PSNA COLLEGE OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

A Project Report On

INDUSTRY SPECIFIC INTELLIGENT FIRE MANAGEMENT SYSTEM

SUBMITTED BY

TEAM ID: PNT2022TMID05047

TEAM MEMBERS:

RAMPRASANTH T - 921319104158
SARAVANA BALAJI N - 921319104169
SABARI KISHORE R K - 921319104161
PALANISAMY S - 911319104141

TABLE OF CONTENTS

1. INTRODUCTION:

1.PROJECT OVERVIEW

Nowadays industry work is hard and produces more accidents. IoT technology uses automation functions to control accidents and disasters. The industry is using smart fire management systems. This smart fire management system uses a GAS Sensor, Flame Sensor, and Temperature Sensor. The gas sensor is used to detect any gas leakage and unwanted gasses in closed areas. If gas is detected in the surroundings sensors are automatically activated. A flame sensor is used to capture the shape of the flame RGB color model to identify the fire. Temperature Sensor, to check the amount of heat that is present in the surroundings. These sensors detect the fire, and when it is identified, then suddenly it forwards the alarm to alert the workers. When the alarm sound is received by the protocol, it releases all the doors in the industry and also alerts those members to get out of work from the industries. The sprinklers are activated and the water spears all the places of fire. This causes the workers to not panic when a flame is caught in the industries. It is very useful for workers and prevents the industries with a short period. When these sensors are not present in the industries, it is very hazardous to all workers and sometimes it creates severe injuries and even death. Emergency alerts are notified to the authorities and the Fire station. Through the smoke and gaseous substances, it can easily be detected by the sensor, due to this, the exhaust fan is turned on

2.PURPOSE

- To give a detect the status of the room with IoT devices
- To turn on sprinkler and exhaust fan when there is accident
- To detect the flow of water
- To send and store the temperature status in a cloud storage
- To give a easy management system on dashboard

2. LITERATURE SURVEY

PAPER TITLE	AUTHOR	OBJECTIVE/OUTCOME
A Survey of Fire Safety Measures for Industry Safety Using IOT	N. Savitha; S. Malathi 2019	In the proposed system the fire safety practices is going to implement for the fire crackers industry. In that the root cause for the fire is to be analyzed and prevent from the fire before it is triggered.
Design of Distributed Factory Fire Alarm Systems	Li Liu ;Yanke C I ; Haosong chen 2020	The Distributed plant fire alarm system can quickly detect the fire and issues an alarm to reduce the damage caused by the fire. The fire alarm system is a control system that integrates signal detection,transmission , processing and control .It mainly complete the basic function of Fire ,smoke and temperature module monitering fire.
A Microcontroller-based Fire Protection System for the Safety of Industries in Bangladesh	Md. Saiam Dept. of Electrical and Electronic Engineering, Khulna University of Engineering & Technology, Khulna, Bangladesh 2021	The affected area is also triggered by the fire extinguishing equipment. At the same time, it also notifies the manager and the nearby fire station via SMS. This paper presents a simulation and practical arrangement of the system to demonstrate

Safety Robot for Flammable Gas and Fire Detection using Multisensor Technology	Sandeep Prabhakaran; Mathan N	In case of fire accidents, the robot alerts the workstation and sends a mail to the firefighting department with the location read from the GPS module. As the robot works as an autonomous system, it does not need to be controlled remotely. Hence this robot is based on the line following mechanism, it is quite easy to install and can cover a large area efficiently.
Computer Vision Based Industrial and Forest Fire Detection Using Support Vector Machine (SVM)	Md. Abdur Rahman; Sayed Tanimun Hasan; Mohammed Abdul Kader 2022	The proposed strategy works on a very large dataset of fire videos that have been collected both in real-life situations and from the internet. This SVM pipeline model shows the maximum accuracy is 93.33%. The system can fulfill the precision and detect faster real-time fire detection.

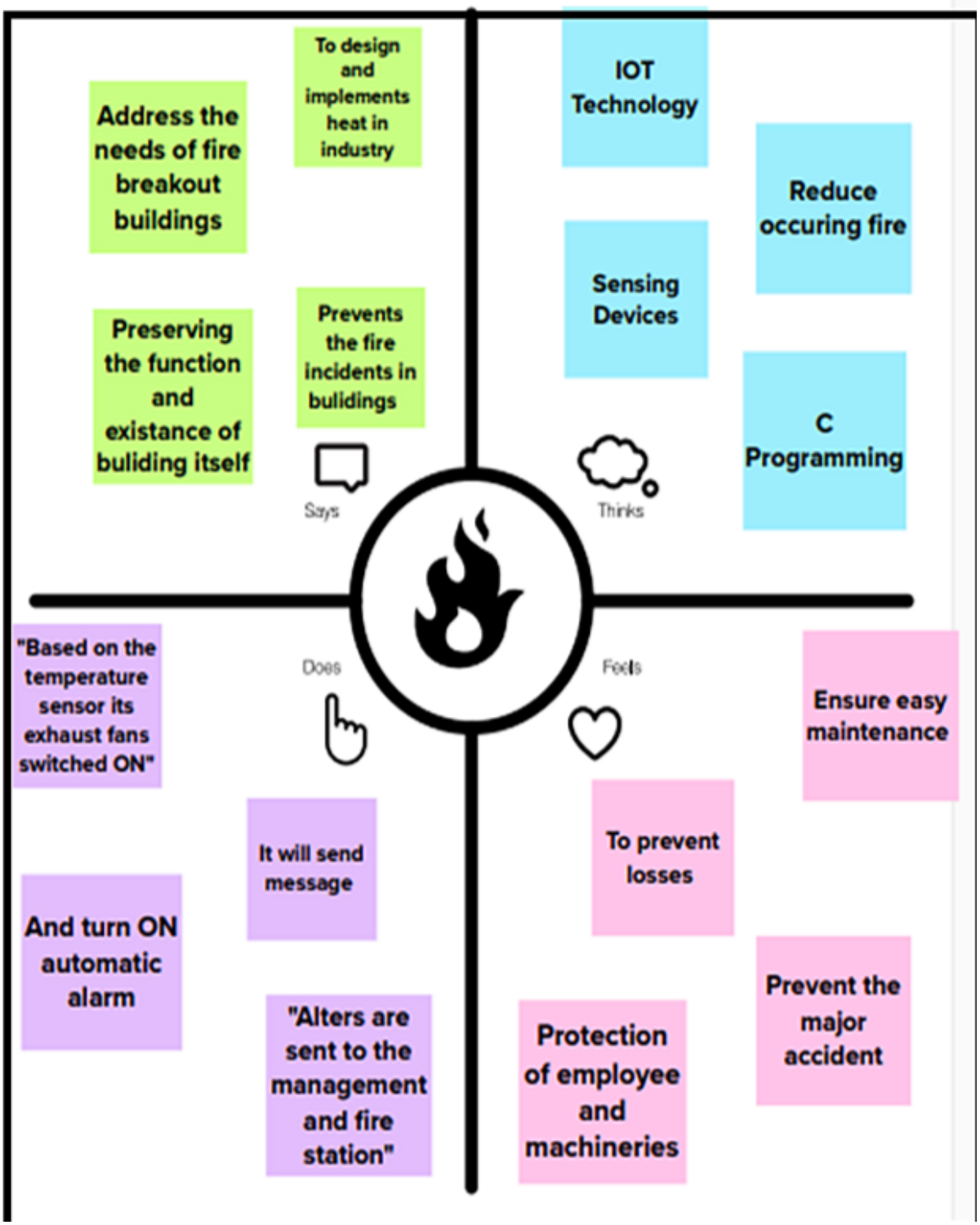
3.IDEATION & PROPOSED SOLUTION

Ideation is the process where you generate ideas and solutions through sessions such as empathy map canvas, ideation and brainstorming. Ideation is also the third stage in the Design Thinking process

1. Empathy Map Canvas

Empathy map consist of six fields,

- How customer Think and Feel?
- How customer See?
- How customer Say and Go?
- How customer Hear?
- What are the advantages ,(Gain) the user can get through this model
- What are the disadvantages, (pain) the user get through this model



2. Ideation & Brainstorming

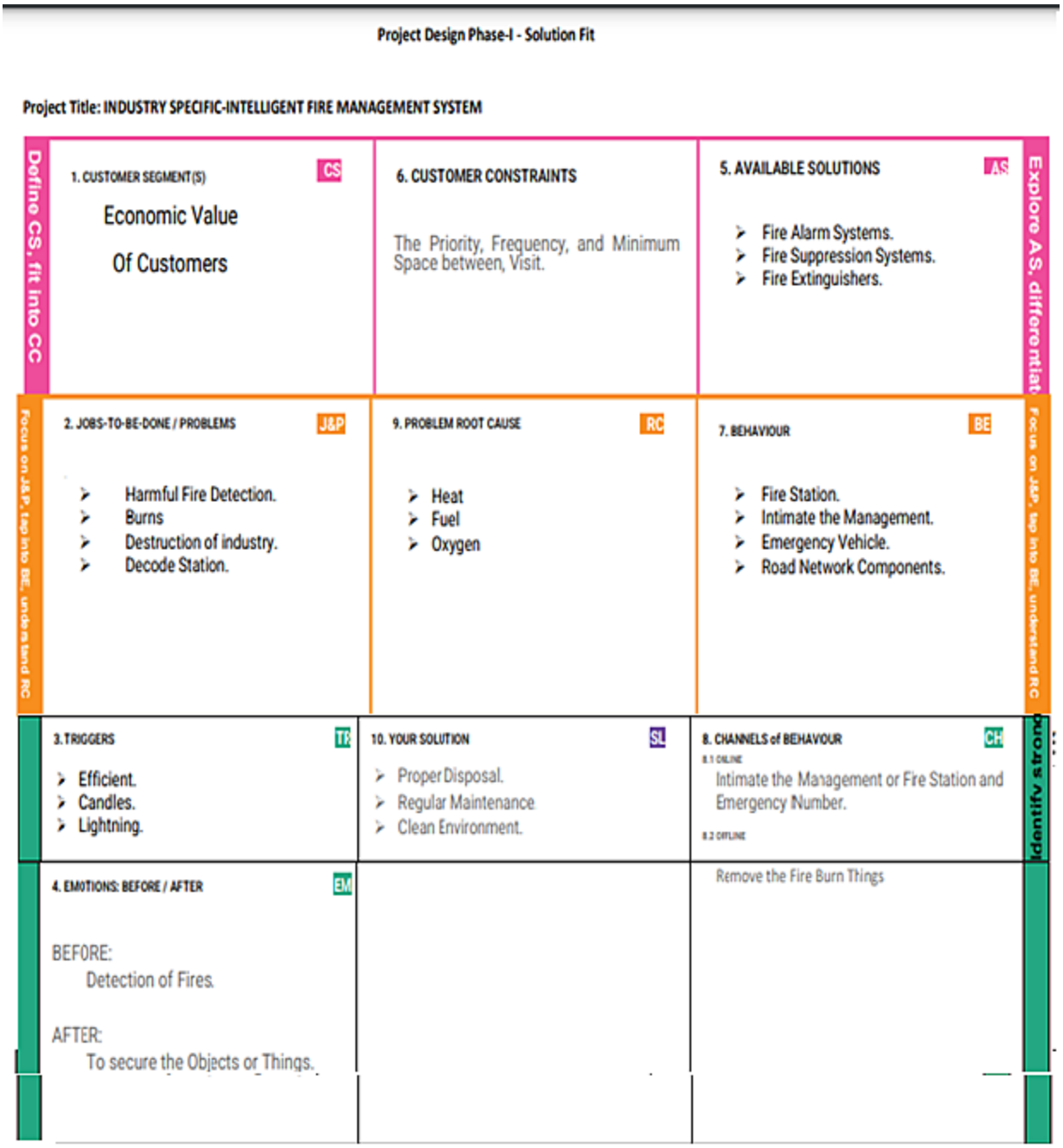
Brainstorming is a group problem-solving method that involves the spontaneous contribution of creative ideas and solutions. This technique requires intensive, freewheeling discussion in which every member of the group is encouraged to think aloud and suggest as many ideas as possible based on their diverse knowledge

3. Proposed Method:

The fire management system can be used to assessing and controlling the fire risks. Passive and active fire prevention.in the above literature they can using the micro controller / multi sensor to controlling the fire risks.In our method we are using the Sensor to predict the living brings are getting stuck inside the room/place. The information will be shared through ETSI (European Telecommunication Standards institute) to the related managements.

4.Proposed Solution

The Problem-Solution Fit simply means that **you have found a problem with your customer and that the solution you have realized for it actually solves the customer's problem**



4.REQUIREMENT ANALYSIS

1. Functional requirement

FR No	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)

FR-1	User Registration	Registration through Form Registration through Gmail Registration through mobile number
FR-2	User Confirmation	Confirmation via Email (OTP) Confirmation via OTP Through GSM
FR-3	Fire Detection Monitoring	In the industry we are monitor the Fire Detection using some sensors
FR-4	Intimate the Fire in the Industry	In case of any fire in industry we intimate the related Management through the Web Application

2. Non-Functional requirement

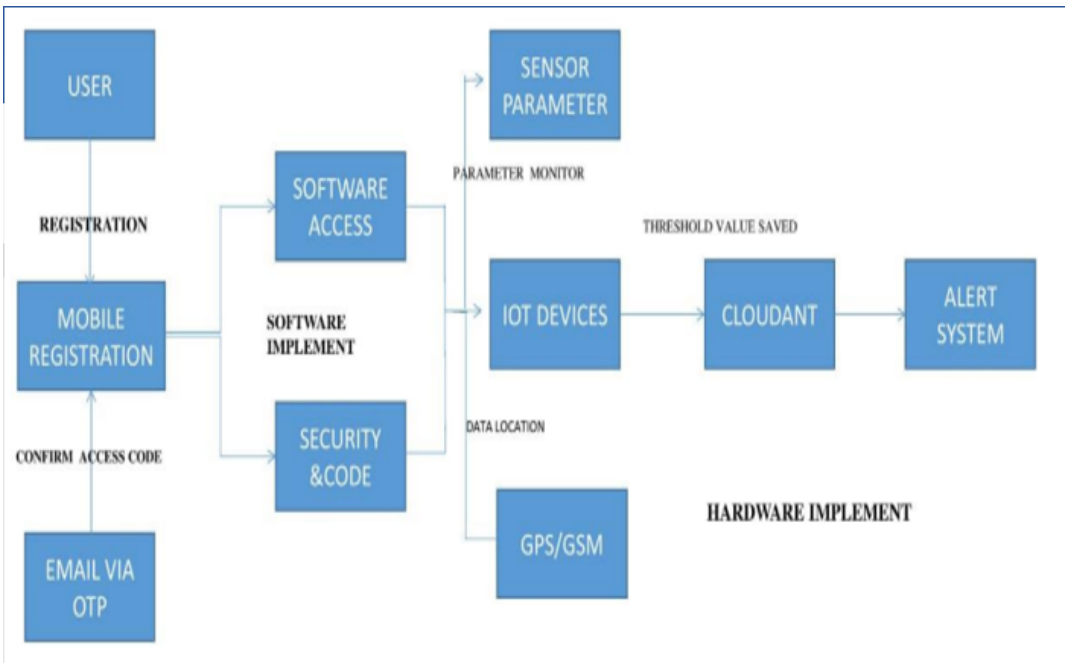
FR No	**Non-Functional Requirement**	**Description**
NFR-1	**Usability**	It is the simple and Economic Easy to use
NFR-2	**Security**	The Web application is highly secured.
NFR-3	**Reliability**	It has high Reliability. The application runs accurately.
NFR-4	**Performance**	Fire detection will intimate immediately through the web application and it also maintain the Records

NFR-5	**Availability**	In our project we are Monitoring the Industry in day and night (24/4). In case of Fire detect we intimate the management.
NFR-6	**Scalability**	We provide a high scalability our project/Application will use ‘n’ number of users

5. PROJECT DESIGN:

1. Data Flow Diagram:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



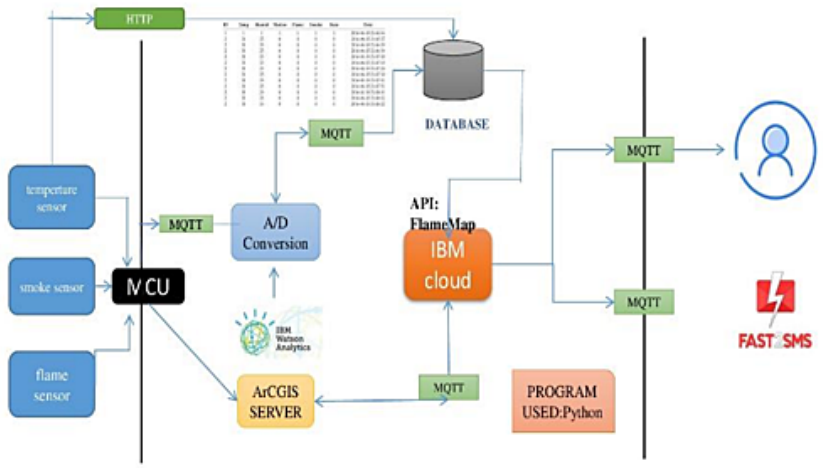
2. Solution and Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

Example: Order processing during pandemics for offline mode

Table-1 :

Components & Technologies:



S.No	Component	Description	Technology
1	User Interface	Web UI/ Mobile App	HTML, CSS, JavaScript
2	Application Logic-1	Logic for a process in the application	Python
3	Application Logic-2	Logic for a process in the application	IBM Watson STT service
4	Application Logic-3	Logic for a process in the application	IBM Watson Assistant
5	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc
6	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloud ant and etc.

7	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
8	External API-1	Purpose of External API used in the application	IBM Weather API, etc
9	External API-2	Purpose of External API used in the application	Mobile API, etc
10	Machine Learning Model	Purpose of Machine Learning Model	Object Recognition Model, etc
11	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	IBM Cloud and etc..

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1	**Open-Source Frameworks**	open-source frameworks used	Technology of Opensource framework
2	**Security Implementations**	The security / access controls implemented, use of firewalls etc.	Encryptions, IAM Controls, OWASP etc.

3	**Scalable Architecture**	The scalability of architecture (3 – tier, Microservices)	IOT AND MOBILE APPLICATION Technology used
4	**Availability**	distributed servers	IBM CLOUD AND WATSON Technology used
5	**Performance**	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	Fast GSM[SMS]

5.3 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)	Registration	USN-1	As a user, I can register for the application by entering my mobile number/email, to verify password, through GSM/MAIL and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive OTP from email/GSM once I registered the mobile application	I can receive confirmation OTP/email and click to confirm	High	Sprint-1
		USN-3	As a user, I can see the all required data about the fire system through the application	I can register & access the dashboard and analysis the details	Medium	Sprint-2
		USN-4	As a user, I can maintain and view the pervious data about the fire management system.		Medium	Sprint-1
	Login	USN-5	As a user, I can monitor the industry fire management system.		High	Sprint-1
	Dashboard					
Customer (Web user)						
Customer Care Executive						
Administrator						

6. PROJECT PLANNING & SCHEDULING

1.Sprint Planning & Estimation

STEP 1 Identify the fire

STEP 2 Prepare an abstract, problemstatement

STEP 3 List required objects needed

STEP 4 Create a code and run it

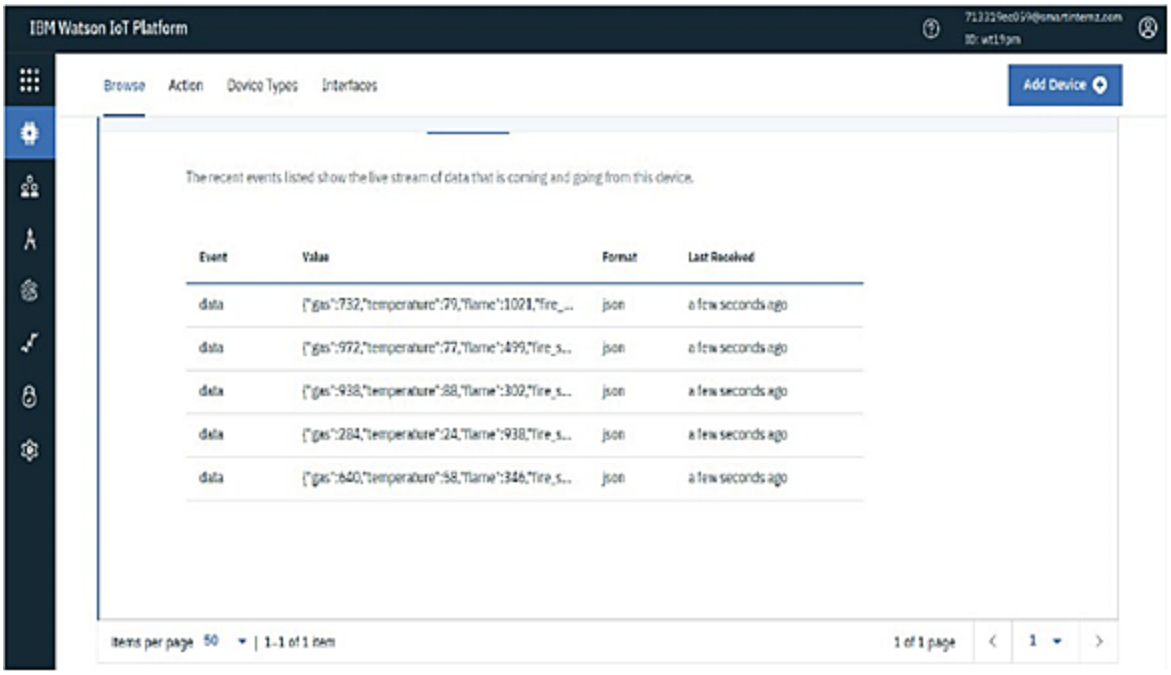
STEP 5 Make a prototype

STEP 6 Test with the created code and check the designed testprototype is working

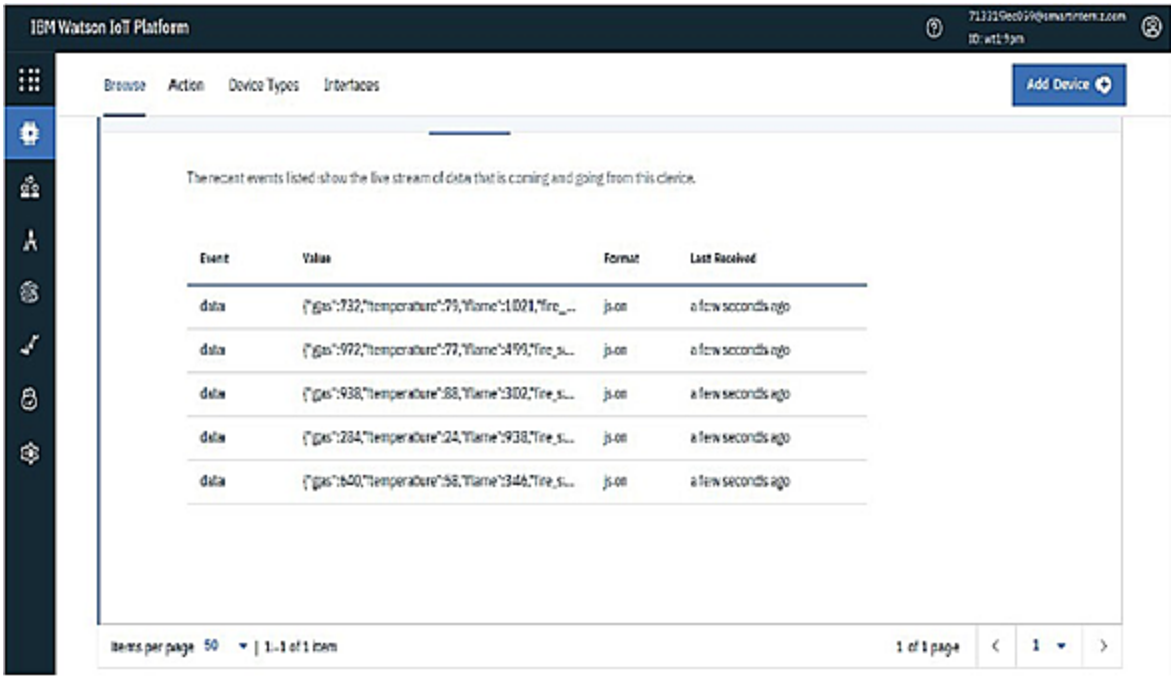
STEP 7 Solution for the problem is found

2. Reports from JIRA

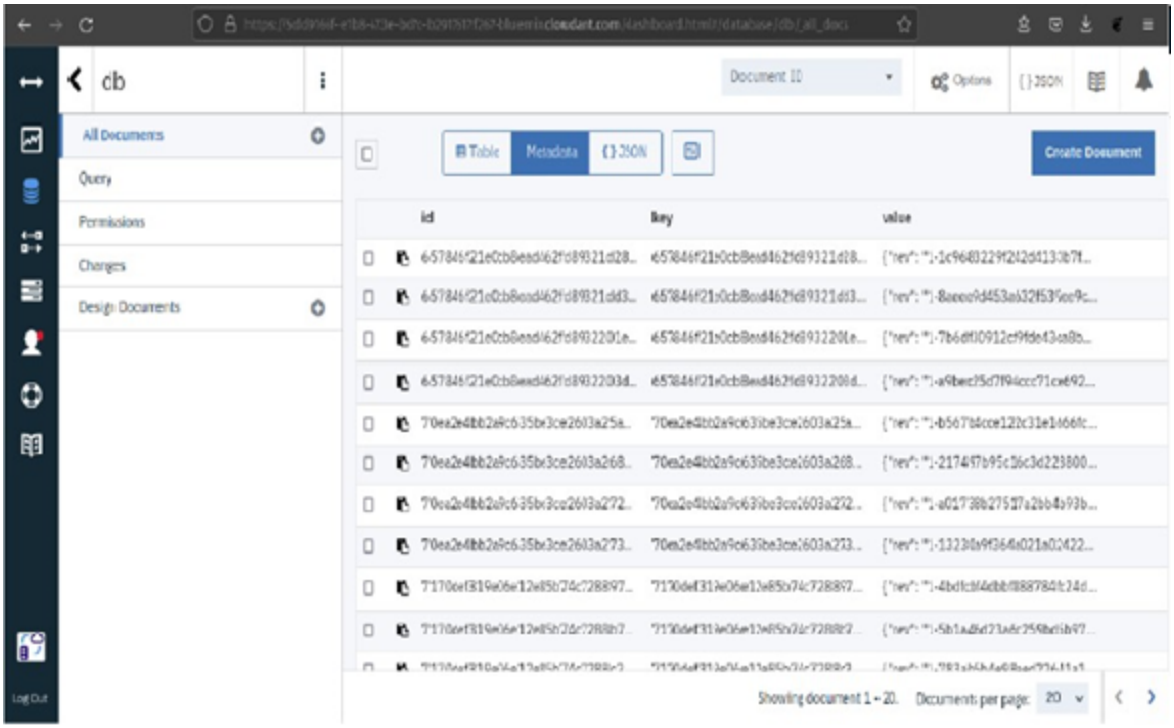
SPRINT 1



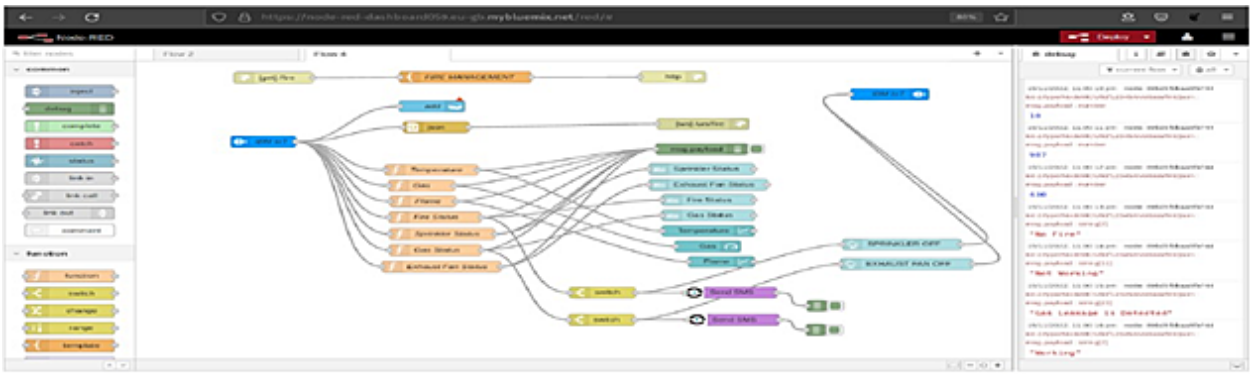
SPRINT 2



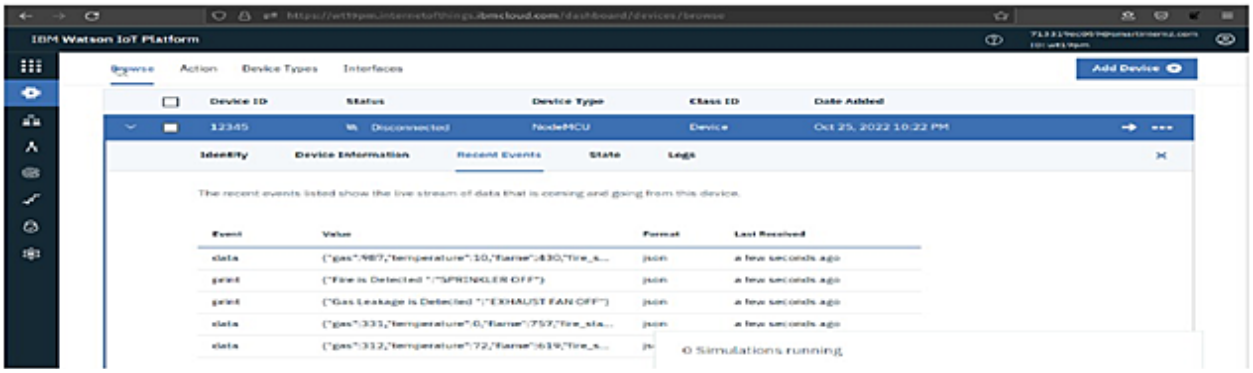
SPRINT 3



SPRINT 4



IBM WATSON IOT:



7.CODING & SOLUTIONING

Feature 1

1. IoT device
2. IBM Watson Platform
3. Node red
4. Cloudant DB
5. Web UI
6. MIT App Inventor
7. Python code

Feature 2

1. Login
2. Verification
3. Ticket Booking
4. Adding rating

8.TESTING AND RESULTS

1. Test Cases -Test case 1:

S.NO	INPUT	OUTPUT	RESULT
1	Gas:42 Temperature:59.30 Flame:267	Exhaust Fan: Not Working Sprinkler: Not Working Status Logged: Done	PASSED
2	Gas:612 Temperature:59.30 Flame:367	Exhaust Fan: Working Sprinkler: Not Working Status Logged: Done	PASSED
3	Gas:327 Temperature:59.30 Flame:841	Exhaust Fan: Working Sprinkler: Working Status Logged: Done	PASSED
4	Gas:13 Temperature:59.30 Flame:601	Exhaust Fan: Not Working Sprinkler: Working Status Logged: Done	PASSED
5	Gas: 123 Temperature:59.30 Flame:385	Exhaust Fan: Working Sprinkler: Not Working Status Logged: Done	PASSED

Test case 2:

				Date	[illegible]							
				Team ID	[illegible]							
				Project Name	Project - Industry specific intelligent fire management system							
				Maximum Marks	4 marks							
Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	TC for Automation(Y/N)	Executed By
Fire detection	Functional	IOT Sensors	Verify the fire detection and sensor working models.	Allocate fire extinguisher for gas requisite	1. Allocate the sensors in correct place. 2.Test check the Flame,gas,temperature in any sensors. 3.If any optimists sensors are detect the fire	1. Flame sensor,2.Gas sensor 3.Temperature sensor	Sensors are identifying the fire	Working as expected	Pass	detect the fire	No	Through IOT devices
Alarm,voice instruction	Functional	Python voice instruction,Alarm	Verify the working of Alarm and voice instruction	Nothing	1. Allocate the sensors in correct place. 2.Test check the Flame,gas,temperature in any sensors. 3.If any optimists sensors are detect the fire 4.Fire activated the fire alarm 5.Next give the voice instruction	[illegible]	Activate the alarm and voice instruction	Working as expected	Pass	Insert the voice instruction and alarm	No	Alarm and Python codes
Database	Functional	Iot and server	Verify to analyse the database storage	Data	1. Allocate the sensors in correct place. 2.Test check the Flame,gas,temperature in any sensors. 3.If any optimists sensors are detect the fire 4.Fire activated the fire alarm 5.Next give the voice instruction 6.Data are stored in databases	Database storage checking	Use to store the database	Working as expected	Pass	Analyse the database	No	SQL in database
Protocol removal	Functional	Emergency security removal	Verify to analyse protocol removal	Protocol	1. Allocate the sensors in correct place. 2.Test check the Flame,gas,temperature in any sensors. 3.If any optimists sensors are detect the fire 4.Fire activated the fire alarm 5.Next give the voice instruction 6.Data are stored in databases 7.Protocol removal	protocol checking and removal	Removal of protocol to reduce the damage control	Working as expected	Pass	Use of protocol removal	No	Protocol removal system

Test case 3:

				Date	[illegible]							
				Team ID	[illegible]							
				Project Name	Project - Industry specific intelligent fire management system							
				Maximum Marks	4 marks							
Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	TC for Automation(Y/N)	Executed By
Fire detection	Functional	IOT Sensors	Verify the fire detection and sensor working models.	Allocate fire extinguisher for gas requisite	1. Allocate the sensors in correct place. 2.Test check the Flame,gas,temperature in any sensors. 3.If any optimists sensors are detect the fire	1. Flame sensor,2.Gas sensor 3.Temperature sensor	Sensors are identifying the fire	Working as expected	Pass	detect the fire	No	Through IOT devices
Alarm,voice instruction	Functional	Python voice instruction,Alarm	Verify the working of Alarm and voice instruction	Nothing	1. Allocate the sensors in correct place. 2.Test check the Flame,gas,temperature in any sensors. 3.If any optimists sensors are detect the fire 4.Fire activated the fire alarm 5.Next give the voice instruction	[illegible]	Activate the alarm and voice instruction	Working as expected	Pass	Insert the voice instruction and alarm	No	Alarm and Python codes
Database	Functional	Iot and server	Verify to analyse the database storage	Data	1. Allocate the sensors in correct place. 2.Test check the Flame,gas,temperature in any sensors. 3.If any optimists sensors are detect the fire 4.Fire activated the fire alarm 5.Next give the voice instruction 6.Data are stored in databases	Database storage checking	Use to store the database	Working as expected	Pass	Analyse the database	No	SQL in database
Protocol removal	Functional	Emergency security removal	Verify to analyse protocol removal	Protocol	1. Allocate the sensors in correct place. 2.Test check the Flame,gas,temperature in any sensors. 3.If any optimists sensors are detect the fire 4.Fire activated the fire alarm 5.Next give the voice instruction 6.Data are stored in databases 7.Protocol removal	protocol checking and removal	Removal of protocol to reduce the damage control	Working as expected	Pass	Use of protocol removal	No	Protocol removal system

Test case4:

				Date	[illegible]							
				Team ID	[illegible]							
				Project Name	Project - Industry specific intelligent fire management system							
				Maximum Marks	4 marks							
Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	TC for Automation(Y/N)	Executed By
Fire detection	Functional	IOT Sensors	Verify the fire detection and sensor working models.	Allocate fire extinguisher for gas requisite	1. Allocate the sensors in correct place. 2.Test check the Flame,gas,temperature in any sensors. 3.If any optimists sensors are detect the fire	1. Flame sensor,2.Gas sensor 3.Temperature sensor	Sensors are identifying the fire	Working as expected	Pass	detect the fire	No	Through IOT devices
Alarm,voice instruction	Functional	Python voice instruction,Alarm	Verify the working of Alarm and voice instruction	Nothing	1. Allocate the sensors in correct place. 2.Test check the Flame,gas,temperature in any sensors. 3.If any optimists sensors are detect the fire 4.Fire activated the fire alarm 5.Next give the voice instruction	[illegible]	Activate the alarm and voice instruction	Working as expected	Pass	Insert the voice instruction and alarm	No	Alarm and Python codes
Database	Functional	Iot and server	Verify to analyse the database storage	Data	1. Allocate the sensors in correct place. 2.Test check the Flame,gas,temperature in any sensors. 3.If any optimists sensors are detect the fire 4.Fire activated the fire alarm 5.Next give the voice instruction 6.Data are stored in databases	Database storage checking	Use to store the database	Working as expected	Pass	Analyse the database	No	SQL in database
Protocol removal	Functional	Emergency security removal	Verify to analyse protocol removal	Protocol	1. Allocate the sensors in correct place. 2.Test check the Flame,gas,temperature in any sensors. 3.If any optimists sensors are detect the fire 4.Fire activated the fire alarm 5.Next give the voice instruction 6.Data are stored in databases 7.Protocol removal	protocol checking and removal	Removal of protocol to reduce the damage control	Working as expected	Pass	Use of protocol removal	No	Protocol removal system

9.ADVANTAGES

1. Easy to manage
2. Medium cost
3. Manage the large database

10. DISADVANTAGES

1. If anyone is sensor is a damagethe total experiment failure

11.CONCLUSION

Industry are work with people and automation machines. The IOT generates new features of industry. In our project we propose a fire detection algorithm which is free from sensors as the ordinary fire detection systems contain. The objective of this project was to create a system which would be able to detect fire as early as possible from a live video feed. System is expected to detect fire while it is still small and has not grown to mammoth proportions. Also, the hardware is minimal and has been already existent in places, thus saving capital. It also saves cost by getting rid of expensive temperature and heat sensors etc. Based on the results produced, the system has proven to be effective at detecting fire. This system is an amalgamation of various fire detection algorithms.The system can be made weather proof Smoke detection along with fire detection can be added as a feature System Optimization and Delay Reductioni.e. Lesser latency may be achieved. System can be used to detect forest fires and may be embedded on a drone orany other UAV for surveillance purposes of property. The system can have military applications. The system can be used for rescueoperations on land and in sea

12.FUTURE SCOPE

This application is ensured for safety for the passengers while they are traveling aloneas well as when they travelwith their familyor friends In future, this application may alsobe used by passengers who travel by bus. By further enhancement of the application the passengers can explore more features regarding their safety.

13.APPENDIX

1.Source Code :

```
<!DOCTYPE html>

<html>

<head>

 <title>Login</title>

 <meta charset="utf-8">

 <meta name="viewport" content="width=device-width, initial-scale=1,
 shrink-to-fit=no">

 <link href="style.css" rel="stylesheet" type="text/css">

 <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.
min.css">
<link href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-
awesome.min.css" rel="stylesheet">

 <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>

</head>

<body>

 <div class="container-fluid">

        <div class="row ">

                <!-- IMAGE CONTAINER BEGIN-->

                <div class="col-lg-6 col-md-6d-none d-md-block infinity-
image-container"></div>

                <!-- IMAGE CONTAINER END -->
```

```
<!-- FORM CONTAINER BEGIN-->

<div class="col-lg-6 col-md-6 infinity-form-container">

<!-- <h4 class = "test"
style="color:#fa6903;">FIREMANAGEMENT</h4> -->

<div class="col-lg-9 col-md-12 col-sm-9col-xs-12
infinity-form">

<!-- Company Logo -->

<div class="text-center mb-3 mt-5">

<h4 class = "test" style="color:#fa6903;">FIREMANAGGMENT</h4>


</div>

<div class="text-center mb-4">


<h4>Login into your account</h4>

</div>

<!-- Form -->

<form class="px-3">

<!-- Input Box -->

<div class="form-input">

<span><i class="fa fa-envelope-
o"></i></span>

<input
type="email" name=""placeholder="Email Address"
tabindex="10"required>
```

```
</div>
                                                    <div class="form-input">

                                                        <span><i class="fa fa-
                                                        lock"></i></span>

                                                        <input type="password"
                                                        name=""
                placeholder="Password" required>

                                                    </div>

                                                    <div class="row mb-3">

                                                    <!-- Remember Checkbox -->

                                  <div class="col-auto d-flex align-items-center">

                                    <div class="custom-control custom-checkbox">

                                      <input type="checkbox" class="custom-control-input"
                                      id="cb1">

                                      <label class="custom-control-label text-black"
                                      for="cb1" style =
                >Remember me</label>

                                    </div>

                                  </div>

                                        </div>

                                        <!-- Login Button -->

                                 <div class="mb-3">

                block">Login</button>
</div>
<button type="submit" class="btn btn
```

```
<div class="text-right ">

<a href="reset.html"class="forget-link">Forgotpassword?</a>

</div>
</div>
</div>
</form></div>
</div>
<!-- FORM CONTAINER END -->
</body>
</html>
```

```
<!DOCTYPE html>

<html>

<head>

 <title>Reset</title>

 <meta charset="utf-8">

 <meta name="viewport" content="width=device-width, initial-scale=1,
 shrink-to-fit=no">

 <link href="style.css" rel="stylesheet" type="text/css">

 <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.
min.css">

 <linkhref="https://stackpath.bootstrapcdn.com/font-
awesome/4.7.0/css/font-awesome.min.css" rel="stylesheet">

 <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>

</head>

<body>

 <div class="container-fluid">
```

```
<div class="row">

<!-- IMAGE CONTAINER BEGIN-->

<div class="col-lg-6 col-md-6d-none d-md-block infinity-
image-container"></div>

<!-- IMAGE CONTAINER END -->



<!-- FORM CONTAINER BEGIN-->

<div class="col-lg-6 col-md-6 infinity-form-container">

<div class="col-lg-8 col-md-12 col-sm-8col-xs-12
infinity-form">

<div class="text-center mb-3 mt-5">

<img src="ps.webp" width="150px">

</div>

<div class="reset-form d-block">

<form class="reset-password-formpx-3">

<h4 class="mb-3" style="color:#fa6903">Reset
Your
password</h4>
<p class="mb-3 text-black">

Please enter your email address and we will send you a password
</p>
<div class="form-input">

<span><i class="fa fa-

<input type="email" name=""placeholder="Email Address"
```

```
                    tabindex="10"required>

                                        </div>

                              <div class="mb-3">

          >Send Reset Link</button>
                                  </form></div>
     </div>

<button type="sUbmit" class="btn"
                                <div class="reset-confirmationd-none px-3">

                                     <div class="mb-4">

                                          <h4 class="mb-3">Linkwas sent</h4>

                                     <h6 class="text-white">Please, check your
                                     inbox for a

          password reset link.</h6>
          Now</button>
  </div>
  <a href="login.html">

     <button type="submit" class="btn">Login
     </a>

</div>

</div>
<div>

                         <!-- FORM CONTAINER END -->
                   </div>

            </div>
```

```
<script
 type="text/jav
 ascript">
 function
 PasswordReset
 () {
            $('form.reset-password-form').on('submit', function(e) {
     e.preventDefault();
     $('.reset-form')
     .removeClass('d-block')
     .addClass('d-none');
    $('.reset-confirmation').addClass('d-block');
            });
 }

 window.addEventListener('l
   oad', function() {
   PasswordReset();
 });
</script>

</body>
</html>
```

```
#include <PubSubClient.h>


#define ORG "wt19pm"

#define

DEVICE_TYPE

"NodeMCU"#define

DEVICE_ID"12345"

#define TOKEN "12345678"

char server[]= ORG

".messaging.internetofthings.ibmcloud.com";char

publishTopic[] = "iot-2/evt/data/fmt/json";

char authMethod[] = "use-token-auth";

char token[] = TOKEN;

char clientId[] = "d:" ORG ":" DEVICE_TYPE

":" DEVICE_ID;

WiFiClient wifiClient;

PubSubClient client(server, 1883, wifiClient);

float temperature = 0;

int gas = 0;

int flame = 0;

String flame_status = "";

String Gas_status = "";
```

```
String exhaust_fan_status = "";

String sprinkler_status = "";

void setup() {

Serial.begin(99900);

wifiConnect();

mqttConnect();

}

void loop() {

srand(time(0));

//initial variables and random generated data

temperature = random(-20,125);

gas = random(0,1000);

int flamereading = random(200,1024);

flame = map(flamereading,200,1024,0,2);

//set a flame status

switch (flame) {

case 0:

flame_status = "No Fire";

break;

case 1:

flame_status = "Fire is Detected";
```

```
break;

}

//send the sprinkler status

if(flame==1){

sprinkler_status = "Working";

}

else{

sprinkler_status = "Not Working";

}

//toggle the fan according to gas reading

if(gas > 100){

Gas_status = "Gas Leakage is Detected";

exhaust_fan_status = "Working";

}

else{

Gas_status = "No Gas Leakage is Detected";

exhaust_fan_status = "Not Working";

}

//json format for IBM Watson

String payload = "{";

payload+="\"gas\":";
```

```
payload+=gas;
payload+=",";
payload+="\"temperature\":";
payload+=(int)temperature;
payload+=",";
payload+="\"flame\":";
payload+=flamereading;
payload+=",";
payload+="\"fire_status\":\""+flame_status+"\",
";
payload+="\"sprinkler_status\":\""+sprinkler_sta
tus+"\",";
payload+="\"Gas_status\":\""+Gas_status+"\",";
payload+="\"exhaust_fan_status\":\""+exhaust_f
an_status+"\"}";
if(client.publish(publishTopic, (char*)
payload.c_str()))
{
Serial.println("Publish OK");
}
else{
```

```
Serial.println("Publish failed");
}
delay(1000);
if (!client.loop())
{
mqttConnect();
}
}
void wifiConnect()
{
Serial.print("Connecting to ");
Serial.print("Wifi");
WiFi.begin("Wokwi-GUEST", "", 6);
while (WiFi.status() != WL_CONNECTED)
{
delay(500);
Serial.print(".");
}
Serial.print("WiFi connected, IP address: ");
Serial.println(WiFi.localIP());
}
```

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


```
{
"version": 1,
"author": "PNT2022TMID51903",
"editor": "wokwi",
"parts": [ { "type": "wokwi-esp32-devkit-v1",
"id": "esp", "top": -110.32, "left": 3.84,
```

```
"attrs": {} } ],
"connections": [ [ "esp:TX0",
"$serialMonitor:RX", "", [] ], [ "esp:RX0",
"$serialMonitor:TX", "", [] ] ]
}
#include <time.h>
#include <WiFi.h>
#include <PubSubClient.h>
#define ORG "wt19pm"
#define DEVICE_TYPE "NodeMCU"
#define DEVICE_ID "12345"
#define TOKEN "12345678"
char server[] = ORG
".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/data/fmt/json";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE
":" DEVICE_ID;
WiFiClient wifiClient;
PubSubClient client(server, 1883, wifiClient);
```

```
float temperature = 0;

int gas = 0;

int flame = 0;

String flame_status = "";

String Gas_status = "";

String exhaust_fan_status = "";

String sprinkler_status = "";

void setup() {

Serial.begin(99900);

wifiConnect();

mqttConnect();

}

void loop() {

srand(time(0));

//initial variables and random generated data

temperature = random(-20,125);

gas = random(0,1000);

int flamereading = random(200,1024);

flame = map(flamereading,200,1024,0,2);

//set a flame status

switch (flame) {
```

```
case 0:
flame_status = "No Fire";
break;
case 1:
flame_status = "Fire is Detected";
break;
}
//send the sprinkler status
if(flame==1){
sprinkler_status = "Working";
}
else{
sprinkler_status = "Not Working";
}
//toggle the fan according to gas reading
if(gas > 100){
Gas_status = "Gas Leakage is Detected";
exhaust_fan_status = "Working";
}
else{
Gas_status = "No Gas Leakage is Detected";
```

```
exhaust_fan_status = "Not Working";

}

//json format for IBM Watson

String payload = "{";

payload+="\"gas\":";

payload+=gas;

payload+=",";

payload+="\"temperature\":";

payload+=(int)temperature;

payload+=",";

payload+="\"flame\":";

payload+=flamereading;

payload+=",";

payload+="\"fire_status\":\""+flame_status+"\",

";

payload+="\"sprinkler_status\":\""+sprinkler_sta

tus+"\",";

payload+="\"Gas_status\":\""+Gas_status+"\",";

payload+="\"exhaust_fan_status\":\""+exhaust_f

an_status+"\"}";

if(client.publish(publishTopic, (char*)
```

```
payload.c_str()))
{
Serial.println("Publish OK");
}
else{
Serial.println("Publish failed");
}
delay(1000);
if (!client.loop())
{
mqttConnect();
}
}
void wifiConnect()
{
Serial.print("Connecting to ");
Serial.print("Wifi");
WiFi.begin("Wokwi-GUEST", "", 6);
while (WiFi.status() != WL_CONNECTED)
{
delay(500);
```

```
Serial.print(".");
}
Serial.print("WiFi connected, IP address: ");
Serial.println(WiFi.localIP());
}
void mqttConnect()
{
if (!client.connected())
{
Serial.print("Reconnecting MQTT client to ");
Serial.println(server);
while (!client.connect(clientId, authMethod,
token))
{
Serial.print(".");
delay(500);
}
Serial.println();
}
}
include <time.h>
```

```
#include <WiFi.h>

#include <PubSubClient.h>

#define ORG "wt19pm"

#define DEVICE_TYPE "NodeMCU"

#define DEVICE_ID "12345"

#define TOKEN "12345678"

char server[] = ORG

".messaging.internetofthings.ibmcloud.com";

char publishTopic[] = "iot-2/evt/data/fmt/json";

char authMethod[] = "use-token-auth";

char token[] = TOKEN;

char clientId[] = "d:" ORG ":" DEVICE_TYPE

":" DEVICE_ID;

WiFiClient wifiClient;

PubSubClient client(server, 1883, wifiClient);

float temperature = 0;

int gas = 0;

int flame = 0;

String flame_status = "";

String Gas_status = "";

String exhaust_fan_status = "";
```

```
String sprinkler_status = "";

void setup() {

Serial.begin(99900);

wifiConnect();

mqttConnect();

}

void loop() {

srand(time(0));

//initial variables and random generated data

temperature = random(-20,125);

gas = random(0,1000);

int flamereading = random(200,1024);

flame = map(flamereading,200,1024,0,2);

//set a flame status

switch (flame) {

case 0:

flame_status = "No Fire";

break;

case 1:

flame_status = "Fire is Detected";

break;
```

```
}

//send the sprinkler status

if(flame==1){

sprinkler_status = "Working";

}

else{

sprinkler_status = "Not Working";

}

//toggle the fan according to gas reading

if(gas > 100){

Gas_status = "Gas Leakage is Detected";

exhaust_fan_status = "Working";

}

else{

Gas_status = "No Gas Leakage is Detected";

exhaust_fan_status = "Not Working";

}

//json format for IBM Watson

String payload = "{";

payload+="\"gas\":";

payload+=gas;
```

```
payload+=",";
payload+="\"temperature\":";
payload+=(int)temperature;
payload+=",";
payload+="\"flame\":";
payload+=flamereading;
payload+=",";
payload+="\"fire_status\":\""+flame_status+"\",
";
payload+="\"sprinkler_status\":\""+sprinkler_sta
tus+"\",";
payload+="\"Gas_status\":\""+Gas_status+"\",";
payload+="\"exhaust_fan_status\":\""+exhaust_f
an_status+"\"}";
if(client.publish(publishTopic, (char*)
payload.c_str()))
{
Serial.println("Publish OK");
}
else{
Serial.println("Publish failed");
```

```
}
delay(1000);
if (!client.loop())
{
mqttConnect();
}
}
void wifiConnect()
{
Serial.print("Connecting to ");
Serial.print("Wifi");
WiFi.begin("Wokwi-GUEST", "", 6);
while (WiFi.status() != WL_CONNECTED)


Serial.print("WiFi connected, IP address: ");
Serial.println(WiFi.localIP());
}
void mqttConnect()
{
if (!client.connected())
{
```

```
Serial.print("Reconnecting MQTT client to ");
Serial.println(server);
while (!client.connect(clientId, authMethod,
token))
{
Serial.print(".");
delay(500);
}
Serial.println();
}
}
```

2.GitHub

DEMO VIDEO LINK:

- **https://drive.google.com/file/d/1uWQbN458ML5SwHxOkCFmKzcG-KQXdg0R/view?usp=drivesdk**
- **https://drive.google.com/file/d/1uWX0-28_jn_Rv93HbDeWs_0R1QwX-e0d/view?usp=drivesdk**

GitHub link:

https://github.com/IBM-EPBL/IBM-Project-8846-1658934532