Final Code

Team Id	*PNT2022TMID51029*
Project	*Gas Leakage Monitoring and Alerting System For Industries*
Team Members	*Shriya.R, Swathy Santhosh, Akila.G,LavanyaDevi.K*

FINAL CODE:

TINKERCARD CODE:

```
int gasSensor=A1;
int buzzer=13;
int led=12;
void setup()
{
   pinMode(A1, INPUT);
   pinMode(buzzer, OUTPUT);
   pinMode(led, OUTPUT);
   Serial.begin(9600);
 }

void loop()
{
int sensorValue=analogRead(gasSensor);
Serial.print("GAS LEVEL:");
Serial.println(sensorValue);
delay(1000);
if (sensorValue>250)
  {
     digitalWrite(buzzer,HIGH);
     digitalWrite(led,HIGH);

  }
else
  {
     digitalWrite(buzzer,LOW);
     digitalWrite(led,LOW);
  }
}
```

PYTHON CODE:

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


#Provide your IBM Watson Device Credentials
organization = "a49saj"
deviceType = "weather_device"
deviceId = "weather_device_1"
authMethod = "token"
authToken = "T?GTpUXf2q7+MrvYil"

# Initialize GPIO


def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="lighton":
        print ("led is on")
    else :
        print ("led is off")

    #print(cmd)




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(0,100)
        Humid=random.randint(0,100)
        hazard=random.randint(0,100)
        pres=random.randint(0,100)

        data = { 'temp' : temp, 'Humid': Humid, 'hazard' : hazard, 'pres' : pres }
```

```
        #print data
        def myOnPublishCallback():
            print ("Published Temperature = %s C" % temp, "Humidity = %s gal" % Humid,
"Hazardous Gas = %s ppm" % hazard, "Pressure = %s %%" % pres, "to IBM Watson")

        success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,
on_publish=myOnPublishCallback)
        if not success:
            print("Not connected to IoTF")
        time.sleep(1)

        deviceCli.commandCallback = myCommandCallback

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

OUTPUT: