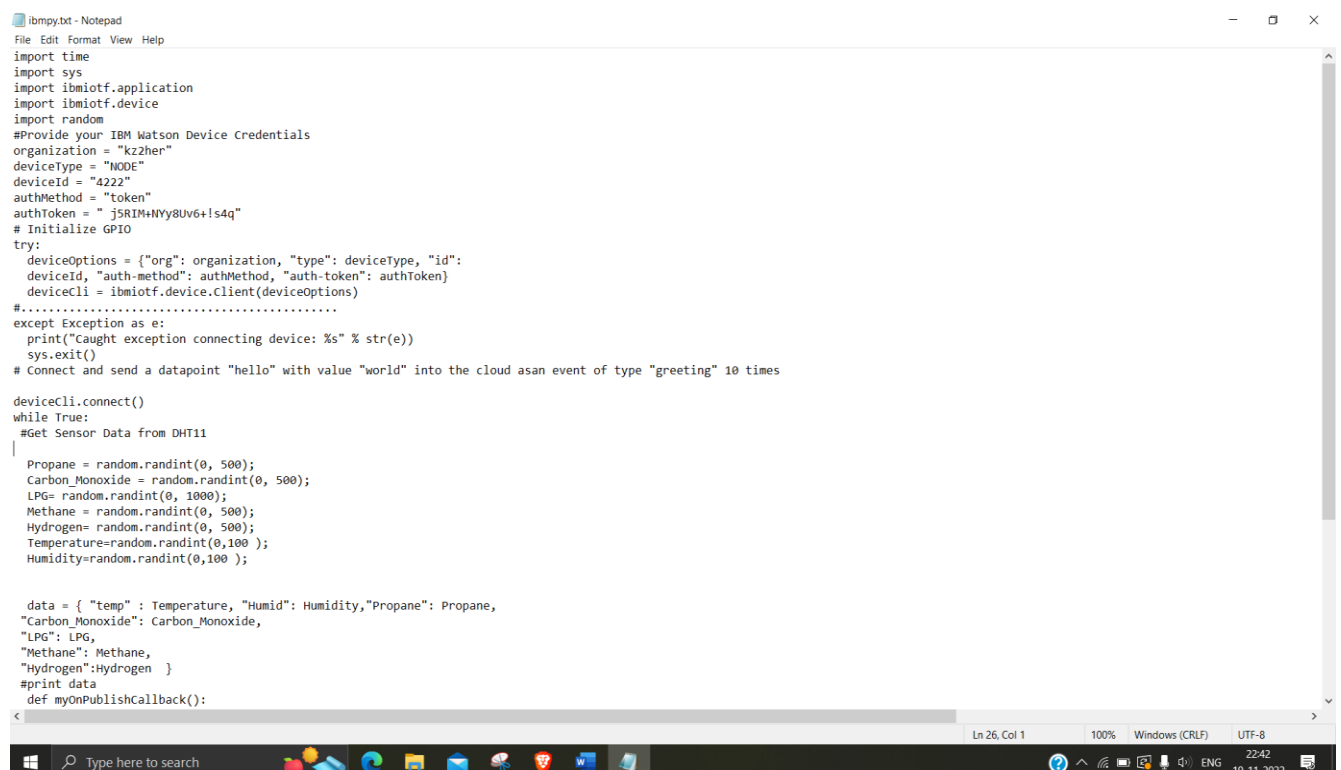


# PROJECT DEVELOPMENT PHASE

## SPRINT 3

TEAM ID	PNT2022TMID04616
PROJECT NAME	GAS LEAKAGE MONITORING AND ALERTING SYSTEM
DATE	31 OCTOBER 2022

**STEP 1:** Write a python code for randomize Temperature, Humidity, Hazardous gases like Methane, Propane, LPG etc.,

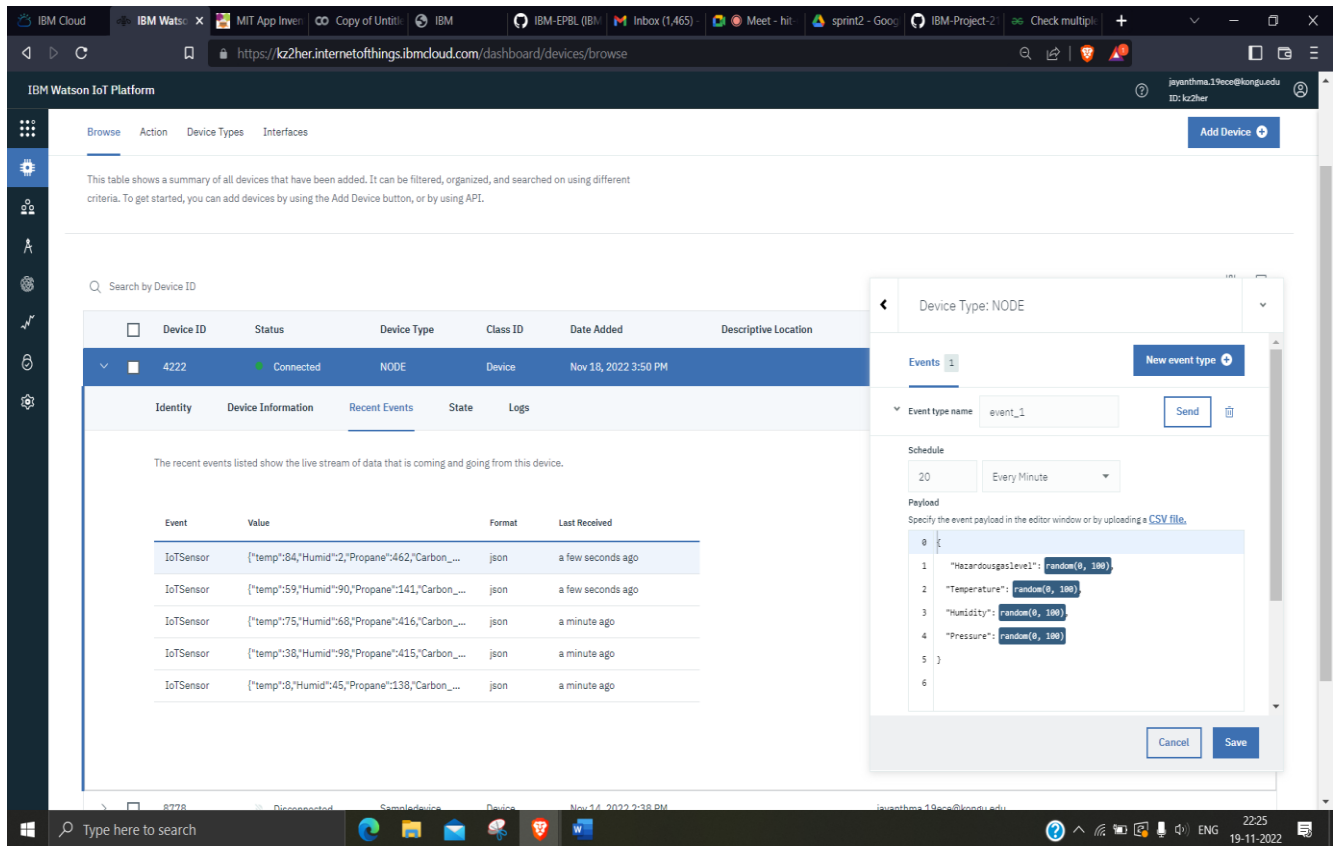


```
ibmpy.txt - Notepad
File Edit Format View Help
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
#Provide your IBM Watson Device Credentials
organization = "k22her"
devicetype = "NODE"
deviceId = "4222"
authMethod = "token"
authToken = " j5RIM+NVy8Uv6!s4q"
# Initialize 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
    |
    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,
        "Carbon_Monoxide": Carbon_Monoxide,
        "LPG": LPG,
        "Methane": Methane,
        "Hydrogen":Hydrogen  }
    #print data
    def myOnPublishCallback():
    <
Ln 26, Col 1    100%    Windows (CRLF)    UTF-8
22:42
19-11-2022
```

STEP 2: Run the python code it send data to IBM IoT Watson Platform.



The screenshot shows the IBM Watson IoT Platform dashboard. The main table lists devices, with device ID 4222 highlighted. The 'Recent Events' tab shows a stream of data from an IoT sensor. A modal window for 'Device Type: NODE' is open, showing a schedule of 'Every Minute' and a payload of random values for various sensors.

Event	Value	Format	Last Received
IoT Sensor	{"temp":84,"Humid":2,"Propane":462,"Carbon_..."	json	a few seconds ago
IoT Sensor	{"temp":59,"Humid":90,"Propane":141,"Carbon_..."	json	a few seconds ago
IoT Sensor	{"temp":75,"Humid":68,"Propane":416,"Carbon_..."	json	a minute ago
IoT Sensor	{"temp":38,"Humid":98,"Propane":415,"Carbon_..."	json	a minute ago
IoT Sensor	{"temp":8,"Humid":45,"Propane":138,"Carbon_..."	json	a minute ago

Device Type: NODE

Events: 1

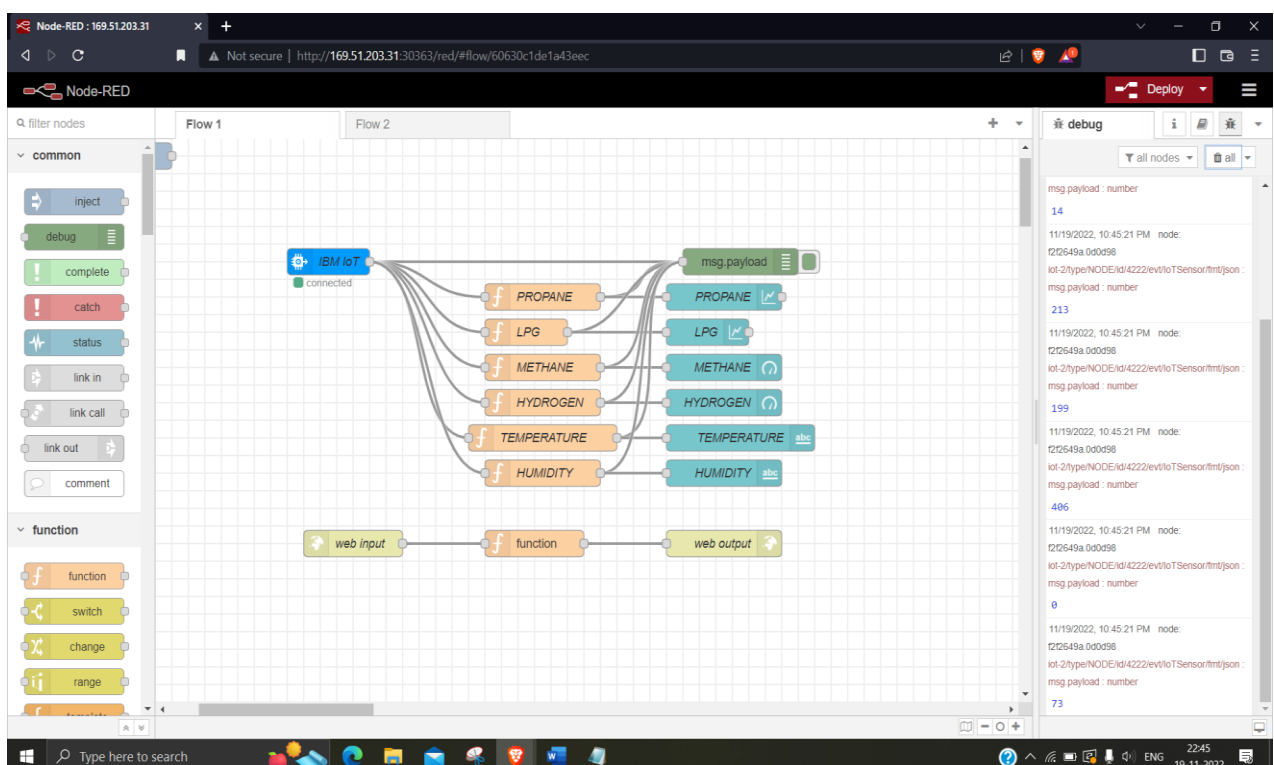
Event type name: event\_1

Schedule: 20 Every Minute

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

```
0 {
1   "Hazardousgaslevel": random(0, 100)
2   "Temperature": random(0, 100)
3   "Humidity": random(0, 100)
4   "Pressure": random(0, 100)
5 }
6
```

STEP 3: Open Node-RED flow dashboard.



The screenshot shows the Node-RED flow dashboard. The flow consists of an 'IBM IoT' node connected to a 'function' node, which then outputs to a 'web output' node. The debug console shows the resulting JSON payload.

```
msg.payload : number
14
11/19/2022, 10:45:21 PM node:
f22649a.0d0d98
iot-2/type/NODE/id/4222/evnt0TSensor/rmt/json :
msg.payload : number
213
11/19/2022, 10:45:21 PM node:
f22649a.0d0d98
iot-2/type/NODE/id/4222/evnt0TSensor/rmt/json :
msg.payload : number
199
11/19/2022, 10:45:21 PM node:
f22649a.0d0d98
iot-2/type/NODE/id/4222/evnt0TSensor/rmt/json :
msg.payload : number
406
11/19/2022, 10:45:21 PM node:
f22649a.0d0d98
iot-2/type/NODE/id/4222/evnt0TSensor/rmt/json :
msg.payload : number
0
11/19/2022, 10:45:21 PM node:
f22649a.0d0d98
iot-2/type/NODE/id/4222/evnt0TSensor/rmt/json :
msg.payload : number
73
```

STEP 4: Open Node-RED user interface to show the Hazardous gases, Humidity and Temperature value in respective charts.

