GOVERNMENT COLLEGE OF ENGINEERING CHETTIKARAI, DHARMAPURI

636704

Affiliated to Anna university, Chennai

GAS LEAKAGE MONITORING AND ALERTING SYSTEM– IOT BASED

IBM NALAIYATHIRAN

Document Title

TITLE	Gas Leakage Monitoring & Alerting System For Industries
DOMAIN NAME	INTERNET OF THINGS
TEAM ID	PNT2022TMID41307
TEAM LEADER NAME	M. VEERAMANI
TEAM MEMBER NAME	M. DEEPAK KUMAR R. NAVEEN KUMAR S. THIRUMAL
MENTOR NAME	Dr. THIYAGARAJAN RATHINAM

1. INTRODUCTION

1.1 Project Overview :

Fires occur in many gas handling industries due to human carelessness. This leads to massive loss of life and economic collapse.So to control this we make it computerized. Now this technology has developed and become IoT technology.The term IoT, or Internet of Things, refers to the collective network of connected devices and the technology that facilitates communication between devices and the cloud, as well as between the devices themselves.Developing the python script for publishing the sensors data values to the IBM IOT platform and alerting the authority when the Gas level crosses its threshold level. The gas level can be monitored through web App.

1.2 Purpose :

Gas detectors measure and indicate the concentration of certain gases in an air via different technologies. Typically employed to prevent toxic exposure and fire, gas detectors are often battery operated devices used for safety purposes. They are manufactured as portable or stationary (fixed) units and work by signifying high levels of gases through a series of audible or visible indicators, such as alarms, lights or a combination of signals.

While many of the older, standard gas detector units were originally fabricated to detect one gas, modern multifunctional or multi-gas devices are capable of detecting several gases at once. Some detectors may be utilized as individual units to monitor small workspace areas, or units can be combined or linked together to create a protection system.

2. LITERATURE SURVEY

2.1 Existing problem :

S.NO	Paper Title	Author Name	Publication Year	Result
1	Internet of Things (IoT) based Gas Leakage Monitoring and Alerting System with MQ-2 Sensor.	Rohan Chandrapandey, Mnish Verma,Lumesh Kumar Sahu	2017	This paper choice of using a real time gas leakage monitoring and sensing the output levels of gas has been clearly observed by the help of this system.
2	Gas Leakage Detection and Smart Alerting and Prediction Using loT	Asmita Varma, Prabhakar S. Kayalvizhi Jayavel	2017	The proposed gas leakage detector is promising in the Field of safety.
3	IOT Based Gas Leakage Detection System with Database Logging, Prediction and Smart Alerting	Chaitali Bagwe,Vidya Ghadi, Vinayshri Naik, Neha Kunte	2018	The system provides constant monitoring and detection of gas leakage along with storage of data in database for predictions and analysis. The IOT components used helps in making the system much more cost effective in comparison with traditional Gas detector systems.
4	Internet of Things (IoT) Based Gas Leakage Monitoring and Alerting System with Mq-6 Sensor	Rohan Chandra Pandey, Manish Verma, Lumesh Kumar Sahu, Saurabh Deshmukh	2018	A discussion on how the aims and objectives are met is presented. An overall conclusion IOT based toxic gas detector is it has become more efficient, more applicable to today's applications and smarter.

5	Gas Leakage Detection and Smart Alerting System Using IoT	Shital Imade, Priyanka Rajmanes, Aishwarya Gavali	2018	In this paper we use IOT technology for enhancing the existing safety standards. While making this prototype has been to bring a revolution in the field of safety against the leakage of harmful and toxic gases.

Link :

https://github.com/IBM-EPBL/IBM-Project-35110-1660281675/blob/main/PROJECT%20DESIGN%20AND%20PLANNING/IDEATION/GAS%20LEAKAGE%20MONITORING%20AND%20ALERTING%20SYSTEM-LITERATURE%20SURVEY%20.pdf

2.2 References :

[1]. Mr. Sameer Jagtap, Prajkta Bhosale, Priyanka Zanzane, Jyoti Ghogare, "LPG Gas Weight and Leakage Detection System Using IoT", International Journal for Research in Applied Science & Engineering Technology", Volume 4, Issue 3, March 2016, Pg – 716 to 720.

[2]. Arun Raj, Athira Viswanathan, Athul T S, "LPG Gas Monitoring System", International Journal of Innovative Technology and Research, Volume 3, Issue 2, February 2015, Pg – 1957 to 1960.

[3]. S Shyamaladevi, V. G. Rajaramya, P. Rajasekar, P. Sebastin Ashok, "ARM7 based automated high-performance system for lpg refill booking & leakage detection", Journal of VLSI Design and Signal Processing", Volume 3, Issue 2, 2014.

[4]. S. Sharma, V. N. Mishra, R. Dwivedi, R. Das, "Classification of gases/odours using Dynamic Response of Thick Film Gas Sensor Array", IEEE Conference on Sensors Journal, 2013.

[5]. Rajeev B. Ahuja, Jayant K. Dash, Prabhat Shrivastava, "A comparative analysis of liquefied petroleum gas (LPG) and kerosene related burns", Burns, Volume 37, Issue 8, December 2011, Pg – 1403 to 1410.

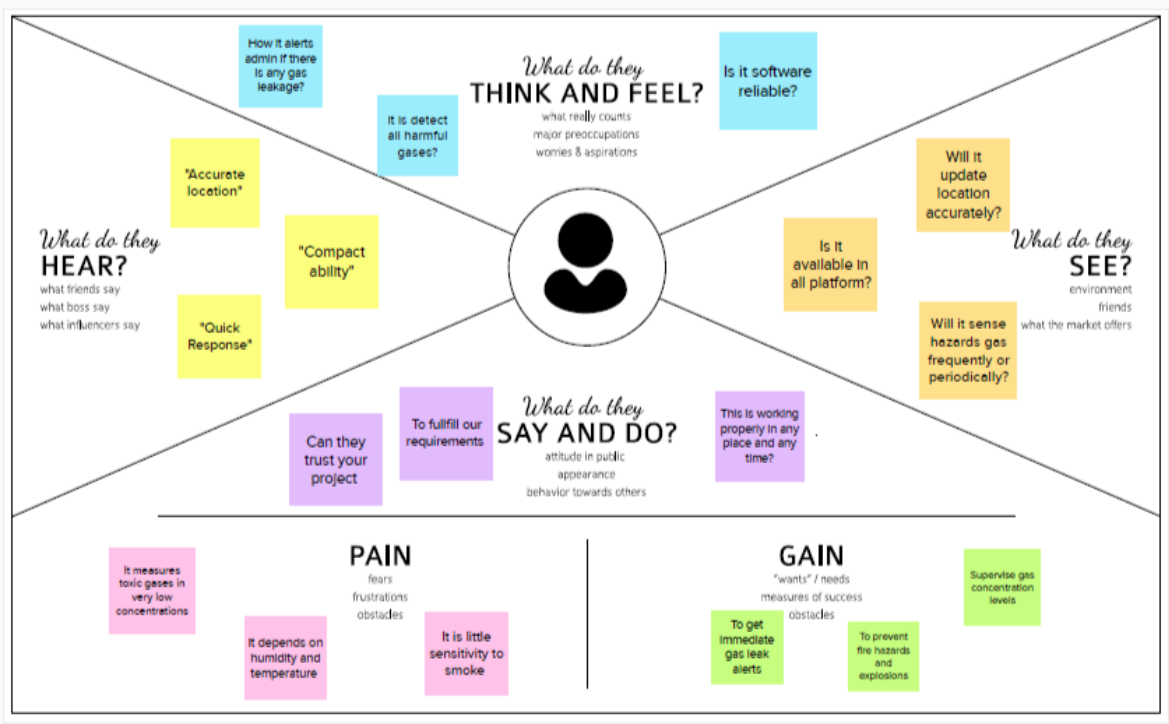
[6]. Prof. Pankaj C. Warule, Shivam Upadhyay, Snehal S. Shelke, Sumitra K. Khandade, "LPG Detection, Metering and Control System Using Microcontroller", IJARIIE, Volume 2, Issue 2, 2016, Pg – 648 to 652.
[7]. Ankit Sood, Babalu Sonkar, Atul Ranjan, Mr. Ameer Faisal, "Microcontroller Based LPG Gas Leakage Detector Using GSM Module", International Journal of Electrical and Electronics Research, Volume 3, Issue2, April- June 2015, Pg – 264 to 269.
[8]. Ashish Shrivastava, Ratnesh Prabhakar, Rajeev Kumar, Rahul Verma, "GSM Based Gas Leakage Detection System", International Journal of
Technical Research and Applications", Volume 1, Issue2, May- June 2013, Pg – 42 to 45.
[9]. Shivalingesh B. M, Ramesh C, Mahesh S. R, Pooja R, Preethi K. Mane, Kumuda S, "LPG Detection, Measurement and Booking System", IJRSI, Volume 1, Issue 4, November 2014, Pg – 7 to 10.
[10]. C. Selvapriya, S. Prabha Sathya, M. Abdulrahim, C. K. Aarthi, "LPG Leakage Monitoring and Multilevel Alerting System", International Journal of Engineering Sciences & Research Technology, Volume 2, Issue 11, November 2013, Pg – 3287 to 3290.
[11]. H. Huang, H. Bainand, S. Zhu, "A Greenhouse Remote Monitoring System Based on GSM", in Proceedings of IEEE International Conference on Information Management, 2011, Pg – 357 to 360.
[13]. J. Tsado, O. Imoru, S.O. Olayemi, "Design and construction of a GSM based gas leak Alert system", IEEE Transaction, IRJEEE Vol. 1(1), pp. 002-006, September, 2014.

2.3 Problem Statement Definition :

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.

3. IDEATION & PROPOSED SOLUTION

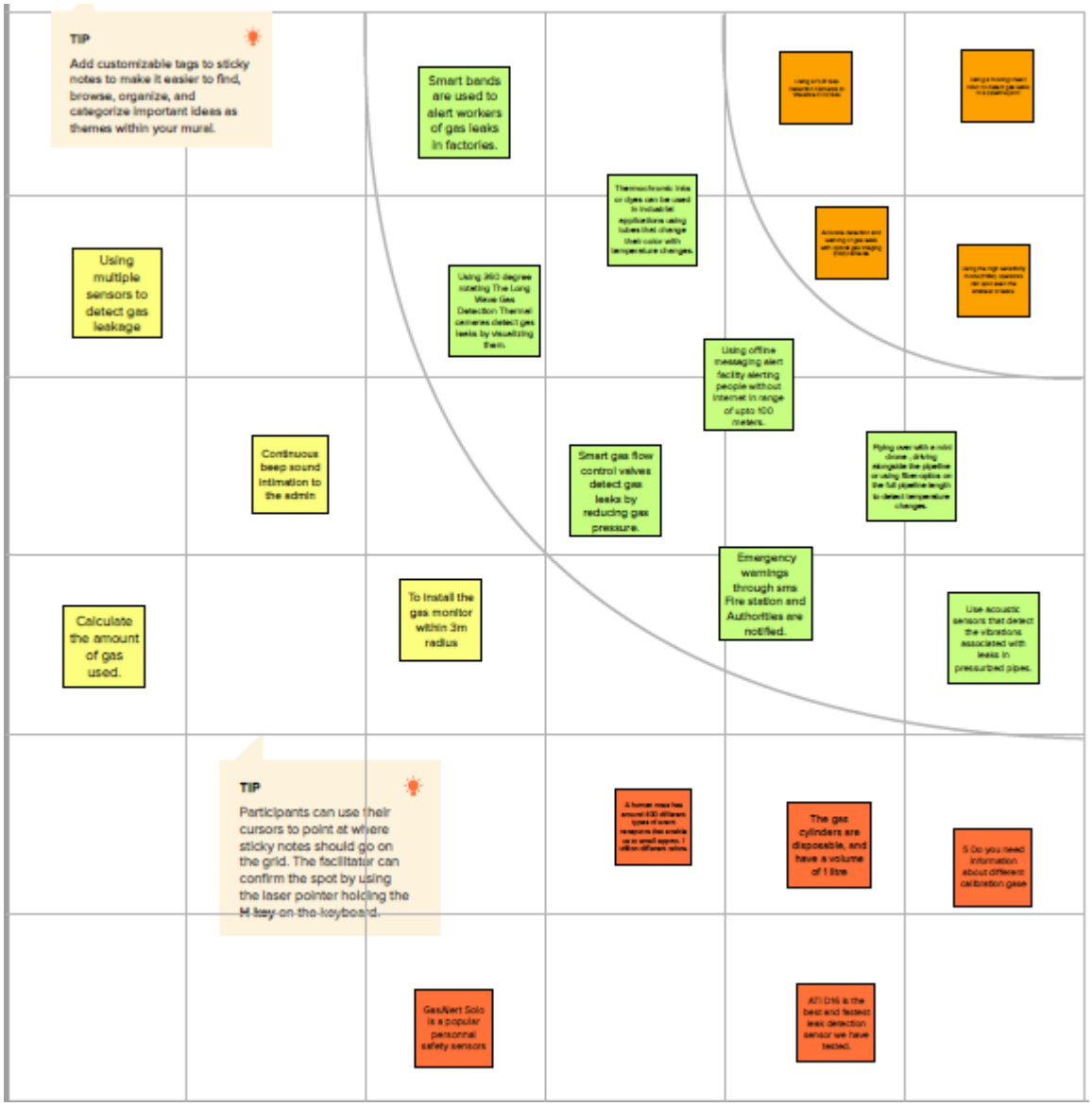
3.1 Empathy Map Canvas :



Link:

https://github.com/IBM-EPBL/IBM-Project-35110-1660281675/blob/main/PROJECT%20DESIGN%20AND%20PLANNING/IDEATION/EMPATHY%20MAP.pdf

3.2 Ideation & Brainstorming :



Link:

https://github.com/IBM-EPBL/IBM-Project-35110-1660281675/blob/main/PROJECT%20DESIGN%20AND%20PLANNING/IDEATION/BRAINSTORMING.pdf

3.3 **Proposed Solution :**

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	1. Develop an efficient system & an application that can monitor and alert the users(workers). 2. This project helps the industries in monitoring the emission of harmful gases. 3. In several areas, the gas sensors will be integrated to Monitor the gas leakage. 4. If in any area gas leakage is detected the admins will be notified along with the Location. 5. In the web application, admins can view the sensor Parameters.
2.	Idea / Solution description	1. Using a moving insect robot to detect gas leaks in a pipeline joint. 2. Using a FLIR Gas Detection Cameras to Visualize CO2 Gas. 3. Calculate the amount of gas used. 4. Smart bands are used to alert workers of gas leaks in factories. 5. Using 360 degree rotating The Long Wave Gas Detection Thermal cameras detect gas leaks by visualizing them. 6. Thermochromic inks or dyes can be used in industrial applications using tubes that change their colour with temperature changes.

		7. Using offline messaging alert facility alerting people without internet in range of upto 100 meters.
3.	Novelty / Uniqueness	1. Using a moving insect robot to detect gas leaks in a pipeline joint. 2. Using 360 degree rotating The Long Wave Gas Detection Thermal cameras detect gas leaks by visualizing them. 3. Fastest alerts to the workers 4. User friendly
4.	Social Impact / Customer Satisfaction	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. Cost efficient 11. Easy installation and provide efficient results 12. Can work with irrespective of fear
5.	Business Model (Revenue Model)	1. It is satisfy to the industries and the industries employees. 2. The product is advertised all over the platforms. Since it is economical, even helps small scale industries from disasters. 3. As the product usage can be understood by everyone, it is easy for them to use it properly for their safest organization.

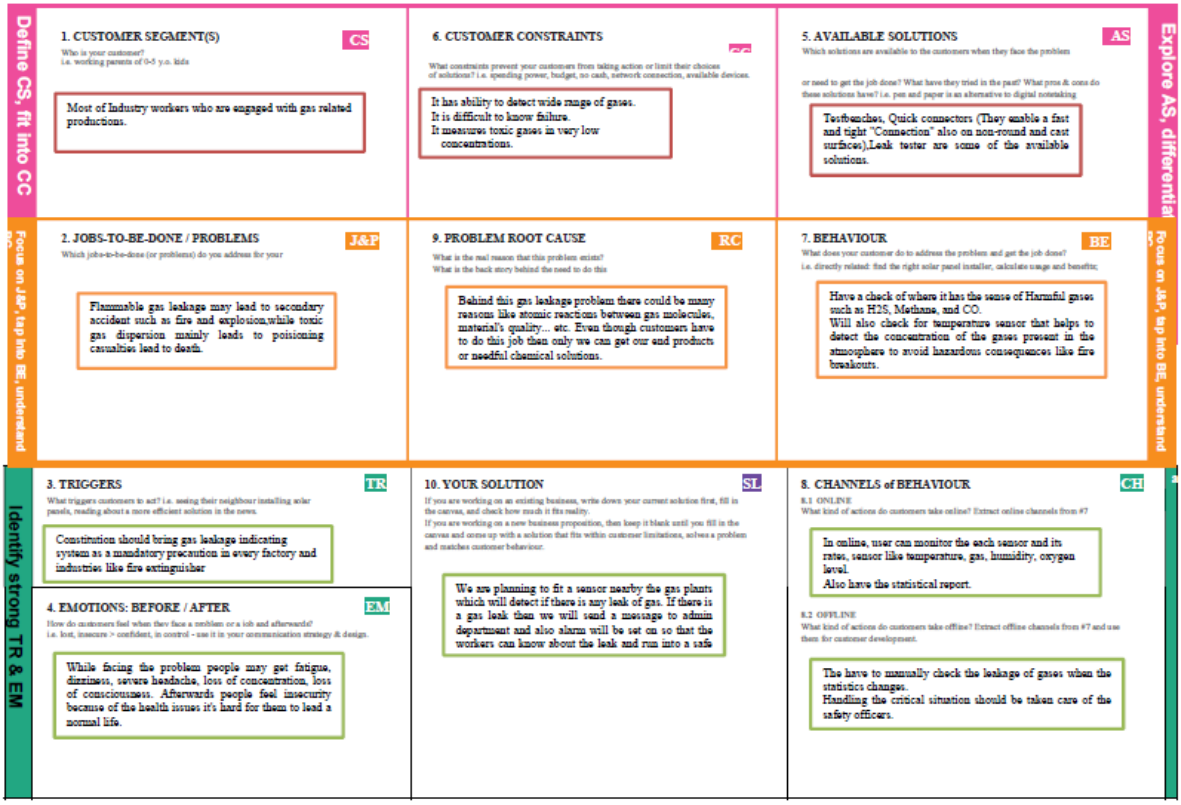
6.	Scalability of the Solution	1. It can automatically detect ,alarm and control gas leakage using an exhaust fan to suck the gas away 2. Since the product is cost efficient, it can be placed in many places in the industries. 3. Even when the gas leakage is more, the product sense the accurate values and alerts the workers effectively.

Link :

https://github.com/IBM-EPBL/IBM-Project-35110-1660281675/blob/main/PROJECT%20DESIGN%20AND%20PLANNING/DESIGN%20PHASE%201/Proposed%20Solution%20(1).pdf

3.4 Problem Solution fit :

Project Title: Gas Leakage Monitoring and Alerting System **Project Design Phase-I - Solution Fit Template**



Link :

https://github.com/IBM-EPBL/IBM-Project-35110-1660281675/blob/main/PROJECT%20DESIGN%20AND%20PLANNING/DESIGN%20PHASE%201/Problem%20solution%20fit.pdf

4. REQUIREMENT ANALYSIS

4.1 Functional requirement :

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Visibility	Level of gas can be monitored by users if there is any leakage, alerts can be sent through messages.
FR-2	User Reception	Level of gas can be monitored by users if there is any leakage, alerts can be sent through messages.
FR-3	User Understanding	The user can monitor the level of gas with the help of the data. If there is an increase in gas level then the alert will be given. They also get notified by the alert
FR-4	User Convenience	Through message we can easily get data of gas level and in case of gas leakage, it can directly send notifications to nearby police station and hospital.
FR-5	User Performance	When the user gets notified, he could turn on the exhaust fan/sprinkler.

4.2 Non-Functional requirements :

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	It updates the data regularly as well as protects the workers.
NFR-2	Security	As a result of emergency alert, we can be able to protect both the humans and properties.

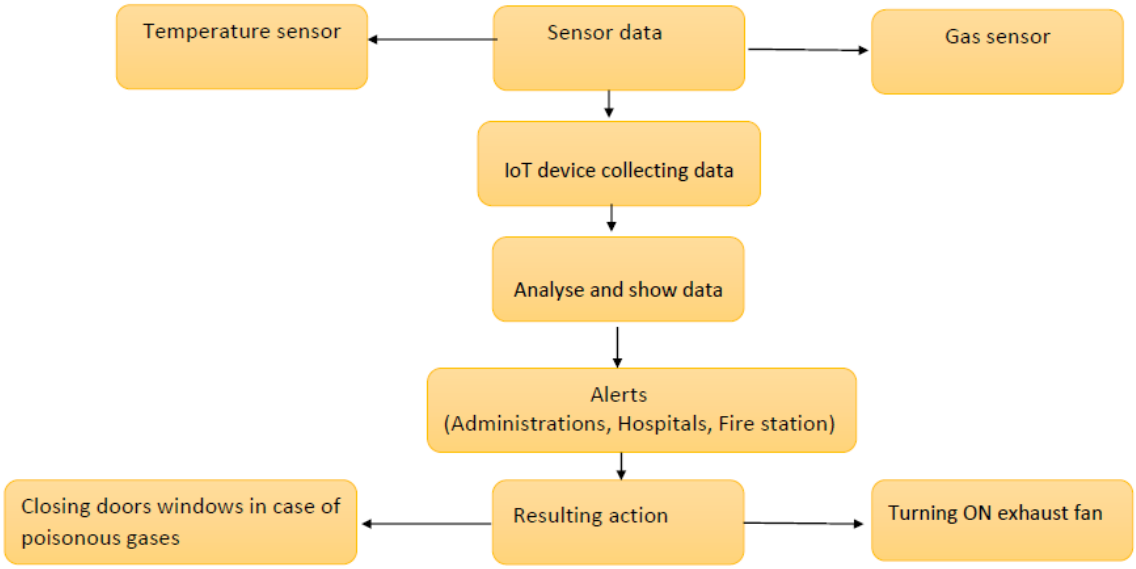
NFR-3	Reliability	Can be able to provide accurate values. It might have a capacity to recognize the smoke accurately and does not give a false
NFR-4	Performance	Sprinklers and exhaust fans are used in case of emergency.
NFR-5	Availability	It can be used for everyday; it includes day and nights.
NFR-6	Scalability	Sensors can be replaced every time it fails.

Link :

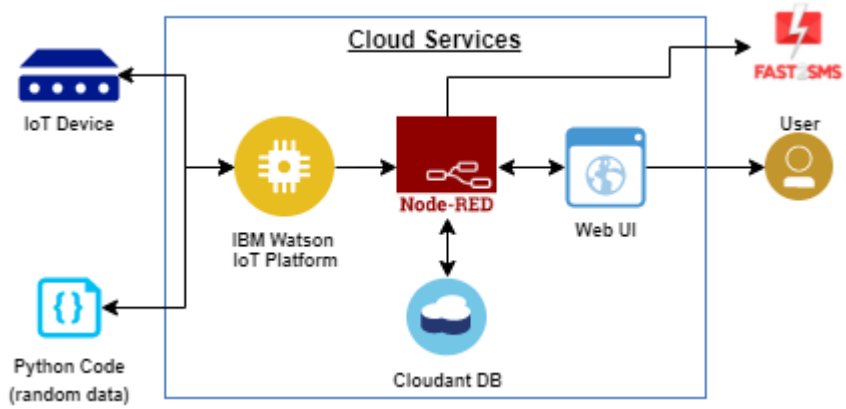
https://github.com/IBM-EPBL/IBM-Project-35110-1660281675/blob/main/PROJECT%20DESIGN%20AND%20PLANNING/DESIGN%20PHASE%202/Solution%20Requirements.pdf

5. PROJECT DESIGN

5.1 Data Flow Diagrams :



5.2 Solution & Technical Architecture :



1. Feed the data from the sensor to the web interface.
2. The data will display in the web page of the authority (user).

3. The collected data is sent to the data base, where the collected data is checked and monitored if the data exceeds then SMS alert send to the admin.
4. The data is provided to the cloud service and stored
5. The authority monitors the web page continuously to collect the data and send the alert to the authority.

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	User can enter into the web application	I can access my account / dashboard	High	Sprint-1
		USN-2	User can register their credentials like email id and password	I can receive confirmation email & click confirm	High	Sprint-1
	Login	USN-3	User can log into the application by entering email & password	I can register & access the dashboard with Facebook Login	High	Sprint-1
	Dashboard	USN-4	User can view the temperature	I can view the data given by the device	High	Sprint-2
		USN-5	User can view the level of gas	I can view the data given by the device	High	Sprint-2
Customer (Web user)	Usage	USN-1	User can view the web page and get the information	I can view the data given by the device	High	Sprint-3
	Working	USN-1	User act according to the alert given by the device	I can get the data work according to it	High	Sprint-3
		USN-2	User turns ON the exhaust fan/sprinkler when the leakage occurs	I can get the data work according to it	High	Sprint-4
Customer Care Executive	Action	USN-1	User solve the problems when someone faces any usage issues	I can solve the issues when some one fails to understand the procedure	High	Sprint-4
Administrator	Administration	USN-1	User stores every information	I can store the gained information	High	Sprint-4

Link :

https://github.com/IBM-EPBL/IBM-Project-35110-1660281675/blob/main/PROJECT%20DESIGN%20AND%20PLANNING/DESIGN%20PHASE%202/Data%20Flow%20Diagrams%20and%20User%20Stories-1.pdf

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation :

S.NO	ACTIVITY TITLE	ACTIVITY DESCRIPTION	DURATION
1	Understanding the project requirement	The Aim is team members are assigned with tasks for each to be executed as a responsible team lead. Also create repository in the Github, Assign members and teach how to use and open the Github and IBM career education portals.	1 WEEK
2	Starting of project	Advisory of team lead to his team members based on regularly attending training sessions for installing and use of prerequiste .Also necessarily attending the training sessions based on python code,development of android app in mobile app invtr.com and working along NodeRed is ensured by the team lead and acknowledged by team members simultaneously .	1 WEEK
3	Attend class	Team members and team lead must watch and learn from classes provided by IBM and NALAYATHIRAN and must gain access of MIT license for their project.	4 WEEK
4	Budget and scope of project	Budgetary planning process taken up on whole as a team to **detect the user** compatible price to the buy the p**roduct based on budgetary** on IOT and component level.	1 WEEK

6.2 Sprint Delivery Schedule :

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Objective	USN-1	As a system, the gas sensor should detect the gas	8	High	Veeramani Deepak kumar
Sprint-1	Features	USN-2	As a system, the gas sensor values should be displayed in a LCD screen	2	Low	Naveen kumar Deepak kumar
Sprint-1	Features	USN-3	As a system, as soon as the detected gas reaches the threshold level, the red color LED should be turned ON.	5	High	Naveen kumar Veeramani
Sprint-1	Features	USN-4	As a system, as soon as the detected gas reaches the threshold level, the siren should be turned ON.	5	High	Veeramani Thirumal
Sprint-2	Focus	USN-5	As a system, it should the send the location where the gas is detected	8	High	Thirumal Naveen kumar
Sprint-2	Focus	USN-6	As a system, it should also send the alerting SMS to the registered phone number	2	Low	Veeramani Thirumal

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

Sprint-3	Data Transfer	USN-13	As an application, it should display the details of the gas level and other details to the user through the frontend of the MIT app.	8	High	Veeramani Thirumal
Sprint-4	Registration	USN-14	As a user, I must first register my email and mobile number in the website	2	High	Deepak kumar Veeramani

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-4	Registration	USN-15	As a user, I must receive confirmation mail and SMS on registration	2	Medium	Thirumal Naveen kumar
Sprint-4	Login	USN-16	As a user, I can login into the web application through email and password.	3	High	Veeramani Naveen kumar
Sprint-4	Dashboard	USN-17	As a user, I can access the dashboard and make use of available resources.	2	Medium	Veeramani Deepak kumar
Sprint-4	Focus	USN-18	As a user, I must receive an SMS once the leakage is detected.	5	High	Deepak kumar Thirumal

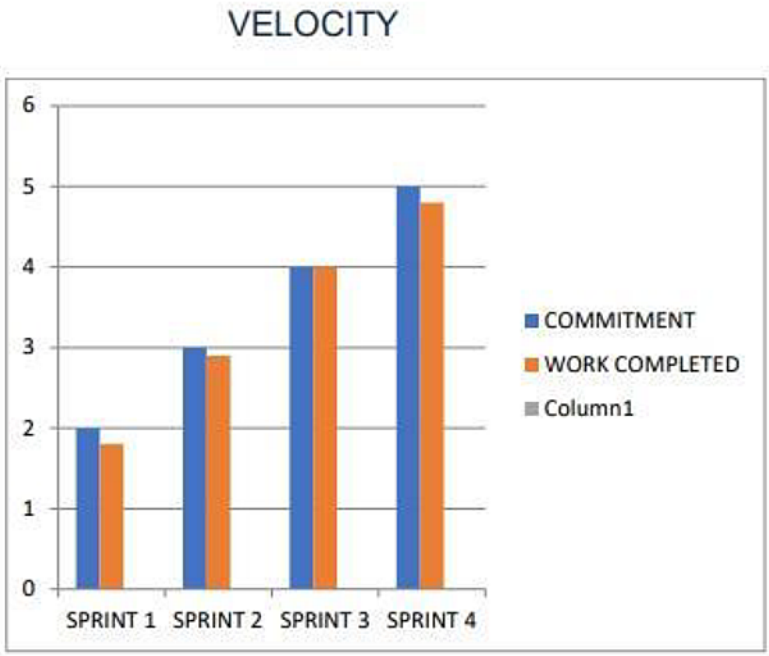
Sprint-4	Allocation	USN-19	As an admin, I must receive information about the leakage along with location and share exact location and route to the person.	3	High	Deepak kumar Naveen kumar
Sprint-4	Allocation	USN-20	As an admin, I must allot particular person to look after the leakage in a particular location.	3	High	Veeramani Thirumal

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)		Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022		20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022		30	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022		49	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022		50	19 Nov 2022

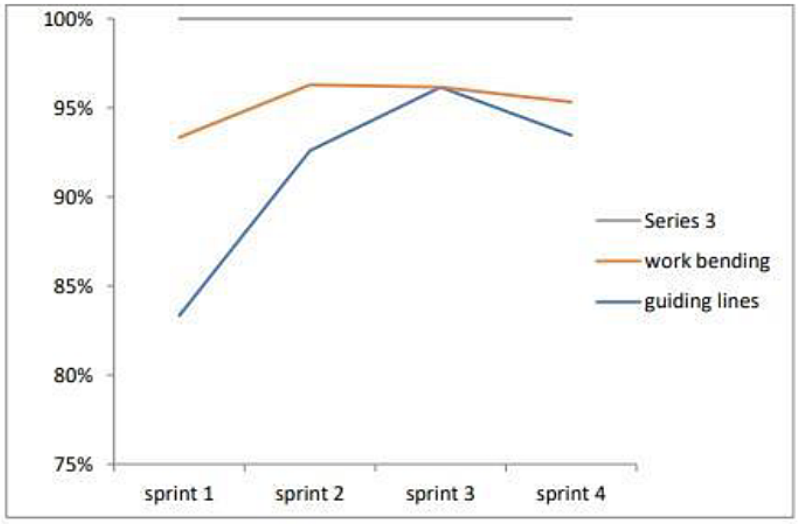
Velocity:

Imagine we have a10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day).

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



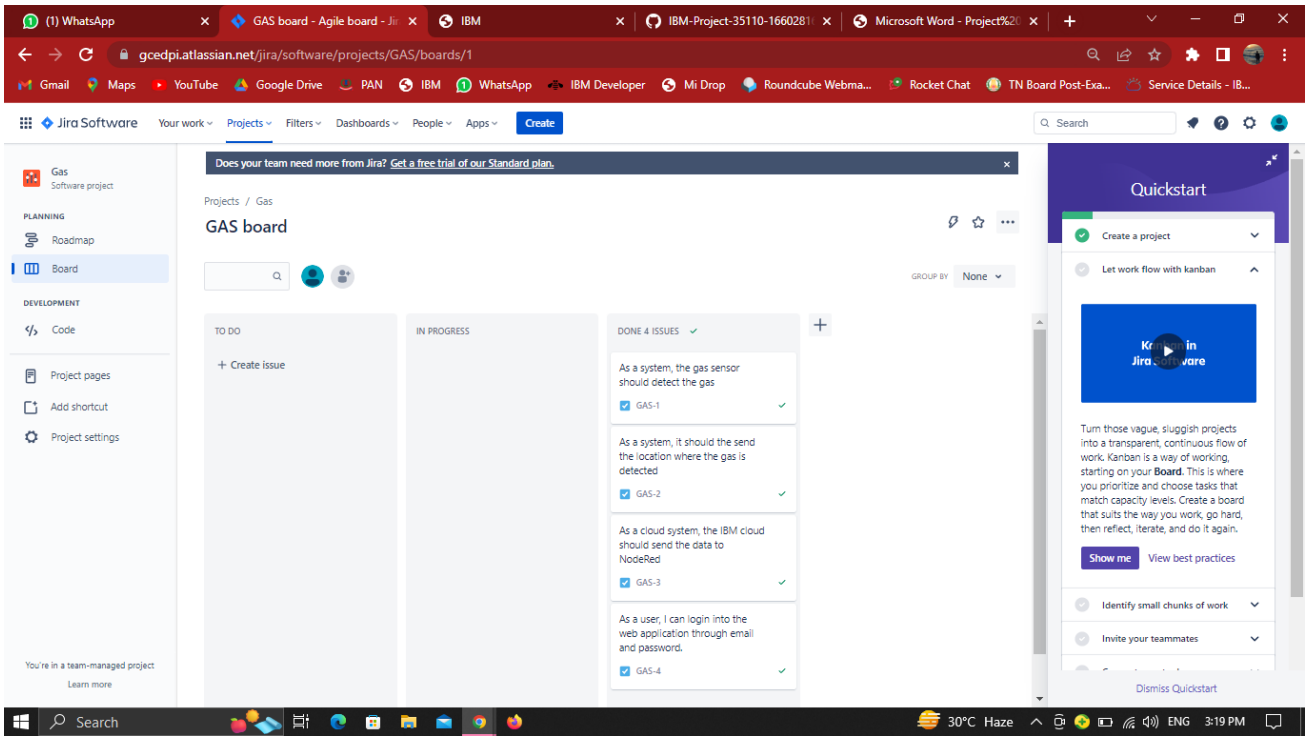
BURNDOWN CHART :

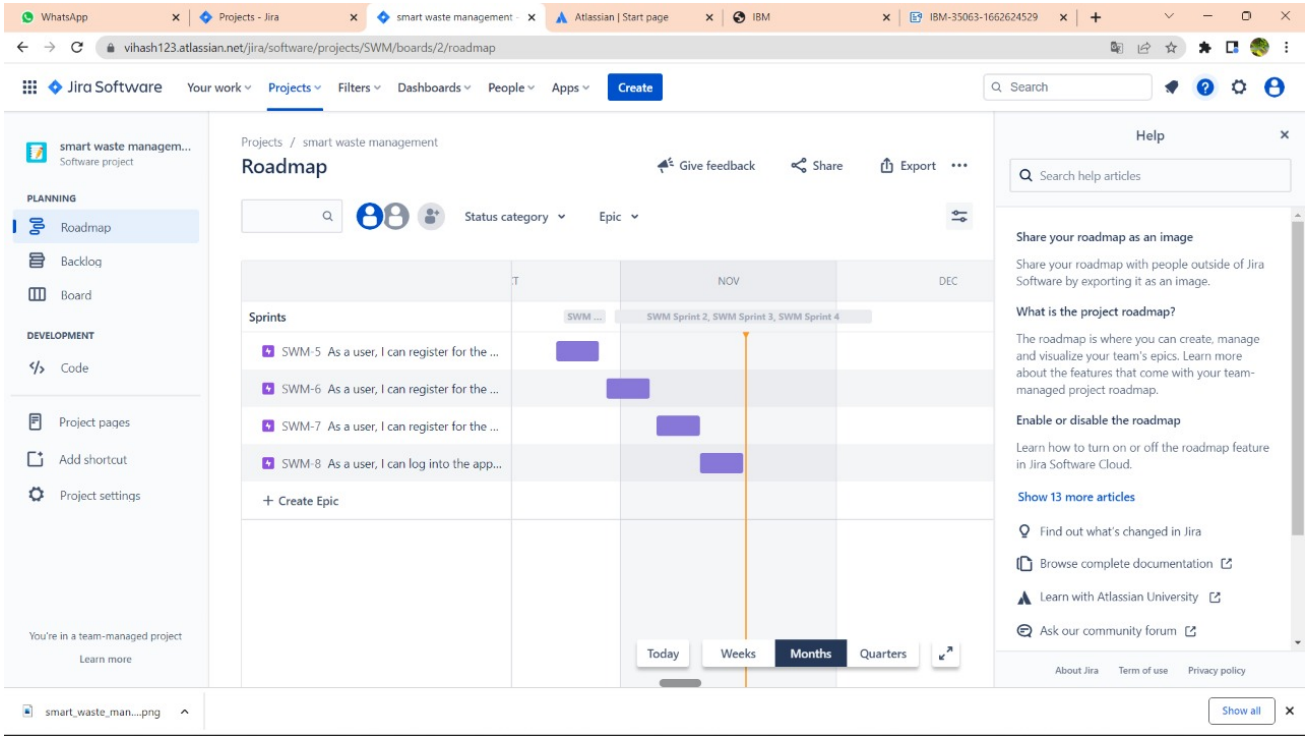


Link :

https://github.com/IBM-EPBL/IBM-Project-35110-1660281675/tree/main/PROJECT%20DESIGN%20AND%20PLANNING/PROJECT%20PLANNING

6.3 Reports from JIRA :





Link :

https://github.com/IBM-EPBL/IBM-Project-35110-1660281675/blob/main/PROJECT%20DEVELOPMENT%20PHASE/REPORT%20FROM%20JIRA.pdf

7. CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 Feature :

```
#IBM Watson IOT Platform
#pip install wiotp-sdk
import wiotp.sdk.device
import time
import random
myConfig = {
    "identity":
    {
        "orgId": "9yby55",
        "typeId": "Gas",
        "deviceId":"18"
    },
    "auth":
    {
        "token": "zIbdsvljWkP@1S34*&"
    }
}

def myCommandCallback(cmd):
    print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="sprinkleron":
        print (" Rainwater sprinkler is ON")
    elif status=="sprinkleroff":
        print (" Rainwater sprinkler is OFF")
    else:
        print ("please send proper command")


client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()

while True:
    TemperatureZ1=random.randint(0,100)
    HumidityZ1=random.randint(0,100)
```

```
    GasLevelZ1=random.randint(0,100)
    PressureZ1=random.randint(0,100)
    TemperatureZ2=random.randint(0,100)
    HumidityZ2=random.randint(0,100)
    GasLevelZ2=random.randint(0,100)
    PressureZ2=random.randint(0,100)

    myData={'TemperatureZ1':TemperatureZ1,'HumidityZ1':HumidityZ1,'GasLevelZ1':GasLevelZ1,
    'PressureZ1':PressureZ1,'TemperatureZ2':TemperatureZ2,'HumidityZ2':HumidityZ2,'GasLevelZ2':
    GasLevelZ2,'PressureZ2':PressureZ2}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
    print("Published data Successfully: %s", myData ,"to the IBM Platform")
    client.commandCallback = myCommandCallback
    time.sleep(5)
client.disconnect()
```

Link :

https://github.com/IBM-EPBL/IBM-Project-35110-1660281675/blob/main/DEVELOP%20A%20PYTHON%20SCRIPT%20TO%20PUBLISH%20AND%20SUBSCRIBE%20TO%20IBM%20IOT%20PLATFORM/PYTHON%20CODE.pdf

8. TESTING

8.1 Test Cases :

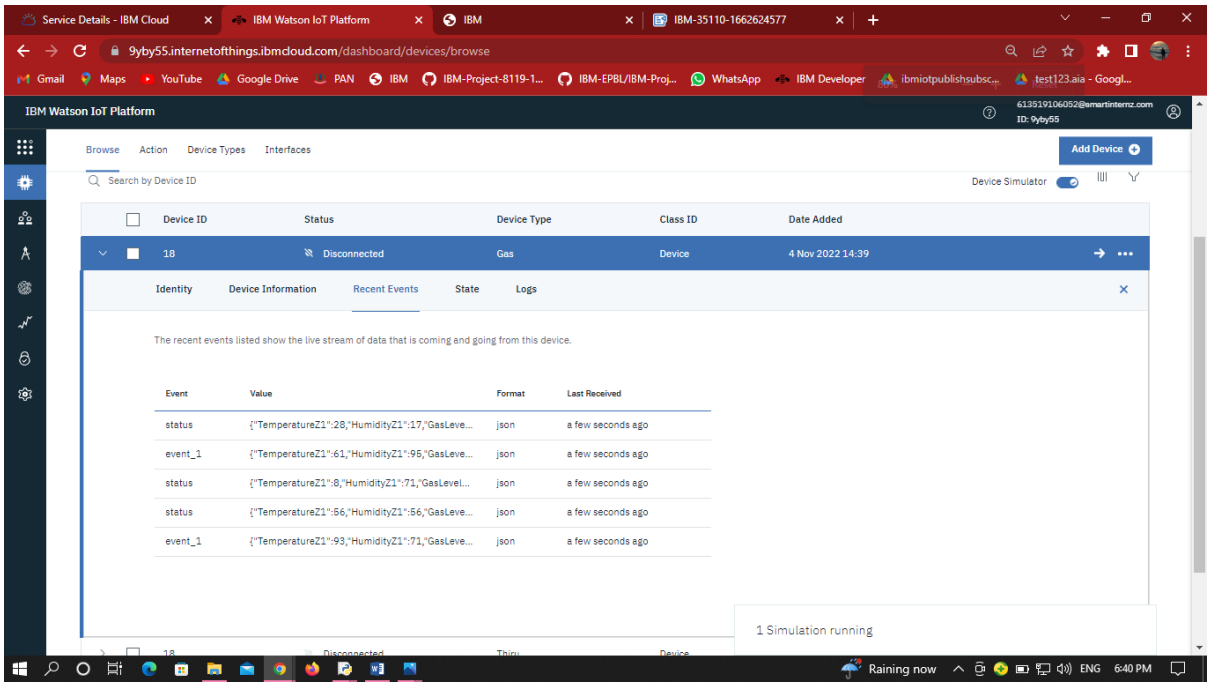
SI.NO	INPUT	OUTPUT	RESULT
01.	Temperature, Humidity, Gas level, Pressure	Gas Level status: high Gas Level status: low	Passed
02.	Temperature, Humidity, Gas level, Pressure	Gas Level status: high Gas Level status: low	Passed
03.	Temperature, Humidity, Gas level, Pressure	Gas Level status: medium Gas Level status: low	Passed
04.	Temperature, Humidity, Gas level, Pressure	Gas Level status: low Gas Level status: low	Passed
05.	Temperature, Humidity, Gas level, Pressure	Gas Level status: high Gas Level status: high	Passed
06.	Temperature, Humidity, Gas level, Pressure	Gas Level status: high Gas Level status: low	Passed
07.	Temperature, Humidity, Gas level, Pressure	Gas Level status: high Gas Level status: medium	Passed
08.	Temperature, Humidity, Gas level, Pressure	Gas Level status: medium Gas Level status: low	Passed
09.	Temperature, Humidity, Gas level, Pressure	Gas Level status: high Gas Level status: low	Passed

10.	Temperature, Humidity, Gas level, Pressure	Gas Level status:low Gas Level status: low	Passed
11.	Temperature, Humidity, Gas level, Pressure	Gas Level status: high Gas Level status: low	Passed
12.	Temperature, Humidity, Gas level, Pressure	Gas Level status: high Gas Level status: low	Passed
13.	Temperature, Humidity, Gas level, Pressure	Gas Level status: high Gas Level status: medium	Passed
14.	Temperature, Humidity, Gas level, Pressure	Gas Level status: high Gas Level status: low	Passed

SI.NO	**INPUT**	**OUTPUT**	**RESULT**
15.	Temperature, Humidity, Gas level, Pressure	Gas Level status: high Gas Level status: medium	Passed
16.	Temperature, Humidity, Gas level, Pressure	Gas Level status: medium Gas Level status: low	Passed
17.	Temperature, Humidity, Gas level, Pressure	Gas Level status: low Gas Level status: low	Passed
18.	Temperature, Humidity, Gas level, Pressure	Gas Level status: high Gas Level status: high	Passed

19.	Temperature, Humidity, Gas level, Pressure	Gas Level status: medium Gas Level status: medium	Passed
20.	Temperature, Humidity, Gas level, Pressure	Gas Level status: low Gas Level status: low	Passed
21.	Temperature, Humidity, Gas level, Pressure	Gas Level status: high Gas Level status: high	Passed
22.	Temperature, Humidity, Gas level, Pressure	Gas Level status: high Gas Level status: medium	Passed
23.	Temperature, Humidity, Gas level, Pressure	Gas Level status: medium Gas Level status: low	Passed
24.	Temperature, Humidity, Gas level, Pressure	Gas Level status: high Gas Level status: low	Passed
25.	Temperature, Humidity, Gas level, Pressure	Gas Level status: high Gas Level status: low	Passed
26.	Temperature, Humidity, Gas level, Pressure	Gas Level status: low Gas Level status: low	Passed
27.	Temperature, Humidity, Gas level, Pressure	Gas Level status: medium Gas Level status: low	Passed
28.	Temperature, Humidity, Gas level, Pressure	Gas Level status: high Gas Level status: low	Passed

Feature code :



8.2 User Acceptance Testing

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 project at the time of the release to User Acceptance Testing (UAT).

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	4	3	2	3	12
Duplicate	0	0	0	0	0
External	1	2	0	4	7
Fixed	5	5	2	6	18
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	0	0	0	0
Totals	10	10	6	14	40

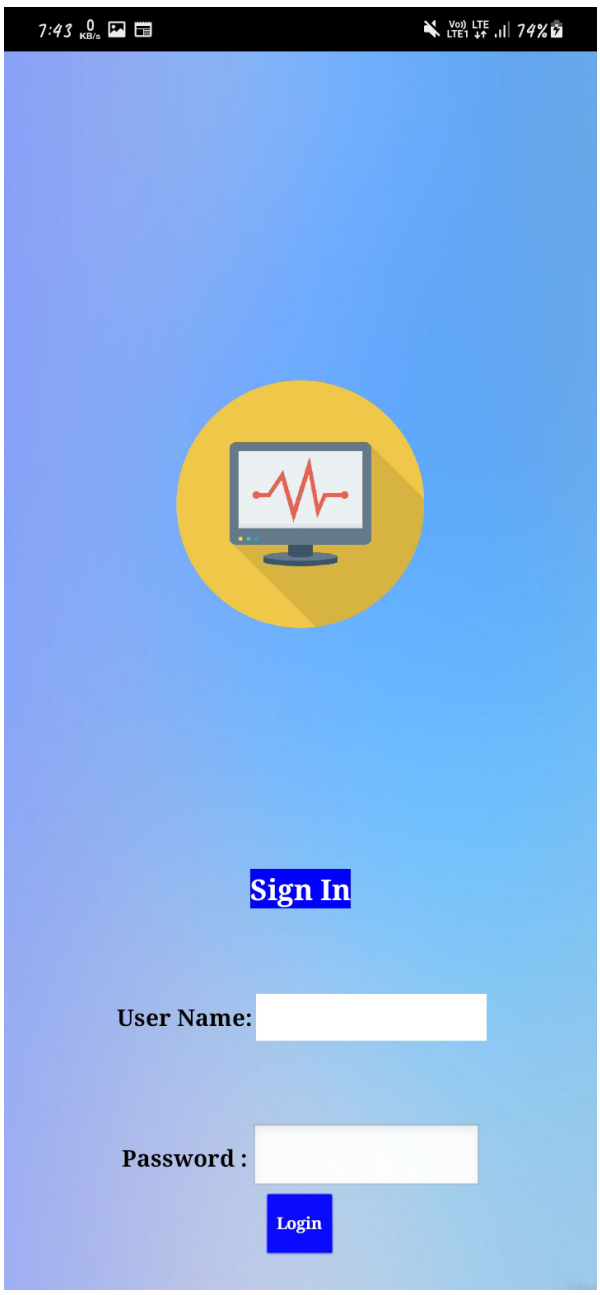
Test Case Analysis :

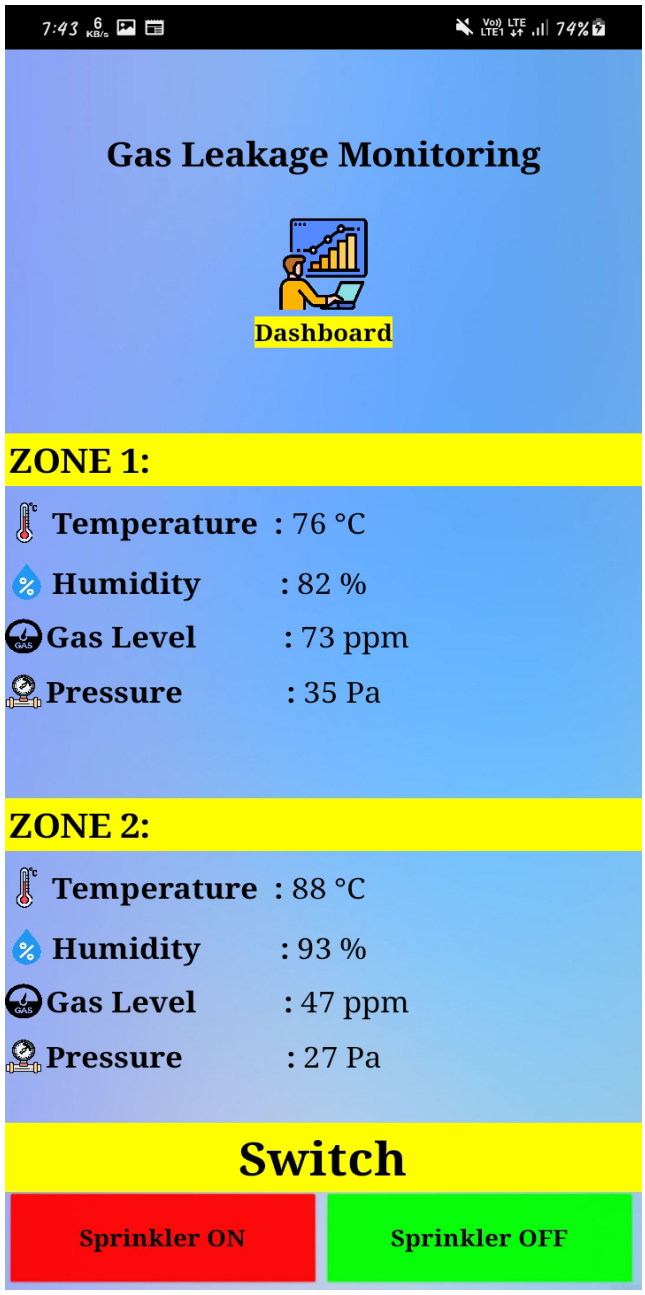
This report shows the number of test cases that have passed, failed,and untested

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	4	0	0	4
Client Application	1	0	0	1
Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9

Final Report Output	4	0	0	4
Version Control	2	0	0	2

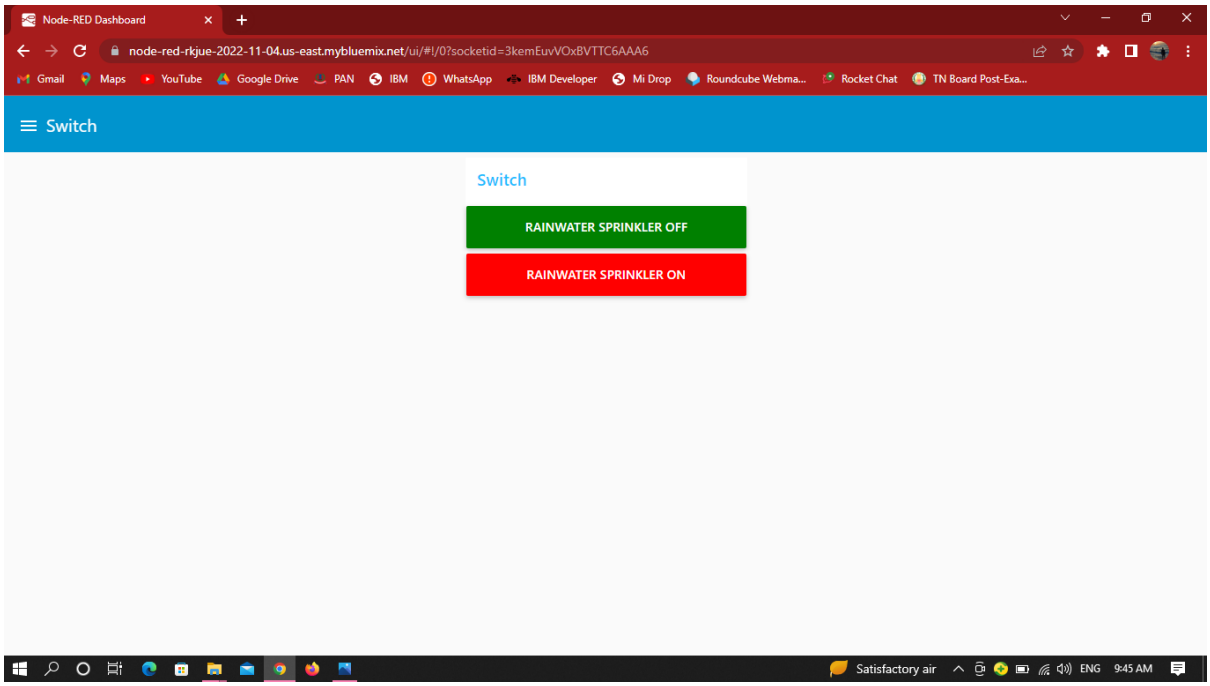
Web UI (App) :

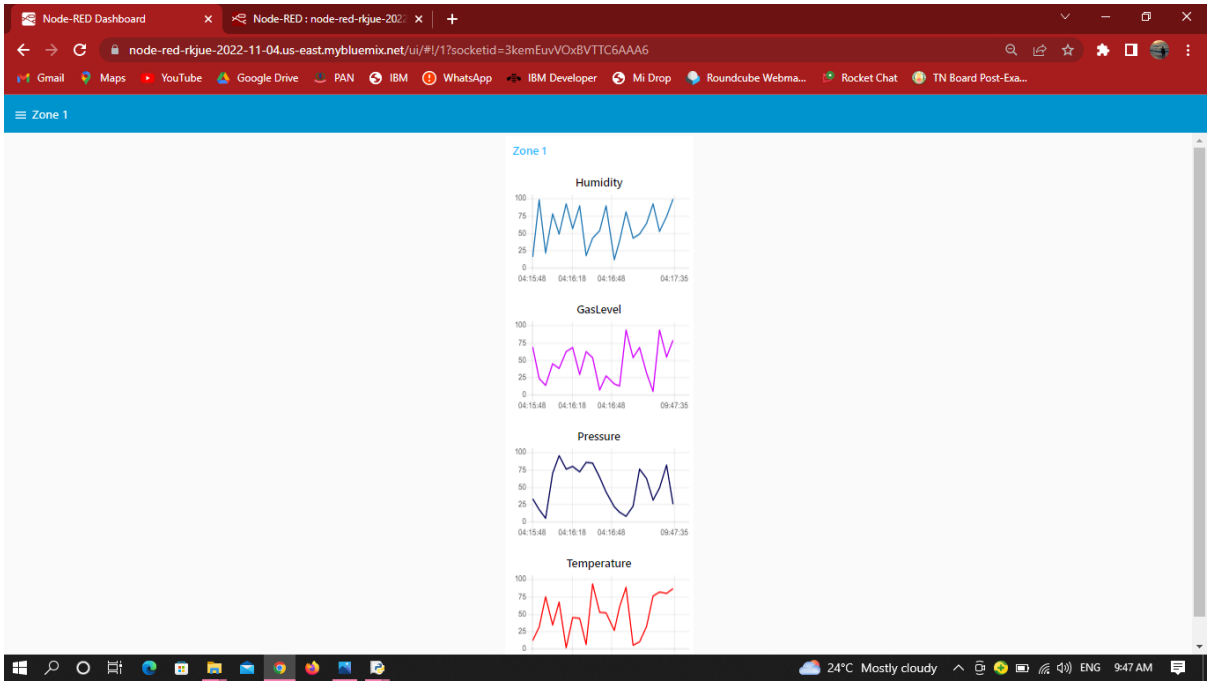


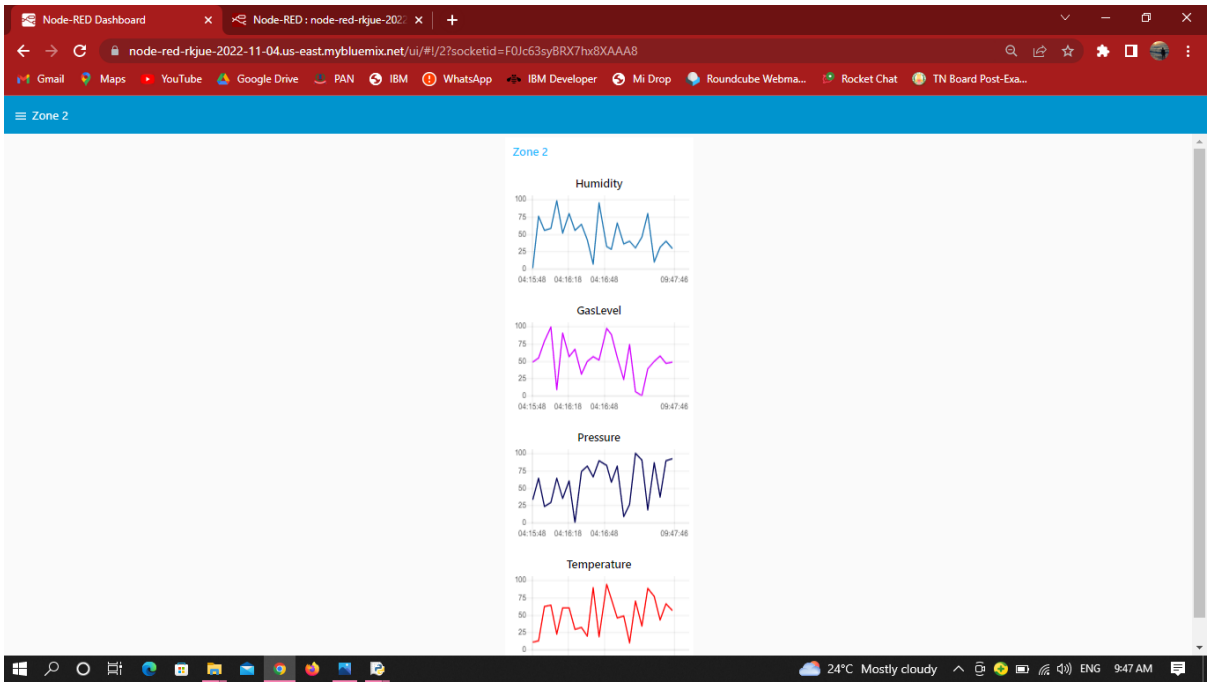


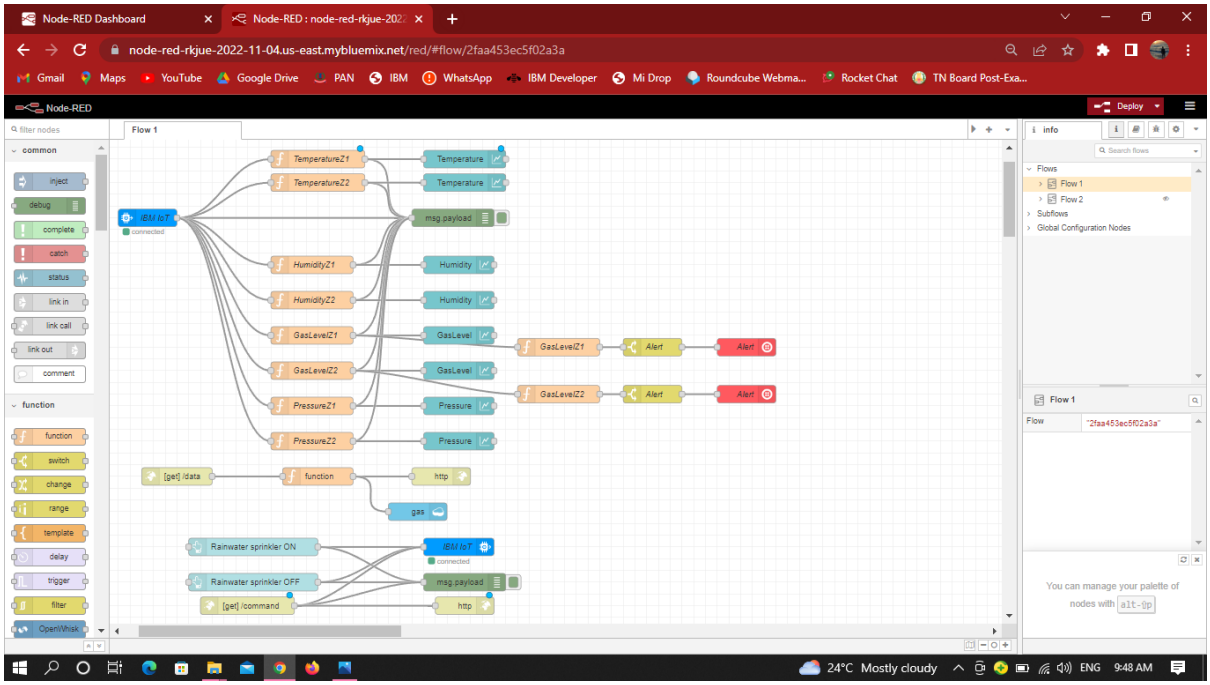
Apk File :

Link :
https://drive.google.com/file/d/1HCuYZdsHB2kt8RCHl9DpR3GtSLgxgZyl/view?usp=drivesdk


Node-RED Dashboard
node-red-rkjue-2022-11-04.us-east.mybluemix.net/ui/#!/0?socketid=3kemEuvVOxBVTTC6AAA6
Switch
Switch
RAINWATER SPRINKLER OFF
RAINWATER SPRINKLER ON

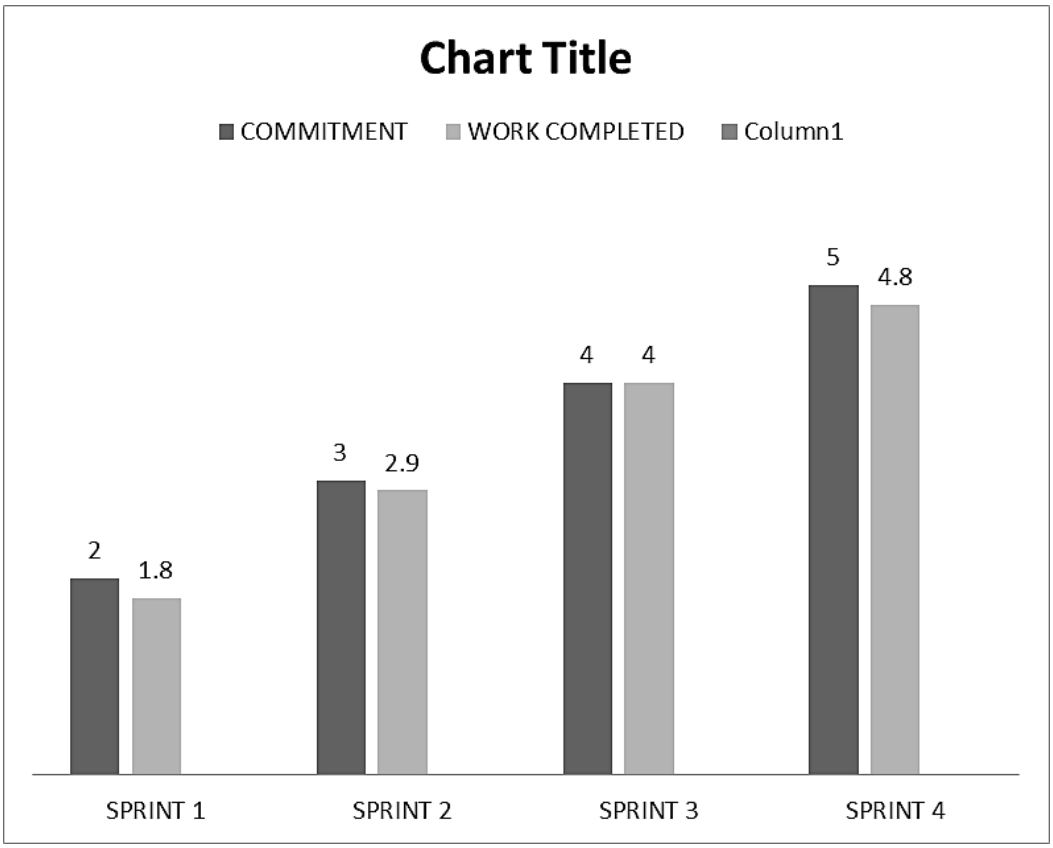

Node-RED Dashboard
Node-RED : node-red-rkjue-202
node-red-rkjue-2022-11-04.us-east.mybluemix.net/ui/#!/1?socketid=3kemEuvVOxBVTTC6AAA6
Zone 1
Zone 1
Humidity
GasLevel
Pressure
Temperature

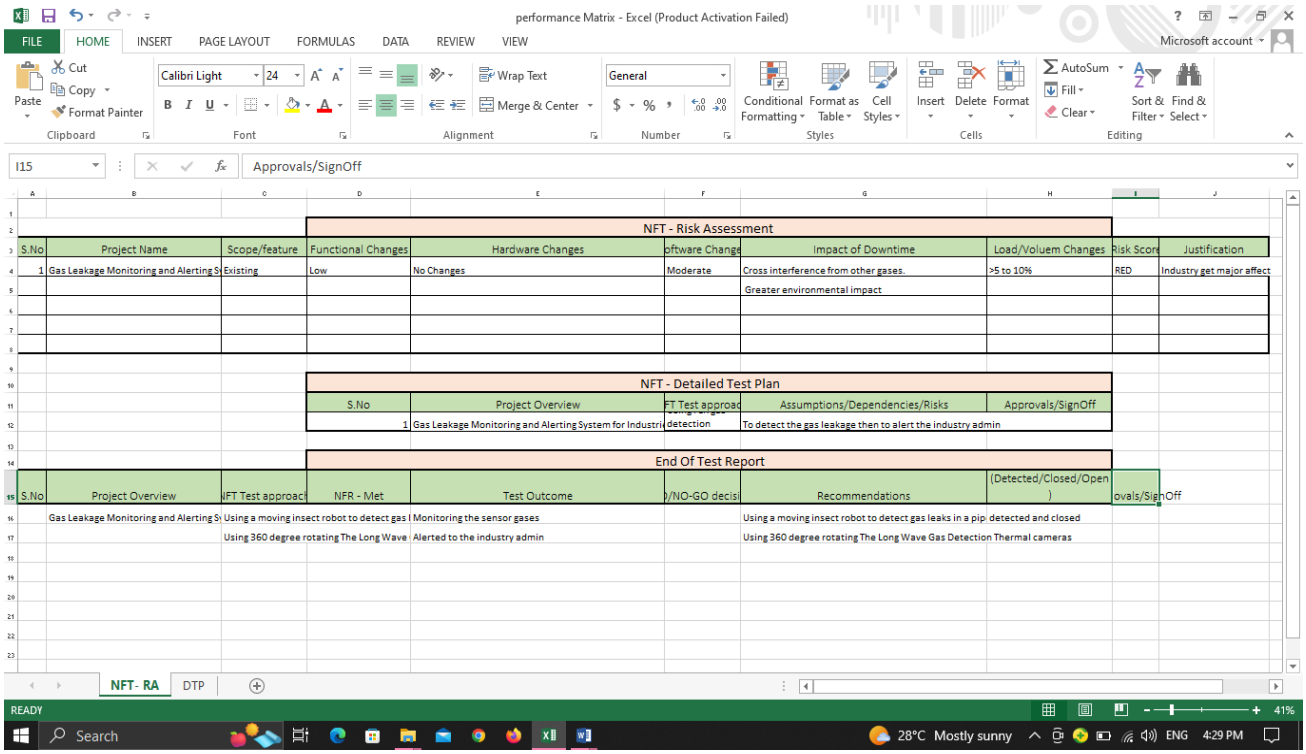

Node-RED Dashboard
node-red-rkjue-2022-11-04.us-east.mybluemix.net/ui/#!/2?socketid=F0Jc63syBRX7hx8XAAA8
Zone 2
Zone 2
Humidity
GasLevel
Pressure
Temperature
24°C Mostly cloudy
ENG
9:47 AM


Node-RED Dashboard
node-red-rkjue-2022-11-04.us-east.mybluemix.net/red/#flow/2faa453ec5f02a3a
Node-RED
Deploy
Flow 1
TemperatureZ1
TemperatureZ2
Temperature
msg.payload
HumidityZ1
HumidityZ2
Humidity
GasLevelZ1
GasLevelZ2
GasLevel
Alert
PressureZ1
PressureZ2
Pressure
[get] /data
function
http
gas
Rainwater sprinkler ON
Rainwater sprinkler OFF
[get] /command
IBM IoT
connected
24°C Mostly cloudy
ENG
9:48 AM

9. RESULTS

9.1 Performance Metrics :





Link :

https://github.com/IBM-EPBL/IBM-Project-35110-1660281675/blob/main/PROJECT%20DEVELOPMENT%20PHASE/performance%20Matrix.xlsx

10. ADVANTAGES & DISADVANTAGES

ADVANTAGES :

1. Helps prevent the high risk of gas explosions.
2. To detect combustible, flammable and toxic gases, and oxygen depletion.
3. Get real-time alerts about the gaseous presence in the atmosphere.
4. Prevent fire hazards and explosions.
5. Supervise gas concentration levels.
6. Ensure worker's health.
7. Real-time updates about leakages.
8. Data analytics for improved decisions.
9. Measure oxygen level accuracy.
10. Get immediate gas leak alerts.

DISADVANTAGES :

1. The instrument needs to be connected to the power supply to work.
2. Not easy to move.
3. The use of the environment (pressure, temperature and humidity) requires a higher.
4. Poor stability.
5. Greater environmental impact.
6. Cross interference from other gases.

11. CONCLUSION

Designing a fire and gas detection system is a significant challenge – the range of scenarios, detector positions, technologies and minimal industry guidance leave the whole process ill-defined, leading to poor potential detection performance in practice.

First and foremost our design should be informed by risk assessment, from which we can define the specific detection requirements, choose appropriate technology and select the best layout strategy. Performing these activities initially will yield the greatest risk reduction benefits and only once these are completed are we in a position to decide whether mapping is needed. Crucially, this pragmatic approach will be reflected in the forthcoming Energy Institute guidance on fire and gas detection design.

12. FUTURE SCOPE

- It could be including a Automatic Shut-off device which will turn off the gas supply whenever it will detect any gas leakage. This system can be implemented in Industries, Hotels and wherever the LPG cylinders are used.
- The gas leakage detection is not only highly accurate but cheap and portable and can be used for industrial and domestic safety.
- This project's future Scope is to use a HOG features for SVM classifier which is used to identify pipeline gas leaks and keep tabs on them.
- In addition, the system utilises an image processing technique to identify pipeline fractures.
- According to the suggested design, the robot capture the image down the pipe, looking for any signs of gas leakage by the Eddy Current method.
- When gas and smoke is detected then system will send short message service (SMS) to the user then user will take respective action.

- We mean to use Xively (new platform) to feed actual time sensor data on the internet. The sensor displays, distinguishes and advances an alarm whenever there is a gas outflow or fire penniless out condition is there.

13. APPENDIX

Source Code :

```
#IBM Watson IOT Platform
#pip install wiotp-sdk
import wiotp.sdk.device
import time
import random
myConfig = {
    "identity":
    {
        "orgId": "9yby55",
        "typeId": "Gas",
        "deviceId":"18"
    },
    "auth":
    {
        "token": "zIbdsvljWkP@1S34*&"
    }
}

def myCommandCallback(cmd):
    print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="sprinkleron":
        print (" Rainwater sprinkler is ON")
    elif status=="sprinkleroff":
        print (" Rainwater sprinkler is OFF")
    else:
        print ("please send proper command")


client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()
```

```
while True:
    TemperatureZ1=random.randint(0,100)
    HumidityZ1=random.randint(0,100)
    GasLevelZ1=random.randint(0,100)
    PressureZ1=random.randint(0,100)
    TemperatureZ2=random.randint(0,100)
    HumidityZ2=random.randint(0,100)
    GasLevelZ2=random.randint(0,100)
    PressureZ2=random.randint(0,100)

    myData={'TemperatureZ1':TemperatureZ1,'HumidityZ1':HumidityZ1,'GasLevelZ1':GasLevelZ1,
    'PressureZ1':PressureZ1,'TemperatureZ2':TemperatureZ2,'HumidityZ2':HumidityZ2,'GasLevelZ2':
    GasLevelZ2,'PressureZ2':PressureZ2}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
    print("Published data Successfully: %s", myData ,"to the IBM Platform")
    client.commandCallback = myCommandCallback
    time.sleep(5)
client.disconnect()
```

GitHub & Project Demo Link :

GitHub Link :

https://github.com/IBM-EPBL/IBM-Project-35110-1660281675

Project Demo Link :

https://drive.google.com/file/d/1H_ApUzlT2CMwlaMSNR_AmQnPjfvPp7cd/view?usp=drivesdk