1 INTRODUCTION

1.1 PROJECT OVERVIEW

Most of the fire-breakouts in industries are due to gas leaks. These cause dreadful damage to the equipment, human life leading to injuries, deaths, and environment. Currently available leakage detectors warn the people around using on-site alarms. So, this project proposes a leakage detector which sends the warning to the concerned people through SMS. This detector senses the presence of harmful gases particularly, LPG, Methane and Benzene. LPG and Methane gases catch fire easily resulting in blasts. Benzene is carcinogen effecting the health of workers, if inhaled in higher concentrations. Hence, detection of these gases is essential. This low cost project includes MQ6, MQ4 and MQ135 gas sensors which detect LPG, Methane and Benzene gas leaks respectively and uses ESP-32 as a Wi-Fi module. The concentration levels of the above mentioned gases are uploaded in the UBIDOTS cloud and the login details are included in the alert message so that the user can check, if needed. The prototype of the proposed system generates a sound alert using buzzer on detection of a dangerous leakage and sends an SMS to the concerned person using IFTTT web service. Different color LEDS are used to specify the gas leaked for example, RED LED indicates the presence of LPG.

1.2 PURPOSE

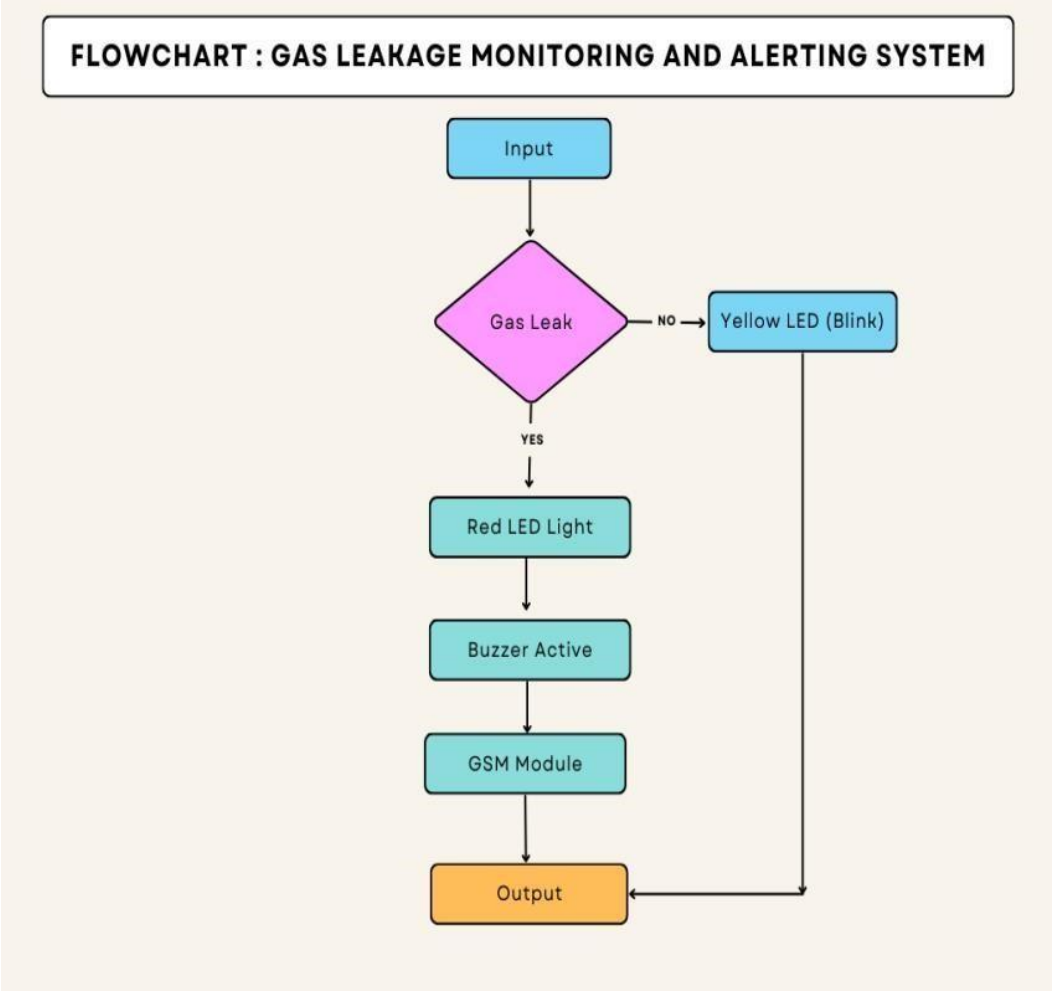
The aim of this project is to detect combustible, flammable and toxic gases and oxygen depletion. This type of device is widely used in industry and can be used in locations such as on oil rigs to monitor manufacturing processes and emerging technologies such as photovoltaic. They may be used in firefighting.

Gas leak detection is the process of identifying potentially hazardous gas leaks by sensors. Additionally, a visual identification can be done using a thermal camera These sensors usually employ an audible alarm to alert people when a dangerous gas has been detected.

2 LITERATURE SURVEY

2.1 EXISTING PROBLEM

Arduino UNO (Atmega-328) is the main unit of the system which performs thefollowing tasks. A signal conditioning of the Arduino UNO is done by output signal ofthe sensor, provided input to Arduino. The detection results displayed on LCD. Indicates the people of danger in work place, factory, home. Buzzer activity with beep(siren) sound is made. Also send alert SMS to the in charge of the plant number is saved in SIM card by using GSM modem. The SMS received depends uponthe leak of gas in the detection area of the sensor.



2.2 REFRENCES

[1] Shrivastava, A., Prabhaker, R., Kumar, R., & Verma, R. GSM based gas leakage detection system. International

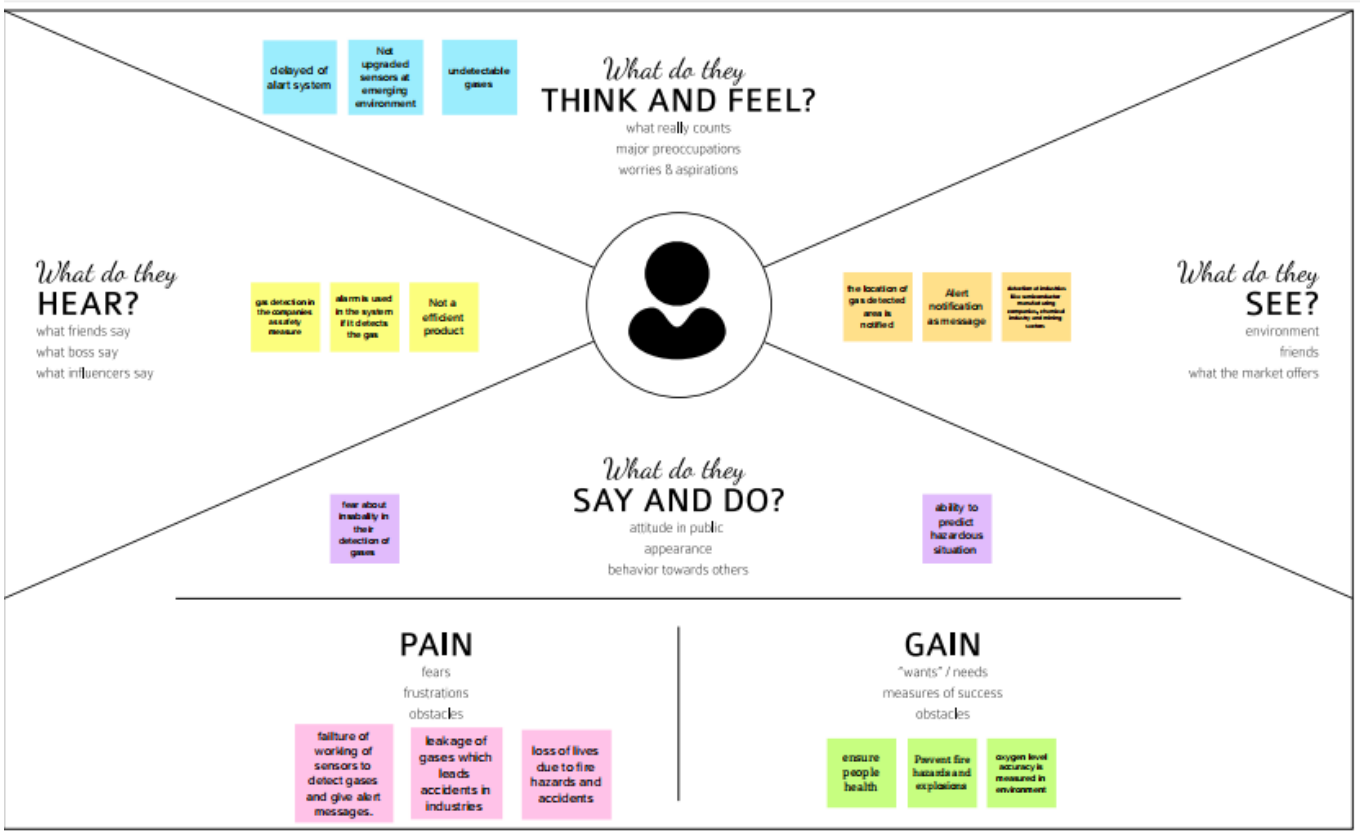
Journal of Emerging Trends in Electrical and Electronics (IJETEE-ISSN: 2320-9569), 2013; 3(2):42-45.
[2] Hema, L. K., Murugan, D., & Chitra, M. WSN based Smart system for detection of LPG and Combustible gases. In
National Conf. on Architecture, Software systems and Green computing-2013.
[3] Ramya, V., & Palaniappan, B. Embedded system for Hazardous Gas detection and Alerting. International Journal of Distributed and Parallel Systems (IJDPS), 2012; 3(3):287-300.
[4] 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).
[5] Jero, S. E., & Ganesh, A. B. 2011, March. PIC18LF4620 based customizable wireless sensor node to detect
hazardous gas pipeline leakage. In 2011 International Conference on Emerging Trends in Electrical and
Computer Technology (pp. 563-566). IEEE.
[6] Anusha, O., & Rajendra prasad, C. H. Experimental investigation on road safety system at crossings. International
Journal of Engineering and Advanced Technology, 2019; 8(2):214–218.
[7] Pravalika, V., & Rajendra Prasad, C. Internet of things based home monitoring and device control using Esp32.
International Journal of Recent Technology and Engineering, 2019; 8(1 Special Issue 4):58–62.
[8] Sanjay Kumar, S., Ramchandar Rao, P., & Rajendra Prasad, C. Internet of things based pollution tracking and
alerting system. International Journal of Innovative Technology and Exploring Engineering, 2019; 8(8):2242–
2245
[9] Deepak, N., Rajendra Prasad, C., & Sanjay Kumar, S. Patient health monitoring using IOT. International Journal of
Innovative Technology and Exploring Engineering, 2018; 8(2):454–457. https://doi.org/10.4018/978-1-5225-
8021-8.ch002
[10] Ramu, M., & Prasad, C. R. Cost effective atomization of Indian agricultural system using 8051 microcontrollers.
International journal of advanced research in computer and communication engineering, 2013; 2(7):2563-2566.

2.3 PROBLEM STATEMENT DEFINITION

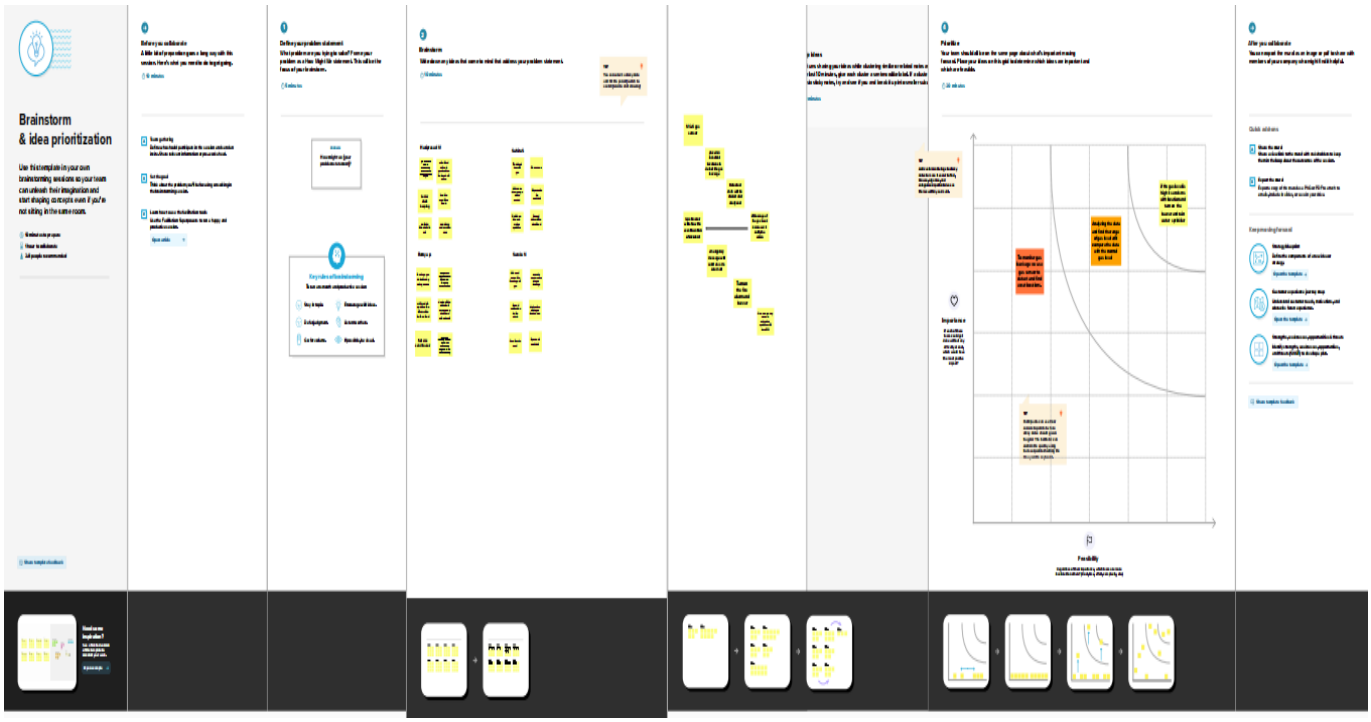
The issue from the environment and the air quality in industrial area is increased the alertness and responsibility regarding the public and workers health. Gas leakage leads to various accidents resulting into both financial loss as well as human injuries. In human's daily life, environment gives the most significant impact to their health issues. The risk of fires, explosion, suffocation, all are on their physical properties such flammability, toxicity etc. The number of based deaths due to the explosion of gas has been increased recently. Thus, a gas detector is invented to ease human on detecting the presence of those dangerous gases within an area to prevent disaster happen. 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 to detect the concentration of the gases present in the atmosphere to avoid hazardous consequences like fire breakouts for humans and workers in industries.

3 IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



3.2 IDEATION AND BRAINSTORMING



3.3 PROPOSED SOLUTION

Proposed Solution Template:

Project team shall fill the following information in proposed solution template.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	In industries gas leakage leads to major accidents and property damage which may affect both the employs as well as the industry.
2.	Idea / Solution description	This project helps the industries in watching the emission of harmful gases In many areas, the gas sensors are integrated to observe the gas outflow If in any space gas outflow is detected the admins are notified together with the placement In the net application, admins will read the sensing element parameters.
3.	Novelty / Uniqueness	In this project with the help of senor we will find the location, where the gas is leaked and turn on the buzzer,
4.	Social Impact / Customer Satisfaction	Inhaling leaked gas in an indoor space, such as your home can result in a lack of oxygen in the air and lead to hypoxia. That can, in turn, lead to severe headaches, fatigue, decreased vision, short breaths, and even loss of consciousness. It can be prevented by our project.

5.	Business Model (Revenue Model)	In industries, home, laboratory, hospital there is possibility of gas leakages. So this model will become an undeniable product or source for almost all companies . To prevent the gas leakage.
6.	Scalability of the Solution	As the product is offered with subscription service , further development in both software and hardware can be made.

3.4 PROBLEM SOLUTION FIT

CUSTOMER SEGMENT Industries and home	**CUSTOMER CONSTRAINTS** • If the power supply fails, the system won't work. • Failure of device/components may have dire consequences ,fatal accidents can occur.	**AVAILABLE SOLUTION** • In the method periodic check done by manually and partial sensing methodology is used. • It raises alarm whenever gas leaked or fire is detected at any place in a factory
JOBS-TO-BE-DONE / PROBLEMS • Needs to sense the gas leakage by sensors • Used to measure the precise value of gas leakage • Needs to send the SMS to the concerned person	**PROBLEM ROOT CAUSE** • Faulty piping and improper use of gas furnace, or appliance. • The leakage of gas may happen due to the human error and false chemical reaction. • Lack of service done in the gas valve.	**BEHAVIOUR** Focus their major concentrate on precautions/ safety measure to make avoid unwanted losses.
TRIGGERS Sudden loss of life of workers and damage to industrial property due to gas leakage **Emotions** BEFORE : insecure, instability AFTER : worry, Sorrow ,pain	**YOUR SOLUTION** By using gas , smoke and temperature sensor it would accurately sense the working environment and concentration of above gases are uploaded in the database and monitored. For additionally security auto air ventilation and automatic door locking mechanism is placed .	**CHANNELS of BEHAVIOUR** ONLINE : Information will be conveyed to concerned person by SMS to avoid risky situations. OFFLINE : If a concerned person does not aware about the gas leakage it is difficult to handle.

4 REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Online Payment for the service
FR-2	User Access	Access the details using web browser Access the details using mobile application
FR-3	User alert	Gets alert as an SMS message Gets alert alarm in the working area.

4.2 NON – FUNCTIONAL REQUIREMENTS

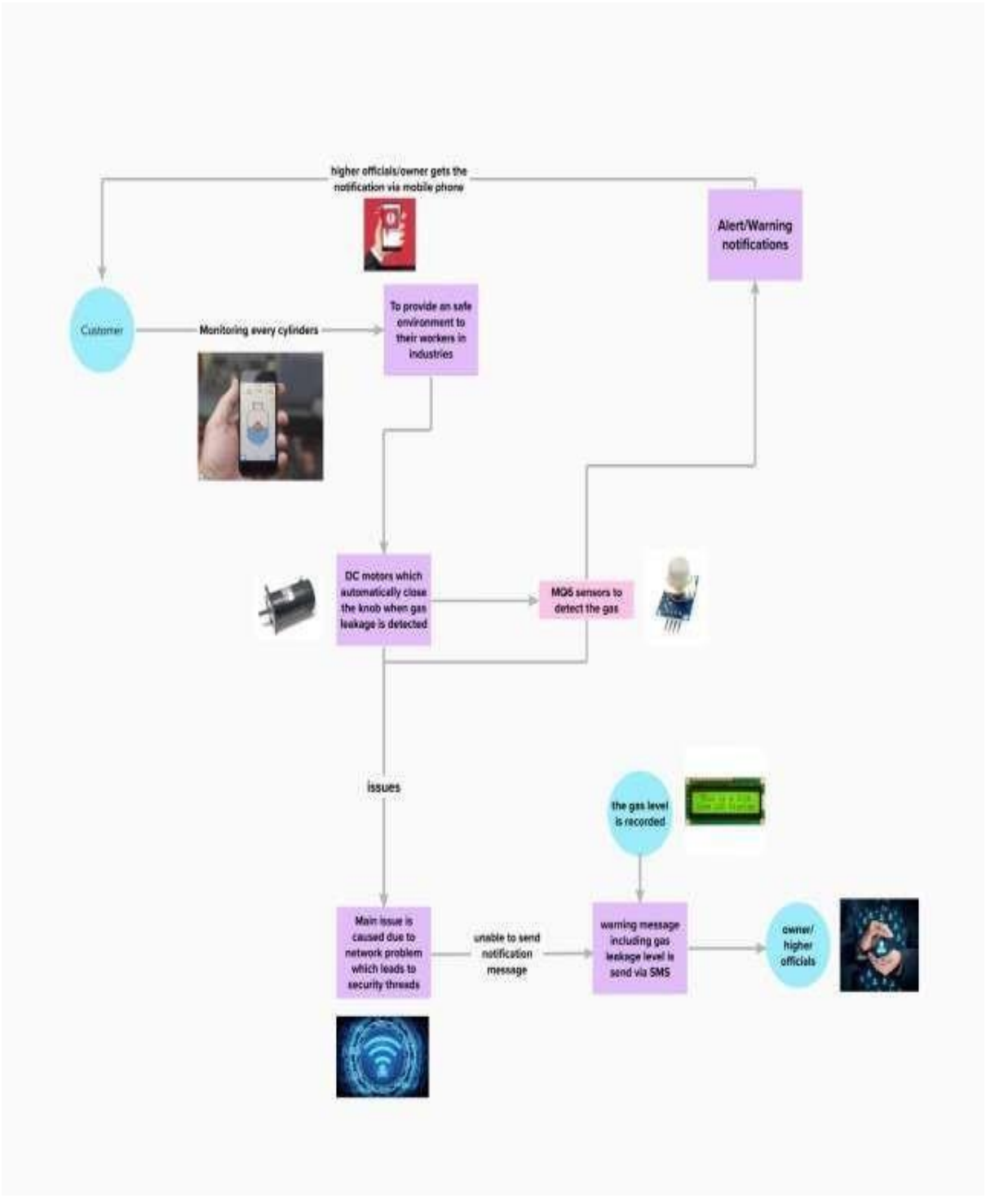
Following are the non-functional requirements of the proposed solution.

NFR No.	Non-Functional Requirement	Description
NFR-1	Usability	The device must be usable by the customer anywhere
NFR-2	Security	Data from the sensors are stored securely and away from other data
NFR-3	Reliability	Data can be retrieved anytime and no data is discarded without customer knowledge

NFR-4	Performance	No performance delay in case of large number of data or more parameters
NFR-5	Availability	The device doesn't fail even under harsh conditions. Device continues to send parameters, even after an alert situation.
NFR-6	Scalability	Device must be capable of measuring conditions even in a larger industry

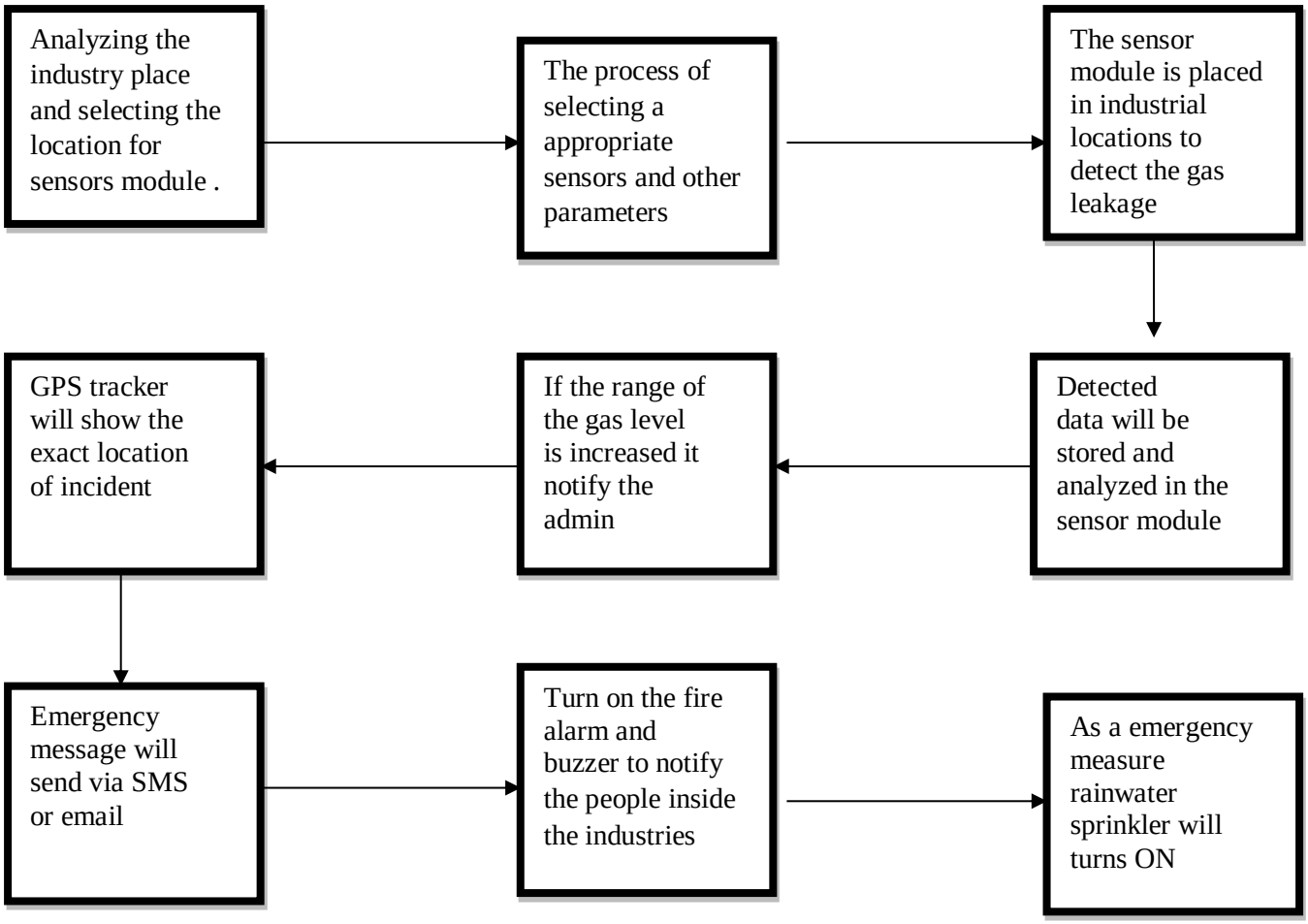
5 PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS


higher officials/owner gets the notification via mobile phone
Alert/Warning notifications
Customer
Monitoring every cylinders
To provide an safe environment to their workers in industries
DC motors which automatically close the knob when gas leakage is detected
MQ6 sensors to detect the gas
issues
the gas level is recorded
Main issue is caused due to network problem which leads to security threads
unable to send notification message
warning message including gas leakage level is send via SMS
owner/ higher officials

5.2 SOLUTION AND TECHNICAL ARCHITECTURE

Gas Leakage Monitoring & Alerting System for Industries has all the features to prevent the gas leakage and notify the authority. This model is built with help of sensors. In this model we use python programming language



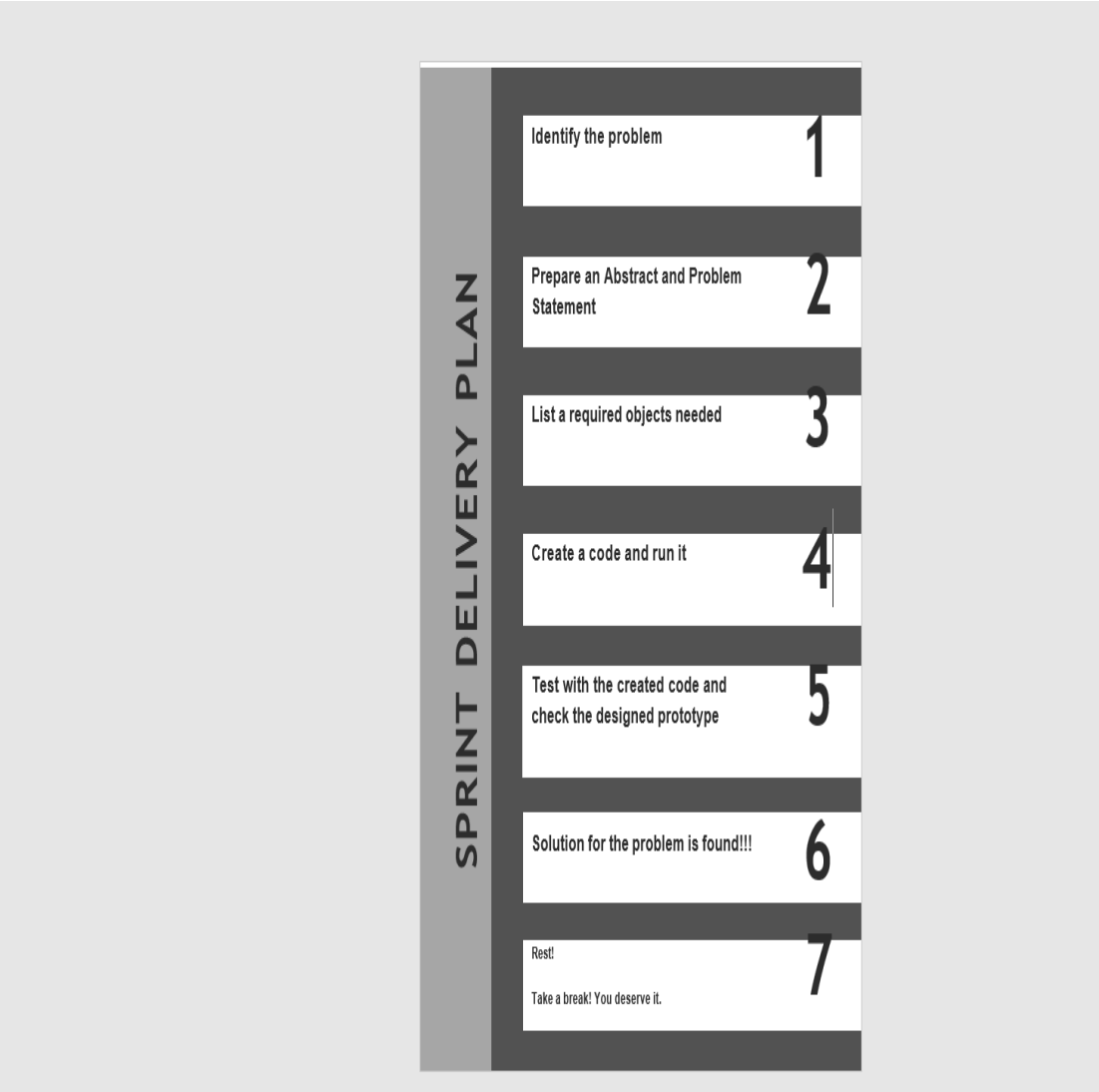
5.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority
Customer(family member/industry owner)	Registration	USN-1	As a user ,I can register for the device in the owners mobile application by entering my email and password	I can access my account / dashboard	High
Customer(higher authority)	confirmation	USN-2	As a user I will receive confirmation message via email and once I received I can install the device in the owners place	I can receive confirmation email & click confirm	High
Customer (fire service 101)	Safety measure register	USN-3	As a register I can register the application in owner/family members mobile phone	I can register & access the dashboard with Facebook Login	Low
Customer (mobile user)	Mobile application	USN-4	As a user I can register by mobile application	I can register for gas detection device with owners mobile number and the alert message will be send by SMS	Medium
Customer (credential)	Login	USN-5	As a user I can log into the device by entering email & password in the owner's mobile application	Mail address and passwords are default	High
Customer (Web user)	Notification	USN-7	As a user when there is a critical situation regarding gas explosion the alert notification will be received through GSM module	Alert message is sent to owners mobile as an SMS	High
Customer care Executive	Network Connectivity	USN-8	When there is a gas leakage is detected in the surrounding	Sensor detect the leakage and notifies the owner via message	High

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority
Administration	Accessing	USN-9	When there is an issue in accessing the device	Admin/Device operator's advice should be undertaken	High

6 PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation



6.2. SPRINT DELIVERY SOLUTION

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	6Days	1Nov2022	5Nov2022	20	52022
Sprint-2	20	6Days	6Nov2022	10Nov2022	20	10Nov2022
Sprint-3	20	6Days	11Nov2022	15Nov2022	20	15Nov2022
Sprint-4	20	6Days	16Nov 2022	19Nov2022	20	19Nov2022

7. CODING & SOLUTIONING

7.1. FEATURE 1

IOT device

Wokwi software

Iot Watson platform

Node red

Cloudant db

Web UI

7.2 FEATURE 2

Registration

Login

Verification

Check the sensors

Buzzer the alarm

Fast sms

DATA BASE SCHEME

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {font-family: Arial, Helvetica, sans-serif;}

/* Full-width input fields */
input[type=text], input[type=password] {
  width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  display: inline-block;
  border: 1px solid #ccc;
  box-sizing: border-box;
}

/* Set a style for all buttons */
button {
  background-color: #04AA6D;
  color: white;
  padding: 14px 20px;
  margin: 8px 0;
  border: none;
  cursor: pointer;
  width: 100%;
}

button:hover {
  opacity: 0.8;
}
```

```
/* Extra styles for the cancel button */
.cancelbtn {
  width: auto;
  padding: 10px 18px;
  background-color: #f44336;
}

/* Center the image and position the close button
*/
.imgcontainer {
  text-align: center;
  margin: 24px 0 12px 0;
  position: relative;
}

img.avatar {
  width: 20%;
  border-radius: 10%;
}

.container {
  padding: 16px;
}

span.psw {
  float: right;
  padding-top: 16px;
}

/* The Modal (background) */
.modal {
  display: none; /* Hidden by default */
  position: fixed; /* Stay in place */
```

```
  z-index: 1; /* Sit on top */
  left: 0;
  top: 0;
  width: 100%; /* Full width */
  height: 100%; /* Full height */
  overflow: auto; /* Enable scroll if needed */
  background-color: rgb(0,0,0); /* Fallback color */
  background-color: rgba(0,0,0,0.4); /* Black w/
opacity */
  padding-top: 60px;
}

/* Modal Content/Box */
.modal-content {
  background-color: #fefefe;
  margin: 5% auto 15% auto; /* 5% from the top,
15% from the bottom and centered */
  border: 1px solid #888;
  width: 80%; /* Could be more or less, depending
on screen size */
}

/* The Close Button (x) */
.close {
  position: absolute;
  right: 25px;
  top: 0;
  color: #000;
  font-size: 35px;
  font-weight: bold;
}

.close:hover,
.close:focus {
```

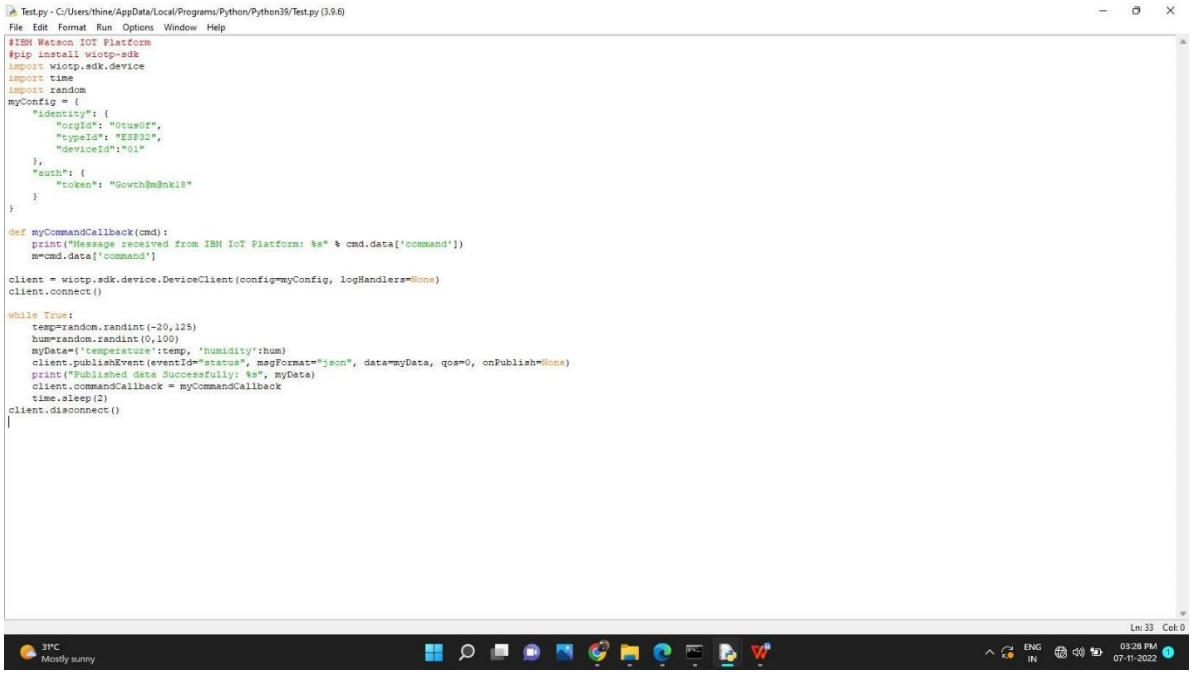
```
  color: red;
  cursor: pointer;
}

/* Add Zoom Animation */
.animate {
  -webkit-animation: animatezoom 0.6s;
  animation: animatezoom 0.6s
}

@-webkit-keyframes animatezoom {
  from {-webkit-transform: scale(0)}
  to {-webkit-transform: scale(1)}
}

@keyframes animatezoom {
  from {transform: scale(0)}
  to {transform: scale(1)}
}

/* Change styles for span and cancel button on
extra small screens */
@media screen and (max-width: 300px) {
  span.psw {
     display: block;
     float: none;
  }
  .cancelbtn {
     width: 100%;
  }

}
</style>
</head>
```

```
<body>

<h1 style="padding-top: 200px; text-align:
center;">Gas Leakage monitoring & Alerting
system for Industries</h1>

<button
onclick="document.getElementById('id01').style.d
isplay='block'" style="width:auto; margin-left:
600px;">Login</button>

<div id="id01" class="modal">

  <form  class="modal-content animate"
method="post">
    <div class="imgcontainer">
      <span
onclick="document.getElementById('id01').style.d
isplay='none'" class="close" title="Close
Modal">&times;</span>
      <img src="account.png" alt="Avatar"
class="avatar">
    </div>

    <div class="container">
      <label
for="uname"><b>Username</b></label>
      <input id="frm1" type="text"
placeholder="Enter Username" name="uname"
required>

      <label for="psw"><b>Password</b></label>
      <input type="password" placeholder="Enter
Password" name="psw" required>
```

```
      <button style="color: black"
onclick="window.location.href=('district.html')"
type="submit">signin</button>
      <label>
        <input type="checkbox" checked="checked"
name="remember"> Remember me
      </label>
    </div>

    <div class="container" style="background-
color:#f1f1f1">
      <button type="button"
onclick="document.getElementById('id01').style.d
isplay='none'"
class="cancelbtn">Cancel</button>
      <span class="psw">Forgot <a
href="#">password?</a></span>
    </div>
  </form>
</div>



<script>

function myFunction() {
  var x = document.getElementById("frm1");
  var text = "";
  var i;
  for (i = 0; i < x.length ;i++) {
    text += x.elements[i].value + "<br>";
  }
```

```
  document.getElementById("demo").innerHTML
= text;

}
// Get the modal action_page.php
var modal = document.getElementById('id01');

// When the user clicks anywhere outside of the
modal, close it
window.onclick = function(event) {
    if (event.target == modal) {
        modal.style.display = "none";
    }
}
</script>

</body>
</html>
```

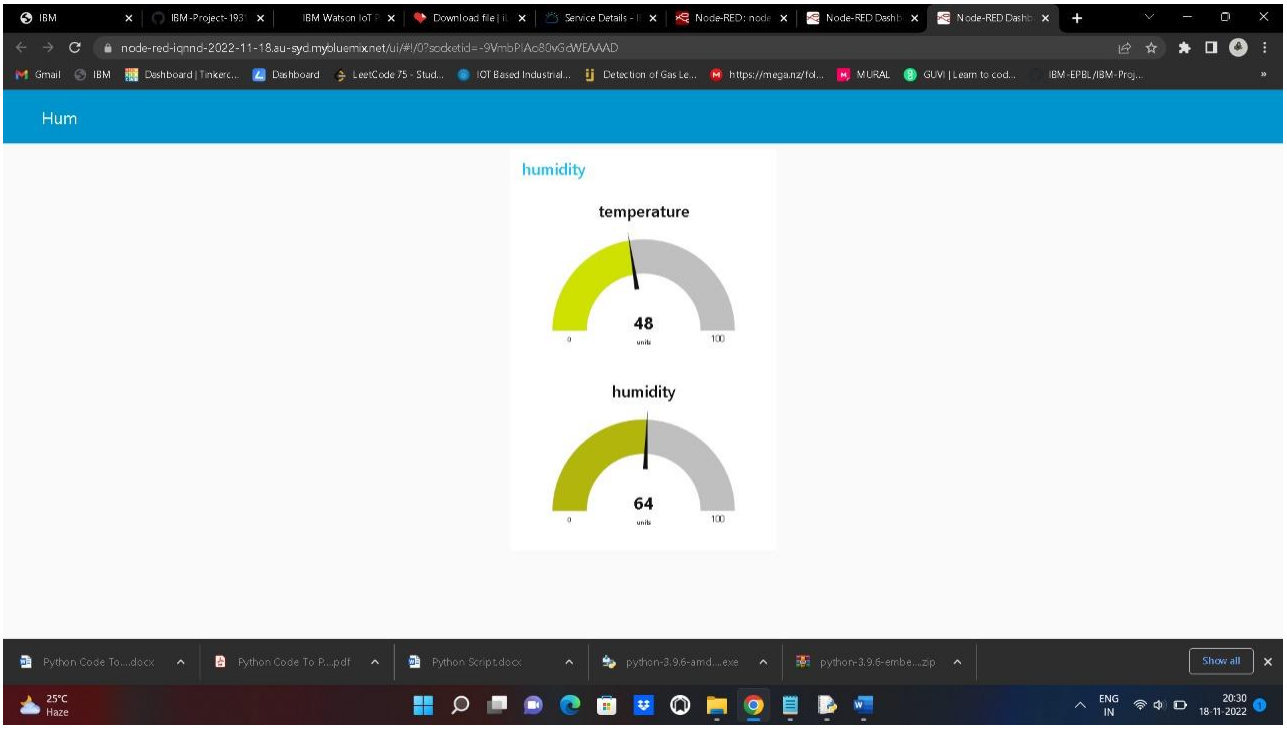
PYTHON CODE:

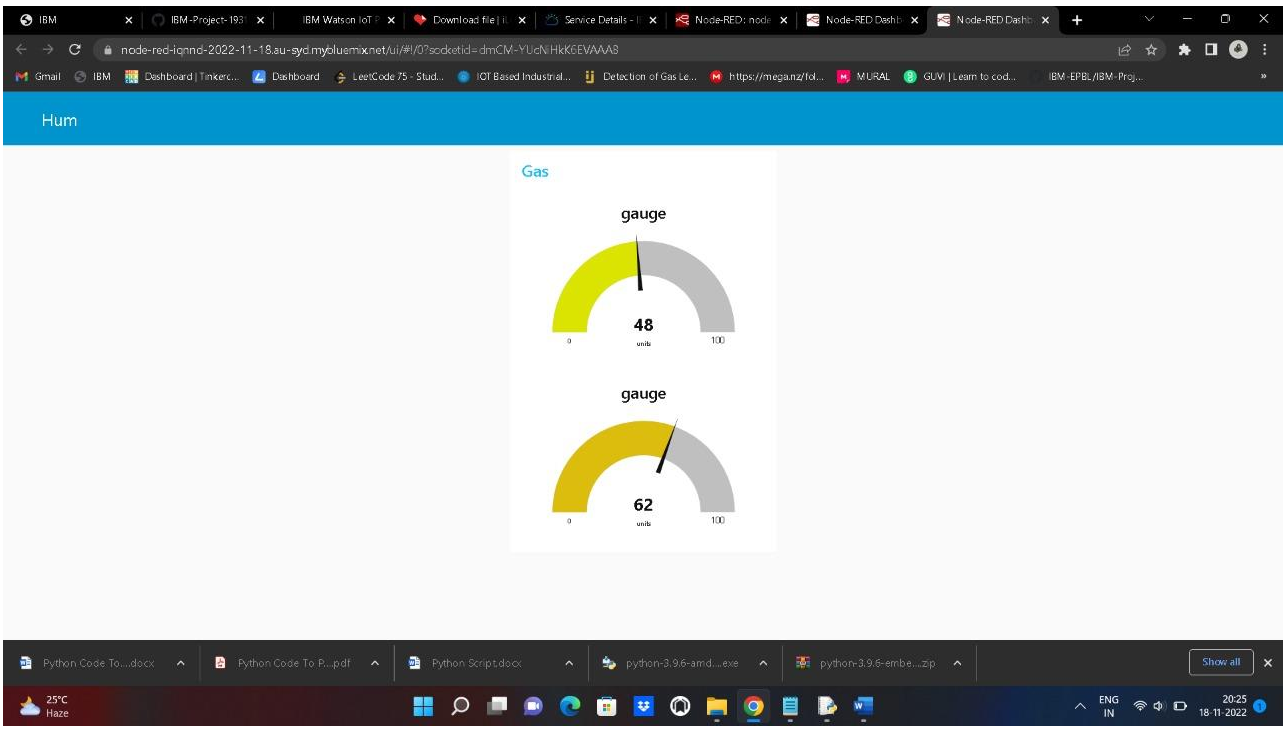

Test.py - C:/Users/thine/AppData/Local/Programs/Python/Python39/Test.py (3.9.6)
File Edit Format Run Options Window Help
#IBM Watson IOT Platform
#pip install wiotp-sdk
import wiotp.sdk.device
import time
import random
myConfig = {
"identity": {
"orgId": "0tus0f",
"typeId": "ESP32",
"deviceId":"01"
},
"auth": {
"token": "Gowth@m@nk18"
}
}
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(-20,125)
hum=random.randint(0,100)
myData={'temperature':temp, 'humidity':hum}
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()
Ln: 33 Col: 0
31°C
Mostly sunny
ENG
IN
03:28 PM
07-11-2022

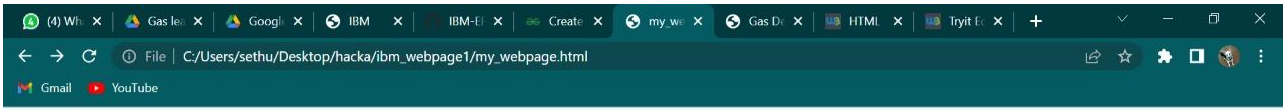
7 TESTING

S NO	TEST CASE	FEATURE	STEPS TO EXECUTE	EXPECTED RESULT	ACTUAL RESULT	EXECUTED BY
1	FUNCTIONAL	LOGIN	LOGIN TO EXECUTE BY FILLING THE DETAILS	CORRECT LOGIN CREDENTIALS	WORKING AS EXPECTED	HARIPRASAD
2	FUNCTIONAL	REGISTRATION	REGISTRATION THROUGH FORMS	REGISTRATION FORM TO BE FILLED AND DISPLAYED	WORKING AS EXPECTED	HARIPRASAD & RAMYA
3	FUNCTIONAL	WOKWI	TO DEVELOP THE IOT DEVICE AND CODE THE IOT DEVICE	SENSE THE DATA	WORKING AS EXPECTED	RAMYA
4	FUNCTIONAL	IBM WATSON	PUSH THE SENSED DATA FROM WOKWI	SENSED DATA IN IBM WATSON	WORKING AS EXPECTED	SUREKA & SUREKA
5	FUNCTIONAL	NODE RED	TO CONNECT WITH THE IBM WATSON AND THEN COLLECT THE SENSED DATA AND DISPLAY IN NODE RED DASHBOARD	VISUAL REPRESENTATION OF SENSED DATA IN NODE RED DASHBOARD	WORKING AS EXPECTED	SUHITHA
6	TESTING	TEST THE ENTIRE WORK	TO CHECK ALL THE MENTIONED TESTCASE ARE WORKING PROPERLY	TEST CASE ARE WOKING PROPERLY	WORKING AS EXPECTED	SUREKA

8 RESULTS

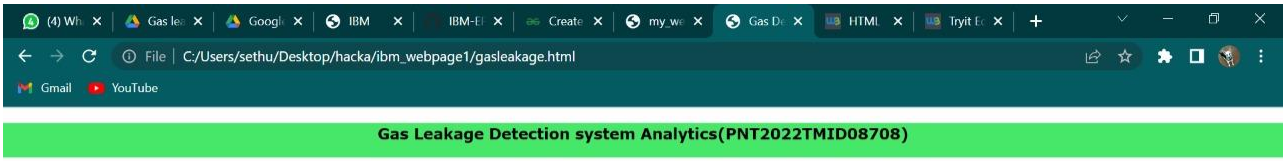

Hum
humidity
temperature
48
humidity
64


Hum
Gas
gauge
48
gauge
62



Gas Leakage monitoring & Alerting system for Industries

Login



Gas Leakage Detection system Analytics(PNT2022TMID08708)

28%

49%

Temperature : 28.00 C

Humidity : 49.50 g/m3

Gas : 287 ppm

9 ADVANTAGES AND DISADVANTAGES

ADVANTAGES

It is a very useful system to implement in the industries or plant facilities to avoid catastrophic explosions. With the help of a gas monitoring solution, you can successfully measure temperature and humidity in the atmosphere, which results in improved plant facilities and ensures employee health safety.

DISADVANTAGES

Poor stability and greater environmental impact; in particular, the selectivity of each sensor is not and the output parameters cannot be determined. Therefore, it should not be used in places where accurate measurement is required.

11.CONCLUSION

This work presents the design and implementation of gas leakage detection system. Various works on gas leakages detection system was reviewed and presented. I was discovered that some of the existent research don't takes in to considerations the cost effectiveness for the purpose of implementation of gas leakages detection at individual/domestic uses, and not easy to be further modified. This research work had advanced in knowledge as it included an embedded system to alert users via multiple mobile phones for further action to be taken when leakage is detected. The device detects gas leakage using a highly sensitive MQ-2 gas sensor to activate a buzzer that alert people of leakages, and also sent an SMS with the information "Gas Leakage Detected" from the SIM800 GSM Module as a backup to alert the appropriate authority or facility owner of a gas leakage.By using various software such as node red, IBM Watson,cloudant we are collecting this information in the cloud and in tha case of emergency it sends message to the user.

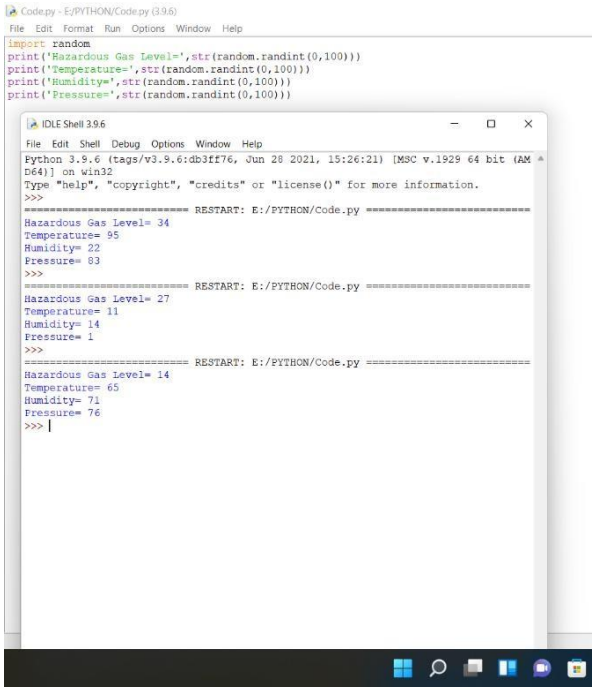
12. FUTURE SCOPE

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 in environment and hence nullify any major or minor hazard being caused due to them. We have used the IOT technology to make a Gas Leakage Detector for society which having Smart Alerting techniques involving sending text message to the concerned authority and an ability performing data analytics on sensor. This system will be able to detect the gas in environment using the gas sensors. This will prevent form the major harmful proble

13. APPENDIX

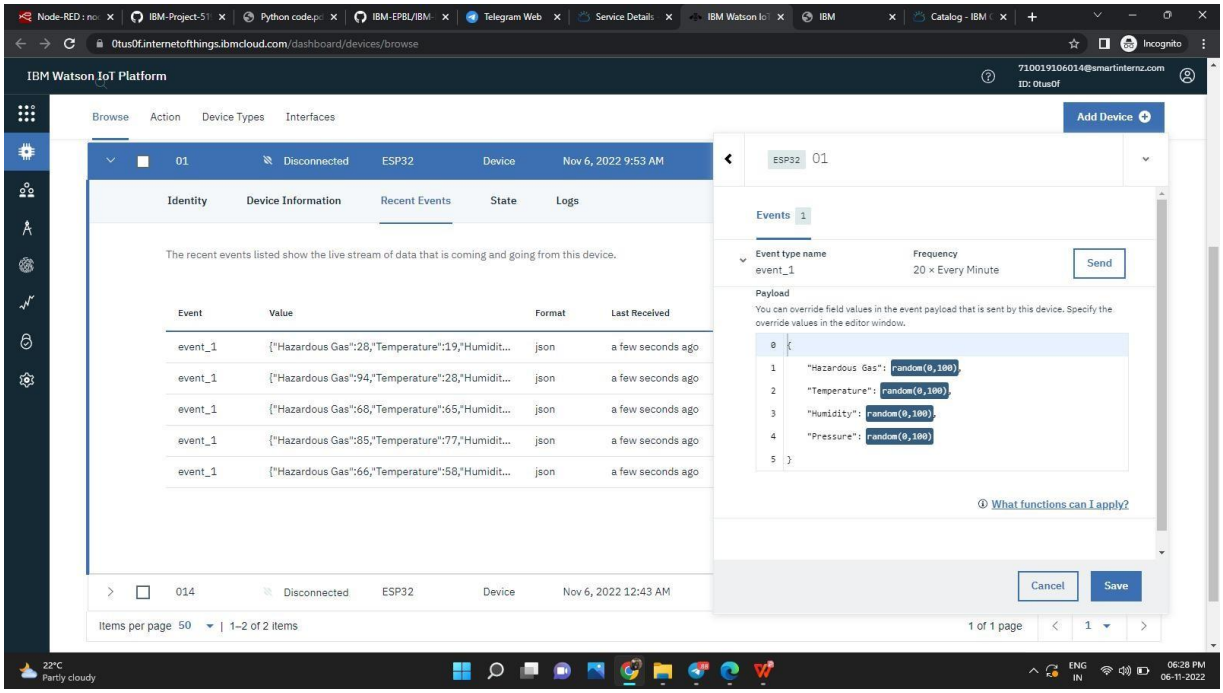
Python Code:

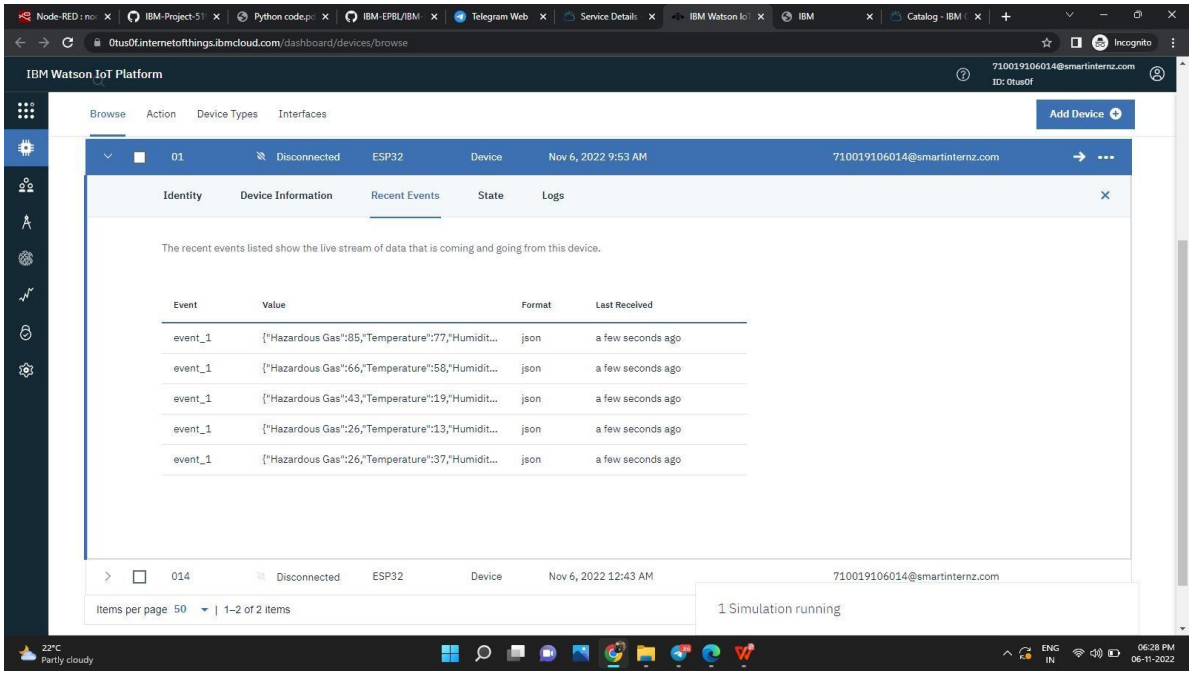
```
import random
print('Hazardous Gas Level=',str(random.randint(0,100)))
print('Temperature=',str(random.randint(0,100)))
print('Humidity=',str(random.randint(0,100)))
print('Pressure=',str(random.randint(0,100)))
```



Watson Code:

```
{
    "Hazardous Gas": random(0,100),
    "Temperature": random(0,100),
    "Humidity": random(0,100),
    "Pressure": random(0,100)
}
```





#IBM Watson IOT Platform

#pip install wiotp-sdk import

wiotp.sdk.device import time

import random

```
myConfig = {
    "identity": {
        "orgId": "0tus0f",
        "typeId": "ESP32",
        "deviceId":"01"
    },
    "auth": {
        "token": "Gowth@m@nk18"
    }
}

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(-
    20,125)
    hum=random.randint(0,100)
    myData={'temperature':temp, 'humidity':hum}
    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()
```

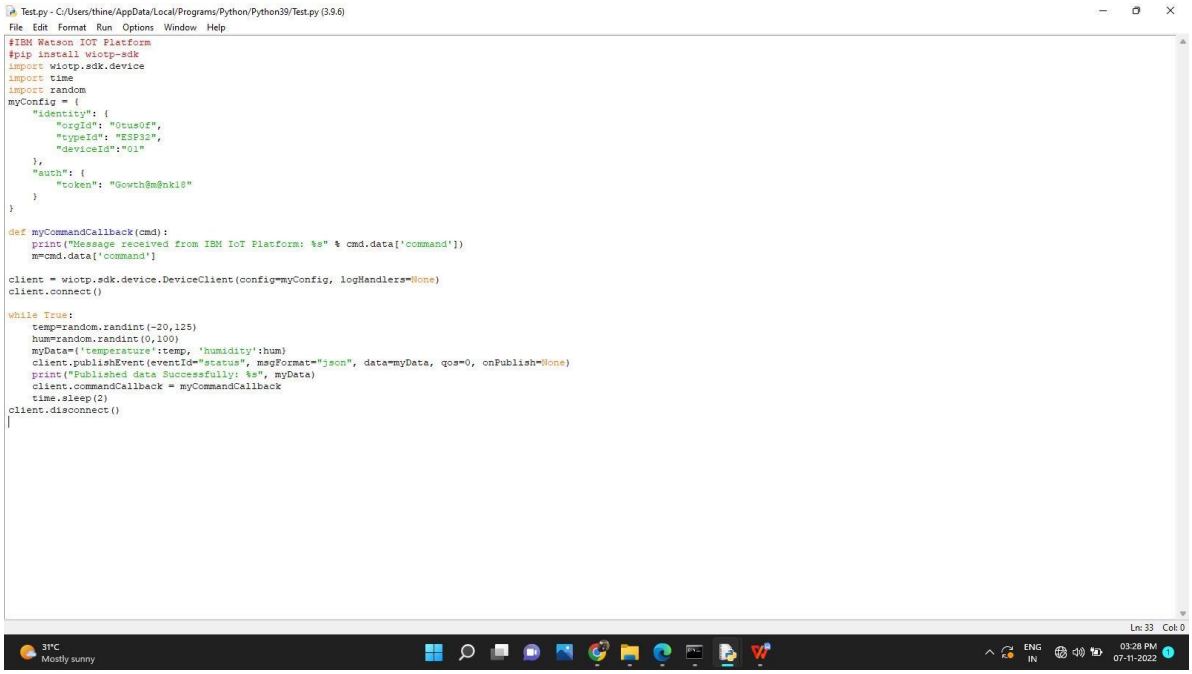
```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {font-family: Arial, Helvetica, sans-serif;}

/* Full-width input fields */
input[type=text], input[type=password] {
  width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  display: inline-block;
  border: 1px solid #ccc;
  box-sizing: border-box;
}

/* Set a style for all buttons */
button {
  background-color: #04AA6D;
  color: white;
  padding: 14px 20px;
  margin: 8px 0;
  border: none;
  cursor: pointer;
  width: 100%;
}

button:hover {
  opacity: 0.8;
}

/* Extra styles for the cancel button */
.cancelbtn {
  width: auto;
  padding: 10px 18px;
  background-color: #f44336;
}

/* Center the image and position the close button */
.imgcontainer {
  text-align: center;
```

```
  margin: 24px 0 12px 0;
  position: relative;
}

img.avatar {
  width: 20%;
  border-radius: 10%;
}

.container {
  padding: 16px;
}

span.psw {
  float: right;
  padding-top: 16px;
}

/* The Modal (background) */
.modal {
  display: none; /* Hidden by default */
  position: fixed; /* Stay in place */
  z-index: 1; /* Sit on top */
  left: 0;
  top: 0;
  width: 100%; /* Full width */
  height: 100%; /* Full height */
  overflow: auto; /* Enable scroll if needed */
  background-color: rgb(0,0,0); /* Fallback color */
  background-color: rgba(0,0,0,0.4); /* Black w/ opacity */
  padding-top: 60px;
}

/* Modal Content/Box */
.modal-content {
  background-color: #fefefe;
  margin: 5% auto 15% auto; /* 5% from the top, 15% from the bottom and centered */
  border: 1px solid #888;
  width: 80%; /* Could be more or less, depending on screen size */
}

/* The Close Button (x) */
.close {
  position: absolute;
  right: 25px;
  top: 0;
  color: #000;
  font-size: 35px;
  font-weight: bold;
}

.close:hover,
.close:focus {
  color: red;
  cursor: pointer;
}

/* Add Zoom Animation */
.animate {
  -webkit-animation: animatezoom 0.6s;
```

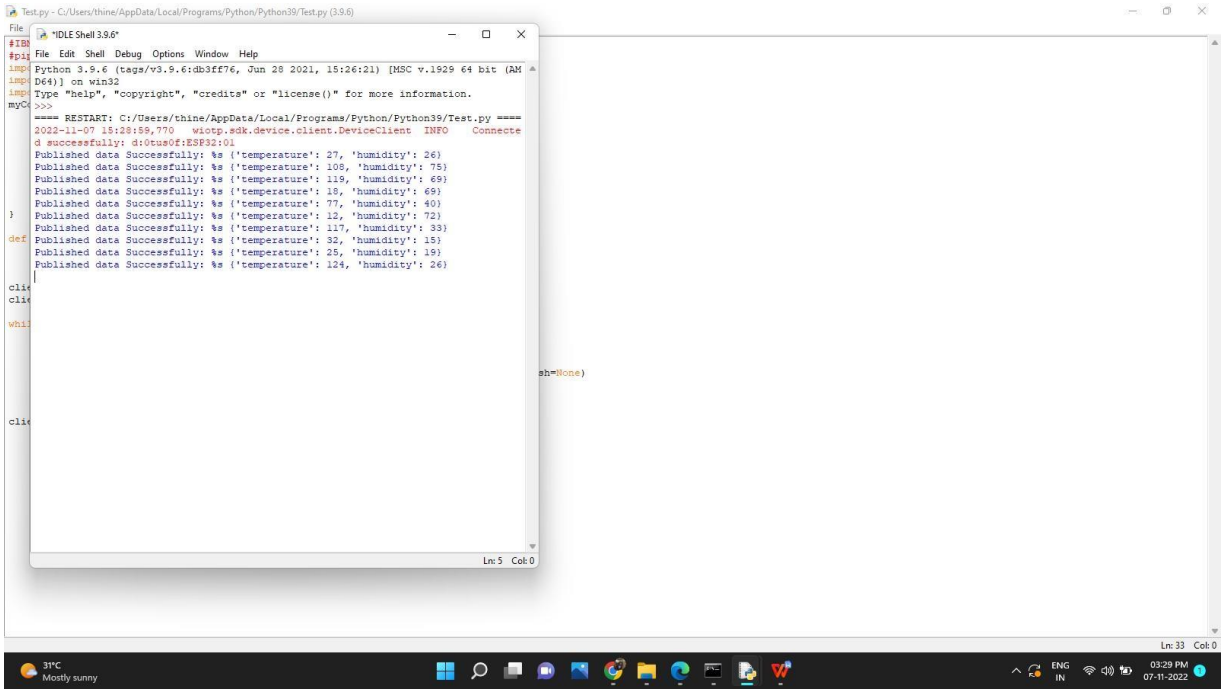
```
  animation: animatezoom 0.6s
}

@-webkit-keyframes animatezoom {
  from {-webkit-transform: scale(0)}
  to {-webkit-transform: scale(1)}
}

@keyframes animatezoom {
  from {transform: scale(0)}
  to {transform: scale(1)}
}

/* Change styles for span and cancel button on extra small screens */
@media screen and (max-width: 300px) {
  span.psw {
     display: block;
     float: none;
  }
  .cancelbtn {
     width: 100%;
  }

}
</style>
</head>
<body>

<h1 style="padding-top: 200px; text-align: center;">Gas Leakage monitoring & Alerting system for Industries</h1>

<button  onclick="document.getElementById('id01').style.display='block'" style="width:auto; margin-left: 600px;">Login</button>

<div id="id01" class="modal">

  <form  class="modal-content animate" method="post">
    <div class="imgcontainer">
      <span onclick="document.getElementById('id01').style.display='none'" class="close" title="Close Modal">&times;</span>
      <img src="account.png" alt="Avatar" class="avatar">
    </div>

    <div class="container">
      <label for="uname"><b>Username</b></label>
      <input id="frm1" type="text" placeholder="Enter Username" name="uname" required>

      <label for="psw"><b>Password</b></label>
      <input type="password" placeholder="Enter Password" name="psw" required>


      <button style="color: black" onclick="window.location.href=('district.html')" type="submit">signin</button>
      <label>
        <input type="checkbox" checked="checked" name="remember"> Remember me
      </label>
    </div>

    <div class="container" style="background-color:#f1f1f1">
      <button type="button" onclick="document.getElementById('id01').style.display='none'"
```

```
class="cancelbtn">Cancel</button>
      <span class="psw">Forgot <a href="#">password?</a></span>
    </div>
  </form>
</div>



<script>

function myFunction() {
  var x = document.getElementById("frm1");
  var text = "";
  var i;
  for (i = 0; i < x.length ;i++) {
    text += x.elements[i].value + "<br>";
  }
  document.getElementById("demo").innerHTML = text;

}
// Get the modal action_page.php
var modal = document.getElementById('id01');

// When the user clicks anywhere outside of the modal, close it
window.onclick = function(event) {
    if (event.target == modal) {
        modal.style.display = "none";
    }
}
</script>

</body>
</html>
```

PYTHON CODE:


Test.py - C:/Users/thine/AppData/Local/Programs/Python/Python39/Test.py (3.9.6)
File Edit Format Run Options Window Help
#IBM Watson IOT Platform
#pip install wiotp-sdk
import wiotp.sdk.device
import time
import random
myConfig = {
"identity": {
"orgId": "0tus0f",
"typeId": "ESP32",
"deviceId":"01"
},
"auth": {
"token": "Gowth@m@nk18"
}
}
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(-20,125)
hum=random.randint(0,100)
myData={'temperature':temp, 'humidity':hum}
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()
Ln: 33 Col: 0
31°C
Mostly sunny
ENG
IN
03:28 PM
07-11-2022

PYTHON OUTPUT:



Watson Cloud IBM: