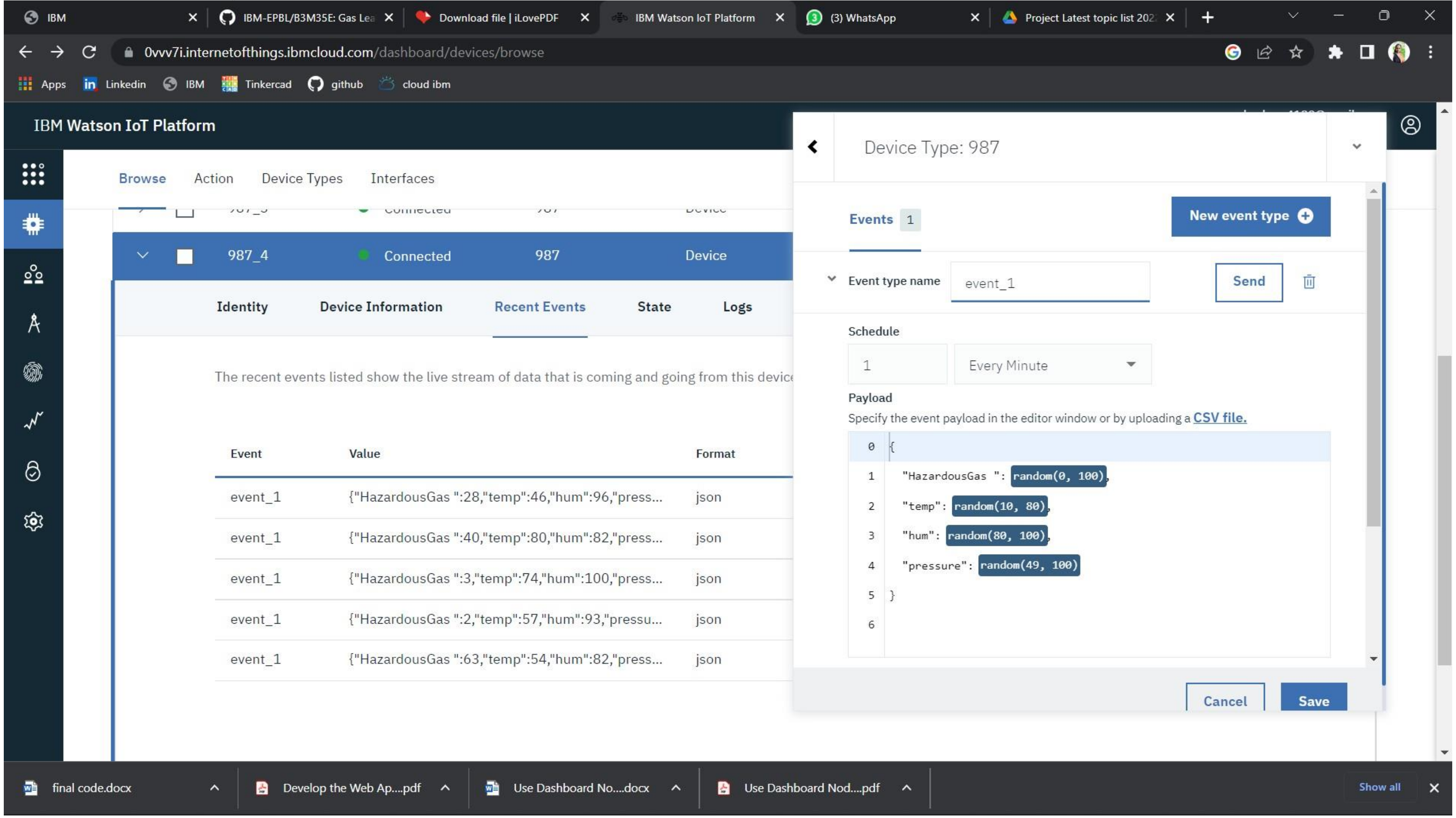
DELIVERY OF SPRINT-3

Date	02 November 2022
Team ID	PNT2022TMID13412
Project Name	Gas Leakage Monitoring and Alerting System

Code:

```
import sys
import ibmiotf.application
import ibmiotf.device
import random
organization = "0vvv7i"
deviceType = "12345"
deviceId = "12"
authMethod ="token"
authToken = "12345678"
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": authMethod, "auth-token" :authToken}
        deviceCli = ibmiotf.device.Client(deviceOptions)
except Exceptions as e:
        print("Caught exception connecting device %s" % str(e))
        sys.exit();

deviceCli.connect()
while True:
        temp=random.randint(90,110)
        Humid=random.randint(60,100)
        data = {'temp' : temp,'Humid' :Humid}
        def myOnPublishCallback():
            print("Published Temperature = %s C" % temp, "Humidity =%s %%" % Humid, "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)
deviceClid.commandCallback=myCommandCallback
deviceCli.disconnect()
```



987_4 Connected 987 Device

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

Event	Value	Format
event_1	{"HazardousGas ":28,"temp":46,"hum":96,"press...	json
event_1	{"HazardousGas ":40,"temp":80,"hum":82,"press...	json
event_1	{"HazardousGas ":3,"temp":74,"hum":100,"press...	json
event_1	{"HazardousGas ":2,"temp":57,"hum":93,"pressu...	json
event_1	{"HazardousGas ":63,"temp":54,"hum":82,"press...	json

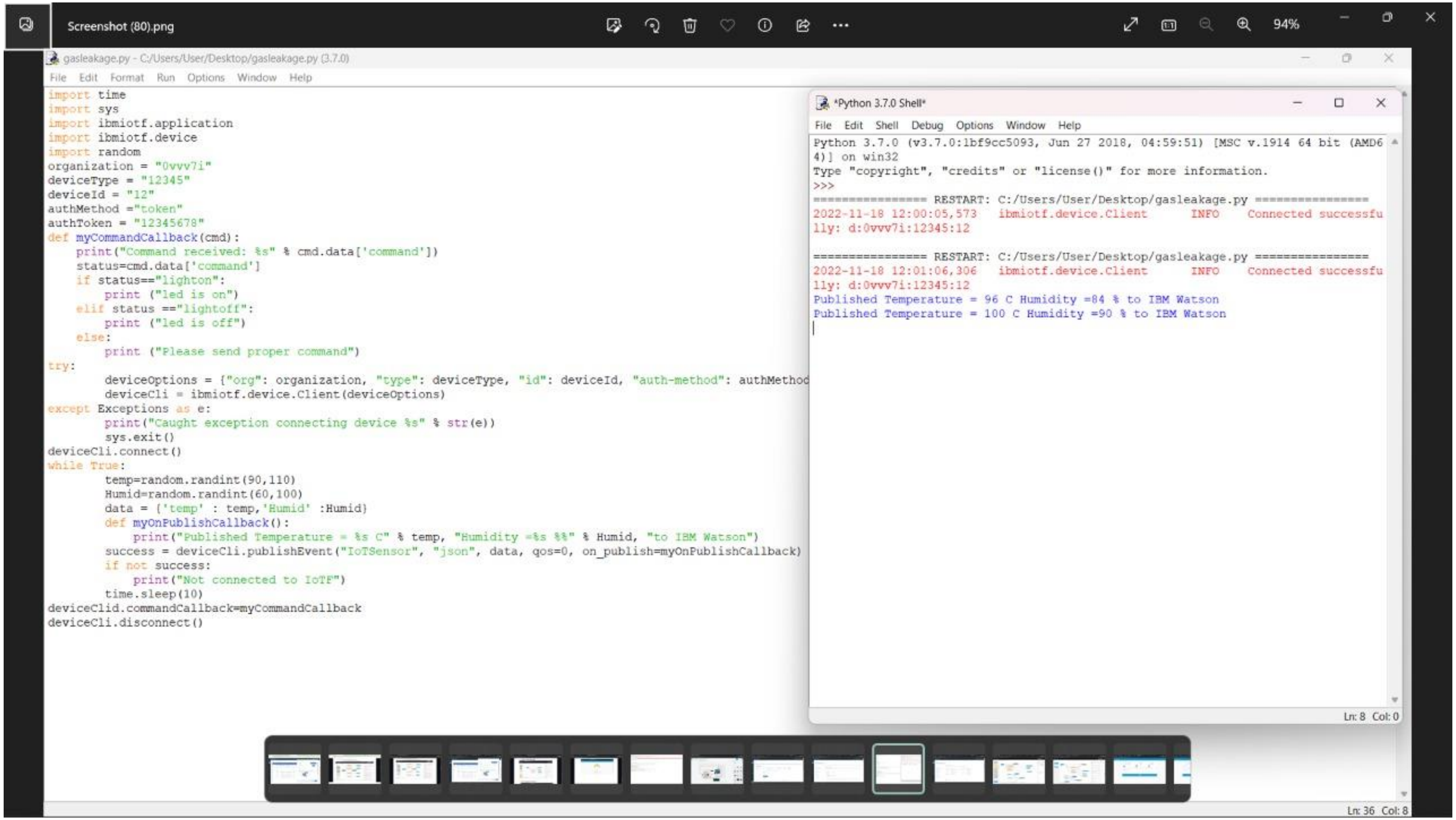
Event type name: event_1

Schedule: 1 Every Minute

Payload

Specify the event payload in the editor window or by uploading a CSV file.

```
{
  "HazardousGas ": random(0, 100),
  "temp": random(10, 80),
  "hum": random(80, 100),
  "pressure": random(49, 100)
}
```



Screenshot (80).png
94%
gasleakage.py - C:/Users/User/Desktop/gasleakage.py (3.7.0)
File Edit Format Run Options Window Help
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Ln: 8 Col: 0
Ln: 36 Col: 8

```python
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
organization = "0vvv7i"
deviceType = "12345"
deviceId = "12"
authMethod ="token"
authToken = "12345678"
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": authMethod
        deviceCli = ibmiotf.device.Client(deviceOptions)
except Exceptions as e:
        print("Caught exception connecting device %s" % str(e))
        sys.exit()
deviceCli.connect()
while True:
        temp=random.randint(90,110)
        Humid=random.randint(60,100)
        data = {'temp' : temp,'Humid' :Humid}
        def myOnPublishCallback():
            print("Published Temperature = %s C" % temp, "Humidity =%s %%" % Humid, "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)
deviceClid.commandCallback=myCommandCallback
deviceCli.disconnect()
```

```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD6
4)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
================ RESTART: C:/Users/User/Desktop/gasleakage.py ================
2022-11-18 12:00:05,573   ibmiotf.device.Client      INFO    Connected successfu
lly: d:0vvv7i:12345:12

================ RESTART: C:/Users/User/Desktop/gasleakage.py ================
2022-11-18 12:01:06,306   ibmiotf.device.Client      INFO    Connected successfu
lly: d:0vvv7i:12345:12
Published Temperature = 96 C Humidity =84 % to IBM Watson
Published Temperature = 100 C Humidity =90 % to IBM Watson
```