SPRINT-2

TEAM ID: PNT2022TMID04616

PROJECT TITLE: GAS LEAKAGE MONITORING AND ALERTING SYSTEM

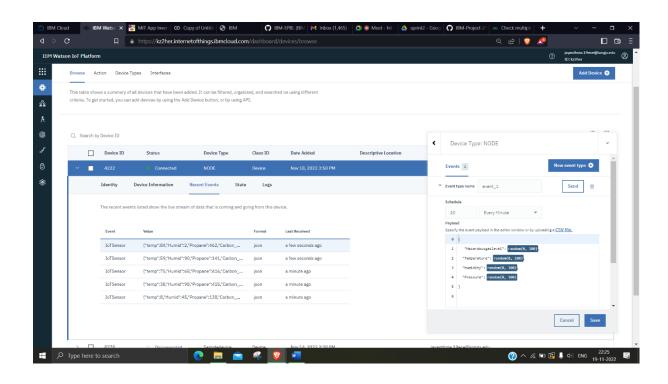
Source code to deployed on IBM Watson lot platform to generate the sensor data.

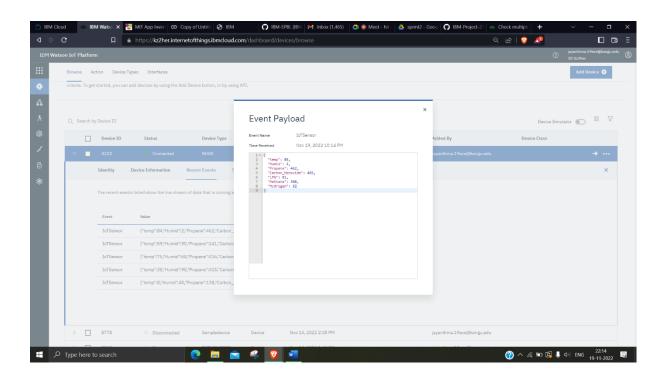
SOURCE CODE:

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
organization = "kz2her"
deviceType = "NODE"
deviceId = "4222"
authMethod = "token"
authToken = " j5RIM+NYy8Uv6+!s4q"
# Initialize GPIO
trv:
  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()
deviceCli.connect()
while True:
 Propane = random.randint(0, 500);
 Carbon Monoxide = random.randint(0, 500);
 LPG= random.randint(0, 1000);
 Methane = random.randint(0, 500);
```

```
Hydrogen= random.randint(0, 500);
  Temperature=random.randint(0,100);
  Humidity=random.randint(0,100);
 data = { "temp" : Temperature, "Humid": Humidity, "Propane": Propane,
 "LPG": LPG,
 "Methane": Methane,
 "Hydrogen": Hydrogen }
 def myOnPublishCallback():
   print ("Published Temperature = %s C" % Temperature, "Humidity = %s%
%" % Humidity,"Propane = %s ppm" % Propane, "LPG = %s ppm" % LPG,"Methan
e = %s ppm" % Methane, "Hydrogen = %s ppm" % Hydrogen, "Carbon monoxide =
%s ppm" % Carbon Monoxide , "to IBM Watson")
    if (Propane or Carbon Monoxide or LPG or Methane or Hydrogen)>150:
    print("GAS LEAKAGE FOUND")
    print("NO LEAKAGE")
  success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,on p
ublish=myOnPublishCallback)
  if not success:
   print("Not connected to IoTF")
  time.sleep(10)
 deviceCli.commandCallback = myCommandCallback
deviceCli.disconnect()
```

SENSOR DATA:





OUTPUT:

