# NAALAIYA THIRAN PROJECT – 2022

# IOT BASED GAS LEAKAGE MONITORING AND ALERTING SYSTEM

**A PROJECT REPORT**

**Submitted by**

| | |
|---|---|
| MANIKANDAN S | 110519106009 |
| KEERHI J | 110519106008 |
| RUPAVATHI S | 1105191060015 |
| LAVANYA V | 110519106701 |

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

**GOJAN SCHOOL OF BUSINESS AND TECHNOLOGY**

**EDAPALAYAM,REDHILLS.**

**CHENNAI-600 052**

# ANNA UNIVERSITY: CHENNAI 600 025
# BONAFIDE CERTIFICATE

Certified that this report **“IOT BSASED GAS LEAKAGE MONITORING AND ALERTING SYSTEM FOR INDUSTRIES”** is the Bonafide work of **MANIKANDAN S**( 110519106009), **KEERTHI J**(110519106008),**RUPAVATHI S**(1105191060015),**LAVANYA V** (110519106701)who carried out research under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.Professional Readiness for Innovation, Employability and Entrepreneurship project offered by IBM and AnnaUniversity ,Chennai.

| | |
|---|---|
| **SIGNATURE OF HOD** | **SIGNATURE OF SUPERVISOR** |
| **Mrs. S.KOKILA** | **Mr.S.VENKAT** |
| Assistant professor & Head Of the Department | Assistant professor |
| Department of ECE | Department of ECE |
| Gojan School Of Business and Technology | Gojan School Of Business and Technology |
| Chennai-600 025 | Chennai-600 025 |

# PROJECT REPORT FORMAT

# 1.INTRODUCTION

## Project Objective

By the end of this project you will:

- Gain knowledge of Watson IoT Platform.
- Connecting IoT devices to the Watson IoT platform and exchanging the sensordata.
- Gain knowledge on IBM Cloudant DB
- Explore Python client libraries of Watson IoT Platform.
- Explore Python library for integrating OpenCV for accessing the Live CameraInput
- Scan the QR code in live streaming and retrieve the QR code details
- Gain knowledge of web application development.
- Gain knowledge of storing the data in Cloudant DB
- Generating QR codes with the required data.

## Project Flow

- The parameters like hazardous gas levels, fire, humidity, andtemperature data are published to the Watson IoT platform
- The device will subscribe to the commands from the application andtakedecisions accordingly to switch on the rainwater sprinkler in case of emergencies
- Sensor data is visualized in the Web Application

## To accomplish this, we have to complete all the activities and taskslisted below:

- Create and configure IBM Cloud Services
  - Create IBM Watson IoT Platform and Device
  - Create Node-RED service
- Develop the Python Script
  - Develop the Python Script
- Develop a web Application using Node-RED Service.
  - Develop the Web application using Node-RED
  - Testing the Web UI by giving the required inputs

# 2 . LITERATURE SURVEY

## LITERATURE 1:

### IOT BASED HOME SAFETY GAS LEAKAGE DETECTION AND AUTOMATIC BOOKING SYSTEM:

Internet of things try towards making life less complex what's more, quicker via robotizing the whole little errands related with the life of human. Today, everything is getting keen because of the innovative advancement, for example, of IOT. As IOT is valuable for robotizing the assignments, the upside of IOT can likewise be far reaching for improving the helpful security strategies. Security plays asignificant role while constructing home, buildings, industries as well as towns. The enlarged focus of certain gases in the environment can be exceptionally unsafe, in recent time, everyone needs a facility which reduces time and effort and expect their work to be as easy as possible. One such region where man wants to get the work quicker and simpler is cooking. Most ordinarily LPG is utilized for cooking reason which was presented by Dr.Walter Snelling. It is a amalgamation of propane and butane alongside soaked substance notwithstanding unsaturated hydrocarbon substance. Gas undertakings utilizes SMS, IVRS or Online reserving for the LPG, which is tedious strategies in individuals' day by day life. However, due to fast nature and high competition, today people look for smarter way of operations than tedious and mechanical as well as manual routine. As such, booking gas has also become one of the tasks where one has tendency to either postpone or forget its booking due to busy schedule and lack of time.

Usually in home or industries, most of the disaster happens due to gas leakages, which leads to several accidents and also causes human life. In order to handle such situation, the proposed gas leakage detection and monitoring system is developed and put forth in this paper. In this layout MQ-6 sensor is used to detect and sense the gas leakage and the temperature sensor is also placed to reduce the false deduction. This proposed system is not only capable of Sensing or detecting the gas leakages as well as alerting the user about the gas leakage by buzzer alarm and also displaying alert message in LCD display simultaneously switch on the exhaust fan and start the stepper motor, external coupling is made to turn off the gas regulator. PIR sensor also placed in the home to notify about the human presence.

## LITERATURE 2:

## ARDUINO BASED GAS LEAKAGE DETECTION SYSTEM USING IOT:

It has become important factor nowadays to bring the technology into our home and office. By making the place smart, the day-to- day activities are becoming more and easier. The development of home automation has become mandatory in homes as people are moving towards to the smart home concepts. The supply gas will also be stopped with the use of solenoid, ultimately preventing the chance of accident. This system will not only able to detect the leakage of gas but also alerting through audible alarms. Presence of excess amounts of harmful gases in environment then this system can notify the user. System can notify to society admin about the condition before mishap takes place through a message. This system will not only able to detect the leakage of gas but also alerting through audible alarms. Presence of excess amounts of harmful gases in environment then this system can notify the user. The people in the neighbors can also be included in case of an emergency. LPG gas sensor is used for input. A buzzer is connected along with the circuit to indicate the use of the output.

## LITERATURE 3:

### IOT BASED DETECTION OF LEKAGAES IN GAS PIPES:

The Internet of things (IOT) is the network of electronic devices, which are related to embedded systems and also other domains through the internet. Liquids and gases are mostly transported in pipelines like oil, natural gases, biofuels and water. It is necessary to check whether the pipes are good enough without cracks. The cracks may lead to disasters. There is a real life incident which needs to be taken seriously. This project might help to get aware of it. The cruel incident took place in Tlahuelilpan town situated in Mexican state. On 18, Jan 2019 a pipeline transporting gasoline exploded taking the life of 96 people and a lots of people gets injured. Pipeline monitoring, control, operation and maintenance are very important activities, which have evolved considerably. The detection and behavior of leaks has deserved special attention by different researchers.
This paper deals with the detection of leakages in gas pipes and thusreducing the man power

## LITERATURE 4:

### A SMART NATURAL GAS LEAKAGE DETECTION AND CONTROL SYSTEMFOR GAS DISTRIBUTION COMPANIES OF BANGLADESHUSING IOT:

This project proposes a smart mobile based model of gas leakage detection and control for gas distribution system of Bangladesh using IoT, called as smart natural gas leakage detection and control system (SNLDCS). The proposed SNLDCS has been implemented in both software and hardware modules. The existing researches are about Liquefied Petroleum Gas (LPG) leakage detection that are used for cylinder gas. Hence, these models are not suitable for gas distributions companies of Bangladesh where natural gasleakage is being controlled from remote places. But the proposed model can quickly detect natural gas leakage by continuous monitoring and can control gas leakage by a smart phone from anywhere. The experimental results confirm that, implementation of SNLDCS model in gas distribution system in

Bangladesh can provide the quickest detection and rapid resolve of gas leakage. As a result, it will increase safety, decreases system loss and reduces Greenhouse Gas (GHG) emission in the air.

## LITERATURE 5:

### DETECTION OFGAS LEAKAGE IN POLYMER INDUSTRIES USING IOT:

Gases leaked from polymer and carbide industries are very harmful to all living things. Major disaster happened at Bhopal on December 3, 1984. Recently an industrial accident occurred at LG polymers chemical plant in the Vishakapattinam. As per the National Disaster Response Force (NDRF), the death toll was 11, and more than 1,000 people became sick after being exposed to the gas. To prevent from these types of accidents, safety system should build in high quality standards. Safety should be ensured byall levels. To incorporate technology in the Safety System, Internet of Things (IOT) technology is used to detect the gas leakage and prevent the disaster before it happened. Internet of Things [IOT] is asystem of interrelated computing devices without human – human or human– computer interaction. IOT is used to automating the daily tasks, the benefits of IOT can also be extendedfor enhancing the existing safety standards. Safety is the most important criterion while designing polymer industries. The spread of highly concentrated gases in the atmosphere can produce extremely dangerous condition. These gases might be flammable at certain temperature and humidity conditions, toxic after exceeding the specified concentrations limits or even a contributing factor in the airpollution of an area leading to problems such as smoke and reduced visibility which can in turn cause several accidents and also have adverse effect on thehealth of people.

## PROBLEM STATEMENT

1. How does the Problem affect?

The main consequences are the emission of flammable substances into the environment, fire, explosion, and distribution of toxic substance. Serious risks of carbon monoxide poisoning in people and animals.

2. What is the issue?

Improper use of gas furnace, stove,

orappliance.Fault in the gas pipeline

3. What is the impact of the issue?Create

potential hazards for the workers in the

industryCreate Pollution to environment

Diseases prone

4. What would happen if this problem isnot solved?

Emission of toxic gases which is harmful to the

environment Leads to explosion or other types of fire

hazard Rusted and poor pipelines creates hazard**.**

5. What would happen if this problem is fixed?

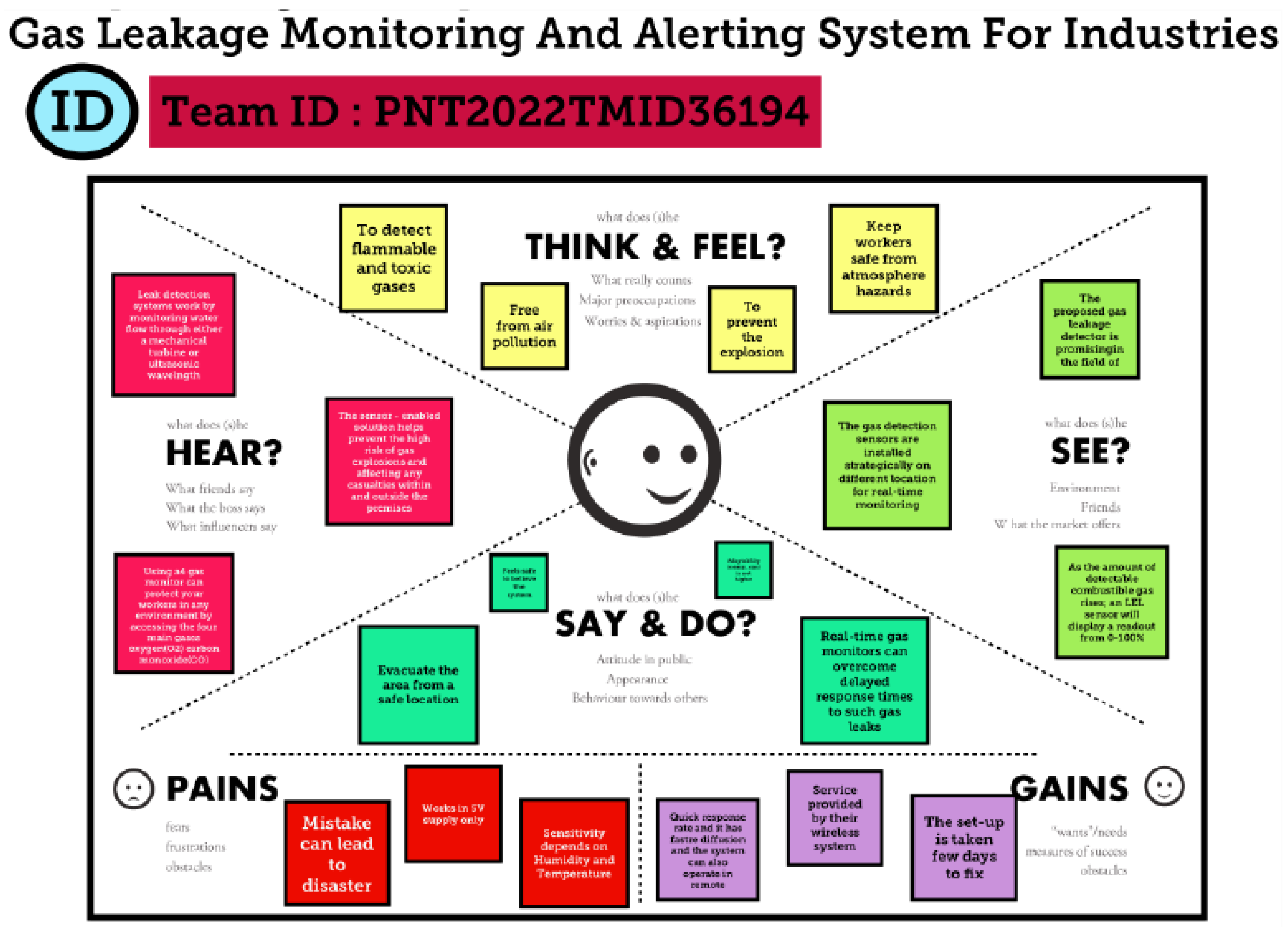
Identifying and solving the issue parameters like pollution and spreading of poisonous gas ,we canfix the problem.

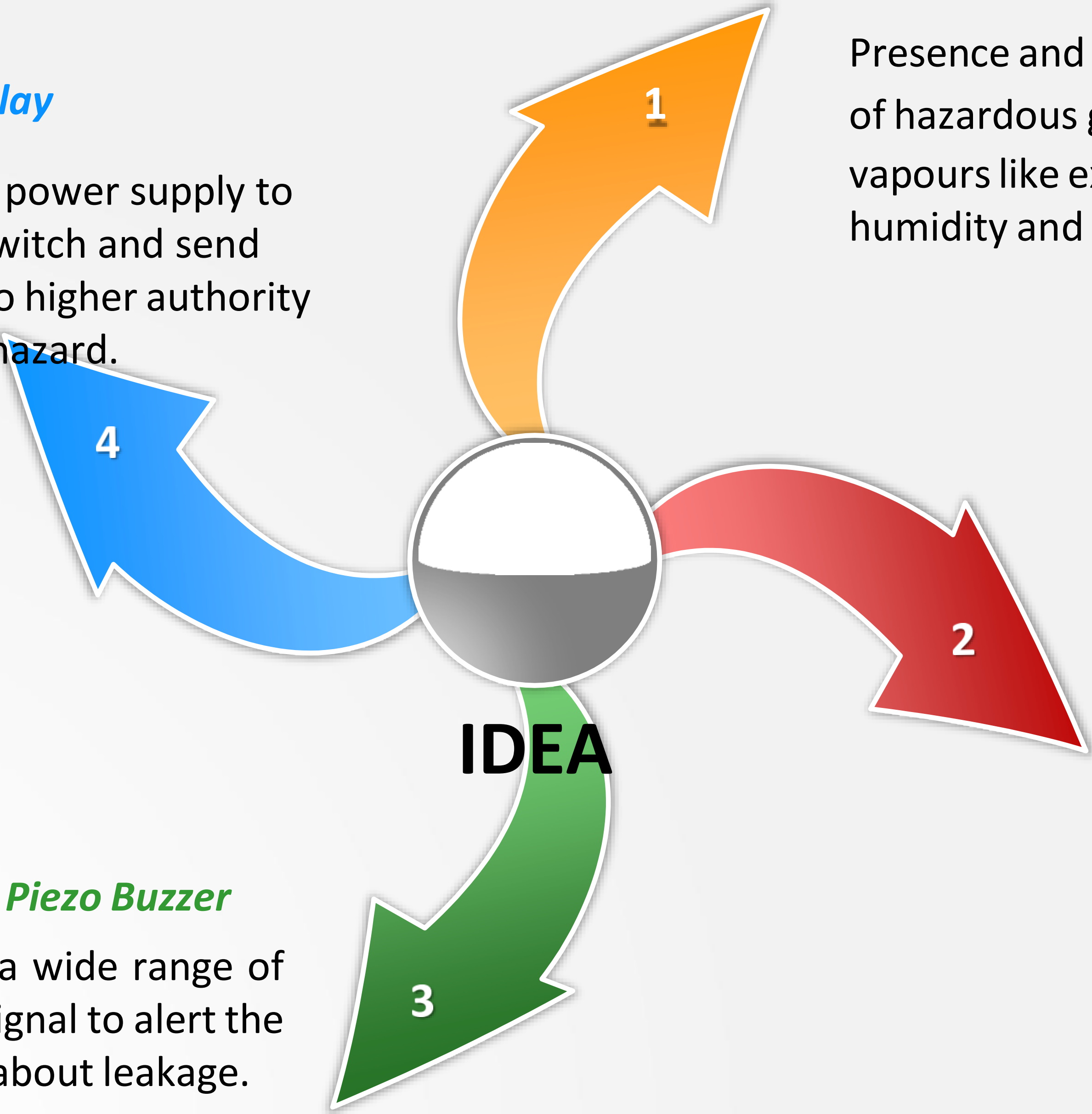
6. Why it is important to fix the problem?

Emission of harmful gases leads to varying of environmental weather and affectsthe earth layerwhich is protected by the UV radiation

# 3． IDEATION AND PROPOSED SOLUTION

## Empathy map Ideation:


Gas Leakage Monitoring And Alerting System For Industries
ID
Team ID : PNT2022TMID36194
what does (s)he
THINK & FEEL?
What really counts
Major preoccupations
Worries & aspirations
To detect flammable and toxic gases
Free from air pollution
To prevent the explosion
Keep workers safe from atmosphere hazards
what does (s)he
HEAR?
What friends say
What the boss says
What influencers say
Leak detection systems work by monitoring water flow through either a mechanical turbine or ultrasonic wavelength
The sensor - enabled solution helps prevent the high risk of gas explosions and affecting any casualties within and outside the premises
Using a4 gas monitor can protect your workers in any environment by accessing the four main gases oxygen(O2) carbon monoxide(CO)
what does (s)he
SEE?
Environment
Friends
What the market offers
The proposed gas leakage detector is promisingin the field of
The gas detection sensors are installed strategically on different location for real-time monitoring
As the amount of detectable combustible gas rises; an LEL sensor will display a readout from 0-100%
what does (s)he
SAY & DO?
Attitude in public
Appearance
Behaviour towards others
Feels safe to believe the system
Evacuate the area from a safe location
Real-time gas monitors can overcome delayed response times to such gas leaks
PAINS
fears
frustrations
obstacles
Mistake can lead to disaster
Works in 5V supply only
Sensitivity depends on Humidity and Temperature
GAINS
"wants"/needs
measures of success
obstacles
Quick response rate and it has faster diffusion and the system can also operate in remote
Service provided by their wireless system
The set-up is taken few days to fix



*Gas Detecting Sensor*

Presence and concentration of hazardous gases and vapours like explosive gases , humidity and odour.

*Harmful gas detected*

Unsafe gas levels pose an immediate danger to workers o equipmen

*Actuates Piezo Buzzer*

Produce a wide range of audible signal to alert the workers about leakage.

*Use Relay*

Cut off power supply to main switch and send email to higher authority about hazard.

# Proposed Solution

**Proposed Solution:**

| S.No. | Parameter | Description |
|---|---|---|
| 1. | **Problem Statement (Problem to be solved)** | ➢ Leaks are considered very dangerous since they can build into an explosive concentration So the proposed solution is used for the development for an efficient system & an application that can monitor and alert the workers |
| 2. | **Idea / Solution description** | ➢ In several areas, the gas sensors will be integrated to monitor the gas leakage<br>➢ The proposed system takes an automatic control action after the detection of 0.001% of LPG leakage.<br>➢ This automatic control action provides a mechanical handle driven by stepper motor for closing the valve<br>➢ We are increasing the security for human by using the combination of a relay and the stepper motor which will shutdown the electric power of the house .Also by using a GSM module, we are sending an alert message by SMS (Short messaging services) to warn the |

| | | |
|---|---|---|
| | | users about the LPG leakage and a buzzer is provided for alerting the neighbors in case of the absence of the users about the LPG leakage<br>➢ The main advantage of this system over the manual method is that, it does all the process automatically and has a quick response time. |
| 3. | **Novelty / Uniqueness** | ➢ User friendly<br>➢ Pioneering study of natural gas detection with CCD in visible range |
| 4. | **Social Impact / Customer Satisfaction** | ➢ Cost efficient<br>➢ Easy installation and provide efficient results. |
| 5. | **Business Model (Revenue Model**) | ➢ With widespread deployment of the urban natural gas industry, the energy security is now becoming one of the priorities in practice.<br>➢ The gas leakage model was applied to analyse the pressure, temperature and flow rate of gas leakage over time under both the steady-state and dynamic conditions.<br>➢ As the product usage can be understood by everyone, it is easy for them to use it properly for their safest organization. |
| 6. | **Scalability of the Solution** | ➢ Establishing fast communication equipment with the nearest fire station and other relief station to have the fastest response in case of an accident.<br>➢ Even when the gas leakage is more, the product sense the accurate values and alerts the workers effectively |

# Problem Solution Fit

| 1.CUSTOMER SEGMENTS | 6. CUSTOMER CONTRAINTS | 5. AVAILABLE SOLUTIONS |
|---|---|---|
| ✓ For industry owner-Ensuring the safety of workers is the main thing.<br>✓ Sometimes it is hard to identify the area where the leakage occurs.<br>✓ The detection of leakage prevents the loss of lives | ✓ Proper maintenance should be taken atleast once in a month and this prevents the customers from taking actions in gas leakage problem. | ✓ Usage of sensors to sense gas Leakage.<br>✓ Buzzer to indicate the leakage.<br>✓ GSM module helps us to get notification when there is a gas leakage. |

## 2. JOBS-TO-BE-DONE / PROBLEMS

- ✓ Capability of the device to withstand in harsh environment is questionable.
- ✓ Due to network issue data couldn't be uploaded to the cloud at all times.

## 9. PROBLEM ROOT CAUSE

- ✓ Sometimes sensor doesn't work properly which can cause the major problem.
- ✓ Location of the device installation and the network plan used by the user are the root cause of the network issue.

## 7. BEHAVIOUR

- ✓ Network issue is very common as most of the industries are located at the country side. Here contact both the developers and the service providers.
- ✓ To determine the gas characteristics and solve the issue, they will locate the leak and identify the warning.

## 3.TRIGGERS

- ✓ Accidents due to gas leakages and loss of physical property and life.
- ✓ Safe precautions for the workers to work without fear.

## 4.EMOTIONS:Before/After

- ✓ Before the action is taken the user feels deceived and cheated.
- ✓ After the problem is resolved user feels the sincerity of the developer

## 10. YOUR SOLUTION

- ✓ Low cost IOT based device that can be easily accessed and fixed by people.
- ✓ Network strength must be boosted in the device.
- ✓ Device can be manufactured in multiple standards based on the environment.

## 8. CHANNELS OF BEHAVIOUR

**ONLINE**

- ✓ Monitor the status of the sensors
- ✓ Notification incase of any gas leakage.

**OFFLINE**

- ✓ Prevent physical damage to sensor.
- ✓ Provide proper network and power supply to sensors.
- ✓ Complaint letters.

# 4 .REQUIREMENT ANALYSIS

## Functional
## Non-Functional Requirements

**Functional Requirements:**

Following are the functional requirements of the proposed solution.

| FR No. | Functional Requirement (Epic) | Sub Requirement (Story / Sub-Task) |
|---|---|---|
| FR-1 | User Registration | Registration through Form<br>Registration through Gmail<br>Registration through LinkedIN |
| FR-2 | User Confirmation | Confirmation via Email<br>Confirmation via OTP |
| FR-3 | Hardware Requirement | Optical<br>Soil<br>Ultra-Sonic Flow Meter |
| FR-4 | Software Requirement | Flow change<br>Pressure point<br>Statistic |
| FR-5 | User Welfare | Calibration<br>No Poisoning of the Sensor<br>Reliable in All Environmental Conditions<br>Easy to Use |

**Non-functional Requirements:**

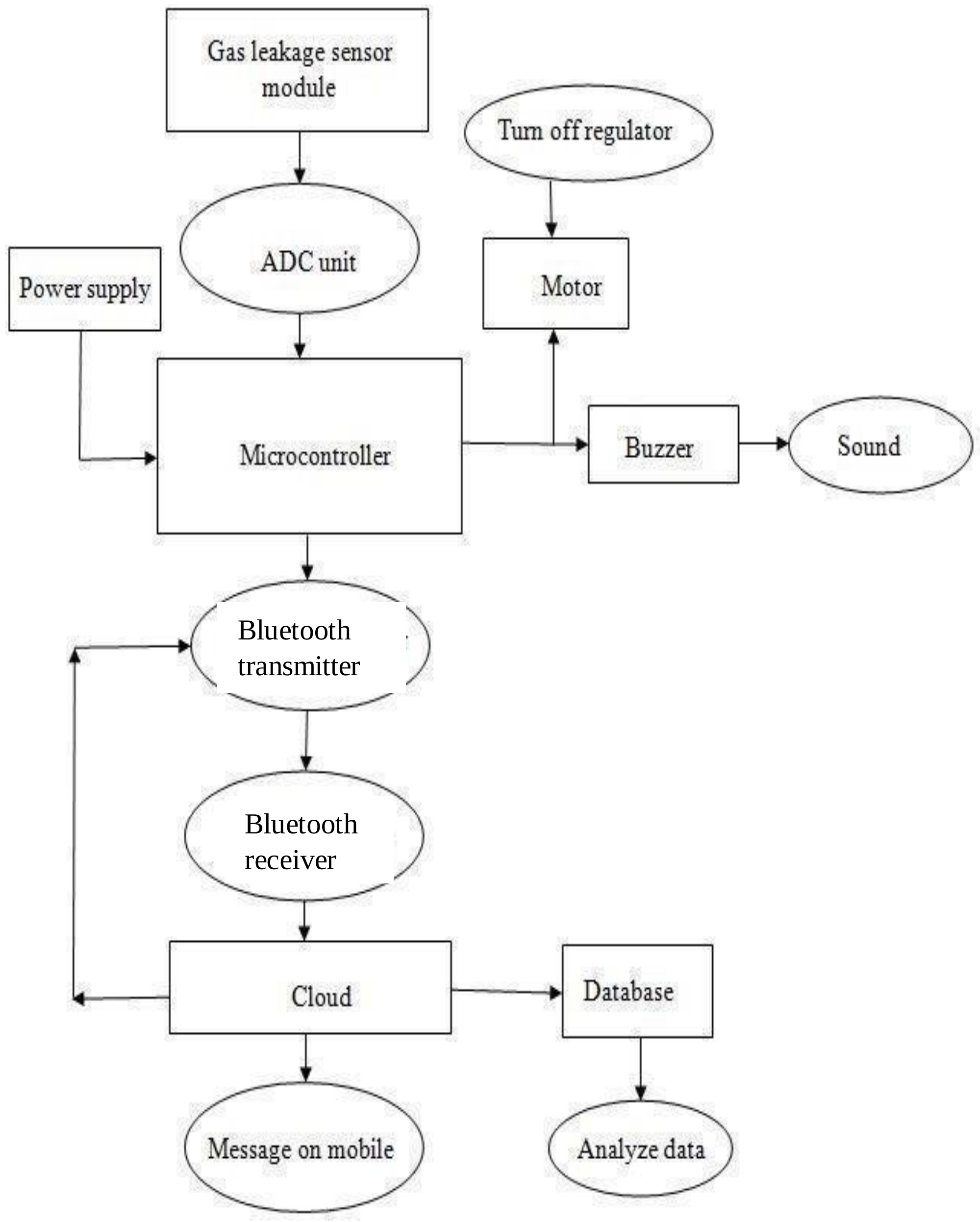
Following are the non-functional requirements of the proposed solution.

| FR No. | Non-Functional Requirement | Description |
|---|---|---|
| NFR-1 | **Usability** | The sensor-enabled solution helps prevent the high risk of gas explosions and affecting any casualties within and outside the premises |
| NFR-2 | **Security** | The device is intended for use in household safety where appliances and heaters that use natural gas and liquid petroleum gas (LPG) may be a source of risk. |
| NFR-3 | **Reliability** | Gas Leakage Detection System (GLDS) can detect leakage at homes, commercial premises or factories. GLDS detects the leakage soon after it happened and sends users an immediate alarm on the incident. |

| | | |
|---|---|---|
| NFR-4 | **Performance** | The Gas Leakage Detector is a wall mounted device fitted close to the floor level with an alarm setting at 20% of lower explosive limit. Whenever there is a leak, the in-built sensor detects and alerts the user in less than 5 minutes, much before it can cause any accidents |
| NFR-5 | **Availability** | The circuit for an LPG leakage detector is readily available in the market, but it is extremely expensive). Presented here is a low-cost circuit for a Gas Leakage Detection that you can build easily. |
| NFR-6 | **Scalability** | The system proves the need for gas detection alarm systems to be 100% reliable. A backup power supply can be included in the system design to augment for power failure condition. Also, calibration of the gas sensor can be done in other for a specific gas to be sensed instead of the LPG numerous gases<br>it sense |

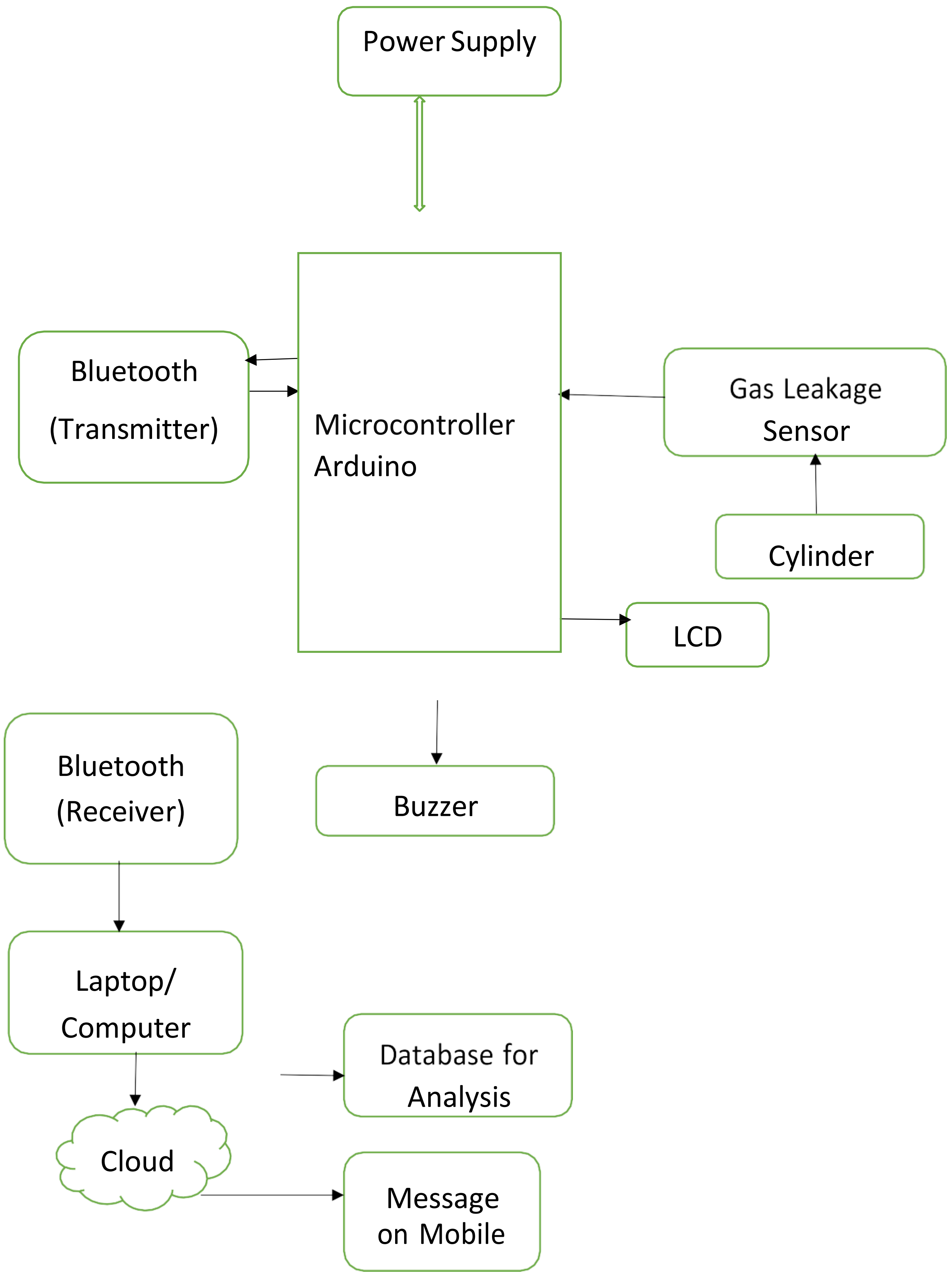
# 5. PROJECT DESIGN

## Data Flow Diagrams



## Solution and Technical Architecture

## User Stories



| JOURNEY STEPS<br>Which step of the experience are you describing? | DISCOVERY<br>Why do they even start the journey? | REGISTRATION<br>Why would they trust us? | ONBOARDING<br>How can they feel successful? | SHARING<br>Why would they invite others? |
|---|---|---|---|---|
| ACTIONS<br>What does the customer do?<br>What they expect? | Leakage of the gas is detected<br>Type of the gas leaked is detected | To share their contact details to reach them out<br>To prioritize delivery | Check for well functioning and faulty devices<br>Ensure all specifications are met | Check for authenticity<br>Test device before sharing |
| NEEDS AND PAINS<br>What does the customer want to achieve and what to avoid? | To prevent future disaster<br>Network failure & human errors | Completely know about the device<br>Not being customer friendly | Achieve maintenance and long life<br>Looks down on expensive | A way of helping society<br>Efforts going unrecognized |
| TOUCHPOINT<br>What part of the service do they interact with? | Through IOT connected devices such as mobile phones or systems | Website<br>Apps | Database management<br>Warnings and buzzers | Contractors<br>Visual demos |
| CUSTOMER FEELING<br>What is the customer feel about product? | Secured feeling<br>Happy about this discovery | Non complex<br>Easy process | Trustable<br>Confident equipment handling | Save peoples life<br>Generate good revenue |

# 6. PROJECT PLANNING AND SCHEDULING

## Sprint Planning And Estimation

### Product Backlog, Sprint Schedule, and Estimation (4 Marks)

| Sprint | Functional Requirement (Epic) | User Story Number | User Story / Task | Story Points | Priority | Team Members |
|---|---|---|---|---|---|---|
| Sprint-1 | Objective | USN-1 | As a system, the gas sensor should detect the gas | 8 | High | MANIKANDA<br>KEERTHI J<br>LAVANYA V<br>RUPAVATHY |
| Sprint-1 | Features | USN-2 | As a system, the gas sensor values displayed onLCD | 5 | High | MANIKANDA<br>KEERTHI J<br>LAVANYA V<br>RUPAVATHY |
| Sprint-1 | Features | USN-3 | As a system, when gas reaches the threshold value,Red color LED will glow | 8 | High | MANIKANDA<br>KEERTHI J<br>LAVANYA V<br>RUPAVATHY |
| Sprint-1 | Features | USN-4 | As a system, then Siren will ON | 2 | Low | MANIKANDA<br>KEERTHI J<br>LAVANYA V<br>RUPAVATHY |

| Sprint | Functional Requirement (Epic) | User Story Number | User Story / Task | Story Points | Priority | Team Members |
|---|---|---|---|---|---|---|
| Sprint-2 | Focus | USN-5 | Asa system, it should send location to the user wheregas leakage is detected | 5 | Medium | MANIKANDA KEERTHI J LAVANYA V RUPAVATHY |
| Sprint-2 | Focus | USN-6 | As a program, it should sendalerting SMStotheregisteredmobilenumber | 8 | High | MANIKANDA KEERTHI J LAVANYA V RUPAVATHY |
| Sprint-2 | Features | USN-7 | As a system, the gas pipe should be closedautomatically once it attains threshold. | 5 | Medium | MANIKANDA KEERTHI J LAVANYA V RUPAVATHY |
| Sprint 3 | Data Transfer | USN-8 | As a program, it should retrieve API key ofthe IBM cloud to send the details | 3 | Low | MANIKANDA KEERTHI J LAVANYA V RUPAVATHY |
| Sprint 3 | Data Transfer | USN-9 | As a system, it should send sensor values along with latitude and longitude to IBM cloud and to NodeRed | 5 | Medium | MANIKANDA KEERTHI J LAVANYA V RUPAVATHY |
| Sprint 3 | Data Transfer | USN-10 | As an application, it should display the gas level details to the user through frontend mitapp | 8 | High | MANIKANDA KEERTHI J LAVANYA V RUPAVATHY |

| Sprint | Functional Requirement (Epic) | User Story Number | User Story / Task | Story Points | Priority | Team Members |
|---|---|---|---|---|---|---|
| Sprint-4 | Registration | USN-11 | As a user, I must receive confirmation mail and SMS onregistration | 3 | High | MANIKANDA KEERTHI J LAVANYA V RUPAVATHY |
| Sprint-4 | Dashboard | USN-12 | As a user, I can login into the web application through email and password. | 2 | Medium | MANIKANDA KEERTHI J LAVANYA V RUPAVATHY |
| Sprint-4 | Dashboard | USN-13 | As a user, I can access the dashboard andmake use of available resources. | 5 | High | MANIKANDA KEERTHI J LAVANYA V RUPAVATHY |
| Sprint 4 | Focus | USN-14 | As a user, I must receive an SMS once the leakage is detected. | 5 | High | MANIKANDA KEERTHI J LAVANYA V RUPAVATHY |
| Sprint-4 | Allocation | USN-15 | As an admin, I must receive information about the leakage along with location and share exact locationand route to the person. I must allot particular person to look after the leakage in a particular location. | 4 | High | MANIKANDA KEERTHI J LAVANYA V RUPAVATHY |

## Project Tracker, Velocity & Burndown Chart: (4 Marks)

| Sprint | Total Story Points | Duration | Sprint Start Date | Sprint End Date (Planned) | Story Points Completed (as on Planned End Date) | Sprint Release Date (Actual) |
|---|---|---|---|---|---|---|
| Sprint-1 | 20 | 6 Days | 24 Oct 2022 | 30 Oct 2022 | 20 | 01 Nov2022 |
| Sprint-2 | 20 | 6 Days | 31 Oct 2022 | 07 Nov 2022 | 20 | 08 Nov 2022 |
| Sprint-3 | 20 | 6 Days | 07 Nov 2022 | 11 Nov 2022 | 20 | 12 Nov 2022 |
| Sprint-4 | 20 | 6 Days | 14 Nov 2022 | 18 Nov 2022 | 20 | 19 Nov 2022 |

### Velocity:

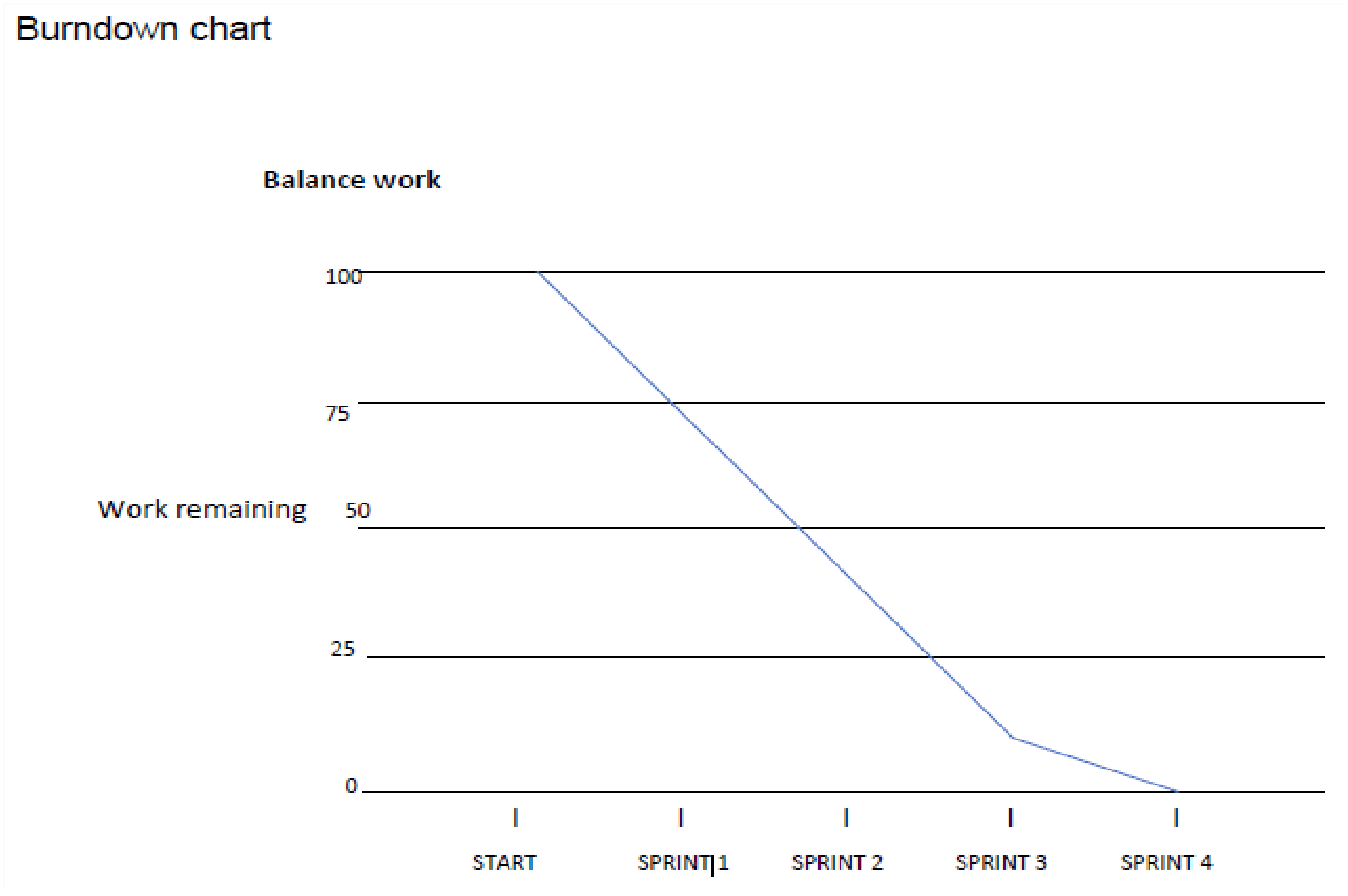
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let’s calculate the team’s average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

## Burndown Chart:

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such
as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.

https://www.visual-paradigm.com/scrum/scrum-burndown-chart/
https://www.atlassian.com/agile/tutorials/burndown-charts



# SPRINT 1

In Sprint 1, the sensor reads the gas value , and if it is above the threshold, the RED led will glow and Buzzer will ON.

If the sensor value is between the low level and threshold, Yellow color LED will glow.

Otherwise, if it is very lower than threshold, the Green color LED will glow.

## CODE:

```
#include<LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(32, 16, 2);
int green = 2;
int yellow = 3;
int red = 4;
int siren = 5;
int gas = A0;
int sensorValue = 0;
void setup()
{
  Serial.begin(9600);
  lcd.init();
  lcd.clear();
  lcd.backlight();
  lcd.setCursor(3,0);
  lcd.print("GAS LEAKAGE");
  lcd.setCursor(4,1);
  lcd.print("DETECTION");
  delay(3000);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Gas Value: ");
  pinMode(green, OUTPUT);
  pinMode(yellow, OUTPUT);
  pinMode(red, OUTPUT);
  pinMode(siren, OUTPUT);
  digitalWrite(red, LOW);
  digitalWrite(yellow, LOW);
  digitalWrite(green, LOW);
}
void loop()
{
  sensorValue = analogRead(gas);
  Serial.println(sensorValue);
  lcd.setCursor(11,0);
  lcd.print(sensorValue);
```

```
if(sensorValue > 500)
{
  lcd.setCursor(0,1);
  lcd.print("GAS DETECTED");
  digitalWrite(red, HIGH);
  digitalWrite(yellow, LOW);
  digitalWrite(green, LOW);
  tone(siren, 200);
}
else if(sensorValue > 281 && sensorValue < 500)
{
  lcd.setCursor(0,1);
  lcd.print("            ");
  digitalWrite(yellow, HIGH);
  digitalWrite(red, LOW);
  digitalWrite(green, LOW);
  noTone(siren);
}
else
{
  lcd.setCursor(0,1);
  lcd.print("            ");
  digitalWrite(green, HIGH);
  digitalWrite(red, LOW);
  digitalWrite(yellow, LOW);
```

```
    noTone(siren);
  }
  delay(1000);
}
```

Thus Sprint 1 is successfully completed.

## SPRINT 2

In Sprint 2, the sensor value is above 500, it will check the gate condition whether it is open or close.

The Servo motor controls the knob of the gate.

If the value is above the threshold, it will send the message “Gas Detected”

**CODE:**

```
#include<Servo.h>

#include <TinyGPS++.h>

#include <SoftwareSerial.h>

#include<LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(32, 16, 2);

int GPSBaud = 9600;

TinyGPSPlus gps;

SoftwareSerial sgps(13, 15); //Rx , Tx gps

SoftwareSerial sgsm(3, 1); // Rx , Tx gsm

#define KNOB 3

#define LEVER 2

Servo myservo;
```

```
int gas = A5;
int sensorValue = 0;
bool gateClosed = true;



void setup()
{
  Serial.begin(9600);
  pinMode(LEVER, INPUT);
  myservo.attach(KNOB);
  myservo.write(90);
  sgsm.begin(9600);
  sgps.begin(9600);
  lcd.init();
  lcd.clear();
  lcd.backlight();
  lcd.setCursor(3,0);
  lcd.print("GAS LEAKAGE");
  lcd.setCursor(4,1);
  lcd.print("DETECTION");
  delay(3000);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Gas Value: ");
}
```

```
void loop()
{
  sensorValue = analogRead(gas);
  Serial.println(sensorValue);
  if(sensorValue > 500 && !gateClosed)
  {
    Serial.println("GAS DETECTED");
    lcd.setCursor(0,1);
    lcd.print("GAS DETECTED ");
    sendSMS("GAS IS DETECTED!!");
    myservo.write(90);
    gateClosed = true;
    sendSMS("THE KNOB IS CLOSED");
    lcd.setCursor(0,1);
    lcd.print("KNOB IS CLOSED");
    delay(1000);
  }
  else if(sensorValue > 500 && gateClosed)
  {
    Serial.println("GAS DETECTED");
    lcd.setCursor(0,1);
    lcd.print("GAS DETECTED ");
    sendSMS("GAS IS DETECTED!!");
    sendSMS("THE KNOB IS ALREADY CLOSED");
    lcd.setCursor(0,1);
    lcd.print("KNOB IS CLOSED");
    delay(1000);
  }
  else
  {
    byte buttonState = digitalRead(LEVER);
    if(buttonState == HIGH)
    {
      myservo.write(0);
      gateClosed = false;
      Serial.println("GATE IS OPENED");
    }
    else
    {
      myservo.write(90);
      gateClosed = true;
      Serial.println("GATE IS CLOSED");
    }
  }
}
void sendSMS(char*message)
{
  while (sgps.available() > 0)
    if (gps.encode(sgps.read()))
    {
      if (gps.location.isValid())
      {
        sgsm.listen();
        sgsm.print("\r");
        delay(1000);
        sgsm.print("AT+CMGF=1\r"); // AT COMMAND TO SEND SMS
        delay(1000);
        /*Replace XXXXXXXXXX to 10 digit mobile number &
        ZZ to 2 digit country code*/
        sgsm.print("AT+CMGS=\"+919025681637\"\r"); // REGISTERED
NUMBER TO SEND SMS
        delay(1000);
        //The text of the message to be sent.
        sgsm.print(message);
        sgsm.print("https://www.google.com/maps/?q="); // MAPS
        sgsm.print(gps.location.lat(), 6); // LAT
        sgsm.print(",");
        sgsm.print(gps.location.lng(), 6); // LONG     delay(1000);
        sgsm.write(0x1A);
        delay(1000);
      }
    }
}
```

Thus Sprint 2 is successfully completed.

## SPRINT 3

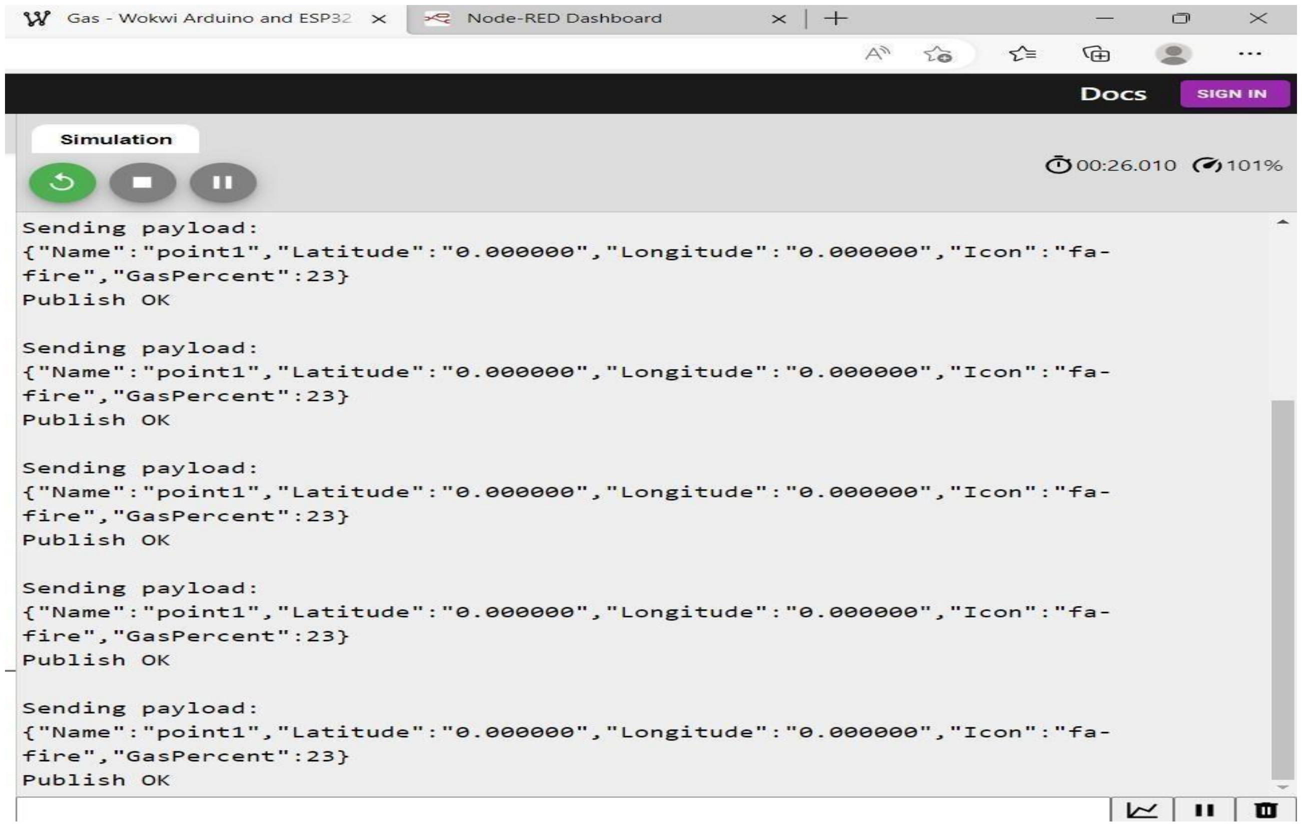
In Sprint 3, the sensor data will send to the IBM Cloud.

Along with the sensor value, it should send the latitude and longitude to IBM Cloud and Node Red.

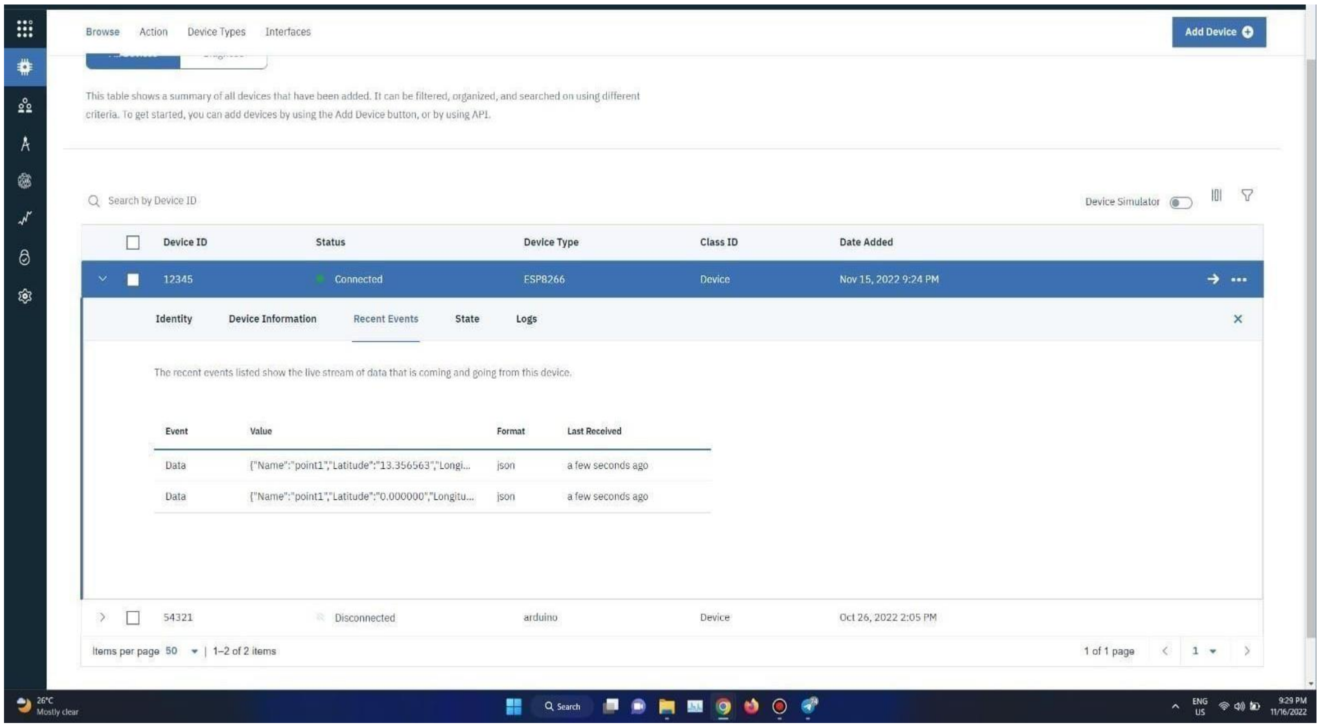
**CODE:**

```
#include <WiFi.h>
#include <PubSubClient.h>
#include <ArduinoJson.h>

WiFiClient  wifiClient;

#define ORG "mz6rat"
#define DEVICE_TYPE "ESP8266"
#define DEVICE_ID "12345"
#define TOKEN "123456789"
#define speed 0.034

char server[] = ORG
".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Data/fmt/json";
char topic[] = "iot-2/cmd/home/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
PubSubClient client(server, 1883, wifiClient);
void publishData();
```

```
const int trigpin=5;
const int echopin=18;
String command;
String data="";
String  lat="13.356563";
String  lon="80.141428";
String name="point1";
String icon="fa-fire";

long  duration;
int dist;

void setup()
{
  Serial.begin(115200);
  pinMode(trigpin, OUTPUT);
  pinMode(echopin, INPUT);
  wifiConnect();
  mqttConnect();
}

void loop() {

  publishData();
  delay(500);

  if (!client.loop()) {
    mqttConnect();
  }
}

void wifiConnect() {
  Serial.print("Connecting to "); Serial.print("Wifi");
  WiFi.begin("Wokwi-GUEST", "", 6);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.print("WiFi connected, IP address: ");
Serial.println(WiFi.localIP());
}

void mqttConnect() {
  if (!client.connected()) {
    Serial.print("Reconnecting MQTT client to ");
Serial.println(server);
    while (!client.connect(clientId, authMethod, token)) {
      Serial.print(".");
      delay(1000);
    }
    initManagedDevice();
    Serial.println();
  }
}

void initManagedDevice() {
  if (client.subscribe(topic)) {
    Serial.println(client.subscribe(topic));
    Serial.println("subscribe to cmd OK");
  } else {
    Serial.println("subscribe to cmd FAILED");
  }
}
void publishData()
{
  digitalWrite(trigpin,LOW);
  digitalWrite(trigpin,HIGH);
  delayMicroseconds(10);
  digitalWrite(trigpin,LOW);
  duration=pulseIn(echopin,HIGH);
  dist=duration*speed/2;
  dist=dist/4;
  dist=100-dist;
  if(dist>80){
    lat="13.356563";
    lon="80.141428";
  }else{
    lat="0.000000";
    lon="0.000000";
  }
  DynamicJsonDocument doc(1024);
  String payload;
  doc["Name"]=name;
  doc["Latitude"]=lat;
  doc["Longitude"]=lon;
  doc["Icon"]=icon;
  doc["GasPercent"]=dist;
  serializeJson(doc, payload);
  delay(3000);
  Serial.print("\n");
  Serial.print("Sending payload: ");
  Serial.println(payload);
  if (client.publish(publishTopic, (char*) payload.c_str()))
{
    Serial.println("Publish OK");
  } else {
    Serial.println("Publish FAILED");
  }
}
```



```
Sending payload:
{"Name":"point1","Latitude":"0.000000","Longitude":"0.000000","Icon":"fa-
fire","GasPercent":23}
Publish OK

Sending payload:
{"Name":"point1","Latitude":"0.000000","Longitude":"0.000000","Icon":"fa-
fire","GasPercent":23}
Publish OK

Sending payload:
{"Name":"point1","Latitude":"0.000000","Longitude":"0.000000","Icon":"fa-
fire","GasPercent":23}
Publish OK

Sending payload:
{"Name":"point1","Latitude":"0.000000","Longitude":"0.000000","Icon":"fa-
fire","GasPercent":23}
Publish OK

Sending payload:
{"Name":"point1","Latitude":"0.000000","Longitude":"0.000000","Icon":"fa-
fire","GasPercent":23}
Publish OK
```



This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.

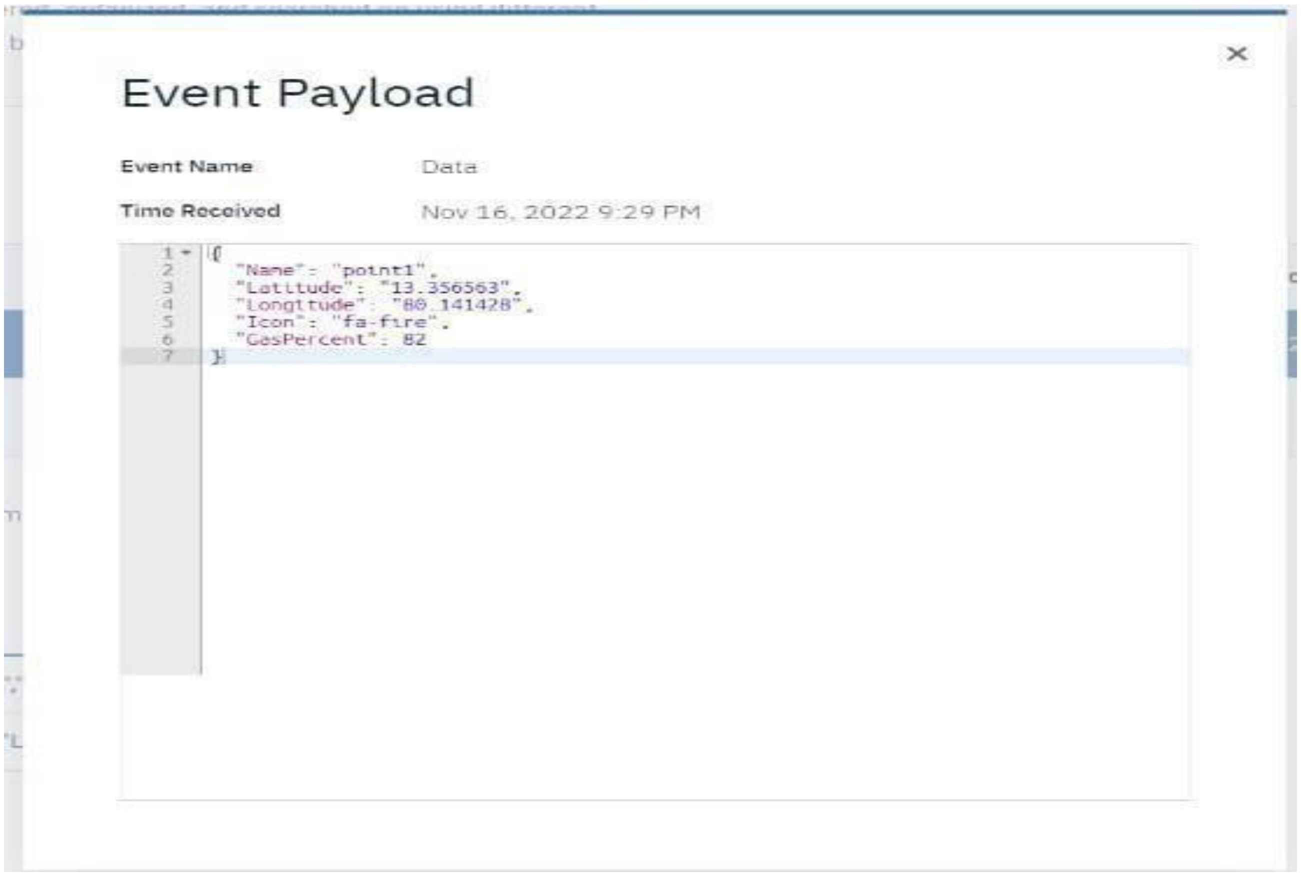
| Device ID | Status | Device Type | Class ID | Date Added |
|---|---|---|---|---|
| 12345 | Connected | ESP8266 | Device | Nov 15, 2022 9:24 PM |
| 54321 | Disconnected | arduino | Device | Oct 26, 2022 2:05 PM |

The recent events listed show the live stream of data that is coming and going from this device.

| Event | Value | Format | Last Received |
|---|---|---|---|
| Data | {"Name":"point1","Latitude":"13.356563","Longi... | json | a few seconds ago |
| Data | {"Name":"point1","Latitude":"0.000000","Longitu... | json | a few seconds ago |

Items per page 50 | 1–2 of 2 items

1 of 1 page 1


Event Payload

Event Name Data

Time Received Nov 16, 2022 9:29 PM

```
{
  "Name": "point1",
  "Latitude": "13.356563",
  "Longtitude": "80.141428",
  "Icon": "fa-fire",
  "GasPercent": 82
}
```

Thus Sprint 3 is successfully completed.

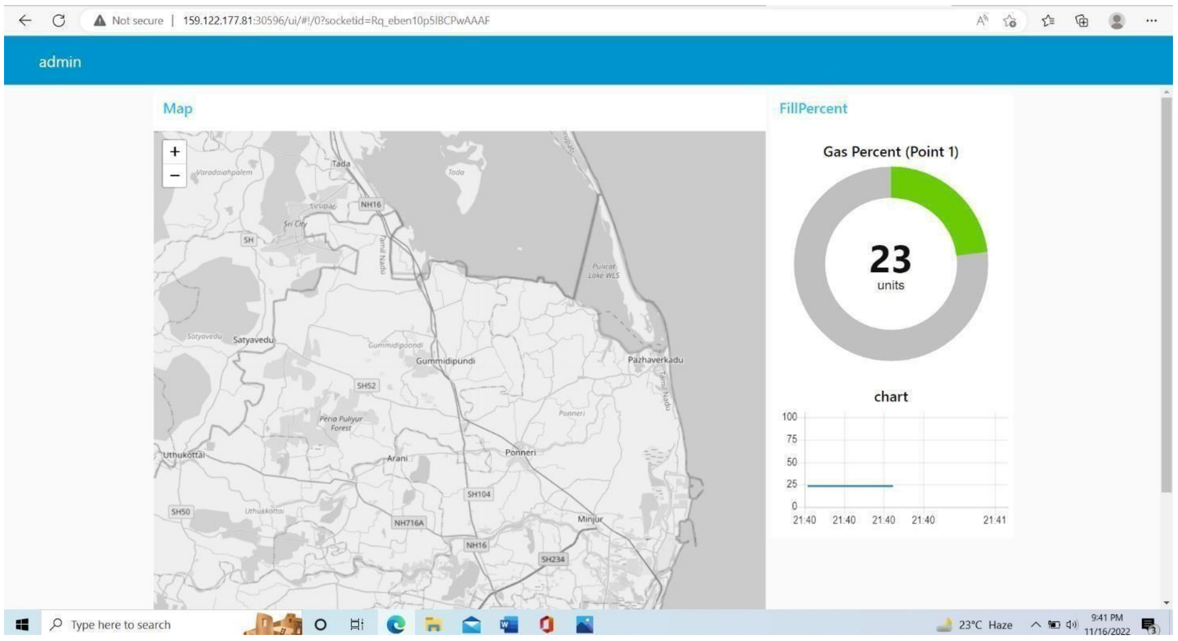
## SPRINT 4

In Sprint 4,the gas percentage value detected by the sensor is displayed on the dashboard.

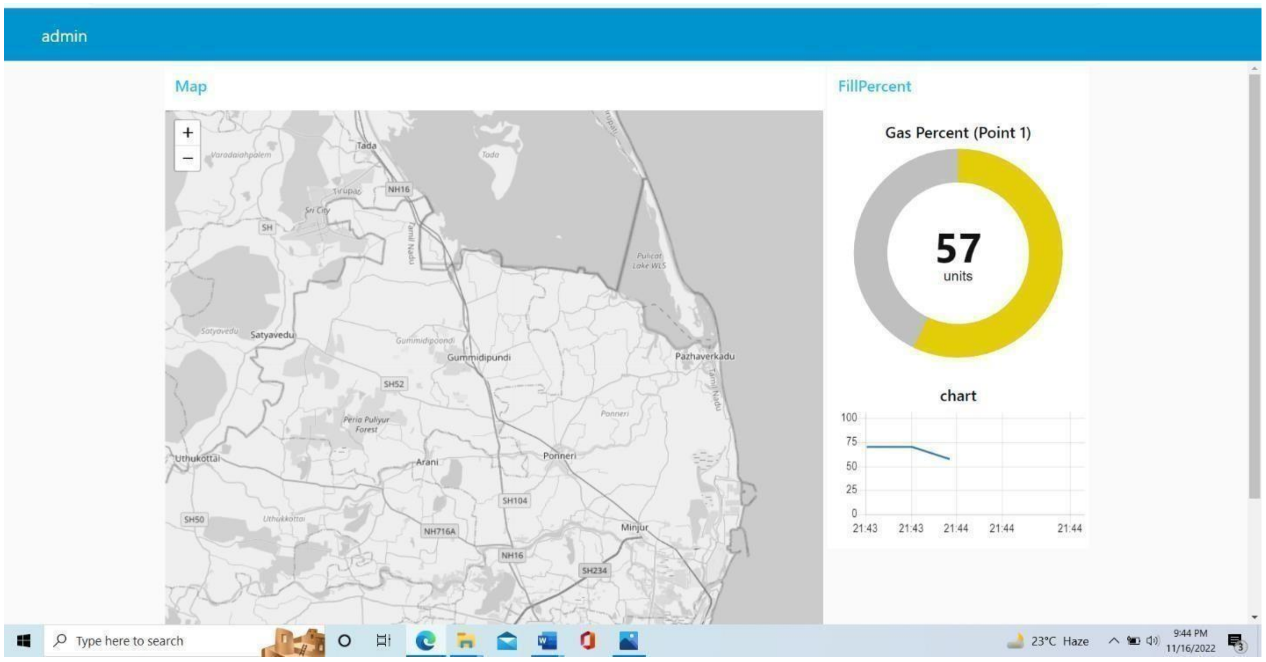
**CODE:**

```
#include <WiFi.h>
#include <PubSubClient.h>
#include <ArduinoJson.h>

WiFiClient  wifiClient;

#define ORG "mz6rat"
#define DEVICE_TYPE "ESP8266"
#define DEVICE_ID "12345"
#define TOKEN "123456789"
#define speed 0.034

char server[] = ORG
".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Data/fmt/json";
char topic[] = "iot-2/cmd/home/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
PubSubClient client(server, 1883, wifiClient);
void publishData();

const int trigpin=5;
const int echopin=18;
String command;
String data="";
String  lat="13.356563";
String  lon="80.141428";
String name="point1";
String icon="fa-fire";

long  duration;
int dist;

void setup()
{
  Serial.begin(115200);
  pinMode(trigpin, OUTPUT);
  pinMode(echopin, INPUT);
  wifiConnect();
  mqttConnect();
}

void loop() {

  publishData();
  delay(500);

  if (!client.loop()) {
    mqttConnect();
  }
}

void wifiConnect() {
  Serial.print("Connecting to "); Serial.print("Wifi");
  WiFi.begin("Wokwi-GUEST", "", 6);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.print("WiFi connected, IP address: ");
Serial.println(WiFi.localIP());
```

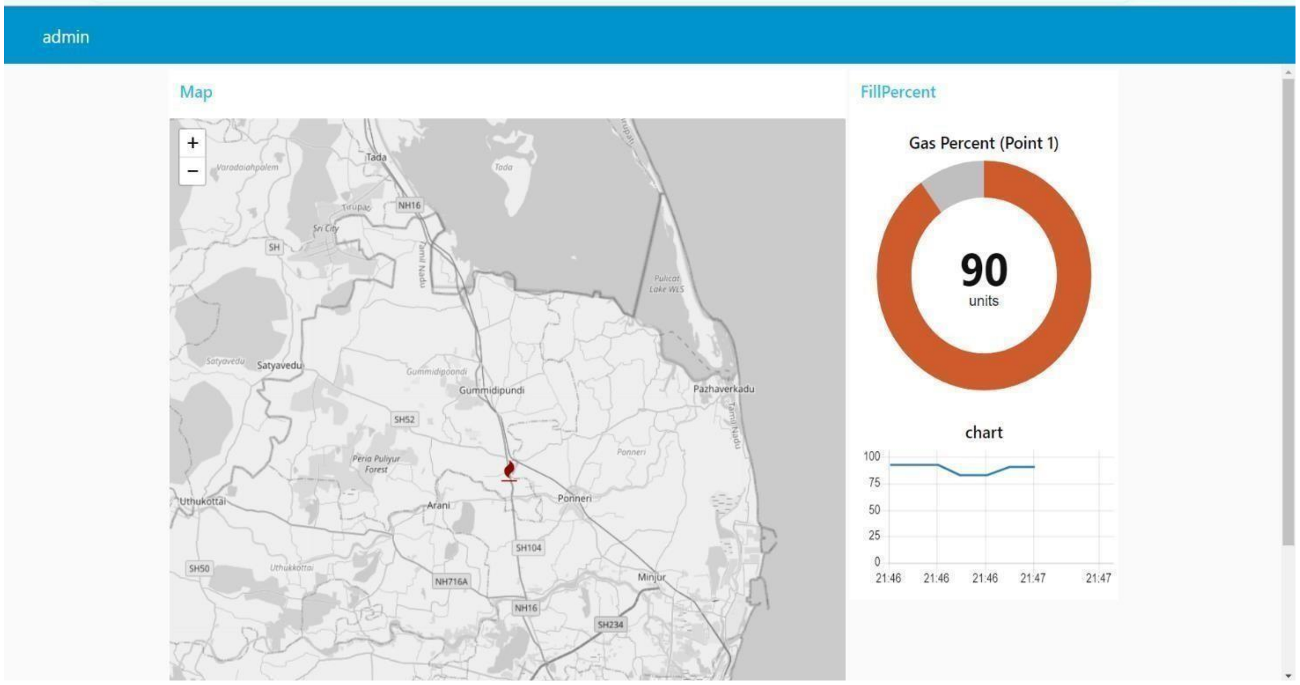
```
}

void mqttConnect() {
  if (!client.connected()) {
    Serial.print("Reconnecting MQTT client to ");
Serial.println(server);
    while (!client.connect(clientId, authMethod, token)) {
      Serial.print(".");
      delay(1000);
    }
    initManagedDevice();
    Serial.println();
  }
}

void initManagedDevice() {
  if (client.subscribe(topic)) {
    Serial.println(client.subscribe(topic));
    Serial.println("subscribe to cmd OK");
  } else {
    Serial.println("subscribe to cmd FAILED");
  }
}
void publishData()
{
  digitalWrite(trigpin,LOW);
  digitalWrite(trigpin,HIGH);
  delayMicroseconds(10);
  digitalWrite(trigpin,LOW);
  duration=pulseIn(echopin,HIGH);
  dist=duration*speed/2;
  dist=dist/4;
  dist=100-dist;
  if(dist>80){
    lat="13.356563";
    lon="80.141428";
  }else{
    lat="0.000000";
    lon="0.000000";
  }
  DynamicJsonDocument doc(1024);
  String payload;
  doc["Name"]=name;
  doc["Latitude"]=lat;
  doc["Longitude"]=lon;
  doc["Icon"]=icon;
  doc["GasPercent"]=dist;
  serializeJson(doc, payload);
  delay(3000);
  Serial.print("\n");
  Serial.print("Sending payload: ");
  Serial.println(payload);
  if (client.publish(publishTopic, (char*) payload.c_str()))
{
    Serial.println("Publish OK");
  } else {
    Serial.println("Publish FAILED");
  }
}
```



The green color indicates, the gas value is very below the threshold value, thus there is no harm.



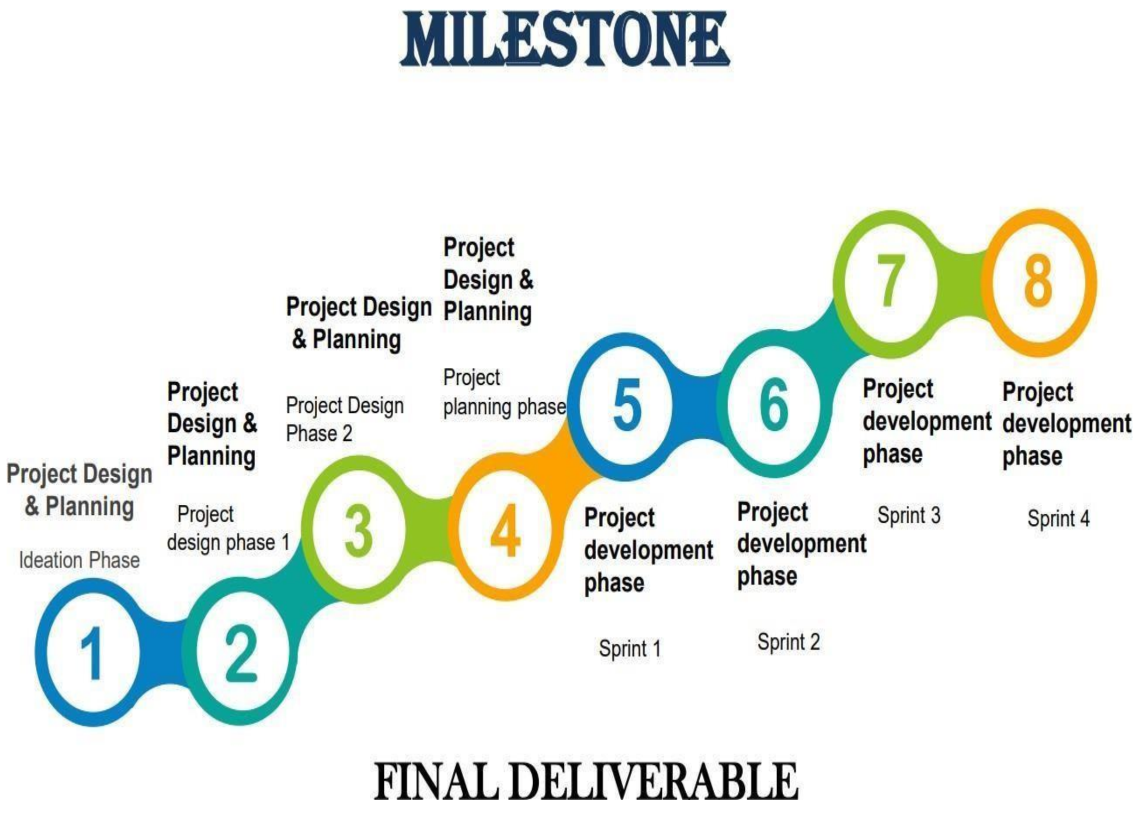
The Yellow color indicates, the gas value is on threshold.



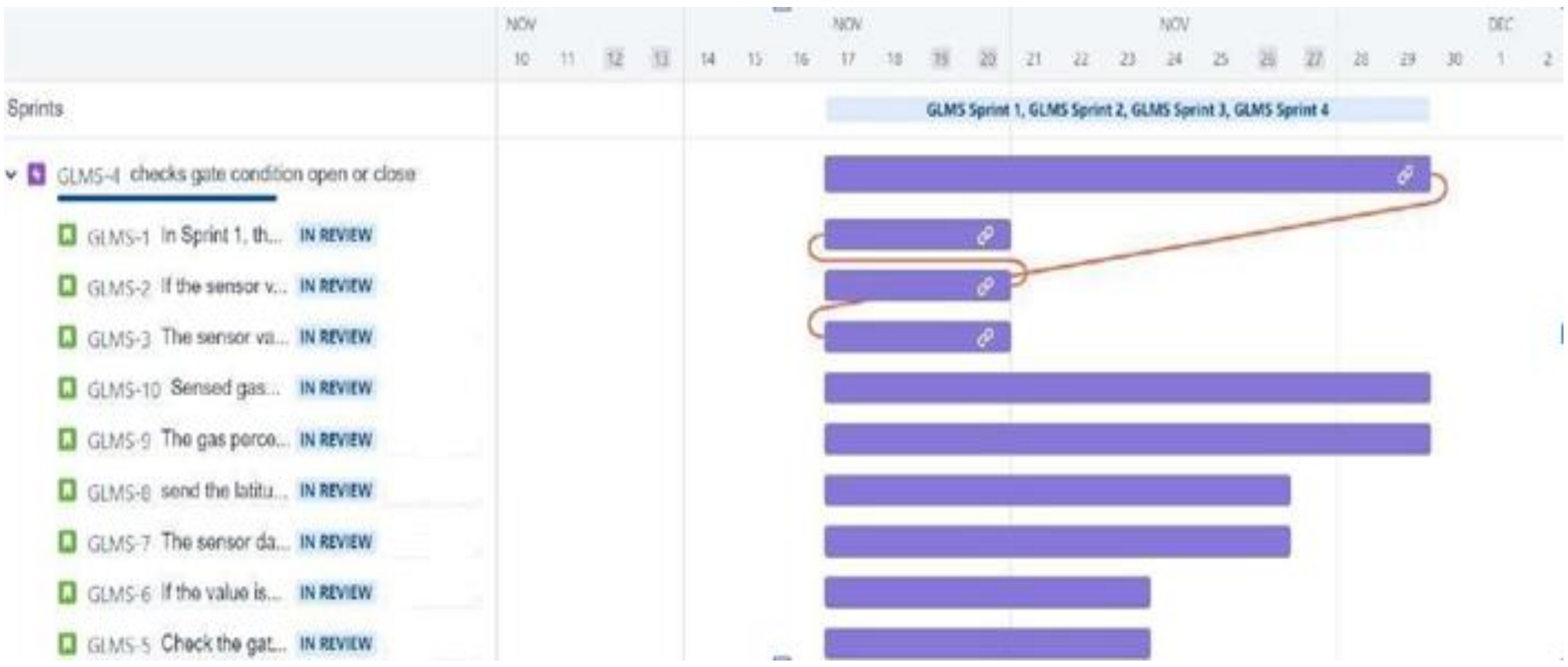
The red color denotes it crosses the threshold value, hence gas leakage is detected.

Thus Sprint 4 is successfully completed.

# SPRINT DELIVERY SCHEDULING


MILESTONE
1
Project Design & Planning
Ideation Phase
2
Project Design & Planning
Project design phase 1
3
Project Design & Planning
Project Design Phase 2
4
Project Design & Planning
Project planning phase
5
Project development phase
Sprint 1
6
Project development phase
Sprint 2
7
Project development phase
Sprint 3
8
Project development phase
Sprint 4
FINAL DELIVERABLE

## REPORTS FROM JIRA



GLMS-4 checks gate condition open or close

GLMS-1 In Sprint 1, th... IN REVIEW

GLMS-2 If the sensor v... IN REVIEW

GLMS-3 The sensor va... IN REVIEW

GLMS-10 Sensed gas... IN REVIEW

GLMS-9 The gas perce... IN REVIEW

GLMS-8 send the latitu... IN REVIEW

GLMS-7 The sensor da... IN REVIEW

GLMS-6 If the value is... IN REVIEW

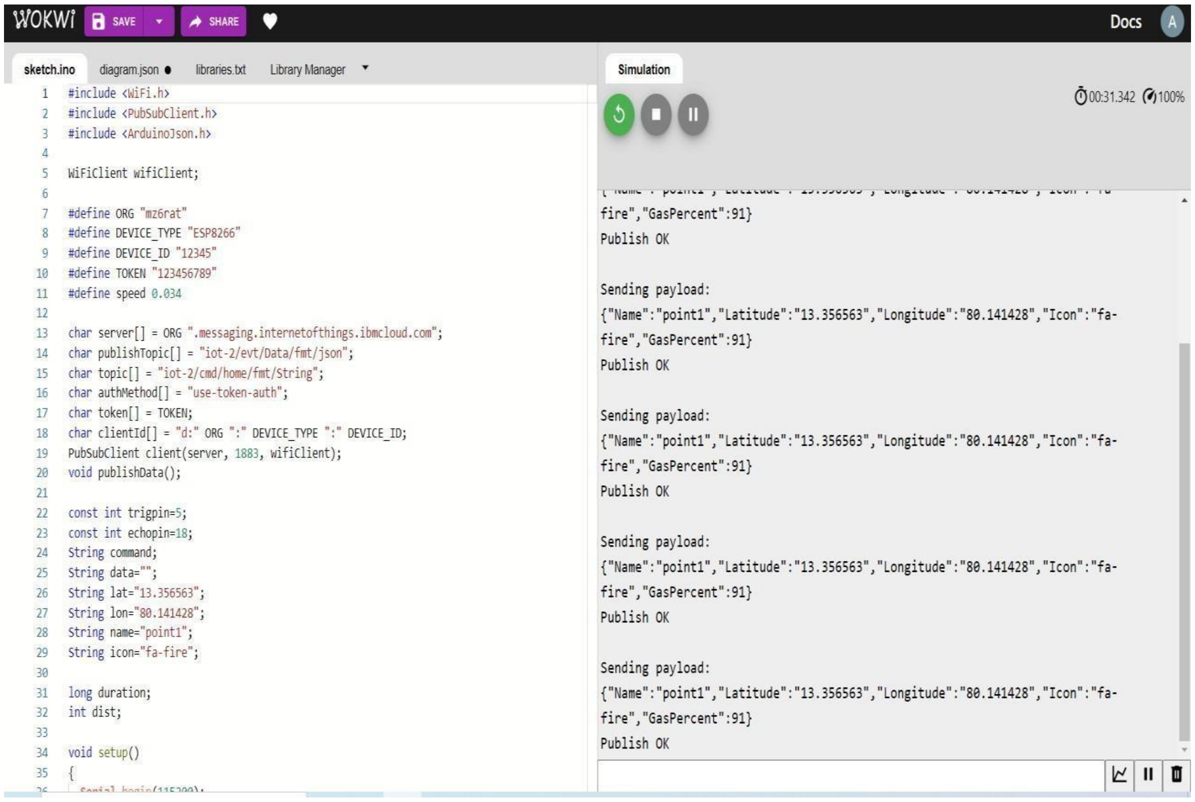
GLMS-5 Check the gat... IN REVIEW

# 7. CODING AND SOLUTIONING

## FEATURE 1

### Step 1:

Find the location of the gas leakage along with gas percentage.

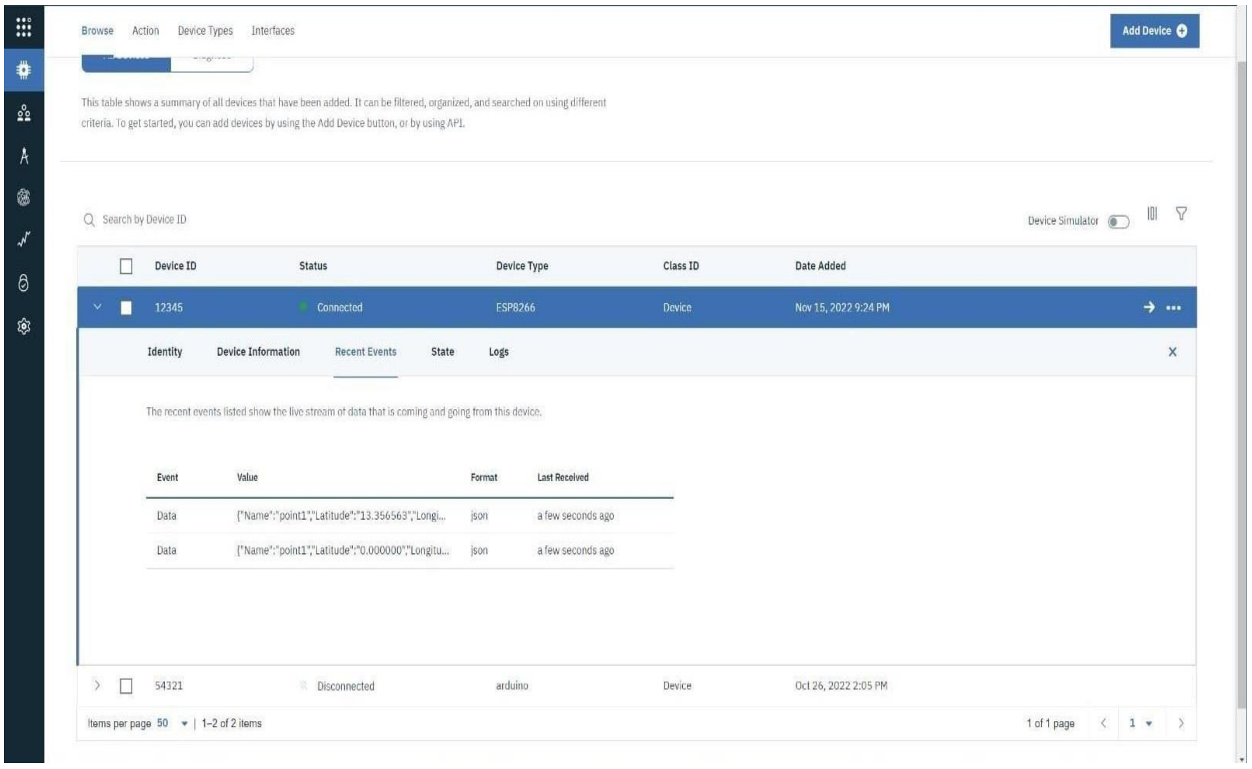


```
#include <WiFi.h>
#include <PubSubClient.h>
#include <ArduinoJson.h>

WiFiClient wifiClient;

#define ORG "mz6rat"
#define DEVICE_TYPE "ESP8266"
#define DEVICE_ID "12345"
#define TOKEN "123456789"
#define speed 0.034

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Data/fmt/json";
char topic[] = "iot-2/cmd/home/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
PubSubClient client(server, 1883, wifiClient);
void publishData();

const int trigpin=5;
const int echopin=18;
String command;
String data="";
String lat="13.356563";
String lon="80.141428";
String name="point1";
String icon="fa-fire";

long duration;
int dist;

void setup()
{
```

```
fire","GasPercent":91}
Publish OK

Sending payload:
{"Name":"point1","Latitude":"13.356563","Longitude":"80.141428","Icon":"fa-
fire","GasPercent":91}
Publish OK

Sending payload:
{"Name":"point1","Latitude":"13.356563","Longitude":"80.141428","Icon":"fa-
fire","GasPercent":91}
Publish OK

Sending payload:
{"Name":"point1","Latitude":"13.356563","Longitude":"80.141428","Icon":"fa-
fire","GasPercent":91}
Publish OK

Sending payload:
{"Name":"point1","Latitude":"13.356563","Longitude":"80.141428","Icon":"fa-
fire","GasPercent":91}
Publish OK
```

**Step 2:**

Output shown on the IBM IOT platform.



This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.

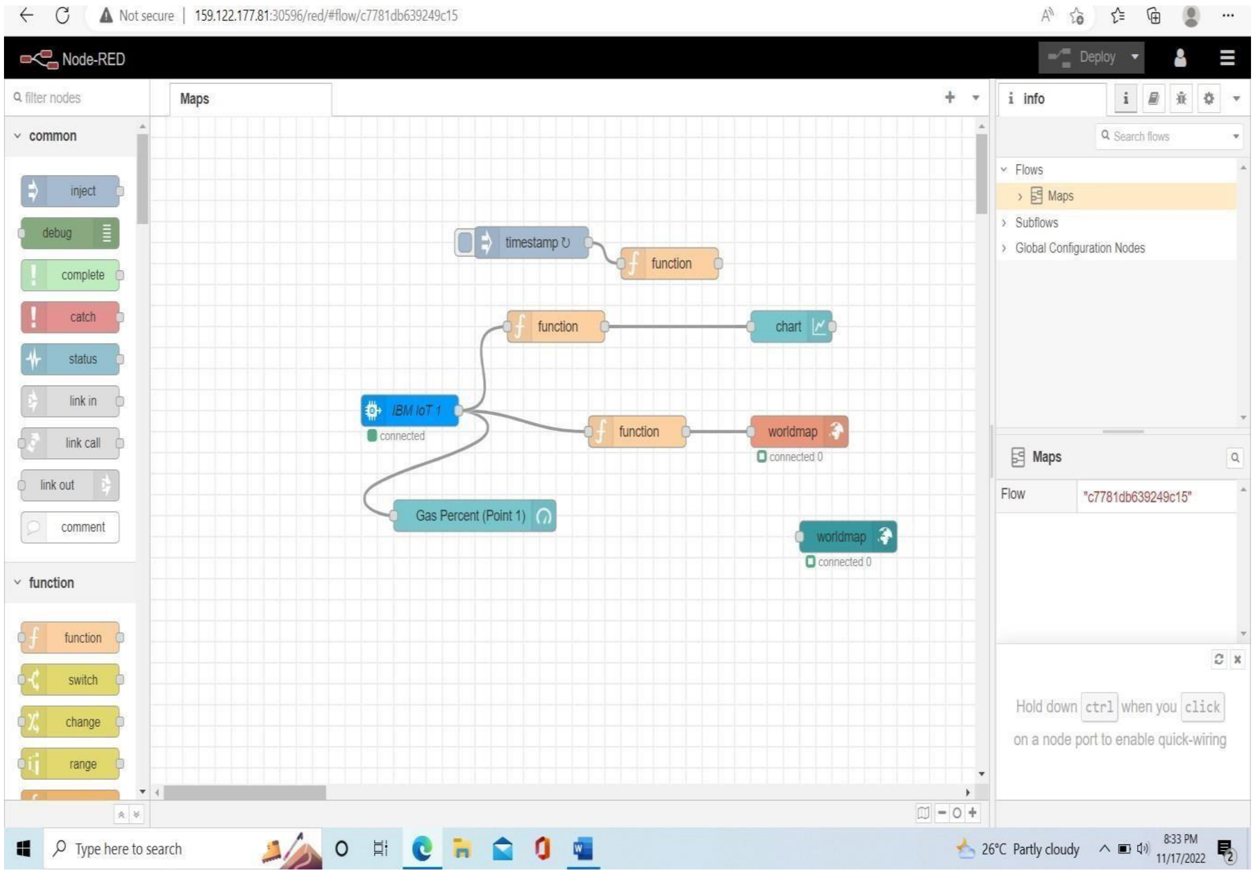
| Device ID | Status | Device Type | Class ID | Date Added |
|---|---|---|---|---|
| 12345 | Connected | ESP8266 | Device | Nov 15, 2022 9:24 PM |
| 54321 | Disconnected | arduino | Device | Oct 26, 2022 2:05 PM |

The recent events listed show the live stream of data that is coming and going from this device.

| Event | Value | Format | Last Received |
|---|---|---|---|
| Data | {"Name":"point1","Latitude":"13.356563","Longi... | json | a few seconds ago |
| Data | {"Name":"point1","Latitude":"0.000000","Longitu... | json | a few seconds ago |

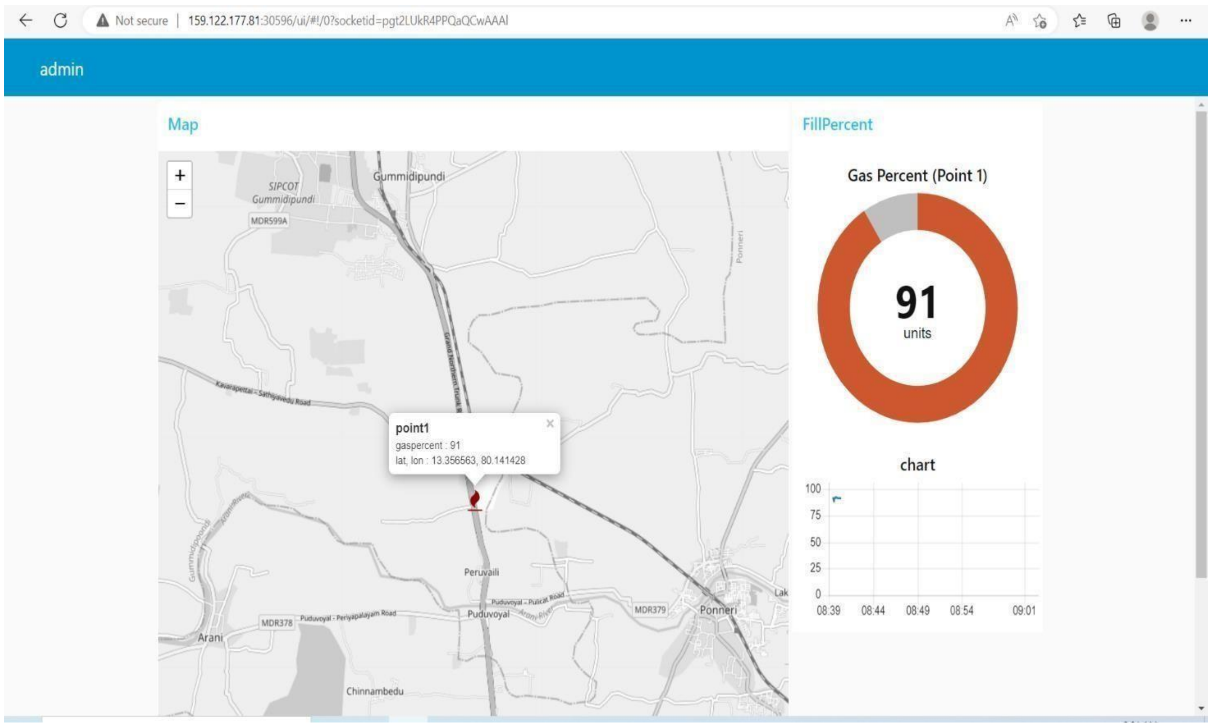
## Step 3:

In this flow we use the IBM IoT node for getting data from IBM Watson IOT platform and changing them into the required format with the help of the function node and passing the values to the Gauge node (UI node) and to the World Map node.



**Step 4:**

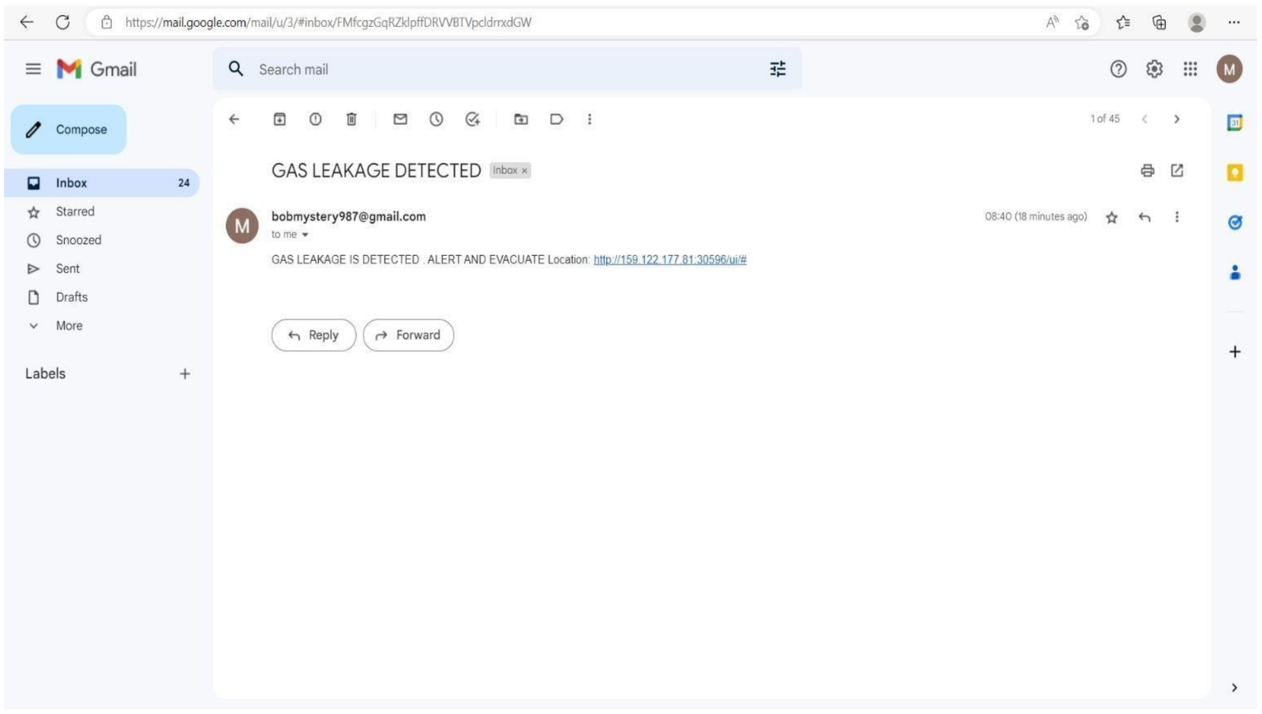
The below figure shows the location along with the gas percentage and it denotes that the leakage is more.



point1
gaspercent : 91
lat, lon : 13.356563, 80.141428

## FEATURES 2

Mail come to the user along with the Alert Message. The mail also contains the location and gas percentage.



GAS LEAKAGE DETECTED Inbox ×

bobmystery987@gmail.com
to me

08:40 (18 minutes ago)

GAS LEAKAGE IS DETECTED . ALERT AND EVACUATE Location: http://159.122.177.81:30596/ui/#

# DATABASE SCHEMA

**<u>Create A Database in Cloudant :</u>**

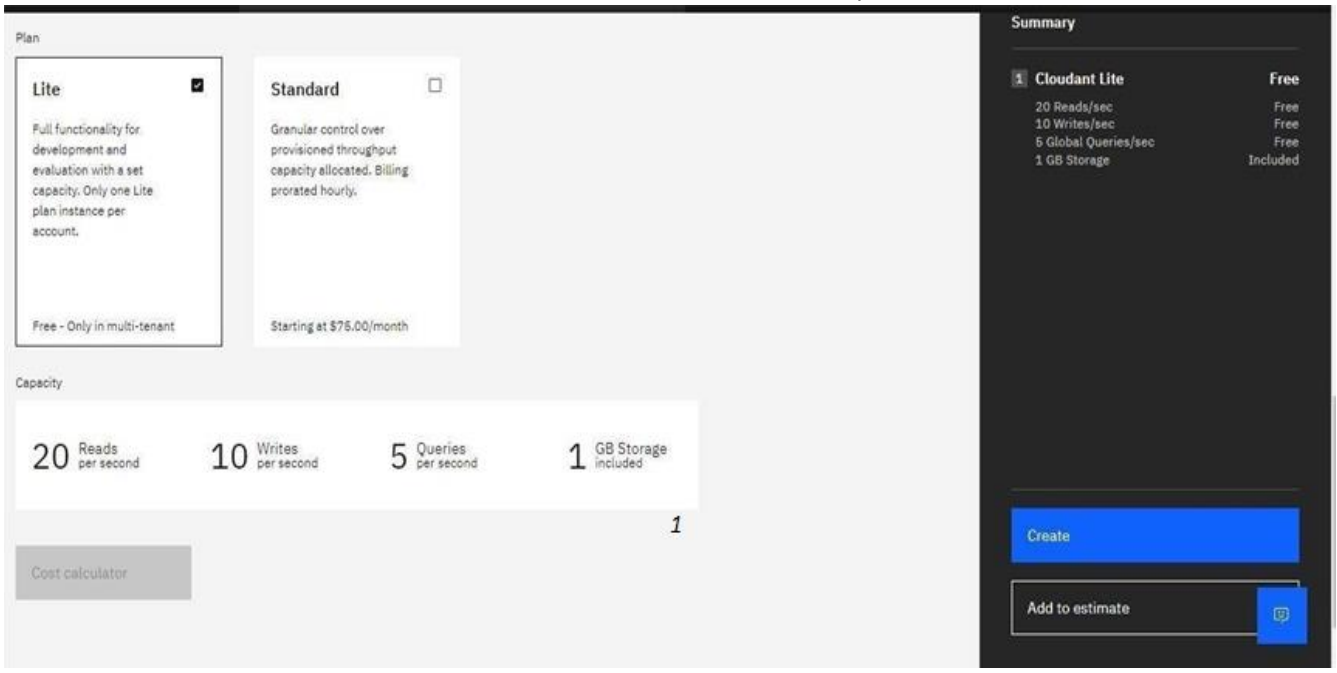
**<u>Aim:</u>**

To create a database in Cloudant to store the location of data.

Steps to be followed:

i. Log in to IBM Cloud account.
ii. Navigated to `./resources`.
iii. Clicked on the “Create Resource +” button.
iv. Searched for “Cloudant”.
v. Chose the “Lite Version” and clicked on “Create”.

The Cloudant database resources is created successfully



Plan

Lite ✓
Full functionality for development and evaluation with a set capacity. Only one Lite plan instance per account.
Free - Only in multi-tenant

Standard ☐
Granular control over provisioned throughput capacity allocated. Billing prorated hourly.
Starting at $75.00/month

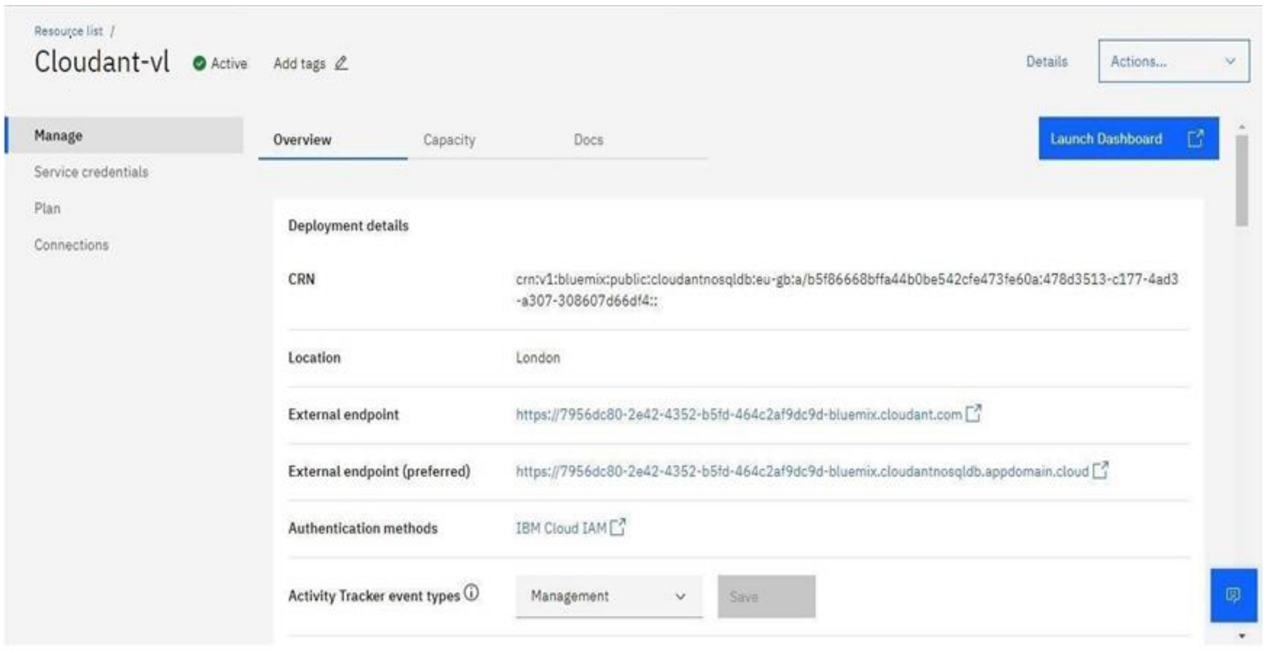
Capacity

20 Reads per second
10 Writes per second
5 Queries per second
1 GB Storage included

1

Summary

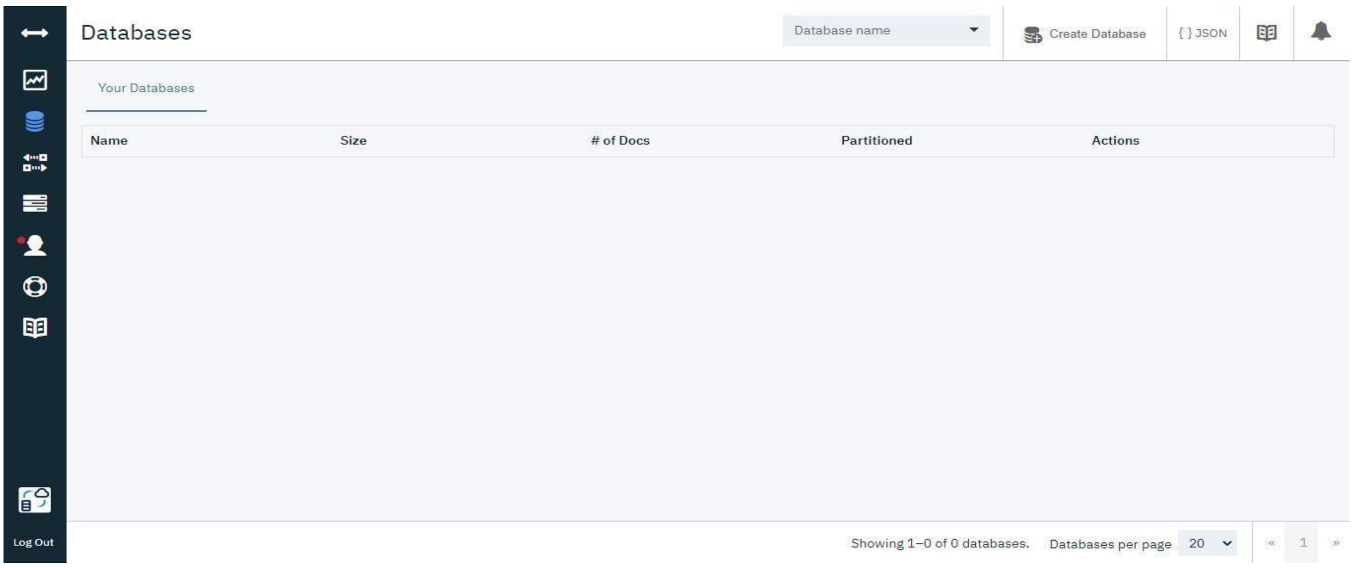
| | |
|---|---|
| 1 Cloudant Lite | Free |
| 20 Reads/sec | Free |
| 10 Writes/sec | Free |
| 5 Global Queries/sec | Free |
| 1 GB Storage | Included |



Deployment details

| | |
|---|---|
| CRN | crn:v1:bluemix:public:cloudantnosqldb:eu-gb:a/b5f86668bffa44b0be542cfe473fe60a:478d3513-c177-4ad3-a307-308607d66df4:: |
| Location | London |
| External endpoint | https://7956dc80-2e42-4352-b5fd-464c2af9dc9d-bluemix.cloudant.com |
| External endpoint (preferred) | https://7956dc80-2e42-4352-b5fd-464c2af9dc9d-bluemix.cloudantnosqldb.appdomain.cloud |
| Authentication methods | IBM Cloud IAM |
| Activity Tracker event types | Management |

Clicked on Launch Dashboard

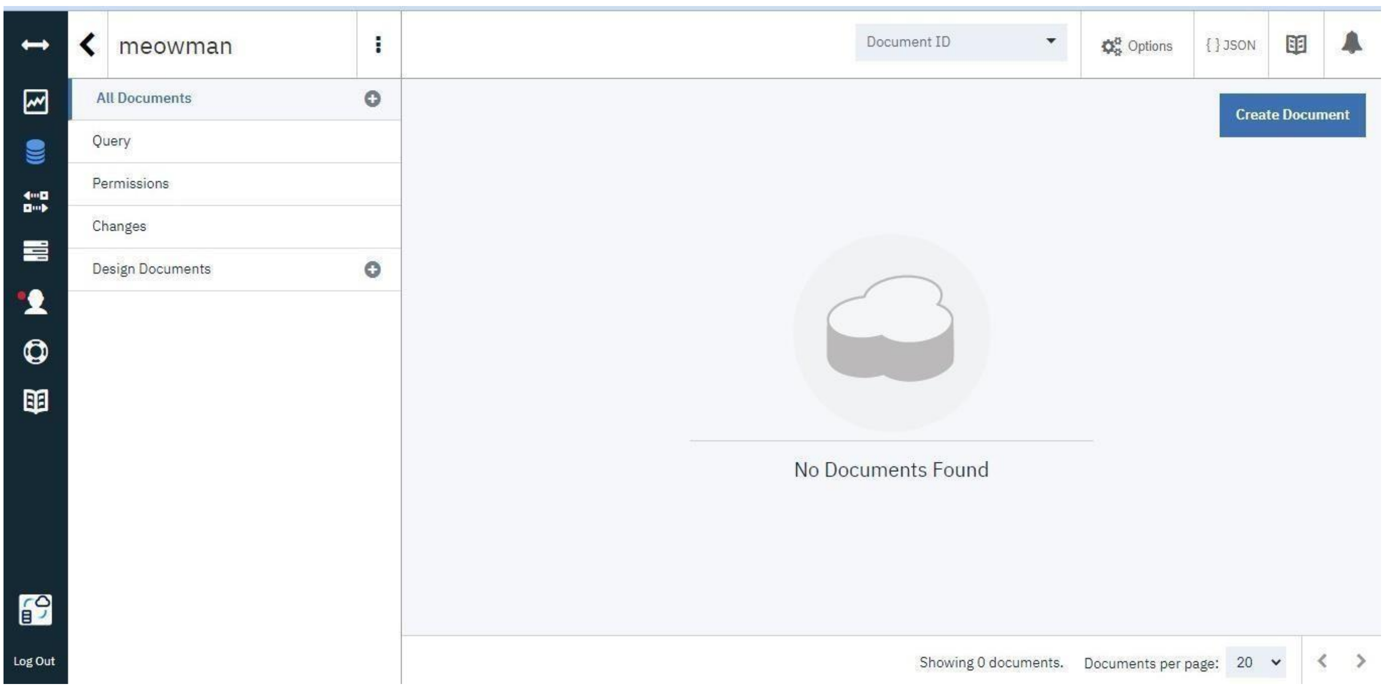


| Name | Size | # of Docs | Partitioned | Actions |
|---|---|---|---|---|

Showing 1–0 of 0 databases. Databases per page 20 « 1 »

Clicked on “Create Database”. Entered “meowman” as the databasename andthe“Non-partitioned” option.

The database “meowman” was created successfully.



**Result:**

A database to store the location data was created successfully on Cloudant database.

# 8. TESTING

## 1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [ProductName] project at the time of the release to User Acceptance Testing (UAT).

# User Acceptance Testing

## 2. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and howthey were resolved

| Resolution | Severity 1 | Severity 2 | Severity 3 | Severity 4 | Subtotal |
|---|---|---|---|---|---|
| By Design | 10 | 4 | 2 | 3 | 19 |
| Duplicate | 1 | 0 | 3 | 0 | 4 |
| External | 2 | 3 | 0 | 1 | 6 |
| Fixed | 10 | 2 | 4 | 20 | 36 |
| Not Reproduced | 0 | 0 | 1 | 0 | 1 |
| Skipped | 0 | 0 | 1 | 1 | 2 |
| Won't Fix | 0 | 5 | 2 | 1 | 8 |
| Totals | 23 | 14 | 13 | 24 | 74 |

# Test Cases

## 3. Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested

| Section | Total Cases | Not Tested | Fail | Pass |
|---|---|---|---|---|
| Print Engine | 7 | 0 | 0 | 7 |
| Client Application | 50 | 0 | 0 | 50 |
| Security | 2 | 0 | 0 | 2 |
| Outsource Shipping | 3 | 0 | 0 | 3 |

|  |  |  |  |  |
|---|---|---|---|---|
| Exception Reporting | 9 | 0 | 0 | 9 |
| Final Report Output | 4 | 0 | 0 | 4 |
| Version Control | 2 | 0 | 0 | 2 |

# 10 . ADVANTAGES & DISADVANTAGES

## ADVANTAGES

- Our system helps customers to upgrade their safety and protect lifeand property from reputed accidents.
- We can able to observe the amount of the gas and also the gas leak.
- By this system, the users can be aware of their gas level and it alsoavoids the prior and delay booking of the cylinder.
- The system is much more efficient and also the sensor used in thesystem MQ-6, is in constant d detection of any change in the environment. Immediate action is taken if there is an accidental leakage insuring 100% safety.
- The proposed system helps the LPG gas customers to lead apleasant life
  - Tiny size compared with other types of load cells.
  - Very little influence because of temperature changes.
  - Extremely precise and linear measurements

## DISADVANTAGES

- Detected Any Moisture
- Will not Find Very small Leak
- Accuracy of Location need to be carefully verified

# 11. CONCLUSION

Gas leaks cause serious disasters that result in property damage and human injuries. The maincauses of gas leaks are poor equipment upkeep and a lack of public awareness. As a result, detecting LPG leaks is critical for avoiding accidents and saving human lives. This paper discussed a system for detecting and alerting LPG leaks. Whenever LPG leakage is detected, this device activates an LED and a buzzer to inform people. This approach is straightforward but dependable.

Internet of Things has gained its wide popularity in recent days due to its various streams of applications which has paved way for smooth, safe and easier mode of living style for human beings. One such area of applications includes gas booking and gas leakage detection for both domestic and commercial purposes. Though, several techniques is existing for the same, yet gas leakage detection is one major concern and a challenge The new proposed system which is microcontroller based application of gas booking and gas detectionsystems using IOT. The sensor used in this model can sense and detect the leakage of the gas and the user gets notification regarding gas leakageand can also monitor the cylinder weight it can be taken to pre-book the new cylinder automatically. This proposed system can be useful in marketing sectors like hotels, shop etc. Themain intention of this work is to ensure safe andeasier way of gas booking and gas leakage detection to avoid disaster that may occur due to negligence

# 12. FUTURE SCOPE

In the future, instead of using AC power, the gas leakage detecting system might be created using photovoltaic panels with abattery as a backup power supply to give a continuous supply, as opposed to the current use of AC power. The protection system employs a combination of MQ6 gas sensors, DHT22 temperature sensors, load sensors, smoke and flame sensors, and PIR sensors.A number of sensors must be calculated, taking into account the room's volume, installation position, and other factors. Thissystem assures that if a gas leak happens, it can be tracked more effectively and that occupants may be notified ahead of time, regardless of whether the leak is visible or not,whether the house is vacant or occupied. The best recommendation for a monitoring system is to utilize a WiFi module that allows the user to monitor the gas level in real-time and automate direct management of the safety device system if an unanticipated occurrence occurs. Finally, the safety device employedwas

the most vital and important aspect. We also suggested that a tripper circuit be built, which would automatically turn off the (MSB) in the event of a fire, and turn off the gas regulator valve via a solenoid valve either from the cylinder from the main switchboard, If an incident occurs, it automatically can switch on the exhaust fan to suck gas to the outsidehouseand sound an alarm and audio buzzer toinform the user or persons around and the user can opening the window, This device monitors the gas and detects any leaks inorder to keep people safe.

# 13 APPENDIX

## SOURCE CODE:

### Detect the gas Leakage

```
#include<Servo.h>
#include <TinyGPS++.h>
#include <SoftwareSerial.h>
#include<LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(32, 16, 2);
int GPSBaud = 9600;
TinyGPSPlus gps;
SoftwareSerial sgps(13, 15); //Rx , Tx gps
SoftwareSerial sgsm(3, 1); // Rx , Tx gsm
#define KNOB 3
#define LEVER 2
Servo myservo;
int gas = A5;
int sensorValue = 0;
```

```
bool gateClosed = true;



void setup()
{
  Serial.begin(9600);
  pinMode(LEVER, INPUT);
  myservo.attach(KNOB);
  myservo.write(90);
  sgsm.begin(9600);
  sgps.begin(9600);
  lcd.init();
  lcd.clear();
  lcd.backlight();
  lcd.setCursor(3,0);
  lcd.print("GAS LEAKAGE");
  lcd.setCursor(4,1);
  lcd.print("DETECTION");
  delay(3000);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Gas Value: ");
}

void loop()
```

```
{
  sensorValue = analogRead(gas);
  Serial.println(sensorValue);
  if(sensorValue > 500 && !gateClosed)
  {
    Serial.println("GAS DETECTED");
    lcd.setCursor(0,1);
    lcd.print("GAS DETECTED ");
    sendSMS("GAS IS DETECTED!!");
    myservo.write(90);
    gateClosed = true;
    sendSMS("THE KNOB IS CLOSED");
    lcd.setCursor(0,1);
    lcd.print("KNOB IS CLOSED");
    delay(1000);
  }
  else if(sensorValue > 500 && gateClosed)
  {
    Serial.println("GAS DETECTED");
    lcd.setCursor(0,1);
    lcd.print("GAS DETECTED ");
    sendSMS("GAS IS DETECTED!!");
    sendSMS("THE KNOB IS ALREADY CLOSED");
    lcd.setCursor(0,1);
    lcd.print("KNOB IS CLOSED");
    delay(1000);
  }
  else
  {
    byte buttonState = digitalRead(LEVER);
    if(buttonState == HIGH)
    {
      myservo.write(0);
      gateClosed = false;
      Serial.println("GATE IS OPENED");
    }
    else
    {
      myservo.write(90);
      gateClosed = true;
      Serial.println("GATE IS CLOSED");
    }
  }
}
void sendSMS(char*message)
{
  while (sgps.available() > 0)
    if (gps.encode(sgps.read()))
    {
      if (gps.location.isValid())
      {
        sgsm.listen();
        sgsm.print("\r");
        delay(1000);
        sgsm.print("AT+CMGF=1\r"); // AT COMMAND TO SEND SMS
        delay(1000);
        /*Replace XXXXXXXXXX to 10 digit mobile number &
        ZZ to 2 digit country code*/
        sgsm.print("AT+CMGS=\"+919025681637\"\r"); // REGISTERED
NUMBER TO SEND SMS
        delay(1000);
        //The text of the message to be sent.
        sgsm.print(message);
        sgsm.print("https://www.google.com/maps/?q="); // MAPS
        sgsm.print(gps.location.lat(), 6); // LAT
        sgsm.print(",");
        sgsm.print(gps.location.lng(), 6); // LONG    delay(1000);
        sgsm.write(0x1A);
        delay(1000);
      }
    }
}
```

**For sending latitude and longitude details to IBM Watson IOT platform**

```
#include <WiFi.h>
#include <PubSubClient.h>
```

```
#include <ArduinoJson.h>

WiFiClient wifiClient;

#define ORG "mz6rat"
#define DEVICE_TYPE "ESP8266"
#define DEVICE_ID "12345"
#define TOKEN "123456789"
#define speed 0.034

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Data/fmt/json";
char topic[] = "iot-2/cmd/home/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
PubSubClient client(server, 1883, wifiClient);
void publishData();

const int trigpin=5;
const int echopin=18;
String command;
String data="";
String lat="13.356563";
String lon="80.141428";
String name="point1";
String icon="fa-fire";

long duration;
int dist;

void setup()
{
  Serial.begin(115200);
  pinMode(trigpin, OUTPUT);
  pinMode(echopin, INPUT);
  wifiConnect();
  mqttConnect();
}

void loop() {

  publishData();
  delay(500);

  if (!client.loop()) {
    mqttConnect();
  }
}

void wifiConnect() {
  Serial.print("Connecting to "); Serial.print("Wifi");
  WiFi.begin("Wokwi-GUEST", "", 6);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.print("WiFi connected, IP address: "); Serial.println(WiFi.localIP());
}

void mqttConnect() {
  if (!client.connected()) {
    Serial.print("Reconnecting MQTT client to "); Serial.println(server);
    while (!client.connect(clientId, authMethod, token)) {
    Serial.print(".");
      delay(1000);
    }
    initManagedDevice();
    Serial.println();
  }
```

```
}

void initManagedDevice() {
  if (client.subscribe(topic)) {
     Serial.println(client.subscribe(topic));
    Serial.println("subscribe to cmd OK");
  } else {
    Serial.println("subscribe to cmd FAILED");
  }
}
void publishData()
{
  digitalWrite(trigpin,LOW);
  digitalWrite(trigpin,HIGH);
  delayMicroseconds(10);
  digitalWrite(trigpin,LOW);
  duration=pulseIn(echopin,HIGH);
  dist=duration*speed/2;
  dist=dist/4;
  dist=100-dist;
  if(dist>80){
  lat="13.356563";
  lon="80.141428";
  }else{
    lat="0.000000";
    lon="0.000000";
  }
  DynamicJsonDocument doc(1024);
  String payload;
  doc["Name"]=name;
  doc["Latitude"]=lat;
  doc["Longitude"]=lon;
  doc["Icon"]=icon;
  doc["GasPercent"]=dist;
  serializeJson(doc, payload);
  delay(3000);
  Serial.print("\n");
  Serial.print("Sending payload: ");
  Serial.println(payload);
  if (client.publish(publishTopic, (char*) payload.c_str())) {
    Serial.println("Publish OK");
  } else {
    Serial.println("Publish FAILED");
  }
}
```

## GITHUB AND DEMOLINK:

### GITHUB:

**https://github.com/IBM-EPBL/IBM-Project-40033-1660620585**

### VIDEO LINK:

**https://www.youtube.com/watch?v=6fPLMS4vgeo&feature=youtu.be**