INDUSTRY-SPECIFIC INTELLIGENT FIRE MANAGEMENT SYSTEM

PROJECT REPORT

Submitted by

SOWNDHARIYA V (737819ECR182)

SOUMIYA K (737819ECR180)

SARVESHAN K (737819ECR168)

SRI HARI V (737819ECR183)

in partial fulfilment

for the award of the degree

of

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

KONGU ENGINEERING COLLEGE

(AUTONOMOUS)

PERUNDURAI, ERODE-638060

PROJECT REPORT

CHAPTER -1

INTRODUCTION

1.1 PROJECT OVERVIEW

There are three primary risks in the process: fire, explosion, and hazardous discharge. whereas fire is the most frequent. More fire accidents are occurring. The need for autonomous intelligent systems has increased in conjunction with property damage. Buildings with both residential and business have fire alarm systems. An intelligent fire alarm system is created with the intention of offering benefits like pinpointing the location of the fire, identifying any wiring issues, and facilitating easier maintenance. To monitor changes in the environment, this system incorporates a gas sensor, a flame sensor, and temperature sensors. The exhaust fans are turned on based on the temperature measurements and, if any gases are present, on the temperature data. Sprinklers will be immediately activated if any flame is detected. Authorities and the local fire station are informed of emergency notifications. Additionally, these contemporary intelligent fire alarm systems are capable of identifying false alarms and are more sensitive than the traditional types.

1.2 PURPOSE OF FIRE ALERTING SYSTEM

1.2.1. DETECTION OF FIRE

Smoke and heat are the two primary ways that your fire alarm system is built to find a fire. If a fire is discovered before smoke or heat are detected by the system's sensors, it should also be possible to manually pull the trigger. When the sprinkler system detects movement, a fire is being responded to by the sprinklers, activating further systems.

Smoke alarms beep loudly when there is a fire

Fig 1.1 Detection of Fire

1.2.2 ALERT OCCUPANTS

Both audible and visual alarms are activated by the fire alarm system to notify building occupants of smoke, heat, or water movement. To encourage people to follow your evacuation plan, these alerts will be bright, loud, annoying, and impossible to ignore. All occupants of the building will be informed by using both types of alerts.

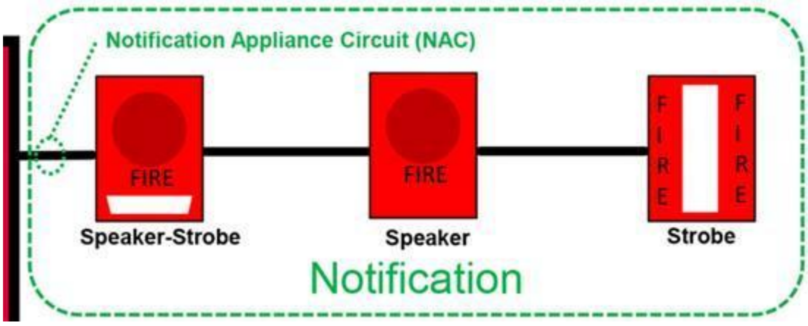


Fig 1.2 Alerting the System

1.2.3 MANAGE RISKS

The third method the fire alarm system in your building safeguards you is by taking action to reduce dangers as they arise. When the alarm is triggered, certain systems carry out a series of actions to protect passengers and stop the spread of fire and smoke. These actions include: automatically closing doors in various zones, turning off the ventilation and air conditioning, or rerouting elevators to transport vehicles to a specific level.

Fig 1.3 Risk rate

1.2.4 NOTIFY AUTHORITIES

Notifying authorities is the fourth function of your fire alarm system. This guarantees that the fire department will arrive as soon as possible, enabling them to respond and put out the fire before it poses a greater hazard.

Fig 1.4 Notification

CHAPTER – 2

2.1 LITERATURE REVIEW

[1] Md Saifudaullah Bin Bahrudin et al.,(2014) proposed a technique to Fire alarm system is a real-time monitoring system that detects the presence of smoke in the air due to fire and captures images via a camera installed inside a room when a fire occurs. The embedded systems used to develop this fire alarm system are Raspberry Pi and Arduino Uno.

[2]Wen-hui Dong et al.,(2016) proposed a technique to meet the application requirements of wireless automatic fire alarm system, based on the analysis of the applicable sites of wireless fire alarm system and the characteristics of communication service, this paper designs a dedicated wireless communication protocol for fire detection and alarm, and develops a complete set of wireless automatic fire alarm system, to achieve rapid fire detection and alarm and state supervision of fire-fighting facilities with low power consumption.

[3] Wilen Melsedec O. Narvios et al., (2017) proposed a technique to Existing buildings without a fire alarm monitoring system need an alternative installation cost-effective, flexible, and scalable and will not affect the building’s aesthetics. In place of the conventional fire alarm monitoring system, an Internet of Things (IoT)fire sensor nodes were designed to monitor each room’s temperature and analog CO level and deploy through point-to-point technique

[4]Karwan Muheden et al., (2019) proposed a technique to the surveillance of home or industrial places through sensors and the prevention of problems via prediction are of vital importance for the safety of these areas. This paper shows how to increase wireless sensor network (WSN) techniques by composing new design methods and improved a low-cost industrial and home safety systems.

[5] Lei Zhang et al., (2009) proposed a technique to an automatic fire alarm system based on wireless sensor networks is developed, which is designed for high-rise buildings. In order to provide early extinguishing of a fire disaster, large numbers of detectors which periodically measure smoke concentration or temperature are deployed in buildings.

[6]Chen Jing et al., (2012) proposed a technique to the problem of false alarm and missing alarm caused by information uncertainty existing in the fire alarm system, Bayesian network (BN) is proposed to analyse fire alarm system. The paper elaborates the internal logic relationship between the fire alarm and the physical-chemical characteristics generated in the process of fire burning by analysing fire mechanism.

[7] Asma Mahgoubet et al.,(2019) proposed a technique to Fire alarm systems are essential in alerting people before fire engulfs their homes. However, fire alarm systems, today, require a lot of wiring and labour to be installed. This discourages users from installing them in their homes. Therefore, we are proposing an IoT based wireless fire alarm system that is easy to install. The proposed system is an ad-hoc network that consists of several nodes distributed over the house.

[8]Seung-WookJee et al.,(2012)) analyses to several problems with conventional FASs and describes an advanced FAS, the traceable FAS, that we developed to solve these problems. The traceable FAS can trace the exact location of a fire, can detect multiple fires that occur simultaneously, and can be used with conventional detectors without disrupting the other detectors operation.

[9]Thou-Ho Chenet et al.,(2006) proposed a technique to identify smoke-detection method for early fire-alarming system based on video processing. The basic strategy of smoke-pixel judgment is composed of two decision rules: a chromaticity-based static decision rule and a diffusion-based dynamic characteristic decision rule.

[10]W. Tjokorda Agung Budiet et al.,(2011) proposed a technique to build a fire alarm system is based on fire detection on video data; this is done with digital image processing techniques and machine vision. Fire detection systems based-on video processing consist of four main stages, namely: motion detection area, detection area of the colour of fire, temporal analysis of probabilistic calculations or applying wavelet analysis, spatial analysis using energy calculations on an area.

2.2 EXISTING METHOD

The combination of fire alarm systems (FAS) with the building's current safety and automation systems. The majority of systems on the market can be used with our solutions. They are capable of collaborating with both wired and wireless systems. Alternatively, we provide complete fire-fighting systems that include detection, alarming, and extinguishing. Modern fire detection systems consist of one or more microprocessor-equipped cabinets with individual field devices attached to the cabinet by wiring and communications protocol. The devices may include the following:

- ❖ Smoke detectors
- ❖ Heat detectors
- ❖ Control units for fans and doors
- ❖ Visual annunciating devices (strobes)
- ❖ Audio annunciating devices (speakers, horns)

2.2.1 SMOKE DETECTORS

In order to maximise sensitivity to smoke, certain smoke detectors may employ both optical and ionisation sensing technologies, which are used by the majority of smoke detectors. Spot smoke detectors that can alert a fire control panel to activate a fire-suppression system are normally part of a full fire protection system. Smoke detectors can function alone, as part of a fire alarm or security system, or they can act in conjunction with other detectors in the vicinity so that they all sound an alert when one is triggered. For the hearing-impaired or deaf, flashing smoke alarms are also available. Smoke detectors need to include integrated carbon monoxide detectors in order to detect carbon monoxide and avoid carbon monoxide poisoning**.**

2.2.2 HEAT DETECTORS

A heat detector is a tool that can sense heat and can function mechanically or electrically. The majority of heat detectors are made to sound alarms and notify systems before smoke even becomes a concern. Fixed-temperature heat detectors are a highly economical option for many property protection applications since they detect heat when ambient temperatures reach a fixed point, generally signalling a fire. When a quick reaction to a fire is essential, rate-of-rise heat detectors are the perfect answer since they can detect temperature spikes that would only occur in the event of a fire emergency.

Combination heat detectors offer both rate-of-rise and fixed detection. This allows the heat detector to alert the central control panel before it reaches its predetermined set point for high rates of rise, allowing for a prompt reaction to both rapid and gradual temperature increases**.**

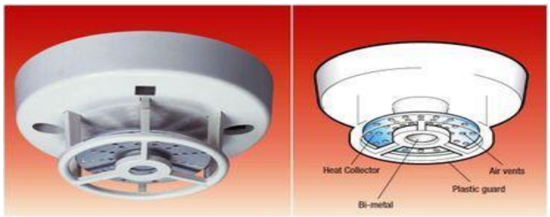


Fig 2.2.1 Heat Detectors

2.2.3 CONTROL UNITS FOR FANS AND DOORS

Modern fire alarm systems, sometimes referred to as fire alarm control panels, are linked to a personal computer (PC) that stores site-specific data, such as the assignment of devices, the naming of devices, the annunciating protocols and messages, and the order of operations for control devices and annunciators. The duration of an audible signal's activity with a general alarm, the format of the messages displayed for any condition, and the time a detector remains in an alarm state before a general alarm is issued are all options for these systems' operations.

2.2.4 VISUAL ANNUNCIATING DEVICES (STROBES)

For the hearing-impaired residents to be warned to leave the area, strobes generate synchronised flashes. Each strobe device, which is frequently mounted on walls or ceilings, has a candela rating, which is a measure of light intensity at its base level. A strobe device's coverage area increases with increasing candela rating**.**

2.2.5 AUDIO ANNUNCIATING DEVICES (SPEAKERS, HORNS)

Fire detection, automatic voice messages, spot announcements, and user-friendly operations are all included in the integrated intelligent addressable fire alarm and public address control panel, which is perfect for all commercial, residential, and industrial applications. Talkback / Public address units at a given place can be used to broadcast information utilising the panel's integrated voice alarm / announcement functionality, which uses addressable technology.

2.3 PROBLEM STATEMENT

Fire safety systems are often well-designed and correctly implemented, as was already said. But at this point, things start to go wrong since maintenance and test planning are frequently done with total contempt. According to Dieken, one-third of the safety systems fail to function effectively in the event of a fire simply due to a lack of inspection, testing, or maintenance of these systems. The author also mentions that roughly 49% of the fire extinguishing systems that were installed failed owing to poor maintenance, resulting in annual property losses of about 15.9 million dollars. Unfortunately, this form of concealed failure is only exposed when a fire develops and the system is required.

CHAPTER – 3

IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

- An empathy map is a straightforward, simple-to-understand picture that reveals information about a user's actions and views.
- It's an effective tool for teams to employ to comprehend their consumers better. Understanding the underlying issue and the individual experiencing it is necessary for developing a workable solution.
- Participants are encouraged to think about issues from the user's perspective, including objectives and obstacles, as part of the process in constructing the map.

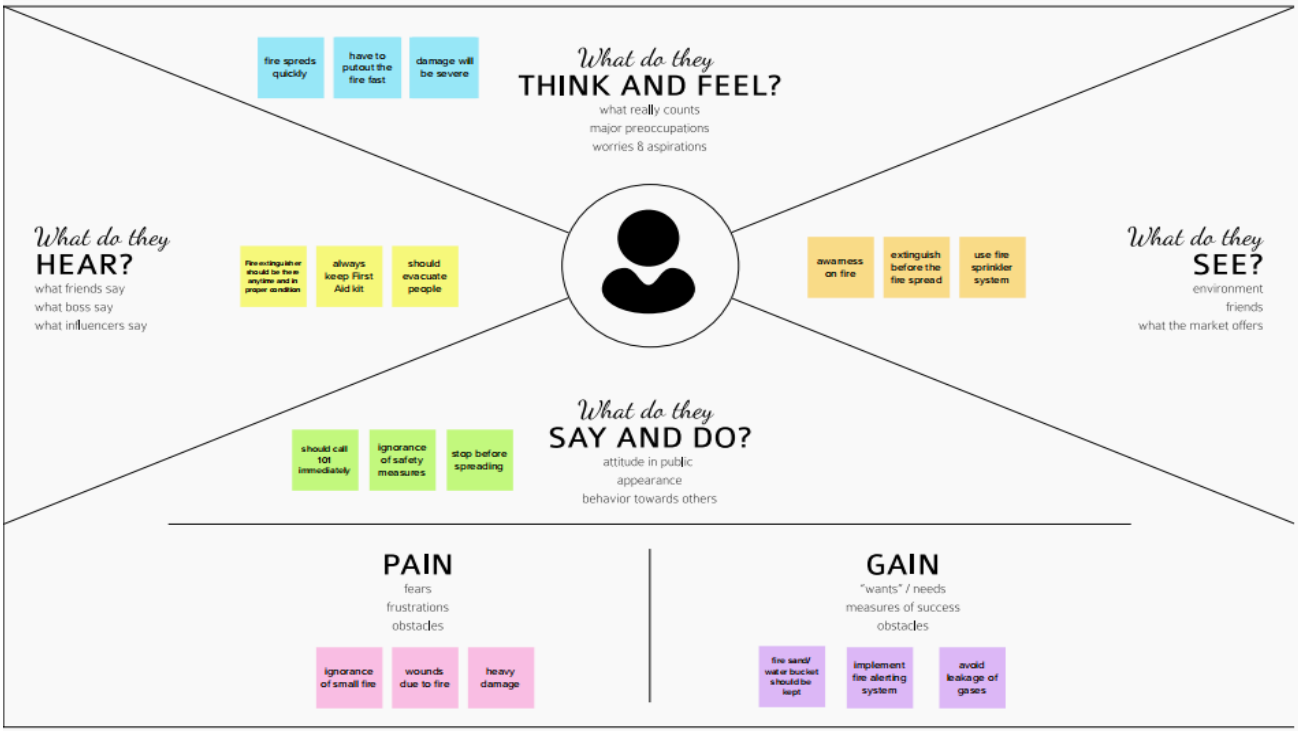


Fig 3.1 Empathy Map

3.2 IDEATION AND BRAINSTORMING

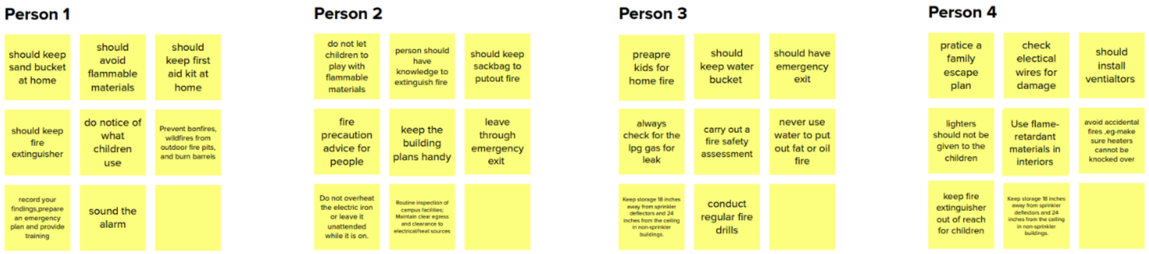
Step 1: Team Gathering, Collaboration and Select the Problem Statement

Team was gathered in mural app for collaboration

The team members are

- Sowndhariya V
- Soumiya K
- Sarveshan K
- Srihari V

Step 2: Brainstorm, Idea Listing and Grouping



Person 1

should keep sand bucket at home

should avoid flammable materials

should keep first aid kit at home

should keep fire extinguisher

do notice of what children use

Prevent bonfires, wildfires from outdoor fire pits, and burn barrels

record your findings,prepare an emergency plan and provide training

sound the alarm

Person 2

do not let children to play with flammable materials

person should have knowledge to extinguish fire

should keep sackbag to putout fire

fire precaution advice for people

keep the building plans handy

leave through emergency exit

Do not overheat the electric iron or leave it unattended while it is on.

Routine inspection of campus facilities; Maintain clear egress and clearance to electrical/heat sources

Person 3

preapre kids for home fire

should keep water bucket

should have emergency exit

always check for the lpg gas for leak

carry out a fire safety assessment

never use water to put out fat or oil fire

Keep storage 18 inches away from sprinkler deflectors and 24 inches from the ceiling in non-sprinkler buildings.

conduct regular fire drills

Person 4

pratice a family escape plan

check electical wires for damage

should install ventialtors

lighters should not be given to the children

Use flame-retardant materials in interiors

avoid accidental fires ,eg-make sure heaters cannot be knocked over

keep fire extinguisher out of reach for children

Keep storage 18 inches away from sprinkler deflectors and 24 inches from the ceiling in non-sprinkler buildings.

Fig 3.2 Brainstorm

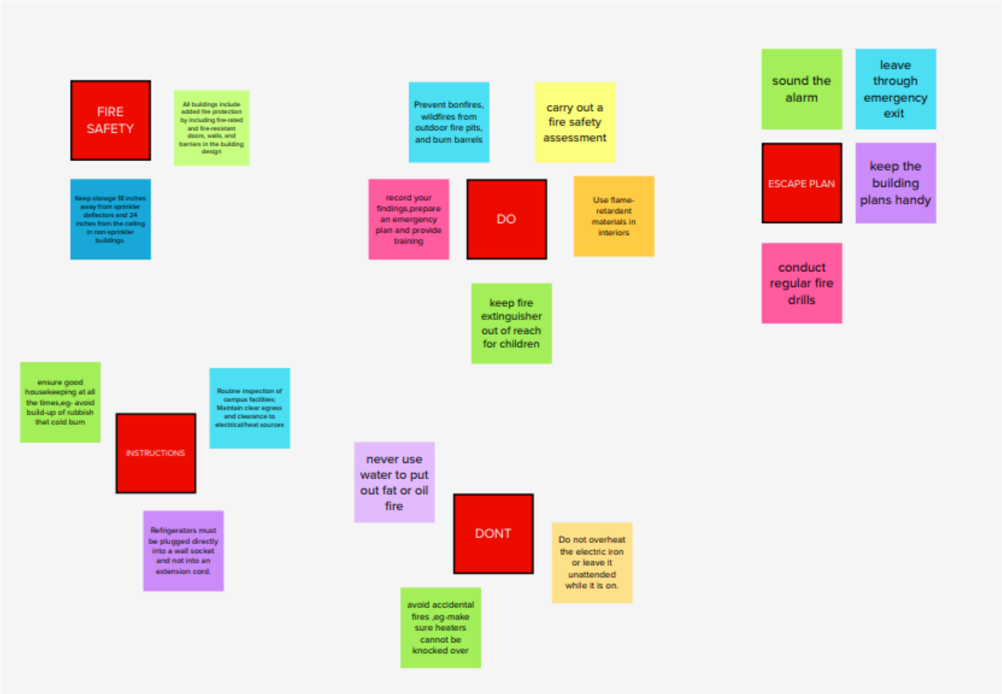

FIRE SAFETY
All buildings include added fire protection by including fire-rated and fire-resistant doors, walls, and barriers in the building design
Keep storage 18 inches away from sprinkler deflectors and 24 inches from the ceiling in non-sprinkler buildings.
Prevent bonfires, wildfires from outdoor fire pits, and burn barrels
carry out a fire safety assessment
record your findings,prepare an emergency plan and provide training
DO
Use flame-retardant materials in interiors
keep fire extinguisher out of reach for children
sound the alarm
leave through emergency exit
ESCAPE PLAN
keep the building plans handy
conduct regular fire drills
ensure good housekeeping at all the times,eg- avoid build-up of rubbish that cold burn
INSTRUCTIONS
Routine inspection of campus facilities; Maintain clear egress and clearance to electrical/heat sources
Refrigerators must be plugged directly into a wall socket and not into an extension cord.
never use water to put out fat or oil fire
DONT
Do not overheat the electric iron or leave it unattended while it is on.
avoid accidental fires ,eg-make sure heaters cannot be knocked over

Fig 3.3 Group Ideas

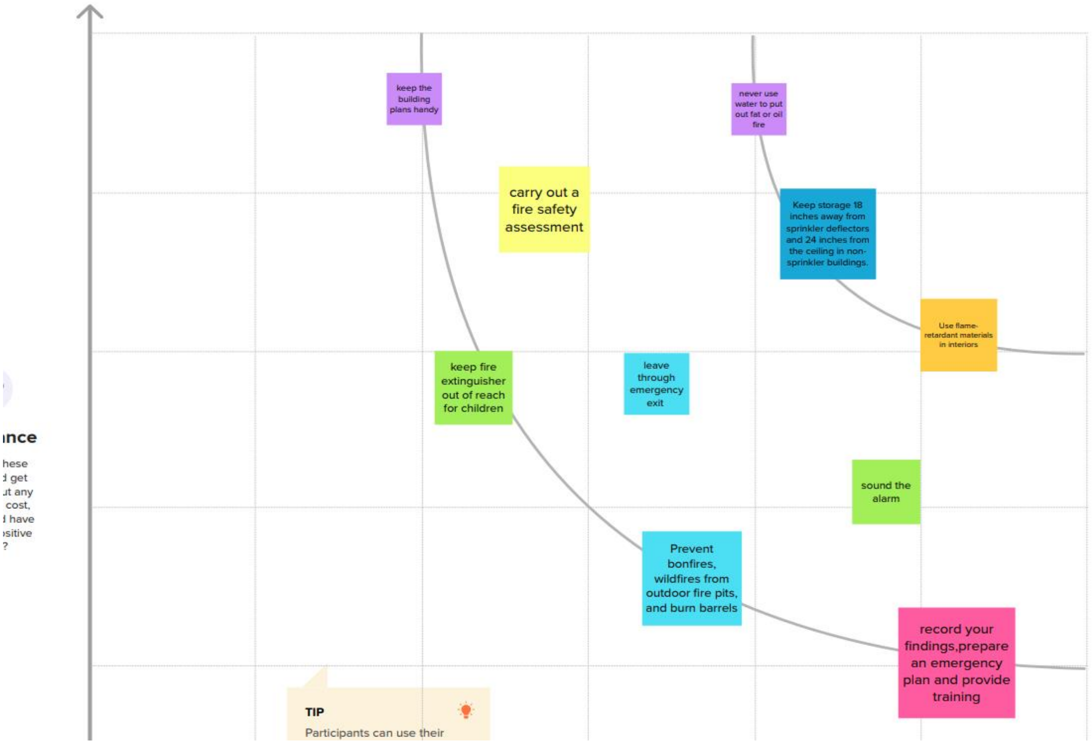
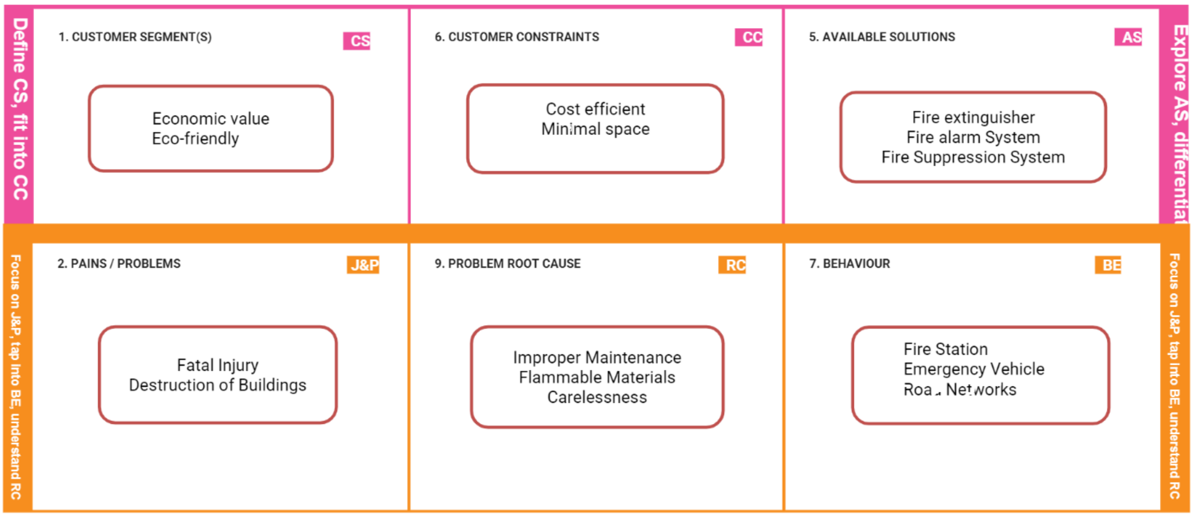

keep the building plans handy
never use water to put out fat or oil fire
carry out a fire safety assessment
Keep storage 18 inches away from sprinkler deflectors and 24 inches from the ceiling in non-sprinkler buildings.
Use flame-retardant materials in interiors
keep fire extinguisher out of reach for children
leave through emergency exit
sound the alarm
Prevent bonfires, wildfires from outdoor fire pits, and burn barrels
record your findings,prepare an emergency plan and provide training
ınce
hese
d get
ıt any
cost,
d have
ositive
?
TIP
Participants can use their

Fig 3.4 Prioritize

3.3 PROPOSED SOLUTION

S. No	Parameter	Description
1	Problem Statement (Problem to be solved)	To enhance the security of the industries that use fire management a warning system to minimize the destruction of live sand property
2	Idea/Solution description	The goal is to find smoke and high temperatures. Additionally, the temperature drops by preserving the air's humidity while putting out fires in the even to fan accident
3	Novelty/Uniqueness	detects the fire even before it begins. Simple administration and effective workflow
4	Social Impact / Customer Satisfaction	Industry workers should put in fearless hours. Substantially reduce on the destruction. To alert everyone, if there is some prudence.
5	Business Model (Revenue Model)	This technology can be used in any environment to detect fires, notify people, and decrease the loss. This system issued in the safety management system to make the most precise predictions.
6	Scalability of the Solution	The size of this system comes up short. Since it needs little time for management, it is simple to keep up with. The system's price is fair.

3.4 Proposed solution fit



Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Economic value
Eco-friendly

6. CUSTOMER CONSTRAINTS CC

Cost efficient
Minimal space

5. AVAILABLE SOLUTIONS AS

Fire extinguisher
Fire alarm System
Fire Suppression System

Explore AS, differentiat

Focus on J&P, tap into BE, understand RC

2. PAINS / PROBLEMS J&P

Fatal Injury
Destruction of Buildings

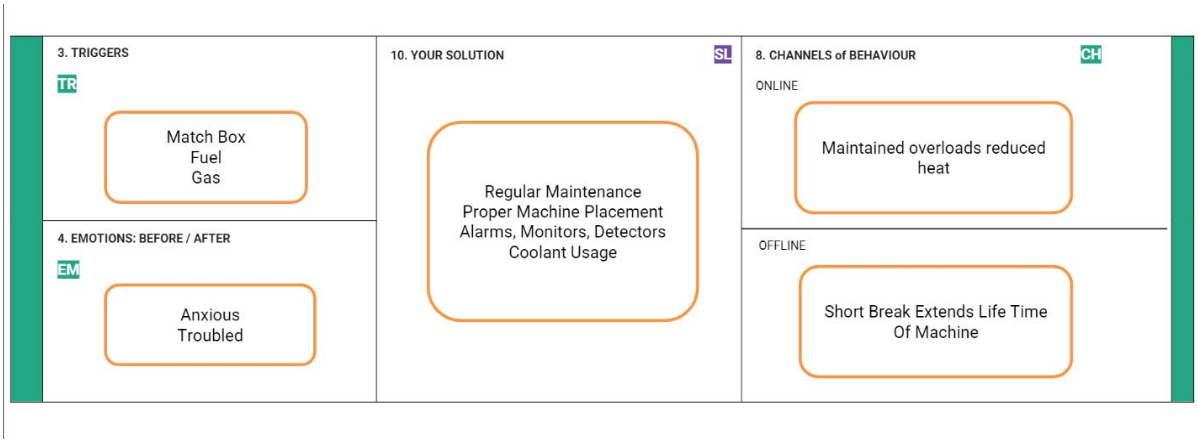
9. PROBLEM ROOT CAUSE RC

Improper Maintenance
Flammable Materials
Carelessness

7. BEHAVIOUR BE

Fire Station
Emergency Vehicle
Road Networks

Focus on J&P, tap into BE, understand RC



3. TRIGGERS TR

Match Box
Fuel
Gas

4. EMOTIONS: BEFORE / AFTER EM

Anxious
Troubled

10. YOUR SOLUTION SL

Regular Maintenance
Proper Machine Placement
Alarms, Monitors, Detectors
Coolant Usage

8. CHANNELS of BEHAVIOUR CH

ONLINE

Maintained overloads reduced heat

OFFLINE

Short Break Extends Life Time Of Machine

CHAPTER – 4

REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

- ❖ A functional requirement establishes the behaviour between inputs and outputs that characterises a system's or a component's function.
- ❖ What should the software system accomplish is specified**?**
- ❖ Defined at a component level
- ❖ Usually easy to define
- ❖ Helps you verify the functionality of the software

FR No.	**Functional Requirement (Epic)**	**Sub Requirement (Story/Sub-Task)**
FR-1	User Registration	**LOWVISION:** ● I want to be able to enlarge the text on the screen because I have flow vision and have difficulty reading. ● Sign up with Gmail
FR-2	User Confirmation	**USERWHOISHEARINGIMPAIRED**: ● I need to enable video captioning to comprehend what is being said in videos. ● Email confirmation required
FR-3	User Login	Utilize the appropriate login and password to log into the website or mobile application.
FR-4	User Access	Checks out the app's prerequisites
FR-5	User Upload	Data uploading by the user is necessary.
FR-6	User Solution	Every 24 hours, a data report should be created and provided to the user.
FR-7	User Data Sync	Enhancing the invoicing system using an API interface

4.2 Non-Functional Requirements

- ❖ A non-functional requirement identifies a software system's quality characteristic.
- ❖ The problem of "How should the software system fulfil the functional requirements?" is constrained.
- ❖ It is not mandatory
- ❖ Applied to system as a whole
- ❖ Usually more difficult to define
- ❖ Helps you verify the performance of the software

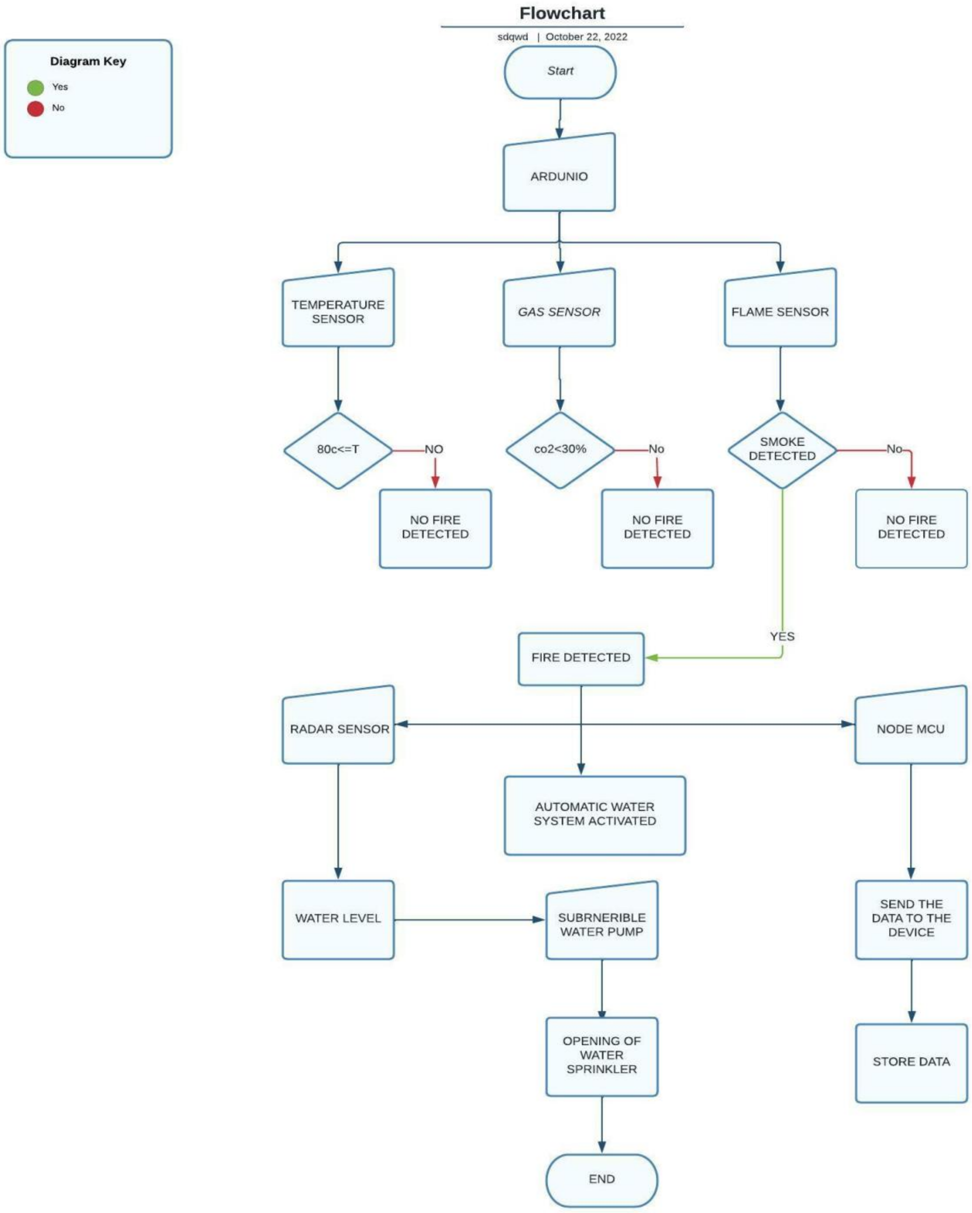
FR No.	**Non-Functional Requirement**	**Description**
NFR-1	**Usability**	● Usability requirements include language barriers and localization tasks. ● Visual and audio signalization ● Protect your property
NFR-2	**Security**	Access permissions for the particular system information may only be changed by the system’s data administrator.
NFR-3	**Reliability**	When any update fails, the database update procedure must roll back any linked updates.
NFR-4	**Performance**	Find a fire. Inform the in habitant soft the fire situation. Turn on the safety control functions. Notify your local fire station.
NFR-5	**Availability**	● Wireless Alarm Systems. ● Wired Alarm Systems. ● Monitored Alarm Systems. ● Unmonitored Alarm Systems

NFR-6	**Scalability**	● In comparison to traditional systems, addressable systems provide a tremendous level of flexibility. ● While the number of devices that any system can support is determined by the manufacturer of the alarm panels, every type of device added to a traditional system necessitates the creation of a new circuit.

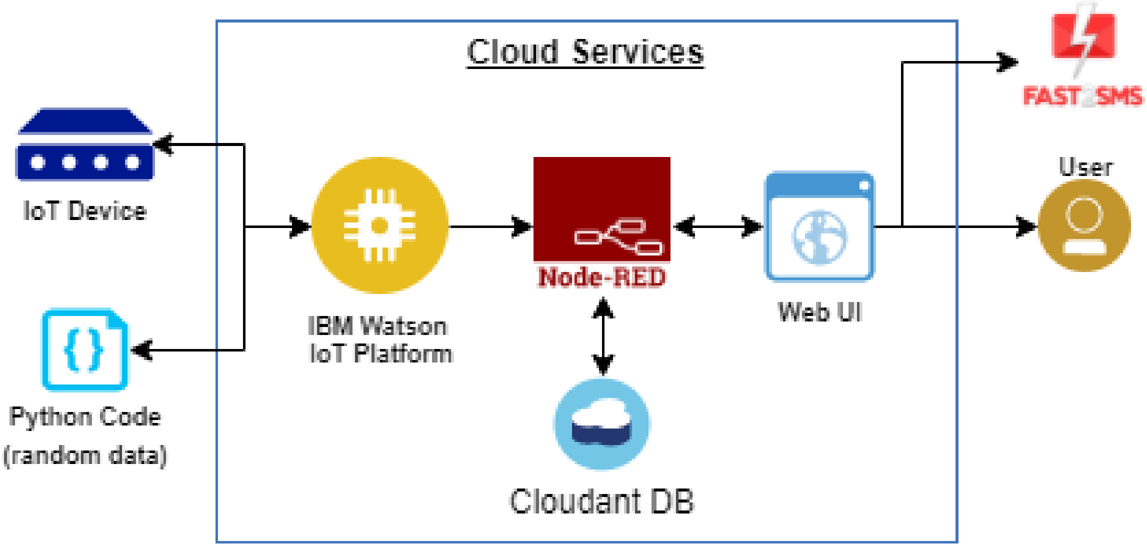
CHAPTER – 5

PROJECT DESIGN

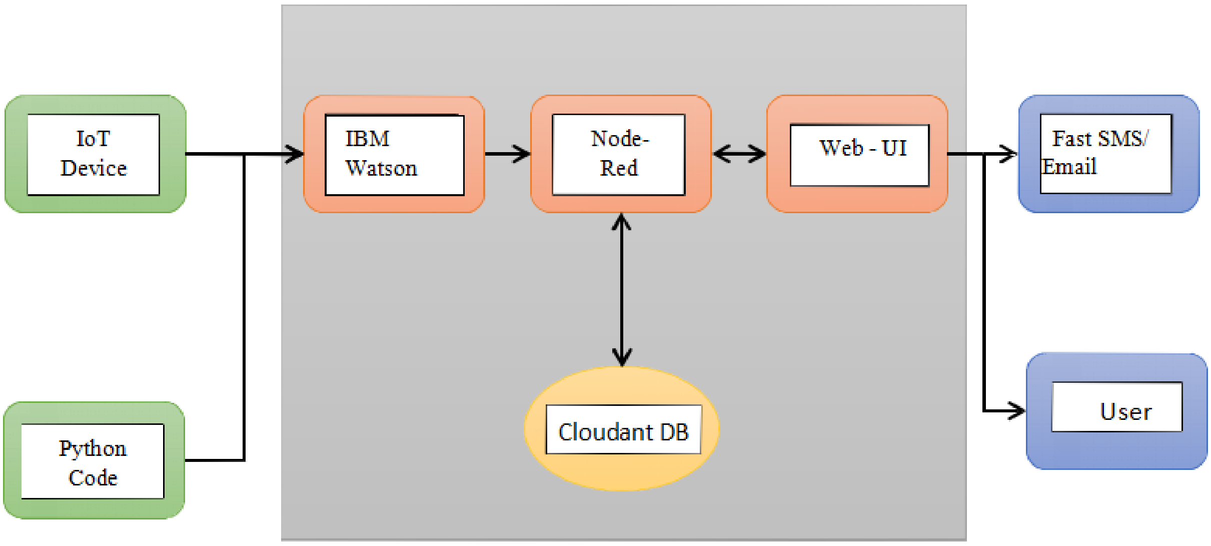
5.1 DATAFLOW DIAGRAM



5.2 SOLUTION AND TECHNICAL ARCHITECTURE:



5.3 CLOUD SERVICES:



5.4 USER STORIES

USER TYPE	FUNCTIONAL REQUIREMENT	USER STORY NUMBER	USER STORY/TASK	ACCEPTANCE CRITERIA	PRIORITY	RELEASE
CUSTOMER (MOBILE USER , WEB USER, CARE EXECUTIVE, ADMINISTRATOR)	REGISTRATION	USN-1	IN ORDER TO REGISTER FOR THE APPLICATION, I MUST ENTER MY EMAIL ADDRESS, PASSWORD, AND CONFIRM MY PASSWORD.	I CAN ACCESS MY ACCOUNT/DASHBOARD	HIGH	SPRINT-1
		USN-2	ONCE I'VE REGISTERED FOR THE APPLICATION, I WILL RECEIVE CONFIRMATION EMAIL AS A USER.	I CAN RECEIVE CONFRIMATION EMAIL & CLICK CONFRIM	HIGH	SPRINT-1
	DASHBOARD	USN-3	I CAN SIGN UP FOR THE APPLICATION AS A USER VIA THE INTERNET.	I CAN REGISTER & ACCESS THE DASHBOARD WITH INTERNET LOGIN	LOW	SPRINT-2

		USN-4	I CAN APPLICATION USER REGISTRATION FOR THE APPLICATION VIA EMAIL	I CAN CONFRIM THE REGISTREATION IN GMAIL	MEDIUM	SPRINT-1
	LOGIN	USN-5	IN ORDER TO ACCESS THE APPLICATION, I MUST ENTER MY EMAIL ADDRESS AND PASSWORD.	I CAN LOGIN WITH MY ID AND PASSWORD	HIGH	SPRINT-1

CHAPTER – 6

PROJECT DESIGN AND PLANNING

6.1 SPRINT PLANNING AND ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story/Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Sowndhariya. V
Sprint-1	Confirmation	USN-2	As a user, I will receive confirmation email once I have registered for the application.	1	High	Soumiya. K
Sprint-	Login	USN-3	As a user, I can log into the application by entering email &password.	1	High	Sarveshan. k

Sprint	**Functional Requirement (Epic)**	**User Story Number**	**User Story/Task**	**Story Points**	**Priority**	**Team Members**
Sprint-2	Simulation	USN-4	As a user, I can register for the application through Facebook.	2	Low	Sri Hari. V
Sprint-2	Software	USN-5	Develop a python code to publish random sensor data.	1	Medium	Sowndhariya. V
Sprint-2	Simulation	USN-6	Connect the data with IBM cloud.	2	Medium	Soumiya. K
Sprint-3	Simulation	USN-7	Establishing Node-RED connection	2	Low	Sarveshan. k
Sprint-3	App development	USN-8	Application development using MIT app inventor.	2	Medium	Sowndhariya. V
Sprint-4	Simulation	USN-9	Connecting the developed application with Node RED.	2	Low	Soumiya. K
Sprint-4	Testing	USN-10	Testing the application.	2	High	Sri Hari. V

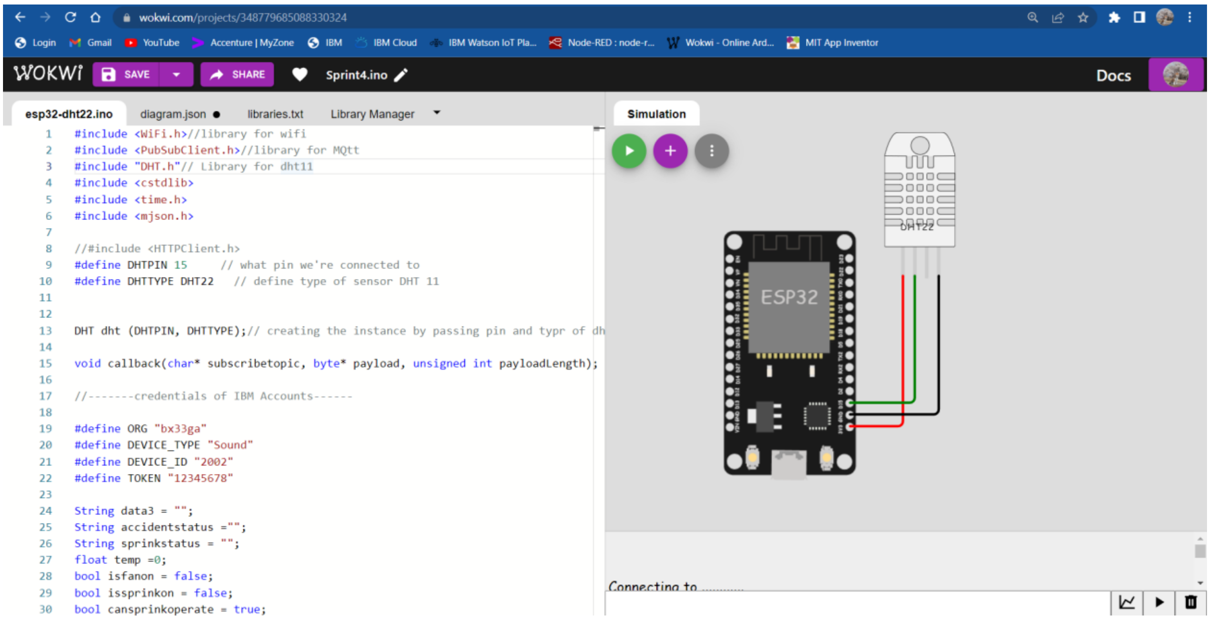
6.2 Sprint delivery schedule

Sprint	**Total Story Points**	**Duration**	**Sprint Start Date**	**Sprint End as on Date (Planned)**	**Story Points Completed (Planned End Date)**	**Sprint Release Date (Actual)**
Sprint-1	20	6Days	24Oct2022	29Oct2022	20	10Nov2022
Sprint-2	20	6Days	31Oct2022	05Nov2022	20	12Nov2022
Sprint-3	20	6Days	07Nov2022	12Nov2022	20	15Nov2022
Sprint-4	20	6Days	14Nov2022	19Nov2022	20	16Nov2022

CHAPTER 7

CODING AND SOLUTION

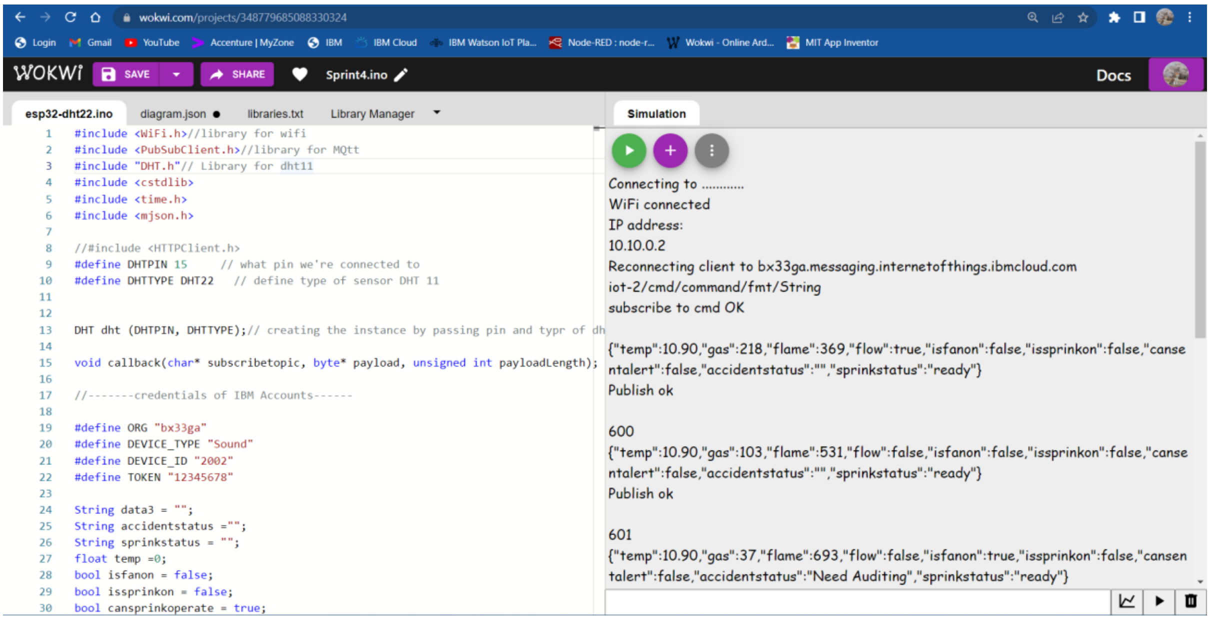
7.1 FEATURE 1:



```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt
#include "DHT.h"// Library for dht11
#include <cstdlib>
#include <time.h>
#include <mjson.h>

//#include <HTTPClient.h>
#define DHTPIN 15     // what pin we're connected to
#define DHTTYPE DHT22   // define type of sensor DHT 11


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

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

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

#define ORG "bx33ga"
#define DEVICE_TYPE "Sound"
#define DEVICE_ID "2002"
#define TOKEN "12345678"

String data3 = "";
String accidentstatus ="";
String sprinkstatus = "";
float temp =0;
bool isfanon = false;
bool issprinkon = false;
bool cansprinkoperate = true;
```

```
Connecting to ............
```

Fig 7.1 Wokwi Code

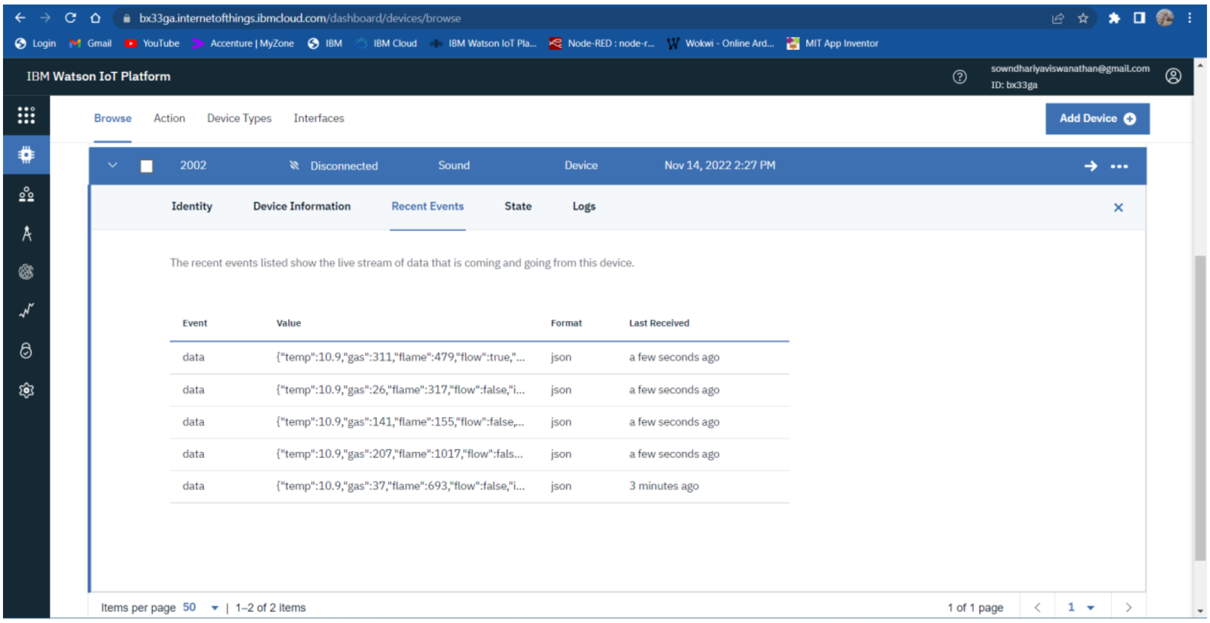


```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt
#include "DHT.h"// Library for dht11
#include <cstdlib>
#include <time.h>
#include <mjson.h>

//#include <HTTPClient.h>
#define DHTPIN 15     // what pin we're connected to
#define DHTTYPE DHT22   // define type of sensor DHT 11


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

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

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

#define ORG "bx33ga"
#define DEVICE_TYPE "Sound"
#define DEVICE_ID "2002"
#define TOKEN "12345678"

String data3 = "";
String accidentstatus ="";
String sprinkstatus = "";
float temp =0;
bool isfanon = false;
bool issprinkon = false;
bool cansprinkoperate = true;
```

```
Connecting to ............
WiFi connected
IP address:
10.10.0.2
Reconnecting client to bx33ga.messaging.internetofthings.ibmcloud.com
iot-2/cmd/command/fmt/String
subscribe to cmd OK

{"temp":10.90,"gas":218,"flame":369,"flow":true,"isfanon":false,"issprinkon":false,"canse
ntalert":false,"accidentstatus":"","sprinkstatus":"ready"}
Publish ok

600
{"temp":10.90,"gas":103,"flame":531,"flow":false,"isfanon":false,"issprinkon":false,"canse
ntalert":false,"accidentstatus":"","sprinkstatus":"ready"}
Publish ok

601
{"temp":10.90,"gas":37,"flame":693,"flow":false,"isfanon":true,"issprinkon":false,"cansen
talert":false,"accidentstatus":"Need Auditing","sprinkstatus":"ready"}
```

Fig 7.2 Wokwi Simulation

7.2 FEATURE 2:

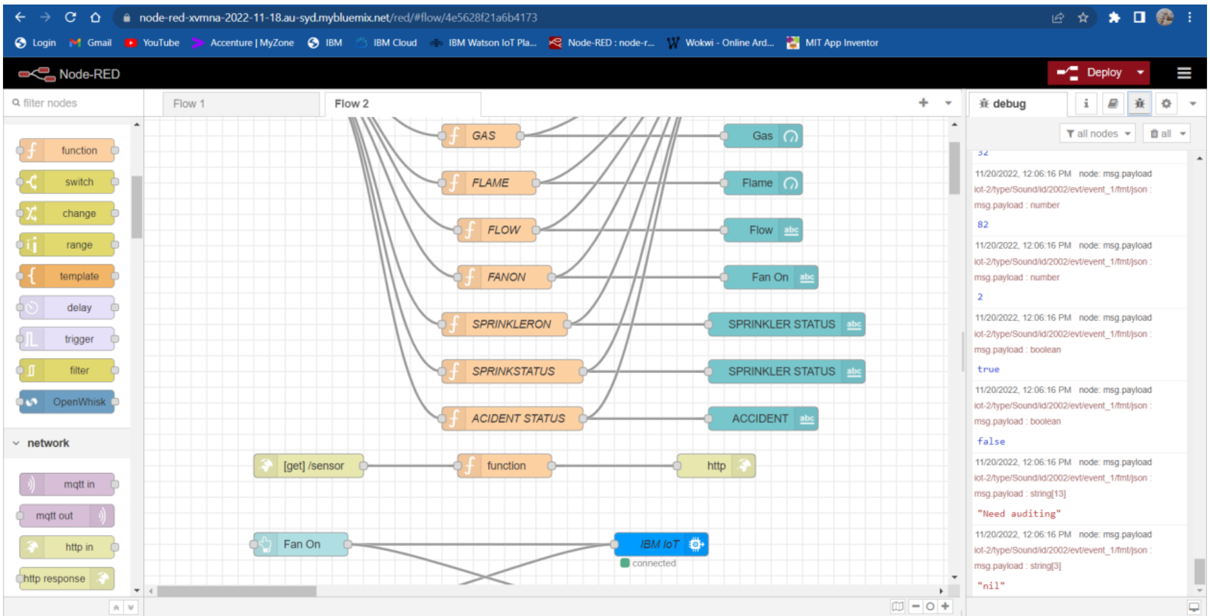


The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
data	{"temp":10.9,"gas":311,"flame":479,"flow":true,"...	json	a few seconds ago
data	{"temp":10.9,"gas":26,"flame":317,"flow":false,"i...	json	a few seconds ago
data	{"temp":10.9,"gas":141,"flame":155,"flow":false,...	json	a few seconds ago
data	{"temp":10.9,"gas":207,"flame":1017,"flow":fals...	json	a few seconds ago
data	{"temp":10.9,"gas":37,"flame":693,"flow":false,"i...	json	3 minutes ago

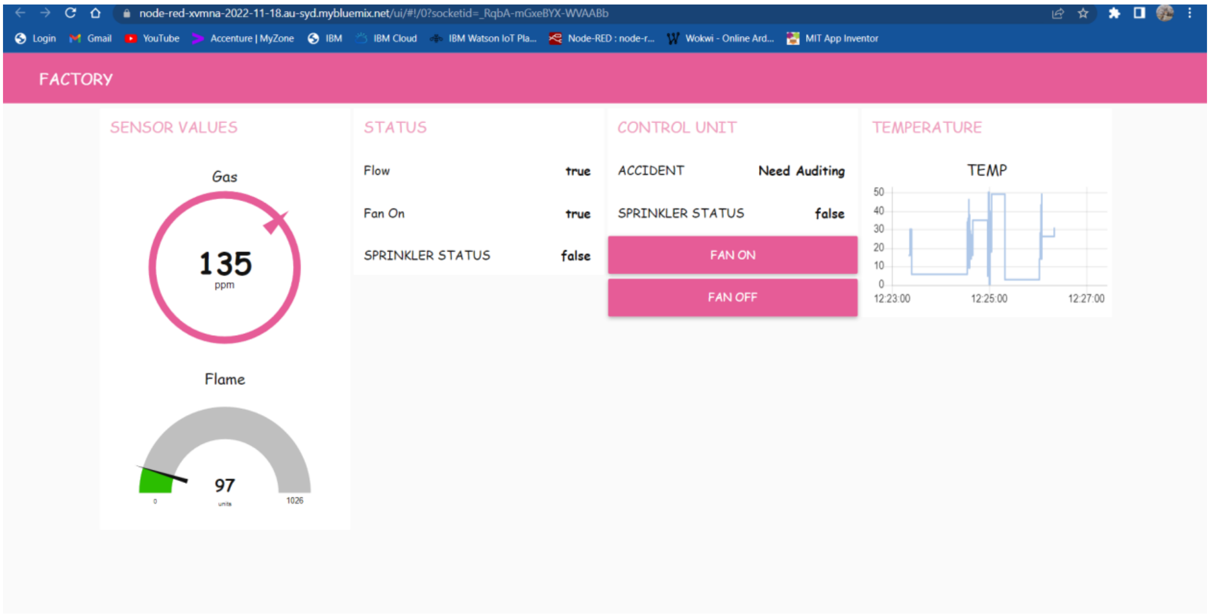
Items per page 50 | 1–2 of 2 items 1 of 1 page 1

Fig 7.3 IBM Watson Output



```
32
11/20/2022, 12:06:16 PM   node: msg.payload
iot-2/type/Sound/id/2002/evt/event_1/fmt/json :
msg.payload : number
82
11/20/2022, 12:06:16 PM   node: msg.payload
iot-2/type/Sound/id/2002/evt/event_1/fmt/json :
msg.payload : number
2
11/20/2022, 12:06:16 PM   node: msg.payload
iot-2/type/Sound/id/2002/evt/event_1/fmt/json :
msg.payload : boolean
true
11/20/2022, 12:06:16 PM   node: msg.payload
iot-2/type/Sound/id/2002/evt/event_1/fmt/json :
msg.payload : boolean
false
11/20/2022, 12:06:16 PM   node: msg.payload
iot-2/type/Sound/id/2002/evt/event_1/fmt/json :
msg.payload : string[13]
"Need auditing"
11/20/2022, 12:06:16 PM   node: msg.payload
iot-2/type/Sound/id/2002/evt/event_1/fmt/json :
msg.payload : string[3]
"nil"
```

Fig 7.4 NODE RED Connection



STATUS

Flow true

Fan On true

SPRINKLER STATUS false

CONTROL UNIT

ACCIDENT Need Auditing

SPRINKLER STATUS false

Fig 7.5 Output

CHAPTER-8

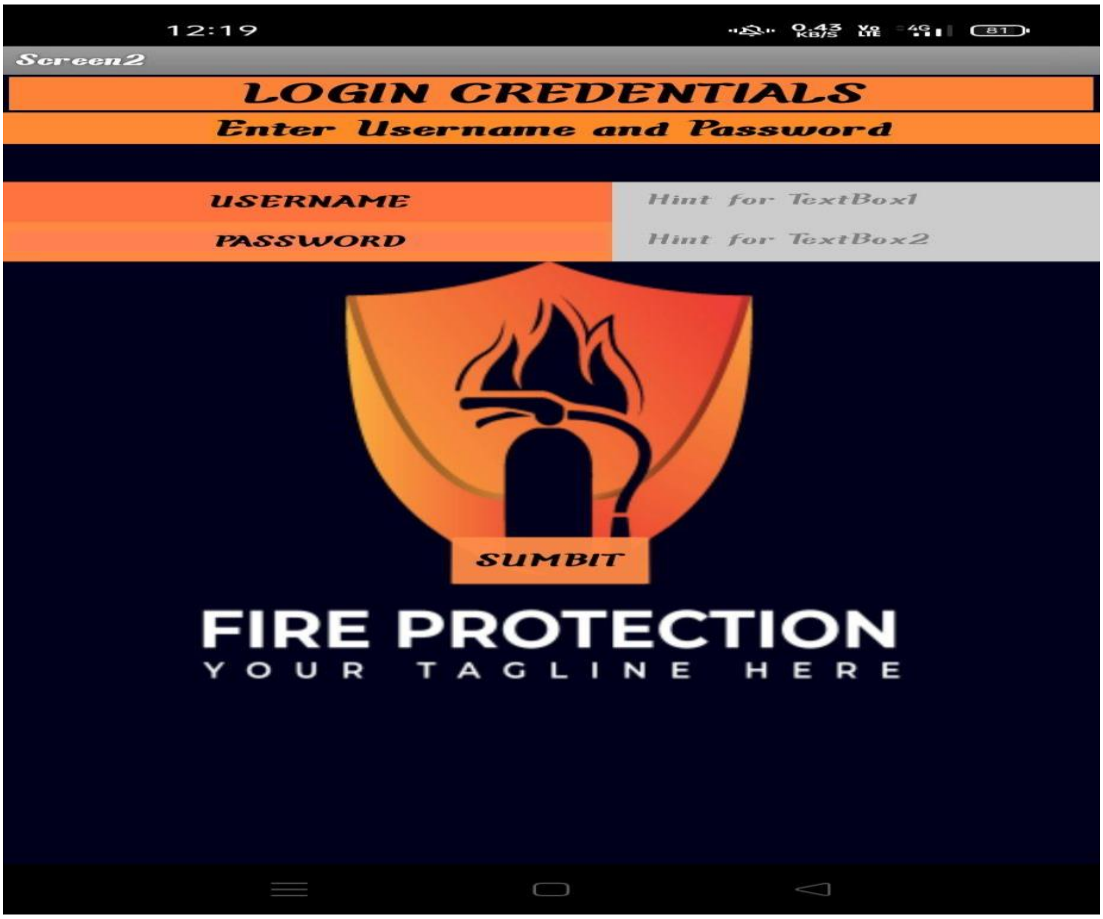
TESTING

8.1 TEST CASES:

Step-1: First the user needs to download the android APK file from MIT app inventor where we developed our mobile application and install in their mobiles.

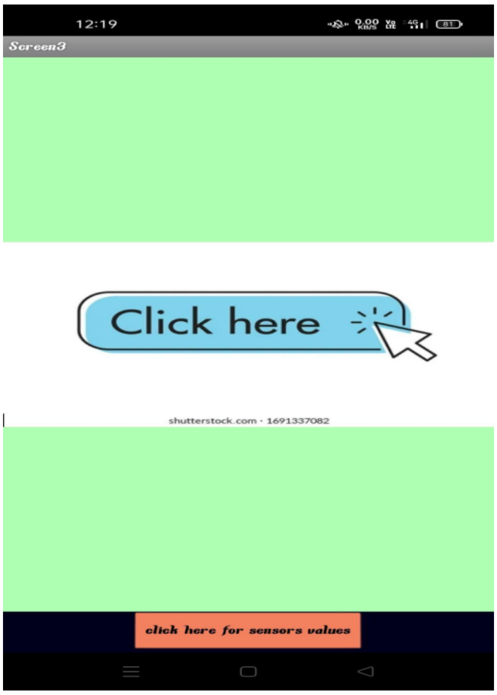
Step-2: After successful installation we can find app icon in our mobile as shown below. After clicking the app icon. The user can see the home page

STEP - 3 The user needs to create username and password.so give username and password and click the signup button. The user can see interface like these as shown below.

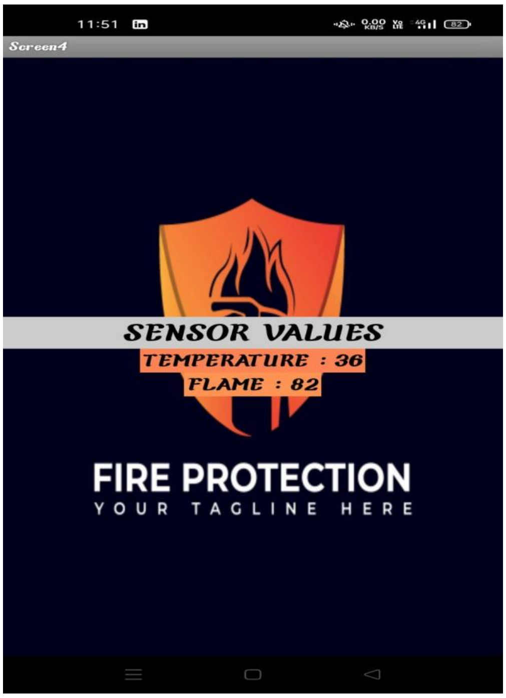


8.2 USER ACCEPTANCE TESTING

After successful login. The next page will be open. In that page we can see the click button to see the temperature values and flame value.



STEP-4 The final values has been displayed



CHAPTER 9

RESULT

9.1 PERFORMANCE METRICS

9.1.1 CPU USAGE

The CPU is utilised to its fullest extent by the tiny version of C++. The programme runs in O(1) time for each loop, ignoring the network and communication. To improve communication with MQTT, the software naps every 1 second. Because the programme runs in O(1) time and the compiler. Because the software is optimised during compilation, there is reduced CPU burden for each cycle. The following instructions are stored on the stack memory and can be popped after execution.

9.1.2 MEMORY USAGE

Sensor values and networking data are saved in the ESP32's sram. It's a lot of info because the ESP32 only has 520 KB of RAM. To preserve memory and ensure optimal programme performance, the exact addresses are rewritten with fresh values for each memory cycle**.**

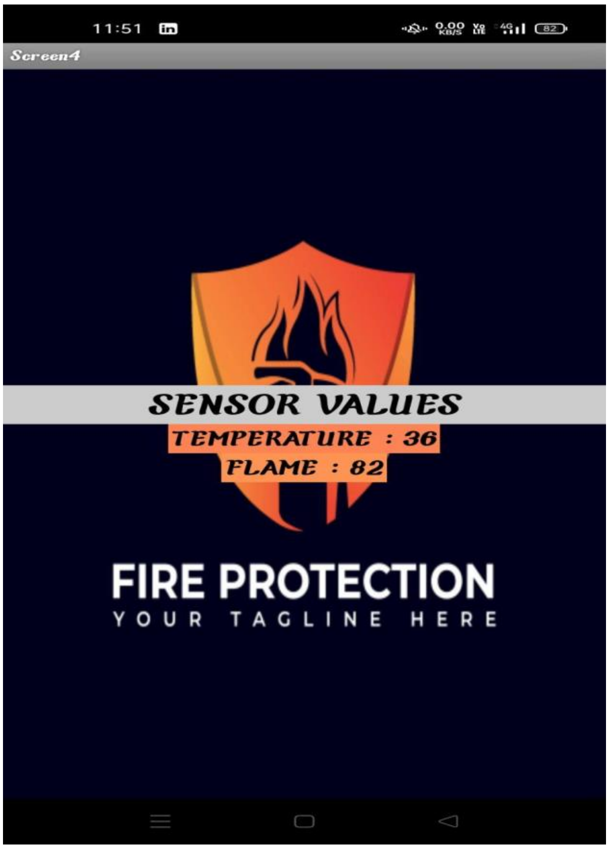
9.1.3 ERROR RATES

Because of backend and dashboard are handled by node-red, the error rates are quite low. Exceptions are handled properly so that the system's usability is not harmed.

9.1.4 GARBAGE COLLECTION:

The Node framework handles garbage collection on the server side. C++ does not have garbage collection features in IoT devices. However, in this case, it is not necessary because the memory will be used again to store the data. There is no allocation of any dangling pointers or poorly handled address space.

9.2 FINAL OUTPUT



SENSOR VALUES

TEMPERATURE : 36

FLAME : 82

CHAPTER 10

ADVANTAGES & DISADVANTAGES

10.1 ADVANTAGES:

- Active fire and gas leak detection
- SMS-based automatic notification of administrative and fire authorities
- Automatically turning on and off sprinklers and exhaust fans
- Authentication is required to manually turn on/off sprinklers and exhaust fans, as well as send SMS alerts.
- It detects false fire breakouts automatically, avoiding unneeded alarm.
- We can validate that the sprinkler system is working properly by using flow sensors.
- A dashboard can display all device statuses.
- A web application allows users to view the dashboard.

10.2 DISADVANTAGES:

- It is always necessary to connect to the internet. [Only for sending SMS alerts]
- If the physical device fails, the entire operation fails.
- Because so much data is kept in the cloud database every second, a huge database is required.

CHAPTER 11

CONCLUSION AND FUTURE SCOPE

11.1 CONCLUSION:

Finally, as part of a building's safety strategy, a fire prevention system should be installed. Without a fire prevention system, persons within the structure face a significant danger of death in the event of an emergency. The systems approved for usage, such as fire alarm systems, sprinkler systems, fire pumps, and smoke control systems, use a variety of measures to notify of fire and smoke conditions, aid limit the spread of the fire, or help put out the fire entirely.

It will be handle using IoT devices by developing a smart management system that overcomes several fundamental flaws in traditional fire management systems, such as actively monitoring for fire breakouts and gas leaking and sending SMS notifications to administrators and fire authorities.

11.2 FUTURE SCOPE:

The future will be fire detection, in which the detector will be more of a sensor, detecting combustion products such as carbon monoxide, carbon dioxide, sulphur dioxide, nitrogen oxides, as well as heat and particulate matter. Sensors will also be able to detect or track whether a room is occupied or not, and will be combined with occupant notification and evacuation. The development of more powerful algorithms and artificial intelligence, both within the sensor and the frontend control unit, will reduce the time between the start of an event and its notification.

CHAPTER 12

APPENDIX

12.1 SOURCE CODE:

```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt
#include "DHT.h"// Library for dht11
#include <cstdlib>
#include <time.h>
#include <mjson.h>

//#include <HTTPClient.h>
#define DHTPIN 15     // what pin we're connected to
#define DHTTYPE DHT22   // define type of sensor DHT 11


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 "bx33ga"
#define DEVICE_TYPE "Sound"
#define DEVICE_ID "2002"
#define TOKEN "12345678"

String data3 = "";
String accidentstatus ="";
String sprinkstatus = "";
float temp =0;
bool isfanon = false;
bool issprinkon = false;
bool cansprinkoperate = true;
bool canfanoperate = true;
bool canalertsent = true;
bool cansentalert = false;
int gas = 0;
int flame = 0;
int flow = 0;
long int cooldown= 600;

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";

char publishTopic[] = "iot-2/evt/data/fmt/json";
char subscribetopic[] = "iot-
2/cmd/command/fmt/String";
```

```
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_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();
  //if real gas sensor is used make sure the senor is heated up for acurate
readings
  /*
    - Here random values for readings and stdout were used to show the
      working  of the devices as physical or simulated devices are not
      available.
  */


  delay(10);
  Serial.println();
  wificonnect();
  mqttconnect();
}

void loop()// Recursive Function
{
  //printf("\n%li\n",time(0));


  temp = dht.readTemperature();
  //setting a random seed
  srand(time(0));

  //initial variable activities like declaring , assigning
  gas = rand()%400;
  int flamereading = rand()%1024;
  flame = map(flamereading,0,1024,0,1024);
  int flow = ((rand()%100)>50?1:0);



  //find the accident status 'cause fake alert may be caused by some mischief
activities
  if(temp < 45 ){
    if(flame > 650 ){
      accidentstatus = "Need Auditing";
      if(canfanoperate)
        isfanon = true;
      else
        isfanon = false;
      issprinkon = false;
    }
    else if(flame <= 10){
      accidentstatus = "nothing happened";
      isfanon = false;
      issprinkon = false;
    }

  }else if(temp >= 45 && temp <= 55 ){
    if(flame <=650 && flame >100  ){

      if(cansprinkoperate)
        issprinkon = true;
      else
        issprinkon = false;
      accidentstatus = "moderate";
      if(gas > 160 && canfanoperate ){
        isfanon = true;
      }
      else{
        isfanon = false;
      }
    }
    else if(flame <= 100 && flame > 10){
     if(cansprinkoperate)
        issprinkon = true;
      else
        issprinkon = false;
      isfanon = false;
      accidentstatus = "moderate";
    }

  }else if(temp > 55){
    if(flame > 650){
      gas = 500 + rand()%500;
      accidentstatus = "severe";
      if(cansprinkoperate)
        issprinkon = true;
      else
        issprinkon = false;
      if(canfanoperate)
        isfanon = true;
      else
        isfanon = false;
    }
    else if(flame < 650 && flame > 400 ){
      gas = 300 + rand()%500;
      accidentstatus = "severe";
      if(cansprinkoperate)
        issprinkon = true;
      else
        issprinkon = false;

      if(canfanoperate)
        isfanon = true;
      else
        isfanon = false;

    }
  }
  else {
    accidentstatus = "Need moderate Auditing";
    isfanon = false;
    issprinkon = false;
  }


  if(issprinkon){
    if(flow){
      sprinkstatus = "working";
    }
    else{
      sprinkstatus = "not working";
    }
  }
  else if(!issprinkon){
    sprinkstatus = "ready";
  }
  else {
    sprinkstatus = "something's wrong";
  }


  PublishData(temp,gas,flame,flow,isfanon,issprinkon);
  printf("\n%li\n",cooldown);
  if(accidentstatus=="severe" && cooldown >= 600){
    cooldown = 0;
    sendalert();
    PublishData(temp,gas,flame,flow,isfanon,issprinkon);
    cansentalert = false;

  }

  if(cooldown > 999999){
    cooldown = 601;
  }
  + delay(1000);
  ++cooldown;
  if (!client.loop()) {
    mqttconnect();
  }
}

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


void PublishData(float temp, int gas ,int flame ,int flow,bool isfanon,bool
issprinkon) {
  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 += "," "\"gas\":";
  payload += gas;
  payload += "," "\"flame\":";
  payload += flame;
  payload += "," "\"flow\":";
  payload += ((flow)?"true":"false");
  payload += "," "\"isfanon\":";
  payload += ((isfanon)?"true":"false");
  payload += "," "\"issprinkon\":";
  payload += ((issprinkon)?"true":"false");
  payload += "," "\"cansentalert\":";
  payload += ((cansentalert)?"true":"false");
  payload += "," "\"accidentstatus\":";
  payload += "\""+accidentstatus+"\"";
  payload += "," "\"sprinkstatus\":";
  payload += "\""+sprinkstatus+"\"";
  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(100);
    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);

  const char *s =(char*) data3.c_str();
  double pincode = 0;

  if(mjson_get_number(s, strlen(s), "$.pin", &pincode)){
```

```
  if(((int)pincode)==67993){
    const char *buf;
    int len;

    if (mjson_find(s, strlen(s), "$.command", &buf, &len))  // And print
it
    {

     String command(buf,len);
     if(command=="\"cantfan\""){

      canfanoperate = !canfanoperate;
     }
     else if(command=="\"cantsprink\""){
      cansprinkoperate = !cansprinkoperate;
     }else if(command=="\"sentalert\""){
      resetcooldown();
     }
    }



  }
 }

 data3="";
}

void resetcooldown(){
 cooldown = 0;
}

void sendalert(){

 cansentalert = true;
 cooldown = 0;


}
```

12.2 GITHUB LINK

https://github.com/IBM-EPBL/IBM-Project-221411659805786