Develop The Python Code

Team ID	PNT2022TMID43411
Team Members	JOTHI KRISHNA T - 715519106018 KARTHIKEYAN A - 715519106020 NITHIYANANTH S - 715519106031 VIPIN L - 715519106059
Project Title	Gas Leakage Monitoring And Alerting System For Industries

Python Code:

```
import ibmiotf.application

import ibmiotf.device

import time

import random

import sys


#ibm watson device credentials


organization="griwxv"

deviceType="ESP32"

deviceid="12345678"

authMethod="token"

authToken="12345678"


#generate random values for gas leakage



def myCommandCallback(cmd):
```

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

#connect and sending data for gas leakage

deviceCli.connect()

while True:
    Gas=random.randint(0,100)
    Temp=random.randint(0,100)
    Hum=random.randint(0,100)
    Fire=random.randint(0,100)
    data={'Gas':Gas,'Temp':Temp,'Hum':Hum,'Fire':Fire}
    print(data)
    def myOnPublishCallBack():
        print("published Gas  %s " %Gas)
        print("published Temp %s " %Temp)
        print("published Hum %s " %Hum)
        print("published Fire %s " %Fire)
    success=deviceCli.publishEvent("IoTSensor","json",data,qos=0,on_publish=myOnPublishCallBack)
    if not success:
        print ("Not connected to IoTF")
```

```
    time.sleep(1)

    deviceCli.commandCallback=myCommandCallback


#disconnect the device from the cloud

deviceCli.connect()
```

OUTPUT: