HINDUSTHAN COLLEGE OF ENGINEERING AND TECHNOLOGY

(AUTONOMOUS)

Team ID	PNT2022TMID10014
Project Name	Project - IOT Gas Leakage Monitoring and Alerting System.

SPRINT 1:

In this sprint we have developed a python code to generate random sensor data and publish that data to the IBM internet of things platform using a python package called ibmiotf. These data will be published to the respected device in that platform.

PYTHON CODE:

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

# IBM Watson Device Credentials
organization = "hfj0vp"  # Organization ID
deviceType = "IOT_Device"  # Device type
deviceId = "Gas_Leakage_Detector"  # Device id
authMethod = "token"
authToken = " "  # Authentication token should be given here. It is not provided here since it is a demo and for security reasons.

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()

deviceCli.connect()

while True:
    # Ransom sensor data generation
    T = random.randint(-40, 80)
    H = random.randint(0, 100)
    G = random.randint(100, 10000)
    A = "OFF"  # Alert flag

    if G >= 1000:  # We can add as many conditions here to check other sensor
data
        A = "ON"

    else:
        A = "OFF"

    # Send sensor data to IBM Watson
    data = {'temperature': T, 'humidity': H, 'gas': G, 'alert': A}

    # print data
    def myOnPublishCallback():
        print("Published Temperature = %s C" % T, "Humidity = %s %%" % H,
"Gas level = %s ppm" % G, "to IBM Watson")


    success = deviceCli.publishEvent("event", "json", data, qos=0,
on_publish=myOnPublishCallback)
    if not success:
        print("Not connected to IoTF")
    time.sleep(5)


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