GAS LEAKAGE MONITORING AND ALERTING SYSTEM

FOR INDUSTRIES

A PROJECT REPORT

Submitted by

PON AJIRITHA M
SANGHAVAI BHUVANESWARI D
SARANYA M
SUBARNA M

TEAM ID:PNT2022TMID35489

CHAPTER 1
INTRODUCTION

1.1 PROJECT OVERVIEW

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

1.2 PURPOSE

- An IoT-enabled gas monitoring system is designed specially to prevent explosions and fire disasters in the facilities and thus save human lives.
- The gas detectors can be used for the detection of combustible, flammable and poisonous gases and for loss of oxygen, and also to detected a gas leak or other pollutants
- When the fire alarm system detects smoke, heat, or water movement, it alerts occupants of the building using both audible and visible alarms

CHAPTER 2
LITERATURE SURVEY

2.1 EXISTING PROBLEM

- Unidentifiable gas leaks give rise to explosions that are harmful to the employees working in the hazardous environment.
- In the automotive industries like oil and gas, hotels, and places where flammable gases are used in abundance, a gas detection system is a basic requirement for safety.

2.2 REFERENCES

- Design of an IoT Based Gas Wastage Monitoring, Leakage Detecting and Alerting System ,Md. Ibtida Fahim;Nowshin Tabassum;Abrar Ahamed Habibullah;Aritra Sarker;Sayeda Islam Nahid;Mohammad Monirujjaman Khan,2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)
- Gas Leakage with Auto Ventilation and Smart Management System Using IoT Afsana Mim Anika;Ms. Nasrin Akter;Md.Niamul Hasan;Jannatul Ferdous Shoma;Abdus Sattar,2021 International Conference on Artificial Intelligence& Smart Systems (ICAIS)
- Gas Leakage and Fire Detection using Raspberry Pi Sourabh Jamadagni;Priyanka Sankpal;Shwetali Patil;Nikita Chougule;Shailesh Gurav,2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)

2.3 PROBLEM STATEMENT DEFINITION

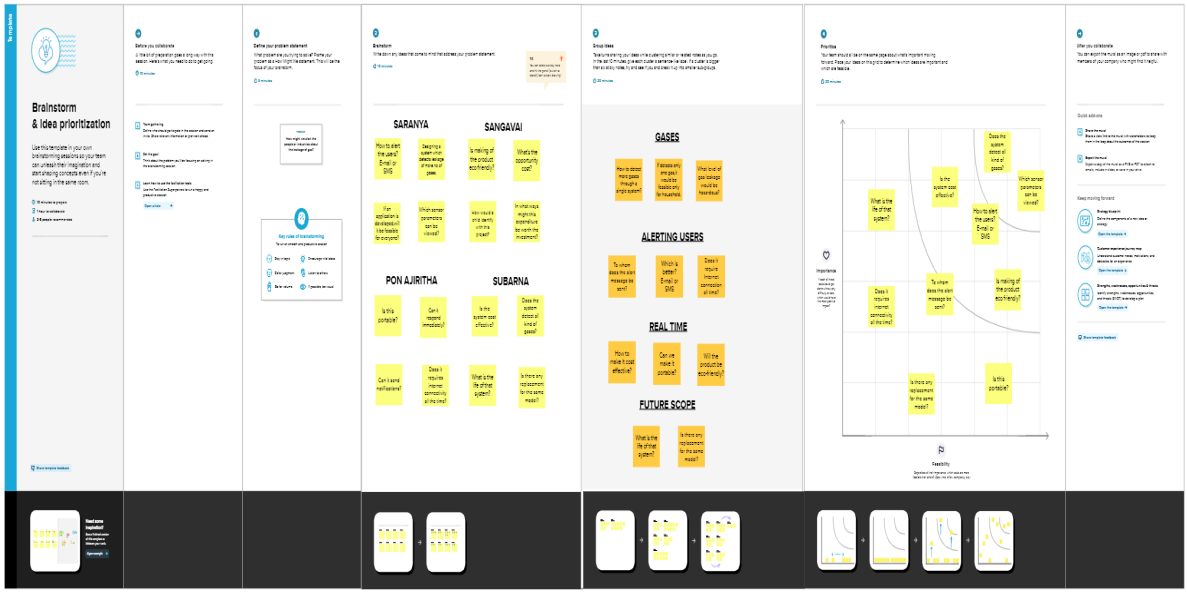
An chemical company ,in the outskirts of the city ,is frequently exposed gas leakage ,this affects the people in and around that place ,other organic life such as animals and plants and mainly the enviroment.

Who does the problem affect?	Elderly people,children ,home makers and industrial workers.
Where these problem occurs?	Home, flats and industries(where gases are used).
What is the issue?	The leakage of gases only can be detected by human nearby and if there are no human nearby, it cannot be detected. But sometimes it cannot be detected by human who has a low sense of smell. And in many cases the large scale leakage might be detected in the later stages which turns out to be disastrous.
When does gas leakage occur ?	Devices such as water heater, dryer, stove, or fireplace,as these appliances age, the seals that connect the piping can corrode and eventually create a gas leak.
Why is the solution for the issue is important?	A gas detector can sound an alarm or send an notification to operators in the area where the leak is occurring. And there are many gases that can be harmful to organic life, such as humans or animals and thus they need to be evacuated from the place of leakage.

CHAPTER 3
IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

3.2 IDEATION AND BRAINSTORMING



3.3 PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	The aim of this project is to detect the leakage of combustible,flammable,poisonous and harmful gases and alerting users about the leakage.
2.	Idea / Solution description	The leakage of harmful gases in the industries will be detected with the help of sensors and the same would be intimated to the user via web application.
3.	Novelty / Uniqueness	Web application and SMS alert is used to notify the user about the leakage of gases which is a great advantage and uniqueness of this system.
4.	Social Impact / Customer Satisfaction	By detecting the leakage of harmful gases and alerting people,this project can prevent the industries from the huge disaster and loss and can also save the life of people residing near the industries.
5.	Business Model (Revenue Model)	This system is of much use for industries that deal with harmful gases.So such industries take lots of precautious measures and this model will be one best solution for them. This shows that the model can earn good revenue.
6.	Scalability of the Solution	We can add an automatic shutoff device which will turn off the gas supply whenever leakage is detected.Apart from its usage in industries, people living in apartments can also avail this system.

3.4 PROBLEM SOLUTION FIT

Project Title: Gas Leakage monitoring & Alerting system for Industries **Project Design Phase-I - Solution Fit Template** **Team ID:** PNT2022TMID35489

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Who is your customer?
i.e. working parents of 0-5 y.o. kids

Industries using harmful gases for their product manufacture.
Household where gas is used in cylinders for cooking purpose.

6. CUSTOMER CONSTRAINTS CC

What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.

Network Connection,Alarm sound,Android device or website

5. AVAILABLE SOLUTIONS AS

Which solutions are available to the customers when they face the problem
or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking

The sensor should sense the leakage of any gas and alert the concerned authorities regarding the leakage. Useful in real time.

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

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

Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides.

1.Detect the leakage of gas and identify which gas has leaked.
2.Alert the customer as well as concerned authorities through E-mail or SMS.

9. PROBLEM ROOT CAUSE RC

What is the real reason that this problem exists?
What is the back story behind the need to do this job?
i.e. customers have to do it because of the change in regulations.

The main root cause of this problem is the industries using the harmful gases for their own benefits without caring about the society.

7. BEHAVIOUR BE

What does your customer do to address the problem and get the job done?
i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

The customer would expect an system which includes an gas sensor to detect the leakage of gases and send necessary information to cloud from where the notification is sent to customer or authority via e-mail or SMS.

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS TR

What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news

Knowing that if harmful gases are released,their lives would be at risk would trigger customer to act.

4. EMOTIONS: BEFORE / AFTER EM

How do customers feel when they face a problem or a job and afterwards?
i.e. lost, insecure > confident, in control - use it in your communication strategy & design.

BEFORE: People would be insecure about their life
AFTER: Customers would be confident that alert message would have reached the authority and they would take necessary steps to save them.

10. YOUR SOLUTION SL

If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.
If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.

On detection of leakage of any gas,the sensor should sense the gas and the system should alert the user and concerned authority regarding the leakage.Even,we can extend this to switch off the gas once the gas leakage is sensed.

8.CHANNELS of BEHAVIOUR CH

8.1 ONLINE
What kind of actions do customers take online? Extract online channels from #7

An website or mobile application which automatically sends notification to customer as well as authority via E-mail or SMS once leakage is detected.

8.2 OFFLINE
What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

The gas sensor which detects the leakage of gases.

Identify strong TR & EM

CHAPTER 4
REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Detection of Leakage of Gases	Detection of gases through sensor. Detect how much amount of gas has leaked. Detect which gas has been leaked and what is the permissible level and has much has it crossed the level.
FR-2	Alerting the user.	Notification via Email Notification via SMS
FR-3	Notification to the admin.	Notification via E-mail or SMS Notifying the exact location where the incident has occurred. Update the person with route to the location.
FR-4	Creation of web application	Update the website with location of place where leakage has happened. Update the level of leakage in the particular industry.

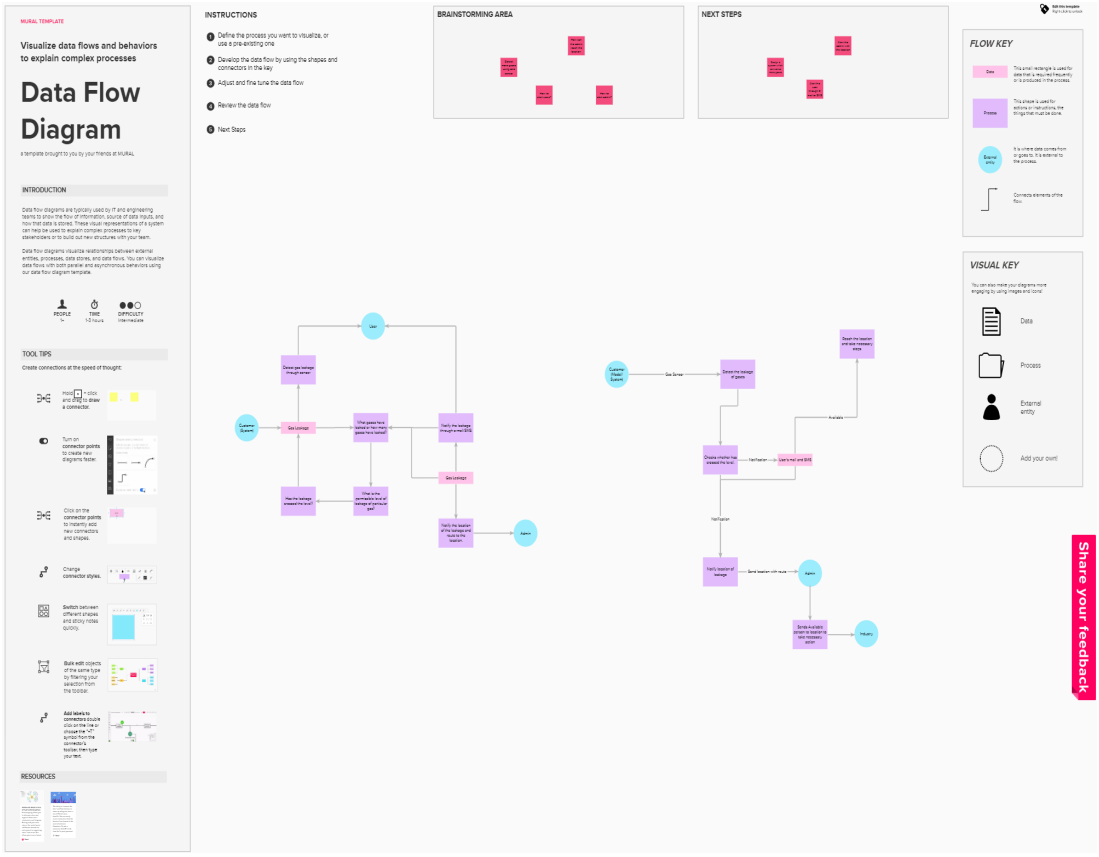
4.2 NON-FUNCTIONAL REQUIREMENTS

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

NFR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The user who has basic knowledge in operating smart phones and internet. The user who knows about the harmfulness of gases.
NFR-2	**Security**	The web application should be accessed by users and admins only through protected login credentials.
NFR-3	**Reliability**	The user must be notified at the correct time so that damage can be prevented. The admin must be notified about the exact location along with the route.
NFR-4	**Performance**	The user and admin must be alerted through notification immediately within seconds to prevent damage.
NFR-5	**Availability**	Once the notification reaches the admin, he must check if some person is available so that he can be sent to the place where leakage has occurred. If not, he must atleast inform the user about how long it will take to reach them.
NFR-6	**Scalability**	There must be at least 20-30 people to address the problem immediately once notified. Leakage must be detected simultaneously at many places.

CHAPTER 5
PROJECT DESIGN

5.1 DATA FLOW DIAGRAM



5.2 SOLUTION AND TECHNICAL ARCHITECTURE

TECHNICAL ARCHITECTURE OF GAS LEAKAGE MONITORING AND ALERTING SYSTEM FOR INDUSTRIES

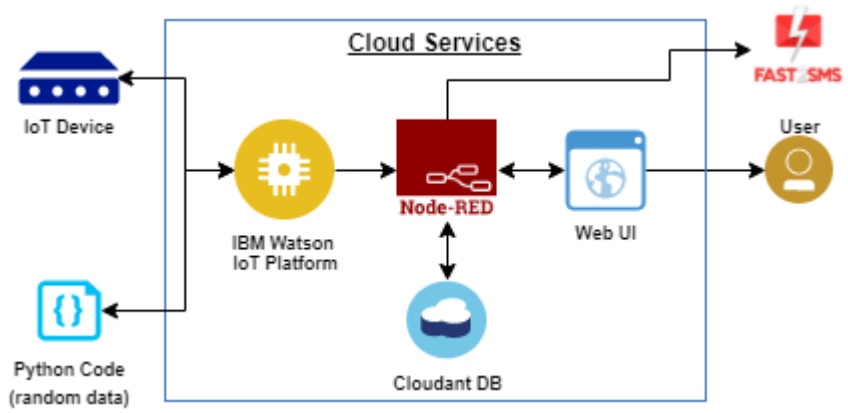


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	Gas Sensor	Gas sensor converts the components and concentrations of various gases into standard electrical signals by using specific physical and chemical effects.	Arduino Software IDE
2.	Buzzer	A buzzeror beeper is an audiosignaling device, which may be mechanical, electromechanical, or piezoelectric.	Arduino Software IDE
3.	Application Building	The process of creating a computer program or a set of programs to perform the different tasks thata business requires.	Python, HTML, CSS
4.	WebApplication	A Web application is an application program that is stored on a remote server and delivered over the internet through a browser interface.	MIT App Invertor

5.	Mobile Application	A software application developed specifically for use on small, wireless computing devices, such as smartphones and tablets, ratherthan desktop or laptop computers	MIT App Invertor
6.	Database	Data Type,Configurations etc.	MySQL, NoSQL, etc.
7.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
8.	File Storage	File storagerequirements	IBM Block Storage or Other Storage Service or Local Filesystem
9.	Node MCU	The Node MCU is an open-source software and hardware development environment built around an inexpensive System-on-a-Chip (SoC) called the ESP8266	Arduino Software IDE
10.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud	Local, Cloud Foundry, Kubernetes, etc.

Table-2: Application Characteristics:

S. No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Technology of Opensource framework
2.	IOTbased communication protocols	Enable IOT devices to communicate with other devices, applications, and services running in thecloud. The internet relieson standardized protocols to ensure communication between heterogeneous devices is secure and reliable	Bluetooth LE, IPv6 technologies(responsible for the logicaldevice addressing and routing of network traffic)

3.	Scalable Architecture	Justifies the scalability of architecture (3 – tier, Micro-services)	Technology used
4.	Availability	Justifies the availability of application (e.g. use ofload balancers, distributed servers etc.)	Technology used
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	Technology used

5.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (System)	Detection	USN-1	As a system, it must sense/detect the leakage of any gas through the sensor.	It must detect the leakage of gas.	High	Sprint-1
		USN-2	As a system , it must detect the amount of gas leaked.	It must measure the exact amount of gas leaked.	Medium	Sprint-2
		USN-3	As a system, it must update the website about the level of leakage in industries surrounding the area.	Web application must be updated with the leakage.	Medium	Sprint-2

Customer (User)	Registration	USN-1	As a user, I must first register my e-mail and mobile number in the website.	I can register my mobile number and E-mail.	High	Sprint-1
		USN-2	As a user, I must receive confirmation mail and SMS on registration.	I can make sure that registration is successful.	High	Sprint-1
	Login	USN-3	As a user, I can login into the web application through e-mail and password.	I can login and view the web application.	Medium	Sprint-3
	Dashboard	USN-4	As a user, I can access the dashboard and make use of available resources.	I can access the dashboard to make use of resources.	Low	Sprint-4
	Notification	USN-5	As a user, I must receive an E-mail once the leakage is detected.	I must receive an e-mail to the registered mail id about the leakage.	High	Sprint-2
		USN-6	As a user, I must receive an SMS once the leakage is detected.	I must receive SMS to the registered Mobile number about the leakage.	High	Sprint-2

Administrator	Registration	USN-1	As an admin, I must first register myself in the portal using E-mail.	I can register myself as an admin in the website.	High	Sprint-2
		USN-2	As an admin, I must receive confirmation mail.	I must receive mail so that my registration is successful.	Medium	Sprint-2
	Login	USN-3	As an admin, I can login into the web application through e-mail and password.	I can login and view the web application.	Medium	Sprint-3
	Dashboard	USN-4	As an admin, I can access the dashboard and make use of available resources.	I can access the dashboard to make use of resources.	Low	Sprint-4
	Notification	USN-5	As an admin, I must receive information about the leakage along with location.	I must be notified with the location of the leakage.	High	**Sprint-1**
	Allocation	USN-6	As an admin, I must allot particular person to look after the leakage in a particular location.	I must allot a person to a particular place.	Medium	**Sprint-2**
		USN-7	As an admin, I must share exact location and route to the person.	The person must know the location and route to the place.	High	Sprint-2

CHAPTER 6
PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Point	Priority	Team Members
Sprint-1	Detection	USN-1	As a system, it must sense/detect the leakageof any gas through the sensor.	4	High	M. Saranya
Sprint-2		USN-2	As a system, it must detectthe amount of gas leaked.	3	Medium	M.Pon Ajiritha
Sprint-2		USN-3	As a system, it must update the website aboutthe level of leakage in industries surrounding the area.	3	Medium	M.Saranya
Sprint-1	Registration	USN-1	As a user, I must first register my e-mail andmobile number in the website.	3	High	M. Pon Ajiritha
Sprint-1		USN-2	As a user, I must receive confirmation mail andSMSon registration.	2	Low	D. Sanghavai Bhuvaneswari
Sprint-2	Login	USN-3	As a user, Ican login into the web application through e-mail and password.	3	Medium	D.Sanghavai Bhuvaneswari
Sprint-4	Dashboard	USN-4	As a user, I can access the dashboard andmake use of available resources.	3	Low	M.Saranya
Sprint-3	Notification	USN-5	As a user,I must receivean E-mail oncethe leakage is detected.	3	High	M.Saranya

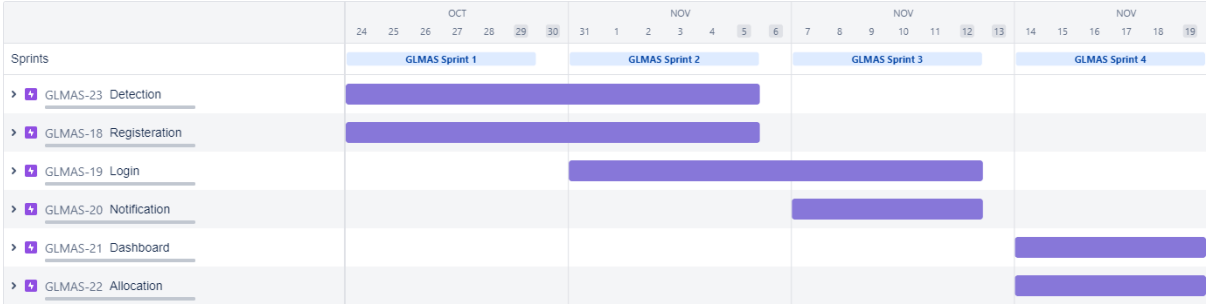
Sprint-3		USN-6	As a user, I must receive an SMS once theleakage is detected.	3	High	M.Pon Ajiritha
Sprint-1	Registration	USN-1	As an admin, I must first register myself in theportal using E-mail.	3	High	M.Subarna

6.2 SPRINT DELIVERY SCHEDULE

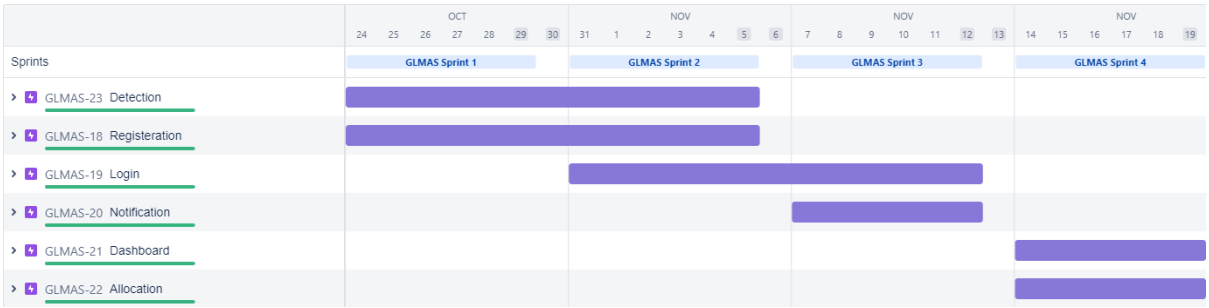
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on PlannedEnd Date)	Sprint Release Date (Actual)
Sprint-1	12	6 Days	24 Oct 2022	29 Oct 2022		29 Oct 2022
Sprint-2	12	6 Days	31 Oct 2022	05 Nov 2022		05 Nov 2022
Sprint-3	12	6 Days	07 Nov 2022	12 Nov 2022		12 Nov 2022
Sprint-4	12	6 Days	14 Nov 2022	19 Nov 2022		19 Nov 2022

6.3 REPORTS FROM JIRA

BEFORE COMPLETING TAKS



AFTER COMPLETING TASKS

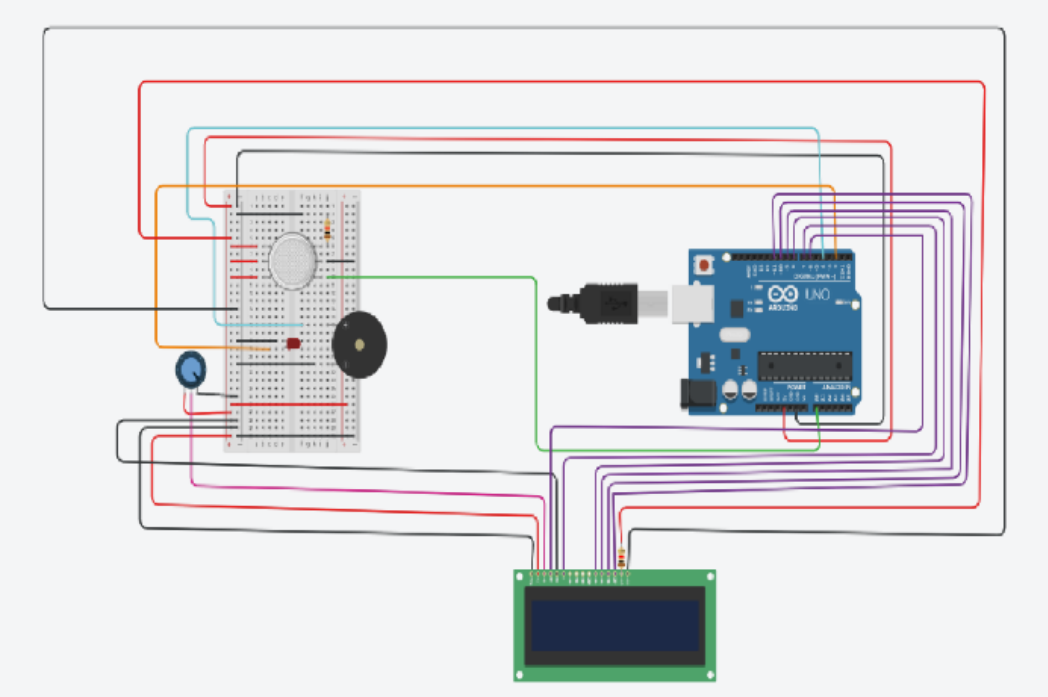


CHAPTER 7
CODING AND SOLUTIONING

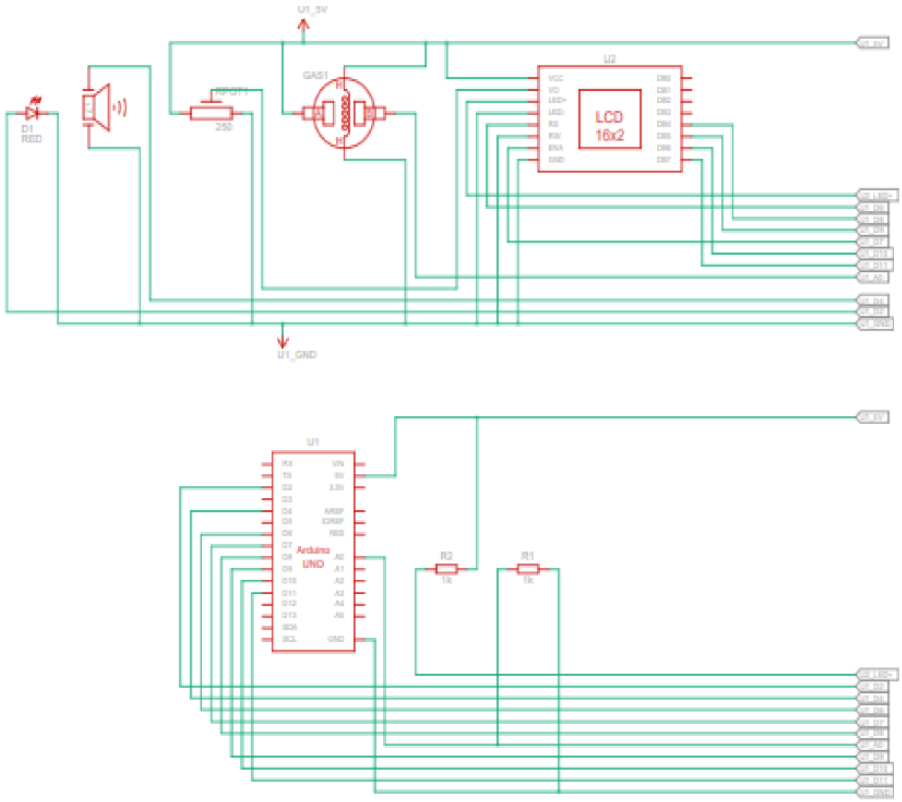
7.1 FEATURE 1

TINKERCAD

CIRCUIT



SCHEMATIC DIAGRAM



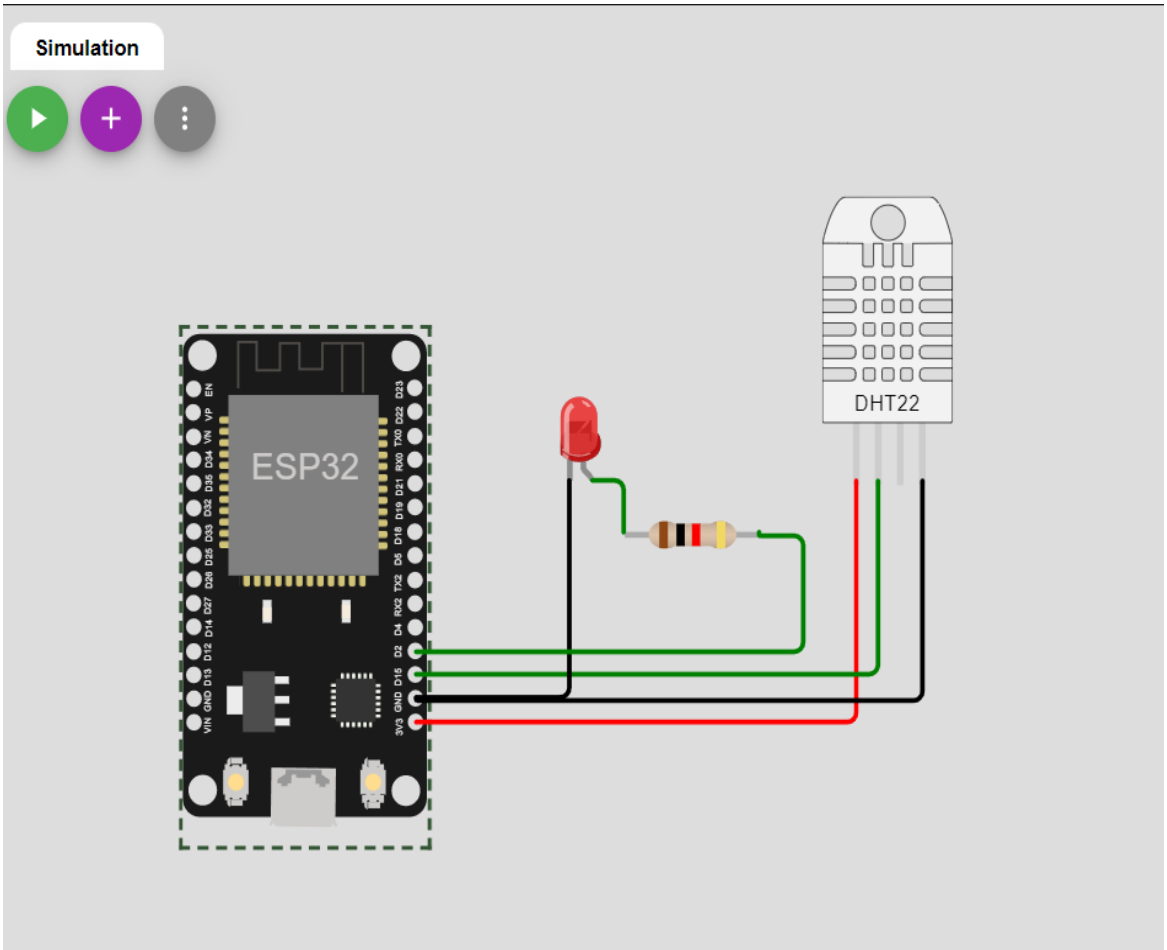
CODE

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(6, 7, 8, 9, 10, 11);
float gasPin = A0;
float gasLevel;
int ledPin = 2;
int buzzPin = 4;

void setup(){
  pinMode(ledPin, OUTPUT);
  pinMode(gasPin,INPUT);
  Serial.begin(9600);
  lcd.begin(16, 2);
  lcd.setCursor(0,0);
  lcd.print("    WELCOME    ");
  lcd.setCursor(0,2);
  lcd.print("GAS ALERT SYSTEM");
  delay(1000);
  lcd.clear();
}
```

```
void loop(){

  gasLevel = analogRead(gasPin);
  gasDetected(gasLevel);
  buzzer(gasLevel);

}

void gasDetected(float gasLevel){
  if(gasLevel >= 250){
   digitalWrite(buzzPin,HIGH);
   digitalWrite(ledPin,HIGH);
   lcd.setCursor(0,0);
   lcd.print("   GAS:");
   lcd.print(gasLevel);
   lcd.setCursor(0,2);
   lcd.print("     ALERT     ");
   delay(1000);
   lcd.clear();
  }
  else{
   digitalWrite(ledPin,LOW);
   digitalWrite(buzzPin,LOW);
   lcd.setCursor(0,0);
   lcd.print("   GAS:");
   lcd.print(gasLevel);
   lcd.setCursor(0,2);
   lcd.print("      SAFE      ");
   delay(1000);
   lcd.clear();
  }
}
void buzzer(float gasLevel){
if(gasLevel>=250)
  {
  for(int i=0; i<=30; i=i+10)
  {
  tone(4,i);
  delay(400);
  noTone(4);
  delay(400);
  }
  }
}
```

WOKWI

CIRCUIT



CODE

```
#include <WiFi.h>//library for Wi-fi
#include <PubSubClient.h>//library for MQTT
#include "DHT.h"// Library for DHT 11

#define DHTPIN 15 // what pin we're connected to
#define DHTTYPE DHT22 // define type of sensor DHT 11
#define LED 2

DHT dht (DHTPIN, DHTTYPE);
void callback(char* subscribetopic, byte* payload, unsigned int
payloadLength);
```

```
//-------credentials of IBM Accounts------
#define ORG "d4fpcb"//IBM ORGANISATION ID
#define DEVICE_TYPE "abcde" //Device type mentioned in IBM Watson
IOT Platform
#define DEVICE_ID "123456" //Device ID mentioned in IBM Watson IOT
Platform
#define TOKEN "1234567890" //Token
String data3;
float h, t;
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";//
Server Name
char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and
type of event perform and format in which data to be send
char subscribetopic[] = "iot-2/cmd/command/fmt/String";//
cmd REPRESENT command type AND COMMAND IS TEST OF FORMAT STRING
char authMethod[] = "use-token-auth";// authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id
WiFiClient wifiClient; // creating the instance for wificlient
PubSubClient client(server, 1883,wifiClient);
void setup()
{
Serial.begin(115200);
dht.begin();
pinMode(LED,OUTPUT);
delay(10);
Serial.println();
wificonnect();
mqttconnect();
}
void loop()
{
h = dht.readHumidity();
t = dht.readTemperature();
Serial.print("temp:");
Serial.println(t);
Serial.print("Humid:");
Serial.println(h);
PublishData(t, h);
delay(1000);
if (!client.loop()) {
mqttconnect();
}
}
```

```
void PublishData(float temp, float humid) {
mqttconnect(); //function call for connecting to IBM
String payload = "{\"temp\":";
payload += temp;
payload += "," "\"Humid\":";
payload += humid;
payload += "}";
Serial.print("Sending payload: ");
Serial.println(payload);
if (client.publish(publishTopic, (char*) payload.c_str())) {
Serial.println("Publish ok");
}
 else {
Serial.println("Publish failed");
}
}

void mqttconnect() {
if (!client.connected()) {
Serial.print("Reconnecting client to ");
Serial.println(server);
while (!!!client.connect(clientId, authMethod, token)) {
Serial.print(".");
delay(500);
}
initManagedDevice();
Serial.println();
}
}
void wificonnect() //function defination for wificonnect
{
Serial.println();
Serial.print("Connecting to ");
WiFi.begin("Wokwi-GUEST", "", 6);//passing the wifi credentials to
establish the connection
while (WiFi.status() != WL_CONNECTED) {
delay(500);
Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
}
```

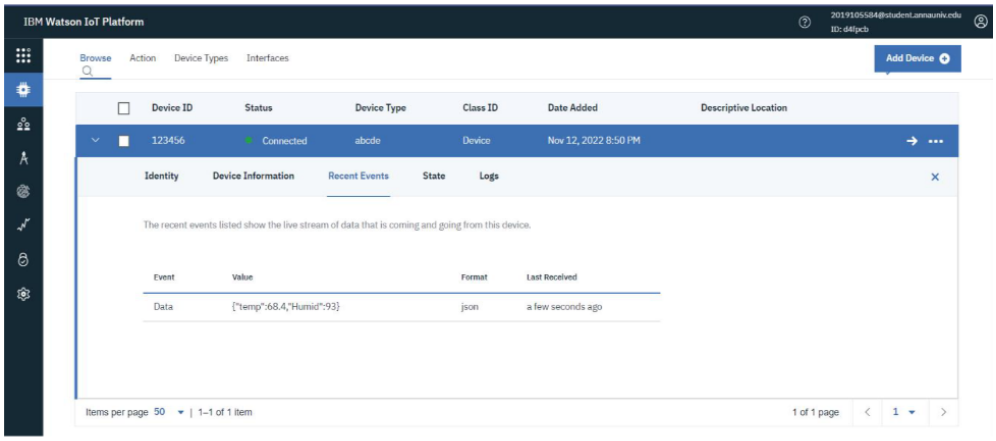
```
void initManagedDevice() {
if (client.subscribe(subscribetopic)) {
Serial.println((subscribetopic));
Serial.println("subscribe to cmd OK");
}
 else {
Serial.println("subscribe to cmd FAILED");
}
}
```

7.2 FEATURE 2

WOKWI OUTPUT

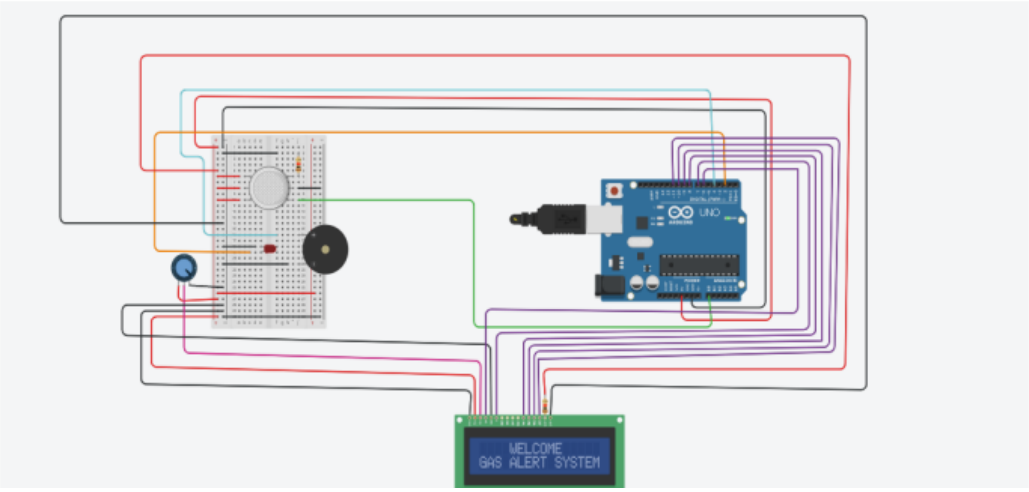
```
Connecting to .....
WiFi connected
IP address:
10.10.0.2
Reconnecting client to d4fpcb.messaging.internetofthings.ibmcloud.com
iot-2/cmd/command/fmt/String
subscribe to cmd OK

temp:68.40
Humid:93.00
Sending payload: {"temp":68.40,"Humid":93.00}
Publish ok
```

IBM CLOUD OUTPUT

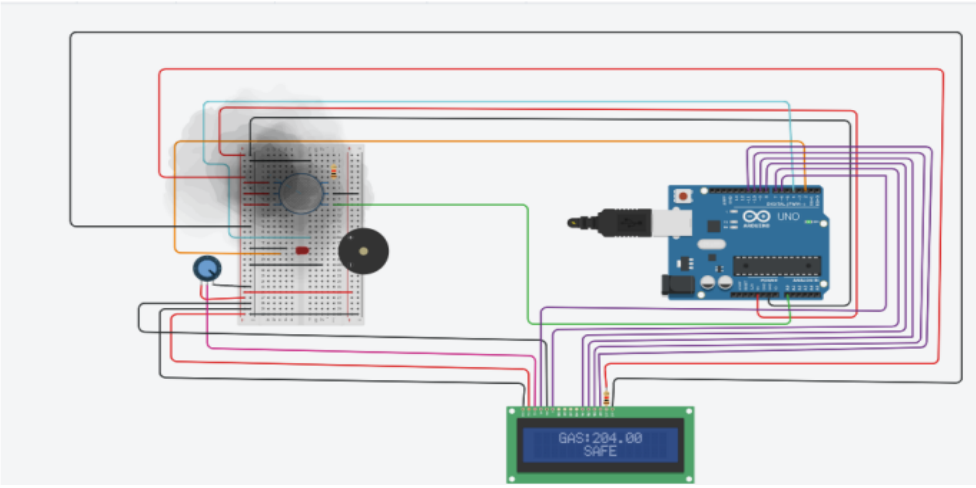


TINKERCAD OUTPUT

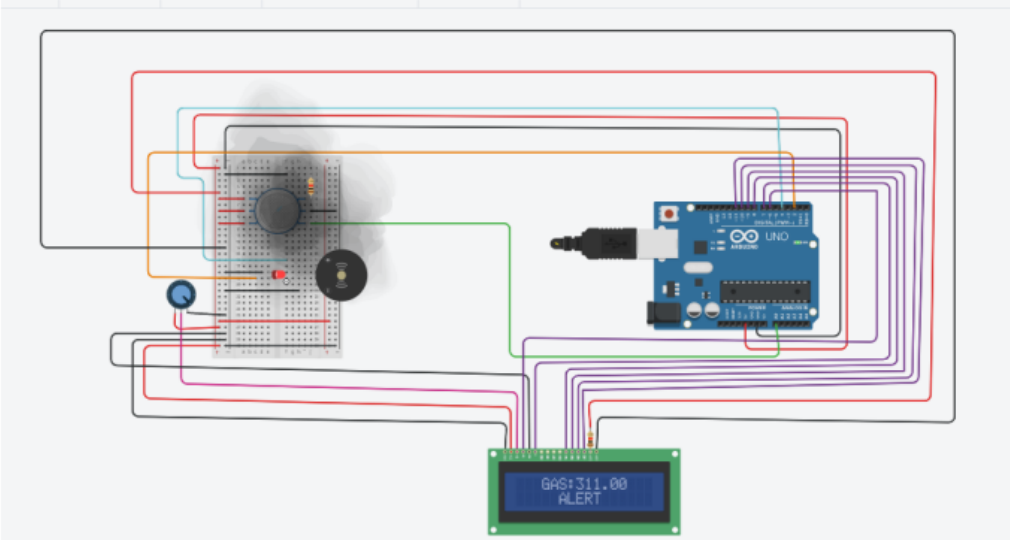
1.INITIAL SCREEN:



2.WHEN THE GAS LEVEL IS LESS THAN 250



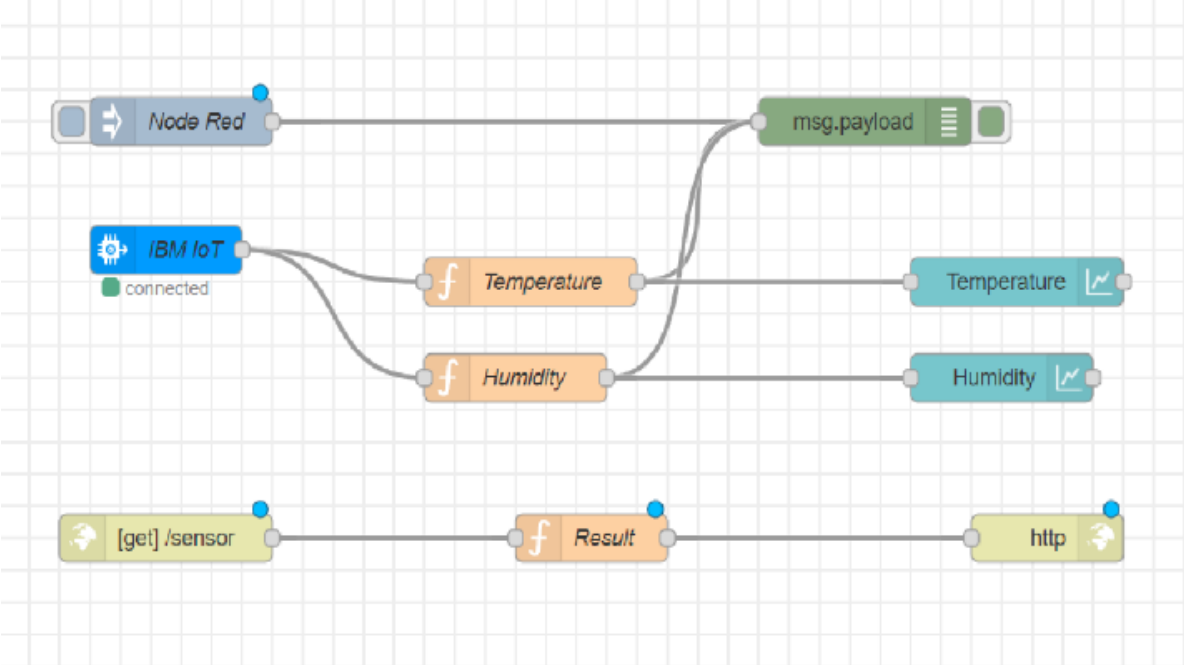
3.WHEN THE GAS LEVEL IS MORE THAN 250:



TINKER CAD LINK:
https://www.tinkercad.com/things/i142DB6zDXB-frantic-maimu/editel?tenant=circuits
DEMO LINK:
https://drive.google.com/drive/u/1/folders/1KK4blSqflC2P_tRCt0yySBf8gn-I2vsp

7.3 DATABASE SCHEMA

NODE RED SCHEMATIC

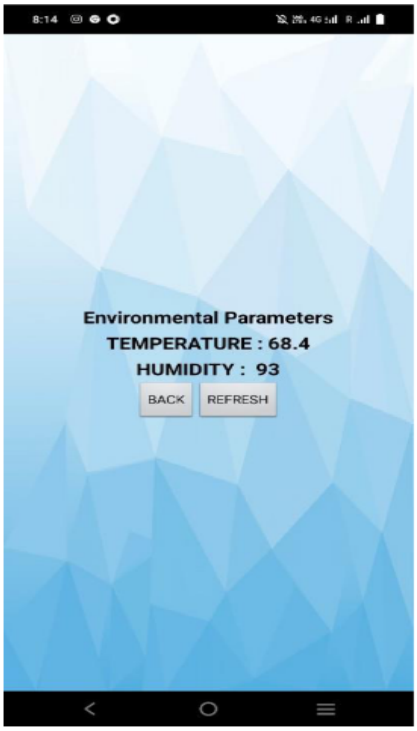


MOBILE APP

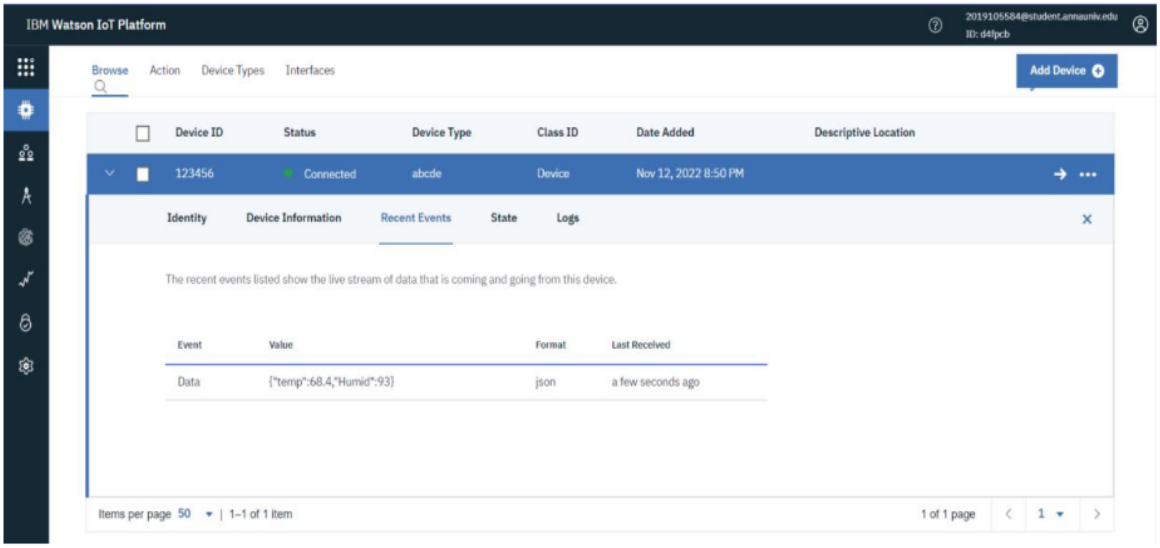
SCREEN 1

SCREEN 2

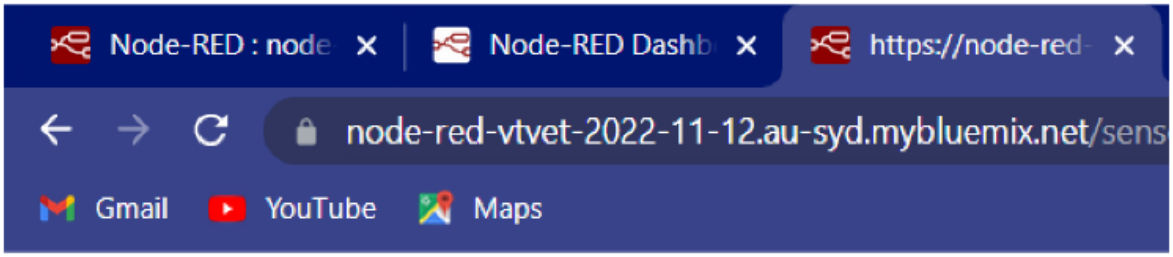
SCREEN 3



IBM WATSON PLATFORM

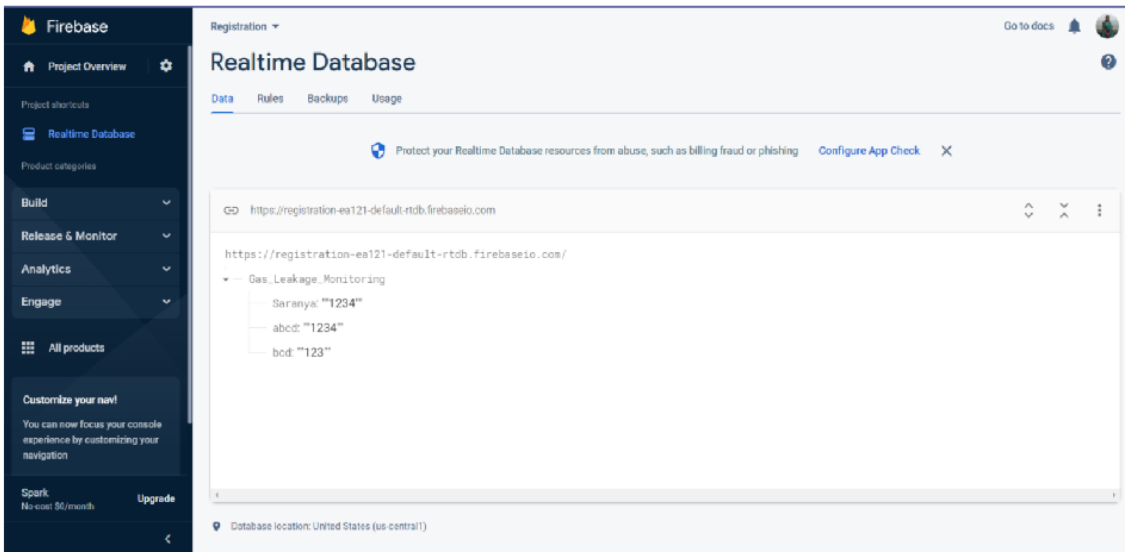


NODE RED



{"temp":68.4,"humid":93}

FIREBASE STORE DATABASE



CHAPTER 8
TESTING

8.1 TEST CASES

				Date	19-Nov-22					
				Team ID	PNT2022TMID35489					
				Project Name	Gas Leakage Monitoring and Alerting System for Industries					
				Maximum Marks	4 marks					
Test case ID	**Feature Type**	**Component**	**Test Scenario**	**Pre-Requisite**	**Steps To Execute**	**Test Data**	**Expected Result**	**Actual Result**	**Status**	**Executed By**
TC_001	Hardware	Tinkercad	Detect the gas level and alert through buzzer.		1.Open Tinkercad 2.Design the circuit. 3.Simulate the circuit.	https://www.tinkercad.com/things/i142DB6zDX9-frantic-maimu/editel	Buzzer will produce sound and LED glows on	Working as expected	Pass	M.Pon Ajiritha,M.Subarna
TC_002	Hardware	Wokwi	Detect the temperature and humidity in the environment		1.Open Wokwi 2.Design the circuit. 3.Simulate the circuit.	https://wokwi.com/projects/346582489078366803	Detect the temperature and humidity in the environment and send data to IBM Cloud.	Working as expected	Pass	M.Saranya,D.Sanghavai Bhuvaneswari
TC_003	Software	IBM Cloud	Receives the temperature and humidity from Wokwi and sends to Node-Red	IBM Watson Account	1.Open IBM Cloud 2.Go to IBM Watson IOT Platform 3.Create a device and note down organisation and device details. 4.Go to Recent events.	https://d4fpcb.internetofthings.ibmcloud.com/dashboard/devices/browse	Temperature and Humidity results are viewed in Recent events.	Working as expected	Pass	M.Subarna
TC_004	Software	NODE Red	Receives the temperature and humidity from IBM Cloud	NODE Red	1.Open Node-Red 2.Create the schematic in the workspace. 3.Give necessary details to connect to IBM Cloud 4.Deploy the schematic. 5.View the dashboard.	https://node-red-vlvet-2022-11-12.au-syd.mybluemix.net/red/#flow/3b182f9eed4c27d5 https://node-red-vlvet-2022-11-12.au-syd.mybluemix.net/sensor	Temperature and Humidity results are viewed in Information tab and Website.Temperature and Humidity variations are seen in the form of line chart in dashboard.	Working as expected	Pass	M.Pon Ajiritha
TC_005	Software	MIT App Inventor	Create a Mobile Application and receive temperature and humidity parameter from Node-Red	MIT App Inventor,MIT AI2 Companion	1.Open MIT App Inventor 2.Design your app in the Designer side(Front-End) 3.For Back-end,go to Blocks. 4.Go to Build to get thr barcode for Mobile App 5.Scan the barcode to get the App installed in your mobile.	http://ai2.appinventor.mit.edu/b/fwwf	User can register and sign in to the App and view the parameters in their place.	Working as expected	Pass	M.Saranya,M.Subarna
TC_006	Functional	Firebase	Collect the database of the user logging in to the Mobile Application	Firebase Account	1.Open Firebase 2.Create a Real time Database 3.Changes the rules to true in read and write and publish it. 4.Collect database of users logging in.	https://registration-ea121-default-rtdb.firebaseio.com/	Collection of database of users	Working as expected	Pass	D.Sanghaval Bhuvaneswari

8.2 USER ACCEPTANCE TESTING

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the Gas Leakage Monitoring and Alerting System for Industries project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

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

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	9	4	3	4	20
Duplicate	0	0	3	0	3
External	2	3	3	1	9
Fixed	11	3	4	25	43
Not Reproduced	0	0	0	0	0
Skipped	0	0	2	1	3
Won't Fix	0	0	2	2	4
Totals	22	10	17	33	82

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	48	0	0	48
Security	10	0	0	10
Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	9	0	0	9
Version Control	2	0	0	2

CHAPTER 9
RESULTS

9.1 PERFORMANCE METRICS

- **Productivity**

  The overall project utilises the given input resources effictively and produces the accurate results. Most probably the productivity is maximum using the software websites for the circuit and the cloud database.

- **Customer Satisfaction**

  As per the need of the customer i.e. alerting when the gas leakage is detected via sms and so preventing them from hazardous incident and also saves their lives. This basically provides security for the customers.

- **Actual Cost**

  The estimated cost and the actual cost required for building the project was more or less equal and thus it is also cost efficient to implement in industries as well as in the household.

- **Cost Variance**

  As the cost planned and the actual cost remains same , the cost variance is positive and our project is under budget, which is the typical measure of success.

CHAPTER 10
ADVANTAGES&DISADVANTAGES

10.1 ADVANTAGES

- Get real-time alerts about the gaseous presence in the atmosphere
- Prevent fire hazards and explosions
- Supervise gas concentration levels
- Cost-effective installation
- Monitor the amount of gases in the environment.

10.2 DISADVANTAGES

- Only one gas can be detected with an instrument.
- It only alerts and cannot prevent fire accidents.

CHAPTER 11
CONCLUSION

This proposed system not only detects the gas but also alerts the users.This system mainly focuses on the design,analysis and implementation of toxic gas leakage control.In order to prevent the existence of gas leakage in or out of our sight, this gas detector project with temperature monitoring system could prevent from gas leakage to happen. It can also send alert notification to user's mobile phone where the user or the family members of the house can be alert and be careful with the probability of the gas leakage to happen.This project has a high chance of success because this innovation has the potential to improve human life. This project allows for the collection and analysis of data related to gas leakage.

CHAPTER 12
FUTURE SCOPE

- Major future scope could be including a Automatic Shut-off device which will turn off the gas supply whenever it will detect any gas leakage.
- Another scope could be usage on a larger scale with differently programmed criticality levels for the use of large scale factories and the same interfacing and construction gas can be implemented for all types of gas sensor.
- It can also be coded to automatically open windows and to turn on exhaust fans in order to blowout the hazardous gas outside the closed area.

CHAPTER 13
APPENDIX

13.1 SOURCE CODE:

13.1.1 WOKWI:

```
#include <WiFi.h>//library for Wi-fi
#include <PubSubClient.h>//library for MQTT
#include "DHT.h"// Library for DHT 11

#define DHTPIN 15 // what pin we're connected to
#define DHTTYPE DHT22 // define type of sensor DHT 11
#define LED 2

DHT dht (DHTPIN, DHTTYPE);
void callback(char* subscribetopic, byte* payload, unsigned int
payloadLength);


//-------credentials of IBM Accounts------
#define ORG "d4fpcb"//IBM ORGANISATION ID
#define DEVICE_TYPE "abcde" //Device type mentioned in IBM Watson
IOT Platform
#define DEVICE_ID "123456" //Device ID mentioned in IBM Watson IOT
Platform
#define TOKEN "1234567890" //Token
String data3;
float h, t;
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";//
Server Name
char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and
type of event perform and format in which data to be send
char subscribetopic[] = "iot-2/cmd/command/fmt/String";//
cmd REPRESENT command type AND COMMAND IS TEST OF FORMAT STRING
char authMethod[] = "use-token-auth";// authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id
WiFiClient wifiClient; // creating the instance for wificlient
PubSubClient client(server, 1883,wifiClient);
```

```
void setup()
{
Serial.begin(115200);
dht.begin();
pinMode(LED,OUTPUT);
delay(10);
Serial.println();
wificonnect();
mqttconnect();
}
void loop()
{
h = dht.readHumidity();
t = dht.readTemperature();
Serial.print("temp:");
Serial.println(t);
Serial.print("Humid:");
Serial.println(h);
PublishData(t, h);
delay(1000);
if (!client.loop()) {
mqttconnect();
}
}
void PublishData(float temp, float humid) {
mqttconnect(); //function call for connecting to IBM
String payload = "{\"temp\":";
payload += temp;
payload += "," "\"Humid\":";
payload += humid;
payload += "}";
Serial.print("Sending payload: ");
Serial.println(payload);
if (client.publish(publishTopic, (char*) payload.c_str())) {
Serial.println("Publish ok");
}
 else {
Serial.println("Publish failed");
}
}
```

```
void mqttconnect() {
if (!client.connected()) {
Serial.print("Reconnecting client to ");
Serial.println(server);
while (!!!client.connect(clientId, authMethod, token)) {
Serial.print(".");
delay(500);
}
initManagedDevice();
Serial.println();
}
}
void wificonnect() //function defination for wificonnect
{
Serial.println();
Serial.print("Connecting to ");
WiFi.begin("Wokwi-GUEST", "", 6);//passing the wifi credentials to
establish the connection
while (WiFi.status() != WL_CONNECTED) {
delay(500);
Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
}
void initManagedDevice() {
if (client.subscribe(subscribetopic)) {
Serial.println((subscribetopic));
Serial.println("subscribe to cmd OK");
}
 else {
Serial.println("subscribe to cmd FAILED");
}
}
```

WOKWI LINK: https://wokwi.com/projects/348582489078366803

13.1.2 TINKERCAD

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(6, 7, 8, 9, 10, 11);
float gasPin = A0;
float gasLevel;
int ledPin = 2;
int buzzPin = 4;

void setup(){
  pinMode(ledPin, OUTPUT);
  pinMode(gasPin,INPUT);
  Serial.begin(9600);
  lcd.begin(16, 2);
  lcd.setCursor(0,0);
  lcd.print("    WELCOME    ");
  lcd.setCursor(0,2);
  lcd.print("GAS ALERT SYSTEM");
  delay(1000);
  lcd.clear();
}
void loop(){

  gasLevel = analogRead(gasPin);
  gasDetected(gasLevel);
  buzzer(gasLevel);

}

void gasDetected(float gasLevel){
  if(gasLevel >= 250){
    digitalWrite(buzzPin,HIGH);
    digitalWrite(ledPin,HIGH);
    lcd.setCursor(0,0);
    lcd.print("   GAS:");
    lcd.print(gasLevel);
    lcd.setCursor(0,2);
    lcd.print("     ALERT     ");
    delay(1000);
    lcd.clear();
  }
```

```
else{
  digitalWrite(ledPin,LOW);
  digitalWrite(buzzPin,LOW);
  lcd.setCursor(0,0);
  lcd.print("   GAS:");
  lcd.print(gasLevel);
  lcd.setCursor(0,2);
  lcd.print("      SAFE      ");
  delay(1000);
  lcd.clear();
 }
}
void buzzer(float gasLevel){
if(gasLevel>=250)
 {
 for(int i=0; i<=30; i=i+10)
 {
 tone(4,i);
 delay(400);
 noTone(4);
 delay(400);
 }
 }
}
```

TINKERCAD LINK: https://www.tinkercad.com/things/i142DB6zDXB-frantic-maimu/editel?tenant=circuits

GITHUB LINK:
https://github.com/IBM-EPBL/IBM-Project-3550-1658578372

DEMO LINK:
https://drive.google.com/drive/folders/1KK4blSqflC2P_tRCt0yySBf8gn-I2vsp