Gas Leakage Monitoring and Alerting System for Industries

PROJECT NAME	**GAS LEAKAGE MONITORING AND ALEARING SYSTEM FOR INDUSTRIES**
TEAM ID	**PNT2022TMID43096**
TEAM MEMBERS	**1.N.LIBINANDHINI** **2.P.KANIPRIYA** **3.R.PRIYADHARSHINI** **4.A.SOWMYA**
BRANCH	**ELECTRONICS AND COMMUNICATION ENGINEERING**

1. INTRODUCTION

1.1 .Project Overview

The purpose of this system to detect gas leakage ,neutralize it,and prevent the explosion.

1.2 . Purpose

Leakage of gas is a major issue in the industrial sector, residential buildings, and gas powered vehicles, one of the preventive methods to stop accidents associated with gas leakage is to install gas leakage detection devices. The focus of this work is to propose a device that can detect gas leakage and alert the owners to avert problems due to gas leakages. The system is based on a microcontroller that employs a gas sensor as well as a stepper motor ,servo motor,ESP 32 module, and a buzzer. The system was designed for gas leakage monitoring and alerts. The output of this research will be significant in averting problems associated with gas leakages now and in future.

2. LITERATURE SURVEY

2.1.Existing Problem

In the present day, they are using gas detectors which detects the gas leakage and intimates the signal to the control room. After getting the signal, they will sent the workers to the specific leak points and they will overcome the problem. The main disadvantage in this system is, its takes more time for this process around 15 to 20 minutes.

2.2. References

ripublication.com

circuitdigest.com

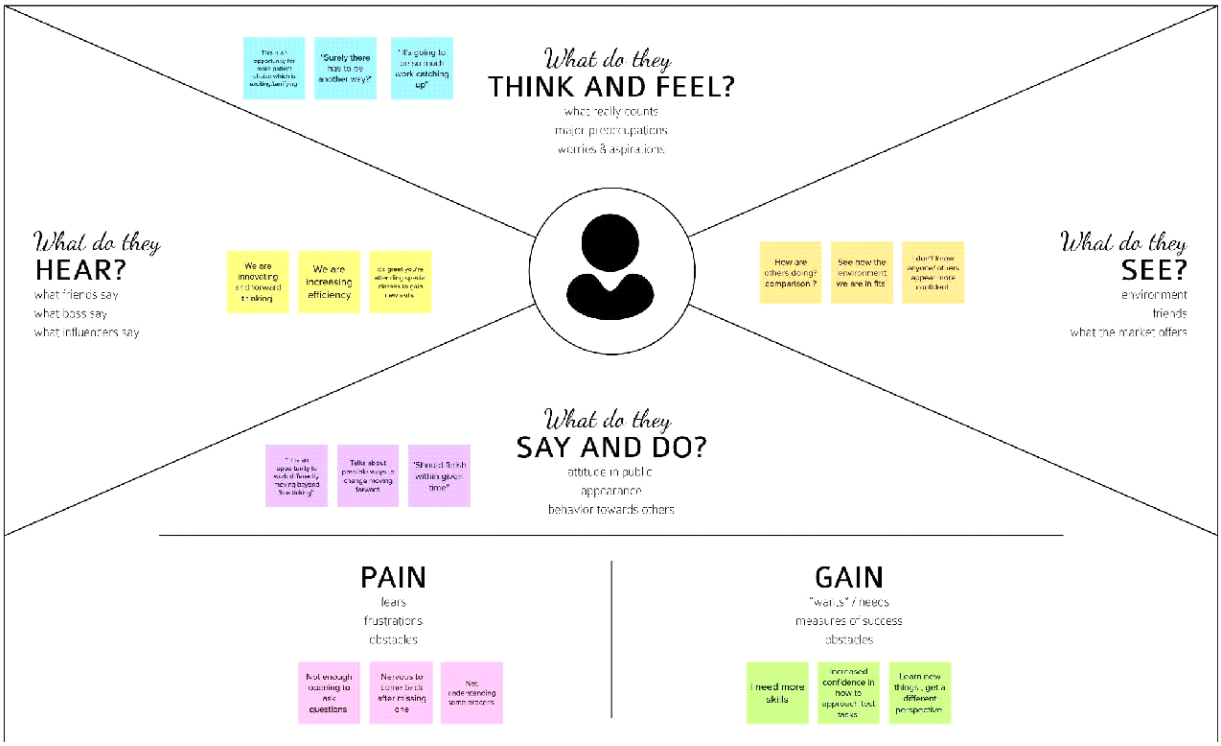
Projectsof8051.com

2.3.Problem Statement Definition

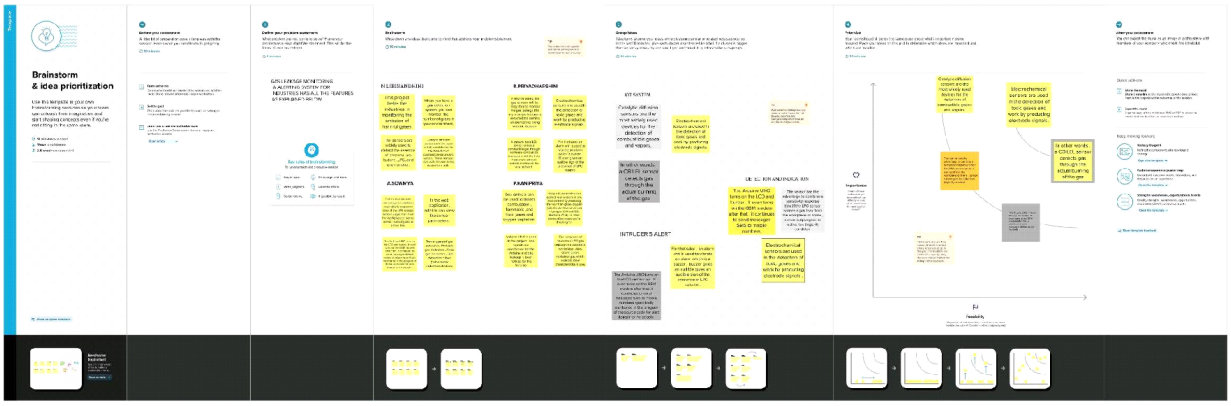
Gas leakage is nothing but the leak of any gaseous molecule from a stove, or a pipeline, or cylinder etc. This can occur either purposefully or even unintendedly. As we are aware that these kinds of leaks are dangerous to our health, and when it becomes explosive it could cause great danger to the people, home, workplace, industry and the environment.Few of the major incidents that took place due to gas leakage include the Bhopal Disaster and the Vizag Gasleak. The Bhopal disaster is known to be the worst industrial accident ever. Approximately 45 tons of Methyl Isocyanate was leaked from this insecticide plant. Methyl Isocyanate is an organic compound and a chemical that could come from the carbamate pesticides. This colorless, poisonous and flammable liquid is something that human beings have to be away from.Vizag Gas leak was a resultant of the escape of styrene that were unattended for a long period. This colorless oily liquid can spread in fumes. So, a detector must be made in such a way that could detect any kind of gas, fume, leak, smoke etc. However harmful and dangerous it can be, the detector could be attached with certain parameters that could help to prevent the issue.

3.IDEATION AND PROPOSED SOLUTION

3.1.Empathy Map Canvas



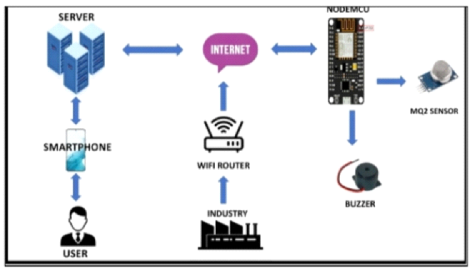
3.2 ideation & Brainstorming



3.3 Proposed Solution

Project team shall fill the following information in proposed solution template

S.No	Parameter	Description
1.	Problem Statement (problem to be solved)	The purpose of this system is to detect gas leakage , neutralize it ,and prevent the explosion.
2.	Idea/ Solution description	Gas detectors can be used to detect combustible, flammable and toxic gases , and oxygen depletion. This type of device is used widely in industry and can be found in locations,such as on oil rigs ,to monitor manufacturing processes and emerging technologies such as photovoltaic. They may be used in firefighting.
3.	Novelty /uniqueness	-Ability to predict the hazardous situation -low cost
4.	Social Impact / Customer Satisfaction	-This model is vital for the industry as there are lot of people to detect the gas leakage prior the fire accident -We have used the IoT technology to make a Gas Leakage Detector for industry which having Smart Alerting techniques involving sending text message to the concerned authority and an ability performing data analytics on sensor readings.

5.	Business Model (Revenue Model)	

3.4. Problem Solution Fit

1. CUSTOMER SEGMENT(S) CS

- Oil, Gas, Polymer Industries
- Hospitals
- Safety Control Personals
- Mining

6. CUSTOMER CONSTRAINTS CC

- Network Connection
- Complexity in Installation
- High budget in installing other products make them to move far from modern technologies

5. AVAILABLE SOLUTIONS AS

- Upgrading to a premium network plan.
- Availing network connection from a reliable Service provider.

2. JOBS-TO-BE-DONE / PROBLEMS J&P

- Suffering from many losses due to gas leakage.
- Having no proper system for controlling or monitoring the leakage.
- Facing heavy budget problems in buying and installing a system for monitoring and controlling

9. PROBLEM ROOT CAUSE RC

- Quality of the material using which the device is made up of plays a vital role in the capability of the device to work in harsh environment.
- Location of the device installation and the network plan used by the user are the cause of Network issue.

7. BEHAVIOUR BE

- Harsh environment is prevailing only on certain industry; thus, the frequency of the said problem is low. In such a case the customer complaints multiple times to get the attention.
- Network issue is very common as most of the industries are located at the country side. Here the contact both the developers and the service providers

3. TRIGGERS TR

- The heavy damages or higher health issues due to the toxic gases urges them to find out a solution as soon as they could possible.
- Usage of the device is portrayed in the news.

4. EMOTIONS: BEFORE/AFTER EM

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

10. YOUR SOLUTION

- Network strength must be boosted in the device
- Device can be manufactured in multiple standards based on the environment.
- Proper evacuation plan and manifestation of emergency drills will help workers to take appropriate step during emergency.

8. CHANNELS OF BEHAVIOUR CH

8.1 ONLINE

- E-Mail to developers
- Online Community

8.2 OFFLINE

- Complaint Letters
- Returning the product is easy

4.REQUIREMENT ANALYSIS

4.1.Functional requirement:

Following are the functional requirements of the proposed solution.

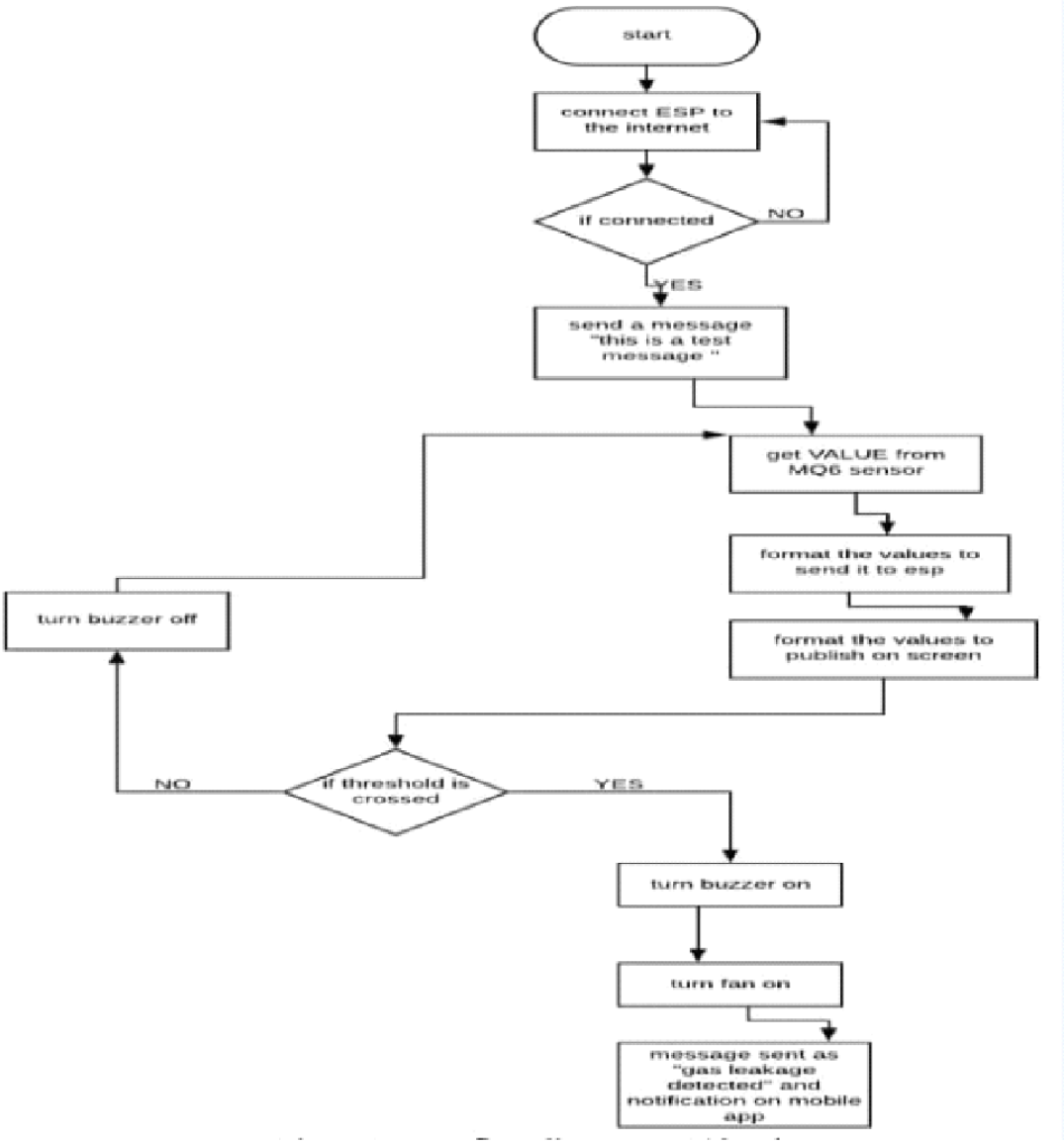
FR NO.	Functional Requirement(Epic)	Sub Requirement (Story /Sub-Task)
FR-1	User Registration	➢ Registration through Form ➢ Offline Registration
FR-2	User Confirmation	➢ Confirmation via Email ➢ Confirmation via OTP
FR-3	User Authentication	➢ User verification through valid User ID and password.
FR-4	User Access	➢ Real time Monitoring of Gas Leakage System, through web portal for Authorized Users.
FR-5	User Alert	➢ User receives an alert through SMS. ➢ Turn on Alerting System in Industry.
FR-6	Review and Feedback	➢ Receive Feedback from Users.

4.2. Non-Functional Requirements:

NFR NO.	Non-Functional Requirement	Description
NFR-1	**Usability**	**➢ Easier Installation process and easy to use**
NFR-2	**Security**	**➢ Data transmission and handling through secured protocols. ➢ Credential encryption & Cloud security.**
NFR-3	**Reliability**	**➢ Only authorized personnel have access to the system. ➢ Assured Data Security and Information conciseness. ➢ Longer Lifetime of Product/Service**
NFR-4	**Performance**	**➢ High Accuracy of gas leakage detection in localized area. ➢ Faster Response to Gas Leakage Detection (SMS alert, valve closing).**
NFR-5	**Availability**	**➢ The user can access the System 24/7. ➢ Real time monitoring system.**
NFR-6	**Scalability**	**➢ The system is scalable even in case of many gas sensors. Or in case of many supervisors**

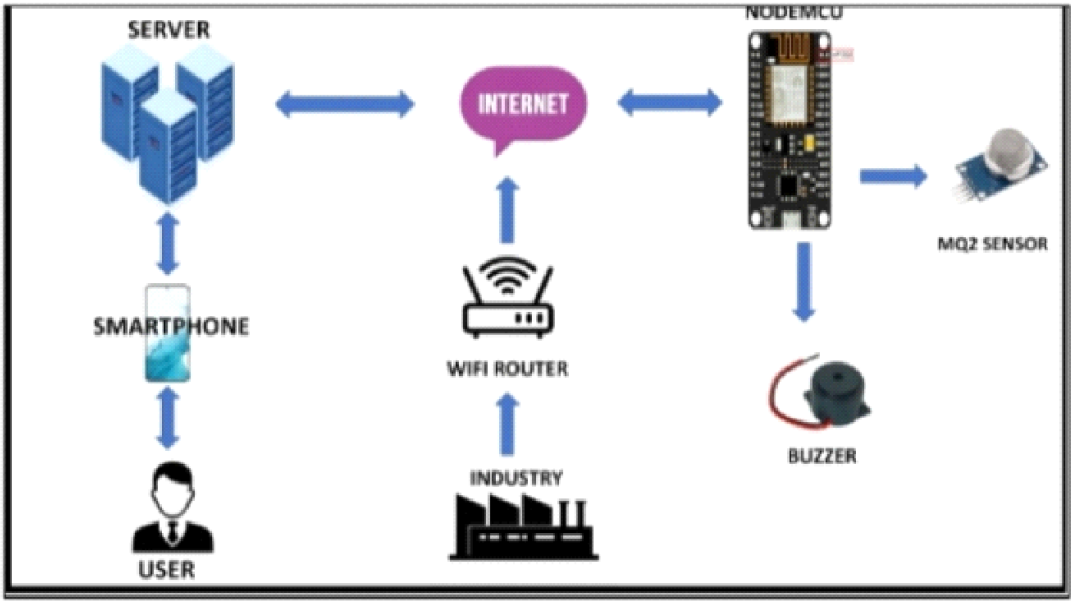
5.PROJECT DESIGN

5.1.Data Flow Diagrams

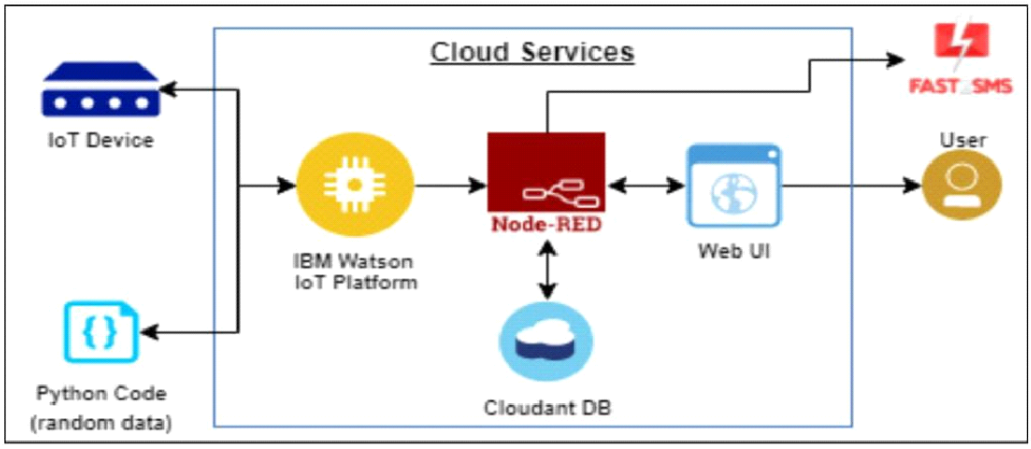

start
connect ESP to the internet
if connected
NO
YES
send a message "this is a test message "
get VALUE from MQ6 sensor
format the values to send it to esp
format the values to publish on screen
if threshold is crossed
NO
YES
turn buzzer off
turn buzzer on
turn fan on
message sent as "gas leakage detected" and notification on mobile app

5.2 Solution & Technical Architecture

SOLUTION ARCHITECTURE



TECHNICAL ARCHITECTURE



5.3.User Stories

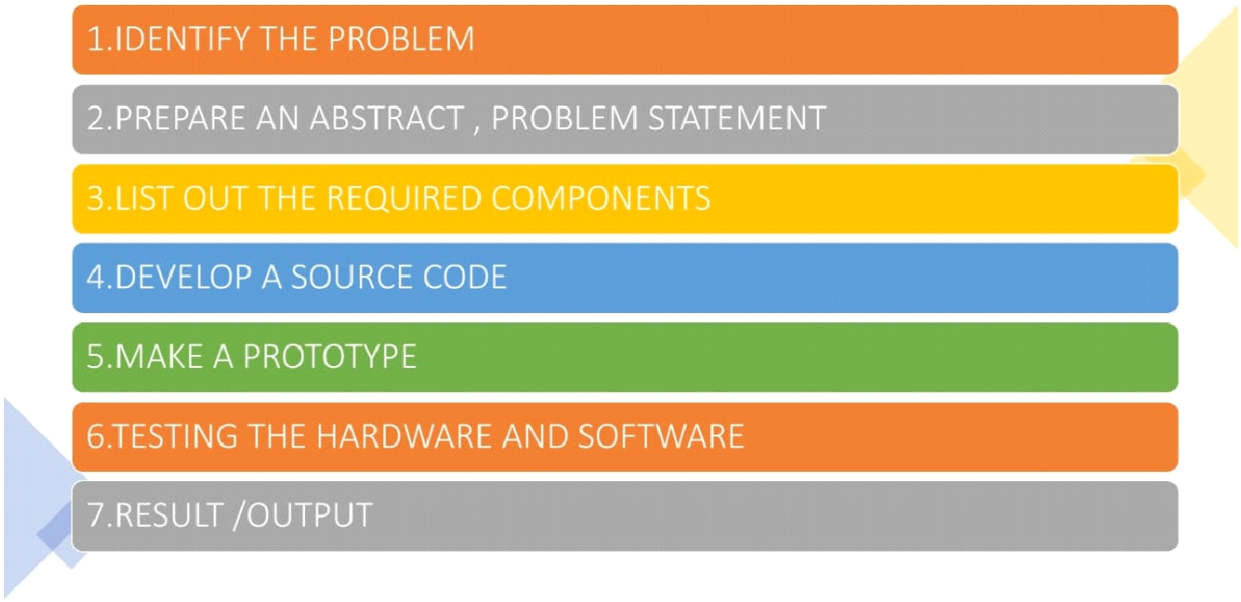
User Type	Functional Requirement(Epic)	User Story Number	User Story /Task	Acceptance criteria	Priority	Release
Customer	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirmin	I can access my account / dashboard .	High	Sprint-1

			g my password.			
Customer	Registration	USN-2	As a user, I will receive confirmation message once I have registered for the application.	I can receive confirmation message & click confirm.	High	Sprint-1
Administrator	Login page	USN-3	As a user entering the username and password which is already existing.	Redirecting to user account.	Medium	Sprint-1
Weather station	Forecasting the current weather	USN-4	As a user, we can monitor the weather fundamentals like (humidity, wind speed, wind direction and rainfall).	Notified about weather conditions.	High	Sprint-1
Controlling the Motor Pump	Controlling	USN-5	It is used to control motors and field sprinklers.	Switching on and off the motor pump manually via mobile application	High	Sprint-2
Fencing	Detecting the motion in certain range	USN-6	Fencing system are helpful in providing security against unauthorized	I can receive notification ; prevention has been taken	High	Sprint-3

			access of human and animal.			
Warehouse management	Collecting database of crops	USN-7	Here famer need to update about expire date of fertilizer and seeds.	Generate the popup message about expire date and stocks and offers	High	Sprint-4

6.PROJECT PLANNING & SCHEDULING

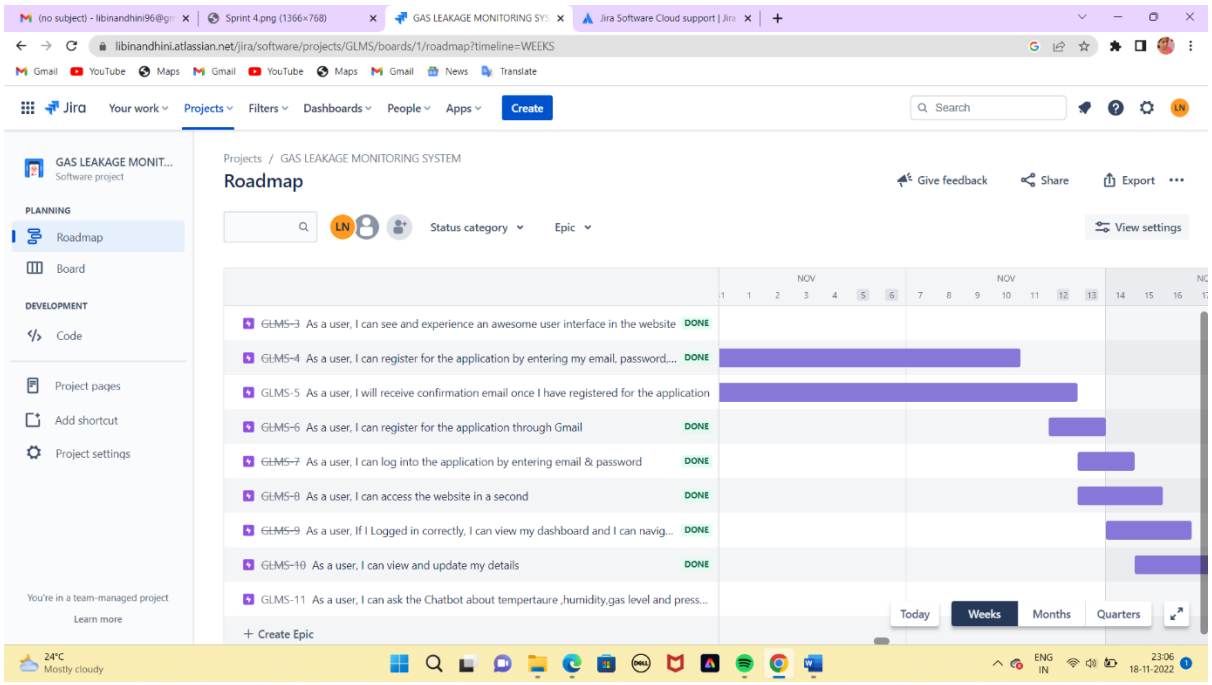
6.1.Sprint Planning & Estimation

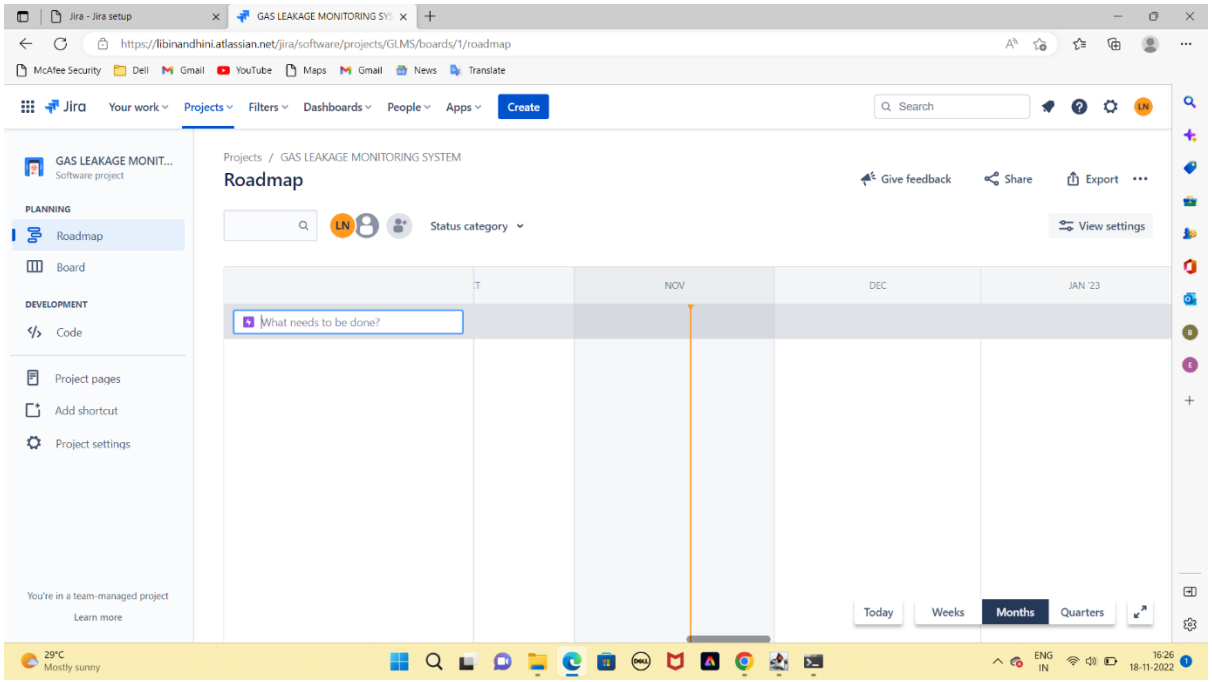


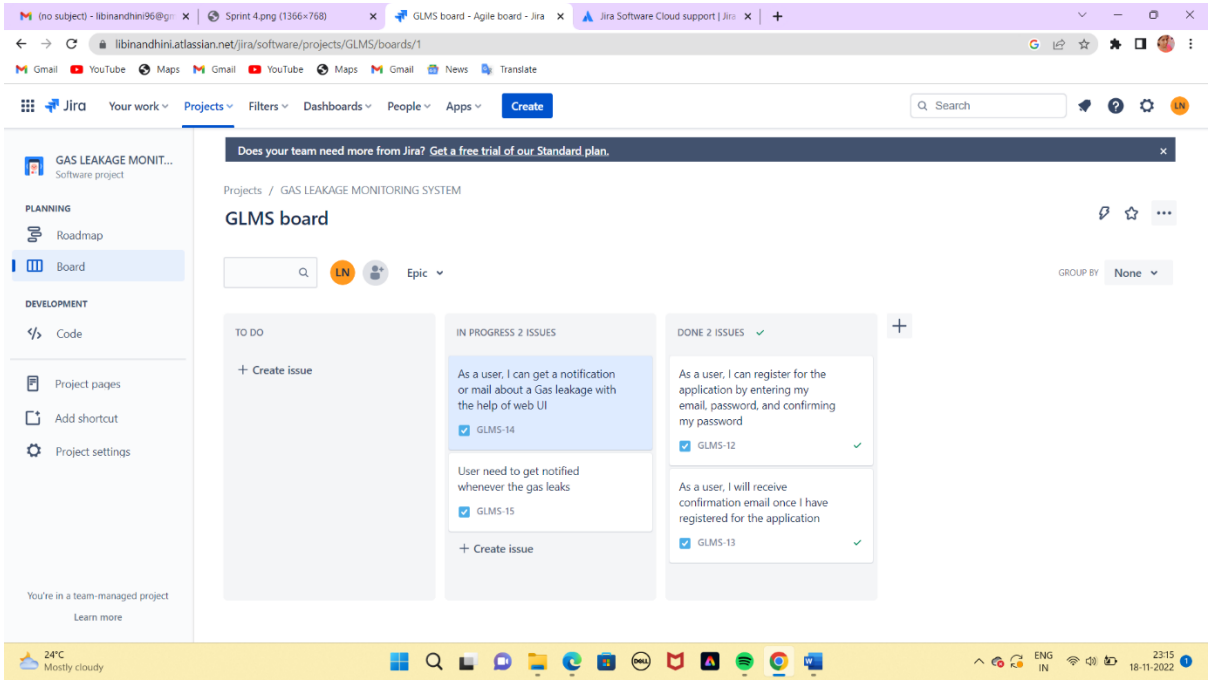
6.2.Sprint Delivery Schedule

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	2022-10-30	2022-11-04	20	2022-11-04
Sprint-2	20	5 Days	2022-11-05	2022-11-04	20	2022-11-04
Sprint-3	20	6 Days	2022-11-10	2022-11-10	20	2022-11-10
Sprint-4	20	4 Days	2022-11-16	2022-11-19	20	2022-11-19

6.3.Reports from JIRA







7.ADVANTAGES

Gas Leakage detection systems are a great way to spend less time setting up and more time on the job. Many factories are required by law to have a gas detection system in place before work can begin. This is also true of drilling rigs. Hardwired gas detectors can take up to two weeks to implement and calibrate before any money can be made on the project. This is

because wires have to be trenched in the ground and buried. Conversely, wireless detectors usually only take a few days at the most to set up. Wireless systems are also made to withstand a wider range of temperatures and weather changes. Hardwired systems sometimes tend to fail in particularly harsh winter climates; wireless detectors can withstand temperatures all the way down to -40 degrees Fahrenheit.
Wireless systems can also be connected to the Internet for available readings at any given time. This allows all safety managers to view gas levels at all points of the project. Systems of this nature are especially beneficial on work sites in which there have been previous accidents and leaks. It is also a great way to closely monitor confined spaces in which higher gas levels can nearly instantly turn the air into poison.

8.DISADVANTAGES

As for disadvantages to the gas leakage detection system, there don't seem to be many. Most users are perfectly happy with the service provided by their wireless system. Some complain that they need to be calibrated more often. This can add up if you are continually buying gas calibration kits. Additionally, the wireless system is not cheap. Anyone looking for an inexpensive detector to do the bare minimum of what is necessary under the law will not be happy with a wireless system. Safety is one of the most important aspects of running a business, especially one that encounters these types of dangers on a regular basis. If you are concerned about budget, a hardwired system will still get the job done.

All things considered, the reliability and real-time level monitoring provided by wireless gas detection systems makes them the top-of-the-line solution for gas leaks. The initial cost may be enough to deter you from pursuing this option, but consider what type of business you run. If you have a small confined space prone to frequent accidents, you will be glad you went with the wireless system. However, if your building is large and open-air, wired systems may be the better solution.

9.CONCLUSION

We have used the IOT technology to make a Gas Leakage Detector for society which having Smart Alerting techniques involving sending text message to the concerned authority and an ability performing data analytics on sensor. This system will be able to detect the gas in environment using the gas sensors. This will prevent form the major harmful problem.

10.FUTURE SCOPE

A mobile gas sensing robot can be constructed to sense the leakage of gas through pipelines as the robot can move on a track which is situated along the length of the pipeline.

11.APPENDIX

Source Code

Including Required Header Files

```
#include &lt;WiFi.h&gt;
#include &lt;PubSubClient.h&gt;
#include &lt;DHTesp.h&gt;
#include &lt;Stepper.h&gt;
#include &lt;ESP32Servo.h&gt;
/*
 NOTE:
//
 As Gas Sensor is not available in Wokwi platform.
 Slide Potentiometer is used instead of Gas Sensor, to variably set
level of gas leakage.
*/
// Defining Constants
#define DHTPIN 15
#define GAS_LEVER 34  // Slide Potentiometer
#define buzzer 13
#define LED 5
const int servoPin = 12;
Servo valve;
DHTesp dhtsensor;
Stepper stepper(1000, 19,21,22,23);

void callback(char* subscribetopic, byte* payload, unsigned int
payloadLength);
#define ORG &quot;wpf8mr&quot;
#define DEVICE_TYPE &quot;ESP32&quot;
#define DEVICE_ID &quot;GAS_LEAKAGE_MONITOR&quot;
#define TOKEN &quot;0123456789&quot;

String data3;
float h, t, g;

int pos=0;
boolean valve_open=true;
//-------- Customise the above values --------
char server[] = ORG &quot;.messaging.internetofthingsx0.ibmcloud.com&quot;;
char publishTopic[] = &quot;iot-2/evt/Data/fmt/json&quot;;
char subscribetopic[] = &quot;iot-2/cmd/test/fmt/String&quot;;
char authMethod[] = &quot;use-token-auth&quot;;
char token[] = TOKEN;
char clientId[] = &quot;d:&quot; ORG &quot;:&quot; DEVICE_TYPE &quot;:&quot; DEVICE_ID;

//-------------------------------------------
WiFiClient wifiClient;
PubSubClient client(server, 1883, callback ,wifiClient);
void setup()
{
  Serial.begin(115200);
  dhtsensor.setup(DHTPIN,DHTesp::DHT22);
```

```
  stepper.setSpeed(100);
  valve.attach(servoPin);
  pinMode(GAS_LEVER, INPUT);
  pinMode(buzzer,OUTPUT);
  delay(10);
  Serial.println();
  wificonnect();
  mqttconnect();
  valve.write(90);

}
void loop()
{
  TempAndHumidity data=dhtsensor.getTempAndHumidity();
  t=data.temperature;
  h=data.humidity;
  g=map(int(analogRead(GAS_LEVER)), 0, 4095, 200, 2000);
  Serial.print(&quot;temperature:&quot;);
  Serial.println(t);
  Serial.print(&quot;Humidity:&quot;);
  Serial.println(h);
  Serial.print(&quot;Gas Level:&quot;);
  Serial.println(g);

  if(g&gt;500){

    tone(buzzer, 1000);
    stepper.step(1000);
    valve.write(180);
  }
  else{
    valve.write(90);
    noTone(buzzer);
  }
  PublishData(t, h, g);
  delay(1000);
  if (!client.loop()) {
    mqttconnect();
  }
}

/*.....retrieving to
Cloud.....*/
void PublishData(float temp, float humid, float gas_level) {
  mqttconnect();
  String payload = &quot;{\&quot;temperature\&quot;:&quot;;
  payload += temp;
  payload += &quot;,&quot; &quot;\&quot;humidity\&quot;:&quot;;
  payload += humid;
```

```
  payload += &quot;,&quot; &quot;\&quot;gas_level\&quot;:&quot;;
  payload += gas_level;
  payload += &quot;}&quot;;

  Serial.print(&quot;Sending payload: &quot;);
  Serial.println(payload);

  if (client.publish(publishTopic, (char*) payload.c_str())) {
    Serial.println(&quot;Publish ok&quot;);
  } else {
    Serial.println(&quot;Publish failed&quot;);
  }

}
void mqttconnect() {
  if (!client.connected()) {

    Serial.print(&quot;Reconnecting client to &quot;);
    Serial.println(server);
    while (!!!client.connect(clientId, authMethod, token)) {
      Serial.print(&quot;.&quot;);
      delay(500);
    }

     initManagedDevice();
     Serial.println();
  }
}
void wificonnect()
{
  Serial.println();
  Serial.print(&quot;Connecting to &quot;);
  WiFi.begin(&quot;Wokwi-GUEST&quot;, &quot;&quot;, 6);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(&quot;.&quot;);
  }
  Serial.println(&quot;&quot;);
  Serial.println(&quot;WiFi connected&quot;);
  Serial.println(&quot;IP address: &quot;);
  Serial.println(WiFi.localIP());
}
void initManagedDevice() {
  if (client.subscribe(subscribetopic)) {
    Serial.println((subscribetopic));
    Serial.println(&quot;subscribe to cmd OK&quot;);
  } else {
    Serial.println(&quot;subscribe to cmd FAILED&quot;);
  }
```

```
}
void callback(char* subscribetopic, byte* payload, unsigned int
payloadLength)
{

  Serial.print(&amp;quot;callback invoked for topic: &amp;quot;);
  Serial.println(subscribetopic);
  for (int i = 0; i &amp;lt; payloadLength; i++) {

    data3 += (char)payload[i];
  }

  Serial.println(&amp;quot;data: &amp;quot;+ data3);

  data3=&amp;quot;&amp;quot;;
  }
```

GitHub & Project Demo Link

https://github.com/IBM-EPBL/IBM-Project-40551-1664442193