Gas Leakage Monitoring And Alerting System For Industries

(Hemachandhar N, Keerthana R, Charudesna K, Dharshini S)

(Team ID: PNT2022TMID32740)

1. INTRODUCTION

Project Overview

The internet of Things is a developing topic of technical, social, and economic significance. The usage of the gas brings great problems in the domestic as well as working places. The inflammable gas, which is excessively used in the work places (Industries). The leakage of the gas causes destructible impact to the lives and as well as to the heritage of the people. Most of the societies have fire safety mechanism. But itcan use after the fire exists. As a result, a system for detecting and monitoring gas leaks is required. Through a flame sensor, the system will sense fire and flame. The buzzer begins to ring when a fire is detected. Tests have shown that the system can keep track of the wastage of gas and leaks and notify the user. The performance that was produced showed that it was successful in reducing the amount of gas that was wasted.

Purpose

The design of a sensor-based automatic gas leakage detector with an alert andcontrol system has been proposed. This is an affordable, less power using, lightweight, portable, safe, user friendly, efficient, multi featured and simple system device for detecting gas. To monitor this gas leak, the system includes an MQ6 gas detector. Thissensor detects the amount of leaking gas present in the surrounding atmosphere. In this way, the consequences of an explosion or gas leak can be avoided**.**

2. LITERATURE SURVEY

Existing Problem

Gas leakage is nothing but the leak of any gaseous molecule from a pipeline, orcylinder etc in the industries. Gas Leakages in open or closed areas can prove to be dangerous .This can occur either purposefully or even unintendedly. As we are aware that these kinds of leaks are dangerous to our health, and when it becomes explosive it could cause great danger to the people, industry and the environment. Therefore, we have used IoT technology to make a Gas Leakage Detector for society which has Smart Alerting techniques involving sending a text message to the concerned authority and the ability to perform data analytics on sensor readings. Our main aim is to propose a gas leakage system for a society where each flat has gas leakage detector hardware. This will detect the harmful gases in the environment and alerting to society members through the alarm and sending notifications.

References

1. Shital Imade, Priyanka Rajmanes, Aishwarya Gavali , Prof. V. N. Nayakwadi “GAS LEAKAGE DETECTION AND SMART ALERTING SYSTEM USING IOT” https://www.pramanaresearch.org/gallery/22.%20feb%20ijirs%20-%20d539.pdf

2. Kumar Keshamoni and Sabbani Hemanth. "Smart Gas Level Monitoring, Booking & Gas Leakage Detector over IoT " International Advance Computing ConferenceIEEE, 2017.

3. Petros Spachos , Liang Song and Dimitrios Hatzinakos. "Gas Leak Detection and Localization System Through Wireless Sensor Networks" The 11th Annual IEEE Consumer Communications and Networking Conference - Demos. IEEE, 2014.

4. "Design and Implementation of an Economic Gas Leakage Detector" National Institute of Health (2004). What you need to know about natural gas detectors. Available:http://www.nidcd.nih.gov/health/smelltaste/gas dtctr.asp.

5. Prof.M.Amsaveni, A.Anurupa, R.S.Anu Preetha, C.Malarvizhi,M.Gunasekaran "Gsm based LPG leakage detection and controlling system" the International Journal of Engineering and Science (IJES) ISSN (e): 2319 – 1813 ISSN (p):2319 –1805 Pages 112-116 March- 2015.

6. Pal-Stefan Murvaya, IoanSileaa "A survey on gas leak detection and localization techniques".

7. Ch. Manohar Raju, N. Sushma Rani, "An android based automatic gas detection and indication robot. In International Journal of Computer Engineering and Applications. 2014;8(1).

8. Falohun A.S., Oke A.O., Abolaji B.M. "Dangerous Gas Detection using an Integrated Circuit and MQ-9" in International Journal of Computer Applications (0975 –8887) Volume 135 – No.7, February 2016.

9. Ashish Shrivastava,Ratnesh Prabhaker, Rajeev Kumar and Rahul Verma "GSM BASED GAS LEAKAGE DETECTION SYSTEM" in International Journal of TechnicalResearch and Applications e-ISSN: 2320- 8163,www.ijtra.com Volume 1, Issue 2(may-June 2013).

10. C.Selvapriya, S.Sathyaprabha, M.Abdulrahim," LPG leakage monitoring and multilevel alerting system", published in 2013.

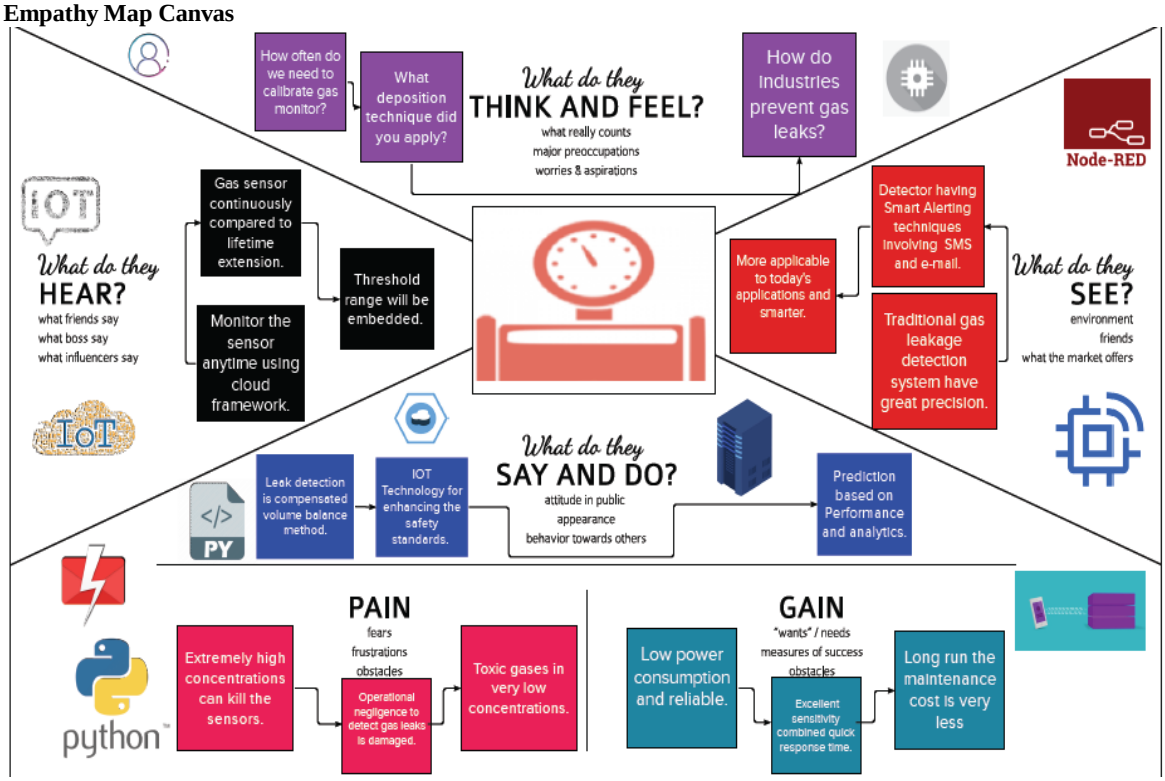
11. Falohun A.S., Oke A.O., Abolaji B.M. "Dangerous gas detection using an integrated circuit and MQ-9. In International Journal of Computer Applications. 2016; 135(7).

Problem Statement Definition

Gas leakage occur. Gas leakage can be detected by human nearby and if there is no human it cannot be detected. But sometimes it cannot be detected by human that has a low sense of smell.Gas leakage leads to various accidents resulting in financial loss as well as injuries to human. In routine life, the environment has the most significant effect on human health.

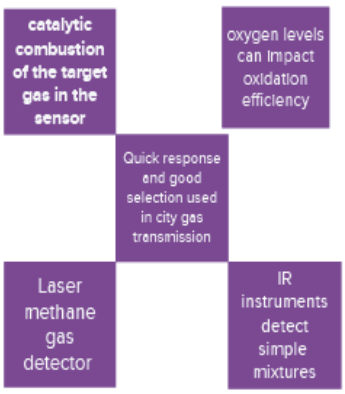
3. IDEATION & PROPOSED SOLUTION

Empathy Map Canvas

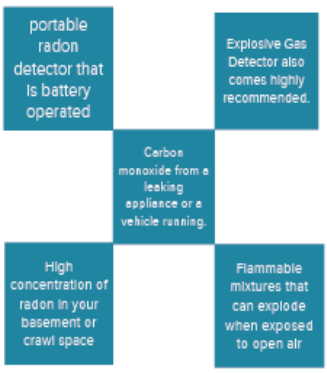


Ideation & Brainstorming

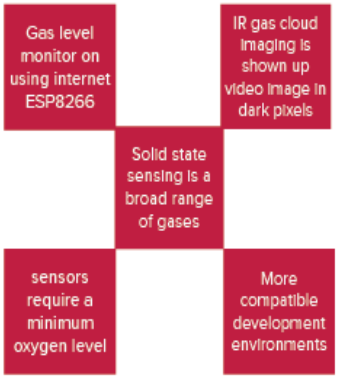
Hemachandhar.N



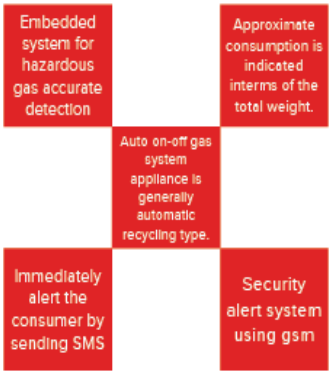
Dharshini . S



Keerthana . R



Charudesna . K



Grouping based on dataset:

1)Data analytics for improved decisions.
2)Cost effective installation
3)High accuracy and reliable technology
4)Prevent free hazards and explosions
5)Monitor the gas concentrate level
6)Ensure the workers health
7)Extremely repeatability

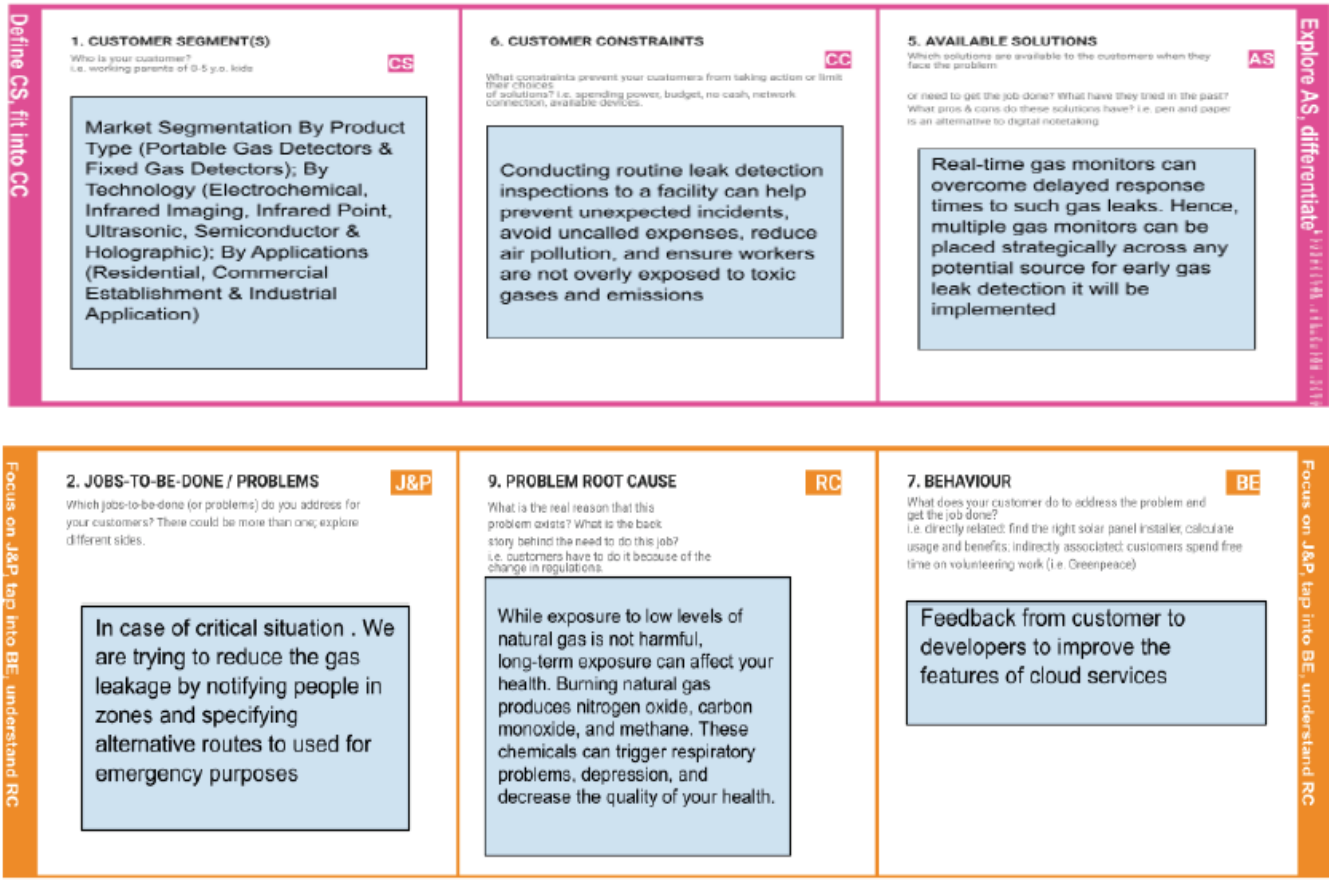
Idea Prioritization:

1) Data analytics for improved decisions.
2) Cost effective installation
3) Reliable technology
4) Extremely precise repeatability
5) Monitor the gas concentrate level
6)Prevent free hazards and explosions
8)High accuracy and reliable technology

Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to besolved)	Gas leakage leads to various accidents resultinginto both financial loss as well as human injuries. In human's daily life, environment gives the most significant impact to their health issues.
2.	Idea / Solution description	**1.** safely detecting any malfunction of a pressurized gas system in order to preventaccumulation of combustible gases so thatdamage or explosion due to such an accumulation of gases is prevented. **2.** gas detection and monitoring system which iseconomical to manufacture and which may be readily installed in conventional trailers ,boats or the like which are normally dependent upon a stored supply of pressurized gas .
3.	Novelty / Uniqueness	**1.** It is fully automatic. **2.** Compact and faster. **3.** Hardware specifically build for thatapplication **4.** Three-level program memory lock.
4.	Social Impact / Customer Satisfaction	**1.** Long run maintanence cost is less. **2.** Sensor has excellent sensitivity combinedwith a quick fast response time **3.** It is possible to get instantaneous results.
5.	Business Model (Revenue Model)	**1.** High quality instruments that can locate those costly leaks by simply aiming it at them. **2.** Included tubular extension and parabola make it possible to probe these difficult locations from a distance to locate the leak.
6.	Scalability of the Solution	**1.** The consumers have to upgrade their safetystandards, act in accordance with statutory requirements Necessity for detection. **2.** The consumers have to upgrade their safety standards, act in accordance with statutory requirements on Environmental commitmentsand most importantly the Basic function being prevented by accidents and protect life andproperty from disasters.

Problem Solution fit



4. REQUIREMENT ANALYSIS

Functional Requirements:

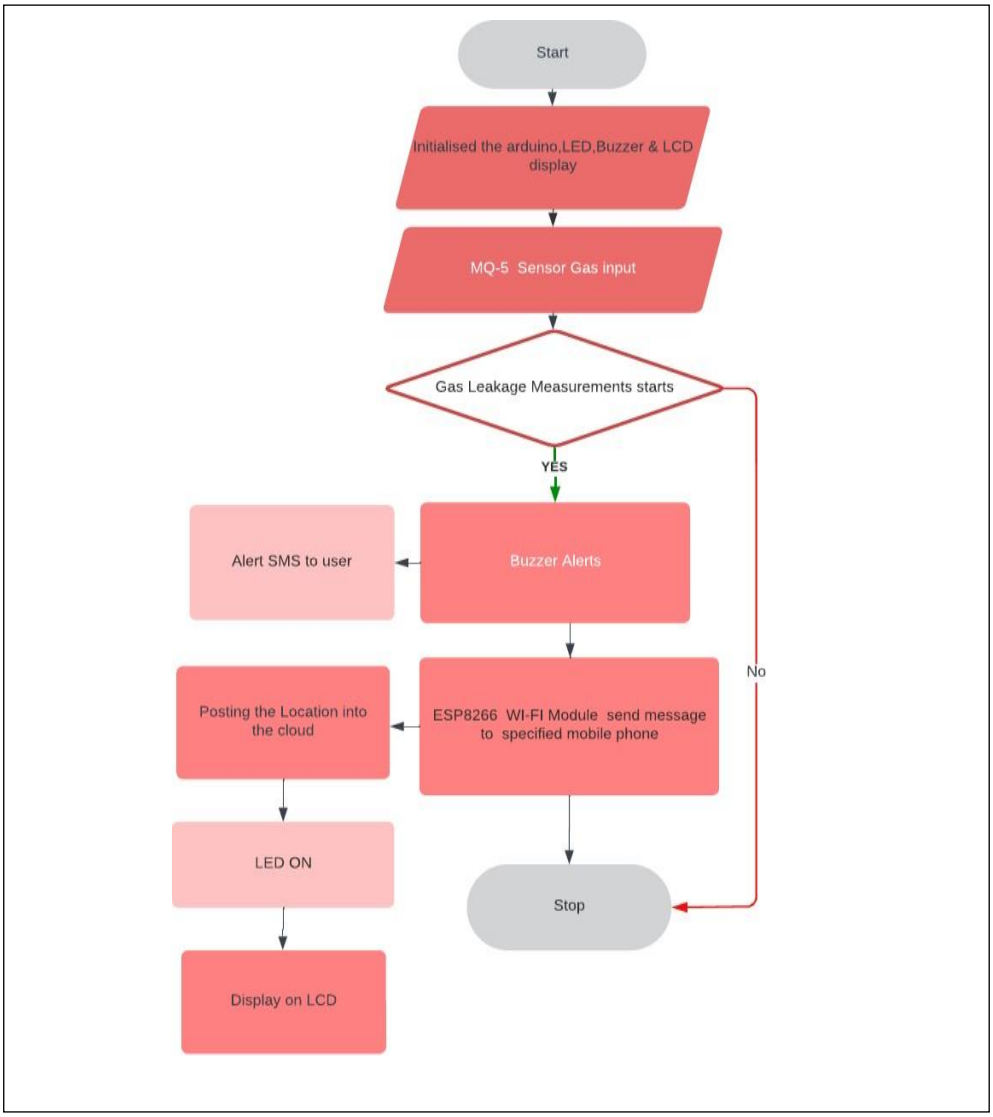
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Methodology	It is a well way to get rapid results in a short time.
FR-2	Impact	Sensor has excellent sensitivity combined with a quickfast response time , possible to get instantaneous results.
FR-3	sensitivity	Specialised of the gas in all similar systems
FR-4	WIFI -Module	Can communicate directly with industrial scientific, consumer technology that is web friendly with no useof shields or any peripherals.
FR-5	operation	The system be operated in android operating system.
FR-6	User Interface	Emergency call, message with application systems

Non-functional Requirements:

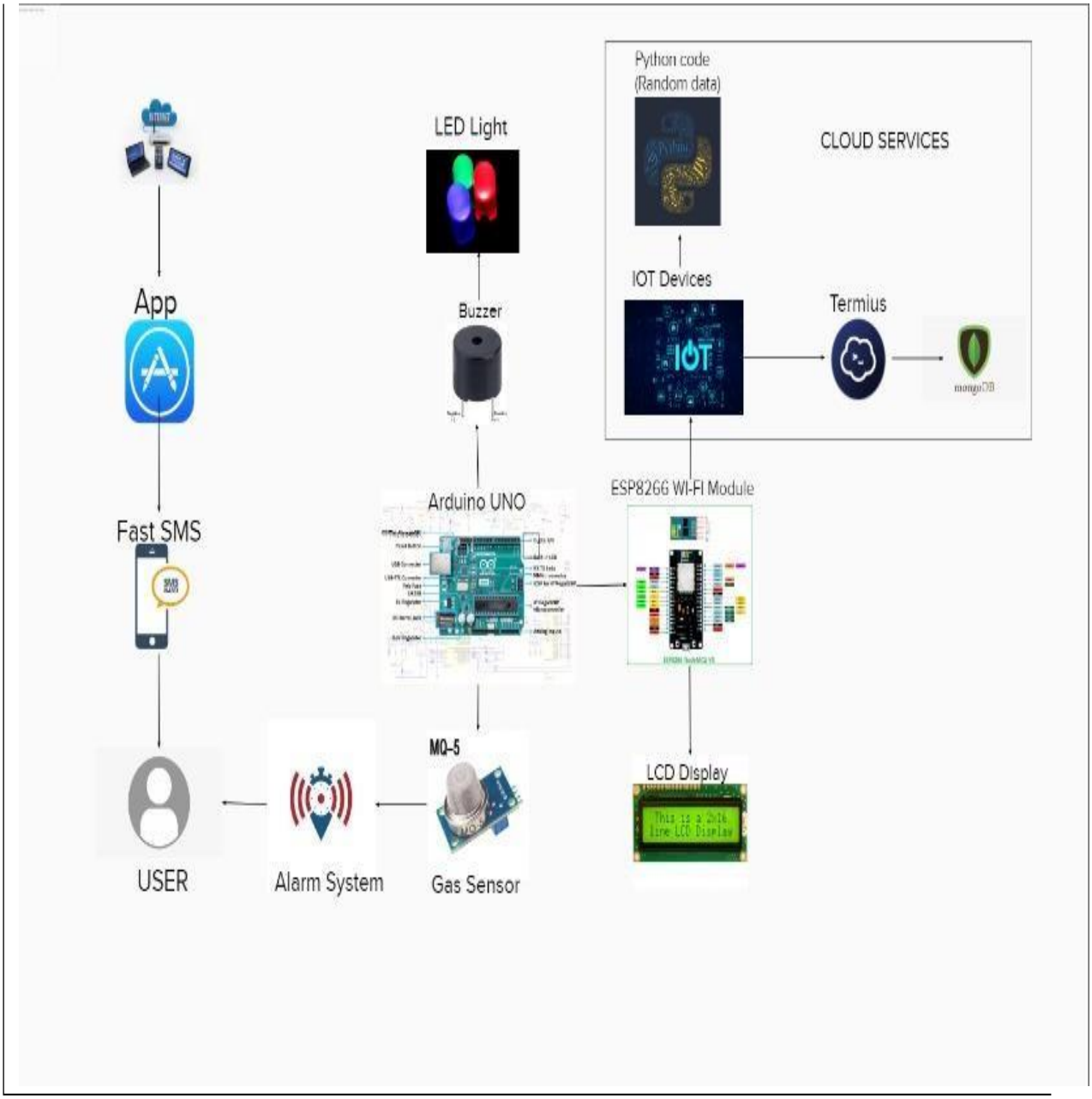
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	It helps prevent the high risk of gas explosions
NFR-2	**Security**	The system should not display the homeownerpersonal information to anyone.
NFR-3	**Reliability**	Unsafe behaviour of personnel has the greatest impact on the probability of gas leakage.
NFR-4	**Performance**	Arduino response time will be fast.
NFR-5	**Availability**	The system should work 24 hours 7 days a week.
NFR-6	**Scalability**	The system interface should be easy andeffective(user-friendly).

5. PROJECT DESIGN

Data flow diagram



Solution & Technical Architecture



6.PROJECT PLANNING & SCHEDULING

User stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Release
Customer (Mobile user)	Registration	USN-1	As The user free Register mobile app makesit easy for you to add the products you own.	I can access my account /dashboard	Sprint-1
		USN-2	As a user I can operate in any location in particular app	I can receive notificatio nthrough g mail	Sprint-1
		USN-3	As a user I can Login in mobile phone activethe extension go to settings and login with phone tab to enable and configure the settings	I can register & access the dashboard with play store Login	Sprint-2
		USN-4	As a user, I can register for the applicationthrough Gmail		Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password		Sprint-1
	Dashboard			Software product accept ythe users	
Customer (Webuser)	Subscrip tion services	USN-6	As a customer to use the web form of thework wide technology	Individual authorized webuser portal	Sprint-1
Customer Care Executive	Email/chat	USN-7	Professional responsible for communicatingThe regarding service expectations	I can servic es for compa ny's succes s	Sprint-2
Administrator	Manager system	USN-8	Descriptive of a product feature from the perspective of the person requesting the features	Express criteria clearly, in simple language	Sprint-1

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority
Sprint-2	Features	USN-7	As a system, the gas leakage pipe should be closed automatically once there it attains the threshold value	5	Medium
Sprint-2	Features	USN-8	As a system, it will indicate that the gas leakage pipe is closed in the LCD screen and send SMS to the registered mobile number.	5	Medium
Sprint-3	Data Transfer	USN-9	As a program, it should retrieve the API key of the IBM cloud to send the details of the system.	2	Low
Sprint-3	Data Transfer	USN-10	As a system, it should send the data of sensor values along with latitudes and longitudes to the IBM cloud	5	Medium
Sprint-3	Data Transfer	USN-11	As a cloud system, the IBM cloud should send the data to NodeRed	2	Medium
Sprint-3	Data Transfer	USN-12	As a system, it should collect the data from the NodeRed and give it to the backend of the mit app.	3	Medium
Sprint-3	Data Transfer	USN-13	As an application, it should display the details of the gas level and other details to the user through the frontend of the mit app.	8	High

Sprint-4	Registration	USN-14	As a user, I must first register my email and mobile number in the website	2	High

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task
Sprint-1	Objective	USN-1	As a system, the gas sensor should detect the gas
Sprint-1	Features	USN-2	As a system, the gas sensor values should be displayed in a LCD screen
Sprint-1	Features	USN-3	As a system, as soon as the detected gas reaches the threshold level, the red color LEDshould be turned ON.
Sprint-1	Features	USN-4	As a system, as soon as the detected gas reaches the threshold level, the siren should be turned ON.
Sprint-2	Focus	USN-5	As a system, it should the send the location where the gas is detected

Sprint Planning & Estimation

Sprint Plan
Analyze The Problem
Prepare An Abstract, Problem
Statement List A Required Object
Needed
Create A Program Code
And Run It Make A
Prototype To Implement
Test With The Created Code And Check The Designed Prototype

7.Sprint delivery schedule

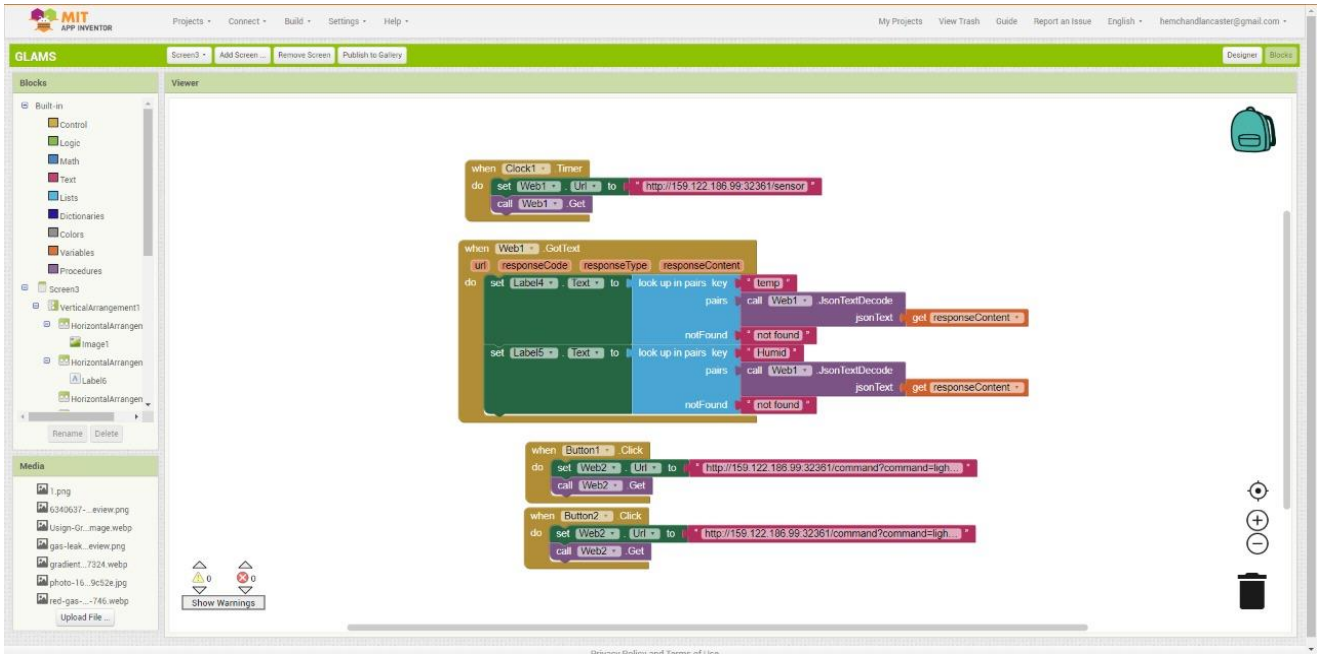
Sprint 1
Sprint 2
Sprint 3
Sprint 4
We are Developing the code in the schedule

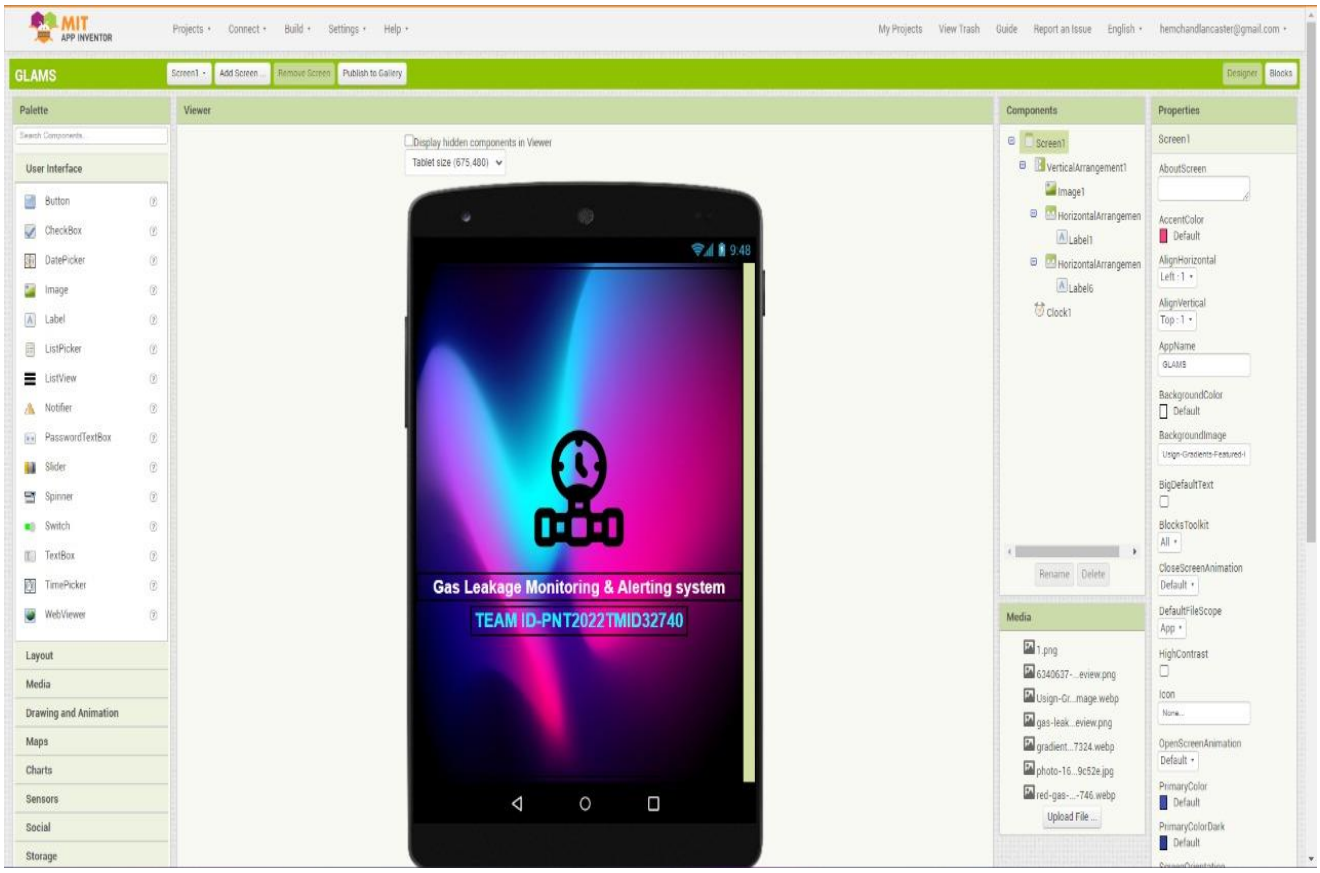
8.Components

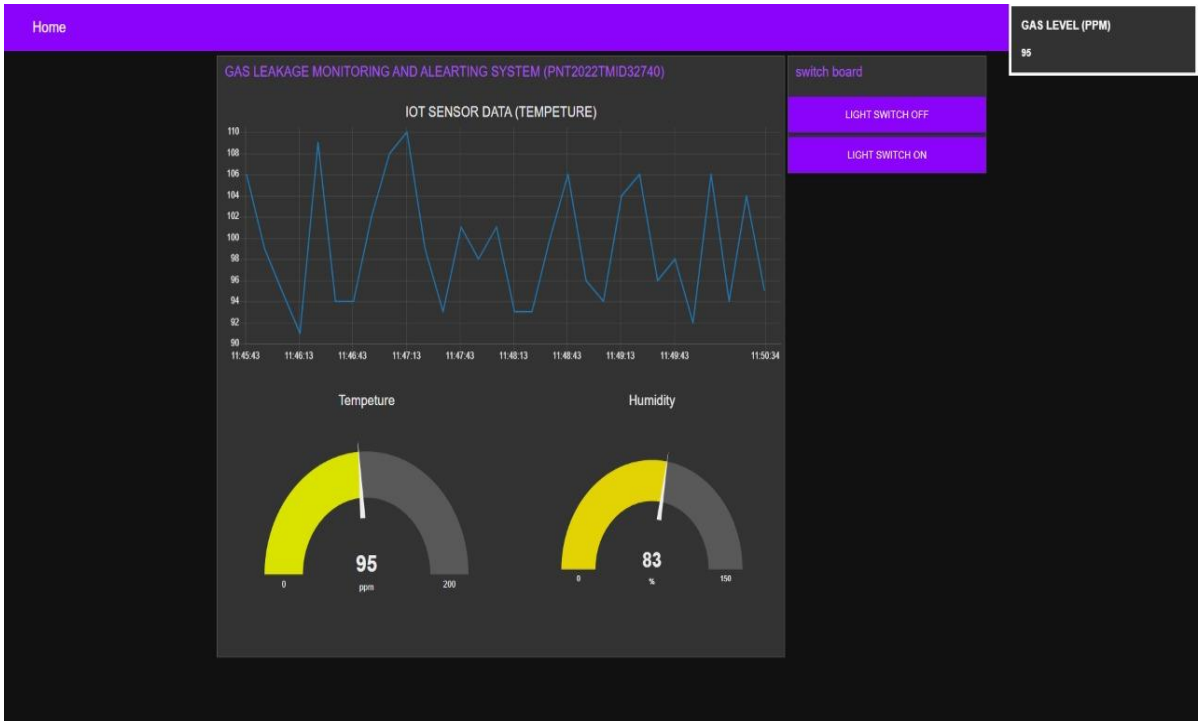
S.No.	Name of the component	Quantity
1.	Arduino UNO R3	1
2.	Breadboard	1
3.	LED	2
4.	Resistor	5
5.	Buzzer	1
6.	Gas sensor	1
7.	LCD 16*2	1

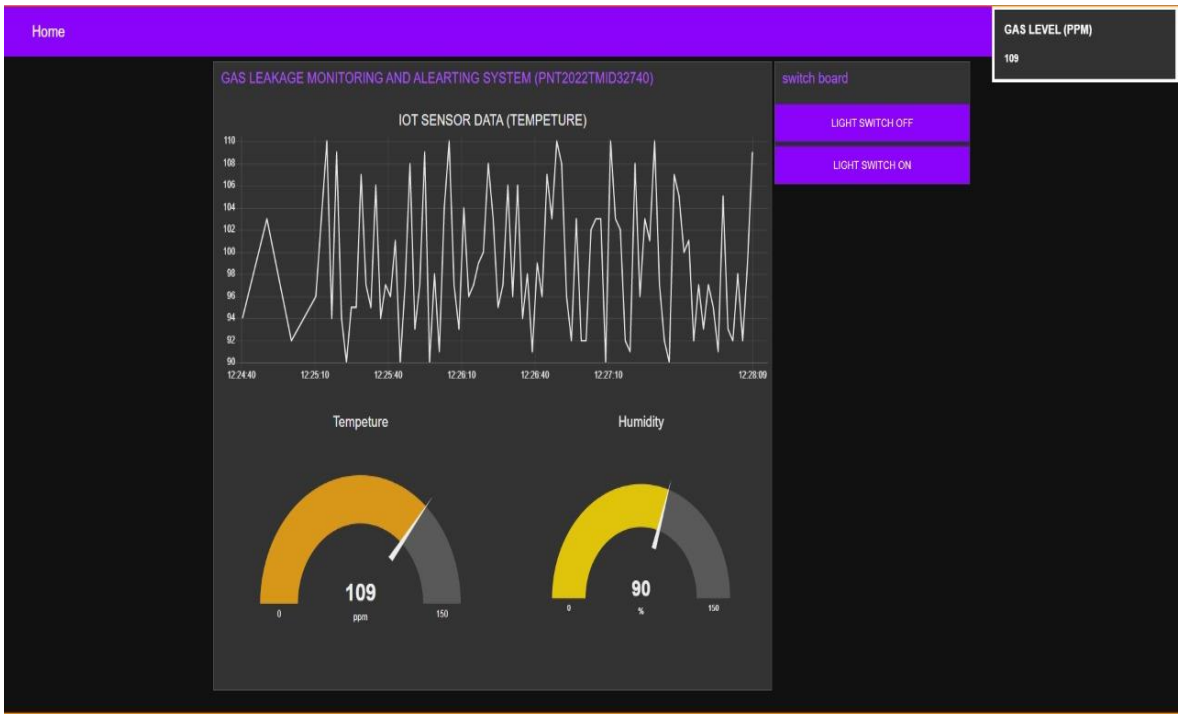
9.MIT app inventer

Testing






Home
GAS LEVEL (PPM)
95
GAS LEAKAGE MONITORING AND ALEARTING SYSTEM (PNT2022TMID32740)
IOT SENSOR DATA (TEMPETURE)
switch board
LIGHT SWITCH OFF
LIGHT SWITCH ON
Tempeture
Humidity
95
ppm
83
%


Home
GAS LEVEL (PPM)
109
GAS LEAKAGE MONITORING AND ALEARTING SYSTEM (PNT2022TMID32740)
IOT SENSOR DATA (TEMPETURE)
switch board
LIGHT SWITCH OFF
LIGHT SWITCH ON
Tempeture
Humidity
109
ppm
90
%

10.Result:

The system can be taken as a small attempt in connecting the existing primary gas detection methods to a mobile platform integrated with IoT platforms. The gases are sensed in an area of 1m radius of the rover and the sensor output data are continuously transferred to the local server. The accuracy of sensors is not up to the mark thus stray gases are also detected which creates an amount of error in the outputs of the sensors, especially in case of methane. Further the availability and storage of toxic gases like hydrogen sulphide also creates problems for testing the assembled hardware. As the system operates outside the pipeline, the complication of system maintenance and material selection of the system in case of corrosive gases is reduced. Thus, the system at this stage can only be use data primary indicator of leakage inside a plant.

11.Advantages/Disadvantages:

Advantages:

1. Get real-time alerts about the gaseous presence in the atmosphere.
2. Prevent fire hazards and explosions.
3. Supervise gas concentration levels.
4. Ensure worker's health.
5. Real-time updates about leakages.
6. Cost-effective installation.
7. Data analytics for improved decisions.
8. Measure oxygen level accuracy.
9. Get immediate gas leak alerts.

Disadvantages:

1. It requires air or oxygen to work.
2. It gets reacted due to heating of wire.
3. It can be poisoned by lead, chlorine and silicon

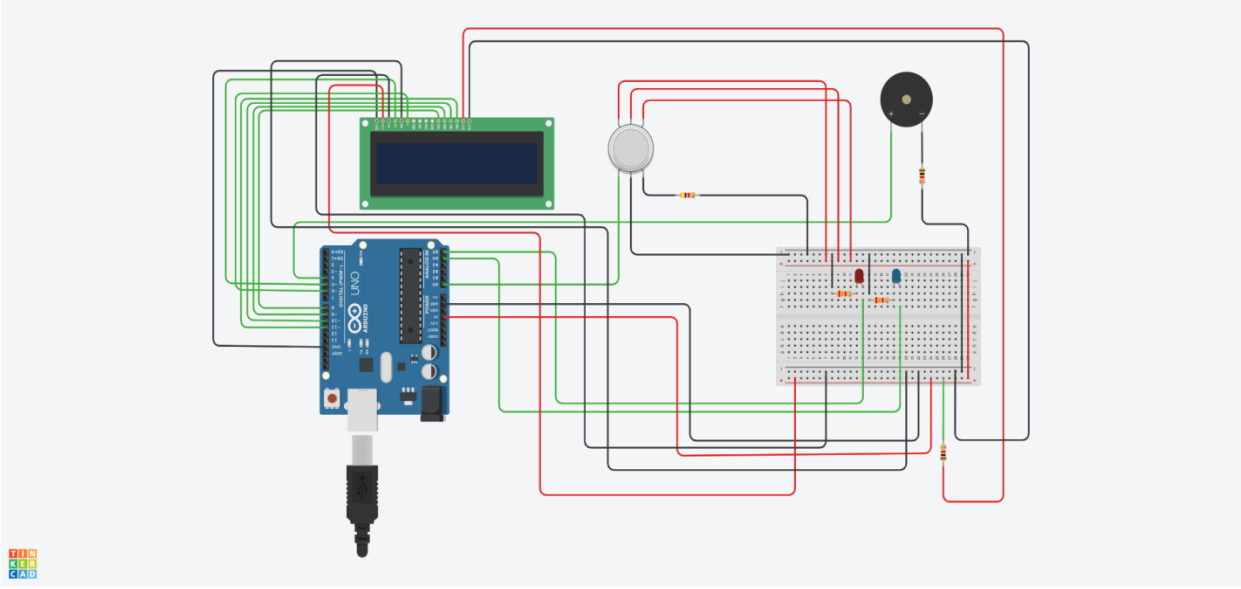
12.CONCLUSION

After completing this project, can conclude that the gas leak detection in the project system is amazing. Usefully applicable in industry and household. In dangerous situations we can save lives with this system. The GSM module displays an alert. A sensor node detects gases su.,h as CO2, oxygen, propane. Estimated transmission distance and power consumption are obtained. The sensor is built using simple techniques and the area of the Arduino UNO micro-controller.

13 .FUTURE SCOPE

This monitoring system can be further improved by using Bluetooth instead of GSM to send alerts to the user, which is supported by another real-time application. For industrial purposes, mobile robots can be developed to detect multiple gas concentrations.

14.APPENDIX



15..CODING & SOLUTIONING:

Source code

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

#Provide your IBM Watson Device Credentials
organization = "qijw2u"
deviceType = "NODEMCU"
deviceId = "glmas1_01"
authMethod = "token"
authToken = "123456789"

# 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()
{
                digitalWrite(greenled,
                HIGH);
                digitalWrite(redled,LO
                W); noTone(buzzer);
                lcd.clear();
                lcd.setCursor(0,0);
                lcd.print("SAFE"
                ); delay(700);
                lcd.clear();
                lcd.setCursor(0,1);
                lcd.print("ALL CLEAR");
                delay(700);
                }
                }
```

ibm code 1.py - C:\Users\HemchandLancaster\AppData\Local\Programs\Python\Python37-32\ibm code 1.py (3.7.0)

File Edit Format Run Options Window Help

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

#Provide your IBM Watson Device Credentials
organization = "qljw2u"
deviceType = "NODEMCU"
deviceId = "glmas1_01"
authMethod = "token"
authToken = "12345678"

# Initialise 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()
```

Ln: 8 Col: 0

"Python 3.7.0 Shell"

File Edit Shell Debug Options Window Help

```
led is off
Published Temperature = 105 C Humidity = 85 % to IBM Watson
Published Temperature = 100 C Humidity = 92 % to IBM Watson
Command received: lightoff
led is off
Published Temperature = 101 C Humidity = 92 % to IBM Watson
Command received: lighton
led is on
Published Temperature = 92 C Humidity = 100 % to IBM Watson
Published Temperature = 97 C Humidity = 74 % to IBM Watson
Command received: lighton
led is on
Published Temperature = 93 C Humidity = 77 % to IBM Watson
Published Temperature = 97 C Humidity = 85 % to IBM Watson
Published Temperature = 95 C Humidity = 79 % to IBM Watson
Published Temperature = 91 C Humidity = 66 % to IBM Watson
Published Temperature = 105 C Humidity = 95 % to IBM Watson
Command received: lighton
led is on
Published Temperature = 93 C Humidity = 80 % to IBM Watson
Published Temperature = 92 C Humidity = 72 % to IBM Watson
Command received: lighton
led is on
Published Temperature = 99 C Humidity = 100 % to IBM Watson
Command received: lighton
led is on
Published Temperature = 92 C Humidity = 74 % to IBM Watson
Published Temperature = 99 C Humidity = 85 % to IBM Watson
Published Temperature = 109 C Humidity = 90 % to IBM Watson
Published Temperature = 97 C Humidity = 95 % to IBM Watson
Published Temperature = 99 C Humidity = 79 % to IBM Watson
Published Temperature = 99 C Humidity = 93 % to IBM Watson
Published Temperature = 102 C Humidity = 89 % to IBM Watson
Published Temperature = 106 C Humidity = 69 % to IBM Watson
Published Temperature = 108 C Humidity = 98 % to IBM Watson
Published Temperature = 96 C Humidity = 76 % to IBM Watson
Published Temperature = 103 C Humidity = 83 % to IBM Watson
Published Temperature = 90 C Humidity = 97 % to IBM Watson
Published Temperature = 91 C Humidity = 93 % to IBM Watson
Published Temperature = 102 C Humidity = 75 % to IBM Watson
Published Temperature = 93 C Humidity = 65 % to IBM Watson
Published Temperature = 100 C Humidity = 62 % to IBM Watson
Published Temperature = 96 C Humidity = 60 % to IBM Watson
Published Temperature = 103 C Humidity = 79 % to IBM Watson
Published Temperature = 105 C Humidity = 96 % to IBM Watson
Published Temperature = 96 C Humidity = 80 % to IBM Watson
Published Temperature = 93 C Humidity = 67 % to IBM Watson
Published Temperature = 104 C Humidity = 79 % to IBM Watson
Published Temperature = 100 C Humidity = 71 % to IBM Watson
Published Temperature = 93 C Humidity = 85 % to IBM Watson
Published Temperature = 104 C Humidity = 62 % to IBM Watson
Published Temperature = 104 C Humidity = 86 % to IBM Watson
Published Temperature = 92 C Humidity = 80 % to IBM Watson
Published Temperature = 105 C Humidity = 71 % to IBM Watson
Published Temperature = 108 C Humidity = 98 % to IBM Watson
Published Temperature = 92 C Humidity = 89 % to IBM Watson
Published Temperature = 107 C Humidity = 74 % to IBM Watson
Published Temperature = 97 C Humidity = 74 % to IBM Watson
Published Temperature = 96 C Humidity = 64 % to IBM Watson
```

Ln: 151 Col: 0

GitHub link:

https://github.com/IBM-EPBL/IBM-Project-14807-1659590201/tree/main

Demo link:

https://youtu.be/GSguGRjCZCg

Tinkercad link:

https://www.tinkercad.com/things/9cqXHiCCyhX