PROJECT REPORT

PROJECT TITLE:INDUSTRY-SPECIFIC INTELLIGENT FIRE MANAGEMENT SYSTEM.

TEAM ID:PNT2022TMID20269

TEAM MEMBERS:
SURENTHIRAN.K.N

YOGI S

SANJAY M

REUBAN S

1.INTRODUCTION

1.1 Project overview

Fire, explosion and toxic release are the three major hazards in the process industry, while fire is the most common one. Increasing number of fire incidents coupled with loss of property has enhanced the demand for automatic intelligent fire alarm systems in residential and commercial buildings. An intelligent fire alarm system is specifically designed to provide advantages such as identification of the fire location, locate any fault in the alarm system wiring, and ensure easier maintenance. This system includes a Gas sensor, Flame sensor and temperature sensors to detect any changes in the environment. Based on the temperature readings and if any Gases are present the exhaust fans are powered ON. If any flame is detected the sprinklers will be switched on automatically. Emergency alerts are notified to the authorities and Fire station. Moreover, these modern intelligent fire alarm systems are more sensitive as compared to the classic models and are competent to detect false alarms.

1.2 Purpose

The primary purpose of fire alarm system is to provide an early warning of fire so that people can be evacuated & immediate action can be taken to stop or eliminate the fire effect as soon as possible.

2.LITERATURE SURVEY

2.1 Existing problem

Fire monitoring systems have usually been based on a single sensor such as smoke or flame. These single sensor systems have been unable to distinguish between true and false presence of fire . Consuming energy all day long and being dependent on one sensor that might end with false alert is not efficient and environmentally friendly. We need a system that is efficient not only in sensing fire accurately, but we also need a solution which is smart. In order to improve upon the results of existing single sensor systems , the smart fire management system includes a Gas sensor, Flame sensor and a temperature sensor . This system also requires a proper network with individual smart devices connected to various panels .

2.2 References

[1]N N Mahzan, N I M Enzai, N M Zin and K S S K M Noh, ”Design of an Arduino-based home fire alarm system with gSM module”, 1st International conference on green and Sustainable computing (ICoGeS), 2017.

[2] ZHANG Ying-Cong, YU Jing, “Study on the Fire IOT Development Strategy”, Shenyang Fire Research Institute --Radiant Energy-Sensing Fire Detectors for Automatic Fire Alarm Signaling, US: ANSI/FMRC, pp. FM3260-2004.

[3] Public Security, Shenyang 110034, China Shenyang Institute of Engineering, Shenyang 110136, China, 2019. Liu Yunhong Qi Meini,"The Design of Building Fire Monitoring System Based on ZigBee-WiFi Networks", Eighth International Conference on Measuring Technology and Mechatronics Automation, IEEE, 2016, pp-733-735

[4] R.A. Sowah, A.R. Ofoli, S.N. Krakani, S.Y. Fiawoo, hardware Design and Web-Based Communication Modules of a Real-Time multisensor Fire Detection and Notification System Using Fuzzy Logic, IEEE Transactions on Industry Applications, 53 (2016) 559-566.

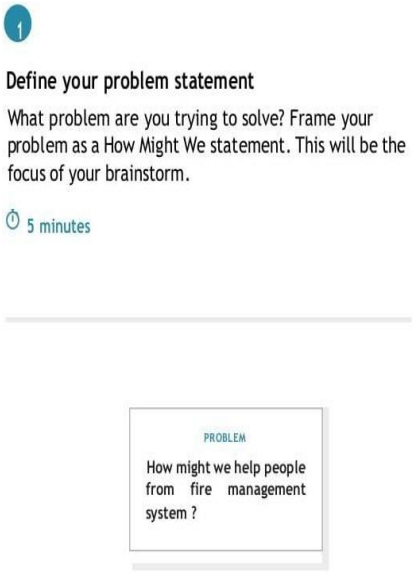
2.3 Problem Statement Definition

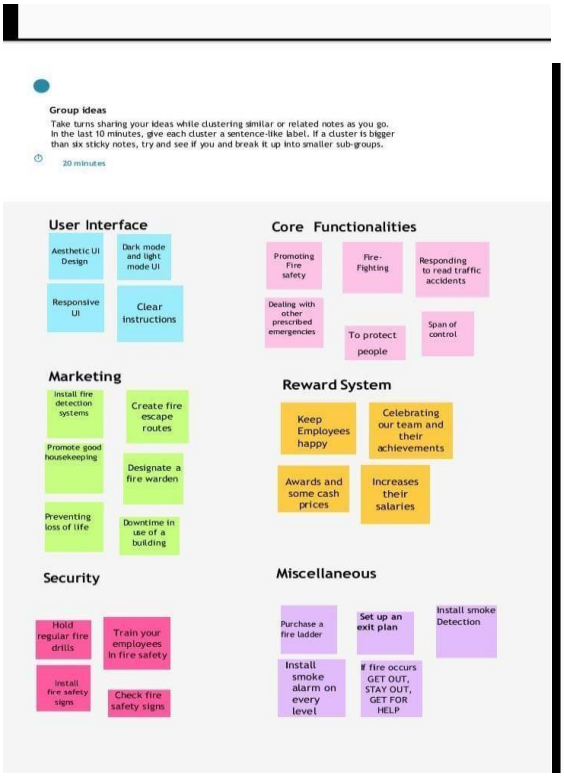
Industry Specific Intelligent fire management system are designed to Prevent fire accidents due to Gas leakage and flame in industry

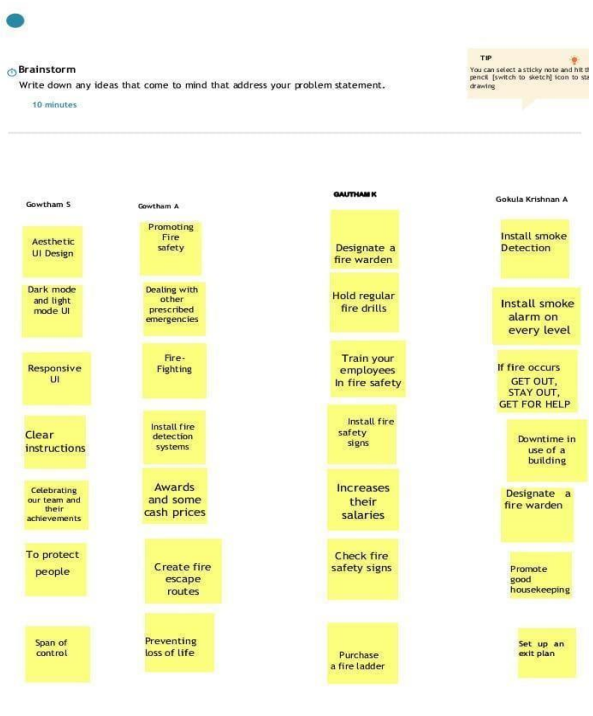
3.IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

3.2 Ideation &Brainstorming



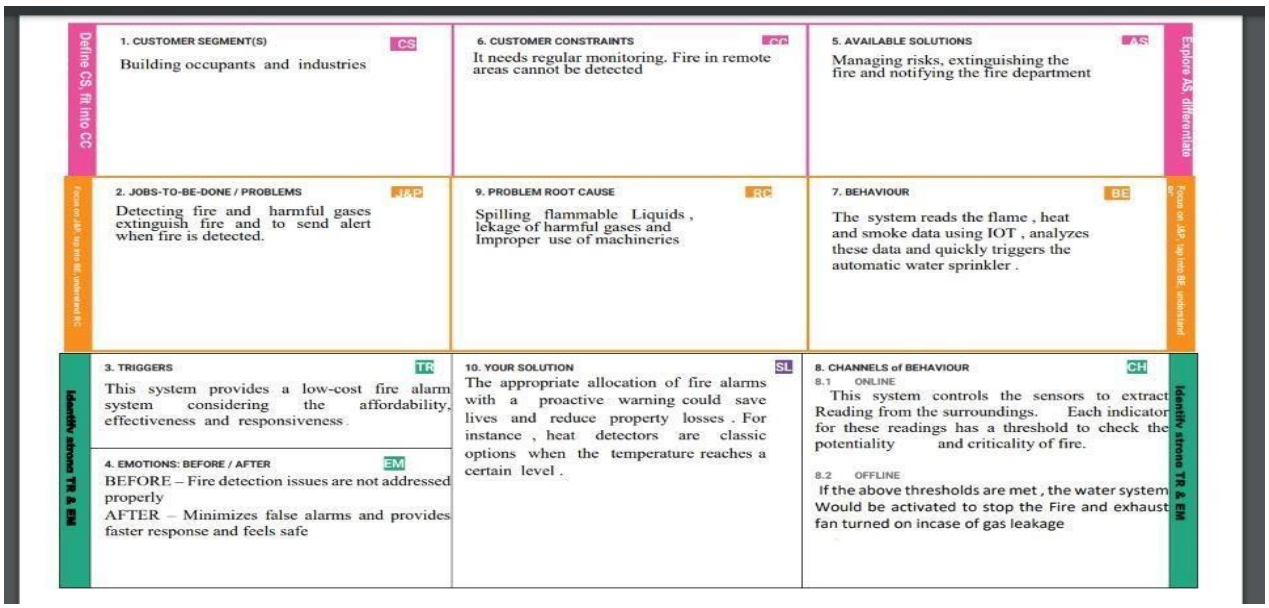




3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	To develop a safety management system for industries for protecting against the fire incidents in industries.
2.	Idea / Solution description	To implement the fire safety system in industries based on IOT using Arduino uno with fire detection and fire extinguisher system using sensors like gas sensor, Flame sensor, temperature sensor .
3.	Novelty / Uniqueness	The integrated system detects fire and gas in the atmosphere and extinguish it automatically. It also send alert via sms when gas and fire are detected.
4.	Social Impact / Customer Satisfaction	The system detects and senses the fire using sensors that we use and it helps the customers to access with the immediate notification and the timely access and helps in early prevention of fire accidents in industries.
5.	Business Model (Revenue Model)	This model is used to calculate the probability of the ignition and spread across a landscape of industries and help in rescuing people and machine from the fire accident.
6.	Scalability of the Solution	This is completely modular system makes it easily expandable and business efficient for the customized fire detection, with the significant cost , Easy operatability and management.

3.4 Problem Solution Fit



4. REQUIREMENT ANALYSIS

4.1 Functional requirement

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 • Registration through Gmail • Registration through LinkedIn
FR-2	User Confirmation	• Confirmation via Email • Confirmation via OTP
FR-3	User Login with respective ID	• Login through website • App using the respective username and password
FR-4	User Access	• Access the app requirements
FR-5	User Upload	• User should be able to upload the data
FR-6	User Solution	• Data report should be generated • Delivered to user for every 24 hours
FR-7	User Data Sync	• API interface to increase to invoice system
FR-8	User Guide	• Guides the basic steps of using the application

4.2 Non-Functional requirement

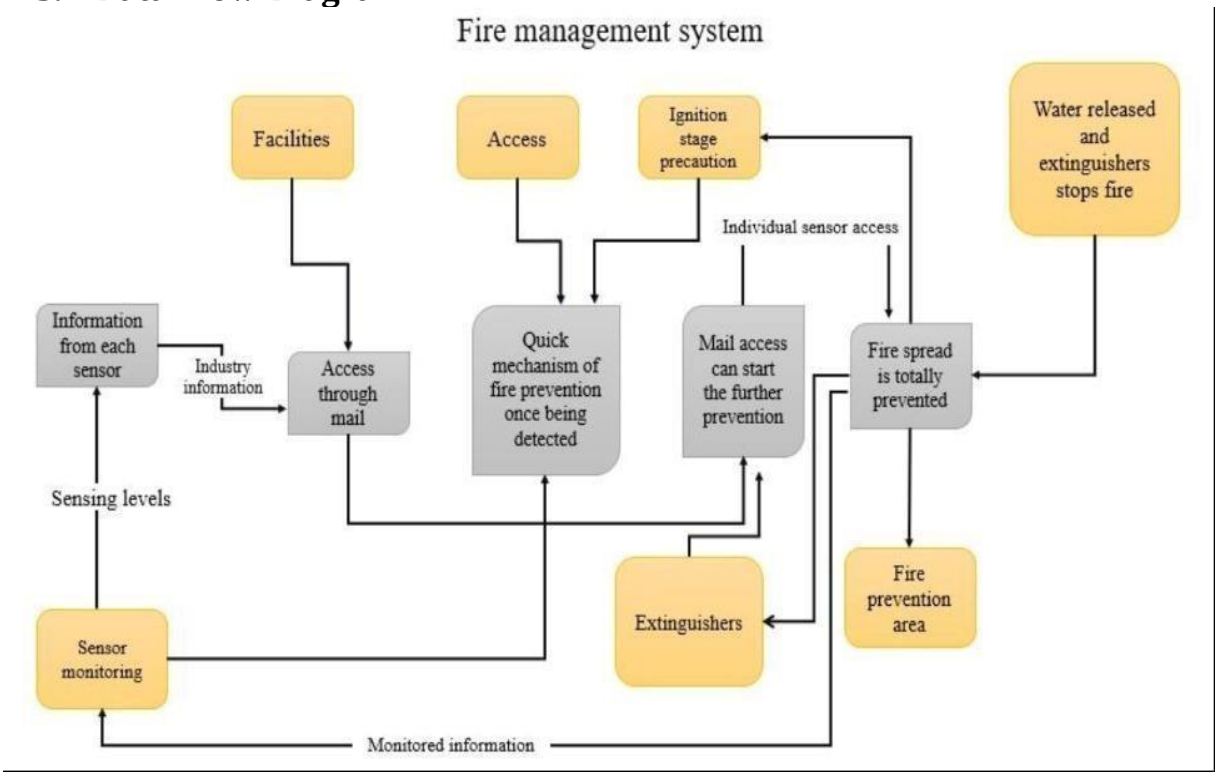
Non-functional Requirements:

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

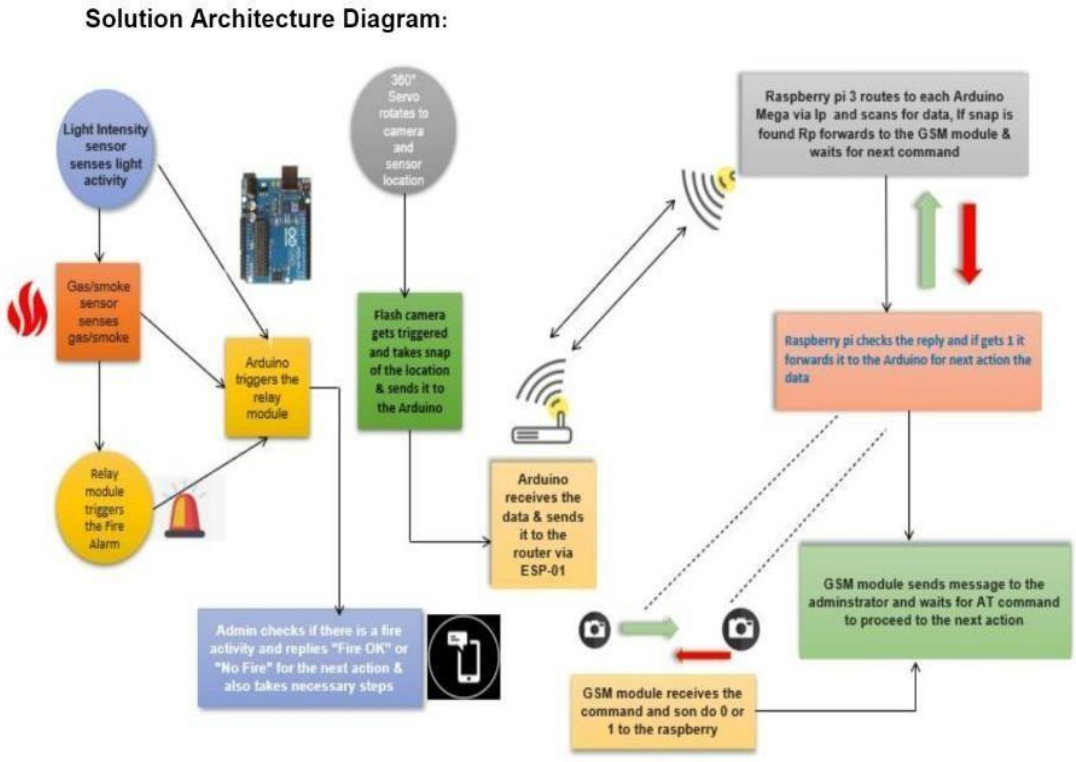
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	• Low perceived workload. • Easy and simple UI. • Usability requirements can consider language barriers and localization tasks. • Usability can be assessed from the below functions. • Efficiency of use.
NFR-2	**Security**	• Access permissions for the system information may only be changed by the system's data administrator.
NFR-3	**Reliability**	• The database update process must roll back all related updates when any update fails.
NFR-4	**Performance**	• The front-page load time must be no more than 2 seconds for users that access the website using an VoLTE mobile connection.
NFR-5	**Availability**	• The rest of the pages that may experience problems must display a notification with a timer showing when the system is going to be up again. • New module deployment mustn't impact front page, product pages, and check out pages availability and mustn't take longer than one hour.
NFR-6	**Scalability**	• We can increase scalability by adding memory, servers, or disk space. On the other hand, we can compress data, use optimizing algorithms. • The website attendance limit must be scalable enough to support 500,000 users at a time.

5. PROJECT DESIGN

5.1 Data Flow Diagram



5.2 Solution Architecture



5.3 User Stories

User Stories

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
• Customer (Mobile user, Web user) • Customer Care • Executive Administrator	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-1

6. PROJECT PLANNING &SCHEDULING

6.1 Sprint Planning & Estimation

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Simulation creation	USN-1	Connect Sensors and Arduino with python code	2	High	VARSHA, LEENA BHARATHI, PREETHI,PRITHIBHA
Sprint-2	Software	USN-2	Creating device in the IBM Watson IoT platform, workflow for IoT scenarios using Node-Red	2	High	VARSHA, LEENA BHARATHI, PREETHI,PRITHIBHA
Sprint-3	Dashboard	USN-3	Design the Modules and test the app	2	High	VARSHA, LEENA BHARATHI, PREETHI,PRITHIBHA
Sprint-4	Web UI	USN-4	To make the user to interact with software.	2	High	VARSHA, LEENA BHARATHI, PREETHI,PRITHIBHA

7. CODING & SOLUTIONING

7.1 Feature 1

- IoT device
- IBM Watson Platform
- Node red
- Cloudant DB
- Web UI
- MIT App Inventor
- Python code

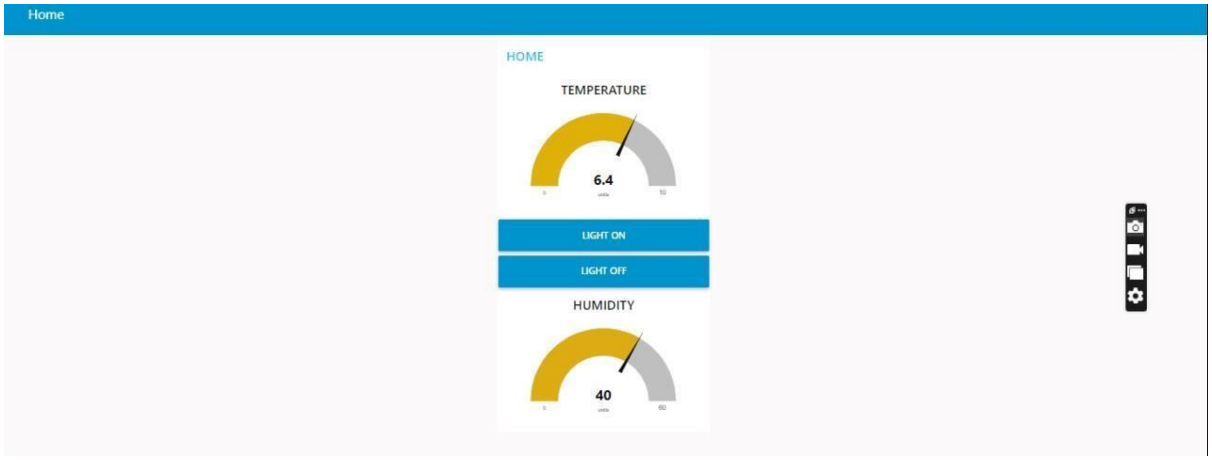
7.2 Feature 2

- Login
- Wokwi

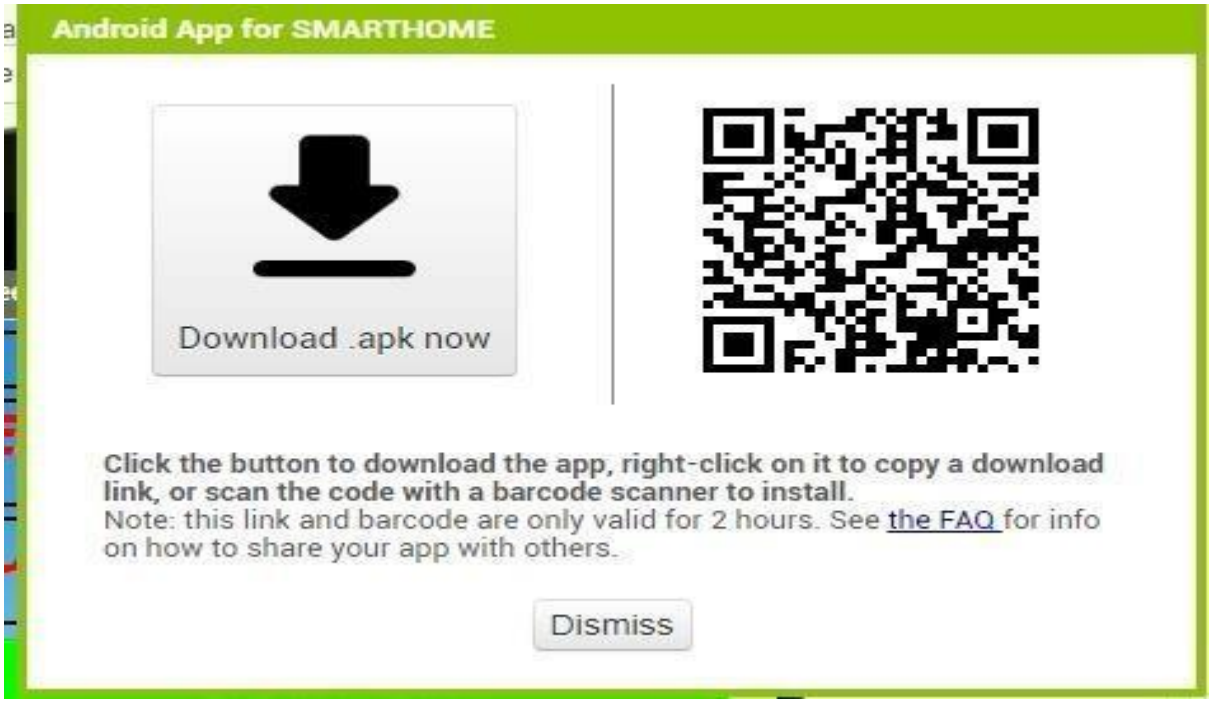
8. TESTING AND RESULTS

8.1 Test Cases

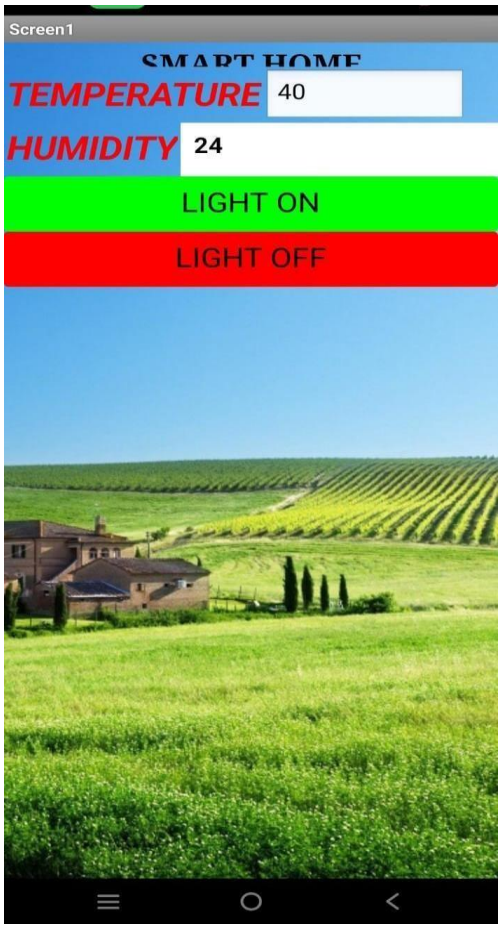
TEST CASE 1



TEST CASE 2



TEST CASE 3



9. ADVANTAGES

- Reduced installation cost.
- They monitor 24/7.
- Improved security in homes, industries and Offices.
- It pin points location of the fire.

10. DISADVANTAGES

- Heat detectors are not considered as life saving devices because they are sensitive only to heat.
- High battery or current consumption will need for these detectors.
- Control pannel may need to be replaced if it becomes damaged.

11. CONCLUSION

This gas leakage system can be applied for household safety and many other applications in the industry . Gas leakages and fire outbreaks in industries as well as houses have lead to wide destruction and losses in the past. Gas leakages and fire outbreaks both spread widely and lead to even greater loss of life and property if proper action is not taken on time. So here we proposed a system that detects gas as well as fire outbreaks and alert us accordingly so that proper action may be taken to control it.

12. FUTURE SCOPE

Smoke detectors and alarms are migrating from just the detection of smoke, to combination detectors and multicriteria detector. The future will be with multicriteria detection in which the detector will be more of a sensor,with the detection more for the products of combustion,such as carbon monoxide, carbon dioxide,sulfur dioxide ,nitrogen dioxide in addition to heat and particulate matter. Within the next decade, video image detection (VID) will become more mainstream in which, through analytics, the image of either smoke or flame will be able to be isolated and detected from within a room or space. The VID system would also be able to detect if an individual is within the space and through the integration with the notification appliances, provide a path of exit.

13. APPENDIX

13.1 Source Code

```
#include <WiFi.h>//library for wifi

#include <PubSubClient.h>//library for MQtt

#include "DHT.h"// Library for dht11

#define DHTPIN 15 // what pin we're connected to

#define DHTTYPE DHT22 // define type of sensor DHT 11

#define LED 2


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

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


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


#define ORG "i3869j"//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 h, t;



//-------- 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/command/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();
Serial.print("temp:");
Serial.println(t);
Serial.print("Humid:");
Serial.println(h);

PublishData(t, h);
delay(1000);
if (!client.loop()) {
```

```
mqttconnect();
}
}

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

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

```
if(data3=="lighton")
{
Serial.println(data3);
digitalWrite(LED,HIGH);
}
else
{
Serial.println(data3);
digitalWrite(LED,LOW);
}
data3="";
}
```

13.2 GitHub

GitHub Link:

https://github.com/IBM-EPBL/IBM-Project-20983-1659768587