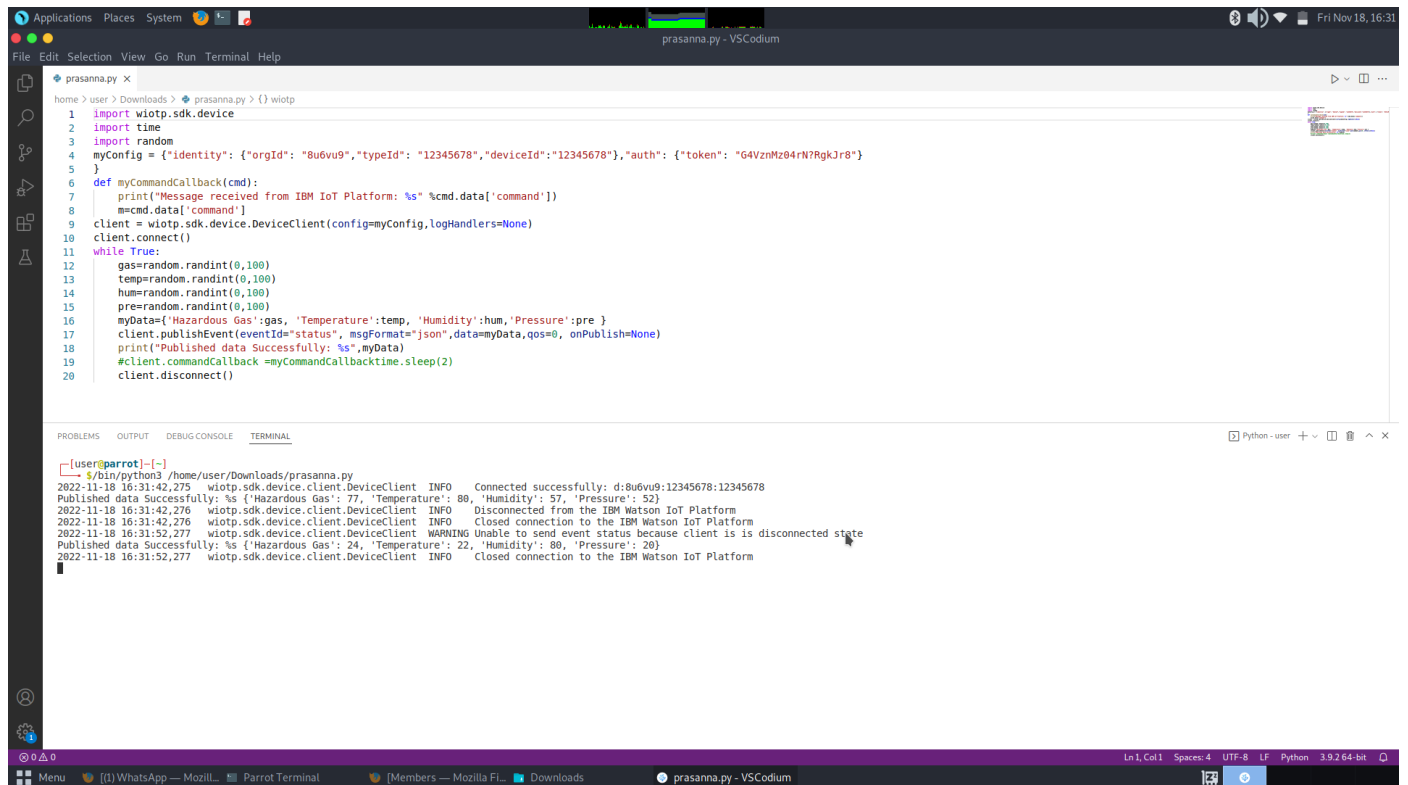


## PYTHON CODE

<b>Date</b>	19 NOVEMBER 2022
<b>Team ID</b>	PNT2022TMID11052
<b>Project Name</b>	GAS LEAKAGE MONITORING AND ALERTING SYSTEM FOR INDUSTRIES

```
import wiotp.sdk.device
import time
import random
myConfig = {"identity": {"orgId": "8u6vu9","typeId":
"12345678","deviceId":"12345678"},"auth": {"token":
"G4VznMz04rN?RgkJr8"}}
}
def myCommandCallback(cmd):
    print("Message received from IBM IoT Platform: %s"
%cmd.data['command'])
    m=cmd.data['command']
client =
wiotp.sdk.device.DeviceClient(config=myConfig,logHandlers=None)
client.connect()
while True:
    gas=random.randint(0,100)
    temp=random.randint(0,100)
    hum=random.randint(0,100)
    pre=random.randint(0,100)
    myData={'Hazardous Gas':gas, 'Temperature':temp,
'Humidity':hum,'Pressure':pre }
    client.publishEvent(eventId="status",
msgFormat="json",data=myData,qos=0, onPublish=None)
    print("Published data Successfully: %s",myData)
    #client.commandCallback =myCommandCallbacktime.sleep(2)
    client.disconnect()
```

# Python Output:-



The screenshot displays a VS Code editor window with a Python script named `prasanna.py` and its terminal output. The script is a loop that generates random data and publishes it to the IBM Watson IoT Platform. The terminal output shows the successful connection, data publishing, and subsequent disconnection of the client.

```
1 import wiotp.sdk.device
2 import time
3 import random
4 myConfig = {'identity': {'orgId': '8u6vu9', 'typeId': '12345678', 'deviceId': '12345678'}, 'auth': {'token': 'G4VznMz04rN7RgkJr8'}}
5 }
6 def myCommandCallback(cmd):
7     print("Message received from IBM IoT Platform: %s" %cmd.data['command'])
8     m=cmd.data['command']
9 client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
10 client.connect()
11 while True:
12     gas=random.randint(0,100)
13     temp=random.randint(0,100)
14     hum=random.randint(0,100)
15     pre=random.randint(0,100)
16     myData={'Hazardous Gas':gas, 'Temperature':temp, 'Humidity':hum, 'Pressure':pre }
17     client.publishEvent(eventId='status', msgFormat='json', data=myData, qos=0, onPublish=None)
18     print("Published data Successfully: %s" %myData)
19     #client.commandCallback =myCommandCallbacktime.sleep(2)
20     client.disconnect()
```

Terminal Output:

```
[user@parrot]-[~]
$ ./bin/python3 /home/user/Downloads/prasanna.py
2022-11-18 16:31:42,275 wiotp.sdk.device.client.DeviceClient INFO Connected successfully: d:8u6vu9:12345678:12345678
Published data Successfully: %s {'Hazardous Gas': 77, 'Temperature': 80, 'Humidity': 57, 'Pressure': 52}
2022-11-18 16:31:42,276 wiotp.sdk.device.client.DeviceClient INFO Disconnected from the IBM Watson IoT Platform
2022-11-18 16:31:42,276 wiotp.sdk.device.client.DeviceClient INFO Closed connection to the IBM Watson IoT Platform
2022-11-18 16:31:52,277 wiotp.sdk.device.client.DeviceClient WARNING Unable to send event status because client is in disconnected state
Published data Successfully: %s {'Hazardous Gas': 24, 'Temperature': 22, 'Humidity': 80, 'Pressure': 20}
2022-11-18 16:31:52,277 wiotp.sdk.device.client.DeviceClient INFO Closed connection to the IBM Watson IoT Platform
```