IBM PROJECT REPORT

PROJECT NAME	**GAS LEAKAGE MONITORING & ALERTING SYSTEM FOR INDUSTRIES**
TEAM ID	**PNT2022TMID16564**
TEAM MEMBERS	**SANTHOSH S** **SARAN S** **PRATHAP A** **SUNIL KUMAR K**

CONTENTS

7. **CODING & SOLUTIONING (Explain the features added in the project along with code)** a. Feature 1
    b. Feature 2
    c. Database Schema (if Applicable)

8. **TESTING**
    a. Test Cases
    b. User Acceptance Testing

9. **RESULTS**
    a. Performance Metrics

10. **ADVANTAGES & DISADVANTAGES**

11. **CONCLUSION**

12. **FUTURE SCOPE**

13. **APPENDIX**

    Source Code

    GitHub & Project Demo Link

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 domes c as well as working places. The inflammable gas, which is excessively used in the work places (Industries). The leakage of the gas causes destruc ble impact to the lives and as well as to the heritage of the people. Most of the socie es have fire safety mechanism. But it can use a er the fire exists. As a result, a system for detec ng 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 no fy 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 automa c 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, mul featured and simple system device for detec ng 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 Exis ng 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 Aler ng techniques involving sending a text message to the concerned authority and the ability to perform data analy cs 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 aler ng to society members through the alarm and sending no fica ons.

2.2 References:

1. Shital Imade, Priyanka Rajmanes, Aishwarya Gavali , Prof. V. N. Nayakwadi “GAS LEAKAGE DETECTION AND SMART ALERTING SYSTEM USING IOT” h ps://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 " Interna onal Advance Compu ng 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 Communica ons 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:h p://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 detec on and controlling system” the Interna onal 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 Interna onal Journal of Technical

Research and Applica ons e-ISSN: 2320- 8163,www.ijtra.com Volume 1, Issue 2 (mayJune 2013).

11. C.Selvapriya, S.Sathyaprabha, M.Abdulrahim," LPG leakage monitoring and mul level aler ng system", published in 2013.

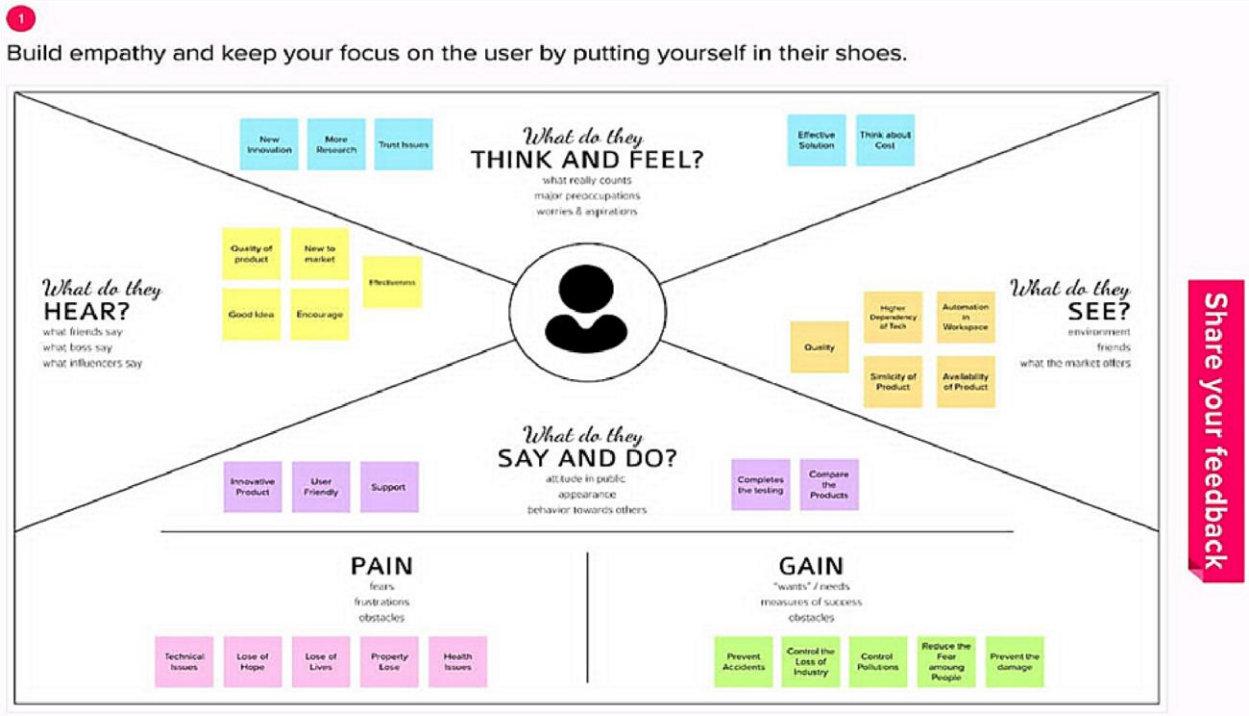
12. Falohun A.S., Oke A.O., Abolaji B.M. "Dangerous gas detec on using an integrated circuit and MQ-9. In Interna onal Journal of Computer Applica ons. 2016; 135(7).

2.3 Problem Statement Defini on:

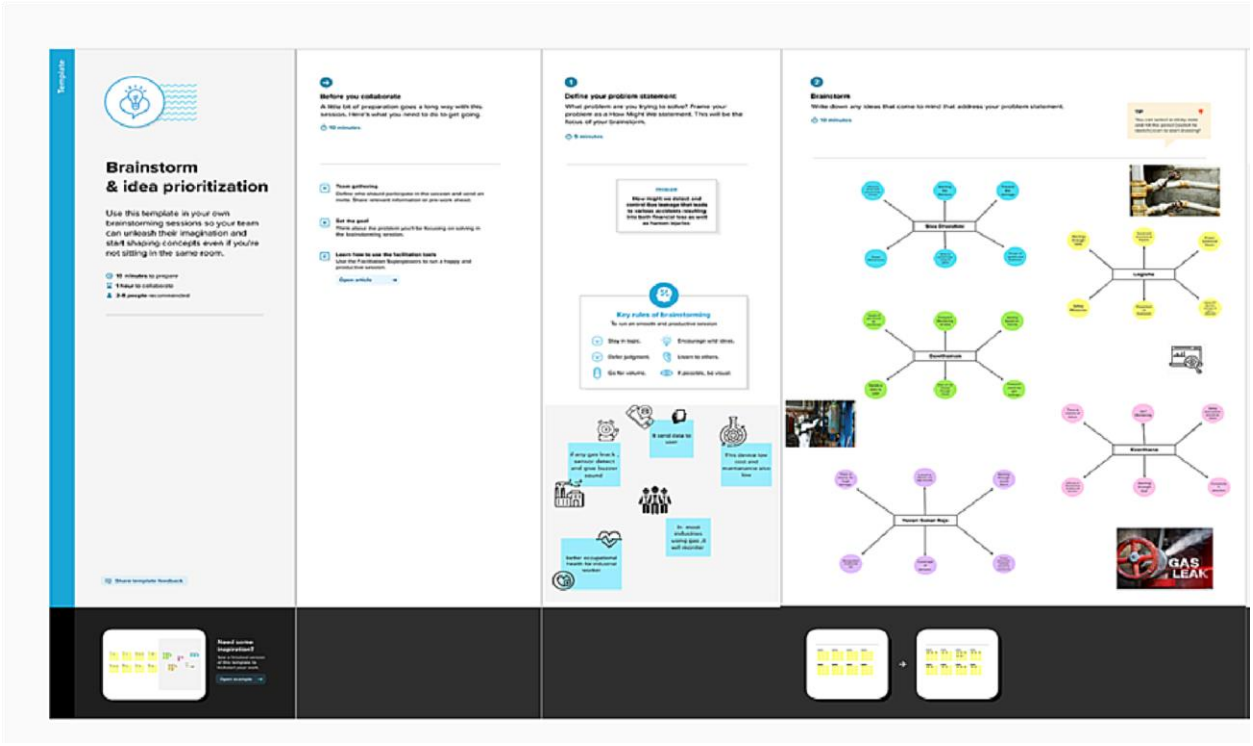
In most industries, one of the key parts of any safety plan for reducing risks to personnel and plant is the use of early-warning devices such as gas detectors. These can help to provide more me in which to take remedial or protec ve ac on. They can also be used as part of a total, integrated monitoring 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. Solu ons 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 automa cally respond to it whenever the leakage occurs.

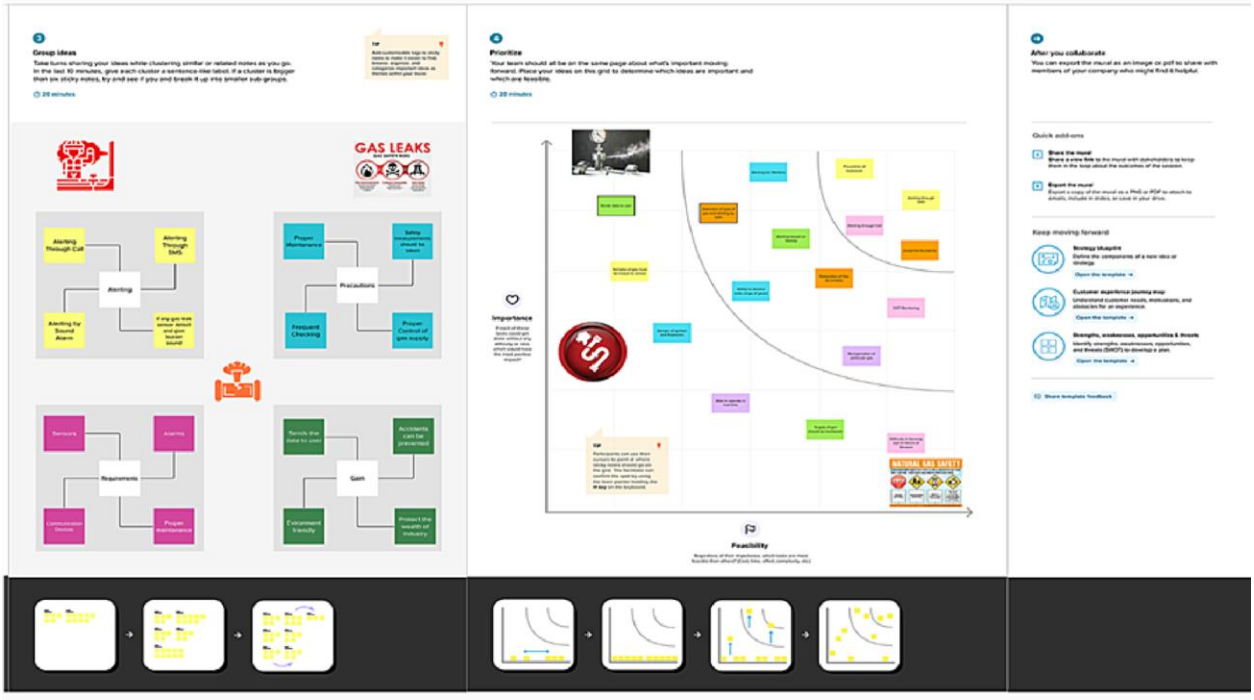
3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas:



3.2 Idea on & Brainstorming:



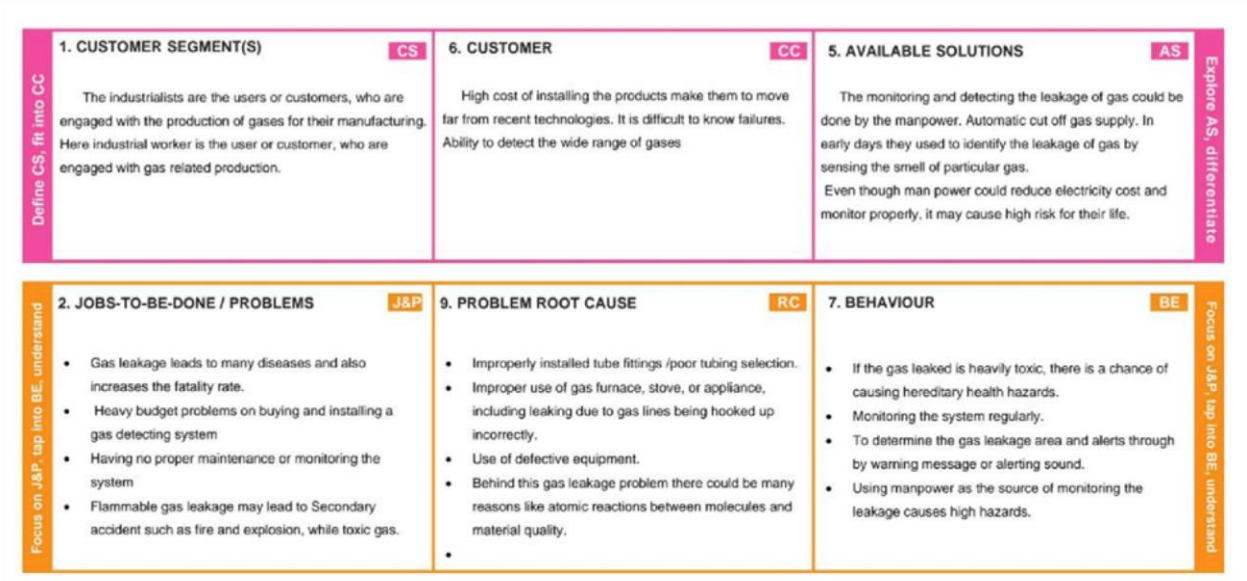


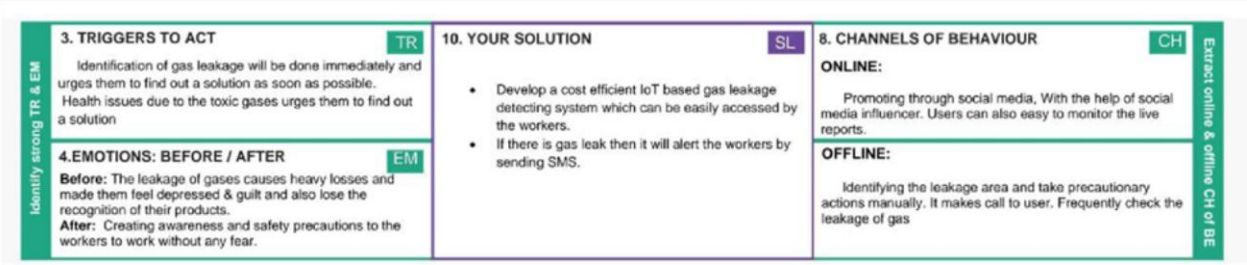
3.3 Proposed Solu on:

S.No	Parameter	Descrip on
1.	Problem Statement (Problem to be solved)	Gas leakage leads to various accidents resul ng in loss of human lives and industry proper es. Some mes, 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 / Solu on descrip on	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 loca on tagging and alert service so that the admin and fire department team will be no fied the exact loca on. The system provides constant monitoring and detec on of gas leakage along with storage of data in database for predic ons and analysis.
4.	Social Impact / Customer Sa sfac on	By implemen ng real- me gas leak detec on, industries can monitor their environmental performance, ensure be er occupa onal health. Also, early detec on of gas leaks can trigger concerned engineers to curtail the spread and keep a safe environment for be er 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 Solu on	The system is very simple and easy to maintain and cost efficient. It has the capability to works for a period of me without any damage in the system components.

3.4 Problem Solu on fit





FR No.	Func onal Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registra on	Registra on through Form Registra on through Gmail Registra on through LinkedIN
FR-2	User Confirma on	Confirma on via Email Confirma on via OTP
FR-3	GPS Access	GPS access to know the loca on
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 preven ve measures can be taken to avoid the loss of proper es.

FR-5	User Requirements	The Gas leakage detec ng system with upgrading technologies which iden fies the leakage of gas and also ensures the workers safety.

4. REQUIREMENT ANALYSIS

4.1Func onal Requirements:

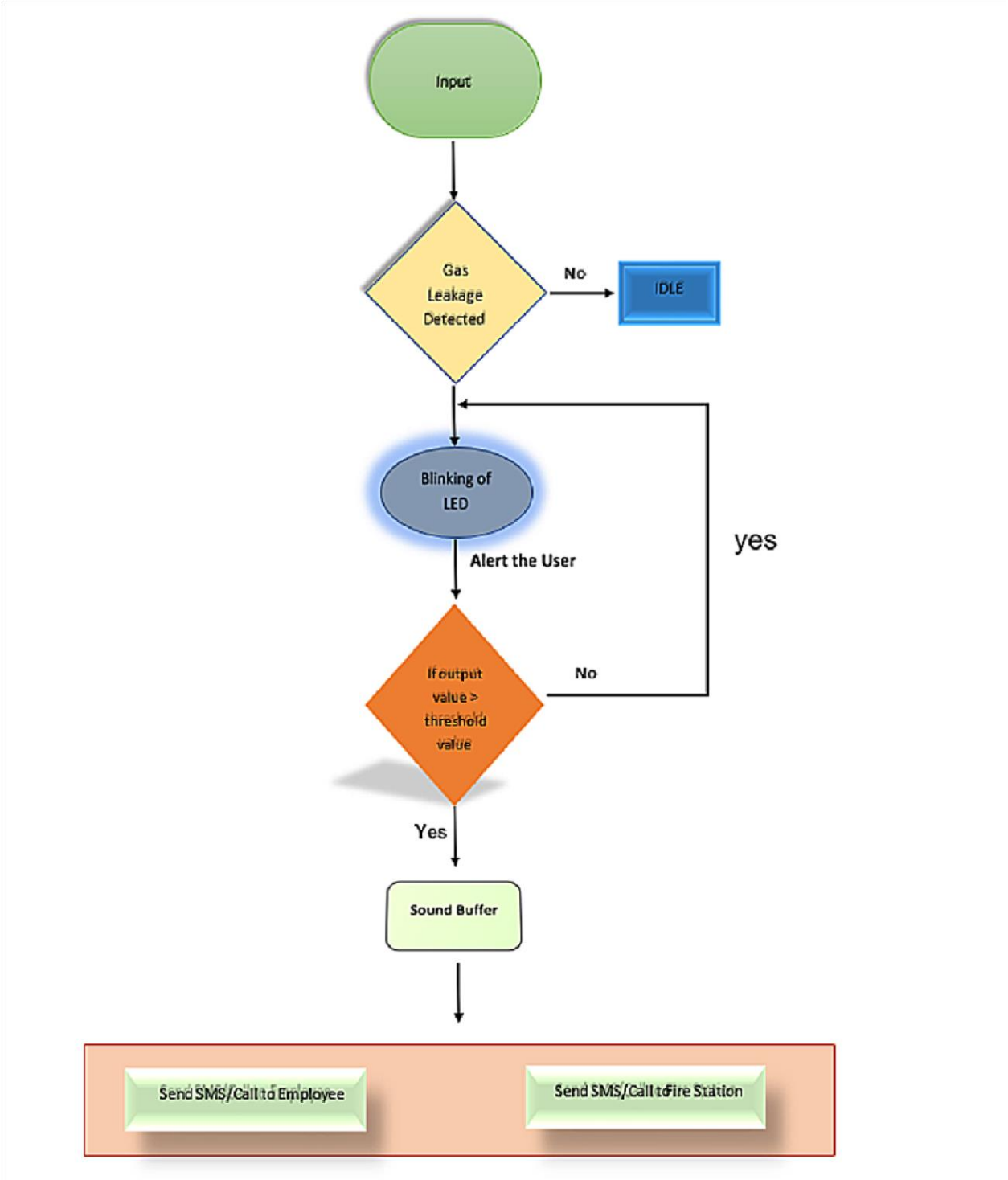
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Methodology	It is a well way to get rapid results in a short me.
FR-2	Impact	Sensor has excellent sensi vity combined with a quick fast response me , possible to get instantaneous results.
FR-3	sensi vity	Specialised of the gas in all similar systems
FR-4	WIFI -Module	Can communicate directly with industrial scien fic, consumer technology that is web friendly with no use of shields or any peripherals.
FR-5	opera on	The system be operated in android opera ng system.
FR-6	User Interface	Emergency call, message with applica on systems

4.2 Non-Func onal requirements:

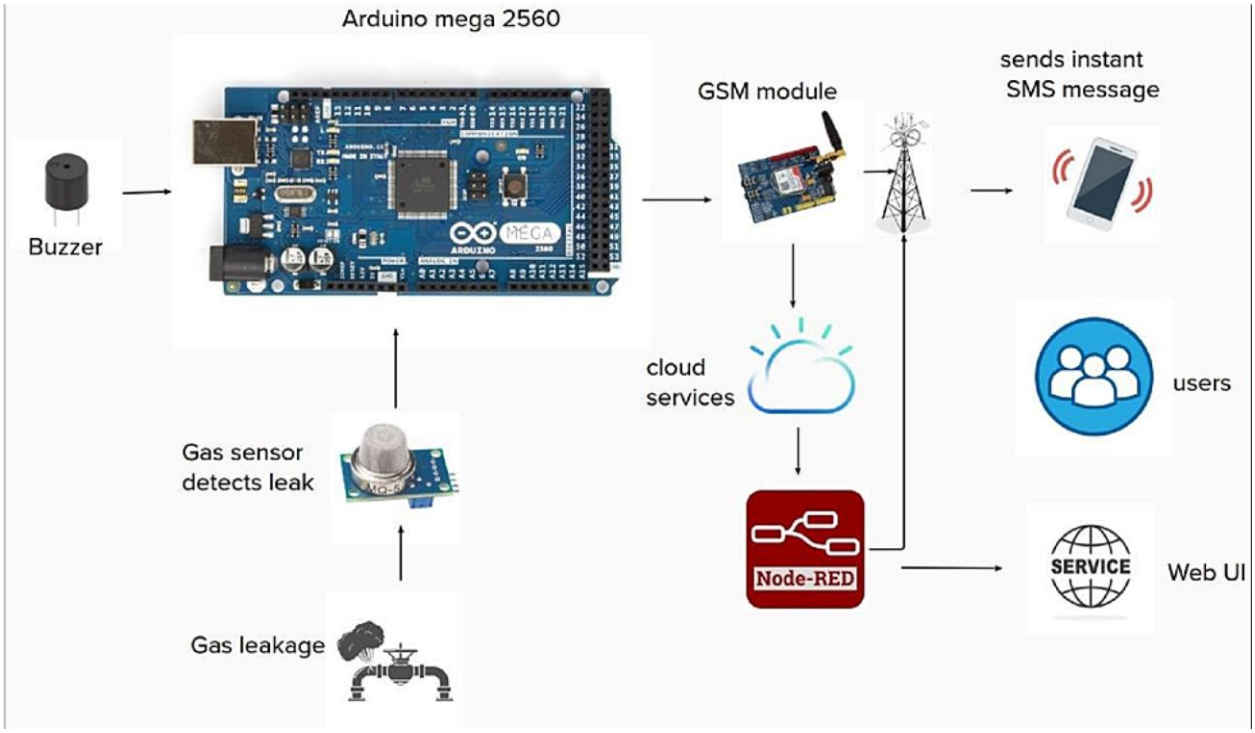
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	It helps prevent the high risk of gas explosions
NFR-2	**Security**	The system should not display the homeowner personal informa on to anyone.
NFR-3	**Reliability**	Unsafe behaviour of personnel has the greatest impact on the probability of gas leakage.
NFR-4	**Performance**	Arduino response me will be fast.
NFR-5	**Availability**	The system should work 24 hours 7 days a week.
NFR-6	**Scalability**	The system interface should be easy and effec ve(user-friendly).

5. PROJECT DESIGN

5.1 Data Flow Diagrams:



5.2 Solu on & Technical Architecture:



5.3 User Stories:

User Type	Func onal Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registra on	USN-1	As a user, I can create an account in the applica on provided.	I can access my account/ dashboard	High	Sprint-1
		USN-2	As a user, I registered using my Gmail.	I can receive confirma on 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)	Registra on	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 applica on.	I can login to the website using my login creden als	High	Sprint-2
Customer Care Execu ve		CCE-1	A customer care execu ve will always be available for the interac on with the customer to clarify the queries.	An execu ve will clarify the doubts and note down the complaints of the applica on if any.	High	Sprint-2
Administrator		ADMIN-1	I as an Admin can access and view the data or informa on provided by the applica on &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 aler ng sound is received.	High	Sprint-1

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Es ma on:

1. SPRINT PLAN
2. ANALYZE THE PROBLEM
3. PREPARE an ABSTRACT, PROBLEM STATEMENT
4. LIST A REQUIRED OBJECT NEEDED
5. CREATE A PROGRAM CODE AND RUN IT
6. MAKE A PROTOTYPE TO IMPLEMENT
7. TEST WITH THE CREATED CODE AND CHECK THE DESIGNED PROTOTYPE

6.2 Sprint Delivery Schedule:

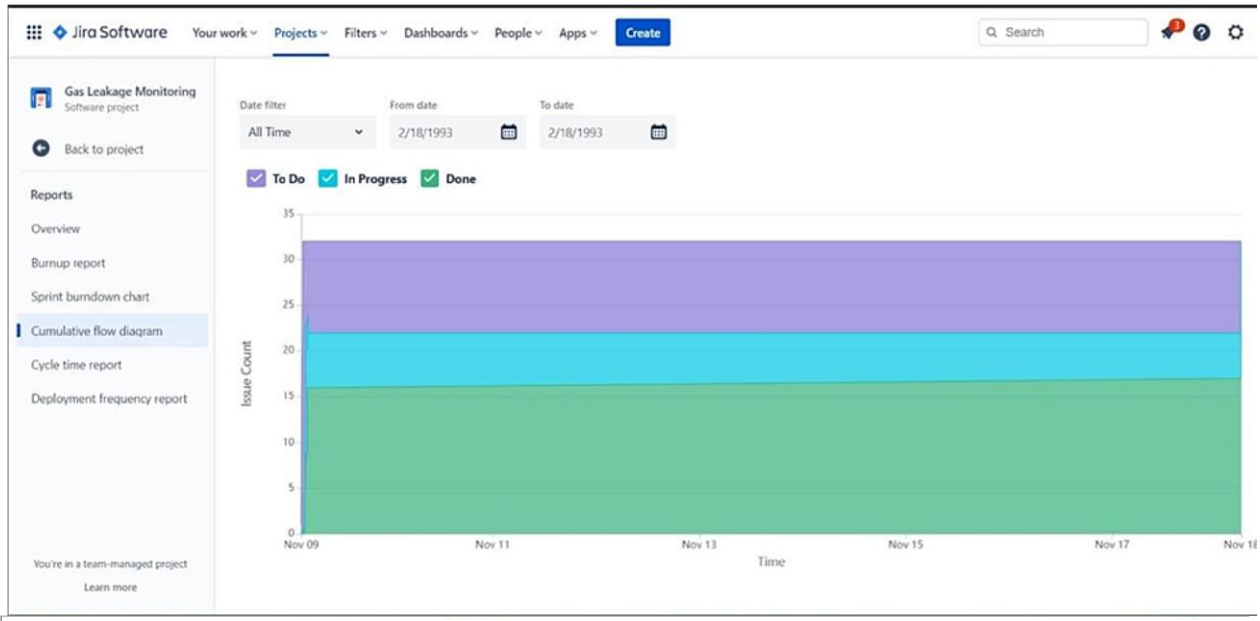
Sprint	Func onal Requirement (Epic)	User Story	User Story / Task	Story Point	Priority
Sprint-1	Create	US-1	Create the IBM Cloud services which are being used in this project.	5	High

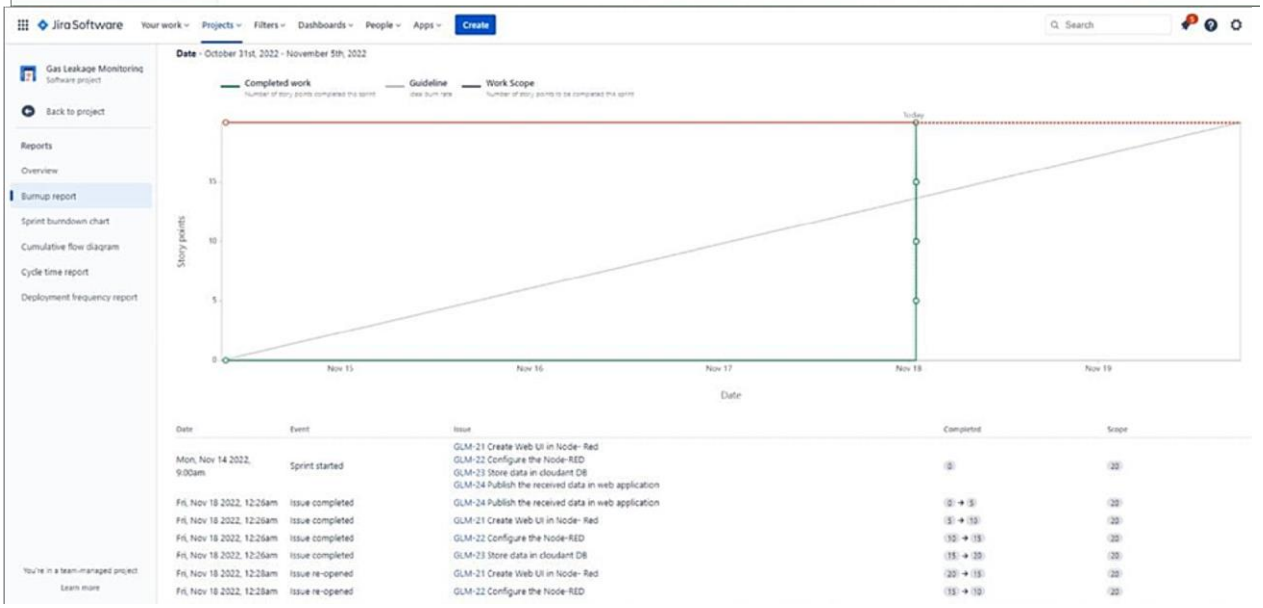
Sprint-1	Configure	US-2	Configure the IBM Cloud services which are being used in comple ng this project.	1	Medium
Sprint-1	Create	US-3	IBM Watson IoT pla orm acts as the mediator to connect the web applica on to IoT devices, so create the IBM Watson IoT pla orm.	1	Medium
Sprint-1	Configure	US-4	Configure 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 pla orm and get the device creden als.	13	High
Sprint-2	Configure	US-2	Configure a device in the IBM Watson IoT pla orm and get the device creden als.	3	Medium

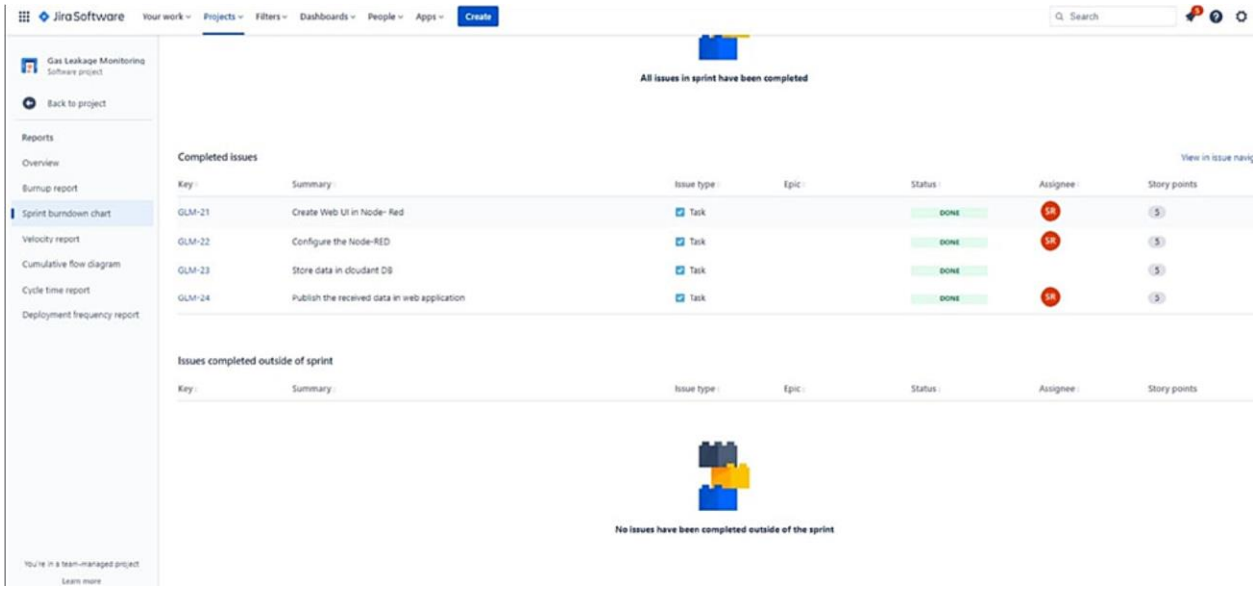
Sprint-2	Create	US-3	Create a Node-RED service.	3	High
Sprint-2	Configure	US-4	Configure the connec on security and create API keys that are used in the Node- RED service for accessing the IBM IoT Pla orm.	1	Medium
Sprint-3	Develop	US-1	Develop a python script to publish random sensor data such as temperature, Flame level and Gas level to the IBM IoTpla orm	1 3	High
Sprint-3	Configure	US-2	A er 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

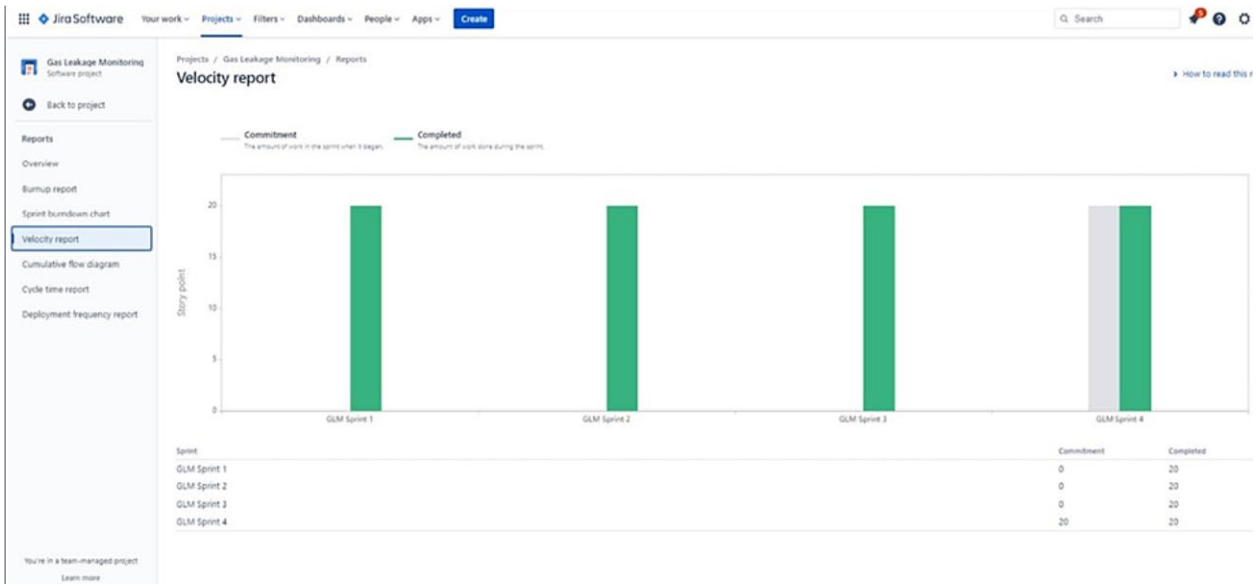
Sprint-3	Publish	US-4	Publish Data to The IBM Cloud	5	High
Sprint-4	Create	US-1	Create Web UI in Node- Red	5	High
Sprint-4	Configure	US-2	Configure the Node- RED flow to receive data from the IBMIoT pla orm	5	High
Sprint-4	Configure	US-3	Use cloudant DB nodes to store the received sensor data in the cloudant DB	5	High
Sprint-4	Publish	US-4	Publish the received data in webapplicati on	5	High

6.3 Report from JIRA:









7. CODING & SOLUTIONING:

```
# Impor ng Required modules
import me   import sys
import  wiotp.sdk.device# IBM IoT Watson Pla orm Module import
ibmio .device
import tkinter as tk # Python GUI Package
from tkinter import k # Python GUI
import me
```

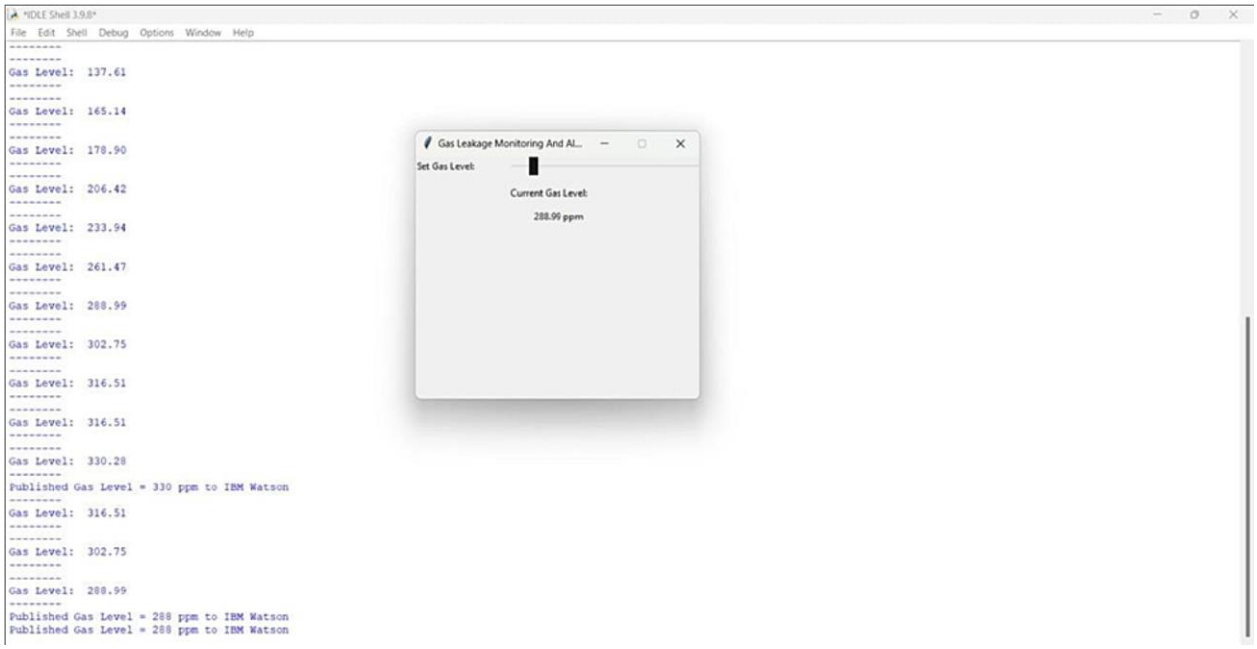
```
from threading import Thread organiza on
= "ioz5i8" # Organiza on ID deviceType =
"raspberrypi"   # Device type
deviceId = "123456"    # Device ID
authMethod = "token"    # Authen ca on Method authToken
= "a-ioz5i8-dl5lbxjraw" #Replace the authtoken
# Tkinter root window
root = tk.Tk() root.geometry('350x300')  # Set size of
root window root.resizable(False, False) # root window
non-resizable
root. tle('Gas Leakage Monitoring And Aler ng System for Industries')
# Layout Configura ons
root.columnconfigure(0, weight=1)
root.columnconfigure(1, weight=3)
current_gas = tk.DoubleVar()
def get_current_gas(): # func on 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 =
k.Label(root,text='Set                Gas                Level:')
slider_gas_label.grid(column=0,row=0,s cky='w')
#  Gas Level slider
slider_gas = k.Scale(root,from_=0,to=3000,orient='horizontal',
command=slider_changed,variable=current_gas)
slider_gas.grid(column=1,row=0,s cky='we')
# current gas level label
current_gas_label = k.Label(root,text='Current Gas Level:')
current_gas_label.grid(row=1,columnspan=2,s cky='n',ipadx=10,ipady=10)
# Gas level label (value gets displayed here)
gas_label = k.Label(root,text=str(get_current_gas()) +" ppm")
gas_label.grid(row=2,columnspan=2,s cky='n')
def publisher_thread():
```

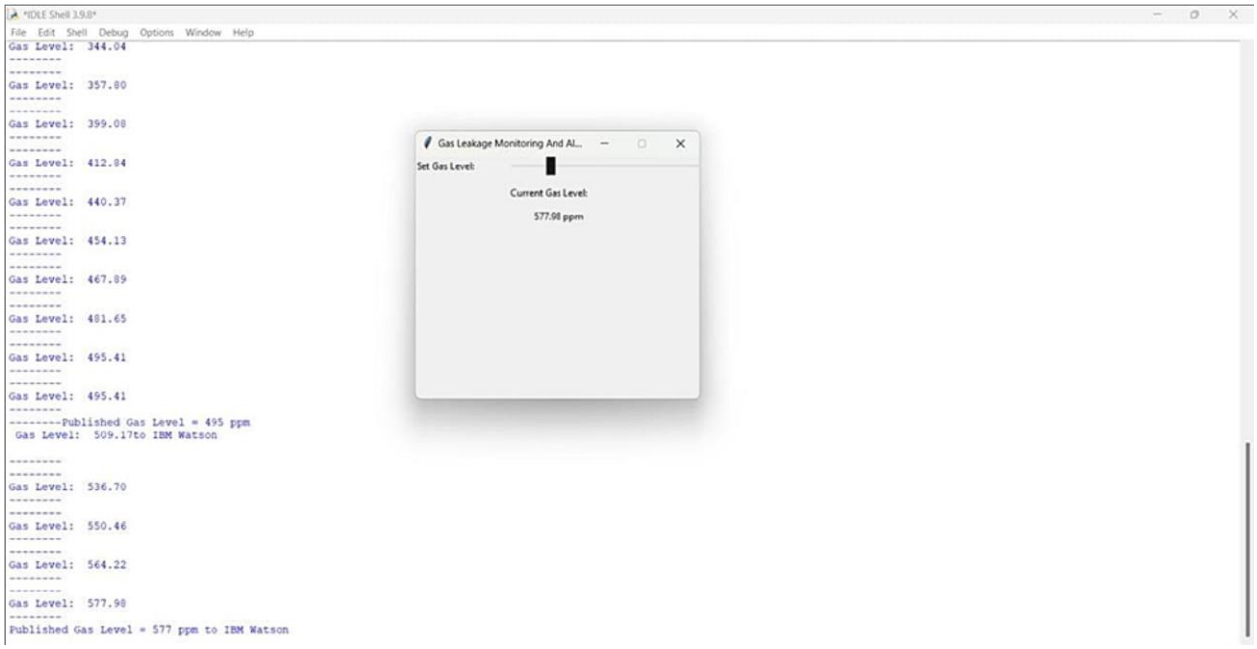
```
   thread = Thread(target=publish_data)
   thread.start() def
publish_data():    #
Excep on Handling
   try:
      deviceOp ons = {"org": organiza on, "type": deviceType, "id": deviceId, "authmethod":
authMethod,
                  "auth-token": authToken}      deviceCli
= ibmio .device.Client(deviceOp ons)    except Excep
on as e:
      print("Caught excep on connec ng device: %s" % str(e))
sys.exit()
   deviceCli.connect()  # Connect to IBM Watson IoT Pla orm
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")
me.sleep(1) publisher_thread()
root.mainloop() # startup Tkinter GUI
# Disconnect the device and applica on from the cloud deviceCli.disconnect()
```

CODE:

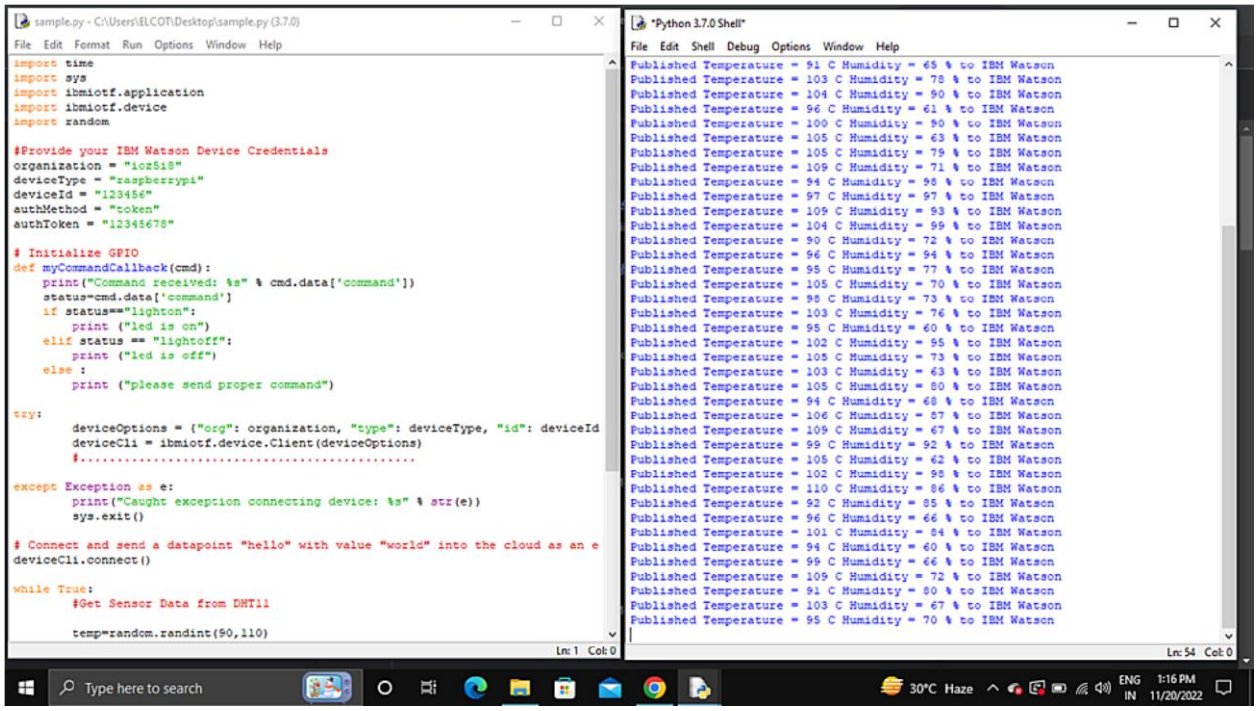
```
File  Edit  Format  Run  Options  Window  Help
# Importing Required modules
import time
import sys
import  wiotp.sdk.device# IBM IoT Watson Platform Module
import ibmiotf.device
import tkinter as tk # Python GUI Package
from tkinter import ttk # Python GUI
import time
from threading import Thread


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')
```

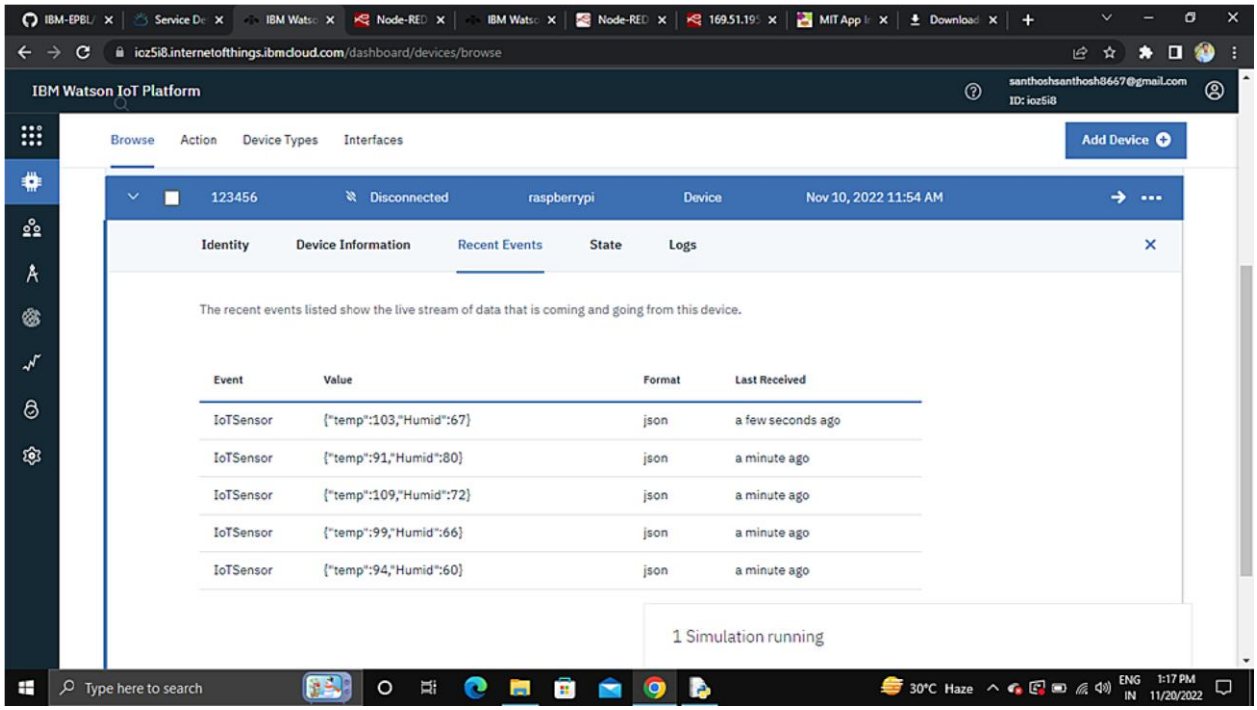
OUTPUT:

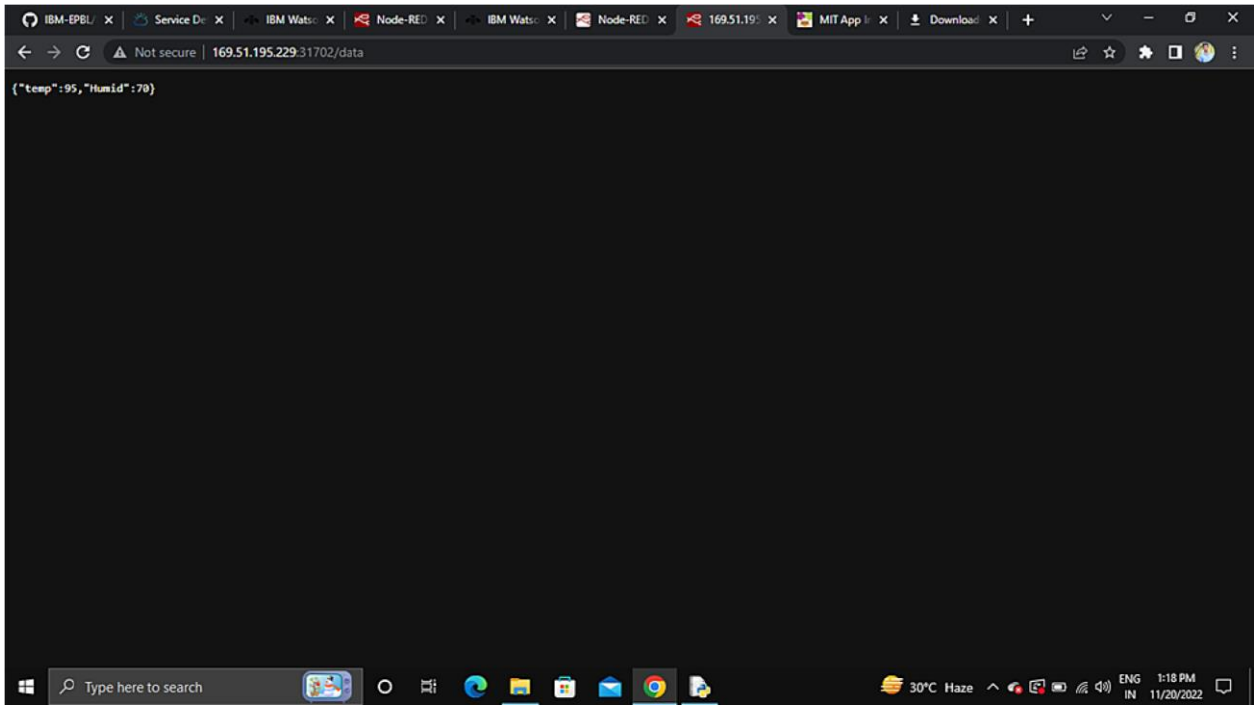

IDLE Shell 3.9.8
File Edit Shell Debug Options Window Help
Gas Level: 137.61
Gas Level: 165.14
Gas Level: 178.90
Gas Level: 206.42
Gas Level: 233.94
Gas Level: 261.47
Gas Level: 288.99
Gas Level: 302.75
Gas Level: 316.51
Gas Level: 316.51
Gas Level: 330.28
Published Gas Level = 330 ppm to IBM Watson
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
Gas Leakage Monitoring And Al...
Set Gas Level:
Current Gas Level:
288.99 ppm

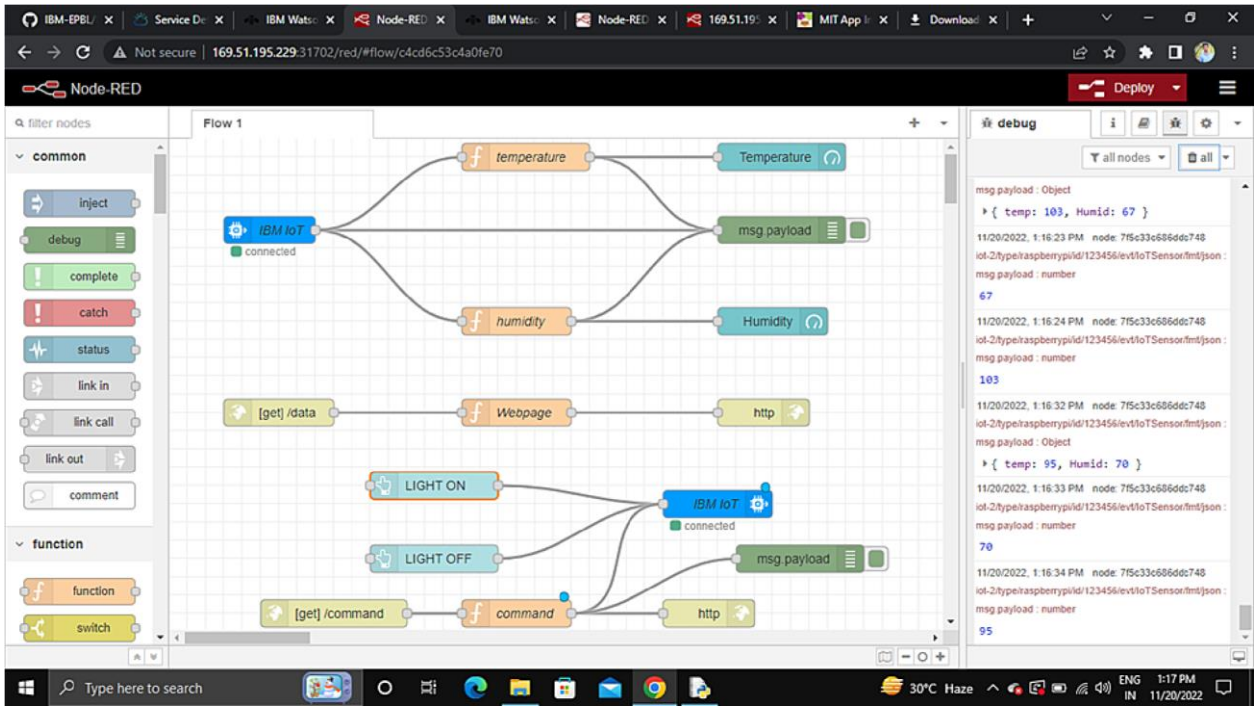

IDLE Shell 3.9.8
File Edit Shell Debug Options Window Help
Gas Level: 344.04
Gas Level: 357.80
Gas Level: 399.08
Gas Level: 412.84
Gas Level: 440.37
Gas Level: 454.13
Gas Level: 467.89
Gas Level: 481.65
Gas Level: 495.41
Gas Level: 495.41
--------Published Gas Level = 495 ppm
Gas Level: 509.17to IBM 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
Gas Leakage Monitoring And Al...
Set Gas Level:
Current Gas Level:
577.98 ppm

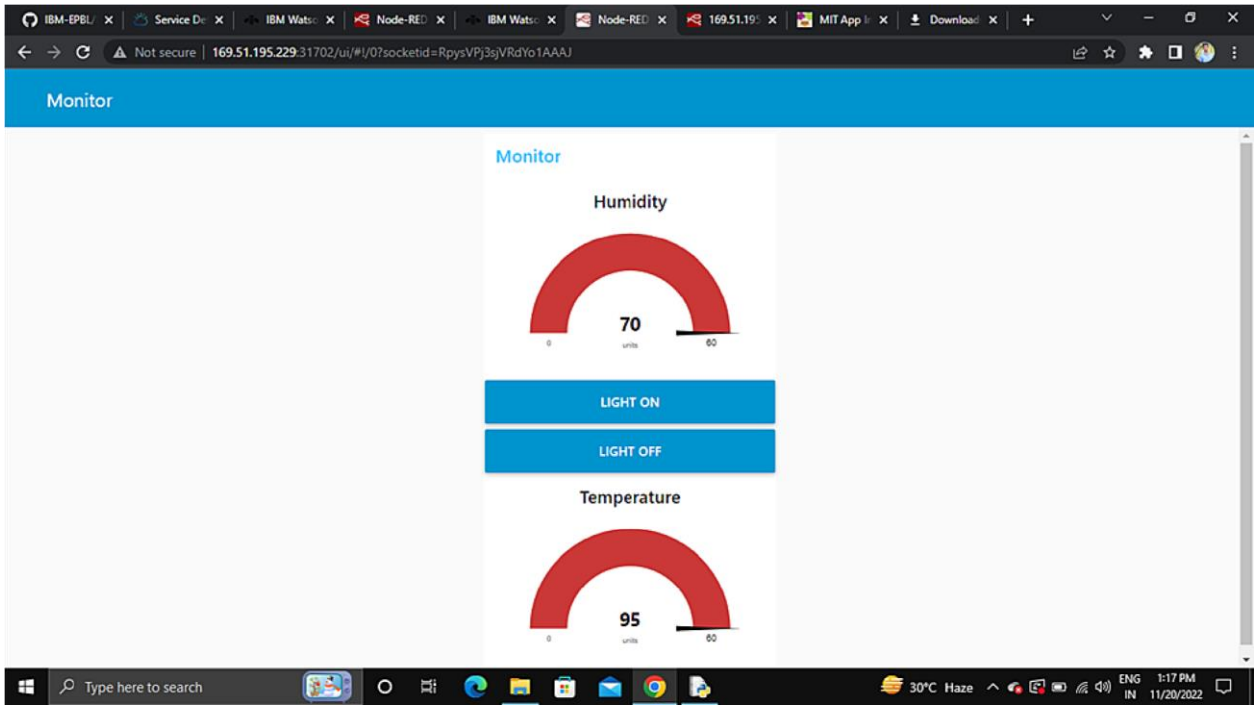
8. Tes ng:

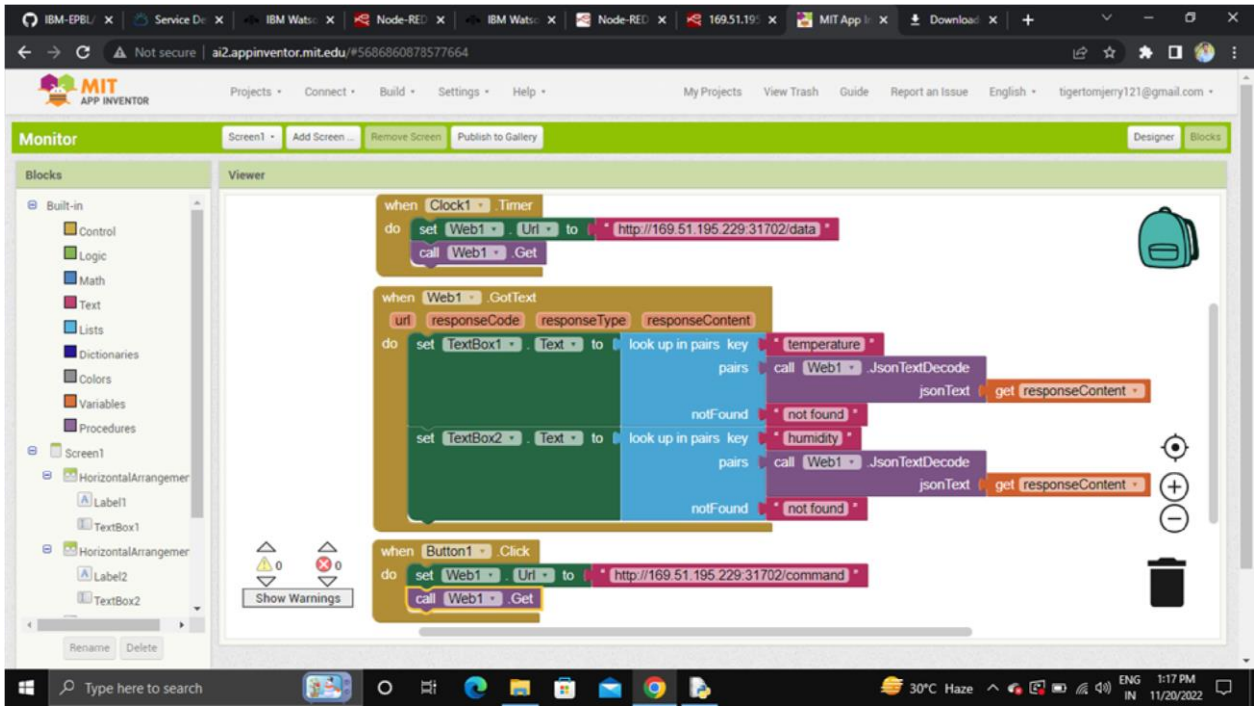

sample.py - C:\Users\ELCOT\Desktop\sample.py (3.7.0)
File Edit Format Run Options Window Help
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
#Provide your IBM Watson Device Credentials
organization = "ioz5i8"
deviceType = "raspberrypi"
deviceId = "123456"
authMethod = "token"
authToken = "12345678"
# Initialize GPIO
def myCommandCallback(cmd):
print("Command received: %s" % cmd.data['command'])
status=cmd.data['command']
if status=="lighton":
print ("led is on")
elif status == "lightoff":
print ("led is off")
else :
print ("please send proper command")
try:
deviceOptions = {"org": organization, "type": deviceType, "id": deviceId
deviceCli = ibmiotf.device.Client(deviceOptions)
#.....
except Exception as e:
print("Caught exception connecting device: %s" % str(e))
sys.exit()
# Connect and send a datapoint "hello" with value "world" into the cloud as an e
deviceCli.connect()
while True:
#Get Sensor Data from DHT11
temp=random.randint(90,110)
Ln: 1 Col: 0
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Published Temperature = 91 C Humidity = 65 % to IBM Watson
Published Temperature = 103 C Humidity = 78 % to IBM Watson
Published Temperature = 104 C Humidity = 90 % to IBM Watson
Published Temperature = 96 C Humidity = 61 % to IBM Watson
Published Temperature = 100 C Humidity = 90 % to IBM Watson
Published Temperature = 105 C Humidity = 63 % to IBM Watson
Published Temperature = 105 C Humidity = 79 % to IBM Watson
Published Temperature = 109 C Humidity = 71 % to IBM Watson
Published Temperature = 94 C Humidity = 98 % to IBM Watson
Published Temperature = 97 C Humidity = 97 % to IBM Watson
Published Temperature = 109 C Humidity = 93 % to IBM Watson
Published Temperature = 104 C Humidity = 99 % to IBM Watson
Published Temperature = 90 C Humidity = 72 % to IBM Watson
Published Temperature = 96 C Humidity = 94 % to IBM Watson
Published Temperature = 95 C Humidity = 77 % to IBM Watson
Published Temperature = 105 C Humidity = 70 % to IBM Watson
Published Temperature = 98 C Humidity = 73 % to IBM Watson
Published Temperature = 103 C Humidity = 76 % to IBM Watson
Published Temperature = 95 C Humidity = 60 % to IBM Watson
Published Temperature = 102 C Humidity = 95 % to IBM Watson
Published Temperature = 105 C Humidity = 73 % to IBM Watson
Published Temperature = 103 C Humidity = 63 % to IBM Watson
Published Temperature = 105 C Humidity = 80 % to IBM Watson
Published Temperature = 94 C Humidity = 68 % to IBM Watson
Published Temperature = 106 C Humidity = 87 % to IBM Watson
Published Temperature = 109 C Humidity = 67 % to IBM Watson
Published Temperature = 99 C Humidity = 92 % to IBM Watson
Published Temperature = 105 C Humidity = 62 % to IBM Watson
Published Temperature = 102 C Humidity = 98 % to IBM Watson
Published Temperature = 110 C Humidity = 86 % to IBM Watson
Published Temperature = 92 C Humidity = 85 % to IBM Watson
Published Temperature = 96 C Humidity = 66 % to IBM Watson
Published Temperature = 101 C Humidity = 84 % to IBM Watson
Published Temperature = 94 C Humidity = 60 % to IBM Watson
Published Temperature = 99 C Humidity = 66 % to IBM Watson
Published Temperature = 109 C Humidity = 72 % to IBM Watson
Published Temperature = 91 C Humidity = 80 % to IBM Watson
Published Temperature = 103 C Humidity = 67 % to IBM Watson
Published Temperature = 95 C Humidity = 70 % to IBM Watson
Ln: 54 Col: 0
Type here to search
30°C Haze
ENG IN
1:16 PM 11/20/2022

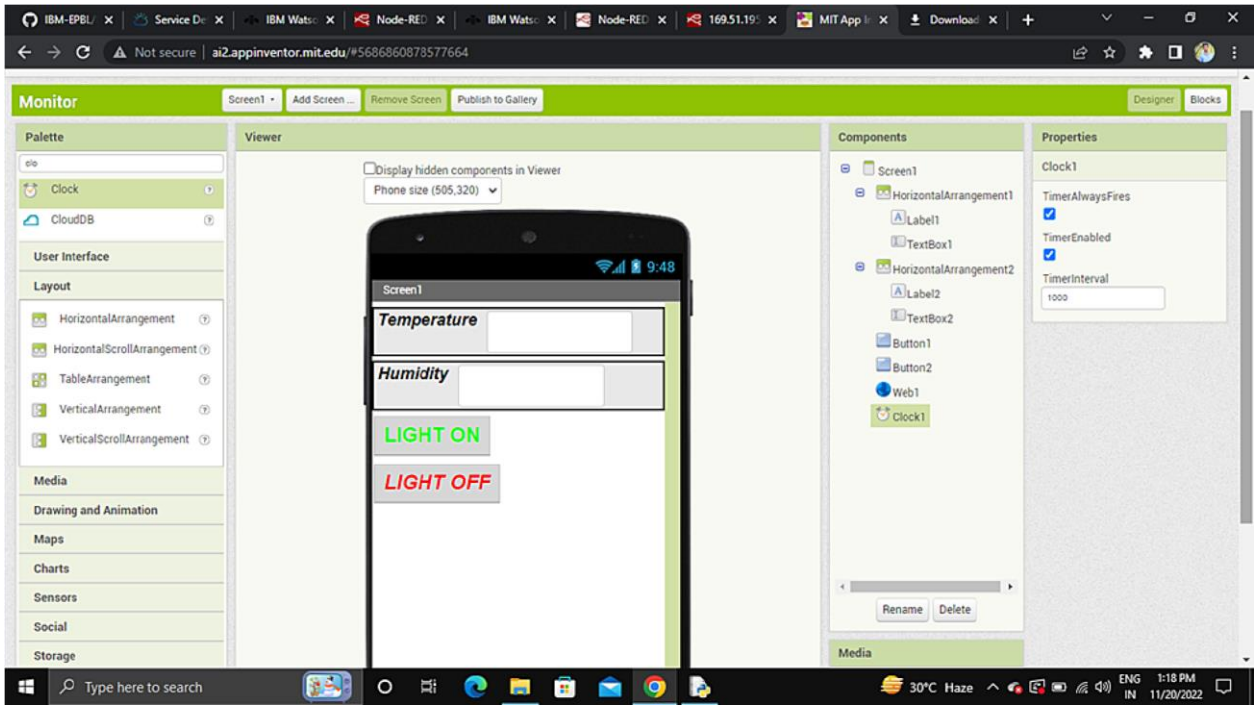

icoz5i8.internetofthings.ibmcloud.com/dashboard/devices/browse
IBM Watson IoT Platform
santhoshsanthosh8667@gmail.com
ID: ioz5i8
Browse
Action
Device Types
Interfaces
Add Device
123456
Disconnected
raspberrypi
Device
Nov 10, 2022 11:54 AM
Identity
Device Information
Recent Events
State
Logs
The recent events listed show the live stream of data that is coming and going from this device.
Event
Value
Format
Last Received
IoTSensor
{"temp":103,"Humid":67}
json
a few seconds ago
IoTSensor
{"temp":91,"Humid":80}
json
a minute ago
IoTSensor
{"temp":109,"Humid":72}
json
a minute ago
IoTSensor
{"temp":99,"Humid":66}
json
a minute ago
IoTSensor
{"temp":94,"Humid":60}
json
a minute ago
1 Simulation running
Type here to search
30°C Haze
ENG IN
1:17 PM 11/20/2022


Not secure | 169.51.195.229:31702/data
{"temp":95,"Humid":70}
Type here to search
30°C Haze
ENG
IN
1:18 PM
11/20/2022


Not secure | 169.51.195.229:31702/red/#flow/c4cd6c53c4a0fe70
Node-RED
Deploy
Flow 1
common
inject
debug
complete
catch
status
link in
link call
link out
comment
function
function
switch
temperature
Temperature
IBM IoT
connected
msg.payload
humidity
Humidity
[get] /data
Webpage
http
LIGHT ON
IBM IoT
connected
LIGHT OFF
msg.payload
[get] /command
command
http
debug
all nodes
all
msg.payload : Object
{ temp: 103, Humid: 67 }
11/20/2022, 1:16:23 PM node: 7f5c33c686ddc748
iot-2/type/raspberrypi/id/123456/evt/IoTSensor/fmt/json : msg.payload : number
67
11/20/2022, 1:16:24 PM node: 7f5c33c686ddc748
iot-2/type/raspberrypi/id/123456/evt/IoTSensor/fmt/json : msg.payload : number
103
11/20/2022, 1:16:32 PM node: 7f5c33c686ddc748
iot-2/type/raspberrypi/id/123456/evt/IoTSensor/fmt/json : msg.payload : Object
{ temp: 95, Humid: 70 }
11/20/2022, 1:16:33 PM node: 7f5c33c686ddc748
iot-2/type/raspberrypi/id/123456/evt/IoTSensor/fmt/json : msg.payload : number
70
11/20/2022, 1:16:34 PM node: 7f5c33c686ddc748
iot-2/type/raspberrypi/id/123456/evt/IoTSensor/fmt/json : msg.payload : number
95
Type here to search
30°C Haze
ENG
IN
1:17 PM
11/20/2022


Not secure | 169.51.195.229:31702/ui/#!/0?socketid=RpysVPj3sjVRdYo1AAAJ
Monitor
Monitor
Humidity
70
LIGHT ON
LIGHT OFF
Temperature
95
Type here to search
30°C Haze
ENG
IN
1:17 PM
11/20/2022


Not secure | ai2.appinventor.mit.edu/#5686860878577664
MIT APP INVENTOR
Projects
Connect
Build
Settings
Help
My Projects
View Trash
Guide
Report an Issue
English
tigertomjerry121@gmail.com
Monitor
Screen1
Add Screen ...
Remove Screen
Publish to Gallery
Designer
Blocks
Blocks
Viewer
Built-in
Control
Logic
Math
Text
Lists
Dictionaries
Colors
Variables
Procedures
Screen1
HorizontalArrangemer
Label1
TextBox1
HorizontalArrangemer
Label2
TextBox2
Rename
Delete
when Clock1 .Timer
do set Web1 . Url to " http://169.51.195.229:31702/data "
call Web1 .Get
when Web1 .GotText
url responseCode responseType responseContent
do set TextBox1 . Text to look up in pairs key " temperature "
pairs call Web1 .JsonTextDecode
jsonText get responseContent
notFound " not found "
set TextBox2 . Text to look up in pairs key " humidity "
pairs call Web1 .JsonTextDecode
jsonText get responseContent
notFound " not found "
when Button1 .Click
do set Web1 . Url to " http://169.51.195.229:31702/command "
call Web1 .Get
Show Warnings
Type here to search
30°C Haze
ENG
IN
1:17 PM
11/20/2022



9. Result:

The system can be taken as a small a empt in connec ng the exis ng primary gas detec on methods to a mobile pla orm integrated with IoT pla orms. The gases are sensed in an area of 1m radius of the rover and the sensor output data are con nuously 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 tes ng the assembled hardware. As the system operates outside the pipeline, the complica on of system maintenance and material selec on 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:

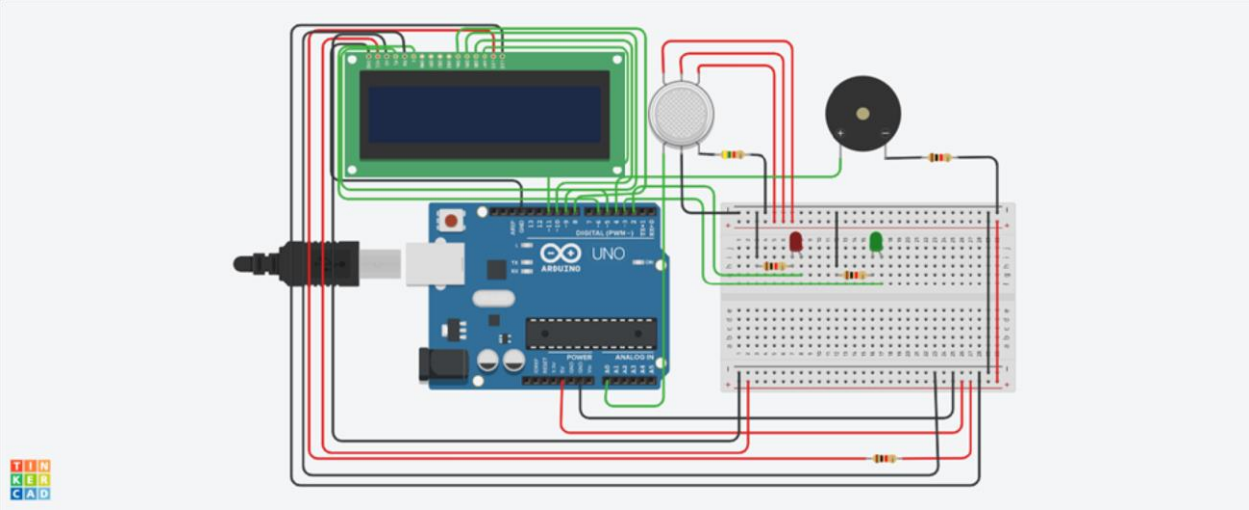
This gas leak detector system contains two features, this includes the SMS Gateway feature for only sending warning informa on regarding the gas leak to user, and the alarm for the warning alert. There is some improvement which can be applied for the future work, such as regarding the SMS Gateway, it need to enhance with feature such as no fying 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 no fica on for no fying the user whenever the sensor not working properly or not connected to the microcontroller for some cases, therefore, it is recommended to add this kind of features in the future work for be er refinement.

12. FUTURE SCOPE:

We propose to build the system using an MQ6 gas detec on 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:



13.2 Components:

The design of a sensor-based automa c gas leakage detector with an alert and control 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

13.3 Source Code:

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

int redled = 2; int
greenled = 3; int buzzer
= 4; int sensor = A0; int
sensorThresh = 100;

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.print(analogValue);
if(analogValue>sensorThresh)
 {
   digitalWrite(redled,HIGH);
digitalWrite(greenled,LOW);
tone(buzzer,1000,10000);
lcd.clear();     lcd.setCursor(0,1);
lcd.print("ALERT");
delay(1000);     lcd.clear();
lcd.setCursor(0,1);
lcd.print("PLEASE EVACUATE");
delay(1000);
 }
 else
 {
   digitalWrite(greenled,HIGH);
digitalWrite(redled,LOW);
noTone(buzzer);     lcd.clear();
lcd.setCursor(0,0);
lcd.print("SAFE");
delay(1000);     lcd.clear();
lcd.setCursor(0,1);
lcd.print("ALL CLEAR");
delay(1000);
 }
}
```

13.4 GITHUB:

Link: https://github.com/IBM-EPBL/IBM-Project-21040-1659770715

13.5 Project Demo Link :

1.Link:
https://drive.google.com/file/d/1lEBF9QZo0IJ3Egmh_plczpDOvLCAL2Oj/view?usp=sharing

2.Link:
https://www.tinkercad.com/things/6kjOhED25pb-gas-leakage-monitoring-and-alerting-system/editel?sharecode=GIe6clcCKeR3foOhvoj1MLQykM0BeNvGsToLSkJDxNA