SPRINT-2

TEAM ID: PNT2022TMID04636

PROJECT TITLE: Gas Leakage monitoring & Alerting system for Industries

Source code to deployed on IBM Watson Iot platform to generate the sensor data.

SOURCE CODE:

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
#Provide your IBM Watson Device Credentials
organization = "iv7q52"
deviceType = "shahidh"
deviceId = "9876"
authMethod = "token"
authToken = "-WmfT+4@nB5cfzR__k"
# Initialize GPIO

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
asan event of type "greeting" 10 times

deviceCli.connect()
while True:
 #Get Sensor Data from DHT11

  temp=random.randint(90,110)
  humid=random.randint(60,100)

  propane = random.randint(0, 2000);
  CO = random.randint(0, 100);
  lpg= random.randint(0, 2000);
  methane = random.randint(0, 1000);
```

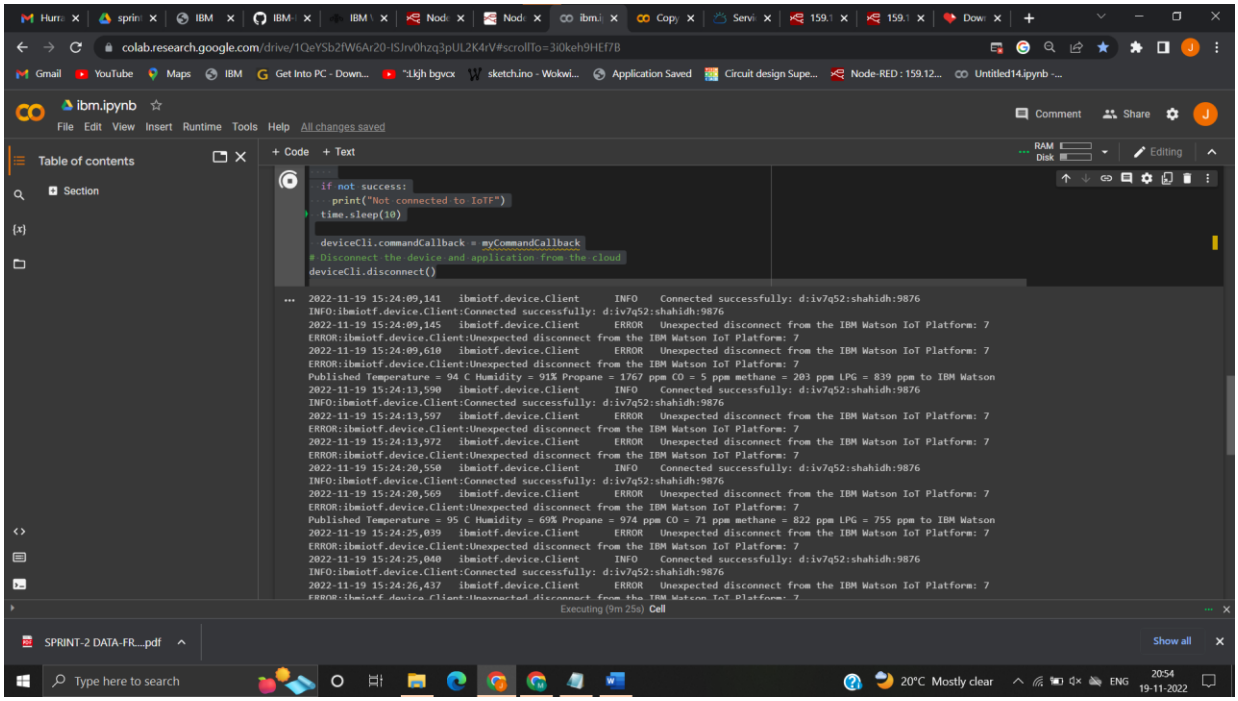
```
  data = { 'temp' : temp, 'humid': humid,"propane": propane,
 "CO": CO,
 "lpg": lpg,
 "methane": methane,
  }
 #print data
  def myOnPublishCallback():
    print ("Published Temperature = %s C" % temp, "Humidity = %s%%" % hum
id,"Propane = %s ppm" % propane,"CO = %s ppm" % CO,"methane = %s ppm" % m
ethane, "LPG = %s ppm" % lpg, "to IBM Watson")


  success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,on_pu
blish=myOnPublishCallback)

  if not success:
    print("Not connected to IoTF")
  time.sleep(10)

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

OUTPUT:



SENSOR DATA: