PROJECT REPORT

PROJECT NAME	GAS LEAKAGE MONITORING & ALERTING SYSTEM FOR INDUSTRIES
TEAM ID	PNT2022TMID09616
TEAM MEMBERS	GIRIDHARAN P AKSHAY RAM RB KARTHIKEYAN S ETHIRAJ M
BRANCH	INFORMATION TECHNOLOGY

1. INTRODUCTION

1.1 Project Overview:

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

1.2 Purpose:

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

2. LITERATURE SURVEY :

2.1 Existing Problem:

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

2.2 References:

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

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

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

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

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

6. Srinivasan,Leela,Jeyabharathi,Kirthika,Rajasree"GAS LEAKAGE DETECTION AND

CONTROL” Scientific Journal of Impact Factor(SJIF): 3.134.

7. Pal-Stefan Murvaya, IoanSileaa “A survey on gas leak detection and localization techniques”.

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

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

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

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

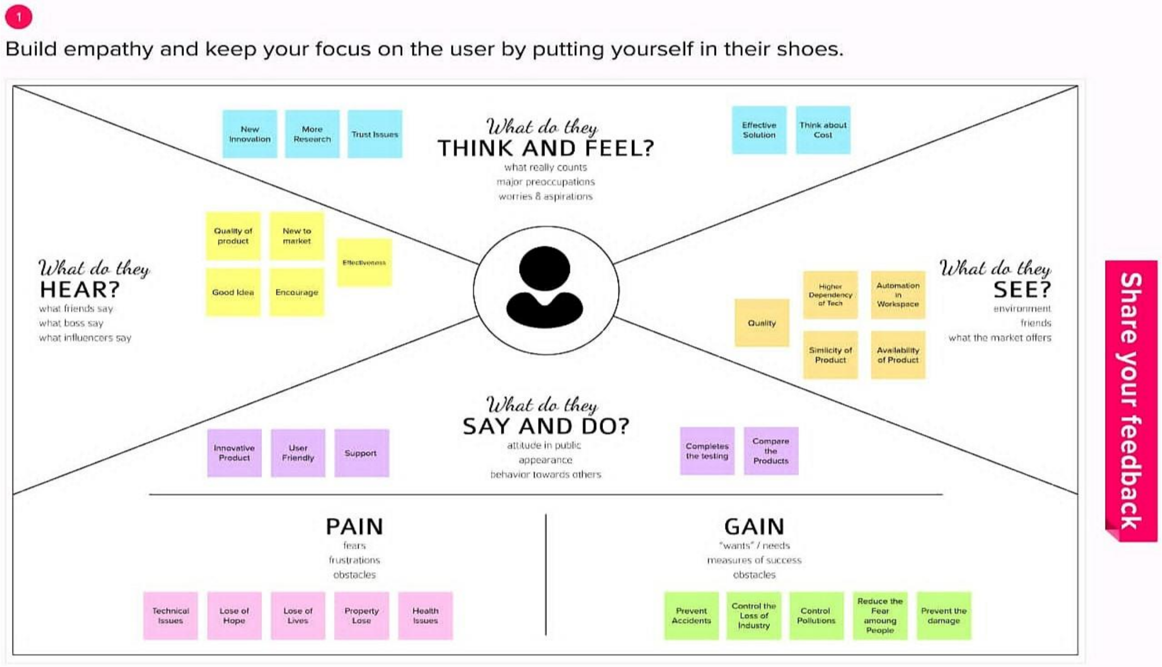
12. Falohun A.S., Oke A.O., Abolaji B.M. “Dangerous gas detection using an integrated circuit and MQ-9. In International Journal of Computer Applications. 2016; 135(7).

2.3 Problem Statement Definition:

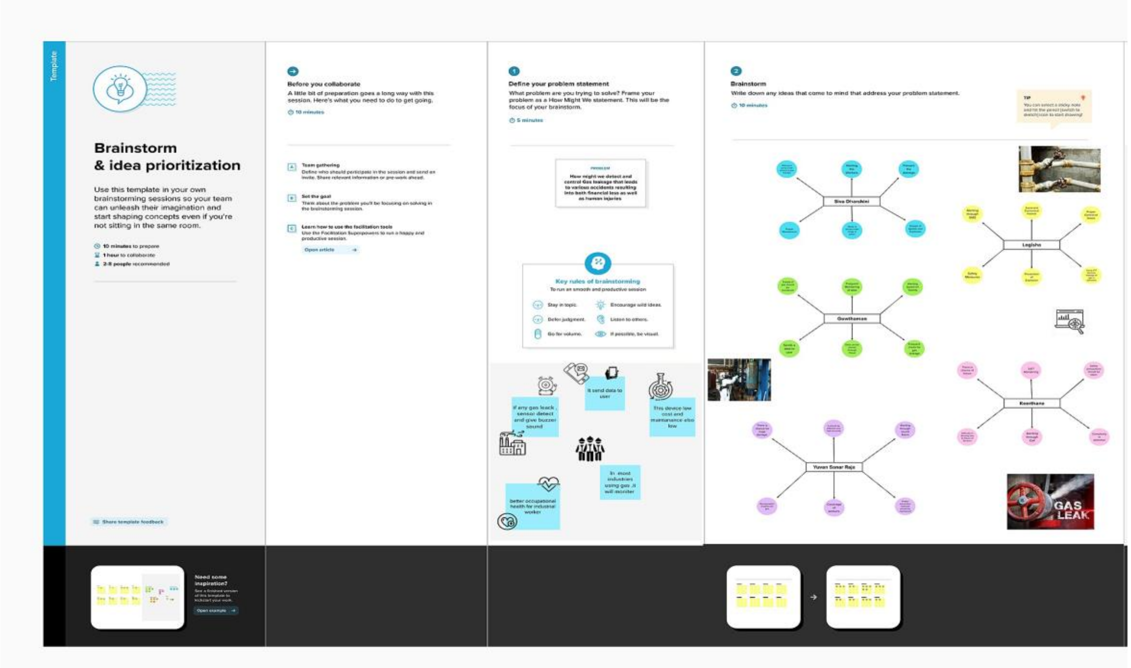
In most industries, one of the key parts of any safety plan for reducing risks to personnel and plant is the use of early-warning devices such as gas detectors. These can help to provide more time in which to take remedial or protective action. They can also be used as part of a total, integrated monitoring and safety system for an industrial plant. Rapid expansion of oil and gas industry leads to gas leakage incidents which are very serious and dangerous. Solutions need to be found out at least to minimize the effects of these incidents since gas leaks also produce a significant financial loss. The challenges are not only to design a prototype of the device that can only detect but also automatically respond to it whenever the leakage occurs.

3. IDEATION & PROPOSED SOLUTION:

3.1 Empathy Map Canvas:



3.2 Ideation & Brainstorming:



Template

Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

- **10 minutes** to prepare
- **1 hour** to collaborate
- **2-8 people** recommended

Share template feedback

Need some inspiration?
See a finished version of this template to kickstart your work.
Open example →

Before you collaborate
A little bit of preparation goes a long way with this session. Here's what you need to do to get going.
10 minutes

A **Team gathering**
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B **Set the goal**
Think about the problem you'll be focusing on solving in the brainstorming session.

C **Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.
Open article →

1 **Define your problem statement**
What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.
5 minutes

PROBLEM
How might we detect and control Gas leakage that leads to various accidents resulting into both financial loss as well as human injuries

Key rules of brainstorming
To run an smooth and productive session

- Stay in topic.
- Encourage wild ideas.
- Defer judgment.
- Listen to others.
- Go for volume.
- If possible, be visual.

If any gas leack , sensor detect and give buzzer sound

It send data to user

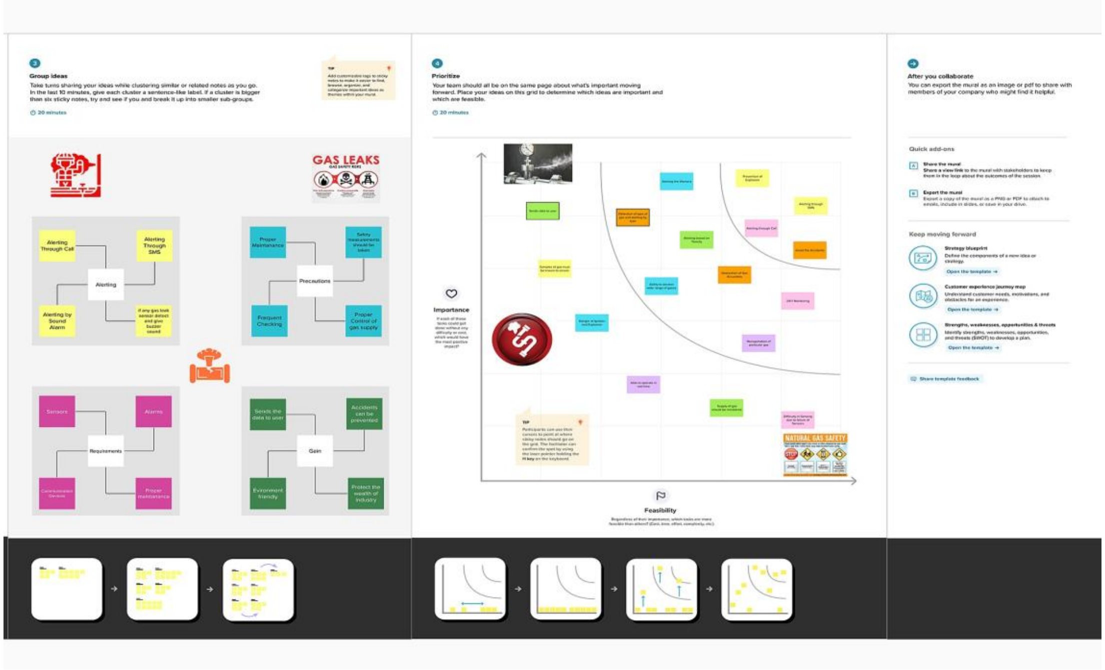
This device low cost and maintanance also low

In most industries using gas ,it will monitor

better occupational health for industrial worker

2 **Brainstorm**
Write down any ideas that come to mind that address your problem statement.
10 minutes

TIP
You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!



3 **Group ideas**
Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.
20 minutes

TIP
Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

4 **Prioritize**
Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.
20 minutes

Importance
If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?
Feasibility
Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)
NATURAL GAS SAFETY
STOP

TIP
Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the **H key** on the keyboard.

After you collaborate
You can export the mural as an image or pdf to share with members of your company who might find it helpful.

Quick add-ons

A **Share the mural**
Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.

B **Export the mural**
Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.

Keep moving forward

Strategy blueprint
Define the components of a new idea or strategy.
Open the template →

Customer experience journey map
Understand customer needs, motivations, and obstacles for an experience.
Open the template →

Strengths, weaknesses, opportunities & threats
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.
Open the template →

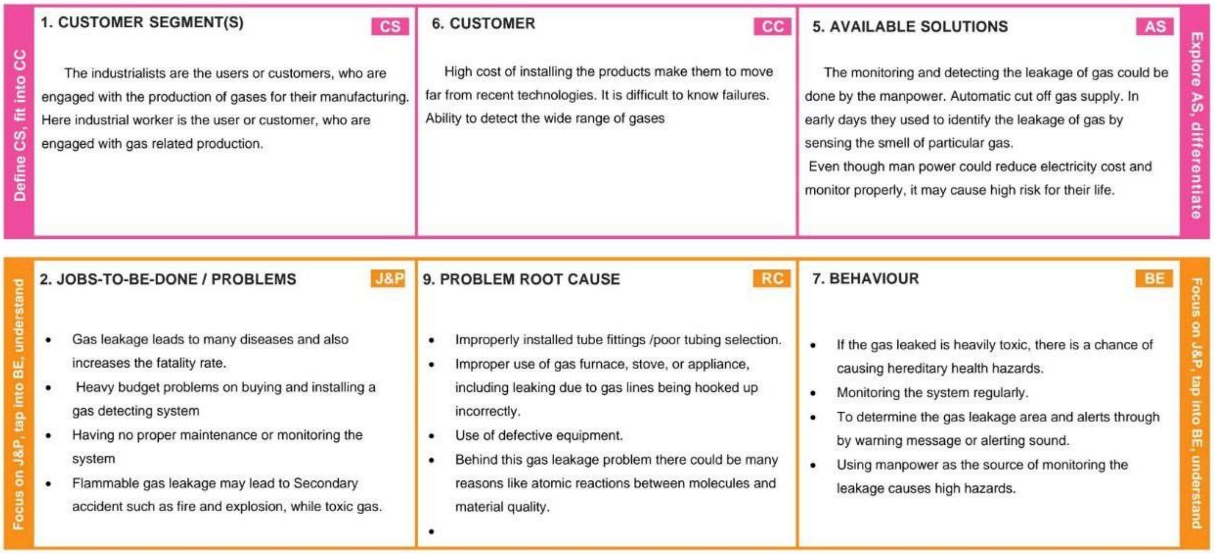
Share template feedback

3.3 Proposed Solution:

S.No	Parameter	Description
1.	Problem Statement (Problem to be solved)	Gas leakage leads to various accidents resulting in loss of human lives and industry properties. Sometimes, the gas leakage cannot be detected by human that has a low sense of smell. Thus, this system will help to detect the presence of gas leakage and alert the users.
2.	Idea / Solution description	It detects the gas leakage by using various sensors. If the gas leakage level is above the threshold level, it sends the alert message through SMS to the user by using GSM module and buzzer the alarm.
3.	Novelty / Uniqueness	We use location tagging and alert service so that the admin and fire department team will be notified the exact location. The system provides constant monitoring and detection of gas leakage along with storage of data in database for predictions and analysis.
4.	Social Impact / Customer Satisfaction	By implementing real-time gas leak detection, industries can monitor their environmental performance, ensure better occupational health. Also, early detection of gas leaks can trigger concerned engineers to curtail the spread and keep a safe environment for better health and safety.

5.	Business Model (Revenue Model)	The product can be made compact, cost efficient and easily installable so that all the industries from small scale to large scale can able to buy the product .
6.	Scalability of the Solution	The system is very simple and easy to maintain and cost efficient. It has the capability to works for a period of time without any damage in the system components.

3.4 Problem Solution fit:



Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

The industrialists are the users or customers, who are engaged with the production of gases for their manufacturing. Here industrial worker is the user or customer, who are engaged with gas related production.

6. CUSTOMER CC

High cost of installing the products make them to move far from recent technologies. It is difficult to know failures. Ability to detect the wide range of gases

5. AVAILABLE SOLUTIONS AS

The monitoring and detecting the leakage of gas could be done by the manpower. Automatic cut off gas supply. In early days they used to identify the leakage of gas by sensing the smell of particular gas.

Even though man power could reduce electricity cost and monitor properly, it may cause high risk for their life.

Explore AS, differentiate

Focus on J&P, tap into BE, understand

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

- Gas leakage leads to many diseases and also increases the fatality rate.
- Heavy budget problems on buying and installing a gas detecting system
- Having no proper maintenance or monitoring the system
- Flammable gas leakage may lead to Secondary accident such as fire and explosion, while toxic gas.

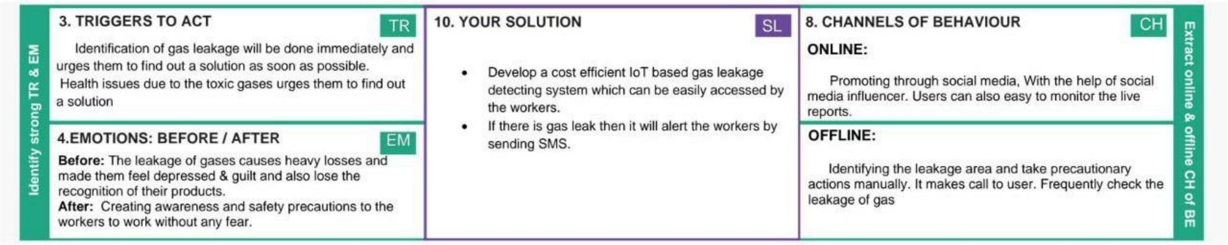
9. PROBLEM ROOT CAUSE RC

- Improperly installed tube fittings /poor tubing selection.
- Improper use of gas furnace, stove, or appliance, including leaking due to gas lines being hooked up incorrectly.
- Use of defective equipment.
- Behind this gas leakage problem there could be many reasons like atomic reactions between molecules and material quality.
-

7. BEHAVIOUR BE

- If the gas leaked is heavily toxic, there is a chance of causing hereditary health hazards.
- Monitoring the system regularly.
- To determine the gas leakage area and alerts through by warning message or alerting sound.
- Using manpower as the source of monitoring the leakage causes high hazards.

Focus on J&P, tap into BE, understand



Identify strong TR & EM

3. TRIGGERS TO ACT TR

Identification of gas leakage will be done immediately and urges them to find out a solution as soon as possible.
Health issues due to the toxic gases urges them to find out a solution

4.EMOTIONS: BEFORE / AFTER EM

Before: The leakage of gases causes heavy losses and made them feel depressed & guilt and also lose the recognition of their products.
After: Creating awareness and safety precautions to the workers to work without any fear.

10. YOUR SOLUTION SL

- Develop a cost efficient IoT based gas leakage detecting system which can be easily accessed by the workers.
- If there is gas leak then it will alert the workers by sending SMS.

8. CHANNELS OF BEHAVIOUR CH

ONLINE:

Promoting through social media, With the help of social media influencer. Users can also easy to monitor the live reports.

OFFLINE:

Identifying the leakage area and take precautionary actions manually. It makes call to user. Frequently check the leakage of gas

Extract online & offline CH of BE

4. REQUIREMENT ANALYSIS:

4.1 Functional requirement:

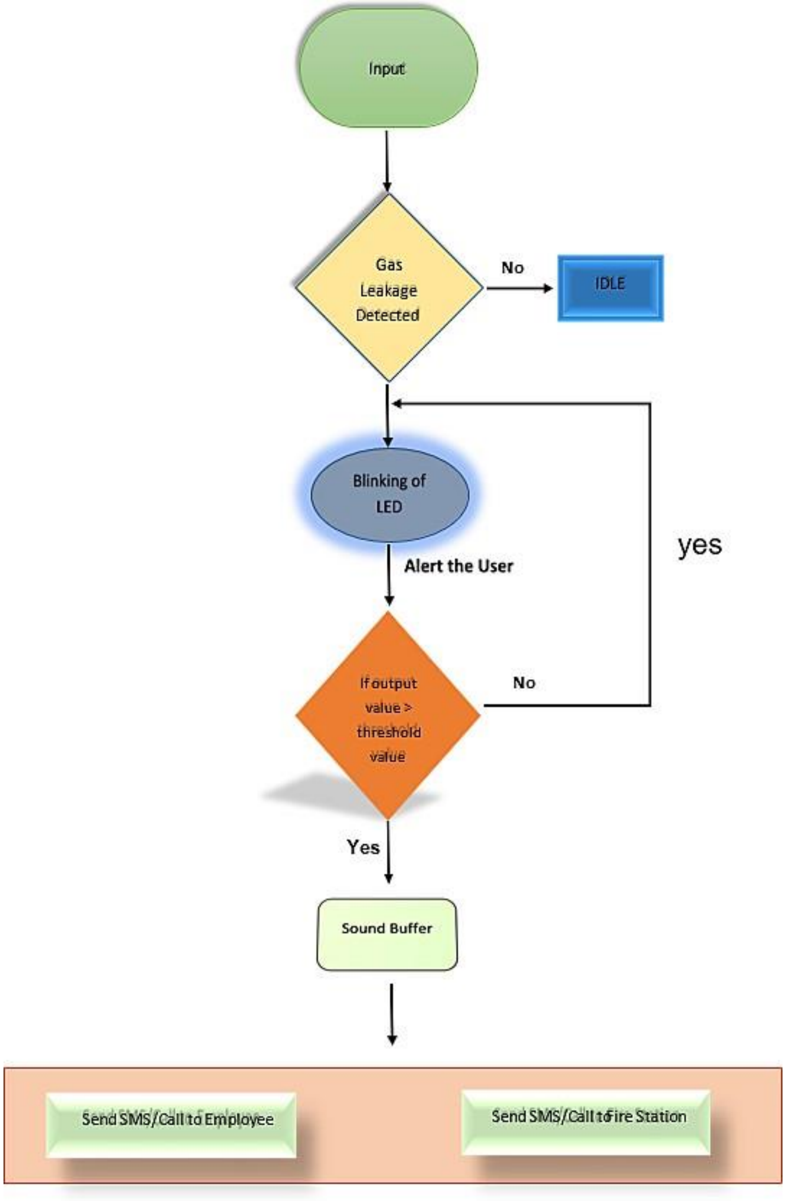
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through FormRegistration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	GPS Access	GPS access to know the location
FR-4	Business Requirements	The device is intended for the use of industries or factories and also for cylinder storage areas. It detects the leakage of gas and sends the data over to a site and preventive measures can be taken to avoid the loss of properties.
FR-5	User Requirements	The Gas leakage detecting system with upgrading technologies which identifies the leakage of gas and also ensures the workers safety.

4.2 Non-Functional requirements:

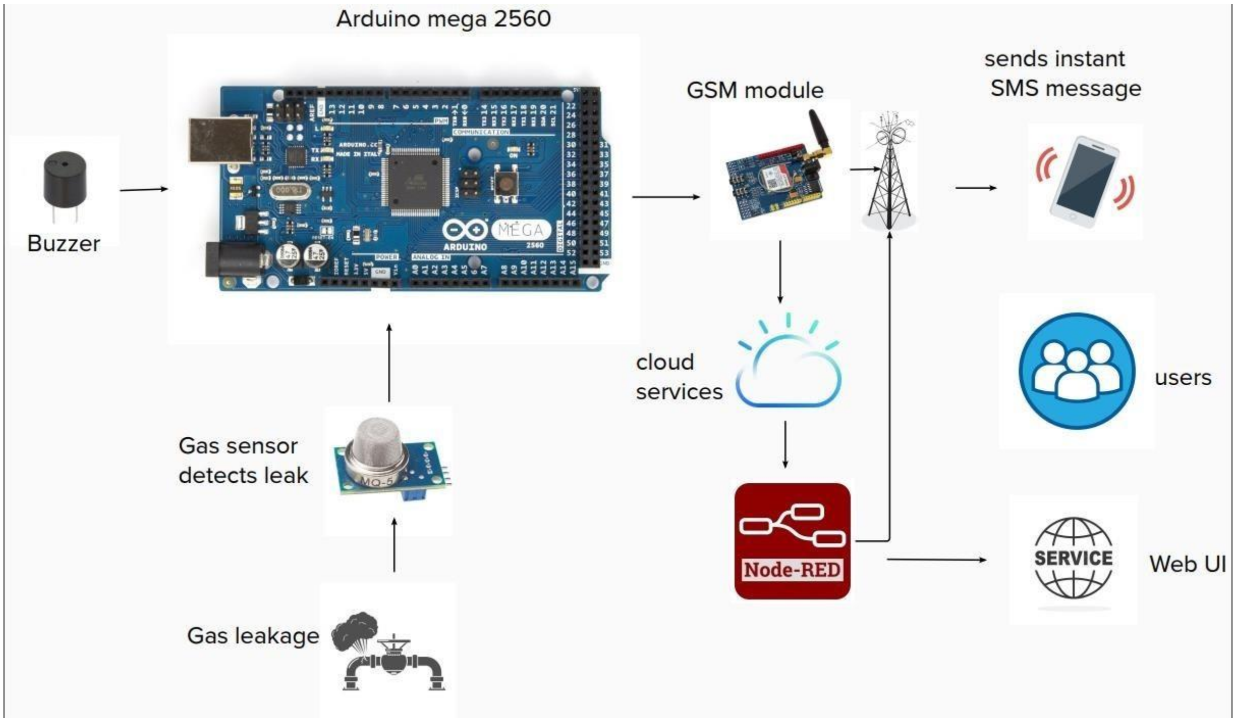
FR No.	Non-Functional Requirement	Description
NFR-1	Usability	The sensors used to detect the gas leakage which helps to prevent the high risk of gas explosion and also can prevent the causalities within and outside the covering area of the industries.
NFR-2	Security	The device is intended for the use of industries or factories, where there is a use of explosive gas is a source of risk. This device will help and secures from the causes.
NFR-3	Reliability	Gas leakage detecting system detects the gas leakage at industries or factories which detects the small amount of gas leakage as soon and sends the alerting SMS to users.
NFR-4	Performance	The Gas leakage detecting system is a device with an alarm setting. Whenever there is a gas leak ,which is greater than the threshold level, the in-build sensor detects and alerts the user within a minute much before it can cause any accidents.
NFR-5	Availability	The gas leakage detecting system is readily available in the market which is extremely expensive, but here we are providing a low-cost circuit for gas leakage detecting system and also it is user friendly
NFR-6	Scalability	The system is very simple and easy to maintain with cost efficient. A backup power supply will be included in the design to prevent from the power failure conditions. It has the capability to works for a period of time without any damage in the system components.

5. PROJECT DESIGN:

5.1 Data Flow Diagrams:


Input
Gas Leakage Detected
No
IDLE
Blinking of LED
Alert the User
yes
If output value > threshold value
No
Yes
Sound Buffer
Send SMS/Call to Employee
Send SMS/Call to Fire Station

5.2 Solution & Technical Architecture:



5.3 User Stories:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can create an account in the application provided.	I can access my account/ dashboard	High	Sprint-1
		USN-2	As a user, I registered using my Gmail.	I can receive confirmation email.	High	Sprint-1
		USN-3	As a user, I can successfully install the app.	I can register and access the dashboard.	Low	Sprint-2
	Login	USN-4	As a user, I can login using my Gmail and password easily.	The login process was easy and simple to access the dashboard.	High	Sprint-1
Customer (Web user)	Registration	WUSN-1	As a web user I can login to web dashboard just like a website.	I can register and access the dashboard.	High	Sprint-2
	Dashboard	WUSN-2	As a user I can view the alert/warning SMS in the web application.	I can login to the website using my login credentials	High	Sprint-2

Customer Care Executive		CCE-1	A customer care executive will always be available for the interaction with the customer to clarify the queries.	An executive will clarify the doubts and note down the complaints of the application if any.	High	Sprint-2
Administrator		ADMIN-1	I as an Admin can access and view the data or information provided by the application &can also check, analyse the threshold value of the gas.	The details of the gas leakage level of the gas are provided to the users through SMS when an alerting sound is received.	High	Sprint-1

6. PROJECT PLANNING & SCHEDULING:

6.1 Sprint Planning & Estimation:

- SPRINT PLAN
- ANALYZE THE PROBLEM
- PREPARE an ABSTRACT, PROBLEM STATEMENT
- LIST A REQUIRED OBJECT NEEDED
- CREATE A PROGRAM CODE AND RUN IT
- MAKE A PROTOTYPE TO IMPLEMENT
- TEST WITH THE CREATED CODE AND CHECK THE DESIGNEDPROTOTYPE

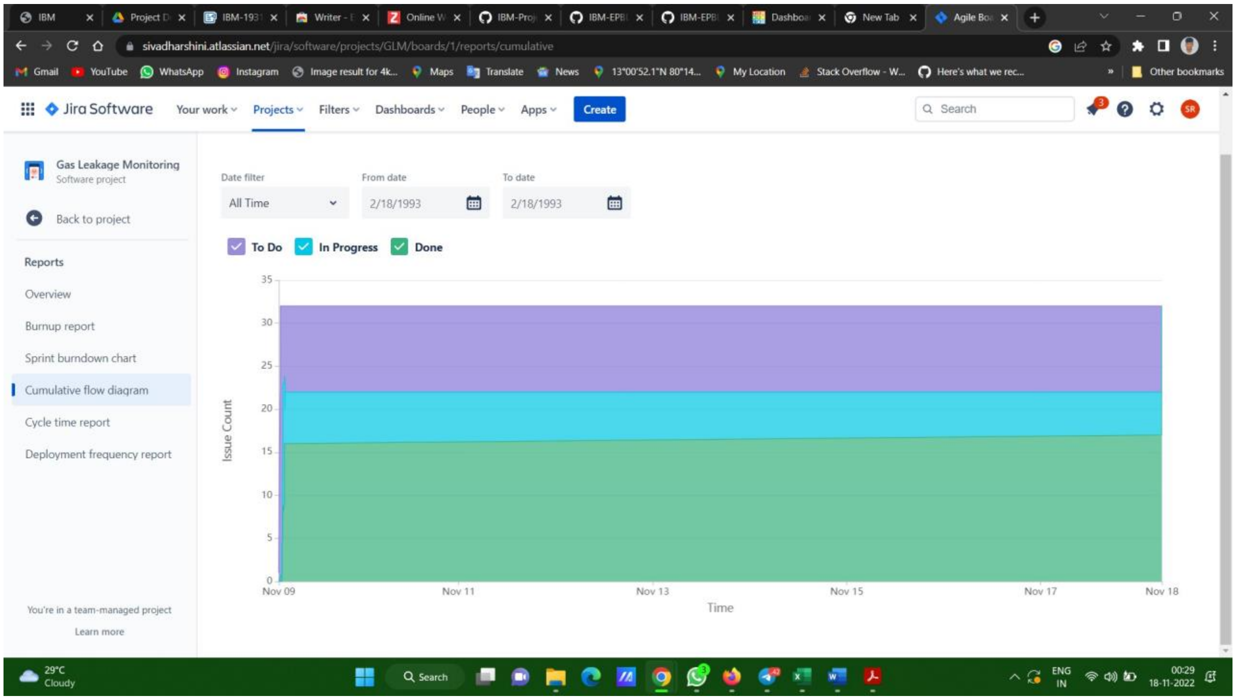
6.2 Sprint Delivery Schedule:

Sprint	Functional Requirement (Epic)	User Story	User Story / Task	Story Point	Priority
Sprint-1	Create	US-1	Create the IBM Cloud services which are being used in this project.	5	High
Sprint-1	Configure	US-2	Configure the IBM Cloud services which are being used in completing this project.	1	Medium
Sprint-1	Create	US-3	IBM Watson IoT platform acts as the mediator to connect the web application to IoT devices, so create the IBM Watson IoT platform.	1	Medium
Sprint-1	Configure	US-4	Configure the IBM Watson IoT which are being used to display the output.	13	High
Sprint-2	Create	US-1	In order to connect the IoT device to the IBM cloud, create a device in the IBM Watson IoT platform and get the device credentials.	13	High

Sprint-2	Configure	US-2	Configure a device in the IBM Watson IoT platform and get the device credentials.	3	Medium
Sprint-2	Create	US-3	Create a Node-RED service.	3	High
Sprint-2	Configure	US-4	Configure the connection security and create API keys that are used in the Node- RED service for accessing the IBM IoT Platform.	1	Medium
Sprint-3	Develop	US-1	Develop a python script to publish random sensor data such as temperature, Flame level and Gas level to the IBM IoTplatform	1 3	High
Sprint-3	Configure	US-2	After developing python code and commands just run the code	1	Medium
Sprint-3	Print	US-3	Print the statements which represent the control of the devices.	1	Low
Sprint-3	Publish	US-4	Publish Data to The IBM Cloud	5	High

Sprint-4	Create	US-1	Create Web UI in Node- Red	5	High
Sprint-4	Configure	US-2	Configure the Node-RED flow to receive data from the IBMIoT platform	5	High
Sprint-4	Configure	US-3	Use cloudant DB nodes to store the received sensor data in the cloudant DB	5	High
Sprint-4	Publish	US-4	Publish the received data inweb-application	5	High

6.3 Report from JIRA:



sivadharshini.atlassian.net/jira/software/projects/GLM/boards/1/reports/cumulative

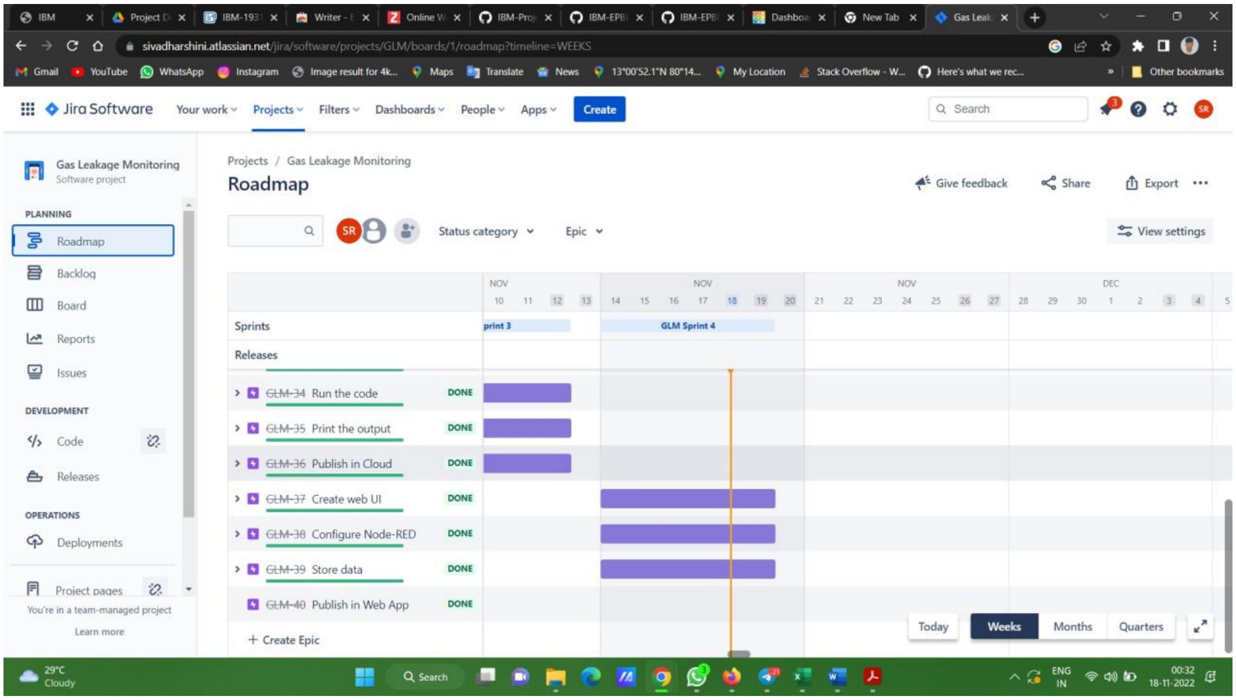
Gas Leakage Monitoring
Software project

Date filter: All Time

From date: 2/18/1993

To date: 2/18/1993

☑ To Do ☑ In Progress ☑ Done

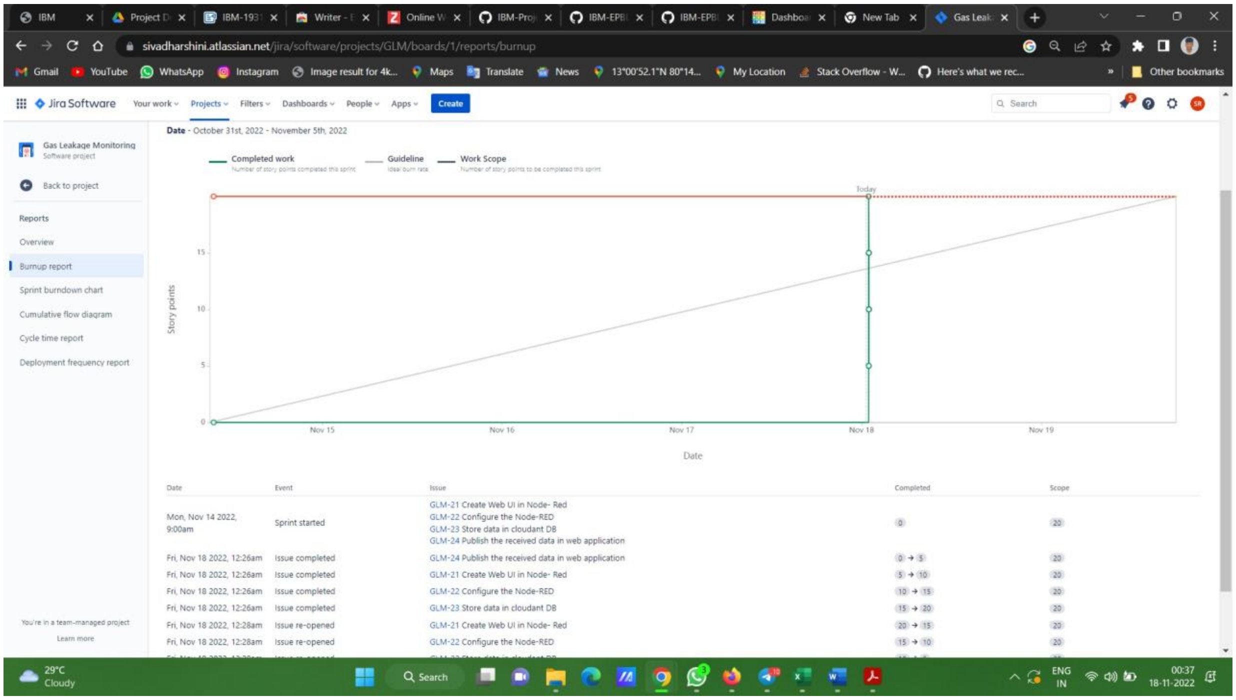


Projects / Gas Leakage Monitoring

Roadmap

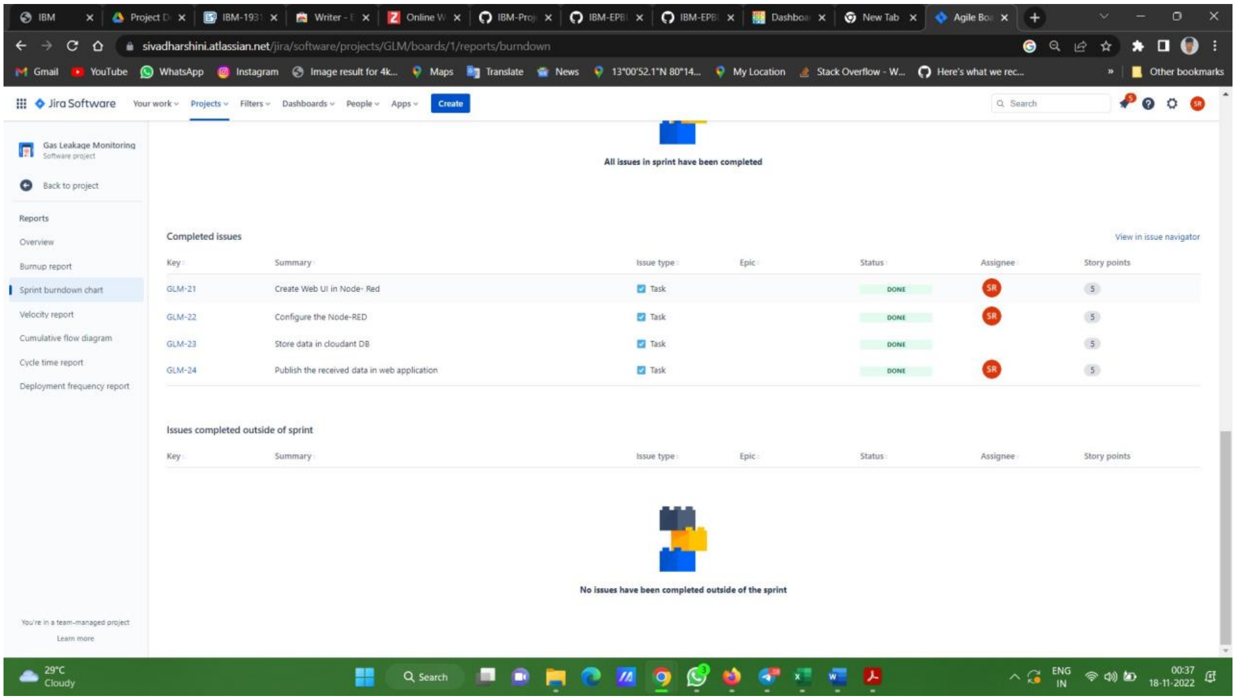
	NOV 10 – 13	NOV 14 – 20	NOV 21 – 27	DEC 28 – 5
Sprints	print 3	GLM Sprint 4		
Releases				
~~GLM-34~~ Run the code DONE				
~~GLM-35~~ Print the output DONE				
~~GLM-36~~ Publish in Cloud DONE				
~~GLM-37~~ Create web UI DONE				
~~GLM-38~~ Configure Node-RED DONE				
~~GLM-39~~ Store data DONE				
~~GLM-40~~ Publish in Web App DONE				

+ Create Epic



Date - October 31st, 2022 - November 5th, 2022

Date	Event	Issue	Completed	Scope
Mon, Nov 14 2022, 9:00am	Sprint started	GLM-21 Create Web UI in Node- Red GLM-22 Configure the Node-RED GLM-23 Store data in cloudant DB GLM-24 Publish the received data in web application	0	20
Fri, Nov 18 2022, 12:26am	Issue completed	GLM-24 Publish the received data in web application	0 → 5	20
Fri, Nov 18 2022, 12:26am	Issue completed	GLM-21 Create Web UI in Node- Red	5 → 10	20
Fri, Nov 18 2022, 12:26am	Issue completed	GLM-22 Configure the Node-RED	10 → 15	20
Fri, Nov 18 2022, 12:26am	Issue completed	GLM-23 Store data in cloudant DB	15 → 20	20
Fri, Nov 18 2022, 12:28am	Issue re-opened	GLM-21 Create Web UI in Node- Red	20 → 15	20
Fri, Nov 18 2022, 12:28am	Issue re-opened	GLM-22 Configure the Node-RED	15 → 10	20



All issues in sprint have been completed

Completed issues

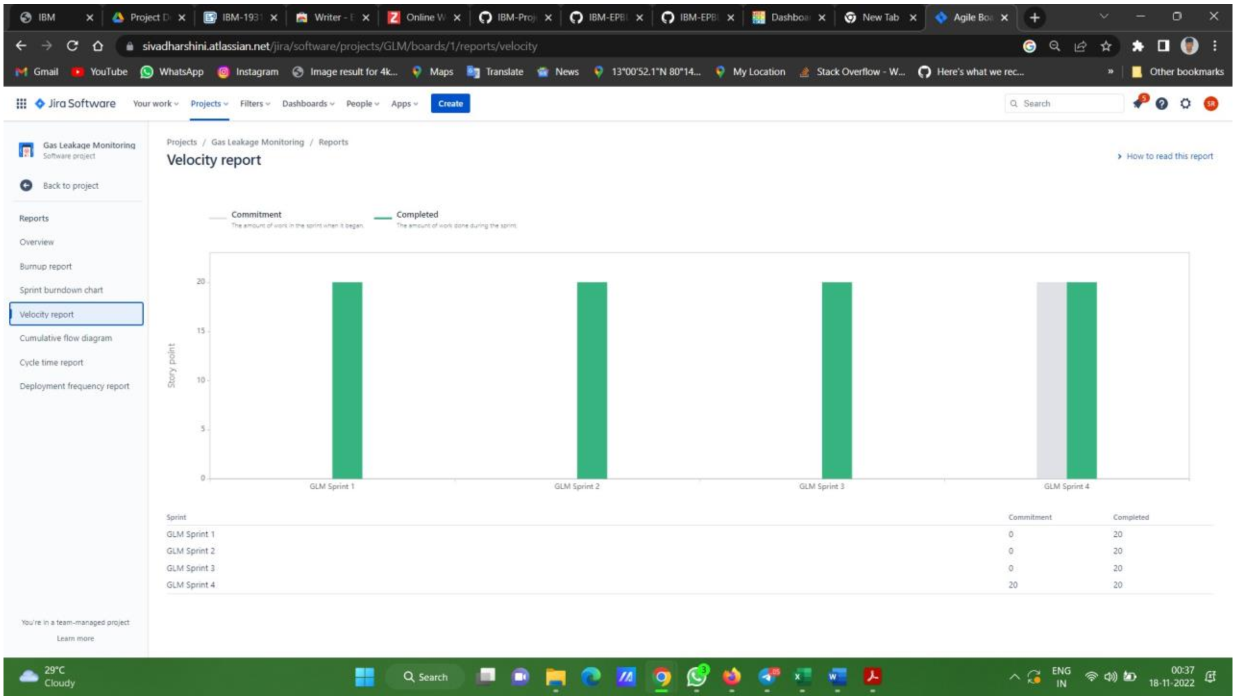
View in issue navigator

Key	Summary	Issue type	Epic	Status	Assignee	Story points
GLM-21	Create Web UI in Node- Red	☑ Task		DONE	SR	5
GLM-22	Configure the Node-RED	☑ Task		DONE	SR	5
GLM-23	Store data in cloudant DB	☑ Task		DONE		5
GLM-24	Publish the received data in web application	☑ Task		DONE	SR	5

Issues completed outside of sprint

Key	Summary	Issue type	Epic	Status	Assignee	Story points

No issues have been completed outside of the sprint



Projects / Gas Leakage Monitoring / Reports

Velocity report

How to read this report

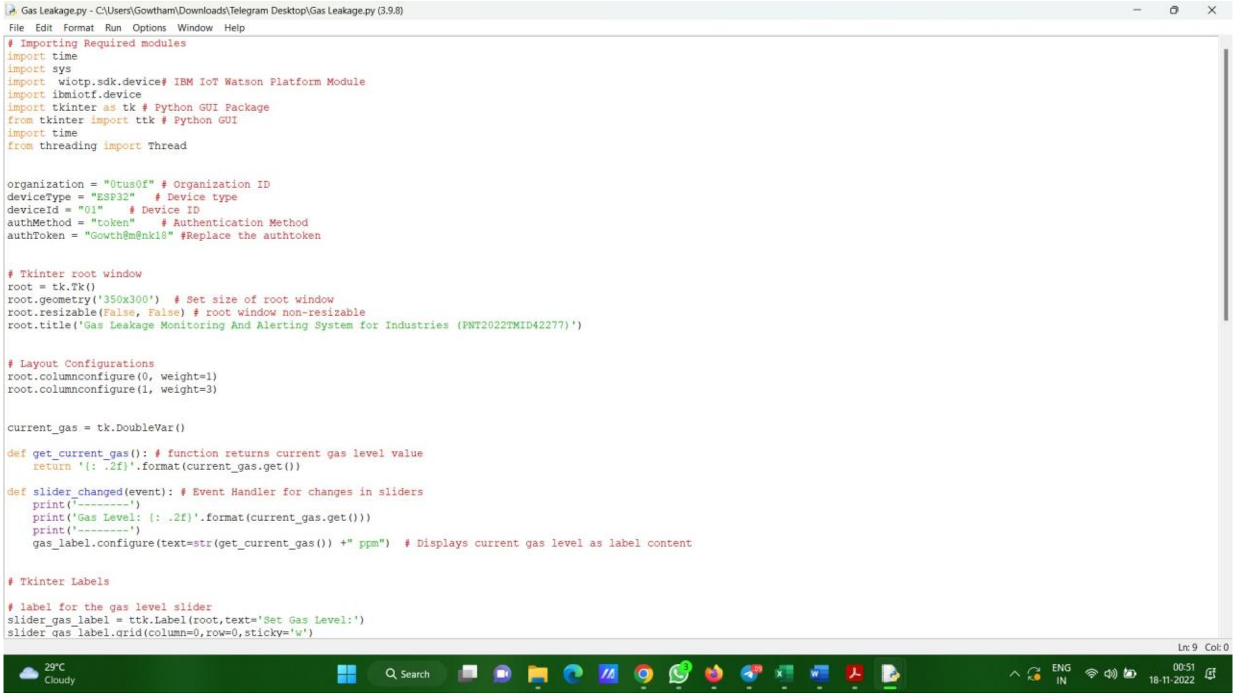
Sprint	Commitment	Completed
GLM Sprint 1	0	20
GLM Sprint 2	0	20
GLM Sprint 3	0	20
GLM Sprint 4	20	20

7. CODING & SOLUTIONING:

```
# Importing Required modules
import time
import sys
import wiotp.sdk.device# IBM IoT Watson Platform Module
import ibmiotf.device
import tkinter as tk # Python GUI Package
from tkinter import ttk # Python GUI
import time
from threading import Thread


organization = "0tus0f" # Organization ID
deviceType = "ESP32" # Device type
deviceId = "01"     # Device ID
authMethod = "token"     # Authentication Method
authToken  =  "Gowth@m@nk18"  #Replace  the  authtoken


# Tkinter root window
root = tk.Tk()
root.geometry('350x300') # Set size of root window
root.resizable(False, False) # root window non-resizable
root.title('Gas Leakage Monitoring And Alerting System for Industries
(PNT2022TMID42277)')


# Layout Configurations
root.columnconfigure(0, weight=1)
root.columnconfigure(1, weight=3)


current_gas = tk.DoubleVar()

def get_current_gas(): # function returns current gas level value
   return '{: .2f}'.format(current_gas.get())

def slider_changed(event): # Event Handler for changes in sliders
   print('------ ')
   print('Gas Level: {: .2f}'.format(current_gas.get()))
   print('------ ')
   gas_label.configure(text=str(get_current_gas()) +" ppm")  # Displays current gas level
as label content
```

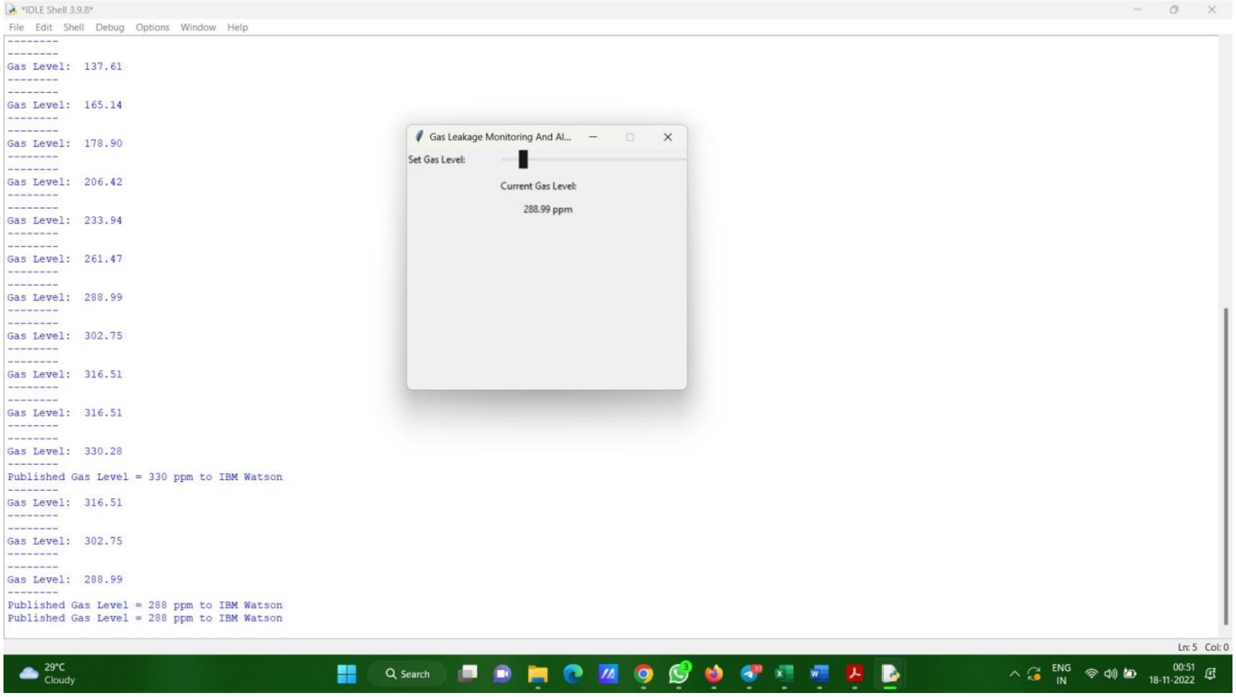
```
# Tkinter Labels

# label for the gas level slider
slider_gas_label = ttk.Label(root,text='Set Gas Level:')
slider_gas_label.grid(column=0,row=0,sticky='w')


#  Gas Level slider
slider_gas = ttk.Scale(root,from_=0,to=3000,orient='horizontal',
command=slider_changed,variable=current_gas) slider_gas.grid(column=1,row=0,sticky='we')


# current gas level label
current_gas_label = ttk.Label(root,text='Current Gas Level:')
current_gas_label.grid(row=1,columnspan=2,sticky='n',ipadx=10,ipady=10)


# Gas level label (value gets displayed here)
gas_label = ttk.Label(root,text=str(get_current_gas()) +" ppm")
gas_label.grid(row=2,columnspan=2,sticky='n')


def publisher_thread():
    thread = Thread(target=publish_data)
    thread.start()

def publish_data():
    # Exception Handling
    try:
        deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-
method": authMethod,
                     "auth-token": authToken}
        deviceCli = ibmiotf.device.Client(deviceOptions)

    except Exception as e:
        print("Caught exception connecting device: %s" % str(e))
        sys.exit()

    deviceCli.connect()  # Connect to IBM Watson IoT Platform

    while True:
        gas_level = int(current_gas.get())
        data = {'gas_level' : gas_level}

        def myOnPublishCallback():
            print("Published Gas Level = %s ppm" % gas_level, "to IBM Watson")

        success = deviceCli.publishEvent("event", "json", data, qos=0,
on_publish=myOnPublishCallback)
        if not success:
            print("Not connected to IoTF")
        time.sleep(1)

publisher_thread()

root.mainloop() # startup Tkinter GUI

# Disconnect the device and application from the cloud
deviceCli.disconnect()
```

CODE:


Gas Leakage.py - C:\Users\Gowtham\Downloads\Telegram Desktop\Gas Leakage.py (3.9.8)
File Edit Format Run Options Window Help
Ln: 9 Col: 0
29°C
Cloudy
Search
ENG
IN
00:51
18-11-2022

```
# Importing Required modules
import time
import sys
import  wiotp.sdk.device# IBM IoT Watson Platform Module
import ibmiotf.device
import tkinter as tk # Python GUI Package
from tkinter import ttk # Python GUI
import time
from threading import Thread


organization = "0tus0f" # Organization ID
deviceType = "ESP32"   # Device type
deviceId = "01"     # Device ID
authMethod = "token"    # Authentication Method
authToken = "Gowth@m@nk18" #Replace the authtoken


# Tkinter root window
root = tk.Tk()
root.geometry('350x300')  # Set size of root window
root.resizable(False, False) # root window non-resizable
root.title('Gas Leakage Monitoring And Alerting System for Industries (PNT2022TMID42277)')


# Layout Configurations
root.columnconfigure(0, weight=1)
root.columnconfigure(1, weight=3)


current_gas = tk.DoubleVar()

def get_current_gas(): # function returns current gas level value
    return '{: .2f}'.format(current_gas.get())

def slider_changed(event): # Event Handler for changes in sliders
    print('--------')
    print('Gas Level: {: .2f}'.format(current_gas.get()))
    print('--------')
    gas_label.configure(text=str(get_current_gas()) +" ppm")  # Displays current gas level as label content


# Tkinter Labels

# label for the gas level slider
slider_gas_label = ttk.Label(root,text='Set Gas Level:')
slider_gas_label.grid(column=0,row=0,sticky='w')
```

OUTPUT:

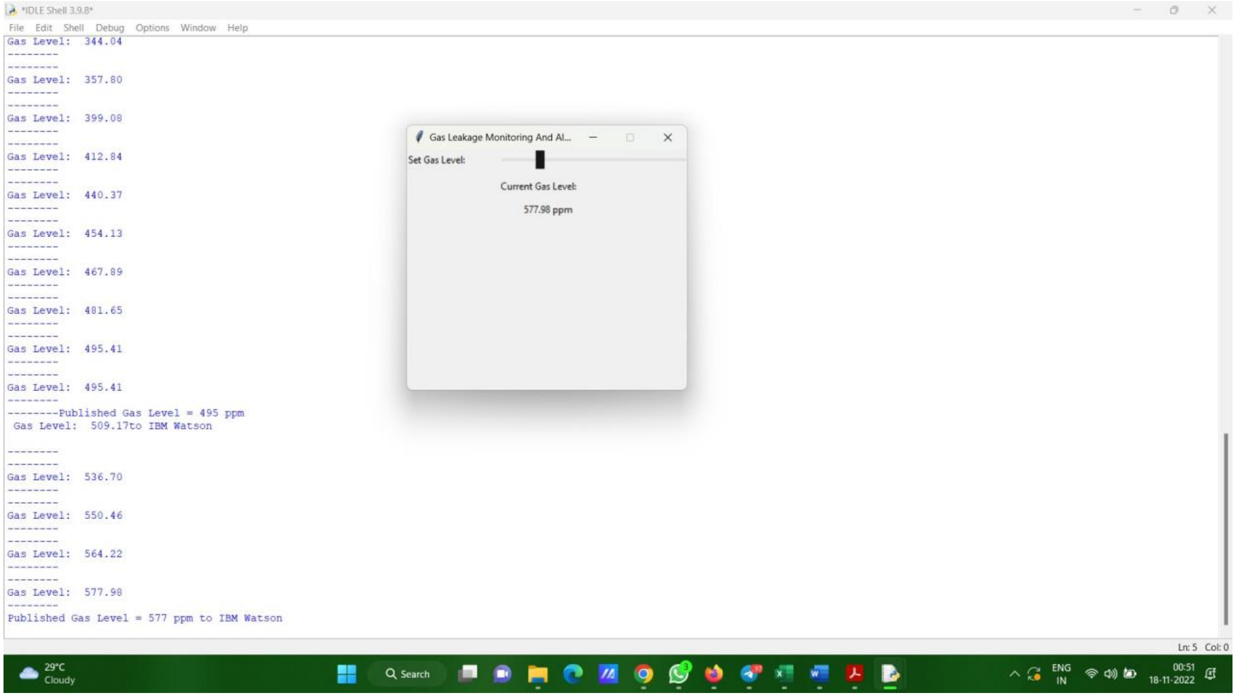

IDLE Shell 3.9.8
File Edit Shell Debug Options Window Help
Gas Leakage Monitoring And Al...
Ln: 5 Col: 0
29°C
Cloudy
Search
ENG
IN
00:51
18-11-2022

```
--------
--------
Gas Level:  137.61
--------
--------
Gas Level:  165.14
--------
--------
Gas Level:  178.90
--------
--------
Gas Level:  206.42
--------
--------
Gas Level:  233.94
--------
--------
Gas Level:  261.47
--------
--------
Gas Level:  288.99
--------
--------
Gas Level:  302.75
--------
--------
Gas Level:  316.51
--------
--------
Gas Level:  316.51
--------
--------
Gas Level:  330.28
--------
Published Gas Level = 330 ppm to IBM Watson
--------
Gas Level:  316.51
--------
--------
Gas Level:  302.75
--------
--------
Gas Level:  288.99
--------
Published Gas Level = 288 ppm to IBM Watson
Published Gas Level = 288 ppm to IBM Watson
```

Set Gas Level:

Current Gas Level:

288.99 ppm


IDLE Shell 3.9.8
File Edit Shell Debug Options Window Help
Gas Leakage Monitoring And Al...
Ln: 5 Col: 0
29°C
Cloudy
Search
ENG
IN
00:51
18-11-2022

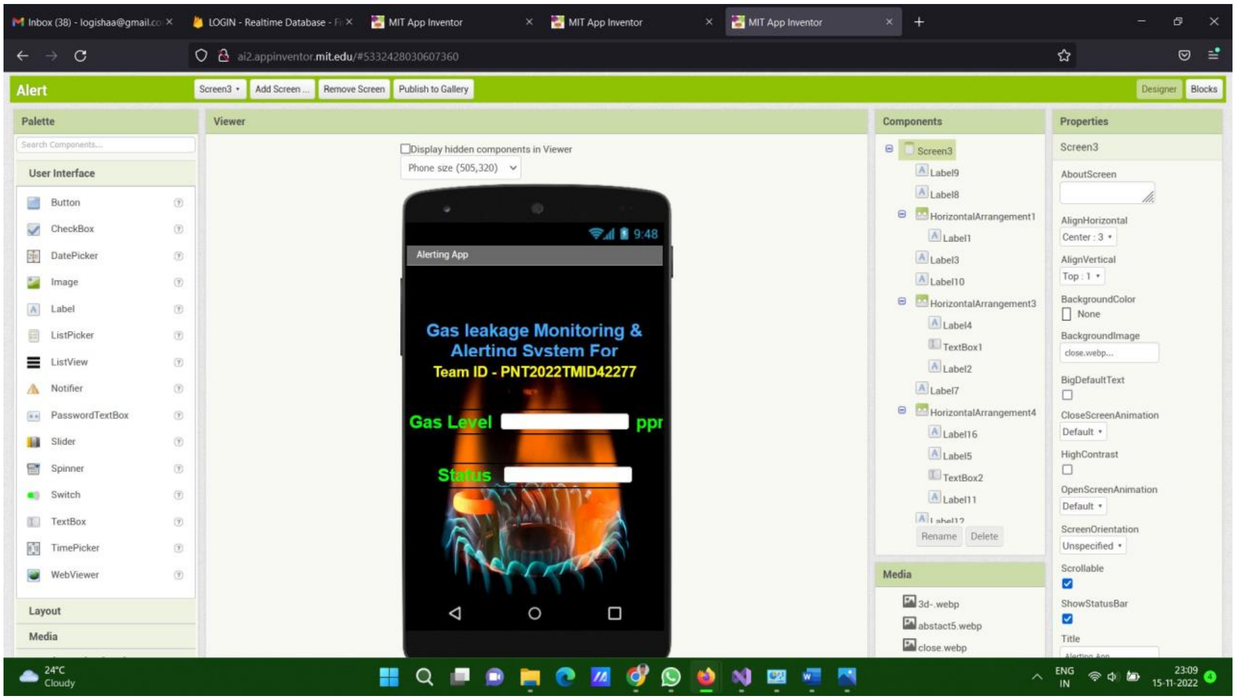
```
Gas Level:  344.04
--------
--------
Gas Level:  357.80
--------
--------
Gas Level:  399.08
--------
--------
Gas Level:  412.84
--------
--------
Gas Level:  440.37
--------
--------
Gas Level:  454.13
--------
--------
Gas Level:  467.89
--------
--------
Gas Level:  481.65
--------
--------
Gas Level:  495.41
--------
--------
Gas Level:  495.41
--------
--------Published Gas Level = 495 ppm
 Gas Level:  509.17to IBM Watson

--------
--------
Gas Level:  536.70
--------
--------
Gas Level:  550.46
--------
--------
Gas Level:  564.22
--------
--------
Gas Level:  577.98
--------
Published Gas Level = 577 ppm to IBM Watson
```

Set Gas Level:

Current Gas Level:

577.98 ppm

8. Testing:



ai2.appinventor.mit.edu/#5332428030607360

Alerting App

Gas leakage Monitoring & Alerting System For

Team ID - PNT2022TMID42277

Gas Level ppm

Status

Properties: Screen3

AlignHorizontal: Center : 3

AlignVertical: Top : 1

BackgroundColor: None

BackgroundImage: close.webp...

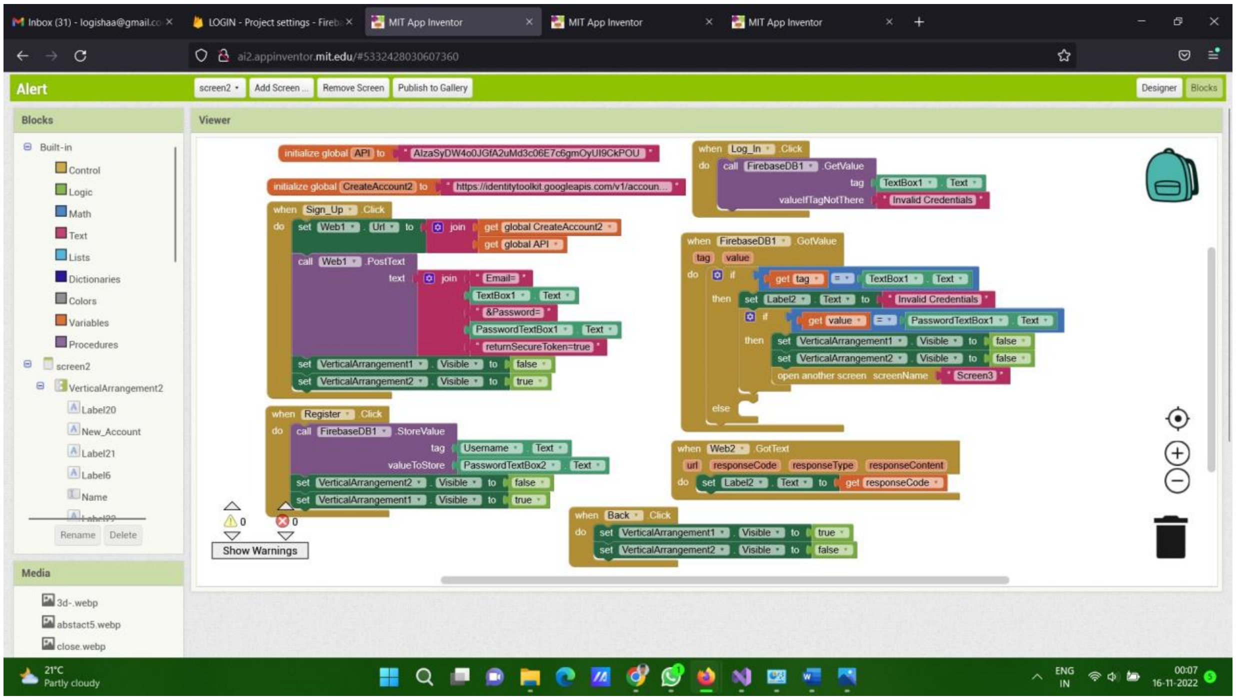
CloseScreenAnimation: Default

OpenScreenAnimation: Default

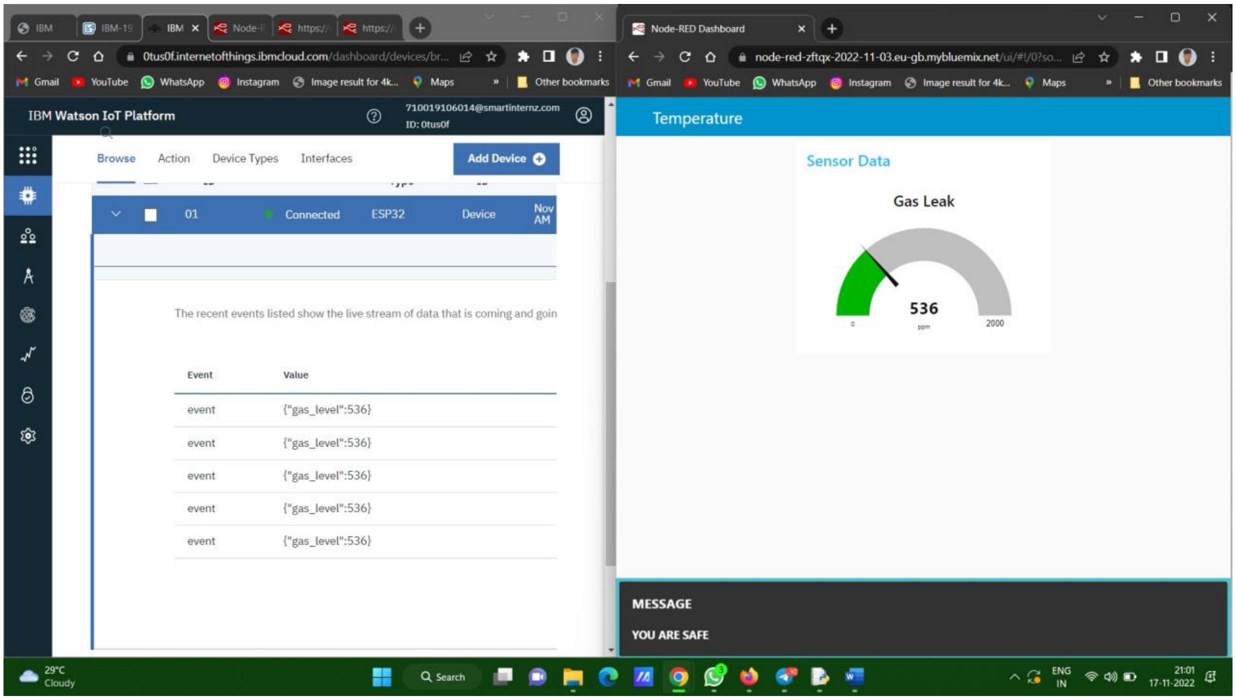
ScreenOrientation: Unspecified

Scrollable: ☑

ShowStatusBar: ☑



```
initialize global API to "AlzaSyDW4o0JGfA2uMd3c06E7c6gmOyUI9CkPOU"

initialize global CreateAccount2 to "https://identitytoolkit.googleapis.com/v1/accoun..."

when Sign_Up.Click
do  set Web1.Url to join  get global CreateAccount2
                          get global API
    call Web1.PostText
         text  join  "Email="
                     TextBox1.Text
                     "&Password="
                     PasswordTextBox1.Text
                     "returnSecureToken=true"
    set VerticalArrangement1.Visible to false
    set VerticalArrangement2.Visible to true

when Register.Click
do  call FirebaseDB1.StoreValue
         tag  Username.Text
         valueToStore  PasswordTextBox2.Text
    set VerticalArrangement2.Visible to false
    set VerticalArrangement1.Visible to true

when Log_In.Click
do  call FirebaseDB1.GetValue
         tag  TextBox1.Text
         valueIfTagNotThere  "Invalid Credentials"

when FirebaseDB1.GotValue
 tag  value
do  if  get tag = TextBox1.Text
    then  set Label2.Text to "Invalid Credentials"
          if  get value = PasswordTextBox1.Text
          then  set VerticalArrangement1.Visible to false
                set VerticalArrangement2.Visible to false
                open another screen  screenName "Screen3"
    else

when Web2.GotText
 url  responseCode  responseType  responseContent
do  set Label2.Text to get responseCode

when Back.Click
do  set VerticalArrangement1.Visible to true
    set VerticalArrangement2.Visible to false
```



IBM Watson IoT Platform

710019106014@smartinternz.com
ID: 0tus0f

01 Connected ESP32 Device Nov AM

The recent events listed show the live stream of data that is coming and goin

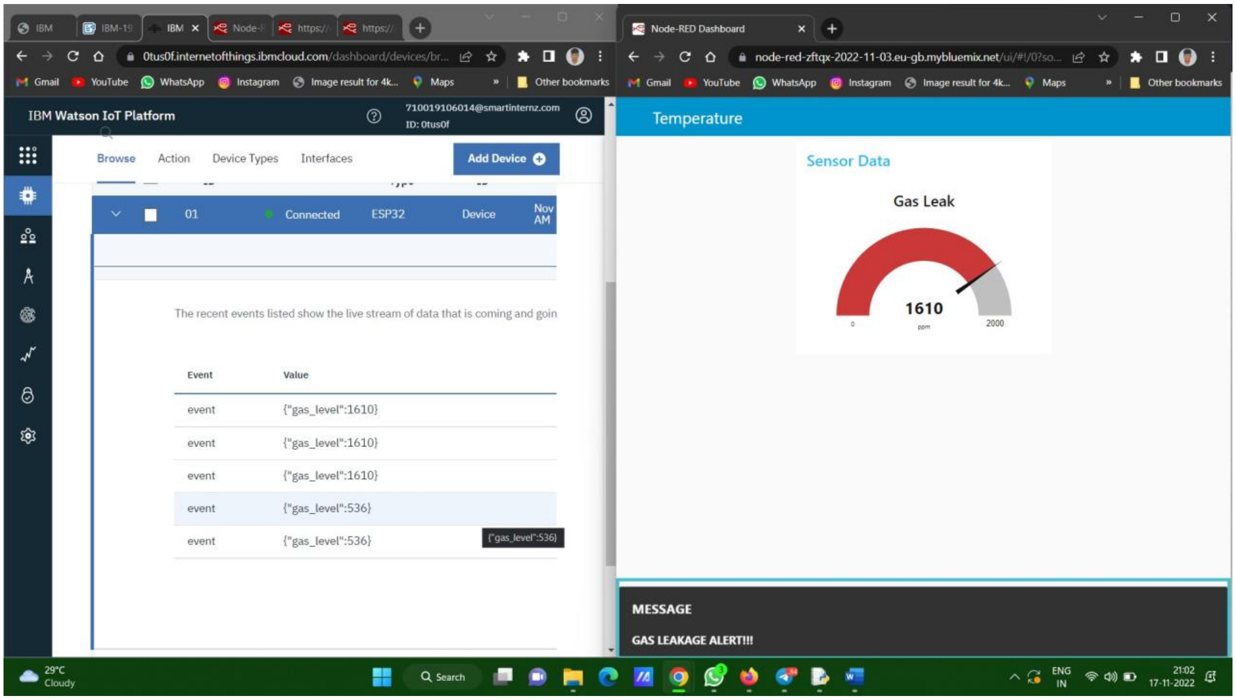
Event	Value
event	{"gas_level":536}
event	{"gas_level":536}
event	{"gas_level":536}
event	{"gas_level":536}
event	{"gas_level":536}

Temperature

Sensor Data

MESSAGE

YOU ARE SAFE



IBM Watson IoT Platform

710019106014@smartinternz.com
ID: 0tus0f

01 Connected ESP32 Device Nov AM

The recent events listed show the live stream of data that is coming and goin

Event	Value
event	{"gas_level":1610}
event	{"gas_level":1610}
event	{"gas_level":1610}
event	{"gas_level":536}
event	{"gas_level":536}

Temperature

Sensor Data

MESSAGE

GAS LEAKAGE ALERT!!!

9. Result:

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

10. Advantages/Disadvantages:

10.1 Advantages:

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

10.2 Disadvantages:

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

11. CONCLUSION:

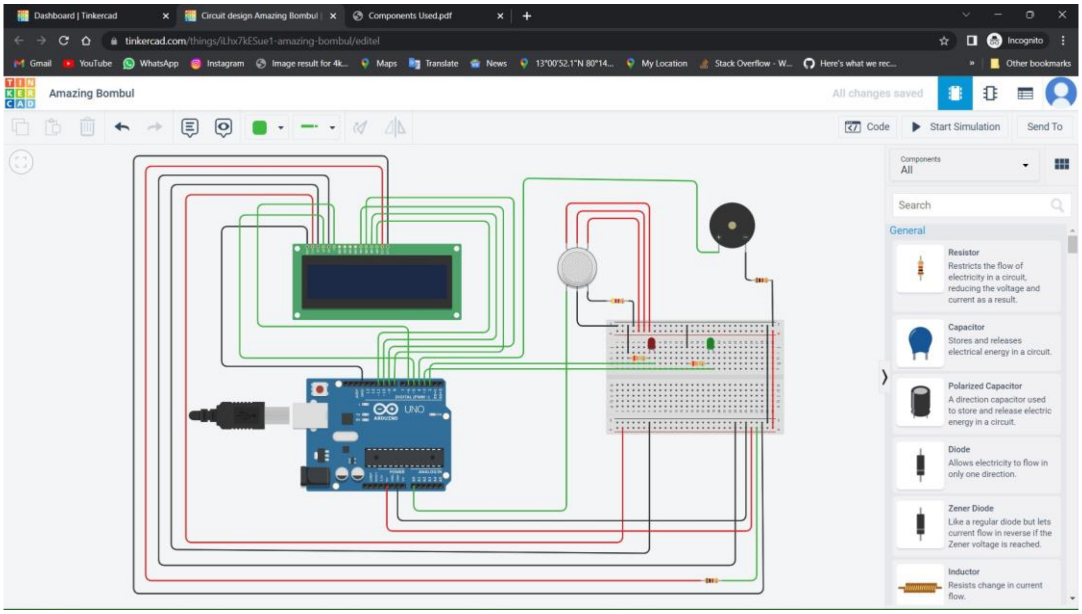
This gas leak detector system contains two features, this includes the SMS Gateway feature for only sending warning information regarding the gas leak to user, and the alarm for the warning alert. There is some improvement which can be applied for the future work, such as regarding the SMS Gateway, it need to enhance with feature such as notifying the user whenever the remaining credit balance is insufficient. Another thing which can be enhanced is regarding the sensor, the sensors in this module do not include somewhat notification for notifying the user whenever the sensor not working properly or not connected to the micro-controller for some cases, therefore, it is recommended to add this kind of features in the future work for better refinement.

12. FUTURE SCOPE:

We propose to build the system using an MQ6 gas detection sensor and interface it with an Aurdino Uno microcontroller along with an LCD Display. This system uses the gas sensor to detect any gas leakages. The gas sensor sends out a signal to the microcontroller as soon as it encounters a gas leakage. The microcontroller processes this signal and a message is displayed on the LCD to alert the user.

13. APPENDIX:

13.1 Circuit Diagram:



Resistor
Restricts the flow of electricity in a circuit, reducing the voltage and current as a result.

Capacitor
Stores and releases electrical energy in a circuit.

Polarized Capacitor
A direction capacitor used to store and release electric energy in a circuit.

Diode
Allows electricity to flow in only one direction.

Zener Diode
Like a regular diode but lets current flow in reverse if the Zener voltage is reached.

Inductor
Resists change in current flow.

13.2 Components:

The design of a sensor-based automatic gas leakage detector with an alert andcontrol system. The components are

S.NO	NAME OF THE COMPONENT	QUANTITY
1	Arduino Uno R3	1
2	LCD 16x2	1
3	Piezo	1
4	Gas sensor	1
5	1 k ohm Resistor	1
6	2.3 k ohm Resistor	1
7	4.7 k ohm Resistor	1
8	Red LED	1
9	Green LED	1

13.4 Source Code:

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(5,6,8,9,10,11);

int redled = A5;
int greenled = A3;
int buzzer = 4;
int sensor = A0;
int sensorThresh = 400;
void setup()
{
pinMode(redled, OUTPUT);
pinMode(greenled,OUTPUT);
pinMode(buzzer,OUTPUT);
pinMode(sensor,INPUT);
Serial.begin(9600);
lcd.begin(16,2);
}
void loop()
{
  int analogValue = analogRead(sensor);
  Serial.println(analogValue);
  if(analogValue>sensorThresh)
  {
    digitalWrite(redled,HIGH);
    digitalWrite(greenled,LOW);
    tone(buzzer,1000,10000);
    lcd.clear();
    lcd.setCursor(0,1);
    lcd.print("ALERT");
    Serial.print("ALERT");
    delay(1000);
    lcd.clear();
    lcd.setCursor(0,1);
    lcd.print("EVACUATE");
    Serial.println(" -- EVACUATE");
    delay(1000);
}
  else
  {
    digitalWrite(greenled,HIGH);
    digitalWrite(redled,LOW);
    noTone(buzzer);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("SAFE");
    Serial.print("SAFE");
    delay(1000);
    lcd.clear();
    lcd.setCursor(0,1);
    lcd.print("ALL  CLEAR");
    Serial.println(" -- ALL CLEAR");
    delay(1000);
}
}
```

13.5 GITHUB:

Link : https://github.com/IBM-EPBL/IBM-Project-24440-1659942903

13.6 Demo Video:

LiNK: https://youtu.be/Eqz5v_VvZKQ