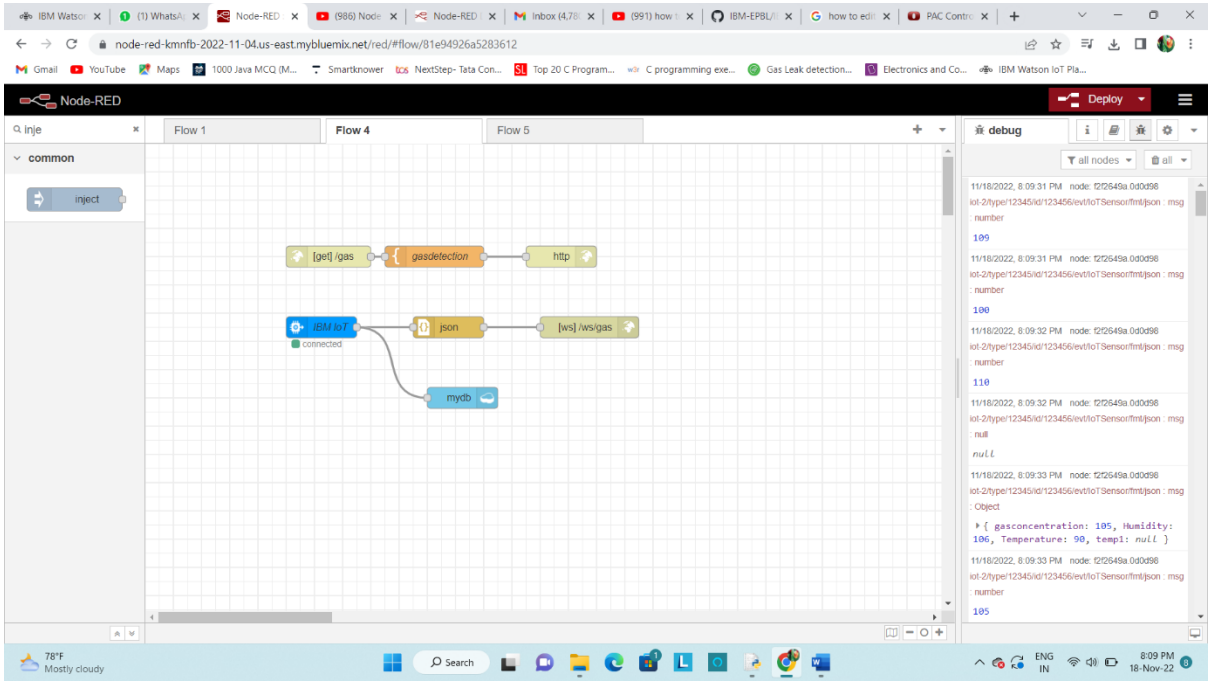
SPRINT 3

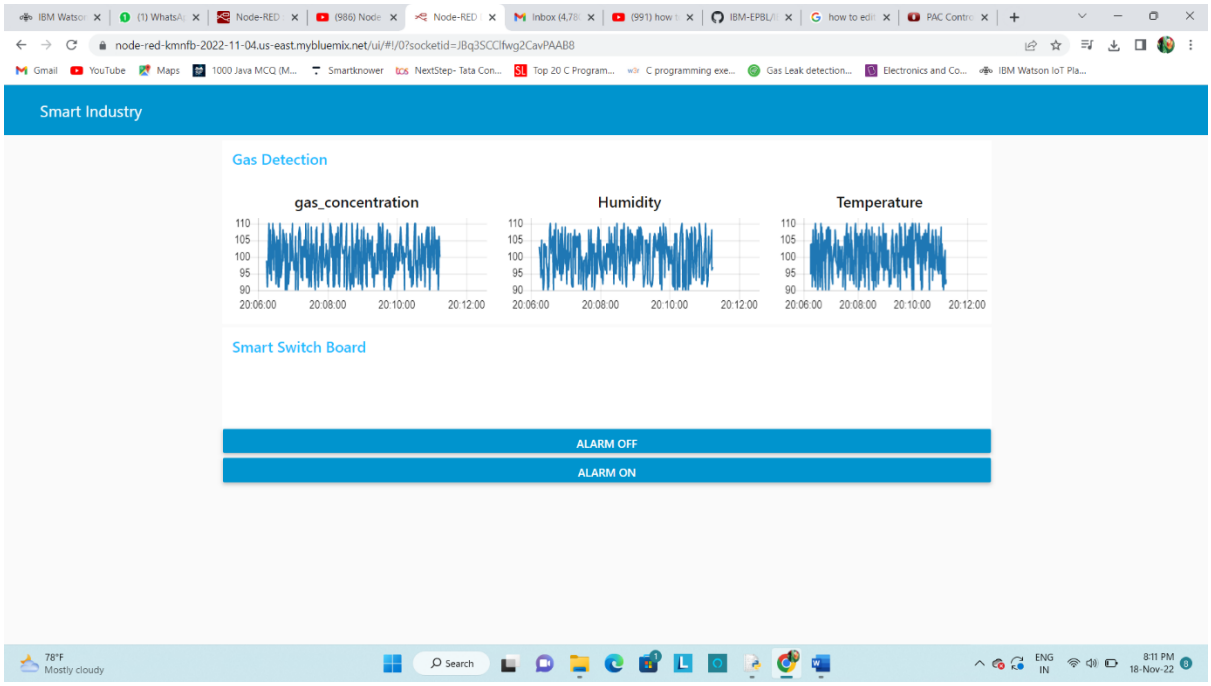
Date	18 November 2022
Team ID	PNT2022TMID13514
Project name	Gas Leakage Monitoring & Alerting System for Industries

NODE RED FLOW

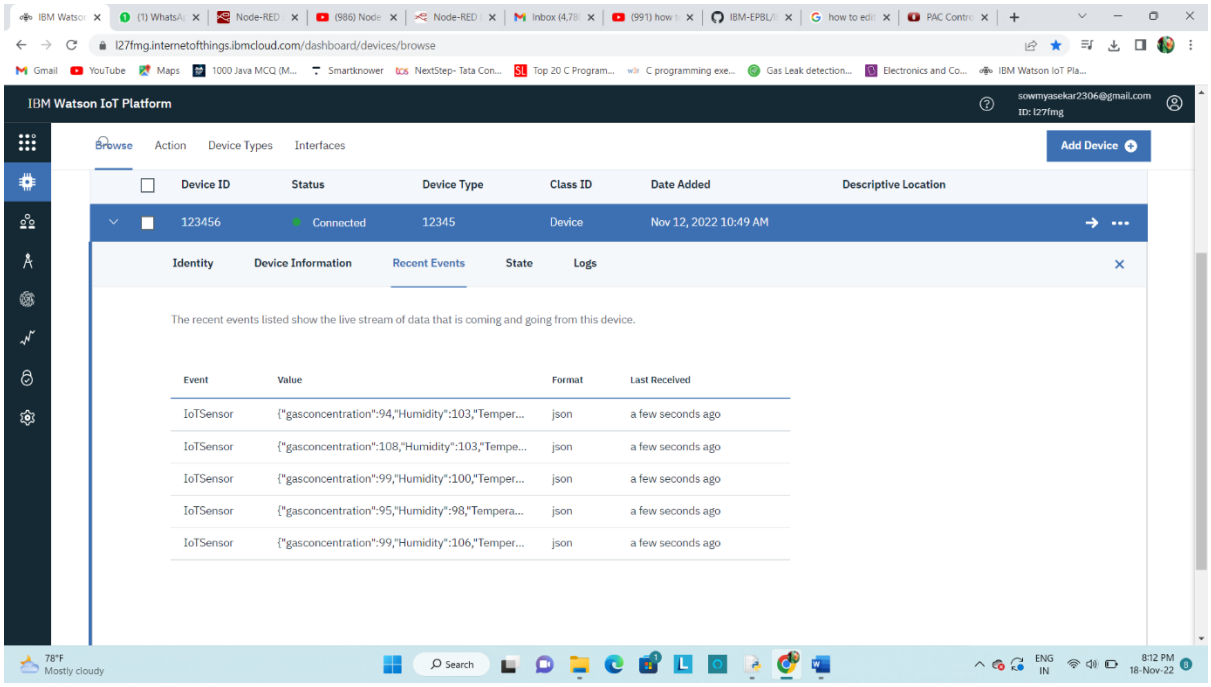


CLOUDANT CONNECTION IN NODE-RED

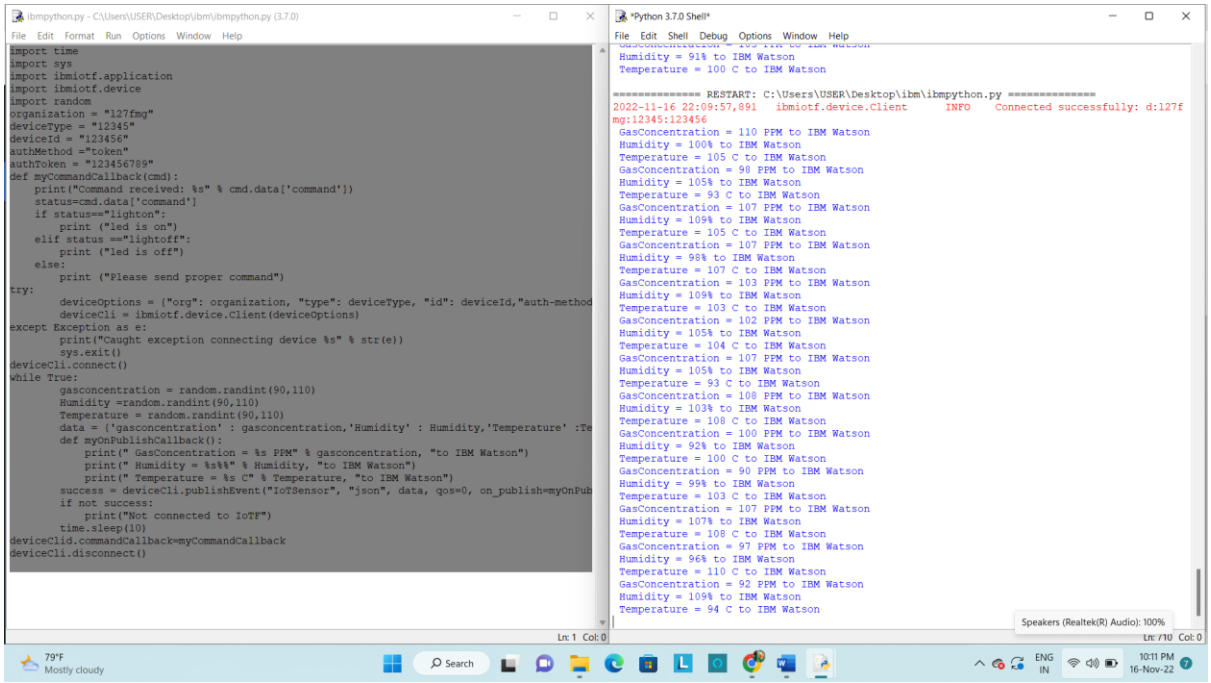




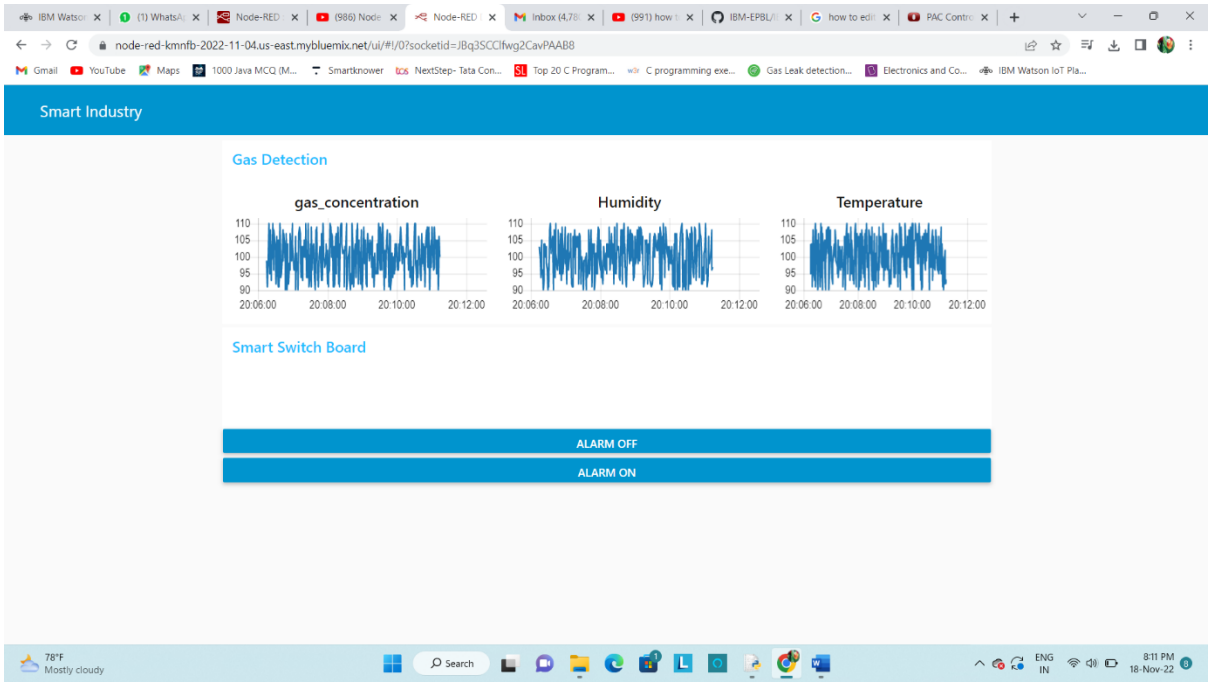
RECEIVE OF MESSAGE FROM WATSON:



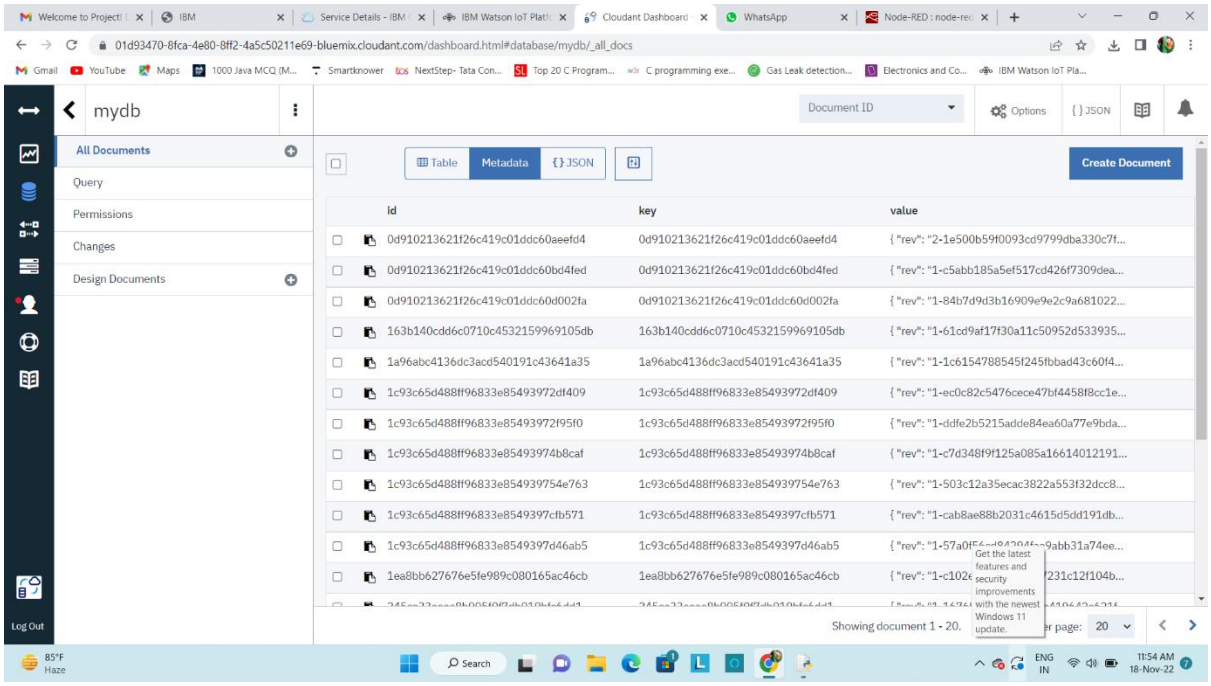
PYTHON CODE OUTPUT:



WEB UI:



CLOUDANT:



PYTHON CODE:

```
import time

import sys

import ibmiotf.application

import ibmiotf.device

import random

organization = "l27fmg"

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()
```