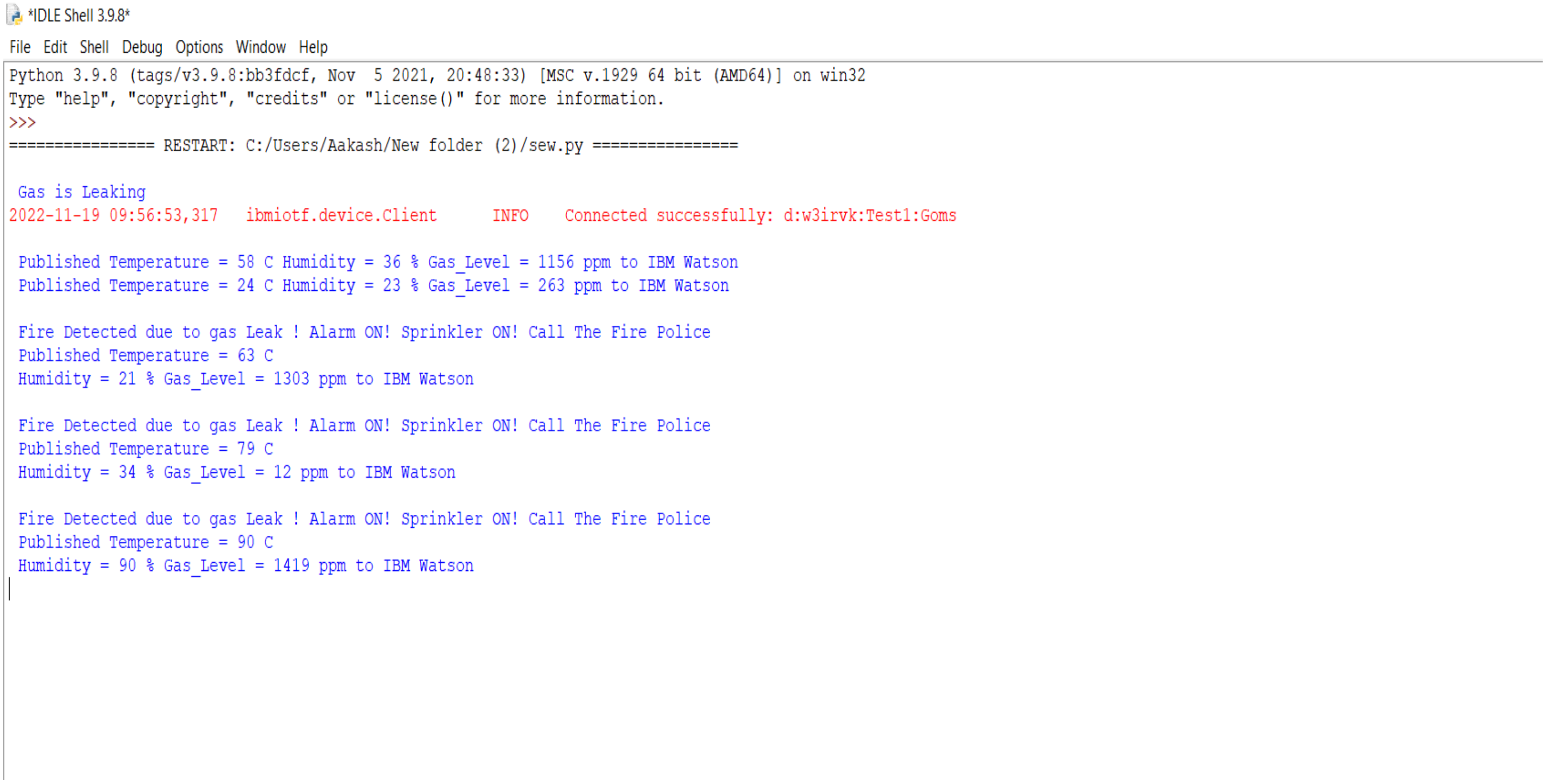
SPRINT 3 :Creating a successful python progam to link with IoT device .

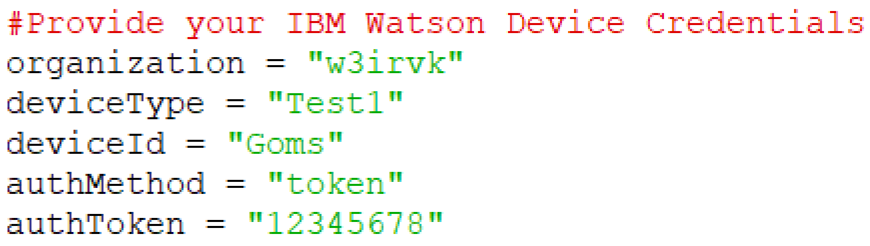
Team ID	PNT2022TMID51059
Project Title	**Gas Leakage Monitoring & Alerting System For Industries**
Date	07 Nov,2022

The Code is given below :

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random


#Provide your IBM Watson Device Credentials
organization = "w3irvk"
deviceType = "Test1"
deviceId = "Goms"
authMethod = "token"
authToken = "12345678"

# Initialize GPIO


def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status == "alarmon":
        print ("Alarm is on please all Evacuate Fans On")
    elif status == "alarmoff":
        print ("Alarm is off and Fans Off")
    elif status == "sprinkleron":
        print ("Sprinkler is On Evacuate Faster")
    elif status == "sprinkleroff":
        print("Sprinkler is Off")
    else:
        print("Please send proper 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 send a datapoint "hello" with value "world" into the cloud as an event of type "greeting" 10 times
deviceCli.connect()

while True:
        #Get Sensor Data from random function

        temp=random.randint(0,120)
        Humid=random.randint(0,100)
        gas=random.randint(0,1500)
        data={'temp':temp,'Humid':Humid,'gas':gas}
        #print data
        def myOnPublishCallback():
            print (" Published Temperature = %s C" % temp, "Humidity = %s %%" % Humid, "Gas_Level = %s ppm" %gas, "to IBM Watson")

        success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=myOnPublishCallback)
        if not success:
            print("\n Not connected to IoTF")
        if temp>60 :
            print("\n Fire Detected due to gas Leak ! Alarm ON! Sprinkler ON! Call The Fire Police \n")
        elif gas>350:
            print("\n Gas is Leaking \n")
        time.sleep(10)

        deviceCli.commandCallback = myCommandCallback

# Disconnect the device and application from the cloud
deviceCli.disconnect()
```



```
Python 3.9.8 (tags/v3.9.8:bb3fdcf, Nov  5 2021, 20:48:33) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
================ RESTART: C:/Users/Aakash/New folder (2)/sew.py ================

 Gas is Leaking
2022-11-19 09:56:53,317   ibmiotf.device.Client      INFO    Connected successfully: d:w3irvk:Test1:Goms

 Published Temperature = 58 C Humidity = 36 % Gas_Level = 1156 ppm to IBM Watson
 Published Temperature = 24 C Humidity = 23 % Gas_Level = 263 ppm to IBM Watson

 Fire Detected due to gas Leak ! Alarm ON! Sprinkler ON! Call The Fire Police
 Published Temperature = 63 C
 Humidity = 21 % Gas_Level = 1303 ppm to IBM Watson

 Fire Detected due to gas Leak ! Alarm ON! Sprinkler ON! Call The Fire Police
 Published Temperature = 79 C
 Humidity = 34 % Gas_Level = 12 ppm to IBM Watson

 Fire Detected due to gas Leak ! Alarm ON! Sprinkler ON! Call The Fire Police
 Published Temperature = 90 C
 Humidity = 90 % Gas_Level = 1419 ppm to IBM Watson
```



```
#Provide your IBM Watson Device Credentials
organization = "w3irvk"
deviceType = "Test1"
deviceId = "Goms"
authMethod = "token"
authToken = "12345678"
```

The above image is our unique device credentials for connecting the python program with the IoT device successfully.

This is the expected output.

The data has been gathered and displayed successfully.