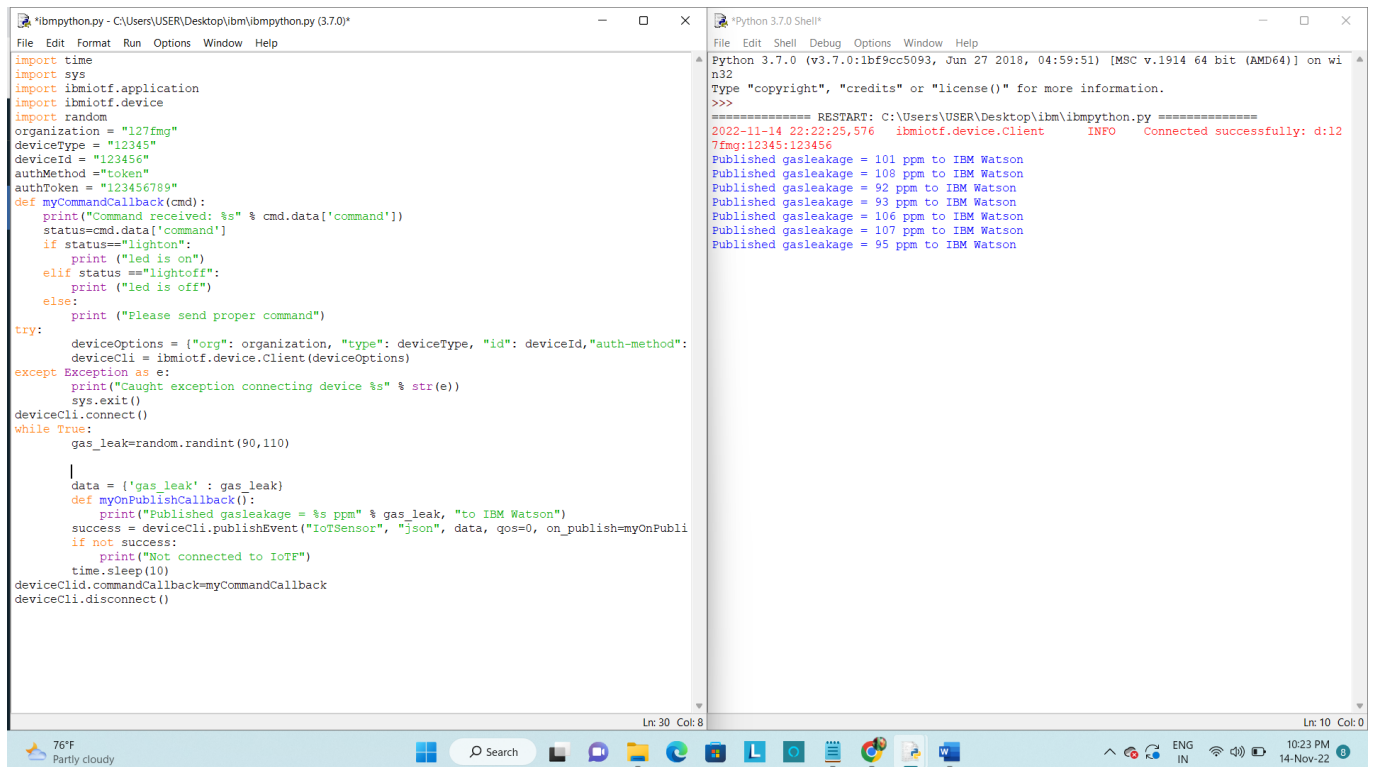


## PUBLISH DATA TO PYTHON CODE

Date	14 November 2022
Team ID	PNT2022TMID13514
Project Name	Gas Leakage Monitoring & Alerting System for Industries
Maximum Marks	4 Marks

### Gas Leakage Monitoring & Alerting System for Industries

Python code :



The image shows a screenshot of a Windows desktop with two windows open. The left window is a text editor titled "ibmpython.py" showing a Python script. The script imports modules like time, sys, random, and ibmiotf. It defines a myCommandCallback function and a while loop that generates random gas leak data and publishes it to IBM Watson IoT. The right window is a "Python 3.7.0 Shell" showing the execution output. It displays a restart message, a successful connection to the IoT device, and a list of published gas leak data points.

```
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":
    deviceCli = ibmiotf.device.Client(deviceOptions)
except Exception as e:
    print("Caught exception connecting device %s" % str(e))
    sys.exit()
deviceCli.connect()
while True:
    gas_leak=random.randint(90,110)
    |
    data = {'gas_leak' : gas_leak}
    def myOnPublishCallback():
        print("Published gasleakage = %s ppm" % gas_leak, "to IBM Watson")
    success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=myOnPubli
    if not success:
        print("Not connected to IoT")
    time.sleep(10)
deviceCli.commandCallback=myCommandCallback
deviceCli.disconnect()
```

```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on wi
n32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\USER\Desktop\ibm\ibmpython.py =====
2022-11-14 22:22:25,576 ibmiotf.device.Client INFO Connected successfully: d:12
7fmg:12345:123456
Published gasleakage = 101 ppm to IBM Watson
Published gasleakage = 108 ppm to IBM Watson
Published gasleakage = 92 ppm to IBM Watson
Published gasleakage = 93 ppm to IBM Watson
Published gasleakage = 106 ppm to IBM Watson
Published gasleakage = 107 ppm to IBM Watson
Published gasleakage = 95 ppm to IBM Watson
```

# IBM Watson Connection :

WhatsApp

IBM-EPBL/IBM-Project-28

IBM

Welcome to Project! Deligh

IBM-Project-2963-165848

IBM Watson IoT Platform

l27fmg.internetofthings.ibmcloud.com/dashboard/devices/browse

Gmail

YouTube

Maps

1000 Java MCQ (M...

Smartknowr

NextStep- Tata Con...

Top 20 C Program...

C programming eve...

Gas Leak detection...

Electronics and Co...

IBM Watson IoT Pla...

IBM Watson IoT Platform

sowmyasekar2306@gmail.com

ID: l27fmg

Browse

Action

Device Types

Interfaces

Add Device

Identity

Device Information

Recent Events

State

Logs

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
IoTSensor	{"gas_leak":106}	json	a few seconds ago
IoTSensor	{"gas_leak":93}	json	a few seconds ago
IoTSensor	{"gas_leak":92}	json	a few seconds ago
IoTSensor	{"gas_leak":108}	json	a few seconds ago

Items per page 50 | 1-2 of 2 items

1 of 1 page

Monday, November 14, 2022

76°F  
Partly cloudy

Search

ENG  
IN

10:23 PM  
14-Nov-22



