A Project Report On

REMOTE GAS PIPELINE TUNNEL TEMPERATURE MONITORING SYSTEM

Submitted for

Professional Readiness Program for Innovation, Employability and Entrepreneurship

On

INTERNET OF THINGS

Submitted By:

1) **M. P. Shri Veena**

Reg. No: 963820106011

2) **A. S. Dharsini**

Reg. No: 963820106004

3) **V. Vibisha**

Reg. No: 963820106012

4) **Avani Anil**

Reg. No: 963820104010

Team ID : NM2023TMID10372

Date : 20.05.2023

Of

GOOD SHEPHERD COLLEGE OF ENGINEERING AND TECHNOLOGY

Maruthamparai, Kaliyal, Kanya Kumari District – 629101

CONTENTS

CHAPTER – 1

INTRODUCTION

1.1 Project Overview

This project is designed to reduce the pipeline explosions and leakges due to the damages in the gas pipeline. This helps in monitoring the temperature levels of the working space through the sensors, with information being sent to the mobile application over the cloud. With this facility the workers will be able to know when the temperature raises sharply through an alarm from the application.

1.2 Purpose

The gas is transported by means of gas pipeline to different places. In these gas pipelines, gas leakage can occur due to damage of pipeline. This occur due to increasing temperature inside the gas pipeline. This can cause pipeline explosions and leakage of gases. These gas leaks lead to severe human and property loss, causing damage to industries and may take several years to cope up with. Hence there is a need for a system to monitor the temperature of the gas pipeline which helps in reducing pipeline explosions and leakages.

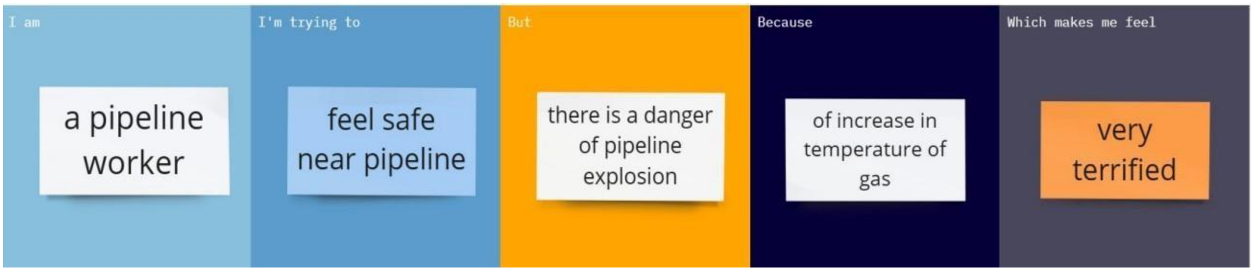
CHAPTER – 2

IDEATION & PROPOSED SOLUTION

2.1 Problem Statement Definition

Pipelines are prone to danger. When the temperature of the gas inside the pipeline increases, there is a chance of pipeline explosion. It can happen because of the expansion of gases inside the pipeline due to increased temperature.

So there is a need to continuously monitor the temperature inside the pipeline to decrease the risk of increasing temperature.



I am	I'm trying to	But	Because	Which makes me feel
a pipeline worker	feel safe near pipeline	there is a danger of pipeline explosion	of increase in temperature of gas	very terrified

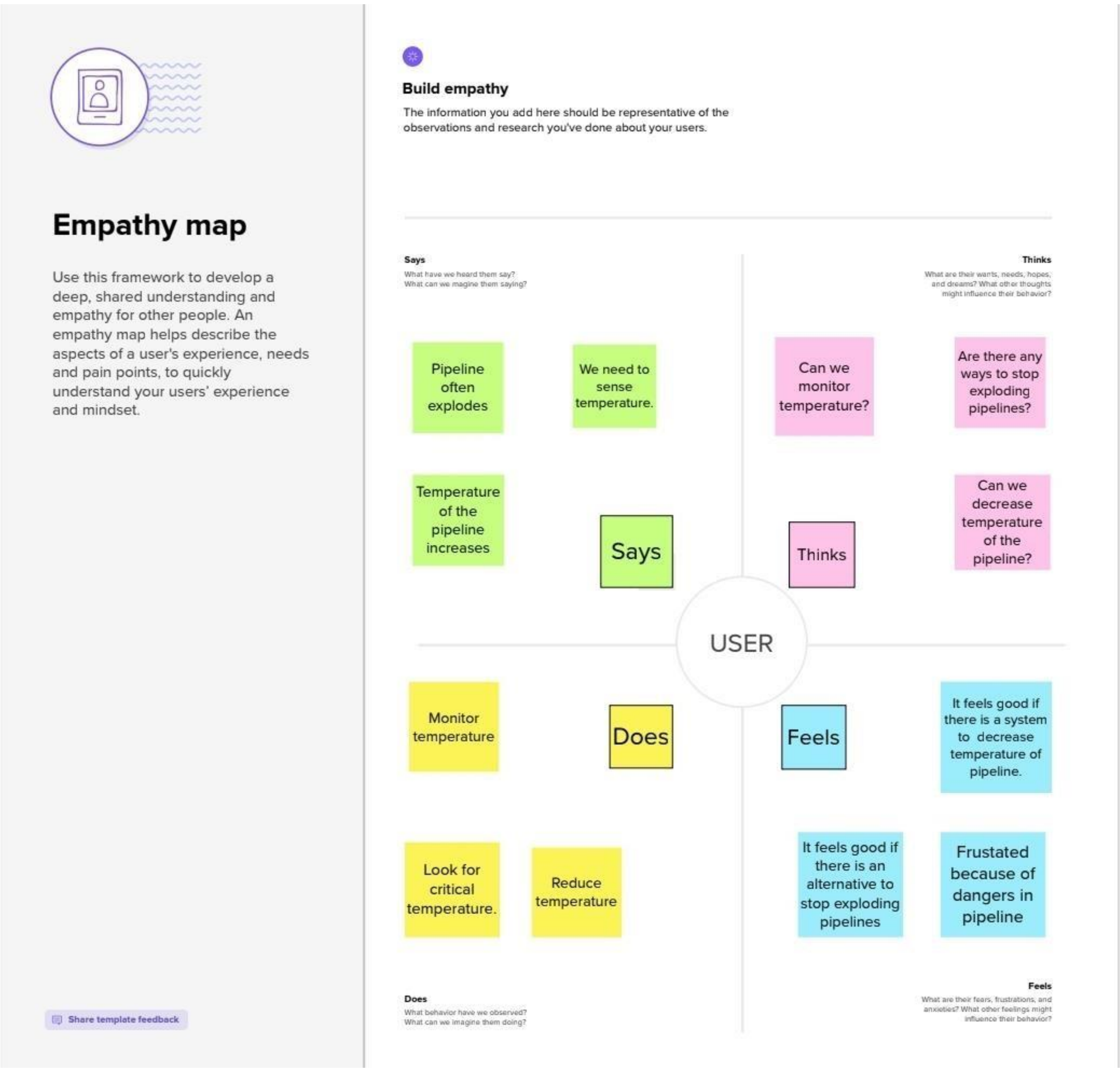
Fig No: 1 Problem Statement

Problem Statement(PS)	I am	I'm trying to	But	Because	Which makes me feel
PS-1	a pipeline worker	Feel safe near pipeline	there is a danger of pipeline explosion	of increase in temperature of gas	Very terrified
PS-2	a customer	Build a system to monitor the temperature of the pipeline	there are difficulties in building the system	Of complexity in structure of circuits	frustrated

Table No: 1 Problem Statement

2.2 Empathy Map Canvas

An empathy map is a collaborative tool teams can use to gain a deeper insight into their customers. It helps us to understand what the customer say, thinks, does and feel. It makes us understandable to the problems faced by the customers and the insights to solve them by providing solutions.



Empathy map

Use this framework to develop a deep, shared understanding and empathy for other people. An empathy map helps describe the aspects of a user's experience, needs and pain points, to quickly understand your users' experience and mindset.

Share template feedback

Build empathy

The information you add here should be representative of the observations and research you've done about your users.

Says

What have we heard them say?
What can we magine them saying?

Pipeline often explodes

We need to sense temperature.

Temperature of the pipeline increases

Says

Thinks

What are their wants, needs, hopes, and dreams? What other thoughts might influence their behavior?

Can we monitor temperature?

Are there any ways to stop exploding pipelines?

Can we decrease temperature of the pipeline?

Thinks

USER

Does

Monitor temperature

Look for critical temperature.

Reduce temperature

Does

What behavior have we observed?
What can we imagine them doing?

Feels

It feels good if there is a system to decrease temperature of pipeline.

It feels good if there is an alternative to stop exploding pipelines

Frustated because of dangers in pipeline

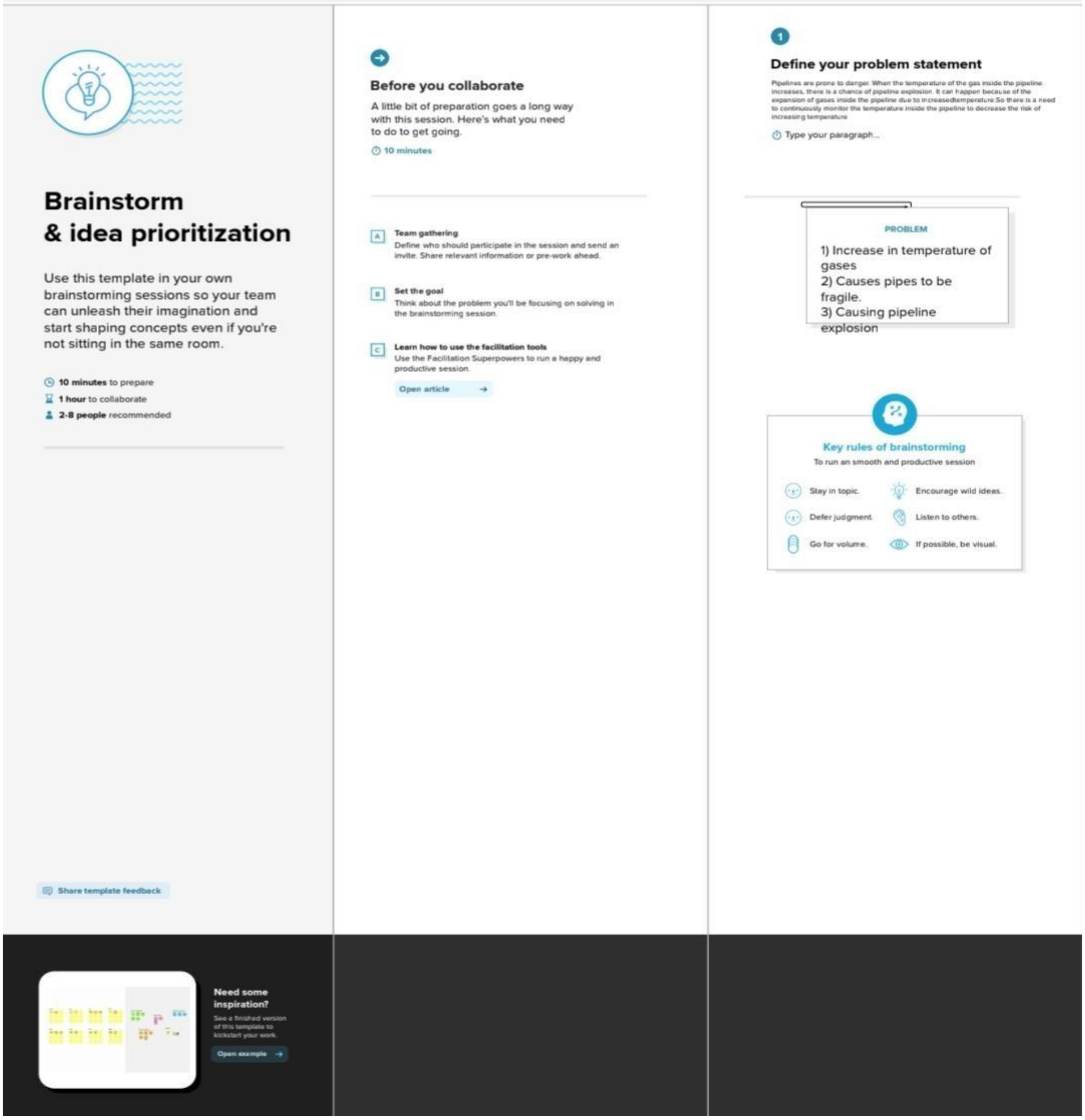
Feels

What are their fears, frustrations, and anxieties? What other feelings might influence their behavior?

Fig No: 2 Empathy Map

2.3 Ideation & Brainstorming

Step 1:Team Gathering, Collaboration and select the problem statement



Brainstorm
& idea prioritization

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

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

Share template feedback

Need some inspiration?

See a finished version of this template to kickstart your work.

Open example →

Before you collaborate

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

10 minutes

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

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

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

Open article →

1

Define your problem statement

Pipelines are prone to danger. When the temperature of the gas inside the pipeline increases, there is a chance of pipeline explosion. It can happen because of the expansion of gases inside the pipeline due to increasedtemperature.So there is a need to continuously monitor the temperature inside the pipeline to decrease the risk of increasing temperature

Type your paragraph...

PROBLEM

1) Increase in temperature of gases
2) Causes pipes to be fragile.
3) Causing pipeline explosion

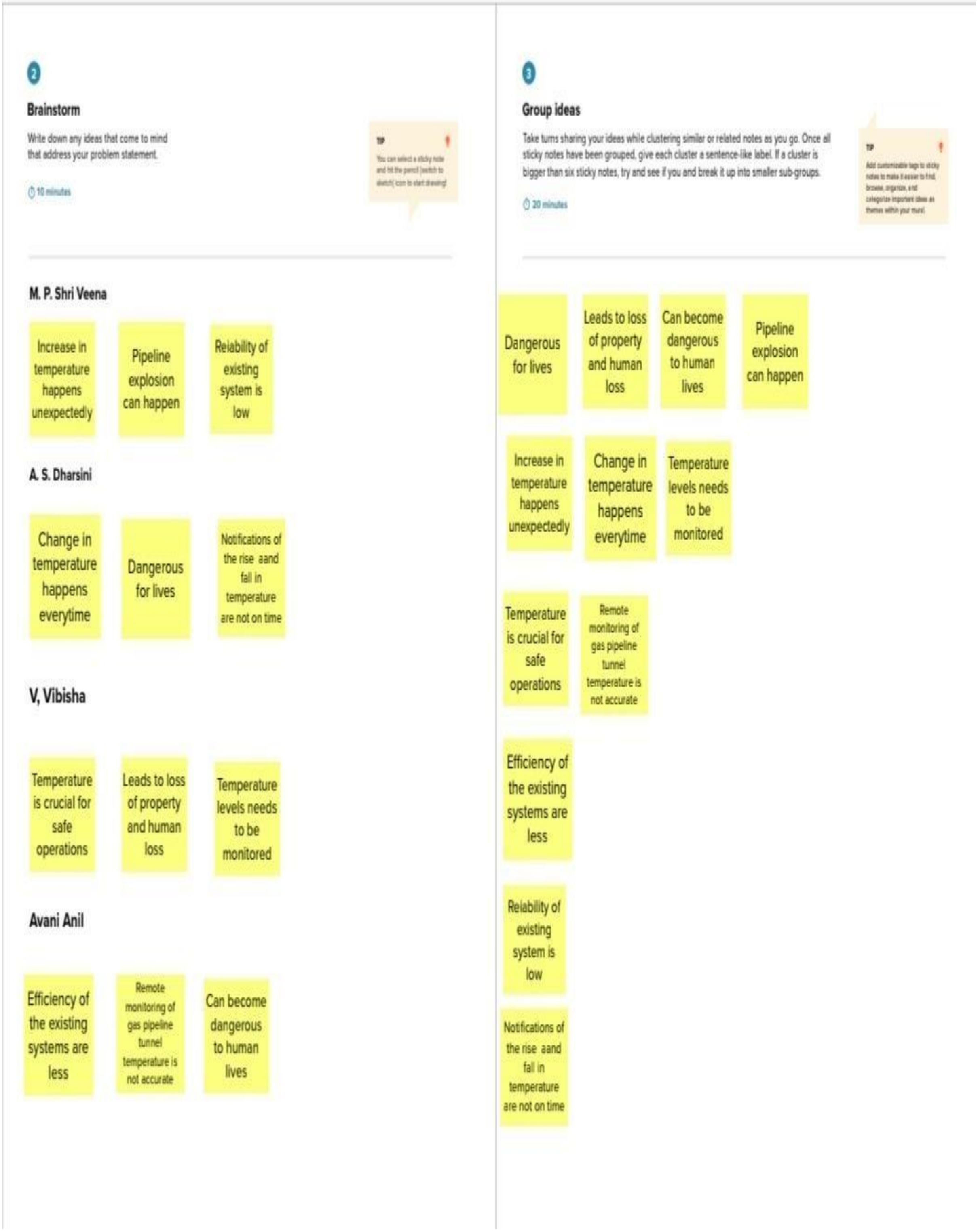
Key rules of brainstorming

To run an smooth and productive session

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

Fig No: 3 a) Brainstorm and idea prioritization

Step 2: Brainstorm, Idea Listing and Grouping



2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP

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

M. P. Shri Veena

- Increase in temperature happens unexpectedly
- Pipeline explosion can happen
- Reliability of existing system is low

A. S. Dharsini

- Change in temperature happens everytime
- Dangerous for lives
- Notifications of the rise aand fall in temperature are not on time

V, Vibisha

- Temperature is crucial for safe operations
- Leads to loss of property and human loss
- Temperature levels needs to be monitored

Avani Anil

- Efficiency of the existing systems are less
- Remote monitoring of gas pipeline tunnel temperature is not accurate
- Can become dangerous to human lives

3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

TIP

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

- Dangerous for lives
- Leads to loss of property and human loss
- Can become dangerous to human lives
- Pipeline explosion can happen

- Increase in temperature happens unexpectedly
- Change in temperature happens everytime
- Temperature levels needs to be monitored

- Temperature is crucial for safe operations
- Remote monitoring of gas pipeline tunnel temperature is not accurate

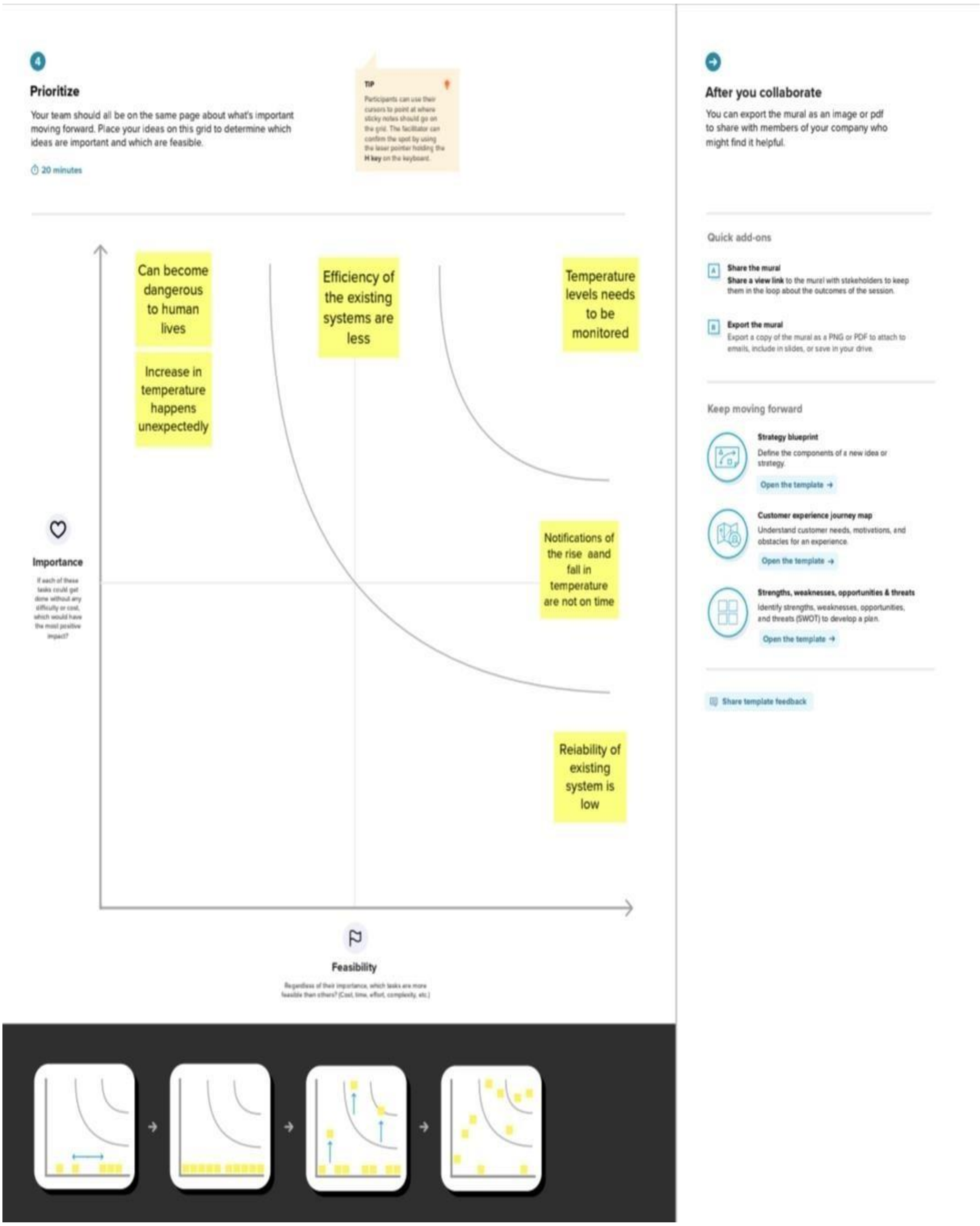
- Efficiency of the existing systems are less

- Reliability of existing system is low

- Notifications of the rise aand fall in temperature are not on time

Fig No: 3 b) Brainstorm and idea listing

Step 3: Idea Prioritization



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

TIP

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

Can become dangerous to human lives

Increase in temperature happens unexpectedly

Efficiency of the existing systems are less

Temperature levels needs to be monitored

Notifications of the rise aand fall in temperature are not on time

Reiability of existing system is low

Importance

If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

Feasibility

Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

After you collaborate

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

Quick add-ons

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

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

Keep moving forward

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

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

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

Share template feedback

Fig No: 3 c) Idea prioritization

2.4 Proposed Solution

S. No	Parameter	Description
1.	Problem Statement	Pipelines are prone to danger. When the temperature of the gas inside the pipeline increases, there could happen many consequences like pipeline explosion, leakage of gases, etc. It happens because of the expansion of gases inside the pipeline due to increase in temperature. So there is a need for a system to monitor the temperature of the gases inside the pipeline and leakage of the gases
2.	Idea / Solution Description	The temperature can be sensed by DHT22. The sensed temperature can be given as an output by ESP32. Thus connecting the ESP32, DHT22 with a potentiometer can monitor the temperature, humidity and leakage of gases.
3.	Novelty / Uniqueness	The uniqueness of the project is the temperature, humididty and leakage threshold can be seen through application developed by Node-Red
4.	Social Impact / Customer Satisfaction	When this project is implemented, huge amount of pipeline explosions and gas leakage can be reduced.
5.	Business Model (Revenue Model)	The business model is the outcome based model. This is because the revenue is earned from the outcome if there is no loss due to gas leakage and pipeline damage.
6.	Scalability of the solution	The project can be operated upto +125 degree Celsius. Hence the scalability of the project is very high.

Table No: 2 Proposed Solution

CHAPTER – 3

REQUIREMENT ANALYSIS

3.1 Functional requirement

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement	Sub Requirement
FR-1	Circuit	DSP22 ESP32 Potentiometer
FR-2	Code	Importing libraries IBM Credentials Creating instance Wi-Fi connection
FR-3	IBM Cloud	IBM Cloud account IBM Watson login
FR-4	Node-Red	Node JS Node-Red installation API Key IBM Node connection

Table No: 3 Functional requirements

3.2 Non-Functional requirements

Following are the non-functional requirements of the proposed solution

NFR No.	Non Functional Requirement	Description
NFR-1	Usability	People can use this product easily. The user needs to have only the knowledge of English to understand the values of temperature, humidity and leakage threshold.
NFR-2	Security	The system provided as solution is completely secure. It requires registering in IBM and Node-Red platform which are highly secure.
NFR-3	Reliability	Time required to get the values of temperature, humidity and leakage threshold values is very low and within milliseconds. Hence reliability is high.
NFR-4	Performance	The performance is high since the accuracy of the values obtained is high. The solution gives the exact values.
NFR-5	Availability	This device can be used by every user who have IBM account.
NFR-6	Scalability	The performance is high since high precision sensor is used.

Table No: 4 Non-Functional requirements

CHAPTER – 4

PROJECT DESIGN

4.1 Data Flow Diagrams

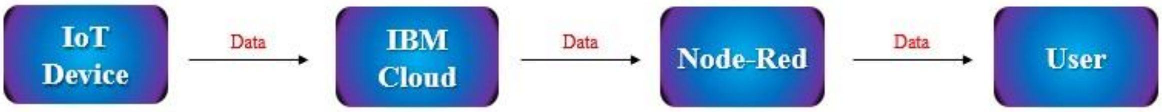


Fig No: 4 Data Flow Diagram

4.2 Solution & Technical Architecture

Solution Architecture:

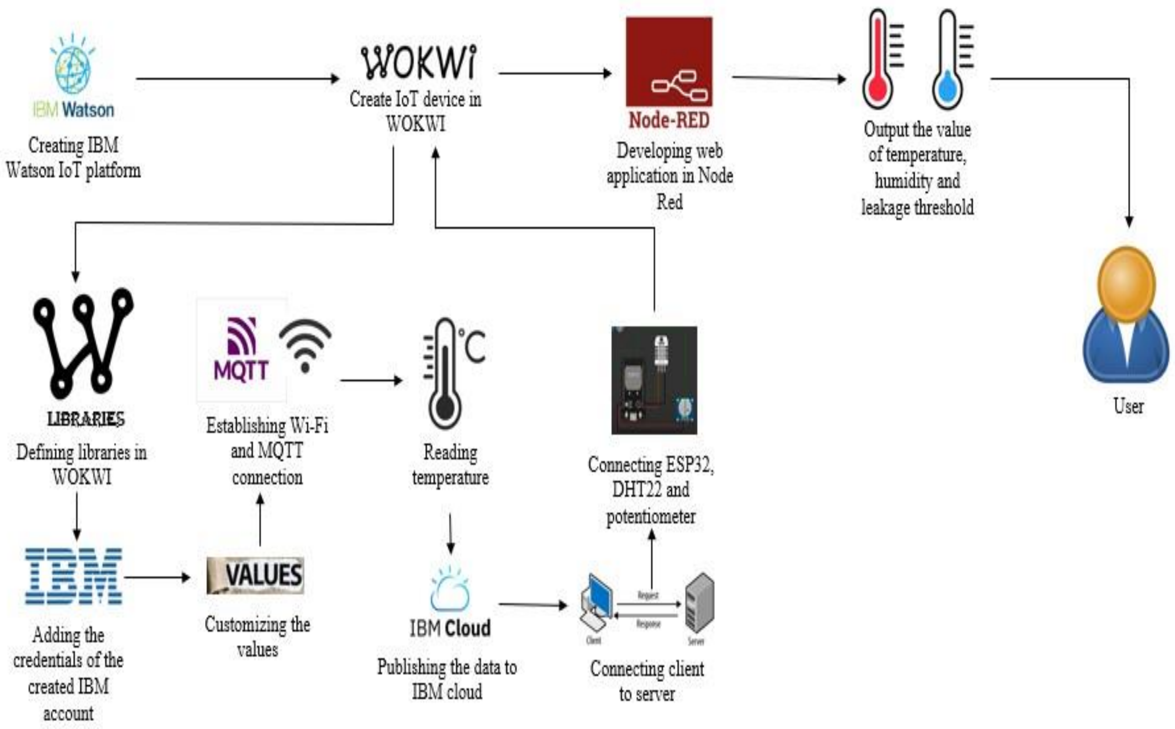

IBM Watson
Creating IBM Watson IoT platform
WOKWI
Create IoT device in WOKWI
Node-RED
Developing web application in Node Red
Output the value of temperature, humidity and leakage threshold
User
LIBRARIES
Defining libraries in WOKWI
MQTT
Establishing Wi-Fi and MQTT connection
°C
Reading temperature
Connecting ESP32, DHT22 and potentiometer
IBM
Adding the credentials of the created IBM account
VALUES
Customizing the values
IBM Cloud
Publishing the data to IBM cloud
Request
Response
Client
Server
Connecting client to server

Fig No: 5 Solution Architecture

Technical Architecture:

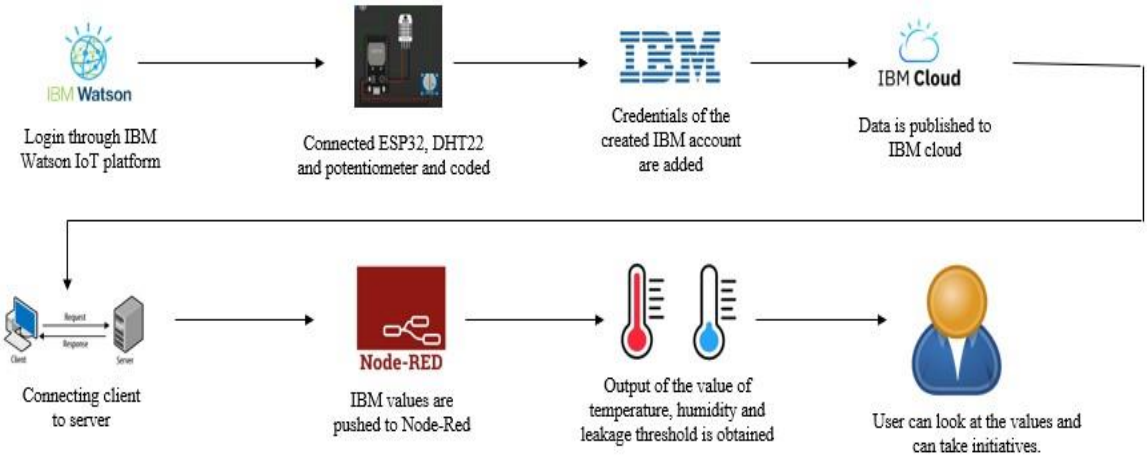

IBM Watson
Login through IBM Watson IoT platform
Connected ESP32, DHT22 and potentiometer and coded
IBM
Credentials of the created IBM account are added
IBM Cloud
Data is published to IBM cloud
Request
Response
Client
Server
Connecting client to server
Node-RED
IBM values are pushed to Node-Red
Output of the value of temperature, humidity and leakage threshold is obtained
User can look at the values and can take initiatives.

Fig No: 6 Technical Architecture

Components & Technology

SI. No	Components	Description	Technology
1	IoT Device	Connection of components such as potentiometer, ESP32, DHT22 virtually	WOKWI
2	Application Logic-1	Logic for a process in the application	IBM Watson
3	Cloud Database	Database Service on Cloud	IBM Cloud
4	Output Visualizer	Visualization tool	Node-Red

Table No: 5 Components & Technology

Application Characteristics:

S.No	Characteristics	Description	Technology
1	Open-Source Frameworks	For programming the IoT device, C++ open source framework is used	C++ in WOKWI
2	Security Implementations	During logging in the IBM Cloud platform, verification code is sent to respective user. Every value are encrypted.	Encryption, Verification by code
3	Scalable Architecture	To operate this device in a seamless way, Wi-Fi technology is used by importing the library.	Wi-Fi
4	Availability	This device can be used by every user who have IBM account.	IBM Server
5	Performance	The performance is high since high precision sensor is used.	DHT22 Sensor, ESP32

Table No: 6 Application Characteristics

4.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Team Member
Customer (Pipeline Worker)	Circuit	USN-1	As a user, I can monitor the circuit used in IoT device	I can view the circuit	High	V. Vibisha
	Code	USN-2	As a user, I can use the code used in IoT device	I can view the code	High	A. S. Dharsini
	IBM Cloud	USN-3	As a user, I can publish the data to cloud	I can view the data in IBM cloud	High	Avani Anil
	Node-Red	USN-4	As a user, I can view the output in Node-Red	I can view the data in Node-Red	High	M. P. Shri Veena

Table No: 7 User Stories

CHAPTER – 5

CODING & SOLUTIONING

5.1 WOKWI Connections:

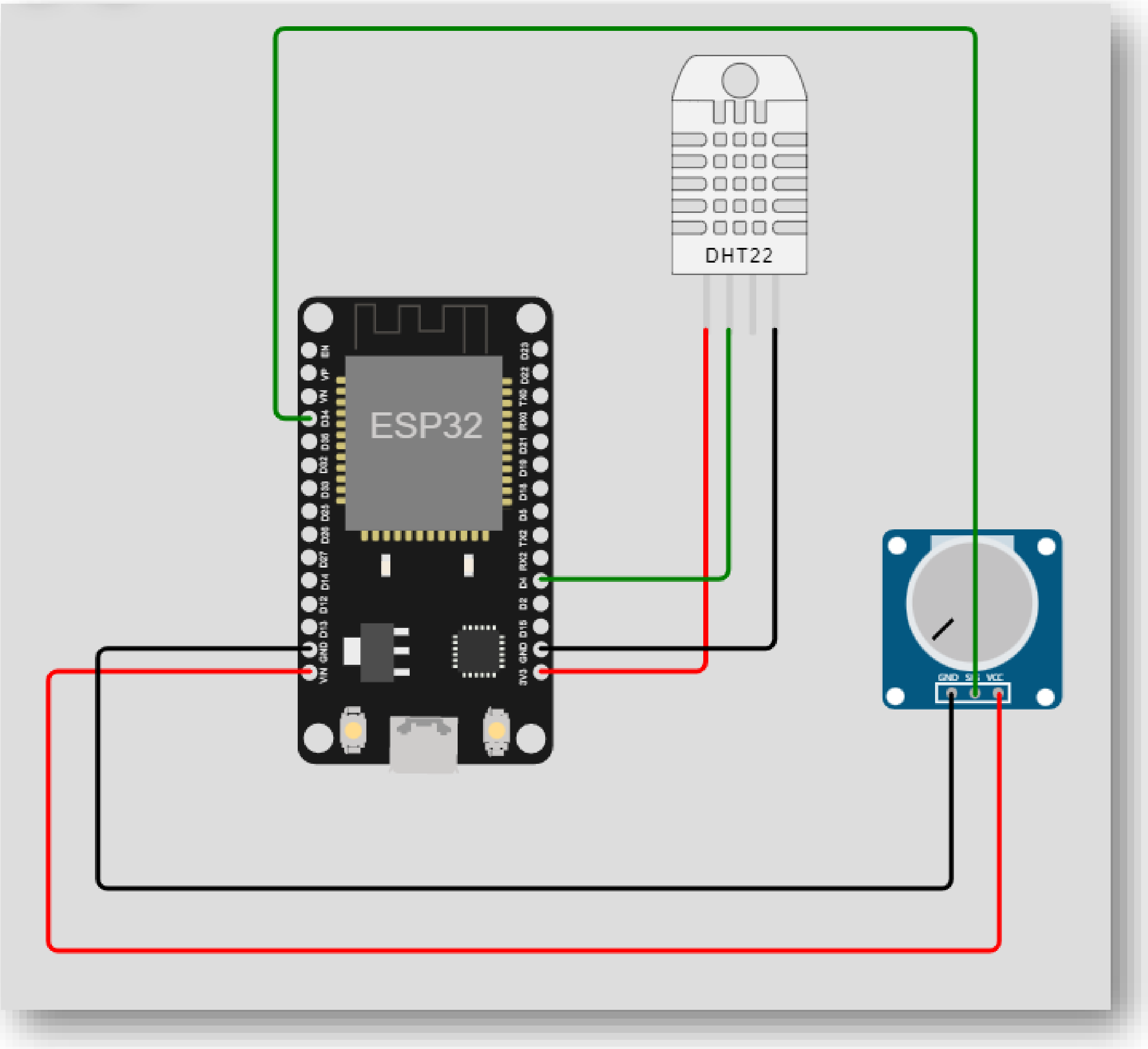


Fig No: 7 WOKWI Connections

The proposed IoT device consists of three components. They are ESP32, DHT22 and potentiometer.

1. DHT22:

DHT22 is a temperature sensor that is used to generate temperature values in Celsius. It can be manually controlled and the temperature values can be changed while the simulation is still running. Click on the DHT22 sensor and a small popup window will open. We can use the temperature and humidity sliders to change the values.

2. Potentiometer:

A potentiometer is a sensor that is used to measure an unknown voltage with the help of a known voltage. It is a knob-controlled variable resistor (linear potentiometer).

3. ESP 32:

The ESP32 is a popular Wi-Fi and Bluetooth-enabled microcontroller, widely used for IoT Projects. ESP32 can perform as a complete standalone system or as a slave device to a host MCU. It can interface with other systems to provide Wi-Fi and Bluetooth.

Connections:

Now the following connections are made.

- DHT22's VCC to any of the power input pins(VIN, 3V3) of ESP-32. It is meant for Power Supply.
- SDA to any of the GPIO pins. SDA stands for digital data pin (input/output).
- GND to one of the GND pins present in ESP 32.
- Potentiometer GND to the remaining GND of the ESP-32.

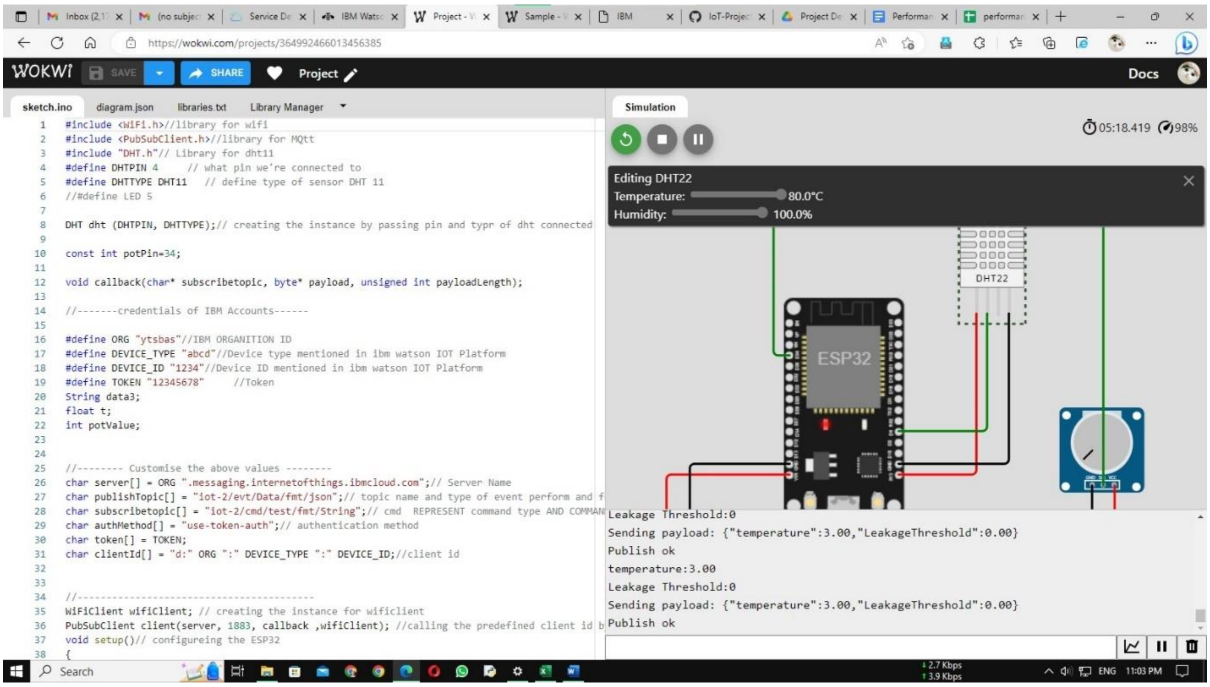
- SIG to any of the GPIO pins. SIG is used for output, generally connected to an analog input pin.
- Potentiometer VCC to the remaining power input pins of ESP-32.

5.2 WOKWI Code

```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt
#include "DHT.h"// Library for dht11
#define DHTPIN 4     // what pin we're connected to
#define DHTTYPE DHT11   // define type of sensor DHT 11
//#define LED 5

DHT dht (DHTPIN, DHTTYPE);// creating the instance by passing pin and typr of
dht connected

const int potPin=34;

void callback(char* subscribetopic, byte* payload, unsigned int
payloadLength);

//-------credentials of IBM Accounts------

#define ORG "ytsbas"//IBM ORGANITION ID
#define DEVICE_TYPE "abcd"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "1234"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "12345678"     //Token
String data3;
float t;
int potValue;


//-------- Customise the above values --------
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Name
char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of
event perform and format in which data to be send
char subscribetopic[] = "iot-2/cmd/test/fmt/String";// cmd  REPRESENT command
type AND COMMAND IS TEST OF FORMAT STRING
char authMethod[] = "use-token-auth";// authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id


//-----------------------------------------
WiFiClient wifiClient; // creating the instance for wificlient
```

```
PubSubClient client(server, 1883, callback ,wifiClient); //calling the
predefined client id by passing parameter like server id,portand
wificredential
void setup()// configureing the ESP32
{
  Serial.begin(115200);
  dht.begin();
  //pinMode(LED,OUTPUT);
  delay(10);
  Serial.println();
  wificonnect();
  mqttconnect();
}

void loop()// Recursive Function
{

  //h = dht.readHumidity();
  t = dht.readTemperature();
  potValue = analogRead(potPin);
  Serial.print("temperature:");
  Serial.println(t);
  Serial.print("Leakage Threshold:");
  Serial.println(potValue);

  //Serial.print("Humidity:");
  //Serial.println(h);

  PublishData(t, potValue);
  delay(1000);
  if (!client.loop()) {
    mqttconnect();
  }
}



/*.....................................retrieving to
Cloud...............................*/

void PublishData(float temp, float potenval) {
  mqttconnect();//function call for connecting to ibm
  /*
     creating the String in in form JSon to update the data to ibm cloud
  */
  String payload = "{\"temperature\":";
  payload += temp;
  payload += "," "\"LeakageThreshold\":";
  payload += potenval;
  payload += "}";


  Serial.print("Sending payload: ");
  Serial.println(payload);


  if (client.publish(publishTopic, (char*) payload.c_str())) {
    Serial.println("Publish ok");// if it sucessfully upload data on the cloud
then it will print publish ok in Serial monitor or else it will print publish
failed
  } else {
    Serial.println("Publish failed");
  }

}
void mqttconnect() {
  if (!client.connected()) {
    Serial.print("Reconnecting client to ");
    Serial.println(server);
    while (!!!client.connect(clientId, authMethod, token)) {
      Serial.print(".");
      delay(500);
    }

     initManagedDevice();
     Serial.println();
  }
}
void wificonnect() //function defination for wificonnect
{
  Serial.println();
  Serial.print("Connecting to ");

  WiFi.begin("Wokwi-GUEST", "", 6);//passing the wifi credentials to establish
the connection
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void initManagedDevice() {
  if (client.subscribe(subscribetopic)) {
    Serial.println((subscribetopic));
    Serial.println("subscribe to cmd OK");
  } else {
    Serial.println("subscribe to cmd FAILED");
  }
}


void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
{

  Serial.print("callback invoked for topic: ");
  Serial.println(subscribetopic);
  for (int i = 0; i < payloadLength; i++) {
    //Serial.print((char)payload[i]);
    data3 += (char)payload[i];
  }

  Serial.println("data: "+ data3);
  data3="";
```

5.3 WOKWI Output



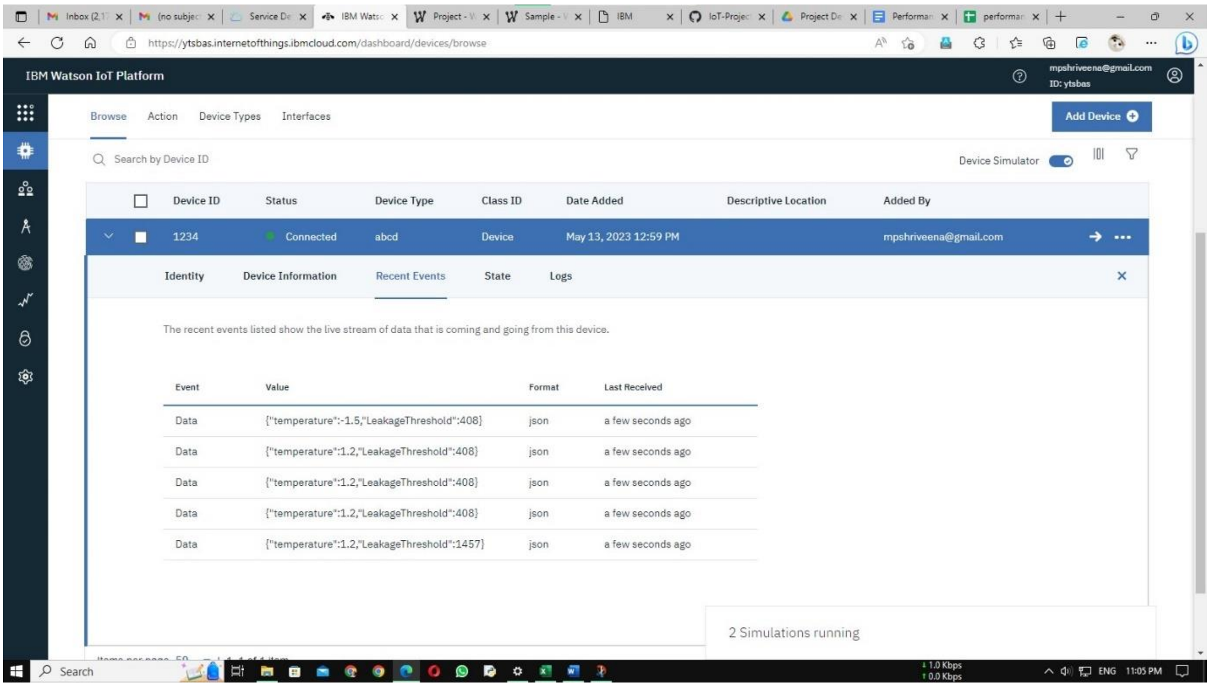
```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt
#include "DHT.h"// Library for dht11
#define DHTPIN 4     // what pin we're connected to
#define DHTTYPE DHT11   // define type of sensor DHT 11
//#define LED 5

DHT dht (DHTPIN, DHTTYPE);// creating the instance by passing pin and typr of dht connected

const int potPin=34;

void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);

//-------credentials of IBM Accounts------

#define ORG "ytsbas"//IBM ORGANITION ID
#define DEVICE_TYPE "abcd"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "1234"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "12345678"     //Token
String data3;
float t;
int potValue;


//-------- Customise the above values --------
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Name
char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of event perform and f
char subscribetopic[] = "iot-2/cmd/test/fmt/String";// cmd  REPRESENT command type AND COMMAN
char authMethod[] = "use-token-auth";// authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id


//-----------------------------------------
WiFiClient wifiClient; // creating the instance for wificlient
PubSubClient client(server, 1883, callback ,wifiClient); //calling the predefined client id b
void setup()// configureing the ESP32
{
```

Editing DHT22
Temperature: 80.0°C
Humidity: 100.0%

```
Leakage Threshold:0
Sending payload: {"temperature":3.00,"LeakageThreshold":0.00}
Publish ok
temperature:3.00
Leakage Threshold:0
Sending payload: {"temperature":3.00,"LeakageThreshold":0.00}
Publish ok
```

Fig No: 8 WOKWI Output

5.4 IBM Output



Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location	Added By
1234	Connected	abcd	Device	May 13, 2023 12:59 PM		mpshriveena@gmail.com

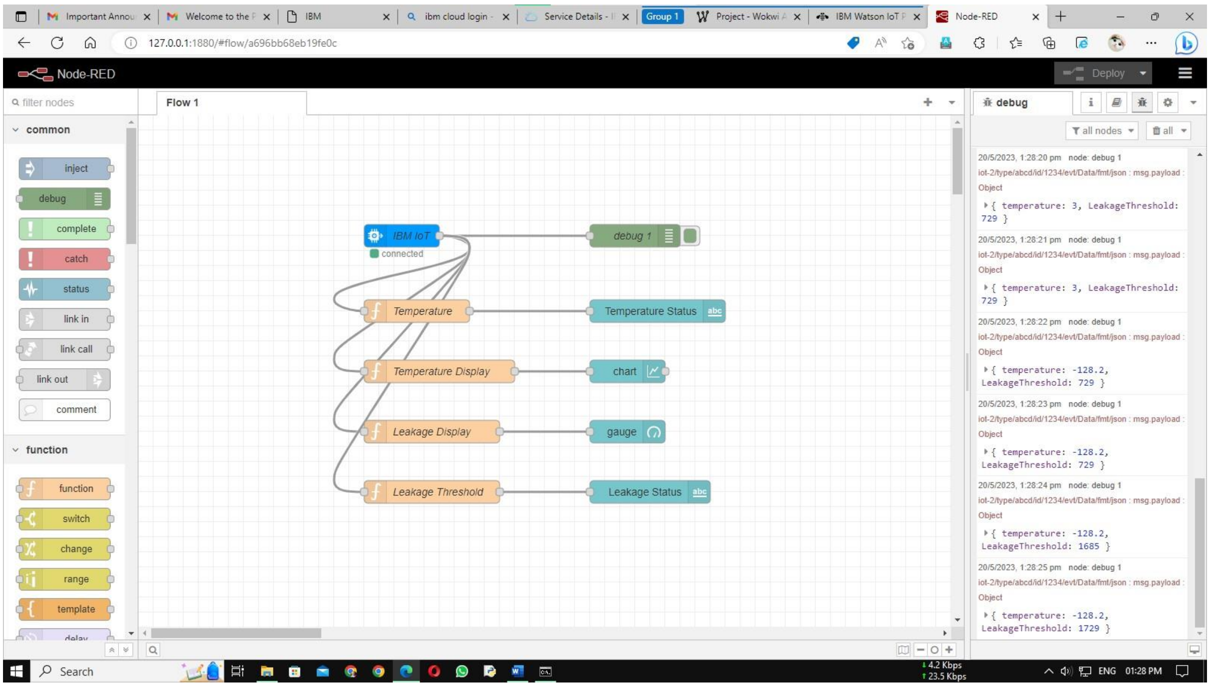
The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
Data	{"temperature":-1.5,"LeakageThreshold":408}	json	a few seconds ago
Data	{"temperature":1.2,"LeakageThreshold":408}	json	a few seconds ago
Data	{"temperature":1.2,"LeakageThreshold":408}	json	a few seconds ago
Data	{"temperature":1.2,"LeakageThreshold":408}	json	a few seconds ago
Data	{"temperature":1.2,"LeakageThreshold":1457}	json	a few seconds ago

2 Simulations running

Fig No: 9 IBM Output

5.5 Node-Red Output

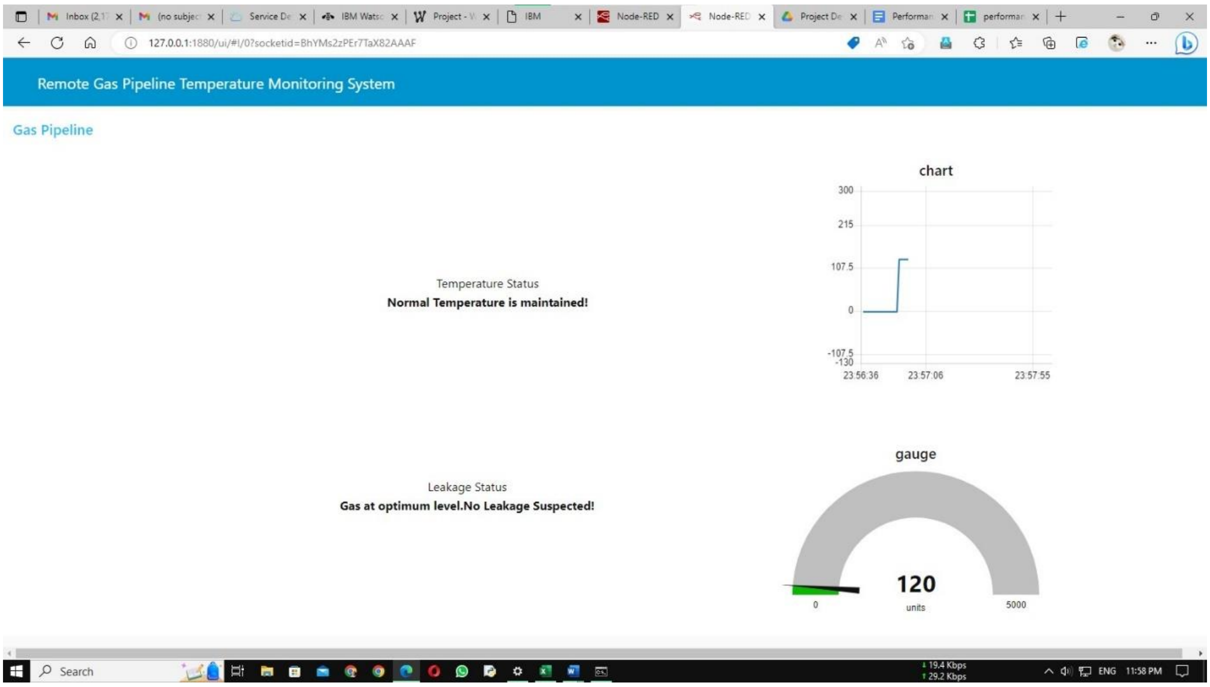


```
20/5/2023, 1:28:20 pm   node: debug 1
iot-2/type/abcd/id/1234/evt/Data/fmt/json : msg.payload :
Object
▸{ temperature: 3, LeakageThreshold:
729 }
20/5/2023, 1:28:21 pm   node: debug 1
iot-2/type/abcd/id/1234/evt/Data/fmt/json : msg.payload :
Object
▸{ temperature: 3, LeakageThreshold:
729 }
20/5/2023, 1:28:22 pm   node: debug 1
iot-2/type/abcd/id/1234/evt/Data/fmt/json : msg.payload :
Object
▸{ temperature: -128.2,
LeakageThreshold: 729 }
20/5/2023, 1:28:23 pm   node: debug 1
iot-2/type/abcd/id/1234/evt/Data/fmt/json : msg.payload :
Object
▸{ temperature: -128.2,
LeakageThreshold: 729 }
20/5/2023, 1:28:24 pm   node: debug 1
iot-2/type/abcd/id/1234/evt/Data/fmt/json : msg.payload :
Object
▸{ temperature: -128.2,
LeakageThreshold: 1685 }
20/5/2023, 1:28:25 pm   node: debug 1
iot-2/type/abcd/id/1234/evt/Data/fmt/json : msg.payload :
Object
▸{ temperature: -128.2,
LeakageThreshold: 1729 }
```

Fig No: 10 Node-Red Output

5.6 Node-Red Dashboard

1) When normal temperature is maintained and no leakage is suspected:



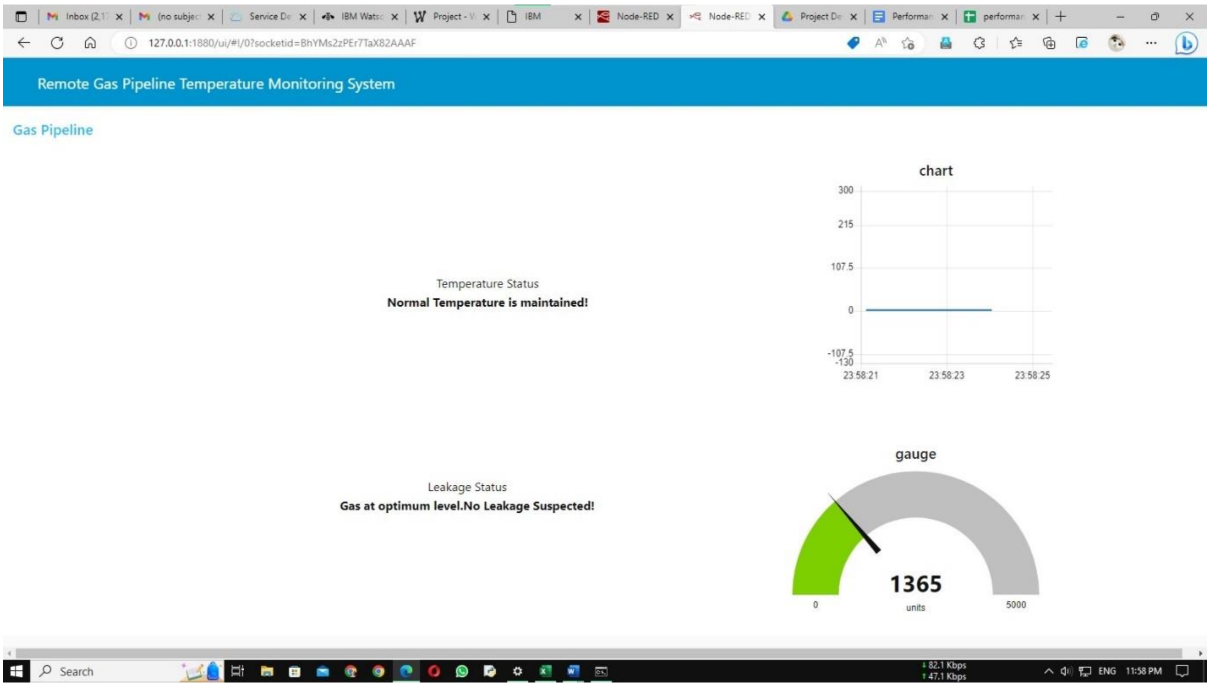
Remote Gas Pipeline Temperature Monitoring System

Gas Pipeline

Temperature Status
Normal Temperature is maintained!

Leakage Status
Gas at optimum level.No Leakage Suspected!

Fig No: 11 a) Node-Red Dashboard when normal temperature is maintained and no leakage is suspected



Remote Gas Pipeline Temperature Monitoring System

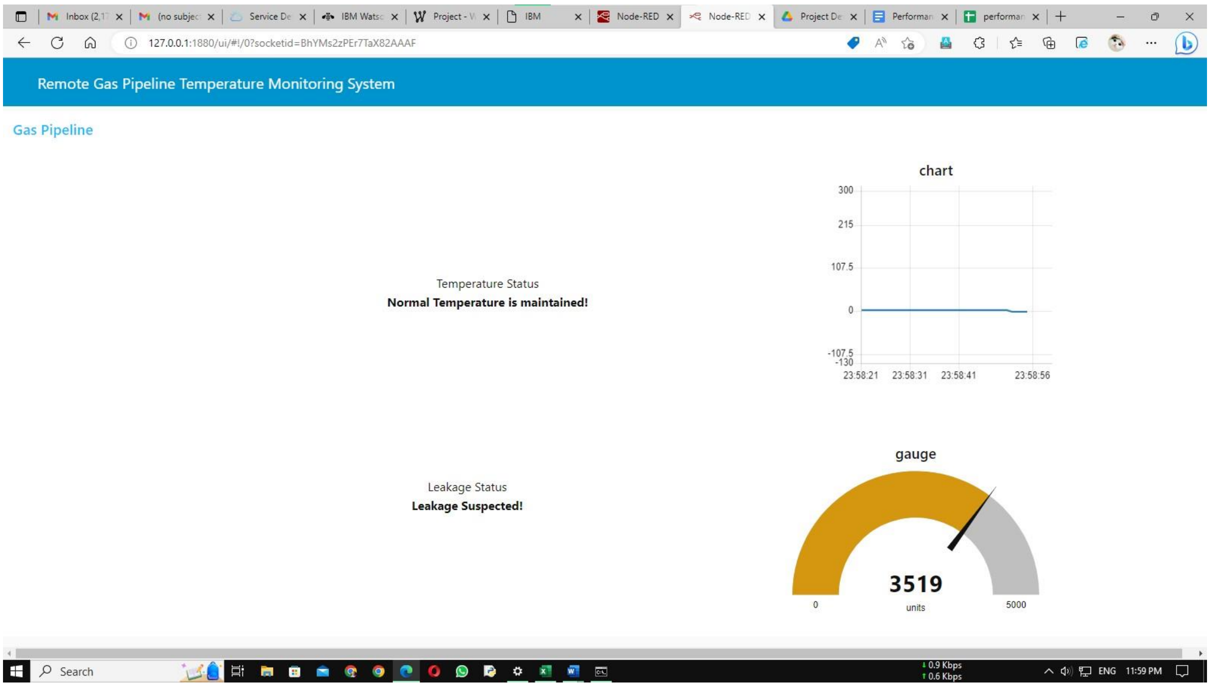
Gas Pipeline

Temperature Status
Normal Temperature is maintained!

Leakage Status
Gas at optimum level.No Leakage Suspected!

Fig No: 11 b) Node-Red Dashboard when normal temperature is maintained and no leakage is suspected

2) When normal temperature is maintained and leakage suspected:



Remote Gas Pipeline Temperature Monitoring System

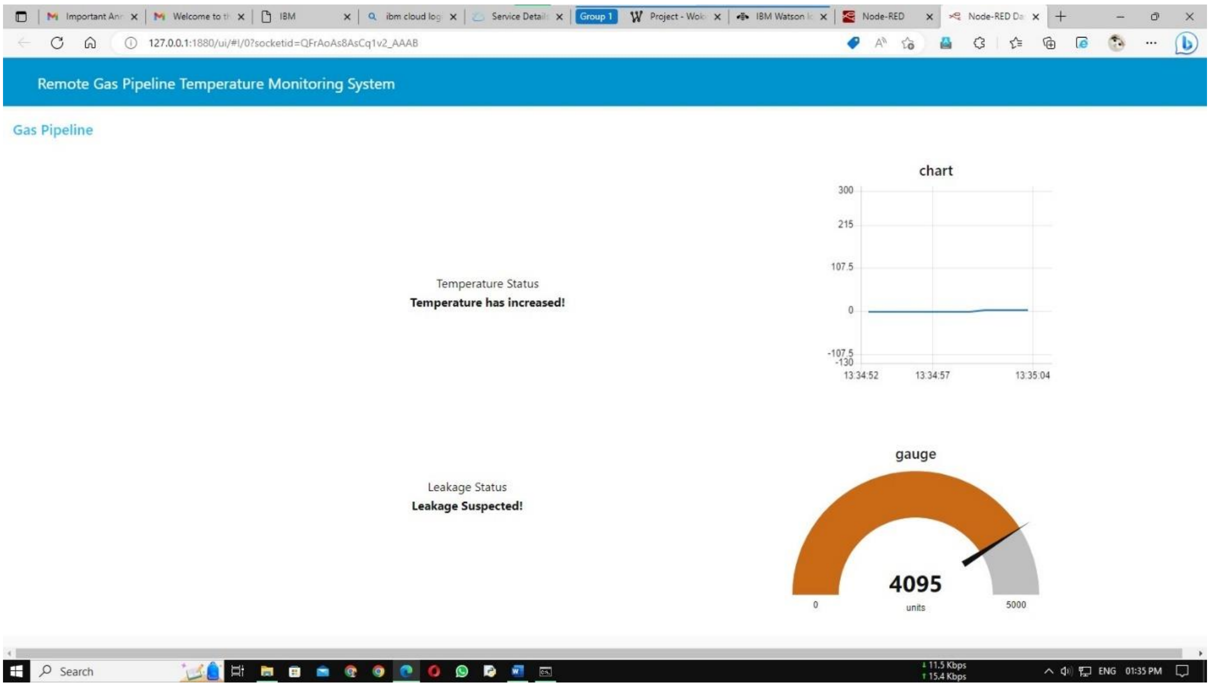
Gas Pipeline

Temperature Status
Normal Temperature is maintained!

Leakage Status
Leakage Suspected!

Fig No: 11 c) Node-Red Dashboard when normal temperature is maintained and leakage suspected

3) When temperature is increased and leakage suspected:



Remote Gas Pipeline Temperature Monitoring System

Gas Pipeline

Temperature Status
Temperature has increased!

Leakage Status
Leakage Suspected!

Fig No: 11 c) Node-Red Dashboard when temperature is increased and leakage suspected

5.7 Feature 1

- The first feature in this project is finding the temperature.
- It gives the accurate value of the temperature by the use of DHT22.
- It also gives the status of the temperature whether the normal temperature is maintained or the temperature has increased.
- The temperature is visualized by chart

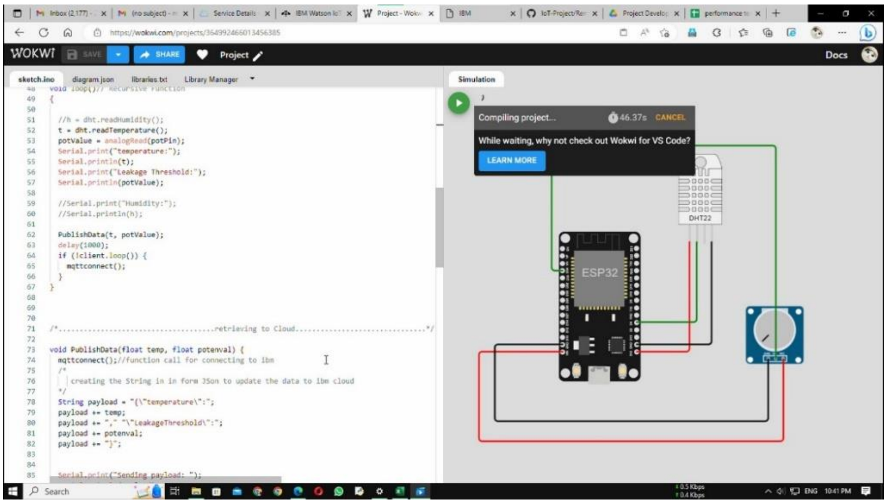
5.8 Feature 2

- The second feature of this project is finding the leakage threshold.
- It finds for leakage of gas if there is damage in the pipeline.
- It also give the status of the leakage whether leakage is suspected or not.
- The leakage is visualized by gauge.

CHAPTER – 6

PERFORMANCE

6.1 Performance Metrics

Parameter	Values	Screenshot
Metrics	Wowki Execution time : 46.37 seconds	**WOKWI Execution Time:**  Fig No: 12 WOKWI Execution Time

```
void loop()// Recursive Function
{

  //h = dht.readHumidity();
  t = dht.readTemperature();
  potValue = analogRead(potPin);
  Serial.print("temperature:");
  Serial.println(t);
  Serial.print("Leakage Threshold:");
  Serial.println(potValue);

  //Serial.print("Humidity:");
  //Serial.println(h);

  PublishData(t, potValue);
  delay(1000);
  if (!client.loop()) {
    mqttconnect();
  }
}



/*....................................retrieving to Cloud...............................*/

void PublishData(float temp, float potenval) {
  mqttconnect();//function call for connecting to ibm
  /*
     creating the String in in form JSon to update the data to ibm cloud
  */
  String payload = "{\"temperature\":";
  payload += temp;
  payload += "," "\"LeakageThreshold\":";
  payload += potenval;
  payload += "}";


  Serial.print("Sending payload: ");
```

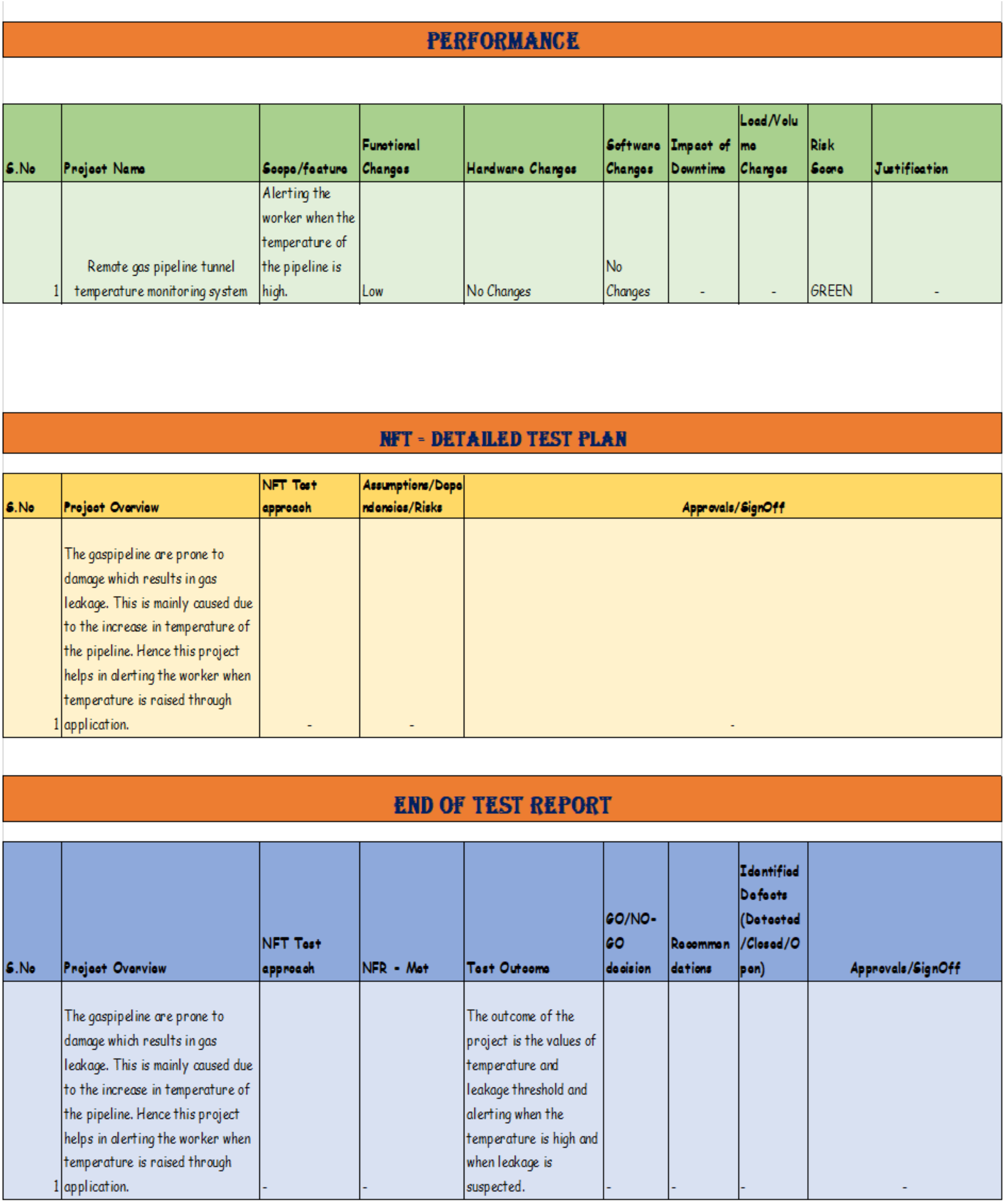
Compiling project... 46.37s CANCEL

While waiting, why not check out Wokwi for VS Code?

LEARN MORE

Table No: 8 Performance Metrics

6.2 Performance Testing:



PERFORMANCE

S.No	Project Name	Scope/feature	Functional Changes	Hardware Changes	Software Changes	Impact of Downtime	Load/Volume Changes	Risk Score	Justification
1	Remote gas pipeline tunnel temperature monitoring system	Alerting the worker when the temperature of the pipeline is high.	Low	No Changes	No Changes	-	-	GREEN	-

NFT - DETAILED TEST PLAN

S.No	Project Overview	NFT Test approach	Assumptions/Dependencies/Risks	Approvals/SignOff
1	The gaspipeline are prone to damage which results in gas leakage. This is mainly caused due to the increase in temperature of the pipeline. Hence this project helps in alerting the worker when temperature is raised through application.	-	-	-

END OF TEST REPORT

S.No	Project Overview	NFT Test approach	NFR - Met	Test Outcome	GO/NO-GO decision	Recommendations	Identified Defects (Detected/Closed/Open)	Approvals/SignOff
1	The gaspipeline are prone to damage which results in gas leakage. This is mainly caused due to the increase in temperature of the pipeline. Hence this project helps in alerting the worker when temperature is raised through application.	-	-	The outcome of the project is the values of temperature and leakage threshold and alerting when the temperature is high and when leakage is suspected.	-	-	-	-

Table No: 9 Performance Testing

CHAPTER – 7

ADVANTAGES & DISADVANTAGES

Advantages:

- This project helps to mintor the temperature.
- It give the temperature status.
- It gives leakage status.
- Simple in design.
- Alerts before it’s too late
- It is cheap.

Disadvantages:

- It requires internet.

CHAPTER – 8

CONCLUSION

The use of temperature monitoring sensors has enabled the monitoring of change in temperature affecting the quality of the products. Their implementation in different sectors has empowered the logistics and warehousing operations of a company. Employing a temperature monitoring system will save you money in the long run, on wasted stock, product recalls etc. If there is a potential problem, you will be notified immediately, before it is too late.

CHAPTER – 9

FUTURE SCOPE

This project shall not remain static. It is always subject to dynamic. Enhancement can be done efficiency in this project. The enhancements can lead to further more modifications which helps to minimize the execution time and increase the accuracy of the system.

CHAPTER – 10

APPENDIX

10.1 Source Code & WOKWI LINK:

Source Code:

https://github.com/naanmudhalvan-SI/IoT-Project/blob/main/Remote_Gas_project

WOKWI link:

https://wokwi.com/projects/364992466013456385

10.2 GitHub & Project Video Demo Link

GitHub link:

https://github.com/naanmudhalvan-SI/PBL-NT-GP--20258-1682662146

Project Video Demo link:

https://drive.google.com/file/d/1OKfOhblWYyX_uSkBHI6kCdSJ6DujtIWa/view?usp=share_link