

PROJECT REPORT

INDEX

1. INTRODUCTION

1.1 Project Overview

1.2 Purpose

2. LITERATURE SURVEY

2.1 Existing problem

2.2 References

2.3 Problem Statement Definition

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

3.2 Ideation & Brainstorming

3.3 Proposed Solution

3.4 Problem Solution fit

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

4.2 Non-Functional requirements

5. PROJECT DESIGN

5.1 Data Flow Diagrams

5.2 Solution & Technical Architecture

5.3 User Stories

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

6.2 Sprint Delivery Schedule

6.3 Reports from JIRA

7. CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 Feature 1

7.2 Feature 2

7.3 Database Schema (if Applicable)

8. TESTING

8.1 Test Cases

8.2 User Acceptance Testing

9. RESULTS

9.1 Performance Metrics

10. ADVANTAGES & DISADVANTAGES

11. CONCLUSION

12. FUTURE SCOPE

13. APPENDIX Source Code GitHub & Project Demo Link

Project Report

1. INTRODUCTION

1.1 Project Overview

In this project "Industry specific intelligent fire management system", we bring together things like Fire detection, evacuation, personal alarm systems, web based application which is used to report fire incidents immediately.

1.2 Purpose

They help in detecting fire or smoke at an early stage and can help in saving lives. Commercial Fire detecting systems usually have an alarm signaling, with the help of a buzzer or Siren. We have designed an IOT based Fire Alerting System using Temperature and a smoke sensor.

2. LITERATURE SURVEY

2.1 Existing problem

We Studied the problems faced by factory workers in times when fire breaks out. With respect to this issue people then proposed a system using Raspberry Pi 3 which is capable of detecting fire and providing information about area of fire. The Raspberry Pi controls multiple Arduino boards which are connected with several motors and cameras to capture the fire incident. In this, they discussed about the modern technology that can be used to reduce extremely unfortunate accidents caused by fire. We designed the whole system and calculated its effectiveness. Also proposed a model for location enhancement and personnel tracking using Wi-Fi networks.

The sensor nodes are placed in important areas of the building, which we create a network and the monitored data is transmitted to control unit through wireless sensor network and if the temperature or pressure reach above the threshold value and building damage is detected automatically, alerts the surroundings and take necessary precautions to prevent the disaster. This safety system that can be used in any constructing and constructed environments. The sensor nodes detects the maximum level that it can withhold, in the mean time it calculates where the damage is occurring and remaining time that the building can offer further resistance to damage.

2.2 References

- Web link:
<https://medcraveonline.com/MOJCE/fire-safety-systems-in-buildings-problems-and-concerns-beyond-the-project.html>
- Tao Chen *et al.*
[Experimental study of evacuation from a 4-storey building](#)
Procedia. Eng.(2013)
- Adam Cowlard *et al.*
[Fire safety design for tall buildings](#) Procedia. Eng.(2013)
- Margrethe Kobes *et al.*
[Building safety and human behaviour in fire: a literature review](#)
Fire Saf. J.(2010)
- M. Agung Wibowo *et al.*
[The analysis of supply chain performance measurement at construction](#)

project

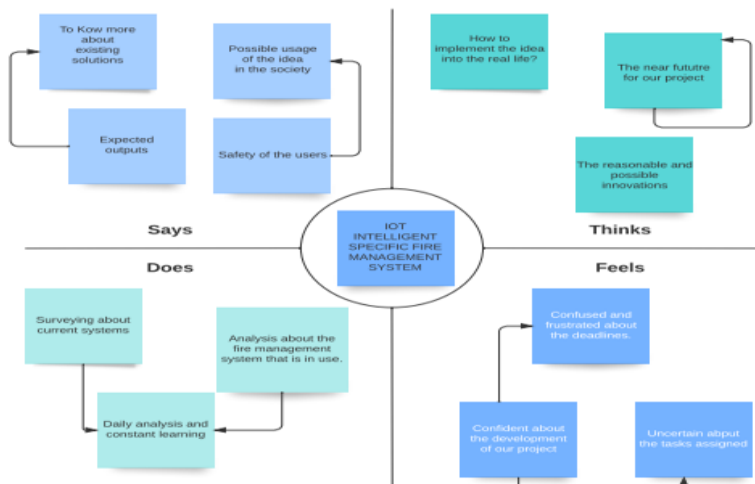
Procedia. Eng.(2015)

2.3 Problem Statement Definition

This intelligent fire alarm system is to sense true occurrences of fire, alert the proper authorities so that people can be evacuated safely.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming

Brainstorm & Idea Prioritization Template:

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

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.

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

The template is divided into two main sections: **Brainstorm & idea prioritization** and **Define your problem statement**.

Brainstorm & idea prioritization:

- Before you collaborate:** A list of all participants gets a hangout with this session. How's that you need to do to get going?
- Team gathering:** Define your problem statement in the session and start on it. How will you collaborate on it?
- Set the goal:** How will the problem get by focusing on solving it in the brainstorming session?
- Start your session:** How will you collaborate on it? How will you collaborate on it?

Define your problem statement:

- Before your problem statement:** Define your problem statement in the session and start on it. How will you collaborate on it?
- Problem:** To detect fire and gas leakage in industries using IOT and to reduce the loss of life and property by deploying IOT based fire detection system.
- Key rules for brainstorming:** Encourage all ideas, no criticism, no evaluation, no judgment, no veto, no criticism, no evaluation, no judgment, no veto.

Step-2: Brainstorm, Idea Listing and Grouping



Step-3: Idea Prioritization



3.3 Proposed Solution

S.No.	Parameter	Description
1	Problem Statement (Problem to be solved)	To improve the safety management system in industries.
2	Idea / Solution description	An Integrated system of temperature monitoring, gas monitoring, fire detection with accuracy of information about locations and response through SMS notification and call.
3	Novelty / Uniqueness	The system utilizes ibm cloud further connected via node red making it reliable.
4	Social Impact / Customer Satisfaction	It avoids the expenditure caused due to the fire in industries. The people in nearby locations can also be alerted .
5	Business Model (Revenue Model)	This product can be utilized by the industries and can be thought of as a productive and helpful item.
6	Scalability of the Solution	The project executes this technique as we need to introduce a system which comprises an Arduino that takes received signals from sensors .The system also provides easy operatability and maintenance.

3.4 Problem Solution fit

Define CS, IR into CL	1. CUSTOMER SEGMENT(S) CS	6. CUSTOMER LIMITATIONS CL	5. AVAILABLE SOLUTIONS AS	Explore AS, differentiate
	Industry members as well as others	The customer should just click the alert message to enhance the further step to stop the fire. Proper network connection and available devices are needed.	The customer used to call for the emergency number 101 to call the fire service team to stop the fire at that time of reporting many products in the industry gets damaged and many lives were death. Now with the use of our product the industry can sense the fire explosion and stop at the initial stage itself. So, it is quite much more easy.	
Focus on PR, Tap into BE, understand RC	2. PROBLEMS / PAINS PR	9. PROBLEM ROOT / CAUSE RC	7. BEHAVIOR BE	Focus on PR, Tap into BE, understand RC
	<ul style="list-style-type: none">We are solving the problem of fire spread by automatically detecting the fire at the ignition stage and stop the fire spread easily using Artificial Intelligence and IOT based ideations.	<ul style="list-style-type: none">The fire causes a lot of damages in the industry. Usually when it gets fired in an industry the fire service team is called to stop the fire. But now our solution use can stop the fire without the help of fire service.	<ul style="list-style-type: none">At once the message is send to the customers mobile from the sensors-controlled intelligence the customer himself can give the access to stop the fire spread on the whole.	
Identify strong TR & EM	3. TRIGGERS TO ACT TR	10. YOUR SOLUTION SL	8. CHANNELS of BEHAVIOR CH	Extract online & offline CH of BE
	We can ask our customer to get an experience about our product. We can insist they must need of our product.	We can just access the message from the IOT devices combined with sensors to stop the fire spread at the ignition stage itself. It is much easier, safe to handle.	ONLINE Notifications send can be accessed.	
	4. EMOTIONS BEFORE / AFTER EM		OFFLINE The sensors with the help of intelligence can stop the fire spread at the initial stage itself.	
	Before: Customer is not finding a proper rid for the fire spread problem. After: Now with the help of our product the customer can easily enhance the problem.			

4. REQUIREMENT ANALYSIS

4.1 Functional requirements

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
1	User Registration	Registration through website or application Registration through Social medias (like Instagram, Facebook)

		Registration through LinkedIN
2	User Confirmation	Verification via Email Verification via OTP
3	User Login	Login through website or App using the respective username and password
4	User Access	Allows the app requirement
5	User Guide	Guides the basic steps of using the application
6	User Upload	User should be able to send the data
7	User Solution	Data report should be generated and delivered to user for per every 24 hours
8	User Data Sync	API interface to increase to invoice system

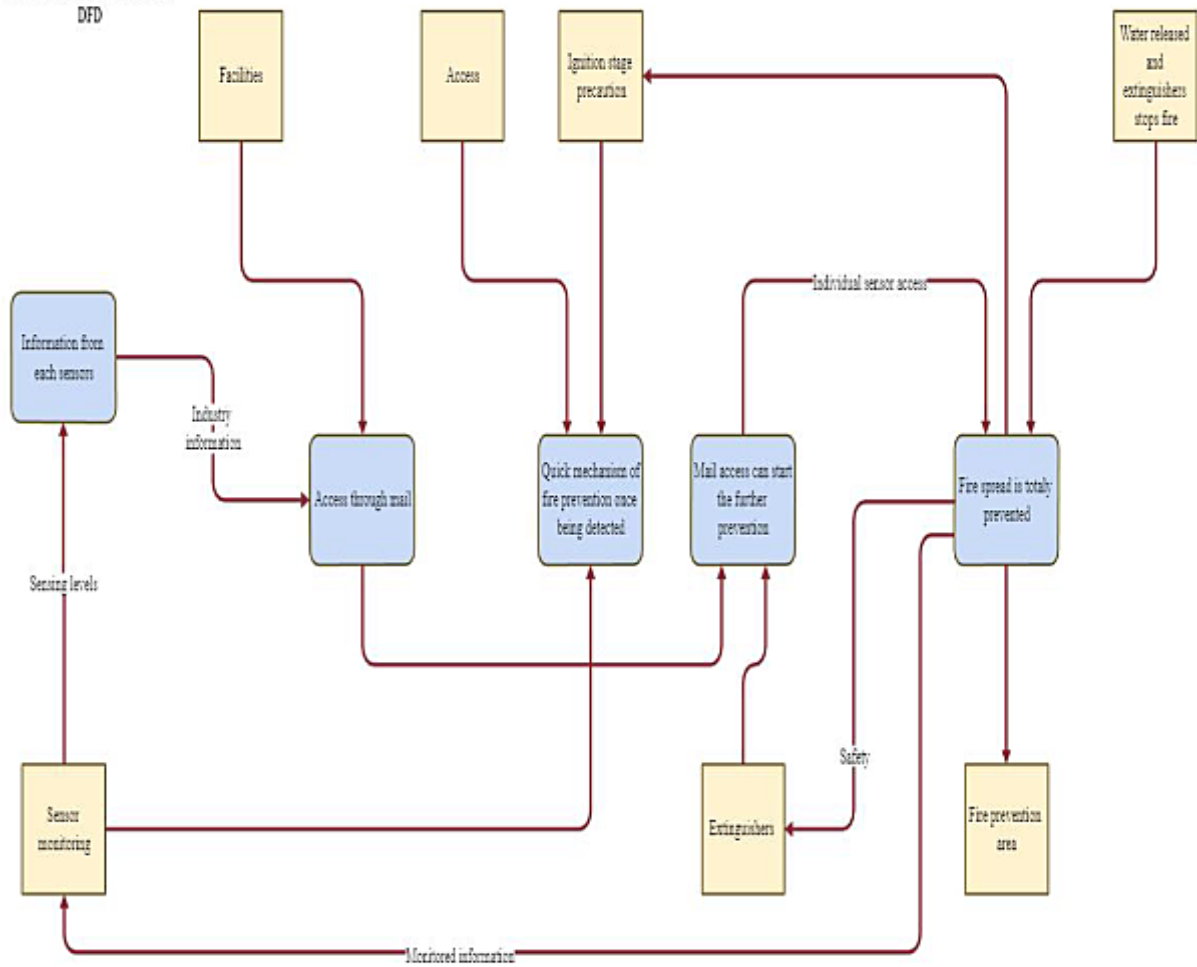
4.2 Non-Functional requirements

NFR No.	Non-Functional Requirement	Description
1	Usability	Usability requirements can consider language barriers and localization tasks. Usability can be assessed from the below functions. Efficiency of use. Low perceived workload. Easy and simple UI.
2	Security	Access permissions for the particular system information may only be changed by the system's data administrator.
3	Reliability	The database update process must roll back all related updates when any update fails.
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.
5	Availability	New module deployment mustn't impact front page, product pages, and check out pages availability and mustn't take longer than one hour. 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.
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

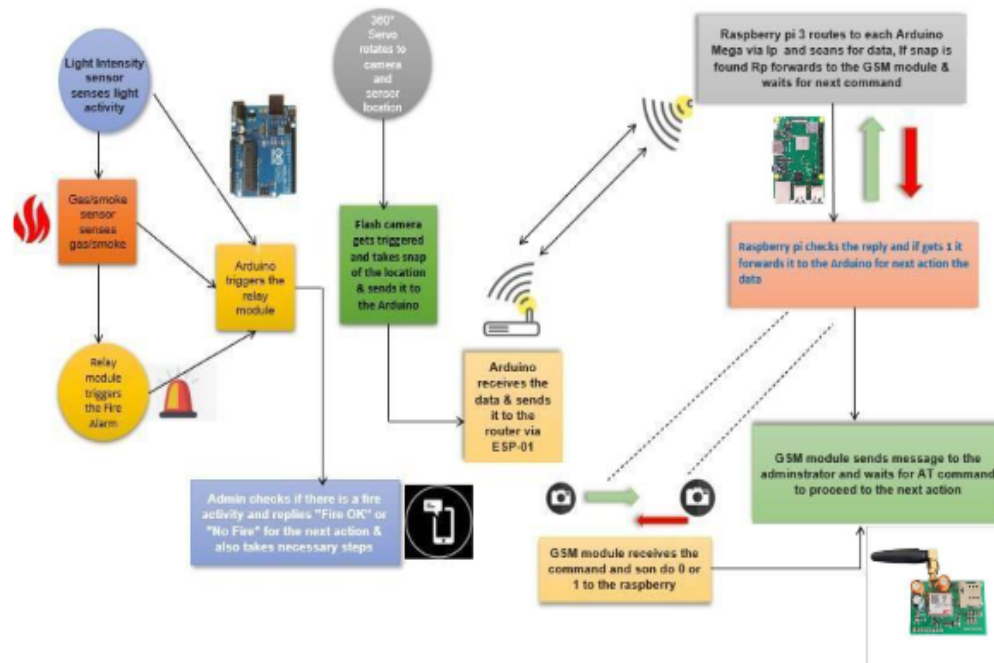
5.1 Data Flow Diagrams

FIRE MANAGEMENT SYSTEM
DFD



5.2 Solution & Technical Architecture

Solution Architecture Diagram:



5.3 User Stories

User Type	Functional requirement	User story number	User story/task	Acceptance criteria	Priority	Release
Customer	Registration	USN-1	As a user , I can register into the application	I can access my account and dashboard	High	Sprint 1
		USN-2	As a user , I can get the confirmation mail.	I can receive the confirmation email & click confirm	High	Sprint 1
	Dashboard	USN-3	As a user , I can register through the internet	I can access my account and dashboard	Medium	Sprint 2
		USN-4	As a user , I can register through mail	I can receive the confirmation email & click confirm	Low	Sprint 1
	Login	USN-5	As a user , I can register into the application by entering my mail id and password .	I can login with my id and password	High	Sprint 1

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Sprint	Functional Requirement	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint 1	Simulation software	USN-1	Using WOKWI, connect temperature, flame, gas sensors to ARDUINO with python script	2	High	AMRITHA J K, ATHULYA A S, MOUNIKA R, YASSHIKAA J
Sprint 2	Cloud software	USN-2	Create device in the IBM Watson IoT platform, and link it to Node-Red	2	High	AMRITHA J K, ATHULYA A S, MOUNIKA R, YASSHIKAA J
Sprint 3	MIT app inventor	USN-3	Develop a mobile application using MIT App inventor	2	High	AMRITHA J K, ATHULYA A S, MOUNIKA R, YASSHIKAA J
Sprint 4	linking	USN-4	Link WOKWI, IBM Cloud and the developed App Application	2	High	AMRITHA J K, ATHULYA A S, MOUNIKA R, YASSHIKAA J
Sprint 4	Dashboard	USN-5	Design the modules and Test the Mobile Application	2	High	AMRITHA J K, ATHULYA A S, MOUNIKA R, YASSHIKAA J

6.2 Sprint Delivery Schedule

Sprint	Total Story	Duration	Sprint Start	Sprint End	Story	Sprint Release
--------	-------------	----------	--------------	------------	-------	----------------

	Points		Date	Date	Points Completed	Date
Sprint 1	20	6 Days	24 oct 2022	29 oct 2022	20	29 oct 2022
Sprint 2	20	6 Days	31 oct 2022	05 nov 2022	20	05 nov 2022
Sprint 3	20	6 Days	07 nov 2022	12 nov 2022	20	12 nov 2022
Sprint 4	20	6 Days	14 nov 2022	19 nov 2022	20	19 nov 2022

7. CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 Feature 1

The ESP32 is connected to the gas sensor and to the temperature sensor which is further connected to the LED and alarm system .Whenever an unusual change in the measurement is observed the sensor senses and indicates it to the user.The temperature data can be viewed by the user continuously via IBM cloud or any other sources like thingspeak(As used in the below code).

Code:

```
#include <WiFi.h>
```

```
#include <Wire.h>
```

```
#include <SPI.h>
```

```
#include "ThingSpeak.h" #include  
<WiFiClient.h>
```

```
unsigned long myChannelNumber = 2; const char * myWriteAPIKey  
= "25V40ZAPI6KIZFGY";
```

```
int LED_PIN = 32; // the current reading from the input pin
```

```
int BUZZER_PIN= 12; const int mq2 = 4;
```

```
int
```

```
value = 0;
```

```
//Flame int flame_sensor_pin = 10 ;// initializing pin 10 as the sensor digital output pin int  
flame_pin = HIGH ; // current state of sensor
```

```
char ssid[] = "a"; char pass[] = "n";
```

```
WiFiClient client;
```

```
#define PIN_LM35 39
```

```

#define ADC_VREF_mV 3300.0
#define ADC_RESOLUTION 4096.0
#define RELAY_PIN 17
#define RELAY_PIN1 27

void setup()
{
  Serial.begin(115200);
  pinMode(RELAY_PIN, OUTPUT);  pinMode(RELAY_PIN1, OUTPUT);
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, pass);  int wifi_ctr
= 0;
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(1000);
    Serial.print(".");
  }
  Serial.println("WiFi connected");
  ThingSpeak.begin(client);  pinMode(LED_PIN, OUTPUT);  pinMode(mq2, INPUT);
  pinMode ( flame_sensor_pin , INPUT ); // declaring sensor pin as input pin for Arduino
  pinMode(BUZZER_PIN, OUTPUT);
}
void temperature()
{
  int adcVal = analogRead(PIN_LM35);  float milliVolt = adcVal *
(ADC_VREF_mV / ADC_RESOLUTION);  float tempC = milliVolt /
10;  Serial.print("Temperature: ");
  Serial.print(tempC);
  Serial.print("°C");  if(tempC
> 60)
  {
    Serial.println("Alert");
    digitalWrite(BUZZER_PIN, HIGH); // turn on
  } else
  {
    digitalWrite(BUZZER_PIN, LOW); // turn on
  }
  int x = ThingSpeak.writeField(myChannelNumber,1, tempC, myWriteAPIKey);  }
void GasSensors()
{
  //mq2

  int gassensorAnalogmq2 = analogRead(mq2);

```

```

Serial.print("mq2 Gas Sensor: ");
Serial.print(gassensorAnalogmq2);
Serial.print("\t");
Serial.print("\t");
Serial.print("\t");

if (gassensorAnalogmq2 > 1500)
{
    Serial.println("mq2Gas");    Serial.println("Alert");    digitalWrite(RELAY_PIN1, HIGH); //
turn on fan 10 seconds    delay(100);
} else
{
    Serial.println("No mq2Gas");    digitalWrite(RELAY_PIN1,
LOW); // turn off fan 10 seconds    delay(100);

}

    int  a  =  ThingSpeak.writeField(myChannelNumber,4,  gassensorAnalogmq2,
myWriteAPIKey);

}

void flamesensor()
{  flame_pin = digitalRead ( flame_sensor_pin ) ; // reading from the sensor if
(flame_pin == LOW ) // applying condition
{
    Serial.println ( " ALERT: FLAME IS DETECTED" ) ;    digitalWrite (BUZZER_PIN, HIGH ) ;// if
state is high, then turn high the BUZZER } else
{
    Serial.println ( " NO FLAME DETECTED " ) ;
    digitalWrite (BUZZER_PIN , LOW ) ; // otherwise turn it low
}  int value = digitalRead(flame_sensor_pin); // read the analog value from sensor

    if (value ==LOW) {    Serial.print("FLAME");
digitalWrite(RELAY_PIN, HIGH);
    } else {
        Serial.print("NO FLAME");    digitalWrite(RELAY_PIN, LOW);
    }

} void loop() {
temperature();  GasSensors();  flamesensor();
}

```

DIAGRAM.JSON:

```

{
  "version": 1,
  "author": "พิทักษ์ สถิตวรรธนะ",
  "editor": "wokwi",
  "parts": [
    { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": -96.39, "left": -7.47, "attrs": {} },
    {
      "type": "wokwi-ntc-temperature-sensor",
      "id": "ntc1",
      "top": -105.69,
      "left": 146.71,
      "rotate": 90,
      "attrs": {}
    },
    {
      "type": "wokwi-led",
      "id": "led1",
      "top": -30.93,
      "left": -84.06,
      "attrs": { "color": "red" }
    },
    {
      "type": "wokwi-resistor",
      "id": "r1",
      "top": 101.21,
      "left": -121.88,
      "attrs": { "value": "5600" }
    }
  ],
  "connections": [
    [ "esp:TX0", "$serialMonitor:RX", "", [] ],
    [ "esp:RX0", "$serialMonitor:TX", "", [] ],
    [ "r1:1", "led1:C", "green", [ "h-7.64", "v2.81" ] ],
    [ "led1:A", "esp:D14", "green", [ "v0" ] ],
    [ "r1:2", "esp:GND.2", "green", [ "h0" ] ],
    [ "ntc1:GND", "esp:GND.2", "black", [ "v0" ] ],
    [ "ntc1:VCC", "esp:VIN", "red", [ "v0" ] ],
    [ "esp:D32", "ntc1:OUT", "green", [ "h-33.04", "v-80.52", "h165.33", "v101.33" ] ]
  ]
}

```

```
}
```

7.2 Feature 2

The user can login into the application entering his details and view the respective temperature and gas sensor sensed data continuously. The user can even remain logged in the application.

code:

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Welcome To Login Form</title>

<!-- Complete css for whole page. -->
<style type="text/css">
/* body css for whole page */
body
{
margin:0px;
background-image:
url("https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.projects.ed.ac.uk%2Fproj
ect%2Fcsg013&psig=AOvVaw2Zlud0tkiB8qE7PAATcOUg&ust=1668527229576000&source
=images&cd=vfe&ved=0CBAQjRxqFwoTCPDglLmCrvsCFQAAAAAdAAAAABAE");
background-size: cover;
color:#f9fcf5;
font-family:Arial, Helvetica, sans-serif;
}
#main{width:600px; height:260px; margin-left:auto; margin-right:auto; border-radius:5px;
padding-left:10px; margin-top:100px;
border-top:3px double #f1f1f1; border-bottom:3px double #f1f1f1;border-right:3px double
#f1f1f1;border-left:3px double #f1f1f1; padding-top:20px;
background: #fff;
}
#main table{font-family:"Comic Sans MS", cursive;}
/* css code for textbox */
#main .tb{
height: 28px;
width: 230px;
border: 1px solid #262b28;
color: #27a465;
font-weight: bold;
opacity: 0.9;
padding: 0 10px;
}
```

```
#main .tb:focus{height:28px; border:1px solid #27a465; outline:none; border-left:5px solid #f7f7f7;}
```

```
/* css code for button*/
```

```
#main .btn{width:60%; height:32px; outline:none; font-weight:bold; border:0px solid #27a465; text-shadow: 0px 0.5px 0.5px #fff; border-radius: 2px; font-weight: 600; color: white; letter-spacing: 1px; font-size:14px; background-color:black; -webkit-transition: 1s; -moz-transition: 1s; transition: 1s;}
```

```
#main .btn:hover{background-color:white; outline:none; border-radius: 2px; color:#f1f1f1; border:1px solid #f1f1f1;-webkit-transition: 1s; -moz-transition: 1s; transition: 1s; }
```

```
</style>
```

```
<!-- Css ending here. -->
```

```
<!-- Complete javascript for login. -->
```

```
<!-- Add url of javascript -->
```

```
<script type="text/javascript" src="http://code.jquery.com/jquery-1.6.min.js"></script>
```

```
<!-- Java Script -->
```

```
<script>
```

```
function login()
```

```
{
```

```
var uname = document.getElementById("email").value;
```

```
var pwd = document.getElementById("pwd1").value;
```

```
var filter = /^[a-zA-Z0-9_\. \- ]+\.@([a-zA-Z0-9\-.]+\.)+([a-zA-Z0-9]{2,4})+$/;
```

```
if(uname=="")
```

```
{
```

```
alert("please enter user name.");
```

```
}
```

```
else if(pwd=="")
```

```
{
```

```
    alert("enter the password");
```

```
}
```

```
else if(pwd=="Athulya" && uname=="Athulya" )
```

```
{
```

```
alert('Login Success...Redirecting to Dashboard');
```

```
    //Redirecting to other page or webste code or you can set your own html page.
```

```
    window.location = "https://node-red-jleja-2022-11-04.eu-gb.mybluemix.net/ui/";
```

```
}
```

```
else
```

```
{
```

```
    alert("Invalid Login Credentials");
```

```
}
```

```
}
```



```

function clearFunc()
{
document.getElementById("email").value="";
document.getElementById("pwd1").value="";
}
    </script>
<!-- Javascript ending here.. -->
</head>

<body>
<!-- Main div code -->
<div id="main">
<div class="h-tag">
<h2><center style="color: black;">Login Form</center></h2>
</div>
<!-- Login box -->
<div class="login">
<table cellspacing="2" align="center" cellpadding="8" border="0">
<tr>
<td style="color: black;">User Name :</td>
<td><input type="text" placeholder="Enter User Name" id="email" class="tb" /></td>
</tr>
<tr>
<td style="color: black;">Password :</td>
<td><input type="password" placeholder="Enter Password" id="pwd1" class="tb" /></td>
</tr>
<tr>
<td></td>
<td><input type="submit" value="Login" class="btn" onClick="login()" /></td>
</tr>
</table>
</div>
    <!-- login box div ending here.. -->
</div>
<!-- Main div ending here... -->
<script>

(function(i,s,o,g,r,a,m){i['GoogleAnalyticsObject']=r;i[r]=i[r]||function(){
(i[r].q=i[r].q||[]).push(arguments)},i[r].l=1*new Date();a=s.createElement(o),
m=s.getElementsByTagName(o)[0];a.async=1;a.src=g;m.parentNode.insertBefore(a,m)
})(window,document,'script','https://www.google-analytics.com/analytics.js','ga');

```

```
ga('create', 'UA-88667581-1', 'auto');
ga('send', 'pageview');
</script>
</body>
</html>
```

8. TESTING

8.1 Test Cases

PROJECT DATA					01 Steps						
Test case ID	Feature Type	Component	Test Scenario	Pre Requisite	Steps to Execute	Test Data	Expected Result	Actual Result	Status	Comments	% for Automation(0/100)
Registration_TC_001	UI	HomePage	Verify user is able to use the homepage button		1.Click the link and download the application 2.Verify homepage test box is displayed	img-2431_registration_tc_001.jpg	Application should show homepage test box	Working as expected	Pass		
Registration_TC_002	UI	HomePage	Verify user is able to use the link and website		1.Click the link and download the application 2.Verify email test box is displayed	img-2432_registration_tc_002.jpg	Application should show email test box	Working as expected	Pass		
Registration_TC_003	UI	HomePage	Verify user is able to use the password method		1.Click the link and download the application 2.Verify password test box is displayed	img-2433_registration_tc_003.jpg	Application should show password test box	Working as expected	Pass		
Registration_TC_004	UI	HomePage	Verify user is able to use the confirm password section		1.Click the link and download the application 2.Verify confirm password test box is displayed	img-2434_registration_tc_004.jpg	Application should show confirm password test box	Working as expected	Pass		
Registration_TC_005	UI	HomePage	Verify user is able to use the submit button		1.Click the link and download the application 2.Verify submit test box is displayed	img-2435_registration_tc_005.jpg	Application should show submit test box	Working as expected	Pass		
Registration_TC_006	Functional	HomePage	Verify user is able to register to the application using valid credentials		1.Click the link and download the application 2.Enter valid username in username test box 3.Enter valid email in email test box 4.Enter valid password in password test box 5.Enter valid confirm password in confirm password test box 6.Enter submit button	Username: testuser@gmail.com Email: testuser@gmail.com Password: test@123 Confirm Password: test@123	Username: testuser@gmail.com Email: testuser@gmail.com Password: test@123 Confirm Password: test@123 User should register to our account page	Working as expected	Pass		
validation_TC_001	Functional	HomePage	Verify user is able to log this application with invalid email		1.Click the link and download the application 2.Enter invalid email in email test box 3.Click on submit button	Email: testuser@gmail.com	Application should show "invalid email"	Working as expected	Pass		

[illegible]

LoggingPage_TL02.2	UI	Logging Page	Verify user is able to see the connection status	1. Click the link and download the application 2. Enter valid username in username text box 3. Enter valid email in email text box 4. Enter valid password in password text box 5. Enter valid confirm password in confirm password text box 6. Enter valid button 7. A new page appears, verify application is running as expected	testfiretestcase@gmail.com testfiretestcase027@gmail.com testfiretestcase027@gmail.com	Application should show connection status	Working as expected			No
LoggingPage_TL02.3	UI	Logging Page	Verify user is able to see alarm on, alarm off connection connection off button	1. Click the link and download the application 2. Enter valid username in username text box 3. Enter valid email in email text box 4. Enter valid password in password text box 5. Enter valid confirm password in confirm password text box 6. Enter valid button 7. A new page appears, verify application is running as expected	testfiretestcase@gmail.com testfiretestcase027@gmail.com testfiretestcase027@gmail.com	Application should show connection status on, alarm on, alarm off button	Working as expected	Pass		No
LoggingPage_TL02.4	Functional	Logging Page	Verify user is able to turn the alarm on whenever alarm on button is clicked	1. Click the link and download the application 2. Enter valid username in username text box 3. Enter valid email in email text box 4. Enter valid password in password text box 5. Enter valid confirm password in confirm password text box 6. Enter valid button 7. A new page appears, verify application is running as expected	testfiretestcase@gmail.com testfiretestcase027@gmail.com testfiretestcase027@gmail.com	Application should turn on alarm	Working as expected	Pass		No

LoggingPage_TL02.5	Functional	Logging Page	Verify user is able to turn the alarm off whenever alarm off button is clicked	1. Click the link and download the application 2. Enter valid username in username text box 3. Enter valid email in email text box 4. Enter valid password in password text box 5. Enter valid confirm password in confirm password text box 6. Enter valid button 7. A new page appears, verify application is running as expected	testfiretestcase@gmail.com testfiretestcase027@gmail.com testfiretestcase027@gmail.com	Application should turn off alarm	Working as expected	Pass		No
LoggingPage_TL02.6	Functional	Logging Page	Verify user is able to turn the connection on whenever connection on button is clicked	1. Click the link and download the application 2. Enter valid username in username text box 3. Enter valid email in email text box 4. Enter valid password in password text box 5. Enter valid confirm password in confirm password text box 6. Enter valid button 7. A new page appears, verify application is running as expected	testfiretestcase@gmail.com testfiretestcase027@gmail.com testfiretestcase027@gmail.com	Application should turn on connection	Working as expected	Pass		No
LoggingPage_TL02.7	Functional	Logging Page	Verify user is able to turn the connection off whenever connection off button is clicked	1. Click the link and download the application 2. Enter valid username in username text box 3. Enter valid email in email text box 4. Enter valid password in password text box 5. Enter valid confirm password in confirm password text box 6. Enter valid button 7. A new page appears, verify application is running as expected	testfiretestcase@gmail.com testfiretestcase027@gmail.com testfiretestcase027@gmail.com	Application should turn off connection	Working as expected	Pass		No

8.2 User Acceptance Testing

8.2.1 Performance Testcases

- To verify if the user can log into the application when correct credentials are used.
- To verify if the "INVALID DATA" message pops up when incorrect credentials are used .
- To verify if the user could access the data in the application .

9. RESULTS

9.1 Performance Metrics

NPT - Risk Assessment							
S.No	Project Name	Scope/Feature	Functional Changes	Hardware Changes	Software Changes	Risk Score	
1	Industry Based Intelligent Fire Management System	Alarm	Low	No Changes	Low	No Changes	DANGER
		Displaying the values			Connection failure		
		Registration			Crashing of server		
		Alarm	Overloading of data				
		Registration					
NPT - Detailed Test Plan							
S.No	Project Overview	NPT Test approach	Approvals/Sign-off				
1	Industry Based Intelligent Fire Management System	Stress testing	By there may be crashing of server stress testing is used Overloading of data is done				
		Load testing					
End Of Test Report							
S.No	Project Overview	NPT Test approach	NPT - Type	Test Outcome	SD/UC/QR/Decision	Identified Defects (Detected/Closed/Open)	Approvals/Sign-off
1	Industry Based Intelligent Fire Management System	Stress testing	Registration - Mid	3 pages requested per second and 3 pages per loaded	60 Decision	No defects	Good
		Displaying the values	- Mid	Due to connection failure, there can be delay in displaying the values	60 Decision	Connection failure due to over load only less number of times	Failure of connection sometimes
		Load testing	Alarm - Mid	Due to overloading, there can be false alarm	60 Decision	False alarm	Chaos environment

10. ADVANTAGES & DISADVANTAGES

10.1 Advantages:

- The main advantage of installing fire alarms is the early warning benefit.

- The fire alarms can be installed just about anywhere in a commercial building and best of all the fire safety measure is highly cost effective for smoke and fire protection.

10.2 Disadvantages:

- Cost, not as competitively priced for smaller applications.
- This panel is computer like and at times there may be issues caused by the firmware (panel software).
- Loose connection results in the fluctuation of the detector loop current. Improper base plug-in connection.

11. CONCLUSION

Fire alarms are a prime necessity in modern buildings and architecture, especially in banks, data centres and gas stations. They detect the fire in the ambience at a very early stage by sensing smoke or flash and heat and raise an alarm which warns people about the fire and furnishes sufficient time to take preventive measures. It not only prevents big losses caused by deadly fire but sometimes proves to be a lifesaver. A fire alarm is a device that detects the presence of fire and atmospheric changes relating to smoke. The fire alarm operates to alert people to evacuate a location in which a fire or smoke accumulation is present. When functioning properly, a fire alarm will sound to notify people on and immediate fire emergency. The distinct sound exists to allow the notification to be heard. The fire alarm constructed by this project is reliable at low-cost.

12. FUTURE SCOPE

The fire safety systems market was valued at USD 10.89 billion in 2020 and is expected to reach USD 16.76 billion by 2026, at a CAGR of 7.5% over the forecast period 2021 - 2026. Knowing all, future fire alarm systems will be software-based, where one needs to find a way to train one or two programmers to avoid being dependent on the equipment supplier. This can also help the supplier and should lead to better pricing. Also with the improvement of the sensors' capabilities and communication channel technology, IoT devices present in industries and residential spaces have boosted the adoption of new-tech fire safety solutions. The scope of the fire safety systems market includes the type of safety system with fire detectors and suppressors, such as gas, foam, and detectors. The increasing focus of the government bodies on implementing fire safety equipment across various industries, such as chemical and petrochemical, oil and gas, pharmaceutical, aerospace, and defense, has led to the growth of the market studied.

13. APPENDIX

Source Code :

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Welcome To Login Form</title>

<!-- Complete css for whole page. -->
<style type="text/css">
/* body css for whole page */
body
{
margin:0px;
background-image:
url("https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.projects.ed.ac.uk%2Fproject%2Fcsg013&psig=AOvVaw2Zlud0tkiB8qE7PAATcOUg&ust=1668527229576000&source=images&cd=vfe&ved=0CBAQjRxqFwoTCPDglLmCrvsCFQAAAAAdAAAAABAE");
    background-size: cover;
color:#f9fcf5;
font-family:Arial, Helvetica, sans-serif;
}
#main{width:600px; height:260px; margin-left:auto; margin-right:auto; border-radius:5px;
padding-left:10px; margin-top:100px;
border-top:3px double #f1f1f1; border-bottom:3px double #f1f1f1;border-right:3px double
#f1f1f1;border-left:3px double #f1f1f1; padding-top:20px;
background: #fff;
}
#main table{font-family:"Comic Sans MS", cursive;}
/* css code for textbox */
#main .tb{
    height: 28px;
    width: 230px;
    border: 1px solid #262b28;
    color: #27a465;
    font-weight: bold;
    opacity: 0.9;
    padding: 0 10px;
}
#main .tb:focus{height:28px; border:1px solid #27a465; outline:none; border-left:5px solid
```

```
#f7f7f7;}
```

```
/* css code for button*/
```

```
#main .btn{width:60%; height:32px; outline:none; font-weight:bold; border:0px solid #27a465;  
text-shadow: 0px 0.5px 0.5px #fff;
```

```
border-radius: 2px; font-weight: 600; color: white; letter-spacing: 1px; font-size:14px;  
background-color:black; -webkit-transition: 1s; -moz-transition: 1s; transition: 1s;}
```

```
#main .btn:hover{background-color:white; outline:none; border-radius: 2px; color:#f1f1f1;  
border:1px solid #f1f1f1;-webkit-transition: 1s; -moz-transition: 1s; transition: 1s; }
```

```
</style>
```

```
<!-- Css ending here. -->
```

```
<!-- Complete javascript for login. -->
```

```
<!-- Add url of javascript -->
```

```
<script type="text/javascript" src="http://code.jquery.com/jquery-1.6.min.js"></script>
```

```
<!-- Java Script -->
```

```
<script>
```

```
function login()
```

```
{
```

```
var uname = document.getElementById("email").value;
```

```
var pwd = document.getElementById("pwd1").value;
```

```
var filter = /^[a-zA-Z0-9_\.\\-]+\@((([a-zA-Z0-9\\-])+\.)+([a-zA-Z0-9]{2,4})+$)/;
```

```
if(uname=="")
```

```
{
```

```
alert("please enter user name.");
```

```
}
```

```
else if(pwd=="")
```

```
{
```

```
    alert("enter the password");
```

```
}
```

```
else if(pwd=="Athulya" && uname=="Athulya" )
```

```
{
```

```
alert('Login Success...Redirecting to Dashboard');
```

```
//Redirecting to other page or webste code or you can set your own html page.
```

```
    window.location = "https://node-red-jleja-2022-11-04.eu-gb.mybluemix.net/ui/";
```

```
}
```

```
else
```

```
{
```

```

        alert("Invalid Login Credentials");
    }
}
function clearFunc()
{
    document.getElementById("email").value="";
    document.getElementById("pwd1").value="";
}
</script>
<!-- Javascript ending here.. -->
</head>

<body>
<!-- Main div code -->
<div id="main">
<div class="h-tag">
<h2><center style="color: black;">Login Form</center></h2>
</div>
<!-- Login box -->
<div class="login">
<table cellspacing="2" align="center" cellpadding="8" border="0">
<tr>
<td style="color: black;">User Name :</td>
<td><input type="text" placeholder="Enter User Name" id="email" class="tb" /></td>
</tr>
<tr>
<td style="color: black;">Password :</td>
<td><input type="password" placeholder="Enter Password" id="pwd1" class="tb" /></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><input type="submit" value="Login" class="btn" onClick="login()" /></td>
<td></td>
</tr>
</table>
</div>
<!-- login box div ending here.. -->
</div>
<!-- Main div ending here... -->
<script>

```

```

(function(i,s,o,g,r,a,m){i['GoogleAnalyticsObject']=r;i[r]=i[r]||function(){
(i[r].q=i[r].q||[]).push(arguments)},i[r].l=1*new Date();a=s.createElement(o),
m=s.getElementsByTagName(o)[0];a.async=1;a.src=g;m.parentNode.insertBefore(a,m)
})(window,document,'script','https://www.google-analytics.com/analytics.js','ga');

ga('create', 'UA-88667581-1', 'auto');
ga('send', 'pageview');
</script>
</body>
</html>

```

WOKWI 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 "t2tj60" //IBM ORGANITION ID
#define DEVICE_TYPE "esp32" //Device type mentioned in ibm watson IOT
Platform
#define DEVICE_ID "12345" //Device ID mentioned in ibm watson IOT
Platform
#define TOKEN "(29a2_Po)AmCk*DHkW" //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();
    }
}

void PublishData(float temp, float humid) {
    mqttconnect();//function call for connecting to ibm

    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");
    } 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()
{
    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="";
}
```

GitHub Link -

<https://github.com/IBM-EPBL/IBM-Project-39916-1660567655>

Project Demo Link -

https://drive.google.com/file/d/1rMKu6T5Ya_TxKfP0wkJ4of9A59wU3KRM/view?usp=sharing