Team ID	PNT2022TMID19555
Project Title	Gas leakage monitoring and alerting system for industries

GAS LEAKAGE MONITORING AND ALERTING SYSTEM FOR INDUSTRIES

1. **INTRODUCTION**
   - 1.1 Project Overview
   - 1.2 Purpose
2. **LITERATURE SURVEY**
   - 2.1 Existing problem
   - 2.2 References
   - 2.3 Problem Statement Definition
3. **IDEATION & PROPOSED SOLUTION**
   - 3.1 Empathy Map Canvas
   - 3.2 Ideation & Brainstorming
   - 3.3 Proposed Solution
   - 3.4 Problem Solution fit
4. **REQUIREMENT ANALYSIS**
   - 4.1 Functional requirement
   - 4.2 Non-Functional requirements
5. **PROJECT DESIGN**
   - 5.1 Data Flow Diagrams
   - 5.2 Solution & Technical Architecture
   - 5.3 User Stories
6. **PROJECT PLANNING & SCHEDULING**
   - 6.1 Sprint Planning & Estimation
   - 6.2 Sprint Delivery Schedule
   - 6.3 Reports from JIRA
7. **CODING & SOLUTIONING (Explain the features added in the project along with code)**
   - 7.1 Feature 1
   - 7.2 Feature 2
   - 7.3 Database Schema (if Applicable)
8. **TESTING**
   - 8.1 Test Cases
   - 8.2 User Acceptance Testing
9. **RESULTS**
   - 9.1 Performance Metrics
10. **ADVANTAGES & DISADVANTAGES**
11. **CONCLUSION**
12. **FUTURE SCOPE**
13. **APPENDIX**
    - Source Code
    - GitHub & Project Demo Link

1.INTRODUCTION

1.1 PROJECT OVERVIEW

Safety plays a major role in today's world and it is necessary that good safety systems are to be implemented in places of education and work. This work modifies the existing safety model installed in industries and this system also be used in homes and offices. Gas Leakages in open or closed areas can prove to be dangerous and lethal. Leakage in any kind of gas are concern in recent years,whether it is an residential area, a cafe, or a canteen or industrial sectors. The traditional Gas Leakage Monitoring Systems though have great precision, fail to acknowledge a few factors in the field of alerting the people about the leakage. Therefore we use latest technology like Internet of Things based Gas leakage monitoring, leakage detecting and alerting system is proposed. The moment gas leakage will probably be recognized, users will be informed via SMS through GSM module. The system will monitor the gas and display through LCD display. Whenever the leakage is detected the buzzer begins sound. Also store the data using cloud data storage. This will detect the harmful gases in environment and alerting to the user through alarm and sending notification.

1.2 PURPOSE

For safety purpose, we use this device. The industries use different types of gases so gas leakage occurs it affect the workers health issue and sometimes it is reason of fire accident.so we have to maintain the gas pipes and monitor the gas if monitor the gas by 24 hrs and leakage occur it alert the buzzer sound and send notification for responsible person.

2.LITERATURE SURVEY

2.1 EXISTING PROBLEM

S.No	TITLE	PROPOSED WORK	TOOLS USED/ ALGORITHM	TECHNOLOGY	ADVANTAGES/ DISADVANTAGES
1	A Mobile Gas Detector with an Arduino Microcontroller	It describes hardware synthesis for a mobile gas detector with an Arduino microcontroller. The main aim of this project is to find the dangerous zone where many stationary detectors are unpractical or too expensive. Thus, we present a self-propelled robotic gas detector.	• Arduino Mega 2560 MC, • L9110 motor driver, • 2Nos. DC motors, • Wi-Fi module ESP8266, • logic level converter, • MQ2 sensor, • HC-SR04 ultrasonic distance sensor	Internet of Things	**ADVANTAGES** • Arduino has some sound advantages, such as large user community, free and broad ranges of libraries of codes, relatively low cost components, and so forth. **DISADVANTAGES** • Its disadvantages are its small and a user has to work in a relatively small space. In many broad and multi-purpose projects, its required to look for third party sources in addition to Arduino scripts.

S.No	TITLE	PROPOSED WORK	TOOLS USED/ ALGORITHM	TECHNOLOGY	ADVANTAGES/ DISADVANTAGES
2	Gas Leakage Detection and Alert System using IoT	We design and develop an propose system which include some safety factors. A safety has been a major issue in today's day to day life.	• Regulator • Rectifier • LCD Display • Gas Sensor • Wi-fi Module • Arduino UNO	Internet of Things	**ADVANTAGES** • The sensor-enabled solution helps prevent the high risk of gas explosions and affecting any casualties within and outside the premises. **DISADVANTAGES** • It is affected due to ambient light interference.

S.NO	TITLE	PROPOSED WORK	TOOLS USED/ ALGORITHM	TECHNOLOGY	ADVANTAGES/ DISADVANTAGES
3	Gas Leakage Detector and Warning Generator	The main aim of this project is to find the dangerous zone where many stationary detectors are unpractical or too expensive. Thus, we present a self-propelled robotic gas detector. It describes hardware synthesis for a mobile gas detector with an Arduino microcontroller.	• Arduino Mega 2560 MC, • L9110 motor driver, • 2Nos. DC motors, • Wi-Fi module ESP8266, • logic level converter, • MQ2 sensor, • HC-SR04 ultrasonic distance sensor	Internet of Things	**ADVANTAGES** The sensor-enabled solution helps prevent the high risk of gas explosions and affecting any casualties within and outside the premises. The gas sensors help detect the concentration of the gases present in the atmosphere to avoid hazardous consequences like fire breakouts **DISADVANTAGES** • It has ability to detect wide range of gases.

2.2 REFERENCE

1. dekitan, A. I., Matthews, V. O., & Olasunkanmi, O. (2018). A microcontroller based gas leakage detection and evacuation system. IOP Conference Series: Materials Science and Engineering, 413, 012008. doi: 10.1088/1757-899x/413/1/012008
2. Shrivastava, A., Prabhaker, R., Kumar, R., & Verma, R. GSM based gas leakage detection system. International Journal of Emerging Trends in Electrical and Electronics (IJETEE-ISSN: 2320-9569), 2013; 3(2):42-45.
3. Priya, P. D., & Rao, C. T. Hazardous Gas Pipeline Leakage Detection Based on Wireless Technology. International Journal of Professional Engineering Studies, India, 2014; 2(1).

2.3 PROBLEM STATEMENT

Software Required:
Python IDLE

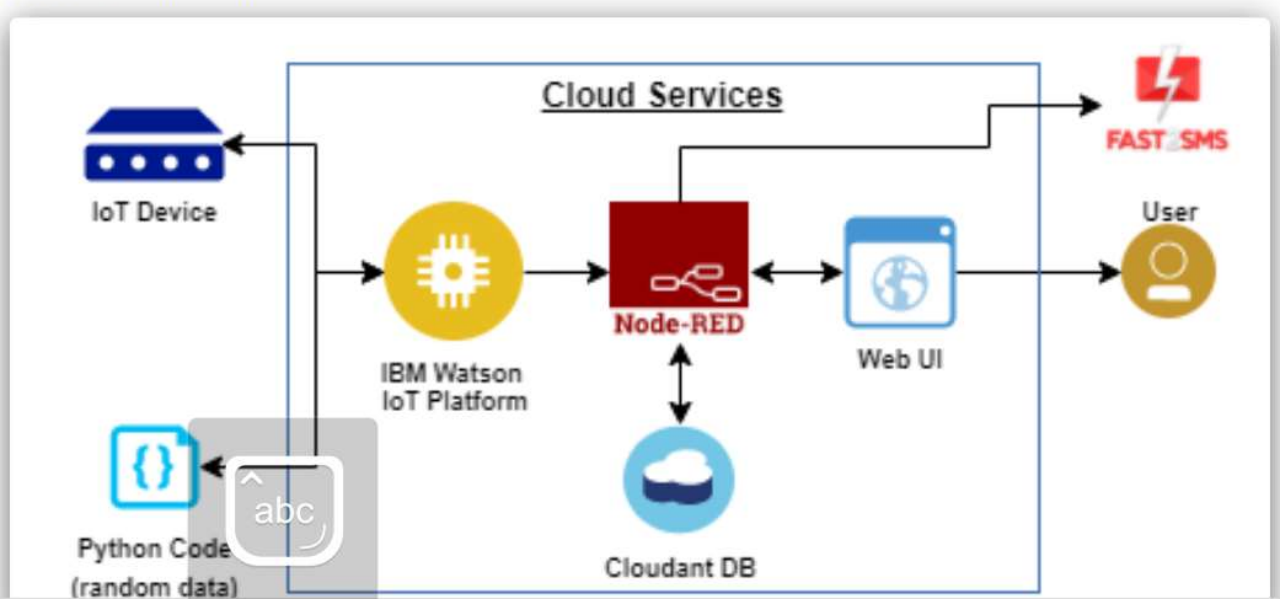
System Required:
RAM-Minimum 4GB Processor-Min. Configuration OS-Windows/Linux/MAC

Gas Leakage Monitoring & Alerting System for Industries has all the features as explained below

Features:

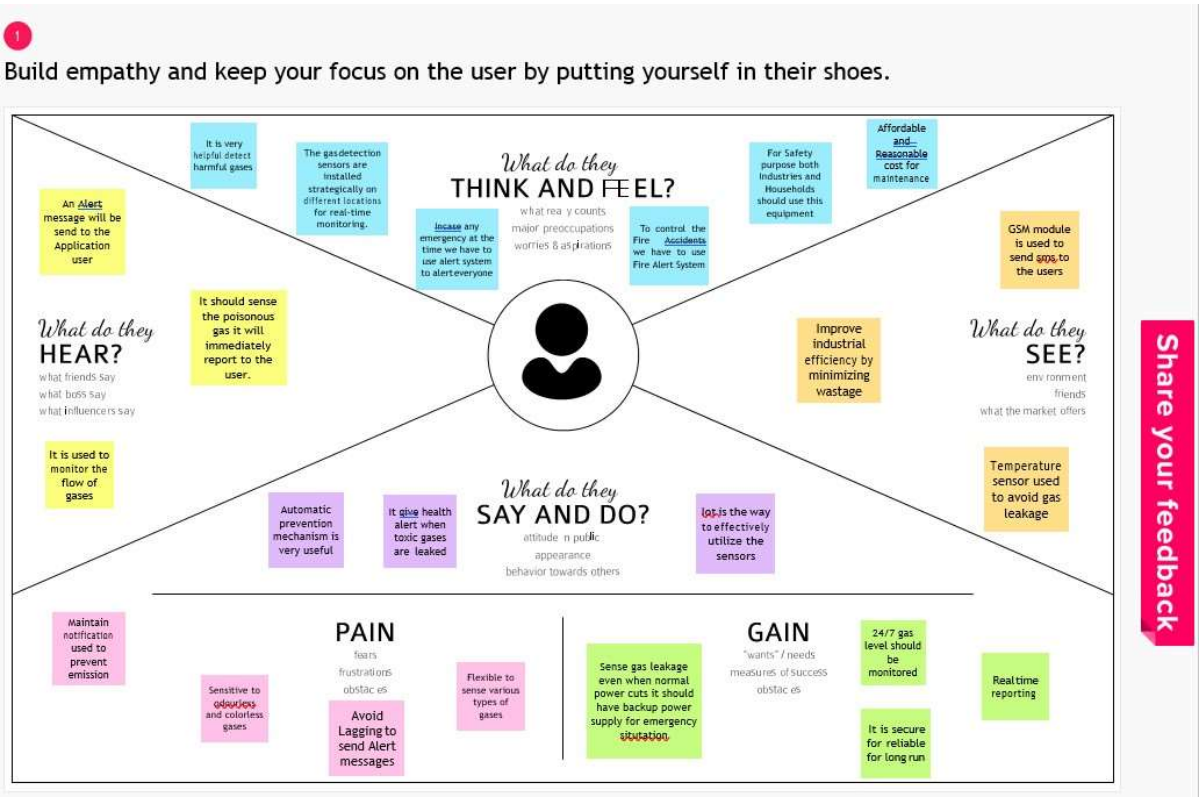
- This project helps the industries in monitoring the emission of harmful gases
- In several areas, the gas sensors will be integrated to monitor the gas leakage
- If in any area gas leakage is detected the admins will be notified along with the location
- In the web application, admins can view the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

Technical Architecture:

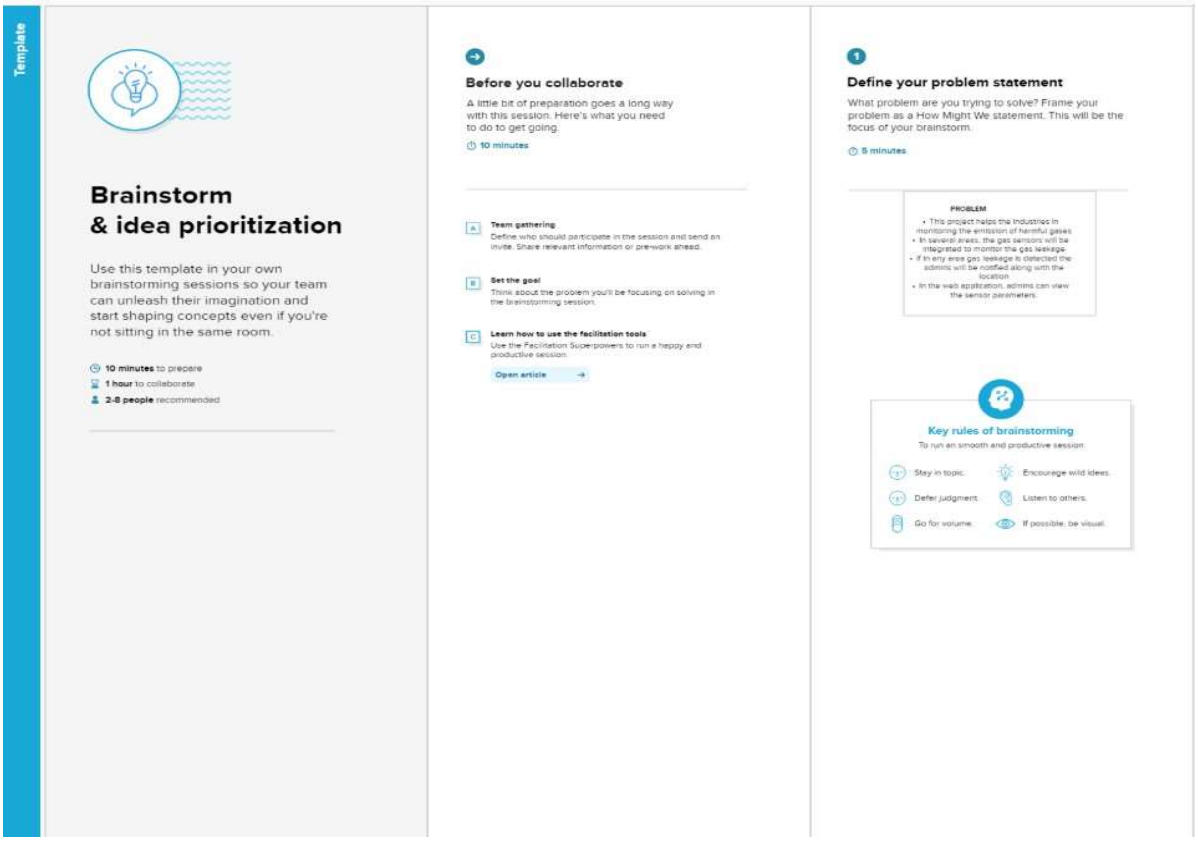


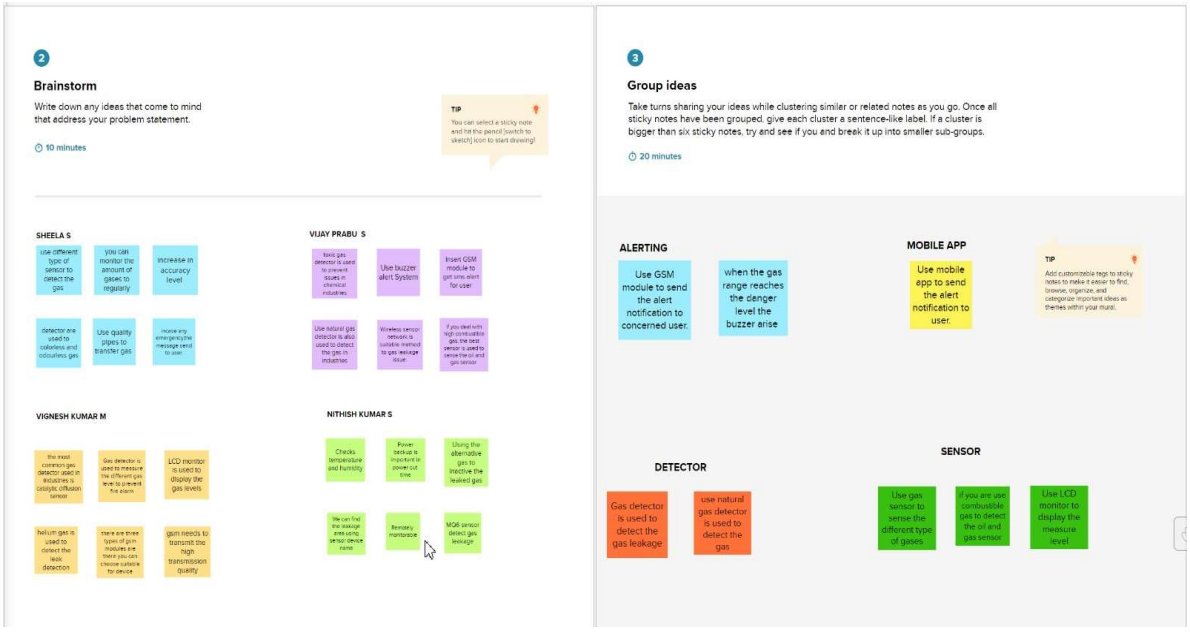
3.IDEATION & PROPOSED SOLUTION

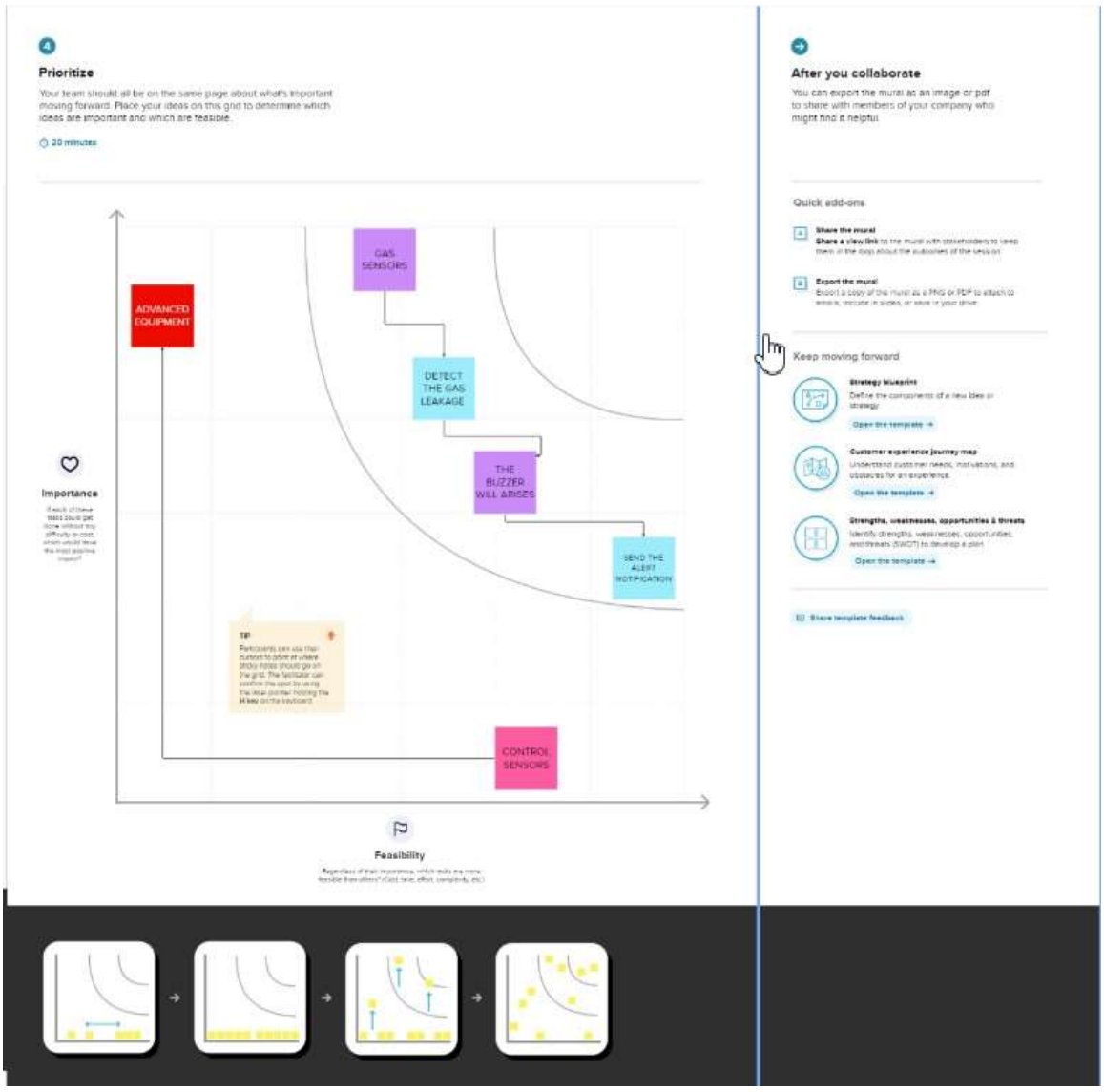
3.1 EMPATHY MAP CANVAS



3.2 IDEATION AND BRAINSTORMING





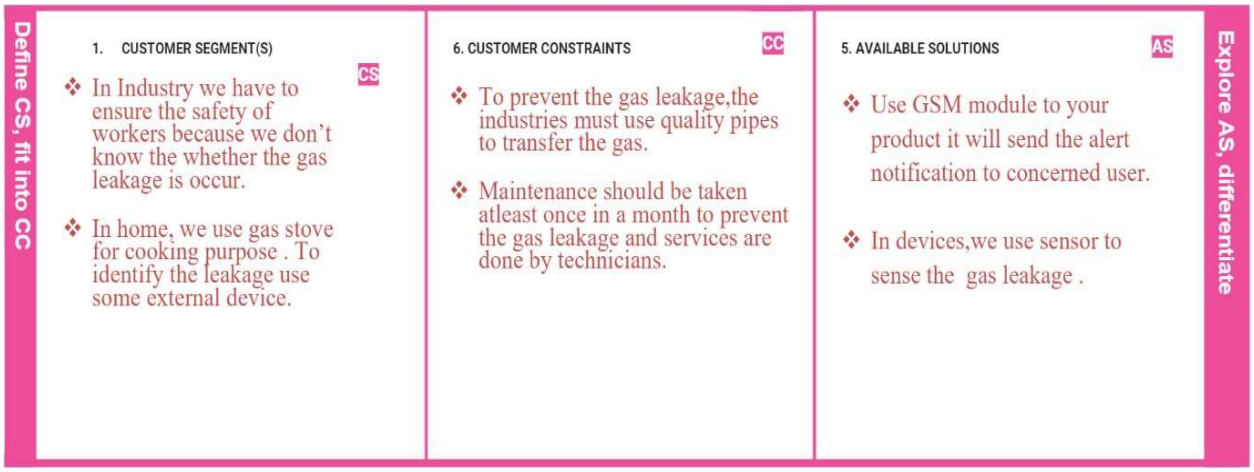

4
Prioritize
Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.
20 minutes
GAS SENSORS
ADVANCED EQUIPMENT
DETECT THE GAS LEAKAGE
THE BUZZER WILL ARISES
SEND THE ALERT NOTIFICATION
CONTROL SENSORS
Importance
Feasibility
TIP
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
Share the mural
Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.
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

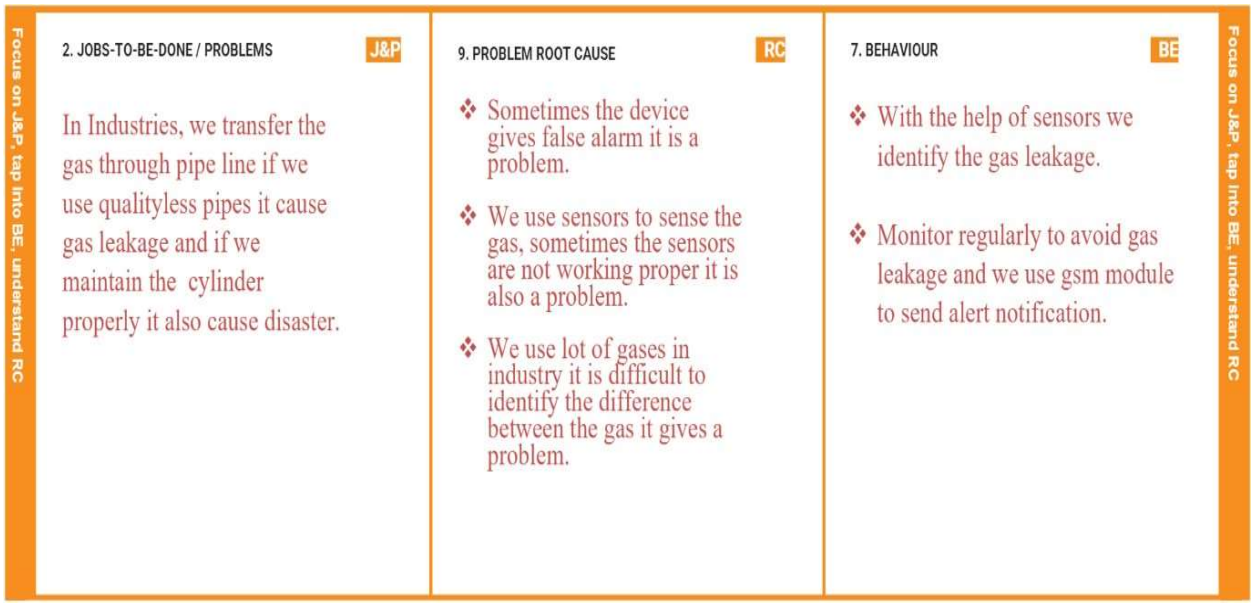
3.3 PROPOSED SOLUTION

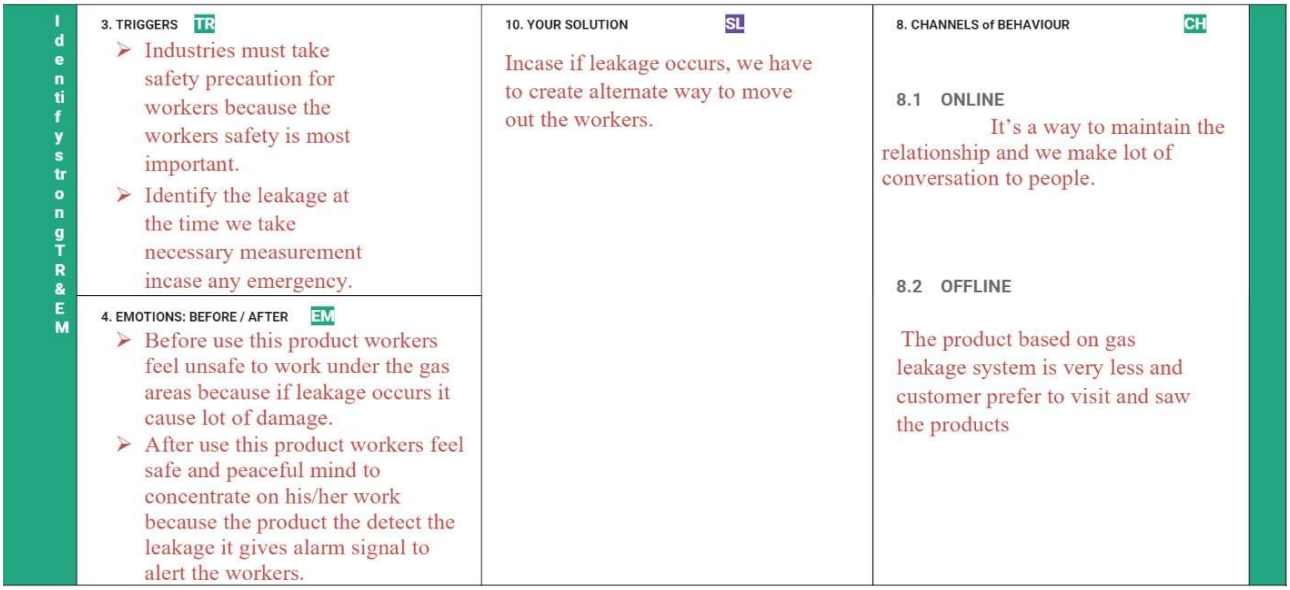
S.NO	PARAMETER	DESCRIPTION
1.	Problem Statement (Problem to be solved)	➢ It helps for monitoring the emission of harmful gases. ➢ If in any area gas leakage is detected the admins will be notified along with the Location and sent alert message to all workers. ➢ Admins can view the sensor Parameters.
2.	Idea / Solution description	➢ Smart bands are used to alert workers of gas leaks in factories. ➢ GSM technique is used to send alert messages to the respective persons and workers, if there is no response it sends the message to fire department. ➢ Using offline messaging alert facility alerting people without internet in range of upto 100 meters.
3.	Novelty / Uniqueness	➢ Visual and Audible alarms are triggered. ➢ It is power consumption and it is effective.
4.	Social Impact / Customer Satisfaction	➢ Get real-time alerts about the gaseous presence in the atmosphere ➢ Prevent fire hazards and explosions ➢ Supervise gas concentration levels ➢ Ensure worker's health using alert fit band. ➢ Cost-effective ➢ Get immediate gas leak alerts

5.	Business Model (Revenue Model)	➢ We can introduce product-based approach to earn a good revenue. The greater number of features attract the end users to use our application.
6.	Scalability of the Solution	➢ Our application can be used by the company which located in under the gas leakage area. We use IBM Watson cloud server to collect the live data the current data. we will ensure the safety of the workers. ➢ Not only triggering alarm it also can perform reactive operations like spreading a particular for making the leaked gas inactive.

3.4 PROBLEM SOLUTION FIT







4.REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

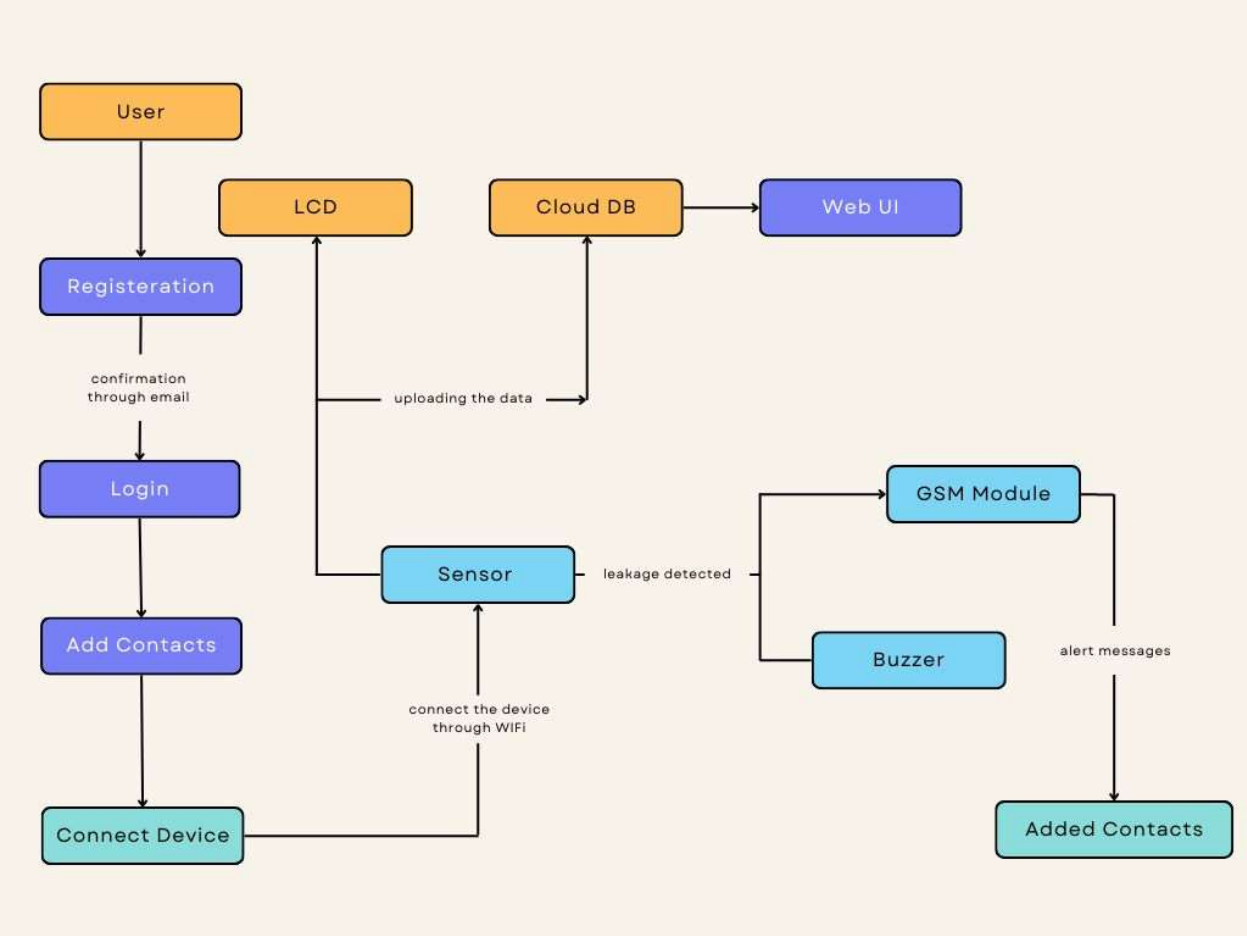
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	❖ Registration through Form ❖ Registration through Gmail
FR-2	User Confirmation	❖ Confirmation via Email ❖ Confirmation via OTP
FR-3	User SIGN IN	❖ User can sign in with username and password
FR-4	Connect the NODEMCU device	❖ Connect the NODEMCU device with application through Wifi.
FR-5	Update contact details	❖ Update the emergency contact number.
FR-6	REALTIME MONITORING	❖ It display the temperature level in pictorial representation using flow chart.
FR-6	Output	❖ In Emergency situation, it sends alert message to emergency contact number. ❖ It sends message to fire service.
FR-7	Review and Feedback	❖ User can share their experience about the app usage. ❖ Provide feedback

4.2 NON-FUNCTIONAL REQUIREMENTS

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	❖ In Industries, we use this device to avoid the fire accidents. ❖ The device can be accessed through Wifi.
NFR-2	**Security**	Only authorised person can access the important details.
NFR-3	**Reliability**	❖ Prevent from accidents. ❖ Avoid false Alarm. ❖ It Should avoid the delay alert message.
NFR-4	**Performance**	In a gas usage industries, the gas is transferred from one end to another end through pipes at the time gas leakage is possible. We use this device to detect the gas leakage and gives buzzer alarm and send alert message to nearest police station and fire service department.
NFR-5	**Availability**	This model is used to continuously monitor and display the gas level and check the gas level is exceed or not.
NFR-6	**Scalability**	Lot of users can access the application at the same time without any inconvenience.

5.PROJECT DESIGN

5.1 DATA FLOW DIAGRAM



5.2 SOLUTION AND TECHNICAL ARCHITECTURE

Table-1: Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript
2.	IOT Application Logic-1	Device Should be connected to System Using python code.	Python
3.	IOT Application Logic-2	NODEMCU device is connected with IBM Watson Platform.	IBM Watson STT service
4.	IOT Application Logic-3	It is connected with Web UI and alert the user through messages.	IBM Watson Assistant

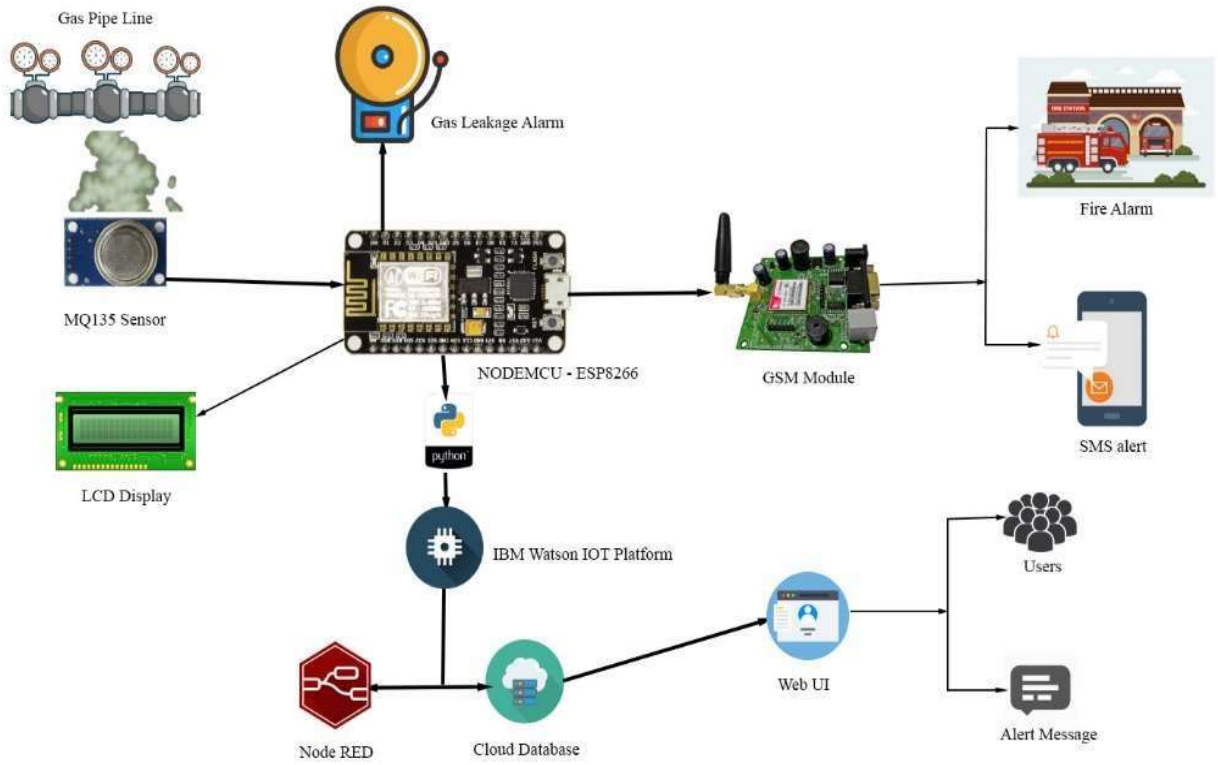
5.	Cloud Database	Database Service on Cloud. Data can be any format based on user.	IBM DB2, IBM Cloudant etc.
6.	File Storage	File Should be named and it contains details of file type, date and time of file is created, memory space.	IBM Block Storage or Other Storage Service or Local Filesystem
7.	External API-1	It is used in device through Wifi communicating and allotting operation efficiency.	Aadhar API, etc.
8.	Machine Learning Model	IOT and machine learning delivers insights otherwise hidden in data for rapid automated response and improved decision making	Object Recognition Model and Danger prediction model etc.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Without using framework,we lot of manual work to do the task, but using framework provide rapid development and in coding part it reduce the code.	Real time monitoring-Gas leakage is monitored continuously using IOT Zeta.
2.	Security Implementations	GSM module is enabled with devices.If leakage is detected send the alert message to concerned user.	e.g. SHA-256-It is used for encryption security.
3.	Scalable Architecture	Any problem occur, the concerned user can see the problem and real time monitoring the gas levels.	Multiple Data store Technologies , Reliable.
4.	Availability	It continuously monitor the gas level and whether the gas leakage occurs it immediatedly sends the SMS alert to user	GSM Module
5.	Performance	The leakage detected without any delay it gives fire alarm and it sends SMS alert to user. WEB UI Is also connected with device so it alert the concerned user. We use high quality pipes and	❖ High durable devices with good quality. ❖ Long Lasting Battery

S.No	Characteristics	Description	Technology
		regular maintanence to avoid the leakage.	

SOLUTION ARCHITECTURE:



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 register for the application by entering my email, password, and confirming my password.	I can access my account	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email	High	Sprint-1
	Login	USN-3	As a user, I can log into the application entering email & password	I can login to my web to my email and password.	High	Sprint-1
	Dashboard	USN-4	As a user, I can login into my account and I can add the members to send alert message	Only added contact numbers can receive SMS alerts.	High	Sprint-1
Customer (Web user)	Registration	USN-1	As a user, I can register for to creating account by entering my email, password, and confirming my password.	I can access my account.	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the web user.	I can receive confirmation email.	High	Sprint-1
	Login	USN-3	As a user, I can log into the website entering email & password	Only valid credentials must be acceptable.	High	Sprint-1
	Dashboard	USN-4	As a user, I can login into my	Only added contact numbers	High	Sprint-1

USER STORIES:

USER TYPE	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
			account and I can add the members to send alert message	can receive sms alerts.		
Administrator	Admin	USN-1	As a Admin, He/ She can add the members those who want to receive alert message.	The Admin have authority can add members and edit the details.	High	Sprint-1
		USN-2	As a Admin, He/ She can monitor real time.	The admin can monitor the process by 24/7 hrs.	High	Sprint-1

6.PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION

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	10	6 Days	24 Oct 2022	29 Oct 2022	10	29 Oct 2022
Sprint-2	12	6 Days	31 Oct 2022	05 Nov 2022	12	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	10	6 Days	14 Nov 2022	19 Nov 2022	10	19 Nov 2022

Velocity:

Imagine we have a 6-day sprint duration, and the velocity of the team is 13(approx) (points per sprint). Let's calculate the team's average velocity(AV) per iteration unit(story points per day)

AV = Sprint duration/Velocity = 13/6=2.2

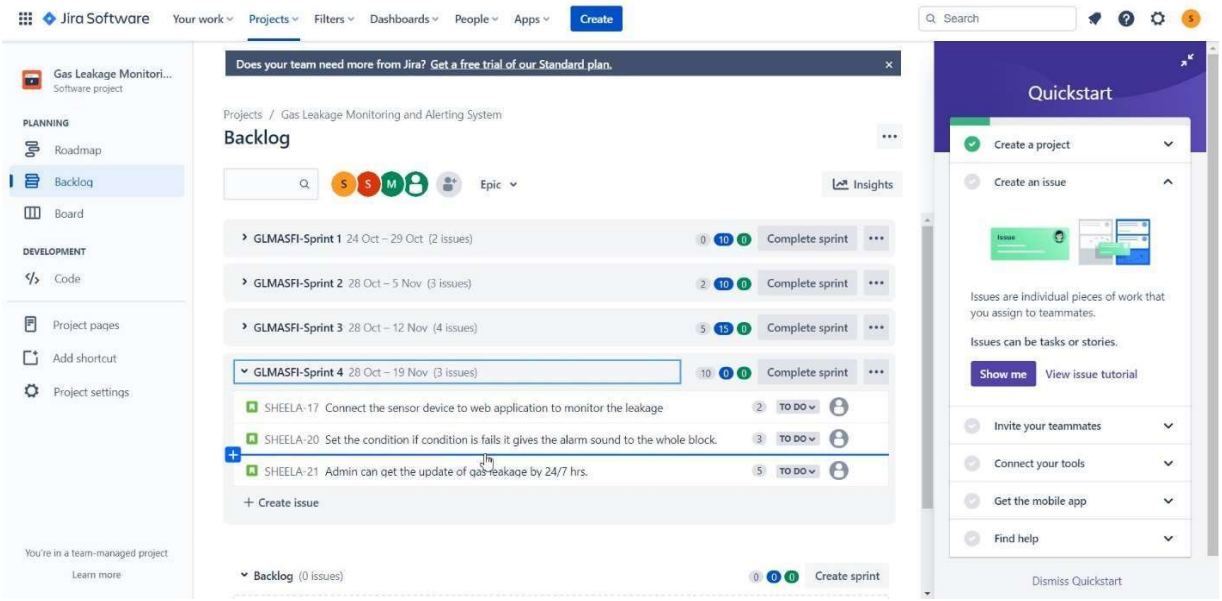
6.2 SPRINT DELIVERY SCHEDULE

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration (Mobile user)	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	5	High	Sheela S
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	5	Medium	Vijay Prabu S
Sprint-2	Login (Mobile User)	USN-3	As a user, I can log into the application entering email & password	5	Low	Vignesh Kumar M
Sprint-2		USN-4	Connect the sensor device to Mobile application to monitor the gas leakage	2	High	Nithish Kumar S
Sprint-2	Dashboard (Mobile User)	USN-5	As a user, I can login into my account and I can add the members to send alert message	5	High	Sheela S
Sprint-3	Registration (Web User)	USN-6	As a user, I can register for to creating account by entering my email, password, and confirming my password.	5	High	Vignesh Kumar M
Sprint-3		USN-7	As a user, I will receive confirmation email once	5	High	Nithish Kumar S

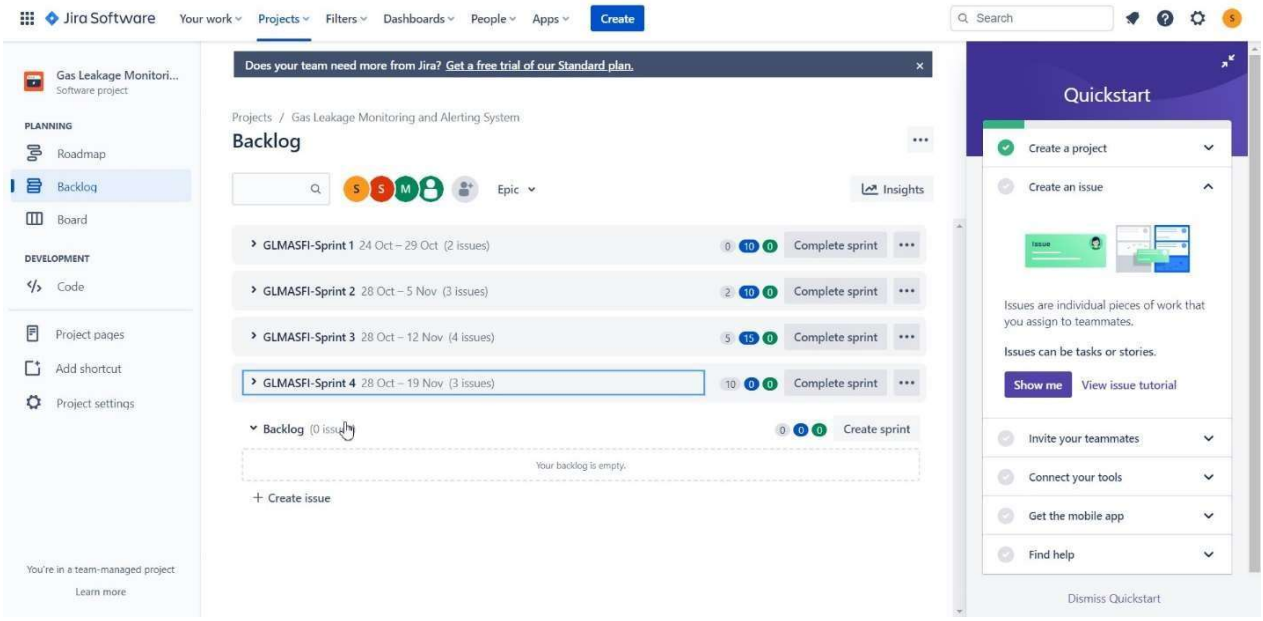
			I have registered for the web user.			
Sprint-3	Login(Web User)	USN-8	As a user, I can log into the website entering email & password	5	Medium	Vijay Prabu S

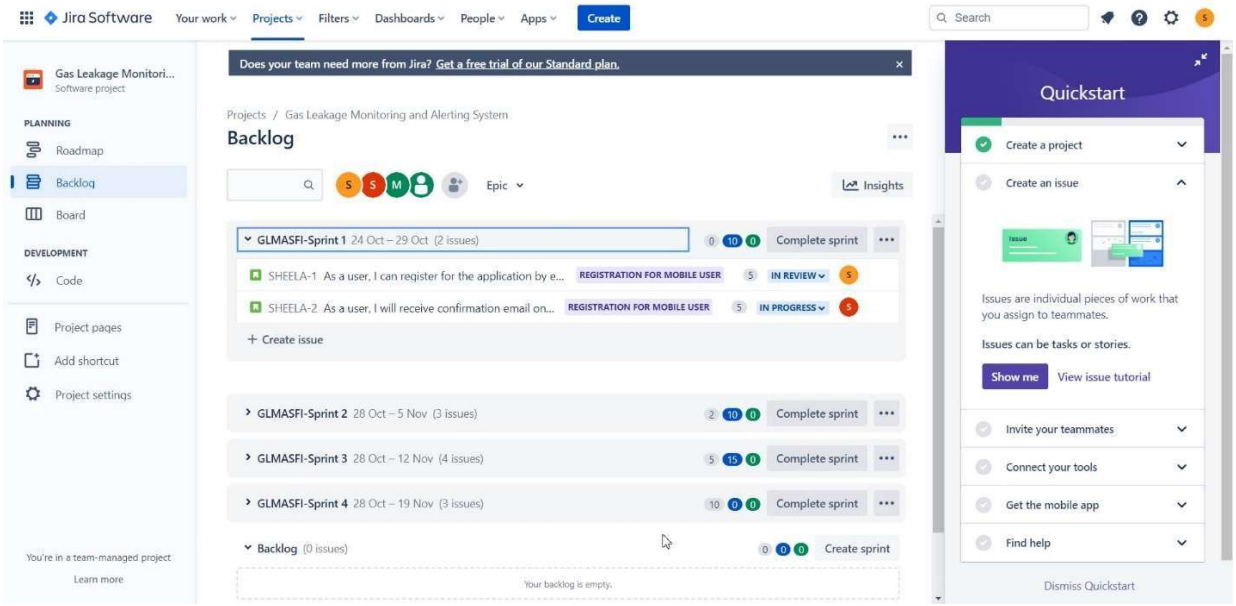
6.3 REPORTS FROM JIRA

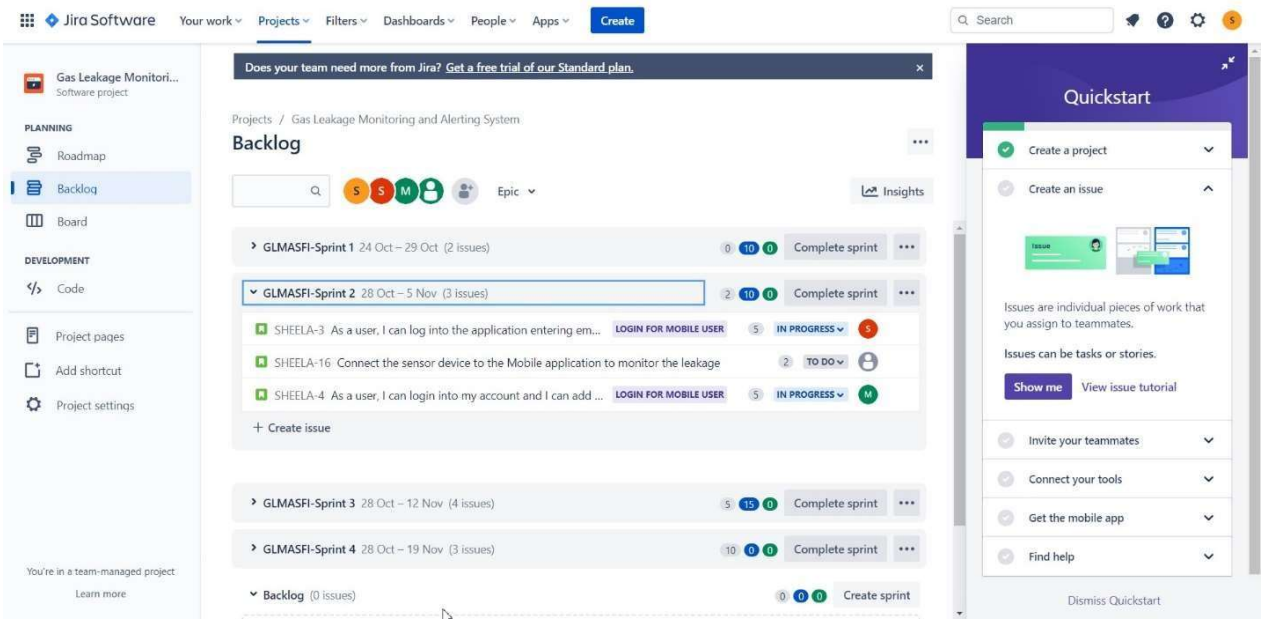
1. Image showing the backlogs create in the JIRA Software (PETA Spring1, PETASpring2, PETA Spring3, PETA Spring4)

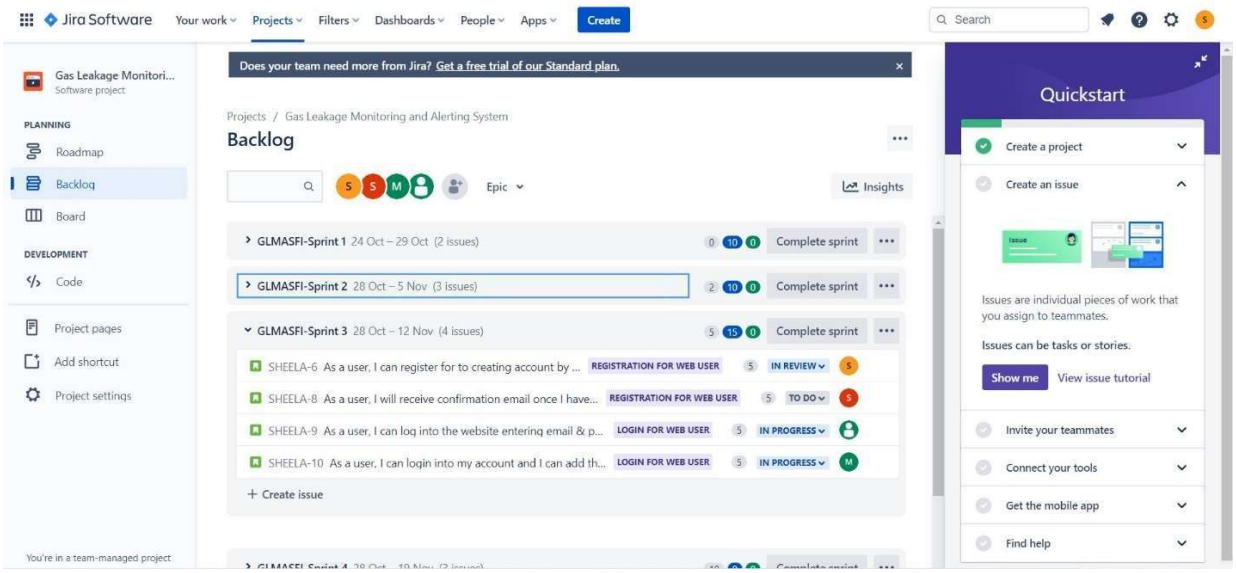


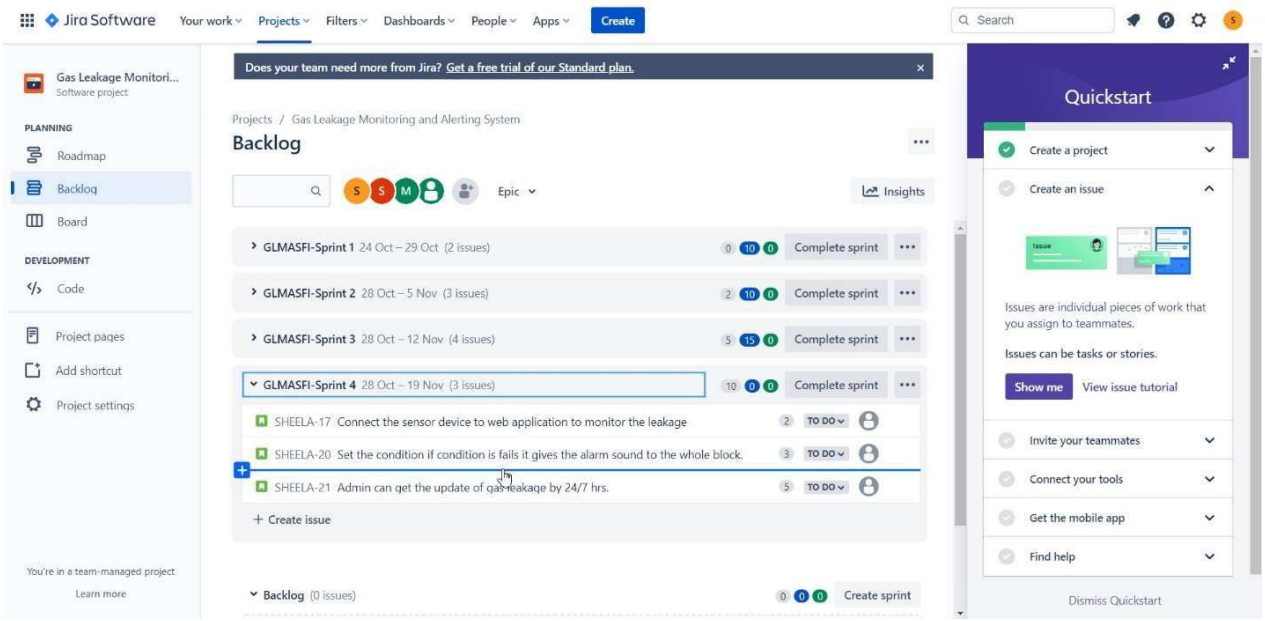
2. Image Showing the User Stories in the respective sprints.






Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
Gas Leakage Monitori...
Software project
PLANNING
Roadmap
Backlog
Board
DEVELOPMENT
Code
Project pages
Add shortcut
Project settings
You're in a team-managed project
Learn more
Does your team need more from Jira? Get a free trial of our Standard plan.
Projects / Gas Leakage Monitoring and Alerting System
Backlog
Epic
Insights
GLMASFI-Sprint 1 24 Oct – 29 Oct (2 issues)
Complete sprint
GLMASFI-Sprint 2 28 Oct – 5 Nov (3 issues)
SHEELA-3 As a user, I can log into the application entering em... LOGIN FOR MOBILE USER IN PROGRESS
SHEELA-16 Connect the sensor device to the Mobile application to monitor the leakage TO DO
SHEELA-4 As a user, I can login into my account and I can add ... LOGIN FOR MOBILE USER IN PROGRESS
Create issue
GLMASFI-Sprint 3 28 Oct – 12 Nov (4 issues)
GLMASFI-Sprint 4 28 Oct – 19 Nov (3 issues)
Backlog (0 issues)
Create sprint
Quickstart
Create a project
Create an issue
Issues are individual pieces of work that you assign to teammates.
Issues can be tasks or stories.
Show me
View issue tutorial
Invite your teammates
Connect your tools
Get the mobile app
Find help
Dismiss Quickstart


Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
Gas Leakage Monitori...
Software project
PLANNING
Roadmap
Backlog
Board
DEVELOPMENT
Code
Project pages
Add shortcut
Project settings
You're in a team-managed project
Does your team need more from Jira? Get a free trial of our Standard plan.
Projects / Gas Leakage Monitoring and Alerting System
Backlog
Epic
Insights
GLMASFI-Sprint 1 24 Oct – 29 Oct (2 issues)
Complete sprint
GLMASFI-Sprint 2 28 Oct – 5 Nov (3 issues)
GLMASFI-Sprint 3 28 Oct – 12 Nov (4 issues)
SHEELA-6 As a user, I can register for to creating account by ... REGISTRATION FOR WEB USER IN REVIEW
SHEELA-8 As a user, I will receive confirmation email once I have... REGISTRATION FOR WEB USER TO DO
SHEELA-9 As a user, I can log into the website entering email & p... LOGIN FOR WEB USER IN PROGRESS
SHEELA-10 As a user, I can login into my account and I can add th... LOGIN FOR WEB USER IN PROGRESS
Create issue
Quickstart
Create a project
Create an issue
Issues are individual pieces of work that you assign to teammates.
Issues can be tasks or stories.
Show me
View issue tutorial
Invite your teammates
Connect your tools
Get the mobile app
Find help



3. Image showing the workspace progress in the sprint

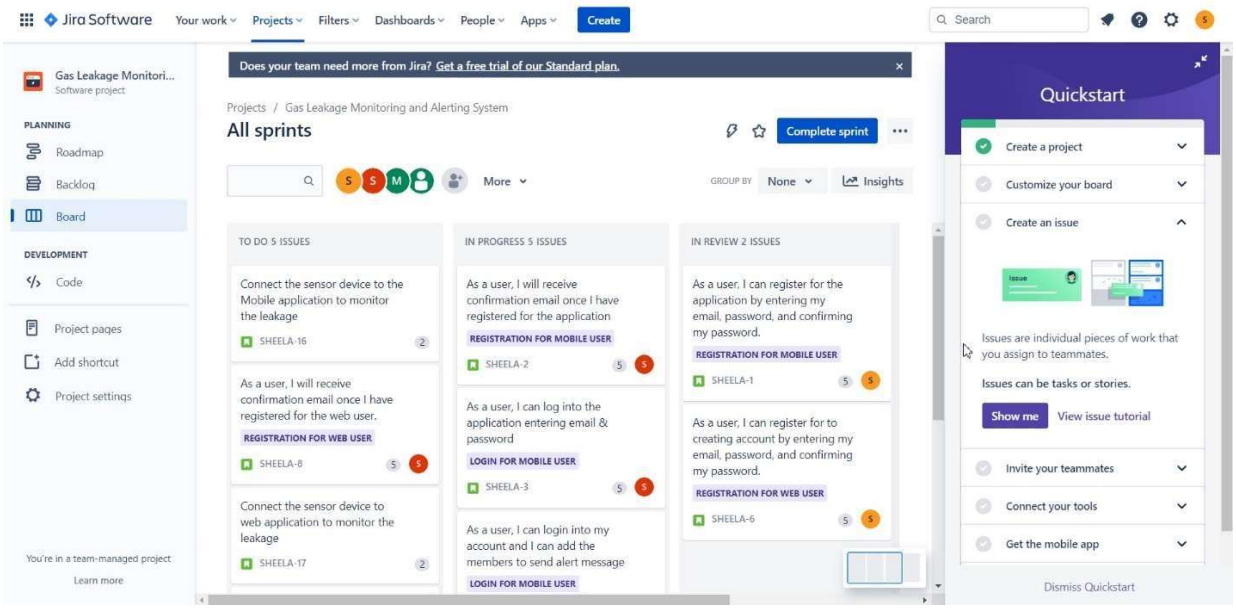
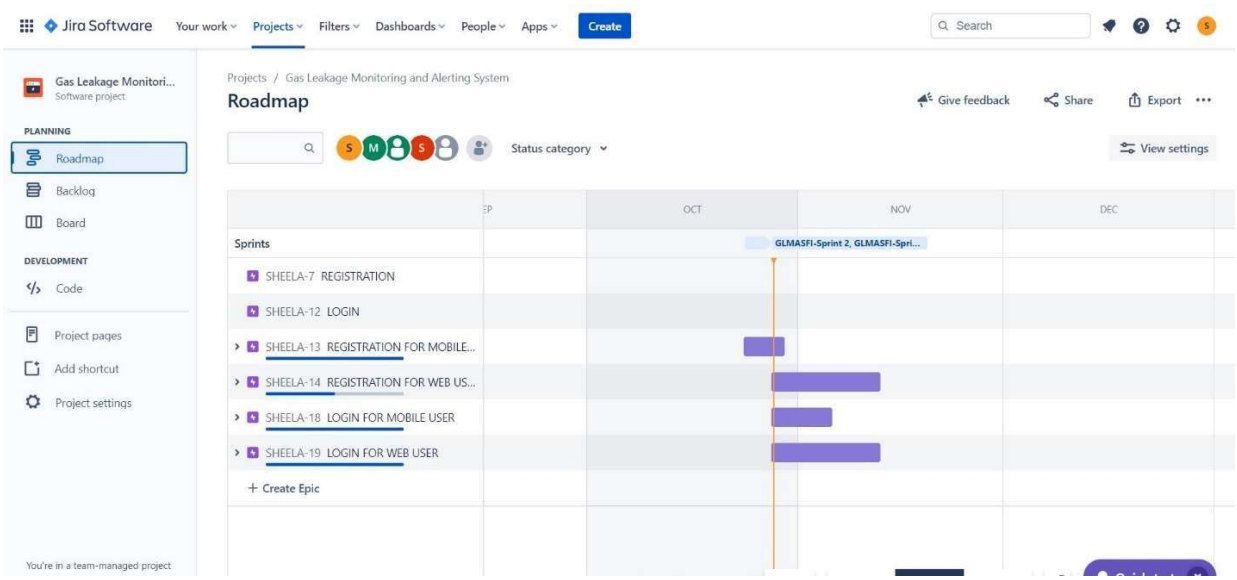
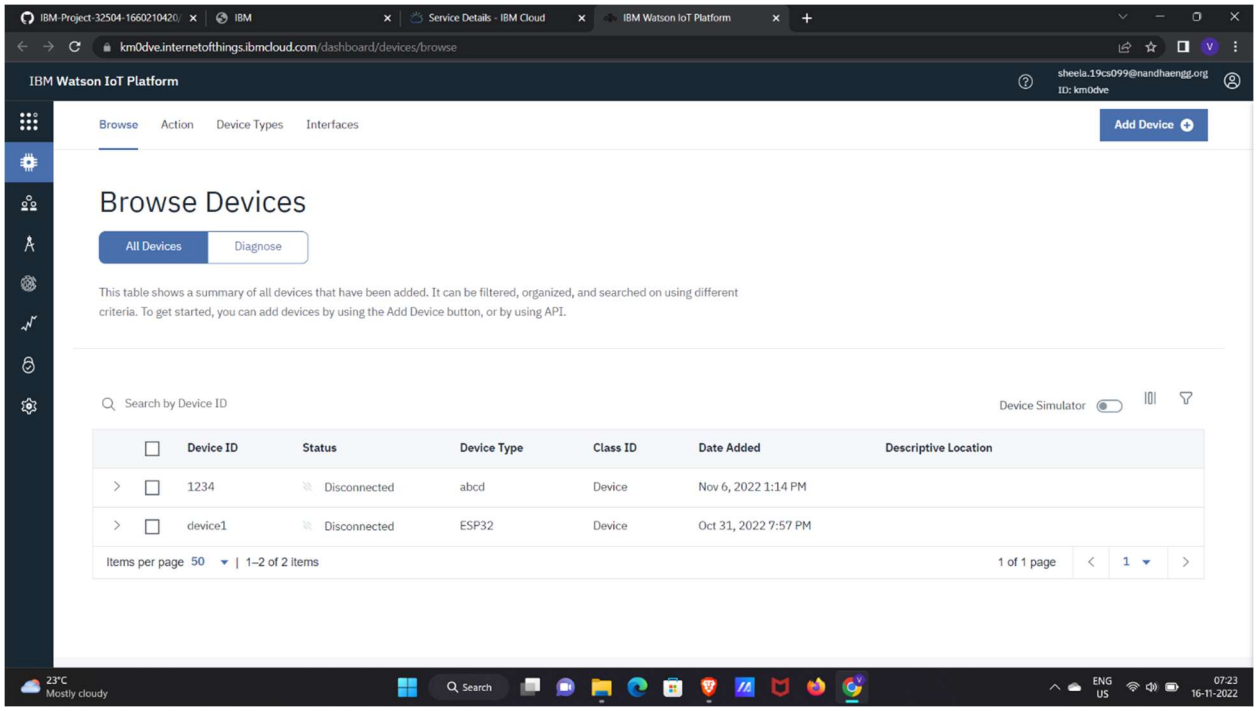


Image Showing the RoadMap of Sprint

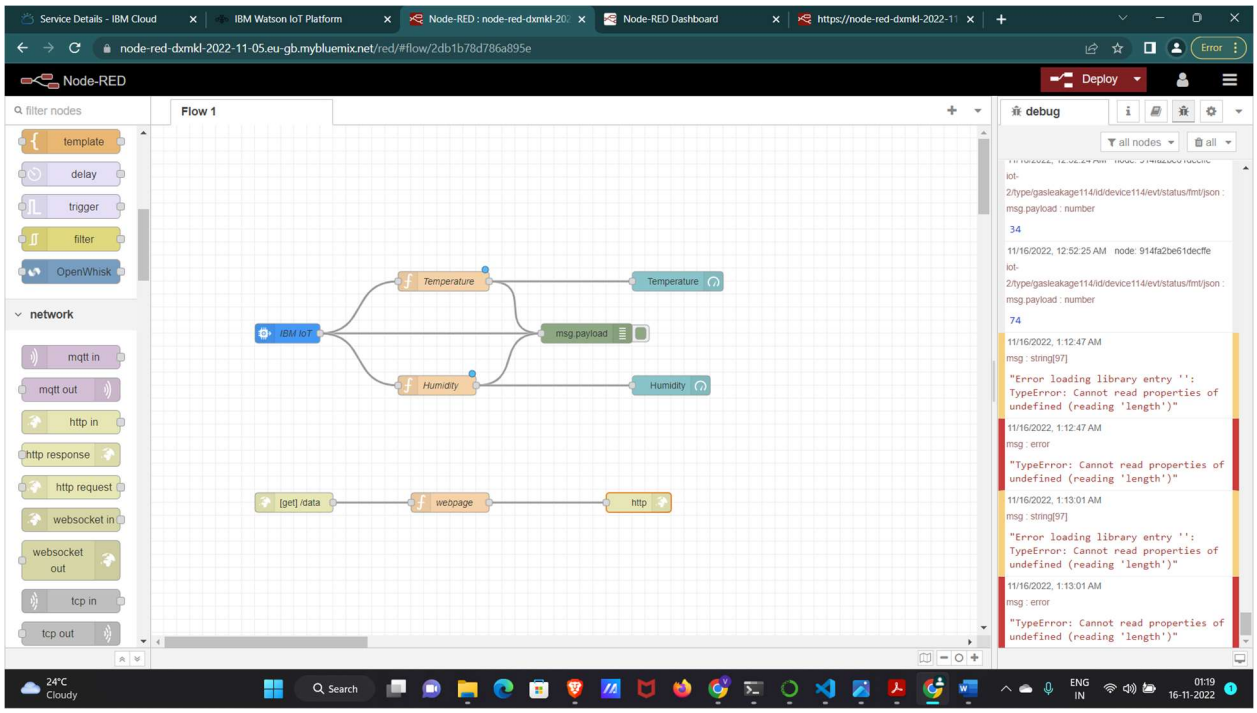


7.SOLUTIONING

7.1 FEATURE 1



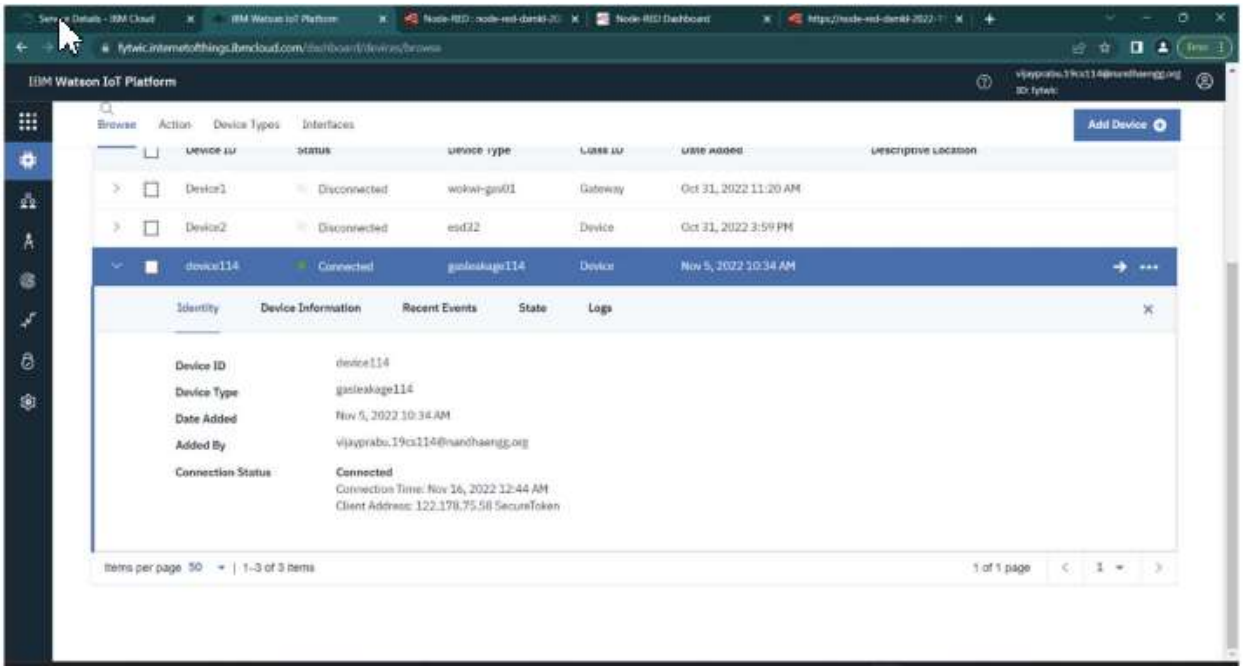
7.2 FEATURE 2

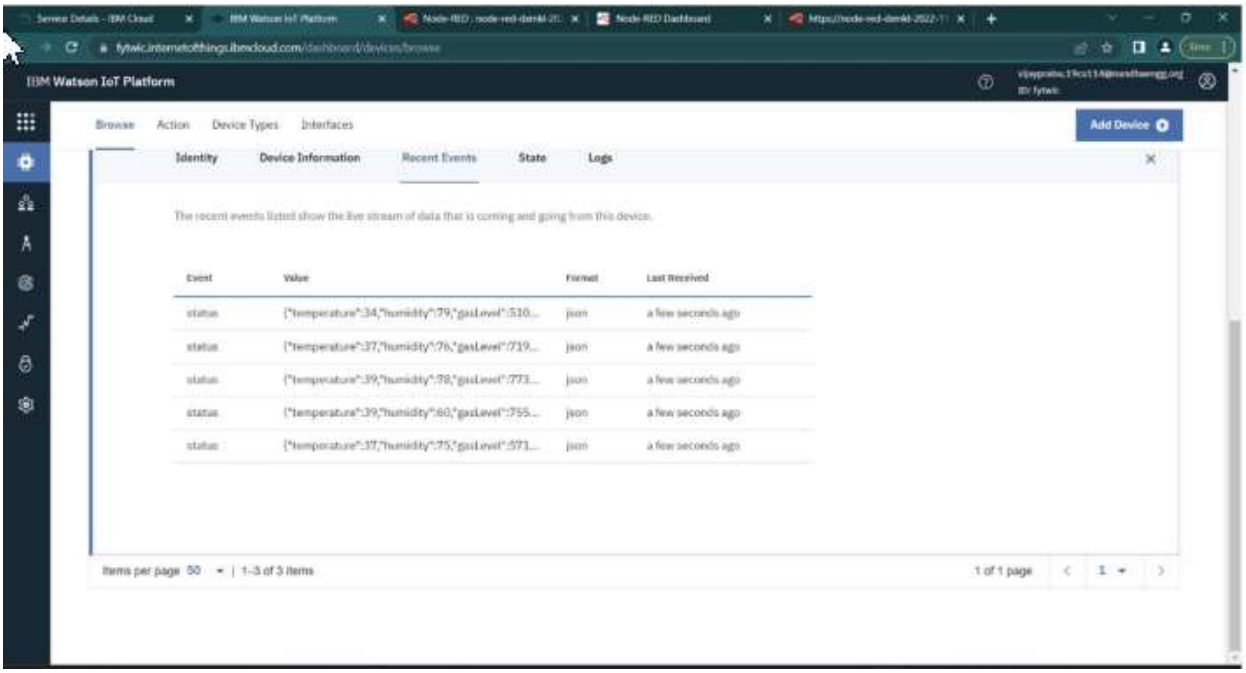


8.TESTING

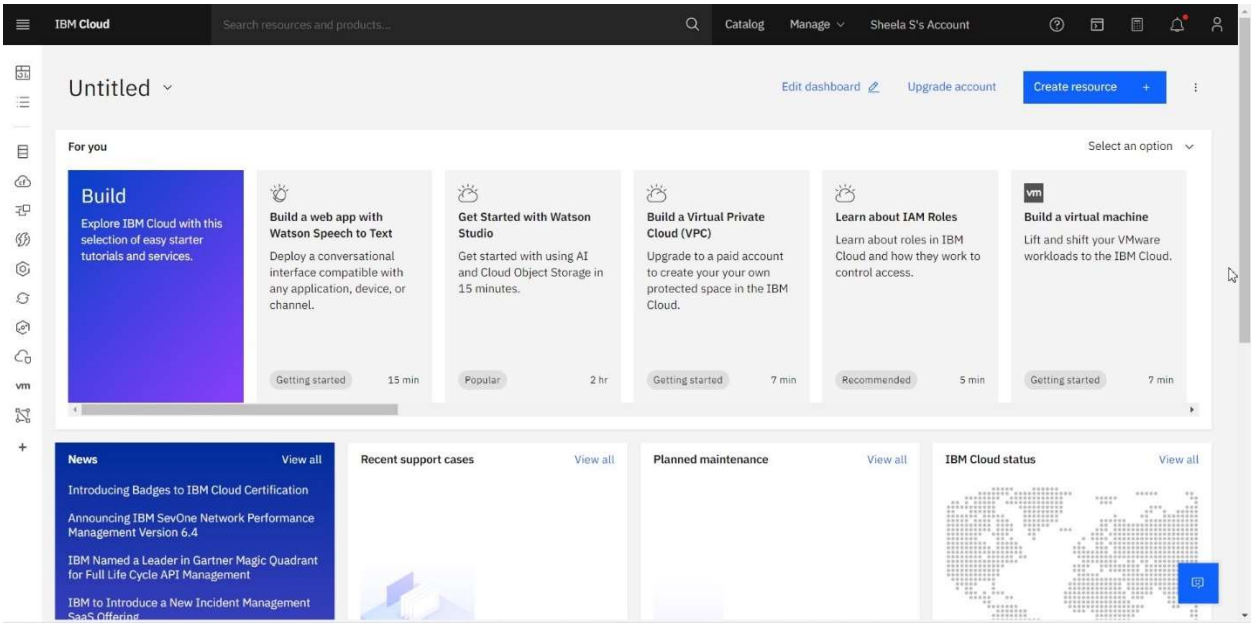
8.1 TEST CASES

Publish data to IBM cloud

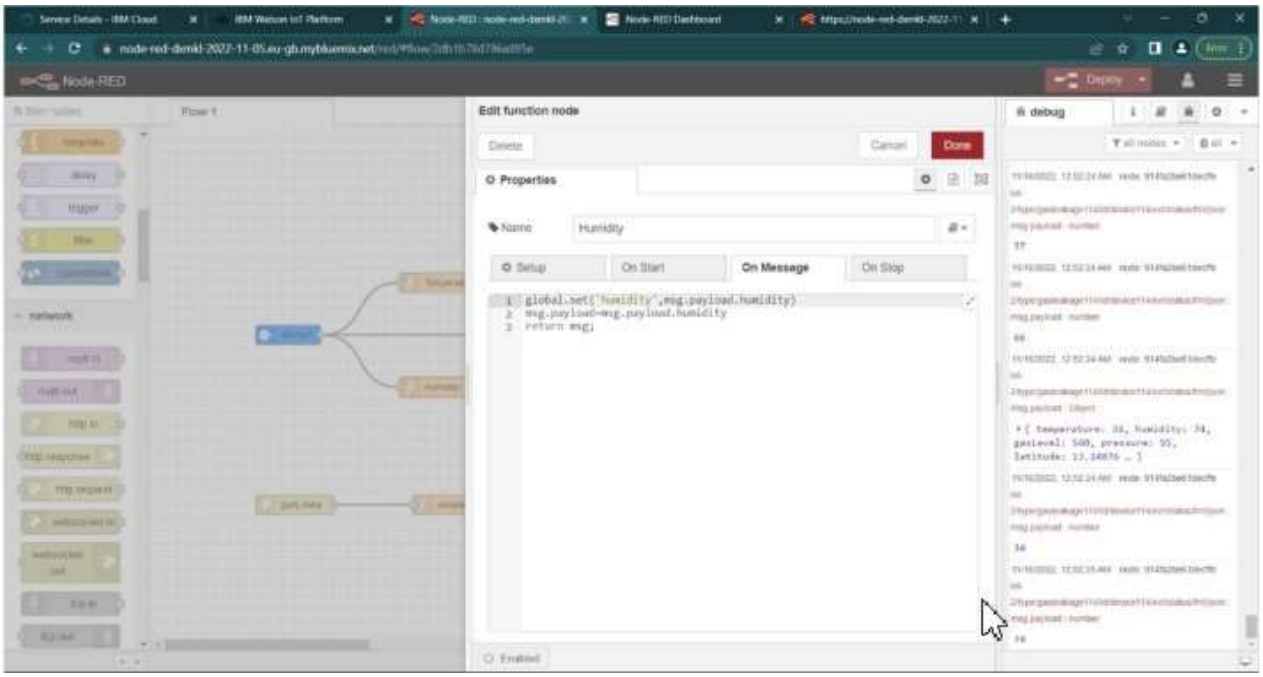


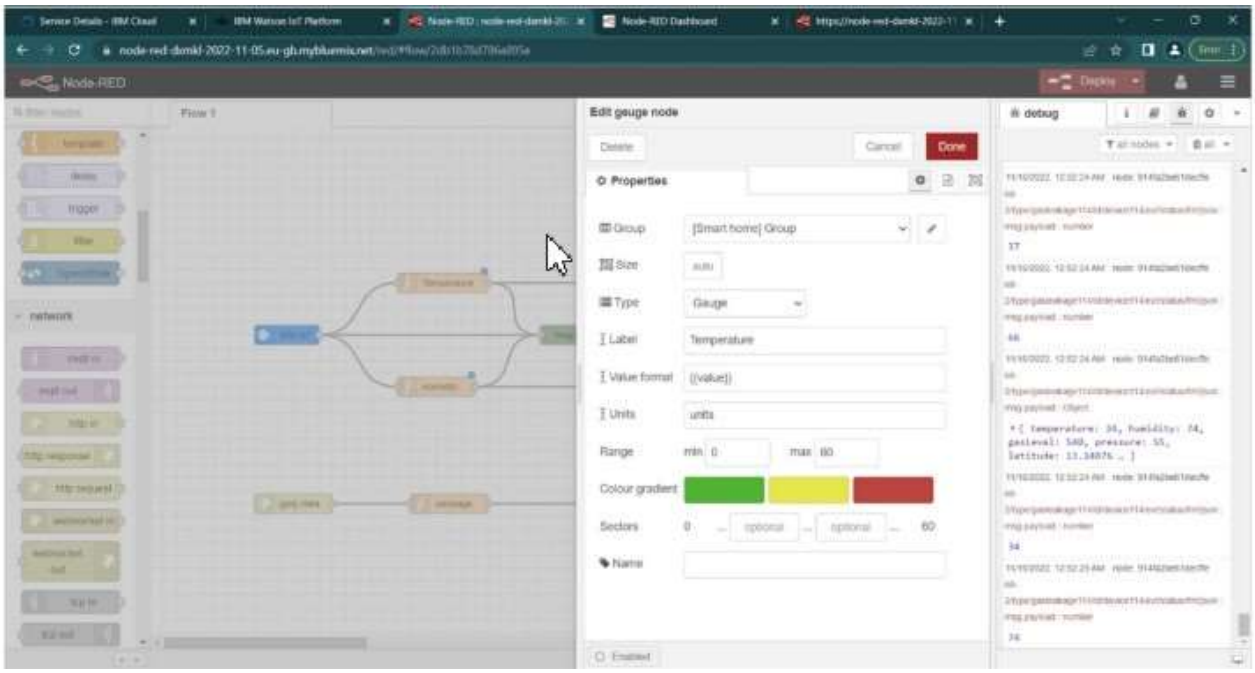


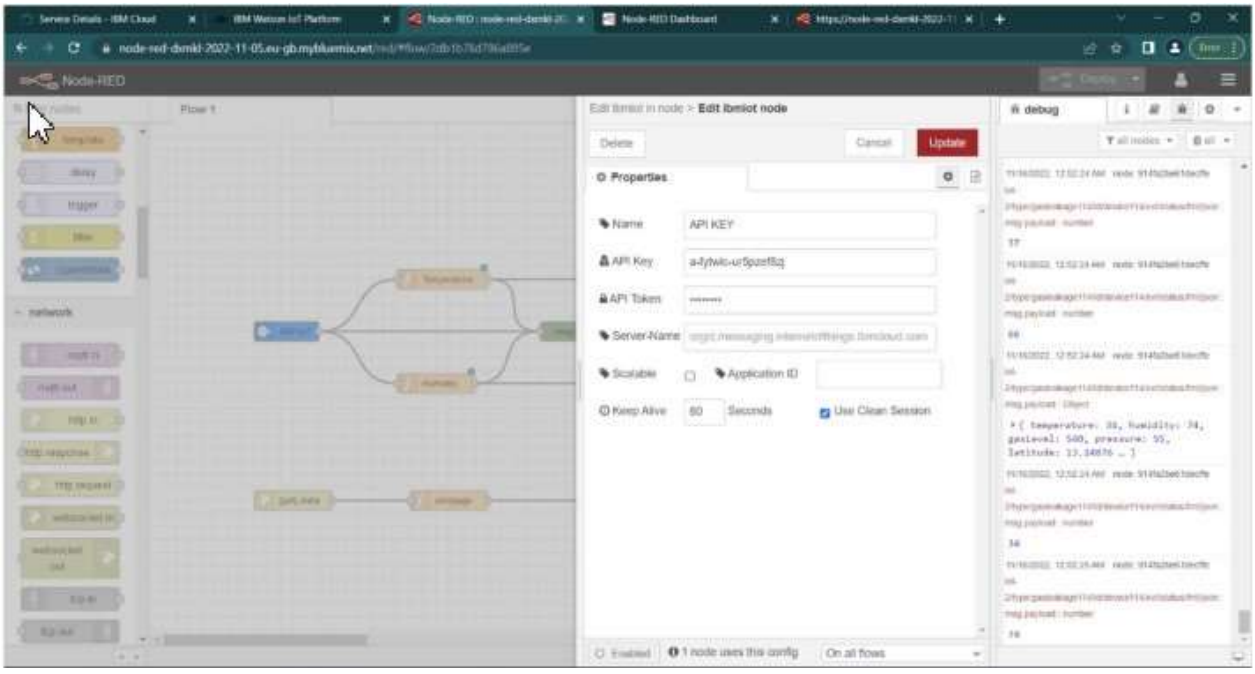
IBM CLOUD USED TO STORE DATA:

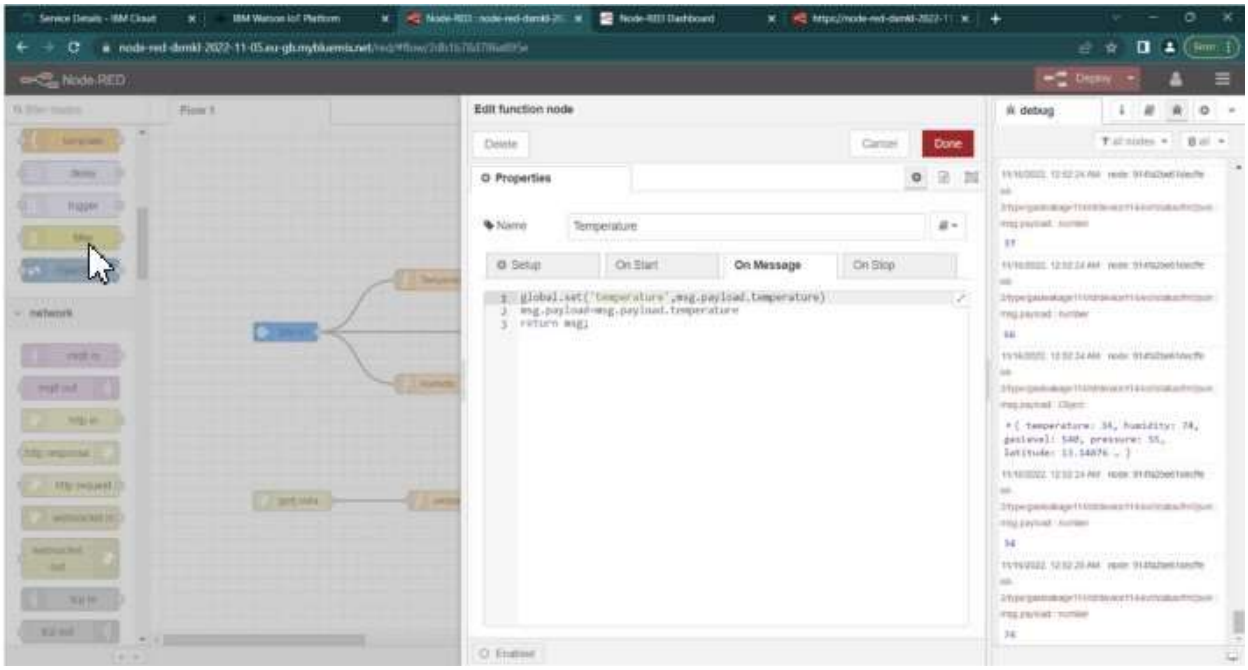


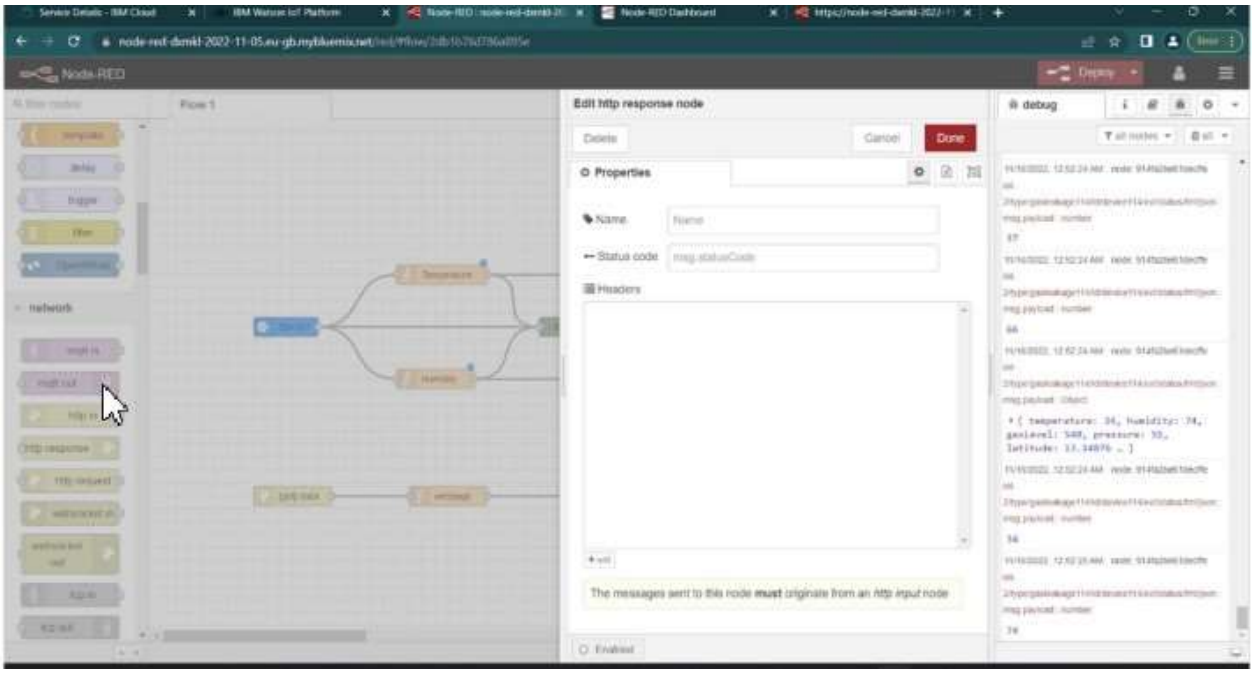
DEVELOP A WEB APPLICATION USING NODE-RED:



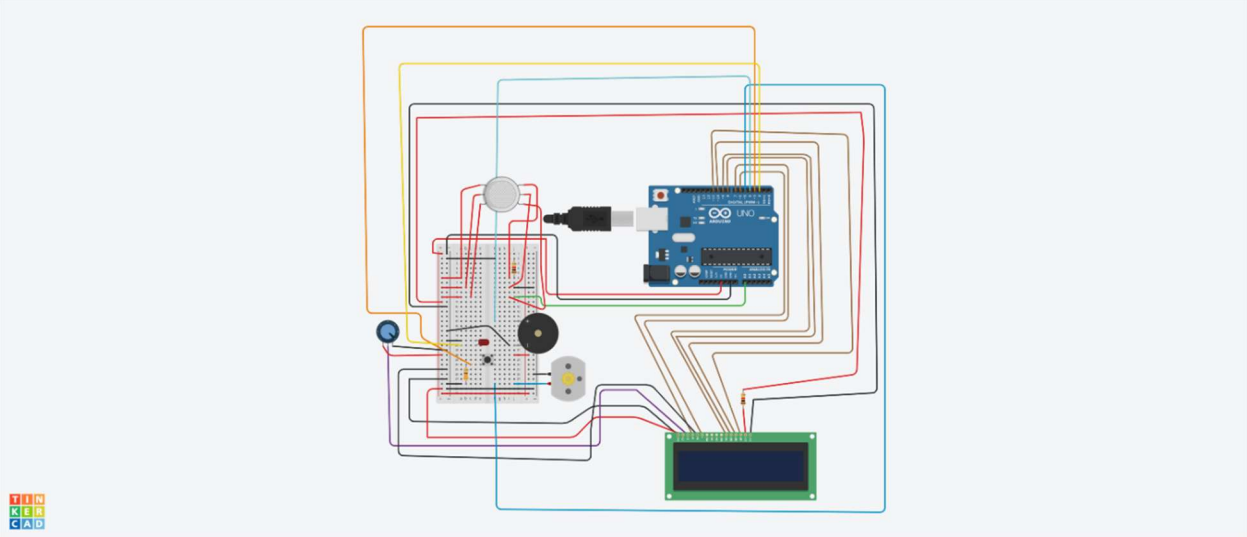

Node-RED
Edit gauge node
Delete
Cancel
Done
Properties
Group
[Smart home] Group
Size
Type
Gauge
Label
Temperature
Value format
{{value}}
Units
units
Range
min 0
max 80
Colour gradient
Sectors
0
60
Name
debug


Node-RED
Edit ibmiot in node > Edit ibmiot node
Delete
Cancel
Update
Properties
Name
API KEY
API Key
API Token
Server-Name
Scalable
Application ID
Keep Alive
60
Seconds
Use Clean Session
1 node uses this config
On all flows
debug





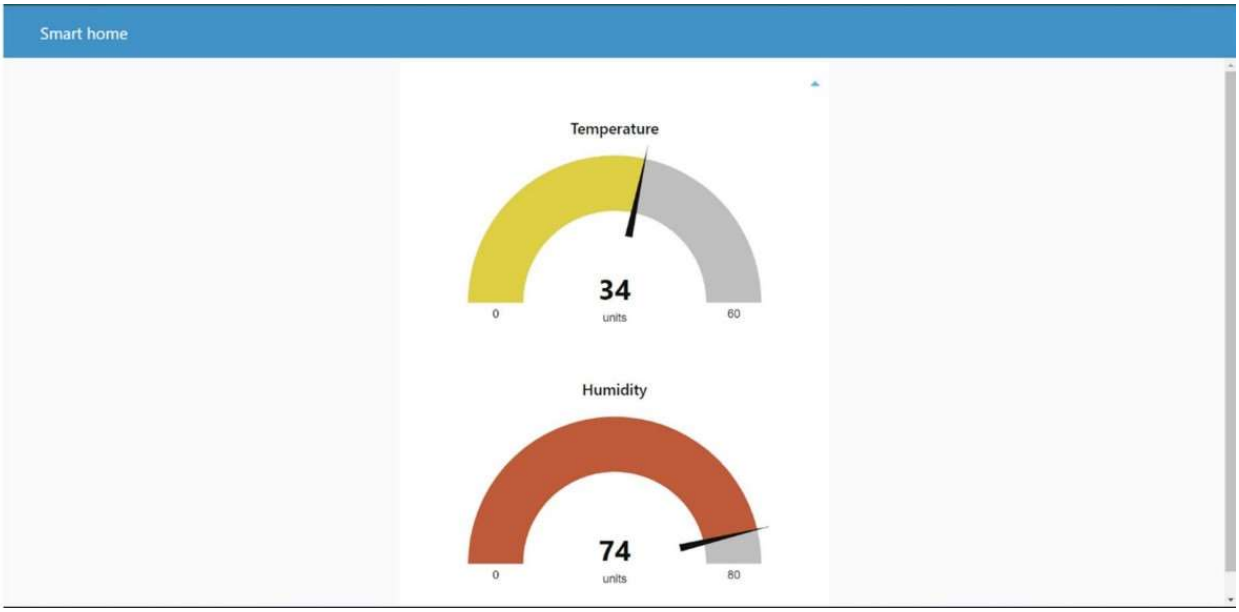
USING TINKERCAD CREATE A CIRCUIT:



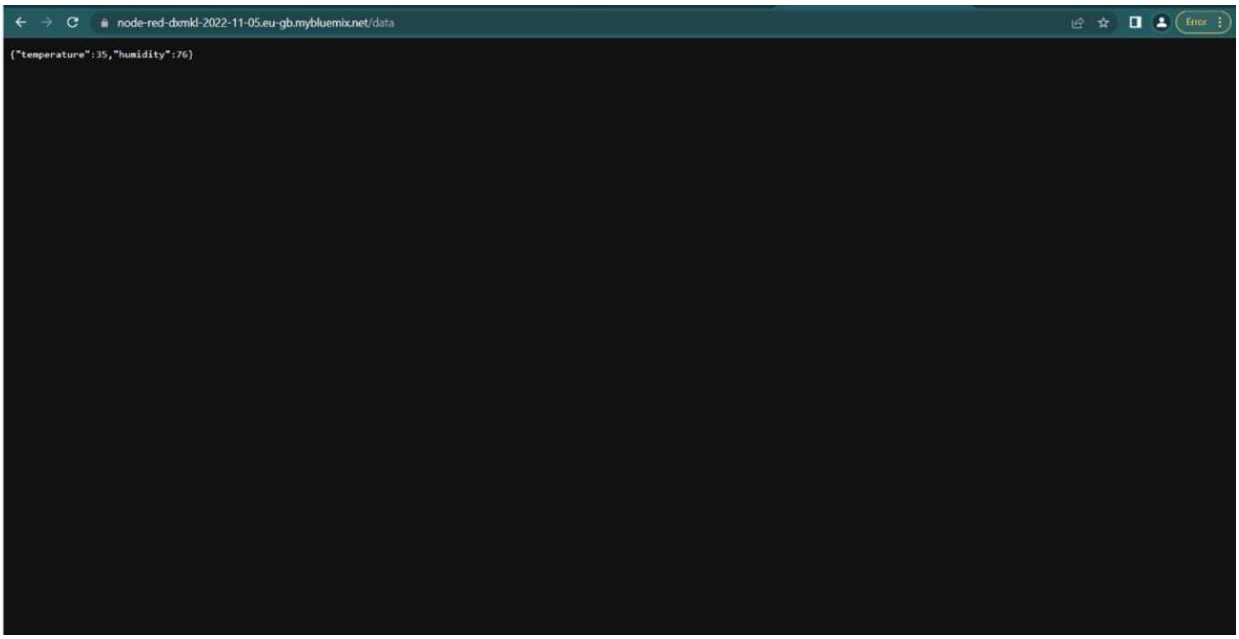
9.RESULTS

9.1 PERFORMANCE METRICS

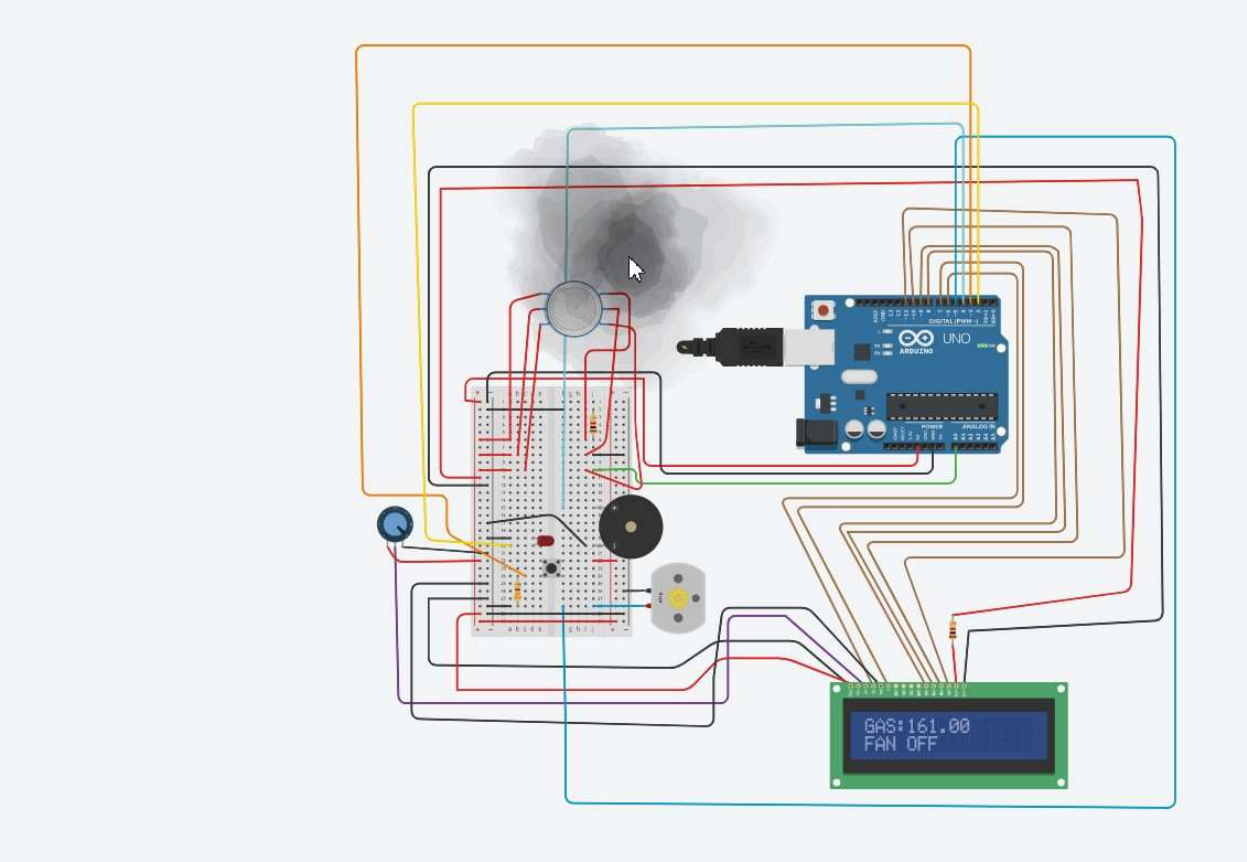
DASHBOARD:



RESULT:



CIRCUIT SIMULATION RESULT:



10.ADVANTAGES & DISADVANTAGES

ADVANTAGES

Some of the advantage of using gas leakage monitoring and alerting system for industries are:

- Get real-time alerts about the gaseous presence in the atmosphere
- Prevent fire hazards and explosions
- Supervise gas concentration levels
- Ensure worker's health
- Real-time updates about leakages
- Get immediate gas leak alerts

DISADVANTAGES

Some of the disadvantages are:

- Sending late notification
- Sometimes false alarm occurs
- Using IOT devices so costly

11.CONCLUSION

This project aims to develop a monitoring and alerting system to meet safety standards and avoid free accidents caused by leakage. The system detects gas in the atmosphere and will continuously update and display the gas value, which the user can easily see via the mobile app. This system has a faster response rate and can disseminate critical information faster than manual methods. In the event of a leak, the system alerts and responds quickly by sending SMS to the appropriate authority

.

12.FUTURE SCOPE

In IOT have different types of devices like sensors. We use only gas sensor to sense the leakage in addition to use temperature sensor. The

temperature sensor can be used which detects the high-pressure gas and display the alert SMS when a high temperature is reached.

13.APPENDIX

SOURCE CODE:

PYTHON CODE:

```
#IBM Watson IOT Platform
#pip install wiotp-sdk
import wiotp.sdk.device
import time
import random
myConfig = {
    "identity": {
        "orgId": "fytwic",
        "typeId": "gasleakage114",
        "deviceId":"device114"
    },
    "auth": {
        "token": "01234567"
    }
}
def myCommandCallback(cmd):
    print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
    m=cmd.data['command']
client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
```

```
client.connect()


while True:
    temp=random.randint(32,40)
    hum=random.randint(60,80)
    gas=random.randint(500,800)
    pres=random.randint(20,80)
    myData={'temperature':temp, 'humidity':hum, 'gasLevel':gas, 'pressure':pres,
'latitude':13.148760, 'longitude':80.229100}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,
onPublish=None)
    print("Published data Successfully: %s", myData)
    client.commandCallback = myCommandCallback
    time.sleep(2)
client.disconnect()
```

FOR CIRCUIT:

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(6, 7, 8, 9, 10, 11);
float gasPin = A0;
float gasLevel;
int ledPin = 2;
int buttonPin = 3;
int buzzPin = 4;
int buttonState;
int fan = 5;

void setup(){
  pinMode(ledPin, OUTPUT);
```

```
  pinMode(buttonPin, INPUT);
  pinMode(gasPin,INPUT);
  pinMode(fan,OUTPUT);
  Serial.begin(9600);
  lcd.begin(16, 2);
  lcd.setCursor(0,0);
  lcd.print(" Welcome");
  lcd.setCursor(0,2);
  lcd.print("PNT2022TMID19555");
  delay(500);
  lcd.clear();
}

void loop(){
  // Read the value from gas sensor and button
  gasLevel = analogRead(gasPin);
  buttonState = digitalRead(buttonPin);
  // call the function for gas detection and button work
  gasDetected(gasLevel);
  buzzer(gasLevel);
  exhaustFanOn(buttonState);
}
// Gas Leakage Detection & Automatic Alarm and Fan ON
void gasDetected(float gasLevel){
  if(gasLevel >= 200){
   digitalWrite(buzzPin,HIGH);
    digitalWrite(ledPin,HIGH);
    digitalWrite(fan,HIGH);
```

```
   lcd.setCursor(0,0);
   lcd.print("GAS:");
    lcd.print(gasLevel);
   lcd.setCursor(0,2);
   lcd.print("FAN ON");
   delay(1000);
   lcd.clear();
  }else{
   digitalWrite(ledPin,LOW);
    digitalWrite(buzzPin,LOW);
    digitalWrite(fan,LOW);
    lcd.setCursor(0,0);
   lcd.print("GAS:");
    lcd.print(gasLevel);
   lcd.setCursor(0,2);
   lcd.print("FAN OFF");
   delay(100);
   lcd.clear();
  }
}
```

//BUZZER

```
void buzzer(float gasLevel){
if(gasLevel>=200)
  {
  for(int i=0; i<=30; i=i+10)
  {
  tone(4,i);
```

```
  delay(300);
  noTone(4);
  delay(4300);
  }
  }
}
```

// Manually Exhaust FAN ON

```
void exhaustFanOn(int buttonState){
  if(buttonState == HIGH){
    digitalWrite(fan,HIGH);
    lcd.setCursor(0,0);
   lcd.print("Button State:");
    lcd.print(buttonState);
    lcd.setCursor(0,2);
    lcd.print("FAN ON");
    delay(10000);
    lcd.clear();
  }}
```

GITHUB PROJECT LINK:

LINK- https://github.com/IBM-EPBL/IBM-Project-32504-1660210420