SPRINT -1

Team ID	PNT2022TMID47529
Project Name	**Project** - Gas Leakage Monitoring and Alerting System
Maximum Marks	2 Marks
Team Members	
Team Head	AJAY KUMAR K – 910419106001
Team Member 1	SNEHA R M – 910419106301
Team Member 2	KANNAKI M – 910419106003
Team Member 3	SUBITSHA R – 910419106009

Code

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
#Provide your IBM Watson Device Credentials
organization = "r0cxw7"
deviceType = "ASKS"
deviceId = "1802"
authMethod = "token"
authToken = "wYQPYB74loS8@uGkWD"
# 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 device:
%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 IoTF")
    time.sleep(10)

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