IBM PROJECT

GAS LEAKAGE MONITORING AND ALERTING SYSTEM FOR INDUSTRIES

Team ID: PNT2022TMID12932

Team Leader: MAITHREYAN M

Team Members:

- RAMKISHORE T
- ABINESH P
- ABISHEK S
- SARAN KUMAR A

CONTENTS

1. INTRODUCTION

1.1 Project Overview:

This project helps the industries in monitoring the emission of harmful gases. In several areas, the integration of gas sensors helps in monitoring the gas leakage. If in any area gas leakage is detected the admins will be notified along with the location. In the web application, admins can view the sensor parameters.

1.2 Purpose:

Inhaling concentrated gas can lead to asphyxia and possible death.To overcome these disasters, we designed a system for monitoring and alerting the leakage of those harmful gases. This makes the industrialists get rid of the fear of any disasters caused by the gases.

2. LITERATURE SURVEY

2.1 Existing Problem:

The number of sensors is unpredictable and the positioning of equipment is improper and also the affordable of the system is high and the systems are sometimes causing heavy disasters.

2.2 References:

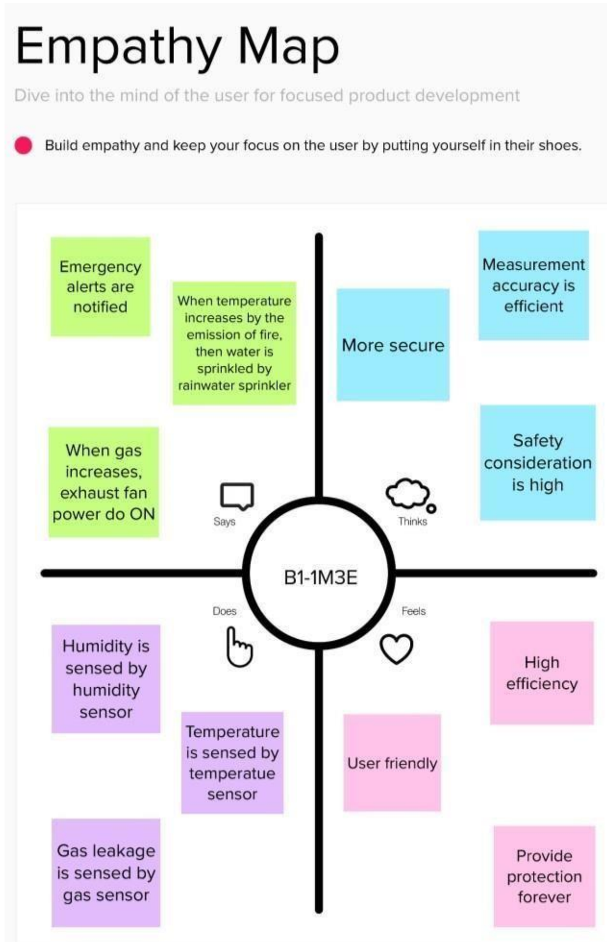
i. Bing Han, Qiang Fu, Hanfang Hou, 'Methane Leakage Monitoring Technology For Natural Gas Stations and Its Application', IEEE 5th International Conference on Computer and Communications,2001. Shruthi Unnikrishnan,1 Mohammad Razil, Joshua Benny, Shelvin Varghese and C.V. Hari, 'LPG Monitoring And Leakage Detection System', Department of Applied Electronics and Instrumentation Engineering, Rajagiri School of Engineering and Technology, Rajagiri Valley, Kakkanad, Kochi, India. J.Vijayalakshmi, Dr.G.Puthilibhai, S.R.Leoram Siddarth, 'Implementation Of Ammonia Gas Leakage Detection & Monitoring System Using Internet Of Things', West Tambaram, Chennai. Makiko Kawada, Tadao Minagawa, Eiichi Nagao, Mitsuhito Kamei, Chieko Nishida and Koji Ueda, 'Advanced Monitoring System For Gas Density Of GIS', Mitsubishi Electric Corporation.

2.3 Problem statement definition:

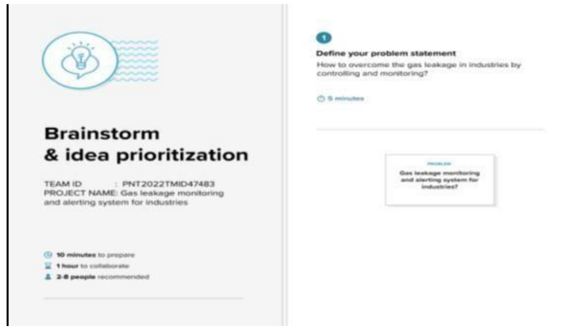
Since the number of sensors is unpredictable, the industrialists feel in secured in handling the gases. Also, the cost price of the products and the complications in installing the systems are high. This makes the customers feel disappointed sometimes.

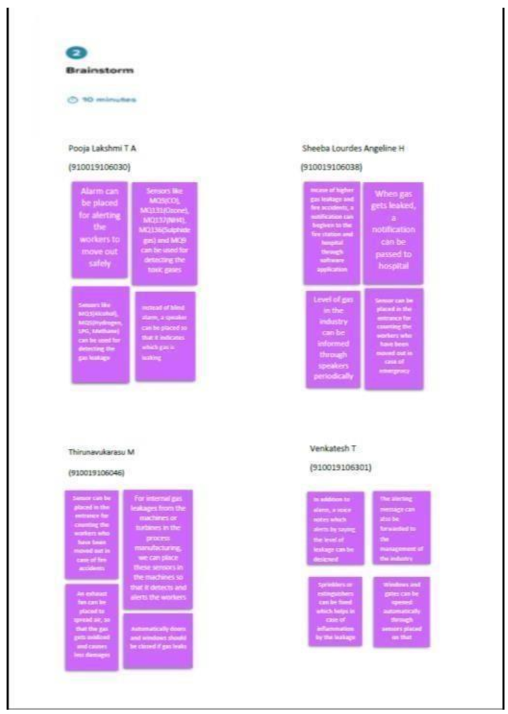
3. IDEATION & PROPOSED SOLUTION

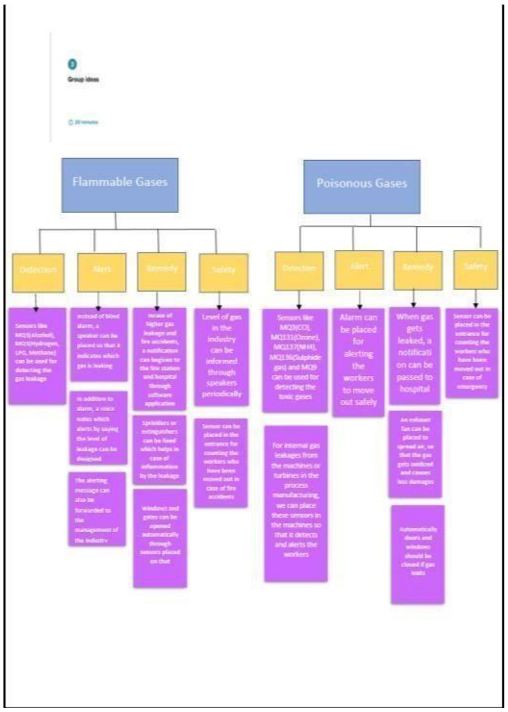
3.1 Empathy Map Canvas:

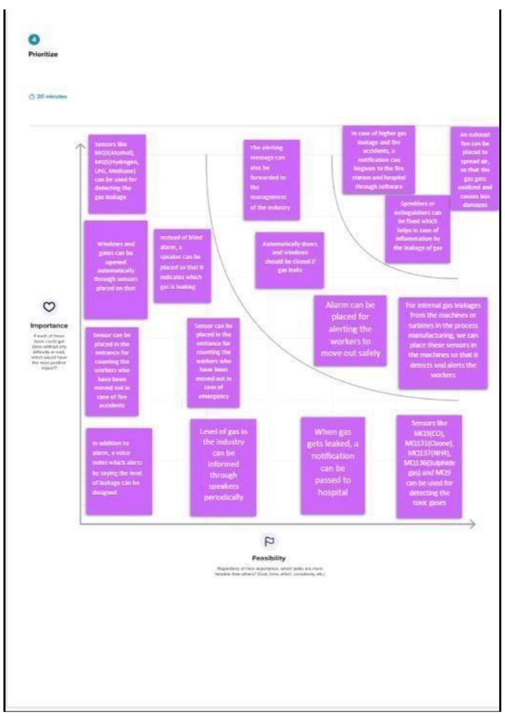


3.2 **Ideation & Brainstorming:**





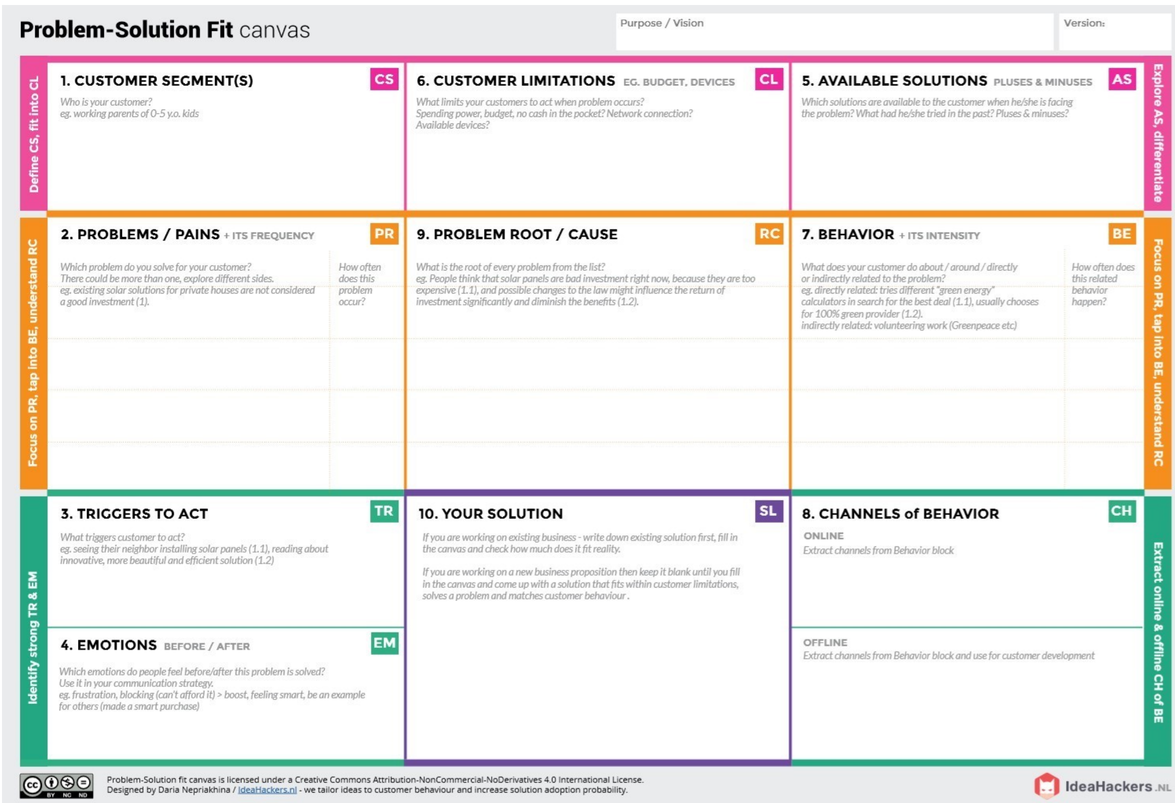

Group ideas
Flammable Gases
Poisonous Gases
Detection
Alert
Remedy
Safety


Prioritize
Importance
Feasibility
Alarm can be placed for alerting the workers to move out safely
Level of gas in the industry can be informed through speakers periodically
When gas gets leaked, a notification can be passed to hospital

3.3 Proposed Solution:

S.No.	Parameter	Description
	Problem Statement	Develop an efficient system & an application that can monitor and alert the users(workers)
	Idea / Solution description	This product helps the industries in monitoring the emission of harmful gasesIn several areas, the gas sensors will be integrated to monitor the gas leakage.If in any area gas leakage is detected the admins will be notified along with the locationIn the web application, admins can view the sensor parameters.
	Novelty / Uniqueness	Fastest alerts to the workersUser friendly
	Social Impact / Customer Satisfaction	Cost efficientEasy installation and provide efficient resultsCan work with irrespective of fear
	Business Model (Revenue Model)	The product is advertised all over the platforms. Since it is economical, it even helps small scale industries from disasters.As the product usage can be understood by everyone, it is easy for them to use it properly for their safest organization.
	Scalability of the Solution	Since the product is cost-efficient, it can be placed in many places in the industry. Even when the gas leakage is more, the product senses the accurate values and alerts the workers effectively.

3.4 **<u>Problem Solution Fit:</u>**



4. REQUIREMENT ANALYSIS

4.1 Functional Requirement:

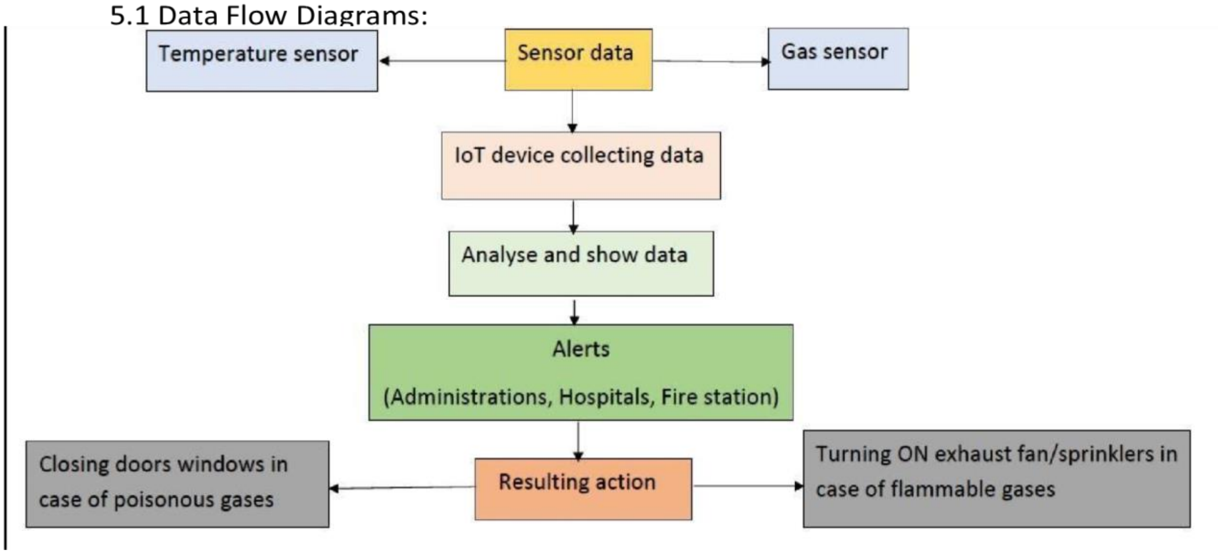
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Visibility	The level of gas can be monitored by users if there is any leakage, alerts can be sent through messages.
FR-2	User Reception	The data like the level of gas can be sent through messages
FR-3	User Understanding	The user can monitor the level of gas with the help of the data. If there is an increase in gas level, then the alert will be given. They also get notified by the alert.
FR-4	User Convenience	Through messages we can easily get data of gas level and in case of gas leakage, it can directly send notifications to nearby police stations and hospitals.
FR-5	User Performance	When the user gets notified, he could turn on the exhaust fan/sprinkler.

4.2 Non-Functional Requirement:

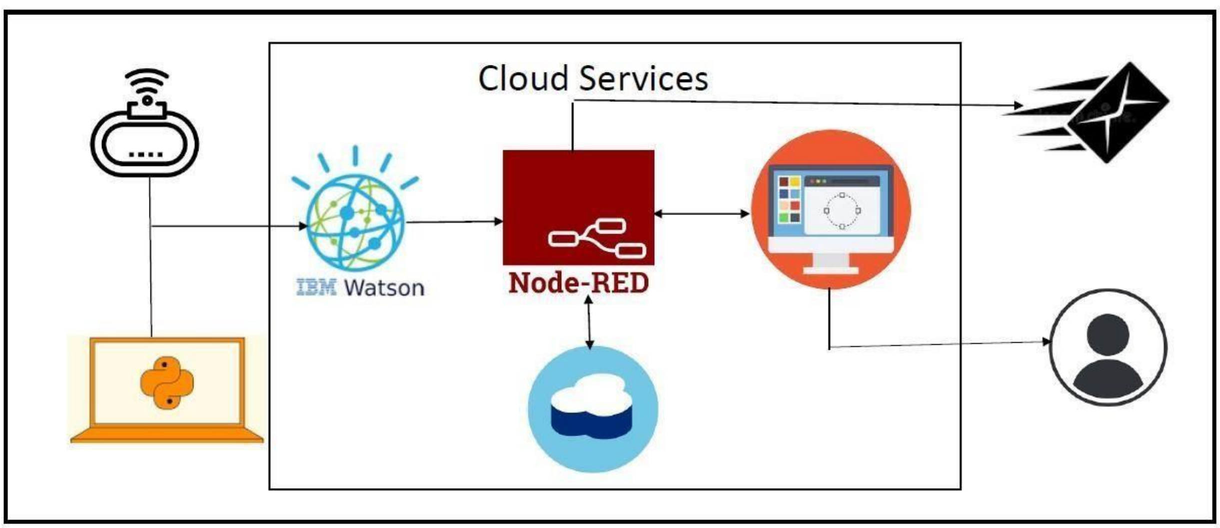
FR No.	Non-Functional Requirement	Description
NFR-1	Usability	It updates the data regularly as well as protects the workers.
NFR-2	Security	As a result of emergency alert, we can be able to protect both humans and properties.
NFR-3	Reliability	Can be able to provide accurate values. It might have a capacity to recognize the smoke accurately and does not give a false
NFR-4	Performance	Sprinklers and exhaust fans are used in case of emergency.
NFR-5	Availability	It can be used for everyday; it includes day and nights.
NFR-6	Scalability	Sensors can be replaced every time it fails.

5. PROJECT DESIGN

5.1 Data Flow Diagrams:



5.2 **Solution & TechnicaArchitecture:**



5.3 **User Stories:**

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	User can enter into the web application	I can access my account / dashboard	High	Sprint-1
		USN-2	Users can register their credentials like email id and password	I can receive confirmation email and click confirm	High	Sprint-1
	Login	USN-3	User can log in to the application by entering email and password	I can login to my account	High	Sprint-1
	Dashboard	USN-4	User can view the temperature	I can view the data given by the device	High	Sprint-2
		USN-5	User can view the level of gas	I can view the data given by the device	High	Sprint-2
Customer (Web user)	Usage	USN-1	User can view the webpage and get the information	I can view the data given by the device	High	Sprint-3
Customer	Working	USN-1	User act according to the alert given by the device	I can get the data work according to it	High	Sprint-3
		USN-2	User turns ON the exhaust fan/sprinkler when the leakage occurs	I can get the data work according to it	High	Sprint-4

Customer Care Executive	Action	USN-1	User solve the problems when someone faces any usage issues	I can solve the issues when someone fails to understand the procedure	High	Sprint-4
Administra tor	Administration	USN-1	User stores every information	I can store the gained information	High	Sprint-4

7. CODING AND SOLUTIONING

7.1 Feature 1

```
import time import sys import
ibmiotf.application import
ibmiotf.device import random
#Provide your IBM Watson Device Credentials organization
= "vq4nsy" deviceType = "PNT2022TMID4465" deviceId =
"PNT2022TMID4465DEVICEID" authMethod =

"token" authToken = "rjghjHFTDHB!"
# Initialize GPIO
def myCommandCallback(cmd):     print("Command received:
%s" % cmd.data['command'])
status=cmd.data['command']     if status=="alarmon":
print ("Alarm is on")     elif (status == "alarmoff") :
```

print ("Alarm is off")   elif status ==

"sprinkleron":   print("Sprinkler is ON")   elif status

== "sprinklerOFF":

print("Sprinkler 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)   gas=random.randint(0,100)

      data = { 'temp' : temp, 'Humid': Humid, 'gas' : gas }

      #print data   def

myOnPublishCallback():

          print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % Humid, "Gas_Level = %s %%" %gas, "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()

7.2 **Feature 2: (Python Output)**

```
File Edit Shell Debug Options Window Help
Published Temperature = 72 C Humidity = 38 % Gas_Level = 93 % to IBM Watson
Published Temperature = 29 C Humidity = 58 % Gas_Level = 63 % to IBM Watson
Published Temperature = 71 C Humidity = 14 % Gas_Level = 87 % to IBM Watson
Published Temperature = 5 C Humidity = 32 % Gas_Level = 92 % to IBM Watson
Published Temperature = 51 C Humidity = 20 % Gas_Level = 82 % to IBM Watson
Published Temperature = 87 C Humidity = 10 % Gas_Level = 62 % to IBM Watson
Published Temperature = 35 C Humidity = 14 % Gas_Level = 19 % to IBM Watson
Published Temperature = 8 C Humidity = 28 % Gas_Level = 81 % to IBM Watson
Published Temperature = 69 C Humidity = 90 % Gas_Level = 50 % to IBM Watson
Published Temperature = 39 C Humidity = 0 % Gas_Level = 51 % to IBM Watson
Published Temperature = 88 C Humidity = 62 % Gas_Level = 27 % to IBM Watson
Published Temperature = 76 C Humidity = 89 % Gas_Level = 98 % to IBM Watson
Published Temperature = 99 C Humidity = 90 % Gas_Level = 12 % to IBM Watson
Published Temperature = 93 C Humidity = 36 % Gas_Level = 7 % to IBM Watson
Published Temperature = 98 C Humidity = 23 % Gas_Level = 40 % to IBM Watson
Published Temperature = 32 C Humidity = 72 % Gas_Level = 62 % to IBM Watson
Published Temperature = 55 C Humidity = 7 % Gas_Level = 80 % to IBM Watson
Published Temperature = 100 C Humidity = 74 % Gas_Level = 29 % to IBM Watson
Published Temperature = 64 C Humidity = 86 % Gas_Level = 13 % to IBM Watson
Published Temperature = 55 C Humidity = 5 % Gas_Level = 17 % to IBM Watson
Published Temperature = 72 C Humidity = 28 % Gas_Level = 37 % to IBM Watson
Published Temperature = 10 C Humidity = 54 % Gas_Level = 65 % to IBM Watson
Published Temperature = 30 C Humidity = 82 % Gas_Level = 82 % to IBM Watson
Published Temperature = 40 C Humidity = 95 % Gas_Level = 57 % to IBM Watson
Published Temperature = 28 C Humidity = 18 % Gas_Level = 17 % to IBM Watson
Published Temperature = 47 C Humidity = 66 % Gas_Level = 50 % to IBM Watson
Published Temperature = 58 C Humidity = 86 % Gas_Level = 50 % to IBM Watson
Published Temperature = 98 C Humidity = 19 % Gas_Level = 87 % to IBM Watson
Published Temperature = 12 C Humidity = 81 % Gas_Level = 40 % to IBM Watson
Published Temperature = 32 C Humidity = 79 % Gas_Level = 75 % to IBM Watson
Published Temperature = 37 C Humidity = 80 % Gas_Level = 24 % to IBM Watson
Published Temperature = 73 C Humidity = 59 % Gas_Level = 40 % to IBM Watson
Published Temperature = 51 C Humidity = 69 % Gas_Level = 34 % to IBM Watson
Published Temperature = 96 C Humidity = 13 % Gas_Level = 68 % to IBM Watson
Published Temperature = 28 C Humidity = 62 % Gas_Level = 7 % to IBM Watson
Published Temperature = 86 C Humidity = 69 % Gas_Level = 34 % to IBM Watson
Published Temperature = 48 C Humidity = 5 % Gas_Level = 40 % to IBM Watson
Published Temperature = 20 C Humidity = 51 % Gas_Level = 78 % to IBM Watson
Published Temperature = 60 C Humidity = 2 % Gas_Level = 91 % to IBM Watson
Published Temperature = 42 C Humidity = 86 % Gas_Level = 64 % to IBM Watson
Published Temperature = 95 C Humidity = 47 % Gas_Level = 99 % to IBM Watson
Published Temperature = 49 C Humidity = 16 % Gas_Level = 84 % to IBM Watson
Published Temperature = 59 C Humidity = 25 % Gas_Level = 66 % to IBM Watson
Published Temperature = 85 C Humidity = 100 % Gas_Level = 56 % to IBM Watson
Published Temperature = 65 C Humidity = 73 % Gas_Level = 13 % to IBM Watson
Published Temperature = 48 C Humidity = 38 % Gas_Level = 38 % to IBM Watson
```

8. ADVANTAGES AND DISADVANTAGES

Advantages:

- Detect the concentration of the gases
- The sensor-enabled solution helps prevent the high risk of gas explosions and affecting any casualties within and outside the premises.
- Get real-time alerts about the gaseous presence in the atmosphere
- Prevent fire hazards and explosions
- Supervise gas concentration levels
- Ensure worker's health
- Real-time updates about leakages
- Cost-effective installation
- Data analytics for improved decisions
- Measure oxygen level accuracy
- Get immediate gas leak alerts

Disadvantages:

- Only one gas can be measured with each instrument.
- When heavy dust, steam or fog blocks the laser beam, the system will not be able to take measurements.

8. CONCLUSION

Gas leakage leads to severe accidents resulting in material losses and human injuries.Gas leakage occurs due to poor maintenance of equipment and inadequate awareness of the people. Hence, gas leakage detection is essential to prevent accidents and to save human lives. This paper presented LPG leakage detection and alert system. This system triggers buzzer and notification to alert people when gas leakage is detected. This system is basic yet reliable.

9. FUTURE SCOPE

Major cities of India are pushing Smart Home application, gas monitoring system is a part of SmartHome application. Enhancing Industrial Safety using IoT. This system can be implemented in Industries, Hotels and wherever the gas cylinders are used. This system can be used in industries involving applications such as Furnace, Boilers, Gas welding, Gas cutting, Steel Plants, Metallurgical industries, Food processing Industries, Glass Industries, Plastic industries, Pharmaceuticals, Aerosol manufacturing. As hospitals require to provide maximum possible safety to patients, this system can be used to keep track of all the cylinders used in it. Some of the cylinders used are Oxygen cylinder, Carbon dioxide cylinder, Nitrous oxide cylinder. As many students are naive the risk of causing accidents is high. Hence, our system can also be used in schools, colleges. Many colleges have well established labs including chemistry lab and pharmaceutical labs where gas burners are used. Several medical equipment requires gas cylinders.

GIT LINK

https://github.com/IBM-EPBL/IBM-Project-19264-1659694977

TINKERCAD LINK:
https://www.tinkercad.com/things/cRl5hzjYGgl-gas-leakage-monitoring-and-alerting-system-for-industries-final/editel?sharecode=x5P2SzqBf6HYo305whOHpVdFtFxtr5Kb8gzoT6rV5A0