

# PROJECT DEVELOPMENT PHASE

## SPRINT-1

Team ID	PNT2022TMID11782
Project Name	Gas Leakage Monitoring And Alerting System

### **PYTHON CODE:**

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

#Provide your IBM Watson Device Credentials
organization = "d19wub"
deviceType = "Arduino"
deviceId = "1234"
authMethod = "token"
authToken = "cfspzFCmWpFlaA*aWR"

# Initialize GPIO
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 Exception as e:
```

```

print("Caught exception connecting : %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 DHT11

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

    data = { 'temp' : temp, 'Humid': Humid }
    #print data
    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 IoT")
        time.sleep(10)

    deviceCli.commandCallback = myCommandCallback

# Disconnect the device and application from the cloud
deviceCli.disconnect()

```

## OUTPUT:

The screenshot shows the IBM Watson IoT Platform dashboard. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. A sidebar on the left contains icons for various functions. The main content area displays a table of devices. The first device is identified as '1234', with a status of 'Connected' and a device type of 'Arduino'. It was last updated on '15 Nov 2022 01:35'. Below the table, a modal window titled 'Showing Raw Data | No Interfaces Available' is open, displaying a table of sensor data:

Property	Value	Type	Event	Last Received
temp	97	Number	IoTSensor	a few seconds ago
Humid	100	Number	IoTSensor	a few seconds ago

At the bottom of the dashboard, it indicates '0 Simulations running'.

The screenshot shows a Python 3.7.0 Shell terminal window. The user has executed a script named 'IOT\_gas.py' located at 'C:\Users\ELCOT\Downloads\IOT\_gas.py'. The terminal output shows several restarts of the script, each resulting in a 'ModuleNotFoundError: No module named 'ibmiotf''. Finally, the script runs successfully, displaying the following output:

```
2022-11-15 02:16:03,548 ibmiotf.device.Client INFO Connected successfully: d:d19wub:Arduino:1234
Published Temperature = 96 C Humidity = 94 % to IBM Watson
Published Temperature = 97 C Humidity = 70 % to IBM Watson
Published Temperature = 92 C Humidity = 61 % to IBM Watson
Published Temperature = 96 C Humidity = 71 % to IBM Watson
Published Temperature = 106 C Humidity = 93 % to IBM Watson
Published Temperature = 96 C Humidity = 95 % to IBM Watson
Published Temperature = 97 C Humidity = 98 % to IBM Watson
```

**RESULT:** The sensor Arduino with python code is connected successfully.