SPRINT-4

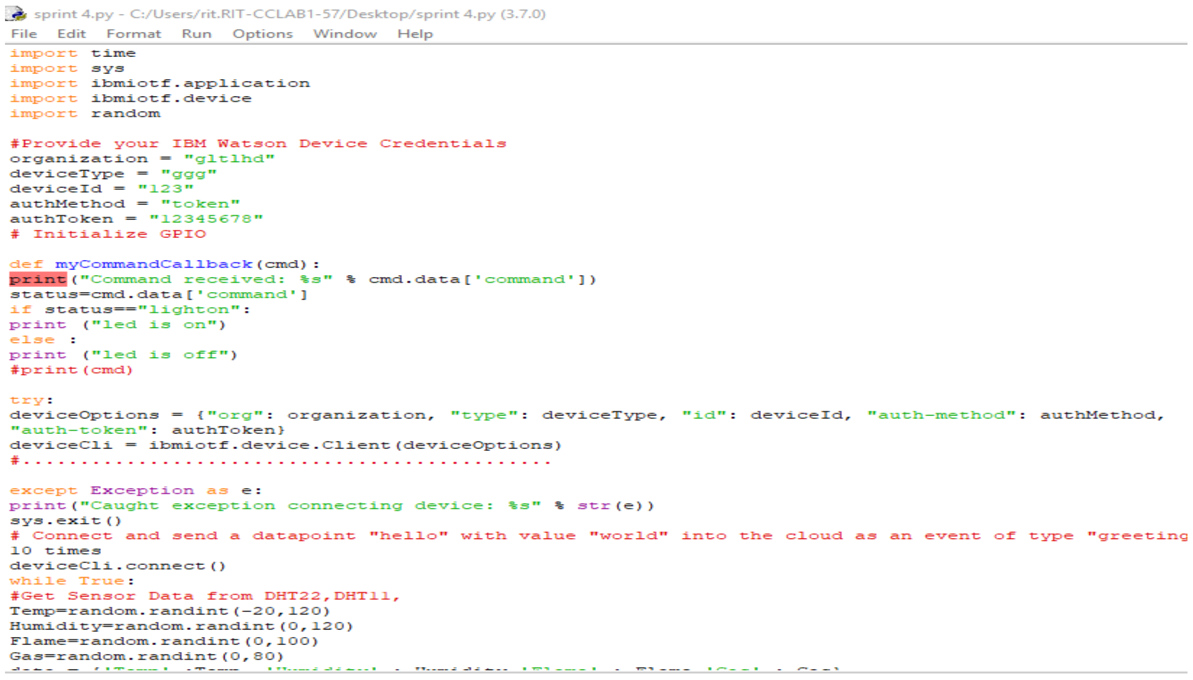
TEAM ID	**PNT2022TMID51070**
PROJECT TITLE	**INDUDTRY-SPECIFIC INTELLIGENT FIRE MANAGEMENT SYSTEM**

PYTHON PROGRAM:

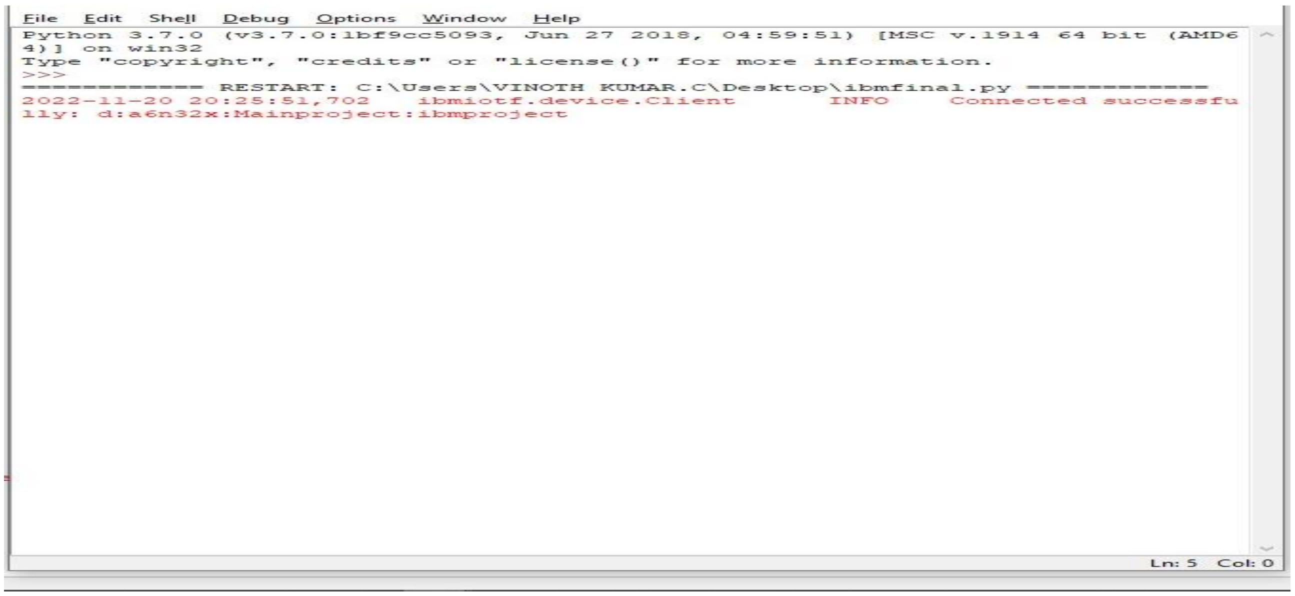
```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random

#Provide your IBM Watson Device Credentials
organization = "gltlhd"
deviceType = "ggg"
deviceId = "123"
authMethod = "token"
authToken = "12345678"
# Initialize GPIO

def myCommandCallback(cmd):
print("Command received: %s" % cmd.data['command'])
status=cmd.data['command']
if status=="lighton":
print ("led is on")
else :
print ("led is off")
#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 DHT22,DHT11,
Temp=random.randint(-20,120)
Humidity=random.randint(0,120)
Flame=random.randint(0,100)
Gas=random.randint(0,80)
```

```
data = {'Temp' :Temp ,'Humidity' : Humidity,'Flame' : Flame,'Gas' : Gas}

def myOnPublishCallback():
if Flame > 100:
data = {'Flame' : Flame}

print ("Temperature =%s c" % Temp ,"Humidity =%s u" % Humidity,"Flame =%s ir" % Flame ,"Gas
=%s ppm" % Gas )
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 and application from the cloud
deviceCli.disconnect()
```

PYTHON CODE OUTPUT:

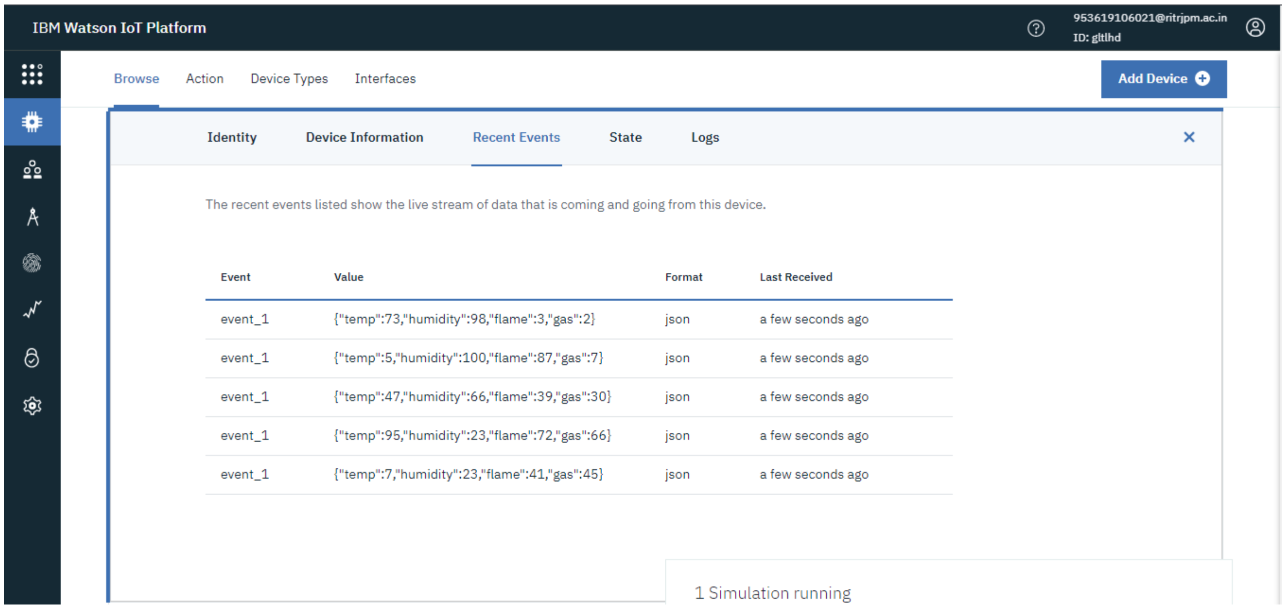

sprint 4.py - C:/Users/rit.RIT-CCLAB1-57/Desktop/sprint 4.py (3.7.0)
File Edit Format Run Options Window Help

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

#Provide your IBM Watson Device Credentials
organization = "gltlhd"
deviceType = "ggg"
deviceId = "123"
authMethod = "token"
authToken = "12345678"
# Initialize GPIO

def myCommandCallback(cmd):
print("Command received: %s" % cmd.data['command'])
status=cmd.data['command']
if status=="lighton":
print ("led is on")
else :
print ("led is off")
#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 DHT22,DHT11,
Temp=random.randint(-20,120)
Humidity=random.randint(0,120)
Flame=random.randint(0,100)
Gas=random.randint(0,80)
```



```
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\VINOTH KUMAR.C\Desktop\ibmfinal.py =============
2022-11-20 20:25:51,702   ibmiotf.device.Client      INFO    Connected successfu
lly: d:a6n32x:Mainproject:ibmproject
```

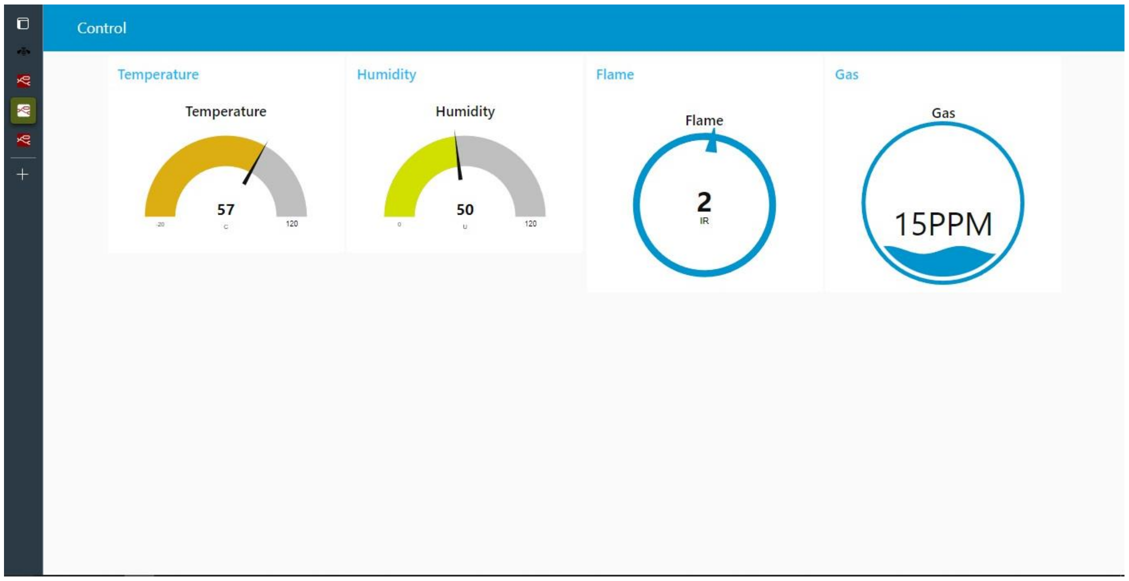
IBM WATSON OUTPUT:



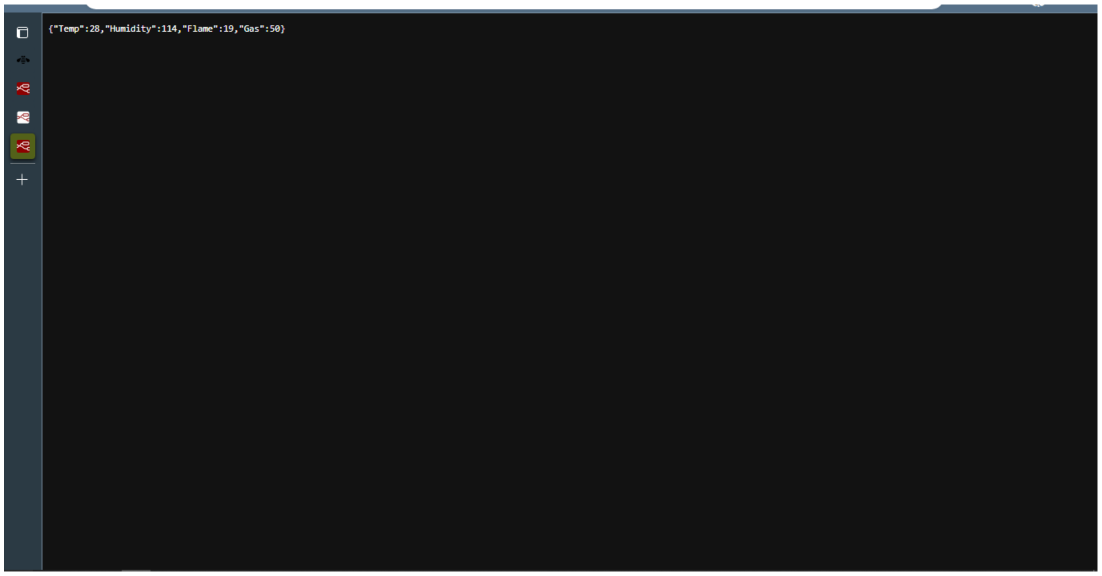
The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
event_1	{"temp":73,"humidity":98,"flame":3,"gas":2}	json	a few seconds ago
event_1	{"temp":5,"humidity":100,"flame":87,"gas":7}	json	a few seconds ago
event_1	{"temp":47,"humidity":66,"flame":39,"gas":30}	json	a few seconds ago
event_1	{"temp":95,"humidity":23,"flame":72,"gas":66}	json	a few seconds ago
event_1	{"temp":7,"humidity":23,"flame":41,"gas":45}	json	a few seconds ago

NODERED UI OUTPUT:



NODE RED SENSOR READING:



```
{"Temp":28,"Humidity":114,"Flame":19,"Gas":50}
```