Project Report

Date	17 November 2022
Team ID	PNT2022TMID15947
Project Name	Industry – specific intelligent fire management system
Team Members	1) Balaji P (Team Leader) 2) Akilan A 3) Dinesh S 4) Arun R

INTRODUCTION:

Fire alarm systems are only effective if they can generate reliable and fast fire alerts with exact location of fire. There is a direct correlation between the amount of damage caused by fire and interventions time in various fire alarm systems. As the time of intervention decreases, the damage also decreases.

Hence the most important factor in a fire alarm system is the reaction or response time of fire alarm system, that is, the time between fire detection and extinguishing.

Project Overview:

The earliest recorded examples of fire protection can be traced back to the Roman Empire and the catastrophic fires that started in Rome. As a result, Emperor Neron has adopted regulations that required fireproof material for walls and buildings restoration to be used.

The second recorded case of adopting fire protection regulations occurred in the year 1666, after the Great fire of London, which destroyed more than 80% of the city.

The fire of London spurred interest in the development of the first equipment for fire suppression in the form of hand pumps and fire hydrant installation for water supply.

Purpose:

A fire management system provides a complete structure for fire safety management, combining all fire-related documentation into one accessible document.

The production of a fire management system includes a review of current processes and production of a bespoke fire safety management system for the building or site – including documentation outlining fire safety systems on-site, emergency procedures and identification of relevant responsibilities.

LITERATURE SURVEY:

This project's purpose is industrial and domestic safety, and the primary concern is to avoid fire hazards. As a solution, a smart fire and high-temperature detection system are designed using GSM technology.

A smoke sensor is used to detect the smoke from the fire and a temperature sensor is used to detect temperature increases inside the building.

In event of a fire, an alert message will be sent to the user via short message service (SMS) via the GSM module. Furthermore, when a fire is detected, a signal will be sent to the fire service department and the manager of the industry.

Developments in Fire Detection Technologies:

Author: *Andrew Kim & Zhigang Liu (2018)*

The progress in fire detection technologies has been substantial over the last decade due to advances in sensor, microelectronics, and information technologies, as well as a greater understanding of fire physics.

This paper provides a review of progress in fire detection technologies over the last decade, including various emerging sensor technologies (e.g., computer vision system, distributed fiber optic temperature sensor, and intelligent multiple sensors), signal processing and monitoring technology

Some problems and future research efforts related to current fire detection technologies are discussed.

Developed an Intelligent Fire Alarm System:

Author: *Hussam Elbehiery Hussam Elbehiery 2012*

A Fire alarm system that provides remote monitoring services can also be used to provide medical alert services. Here a person with health problems who lives alone carries a radio transmitter that can trigger the system in case they need assistance.

Signals received at the monitoring station are identified by type (fire, burglary, medical alert) so that a proper response can be made.

Fire Safety and Alert System Using Arduino Sensors with IoT Integration:

Author: S. Perilla (2015)

In this paper, Integrating IoT on a fire safety system greatly increases its effectiveness and efficiency. With the use of sensors, fire indications like increase of temperature, presence of flames, gases and smoke are detected effectively.

Building occupants and fire-fighting authorities are notified in real-time through distress sound and light alarms, and SMS messages sent by the modules integrated in this system. Critical situations are solved and addressed quickly over the traditional systems which requires large amount of time and effort.

Smart Fire Alarm System:

Author: Gaurav Pawar (2021)

The paper depicts the necessity and an efficient solution for fire safety. Internet of Things was the main concept used and the project mainly builds on the techniques which are already present and also it has overcome obstacles present in the previous systems. But still, there are a few tweaks and remodeling required to get the coefficient and working model.

The time taken for the process is to be reduced for practical use Directly call the emergency services and key contacts to minimize the time it takes for the fire brigade to attend the site.

TABULATION FOR LITERATURE SURVEY :

S.NO	Year	Researcher	Title	Technology	Remark
01	2021	Gaurav Pawar	Smart Fire Alarm System	IOT	Highest Accuracy of about 90% has attained
02	2015	S.Perilla	Fire Safety and Alert System Using Arduino Sensors with IoT Integration	IOT	it was found that training accuracy was 90.82%, and testing accuracy was 83.63%. By the using of Arduino Sensors
03	2018	Mr.C.Santhana Krishnan	A Survey on the implementation of Fire detection System Based on ZigBee Wi-Fi Networks	ZigBee Module	On the basis of using the ZigBee Module ,which greatly improves and result shown 90% of this module
04	2018	Malathi Subramanian	A Survey on Fire Safety Measures for Industry Safety Using IoT	IOT	achieve accuracy of 96%, this was done in this project.
05	2019	Mr. Aneesh & Mr. Shafeek Basheer	A Smart Real-Time Fire and Smoke Detection System	Tensilica ESP8266 processor	Results have shown 92.5%, By using of Tensilica ESP8266 processor

Problem statement :

We need to design a fire alarm system that all family members can use in single-family residences.

It must be able to detect fires at all locations, residents must be able to activate it from convenient locations themselves, and it must alert residents in all portions of the house.

Solution Statements:

- A properly designed, installed, operated, and maintained fire alarm system can reduce the losses associated with an unwanted fire in any building.

- The control centre will provide various functionalities to the end users such as visuals.

- The acoustic alarms in case of fire/smoke detection and extreme weather conditions, easy access to camera streams and sensor measurements, manipulation of cameras and sensors.

- The video on demand, maps for location and visualisation, visualization of fire propagation estimation, etc through a user friendly interface.

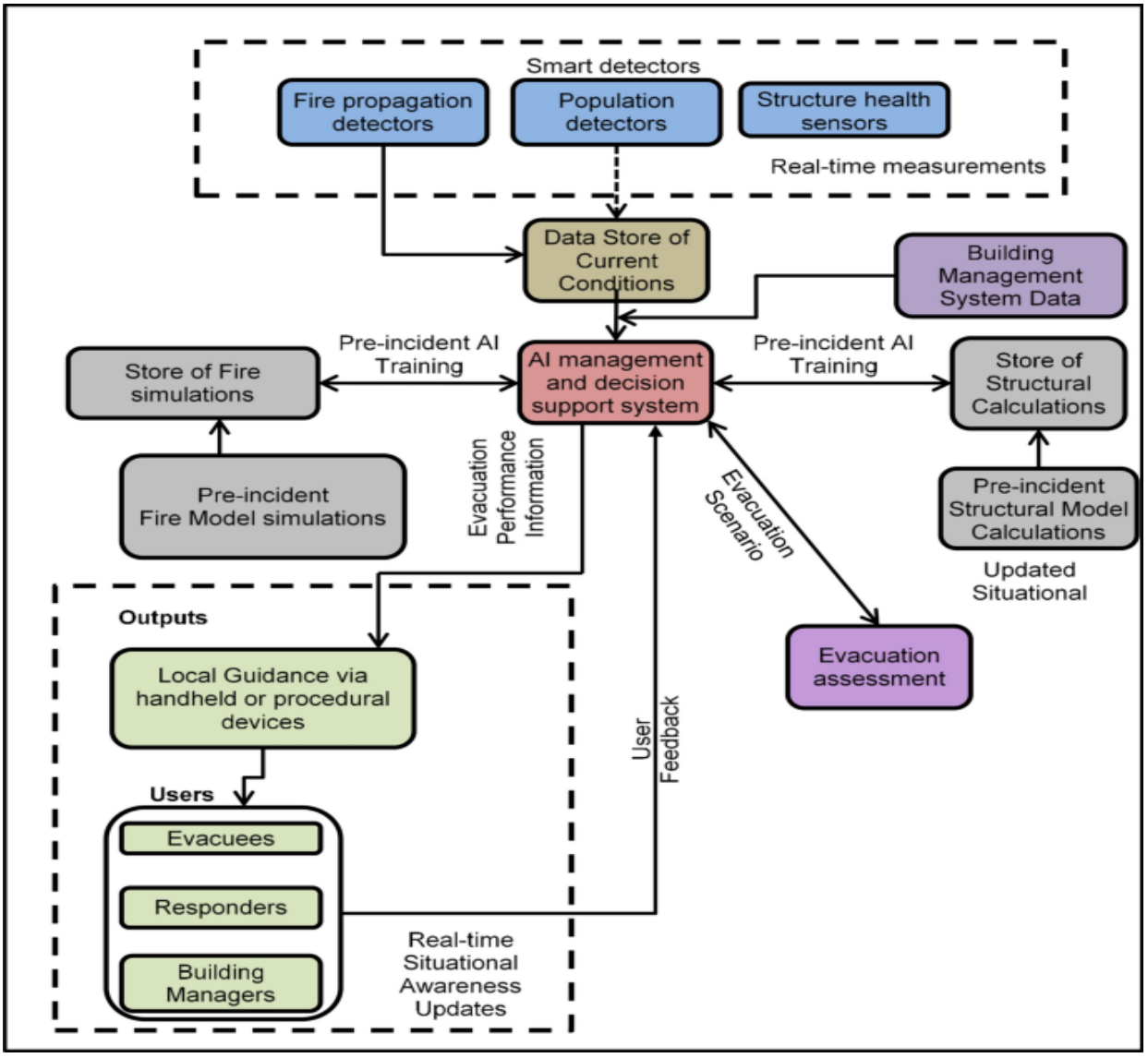
- Hence electronic circuits can be designed for the fire based alarms and they provide very high efficiency and can be used for the security reasons.

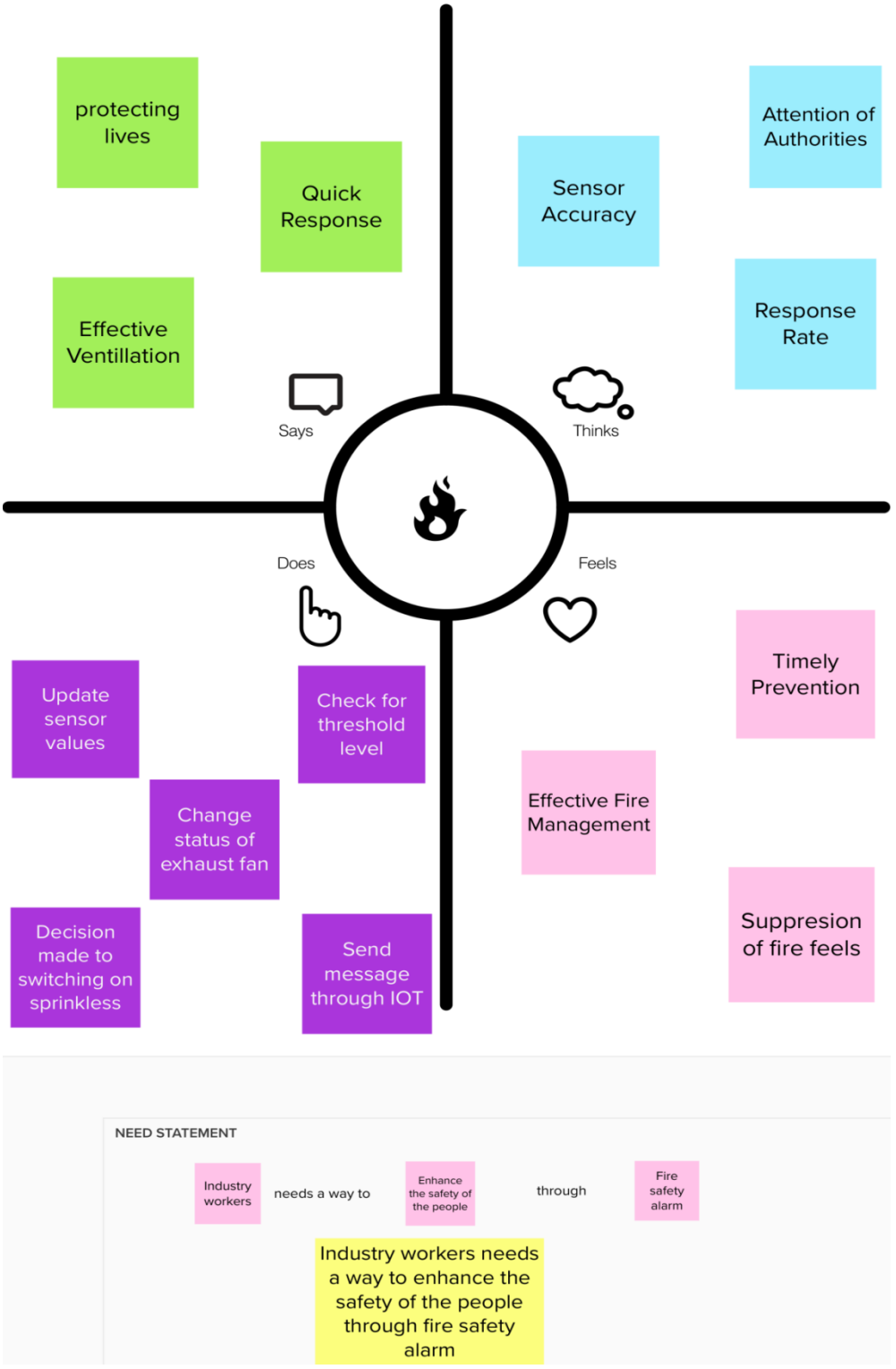
- Early fire detection is best achieved by the installation and maintenance of fire detection equipment in all rooms and areas of the house or building.

IDEATION & PROPOSED SOLUTION :

- Empathy Map Canvas
- Ideation & Brain storming
- Proposed Solution
- Problem Solution Fit

Empathy Map Canvas:





Proposed Solution :

Proposed Solution Template:

Project team shall fill the following information in proposed solution template.

S. No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	To improve the safety management system in industries. Improving the safety management system against the fire incidents in industries.
2.	Idea / Solution description	To implement the fire safety management in industry based on IOT using Arduino uno board with fire detection and fire extinguisher system. And using some sensors (Humidity sensor, Flame sensor, smoke sensor) with GPS tracking system.
3.	Novelty / Uniqueness	An Integrated system of temperature monitoring, gas monitoring, fire detection automatically fire extinguisher with accuration of information about locations and response through SMS notification and call.
4.	Social Impact / Customer Satisfaction	It early prevents the accident cost by fire in industries. Nearby locations so maximum extend more accurate reliability. Compatability design integrated system.
5.	Scalability of the Solution	It is trying to execute this technique as we need to introduce an arduino gadget which was modified with an Arduino that takes received signals from sensors. Easyoperatability and maintenance. Required low time for maintain. Cost is reasonable value.
6.	Business Model (Revenue Model)	This product can be utilized by a industries .this can be thought of as a productive and helpful item as industries great many current rescuing people and machine from the fire accident.

Problem Solution Fit:

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Who is your customer?

Customers of the fire prevention bureau are both internal and external, as is common in many major organisations.

6. CUSTOMER CONSTRAINTS CC

What constraints prevent your customers from taking action or limit their choices of solutions?

Place and keep smoke alarms. Install smoke alarms in and around bedrooms on every level of your house. Every month, test your smoke alarms.

5. AVAILABLE SOLUTIONS AS

Which solutions are available to the customers when they face the problemor need to get the job done? What have they tried inthe past?What pros & cons do these solutions have?

Utilizing a functioning smoke alarm cuts the danger of dying in a fire in half. Additionally, because the firefighters are dispatched to the fire sooner, it significantly reduces material losses.

Explore AS, differentiate

Focus on J&P, tap into

2. JOBS-TO-BE-DONE / PROBLEMS J&P

Which jobs-to-be-done (or problems) do you addressfor your customers?

the outside of the building using materials that could be discovered nearby in a vandalism crime.
Therefore, this particular issue can be greatly reduced by using appropriate security measures, such as the protection of stored products and the effective and fast evacuation of trash.

9. PROBLEM ROOT CAUSE RC

What is the real reason that this problem exists? What is the back story behind the need to do this job?

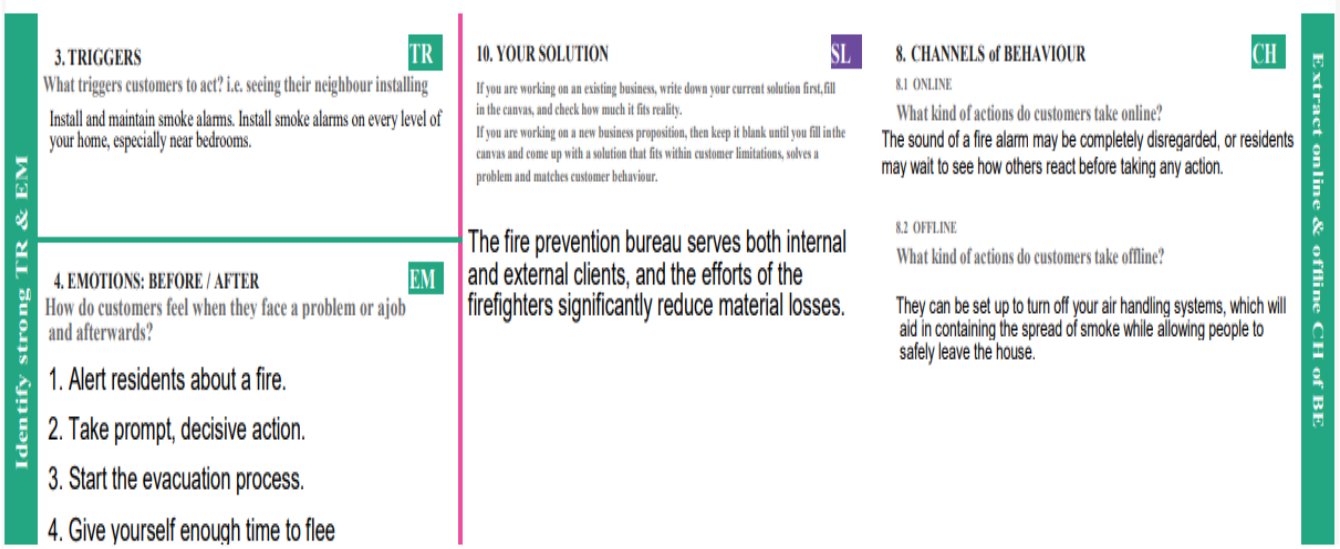
Because you have a lot of control over it, you should consider the likelihood of arson while assessing your risks. The majority of intentionally started fires occur in locations where vandalism or fire-setting had previously been documented. Typically, neighbourhood children set fire to neighbouring objects as a form of outside vandalism against the building.
The implementation of proper security measures, such as the protection of stored goods and the quick and efficient evacuation of waste, can therefore significantly lessen this particular problem.

7. BEHAVIOUR BE

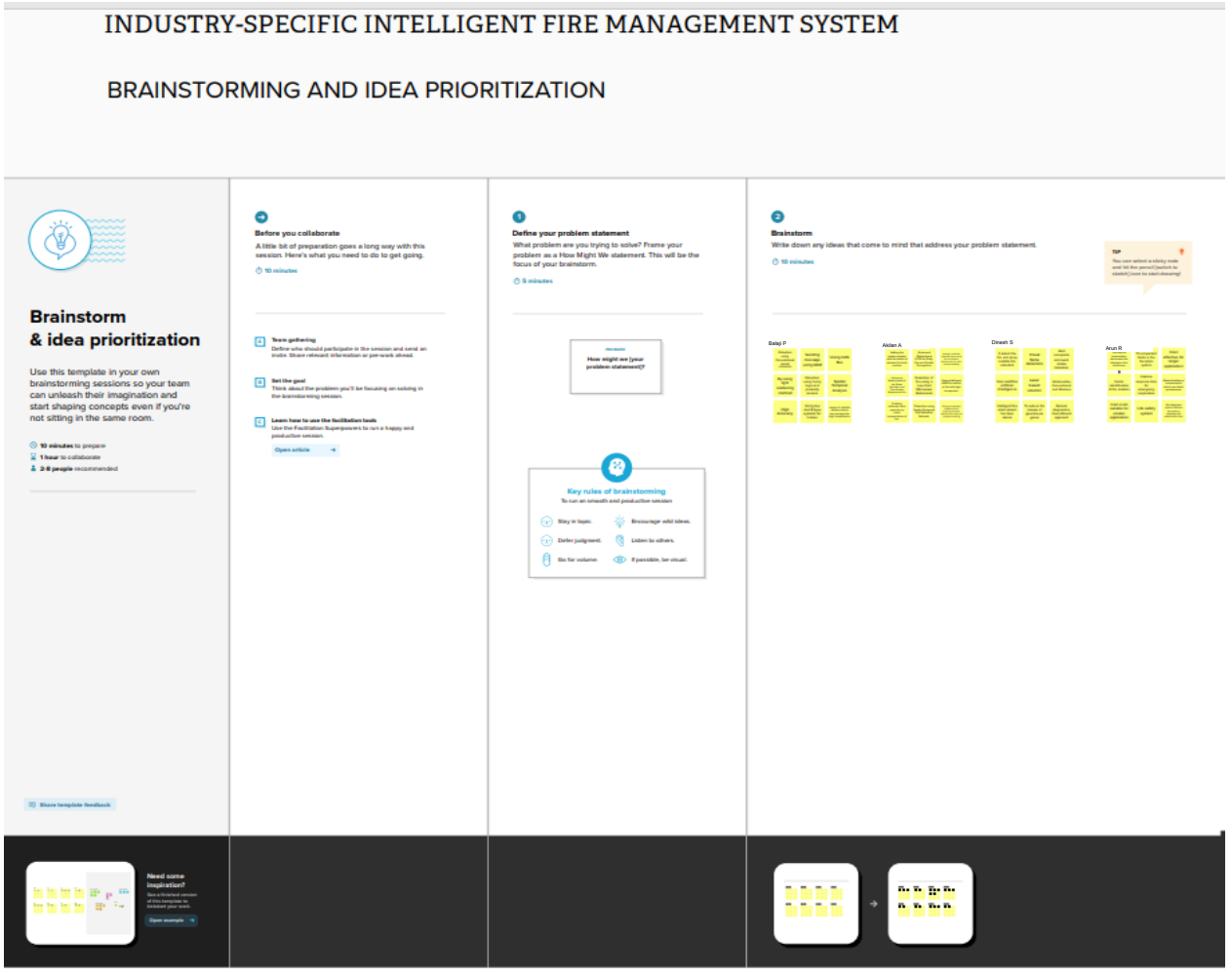
What does your customer do to address the problem andget the job done?

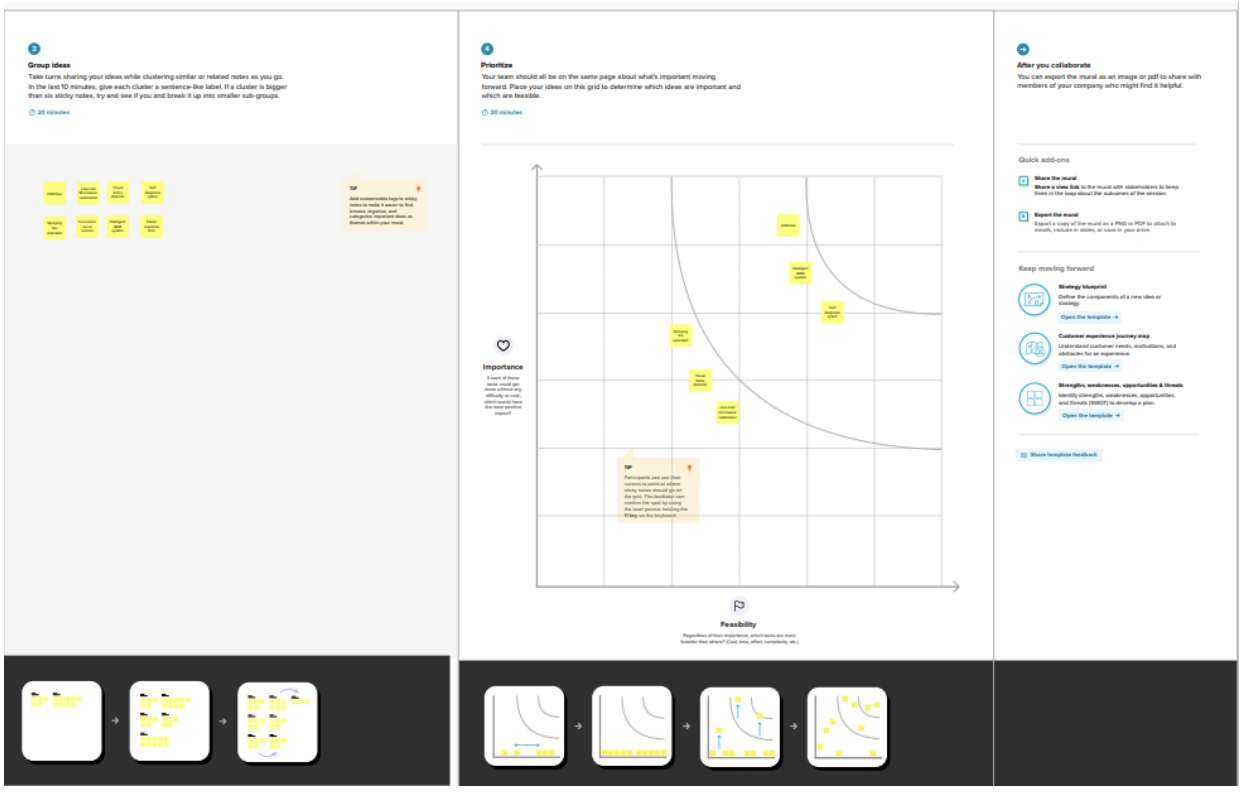
Lock the door to the space where the fire is. Moreover, activate the nearby fire alarm system.

Focus on J&P, tap int C



BRAINSTORMING AND IDEA PRIORITIZATION :





Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Usability requirements includes language barriers and localization tasks. Usability can be assessed byEfficiency of use.
NFR-2	**Security**	Access permissions for the particular systeminformation may only be changed by the system's data administrator.
NFR-3	**Reliability**	The database update process must roll back allrelated updates when any update fails.
NFR-4	**Performance**	The front-page load time must be no more than 2seconds for users that access the website.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through website or application Registration through Social medias Registration through LinkedIN
FR-2	User Confirmation	Verification via Email or OTP
FR-3	User Login	Login through website or App using the respective username and password
FR-4	User Access	Access the app requirements
FR-5	User Upload	User should be able to upload the data
FR-6	User Solution	Data report should be generated and delivered to user for every 24 hours
FR-7	User Data Sync	API interface to increase to invoice system

Non-functional Requirements:

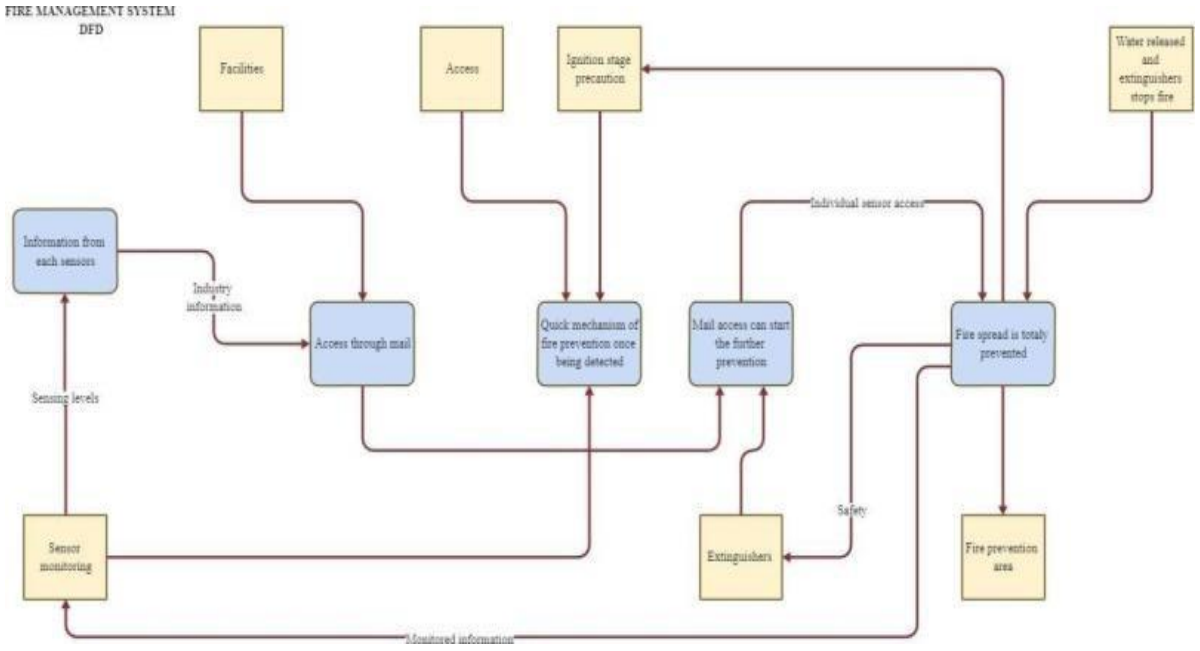
Following are the non-functional requirements of the proposed solution.

NFR-5	**Availability**	New module deployment must not impact front page, product pages, and check out pages availability and mustn't take longer than one hour.
NFR-6	**Scalability**	We can increase scalability by adding memory, servers, or disk space. On the other hand, we can compress data, use optimizing algorithms.

Project Design:

- Data Flow Diagram
- Technical Architecture
- User Stories

Data Flow Daigram:



Technical Architecture :

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

Guidelines:

In Points**:**

1. Include all the processes (As an application logic / Technology Block)
2. Provide infrastructural demarcation (Local / Cloud)
3. Indicate external interfaces (third party API's etc.)
4. Indicate Data Storage components / services
5. Indicate interface to machine learning models (if applicable)

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Web UI, Node-RED, MIT app	IBM IoT Platform, IBM Node red, IBM Cloud
2.	Application Logic-1	Create Ibm Watson IoT platform and create node-red service	Ibm Watson, ibm cloudant service,ibm node-red
3.	Application Logic-2	Develop python script to publish and subscribe to IBM IoT Platform	python
4.	Application Logic-3	Build a web application using node-red service	IBM Node-red
5.	Database	Data Type, Configurations etc.	MySQL
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant
7.	File Storage	Developing mobile application to store and receive the sensors information and to react accordingly	Web UI,python
8.	External API-1	Using this IBM fire management API we can track the temperature of the incident place and where the fire had been attacked.	IBM fire management API
9.	External API-2	Using this IBM Sensors it detects the fire, gas leaks , temperature and provides the activation ofsprinklers to web UI	IBM Sensors
10.	Machine Learning Model	Using this we can derive the object recognition Model	Object Recognition Model

11.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Cloud Server Configuration	IBM cloudant, IBM IoT Platform

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	MIT app Inventor	MIT License
2.	Security Implementations	IBM Services	Encryptions, IBM Controls
3.	Scalable Architecture	sensor-IoT Cloud based architecture	cloud computing and AI
4.	Availability	Mobile, laptop, desktop	MIT app
5.	Performance	Detects the Fire, gas leak,temperature	sensors

Users 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	As a user, I can register for the application by entering my mail, password, and confirming my password	I can access my account/ dashboard	High	Sprint-1

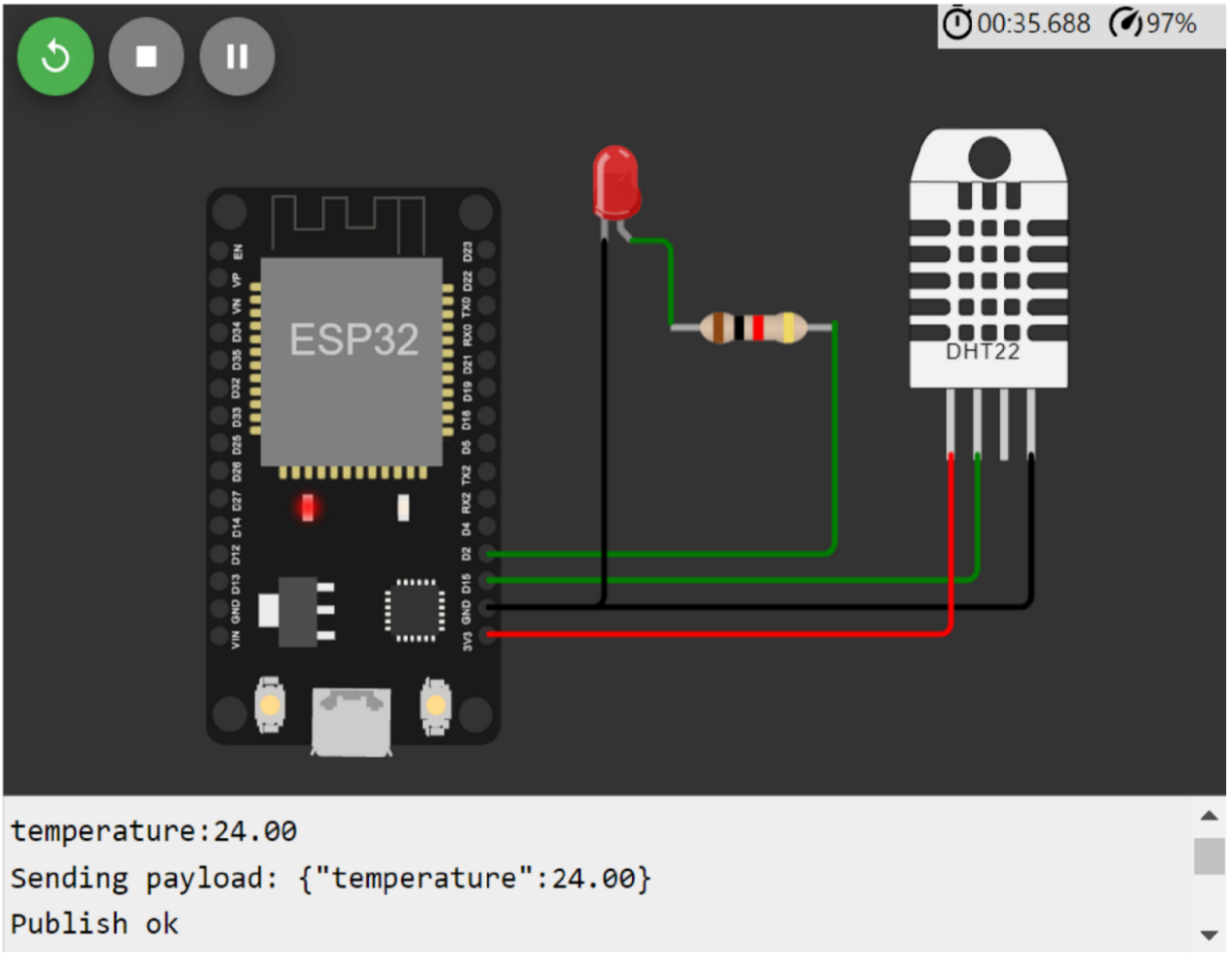
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
	Dashboard	USN-3	As a user, I can register for the application through internet	I can register & access the dashboard Eith with Internet login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can confirm the registration in Gmail	Medium	Sprint-1
	Login	USN-5	As a user log into Application.	I can login with my id password	High	Sprint-1

Project Planning & Scheduling :

Project Development Delivery Of

Sprint-1:

Display the temperature values:



Screen Shot

Program:

```
#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 "zbgr67"
//IBM ORGANITION ID
#define DEVICE_TYPE "fershidevicetype"

//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "fershideviceid"

//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "fershiageona"//Token String data3; float 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();
```

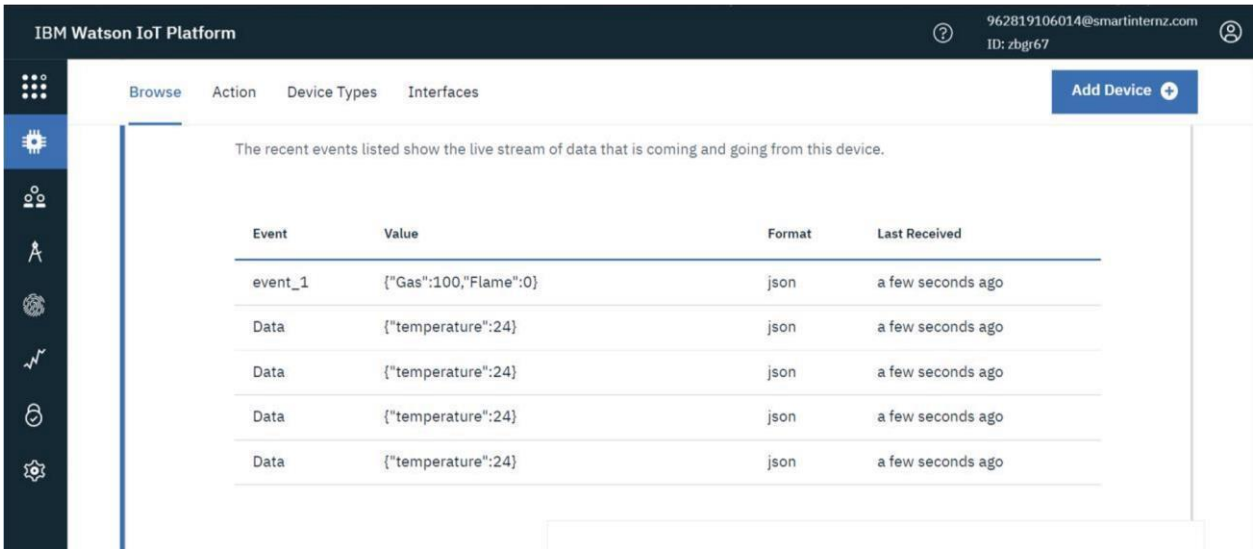
```
} void loop()
// Recursive Function
{
t= dht.readTemperature();
Serial.print("temperature:");Serial.println(t);
 PublishData(t);
delay(1000);
 if (!client.loop())
{ mqttconnect();
}
}
/*.....retrieving to
Cloud */
void PublishData(float temp)
{ mqttconnect();
//function call for connecting to ibm
/*
creating the String in in form JSon to update the data to ibm cloud
*/
String payload = "{\"temperature\":";  payload
+= temp;      payload += "}";
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 thecloud then it will print publish ok in Serial monitor or else it will print publish failed

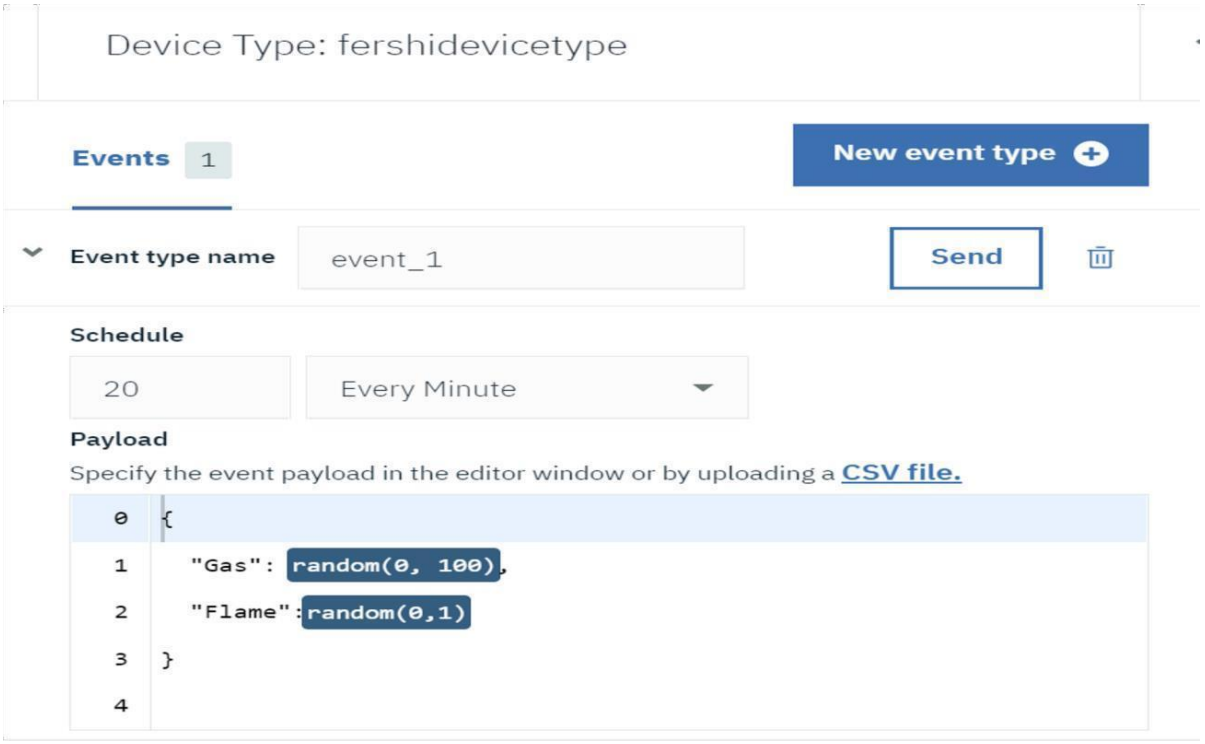
```
} else {
Serial.println("Publish failed");
}
 } void mqttconnect() {
if (!client.connected()) {
Serial.print("Reconnecting client to "); Serial.println(server);
while (!!!client.connect(clientId, authMethod, token))
{ Serial.print(".");  delay(500);
}
initManagedDevice();
Serial.println();
} } void wificonnect() //function defination for wificonnect
{
Serial.println("data: "+ data3);
if(data3=="lighton")
{
Serial.println(data3);
digitalWrite(LED,HIGH);
}        else
{
Serial.println(data3);
digitalWrite(LED,LOW);
} data3="";
}
```

Sprint 2:

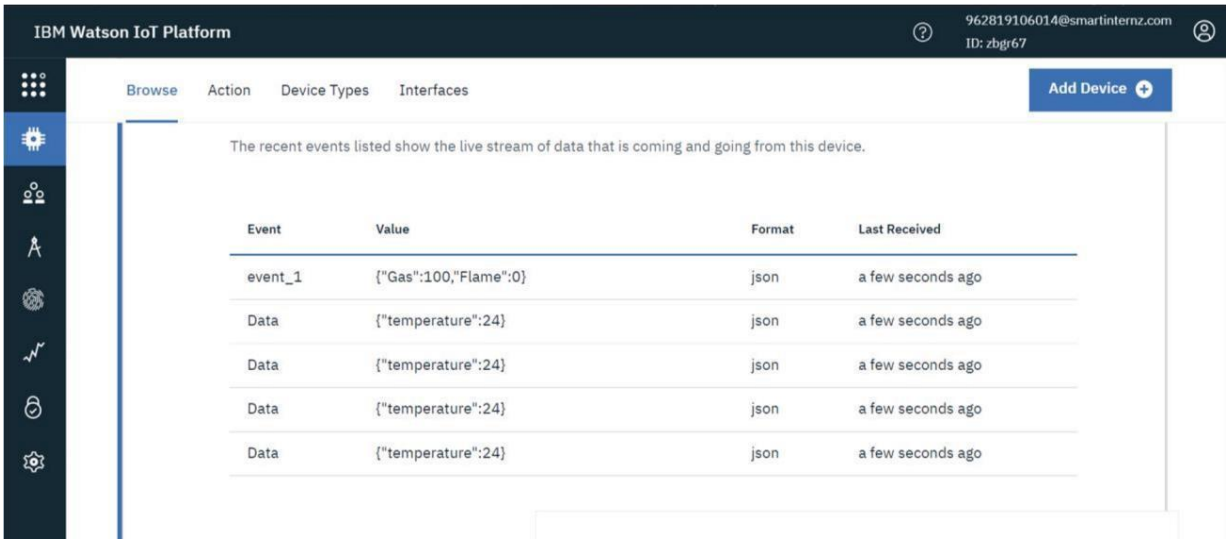
Displaying flame sensor values:



Code:

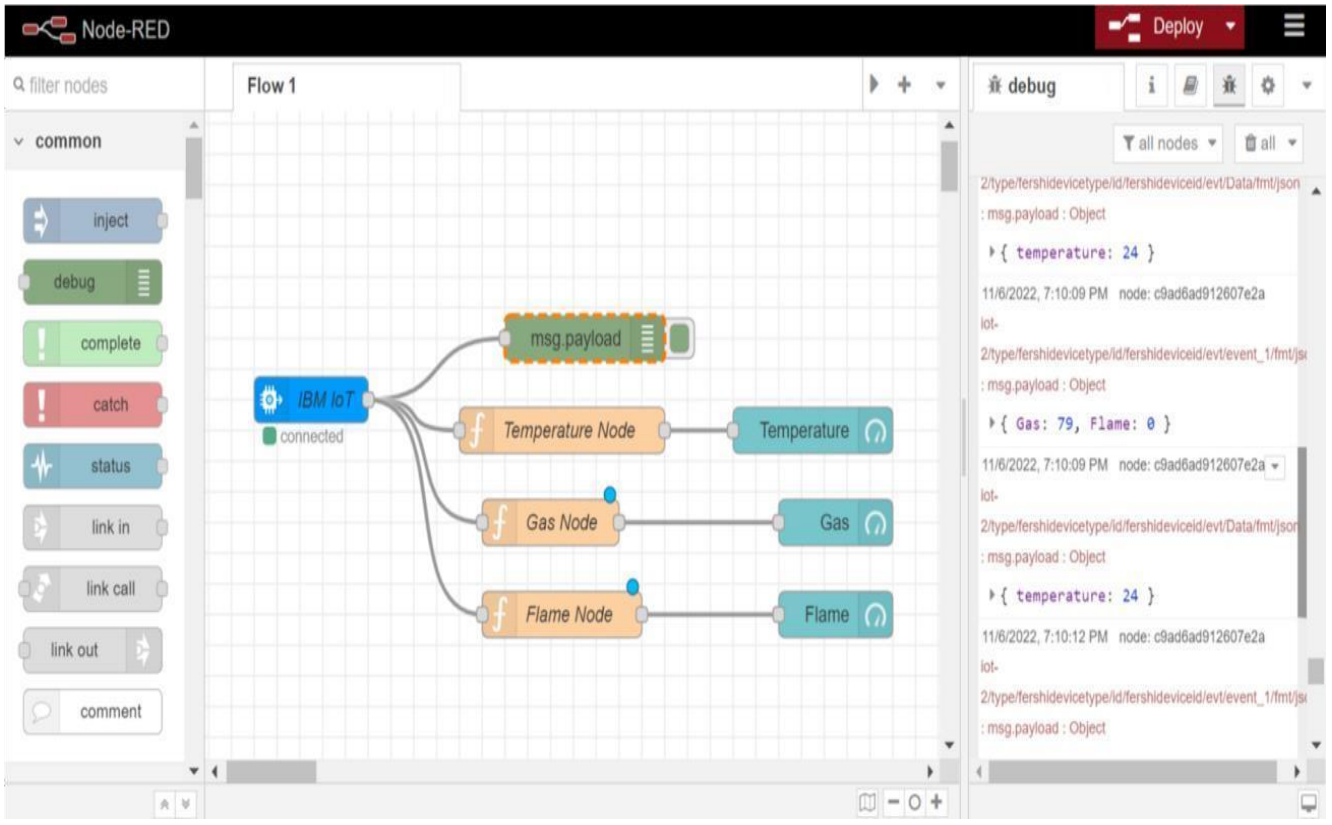


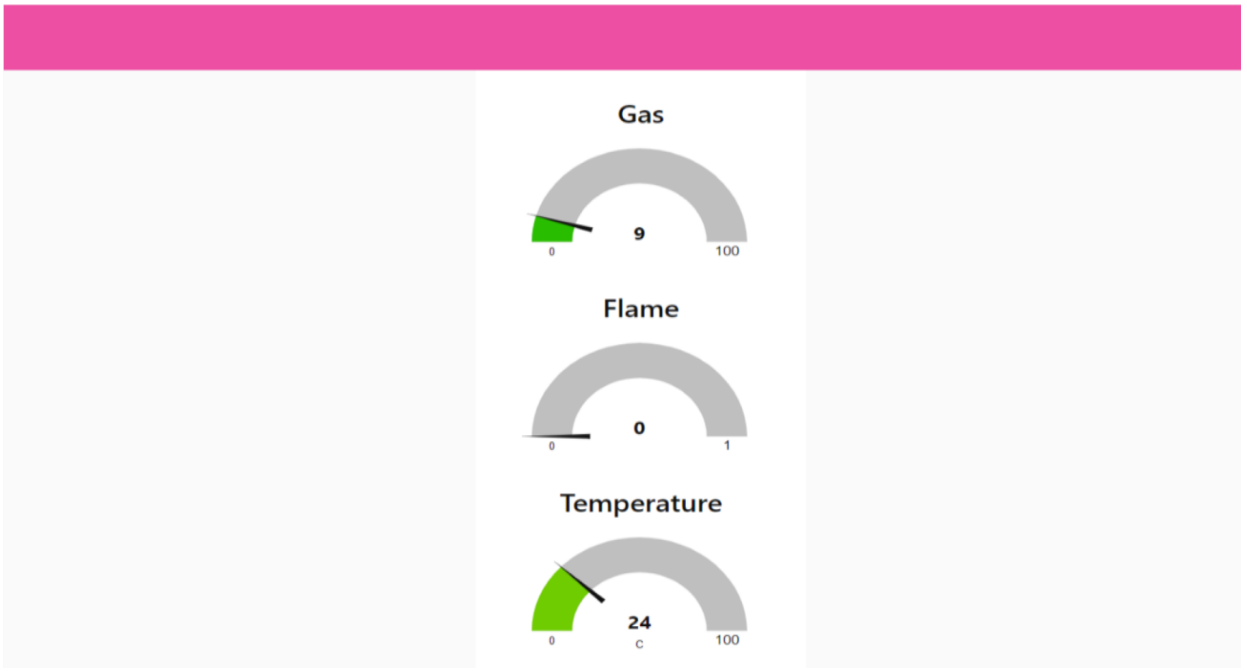
Displaying gas sensor values:



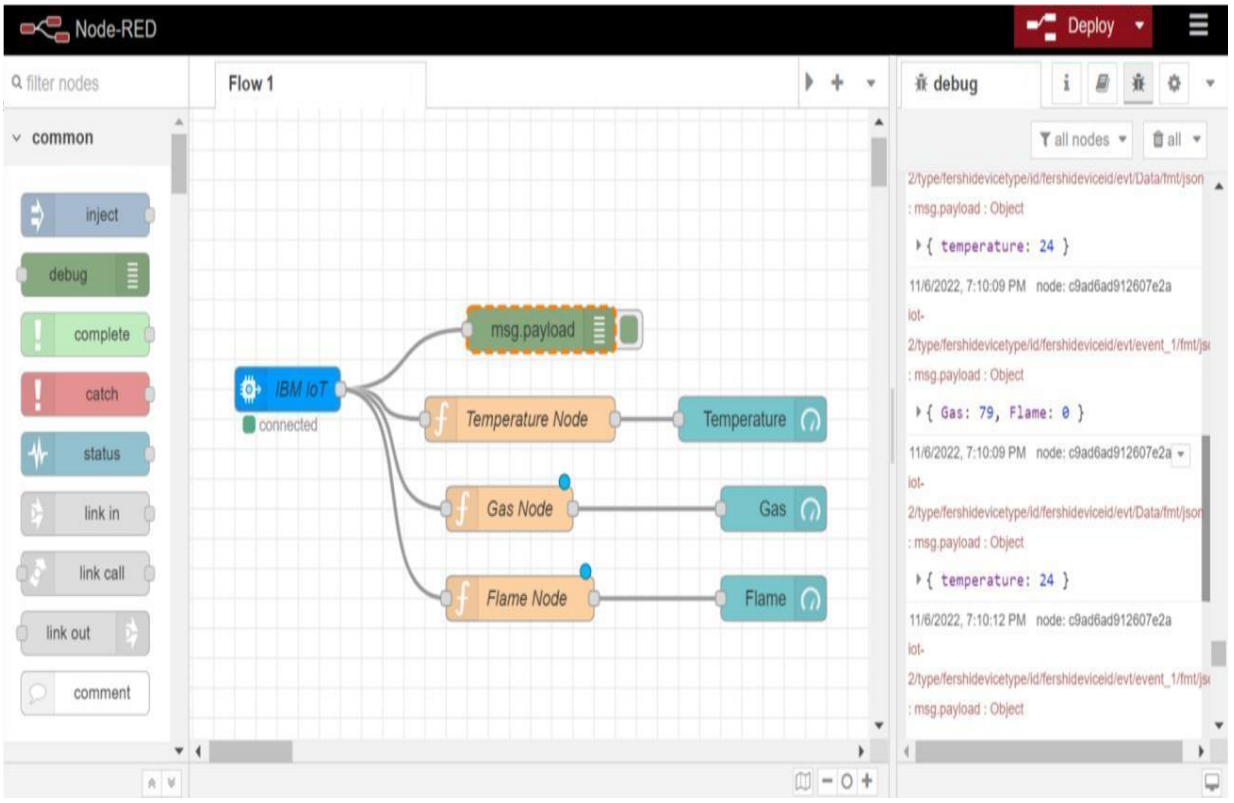
MONITORING SENSOR VALUES:

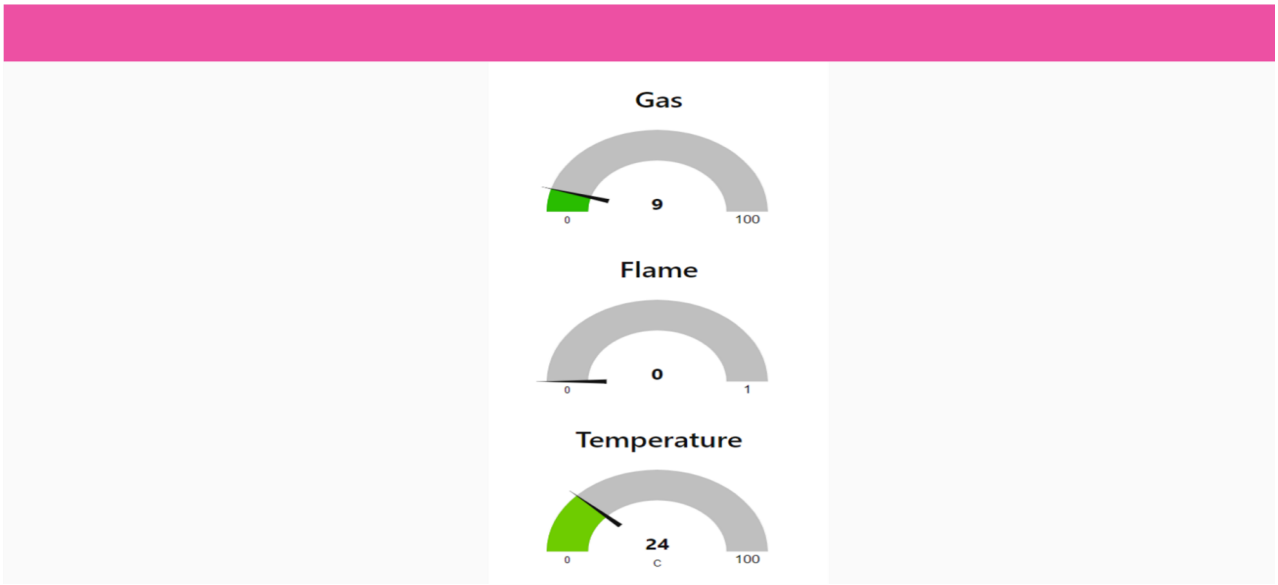
Display the temperature values in the dashboard:



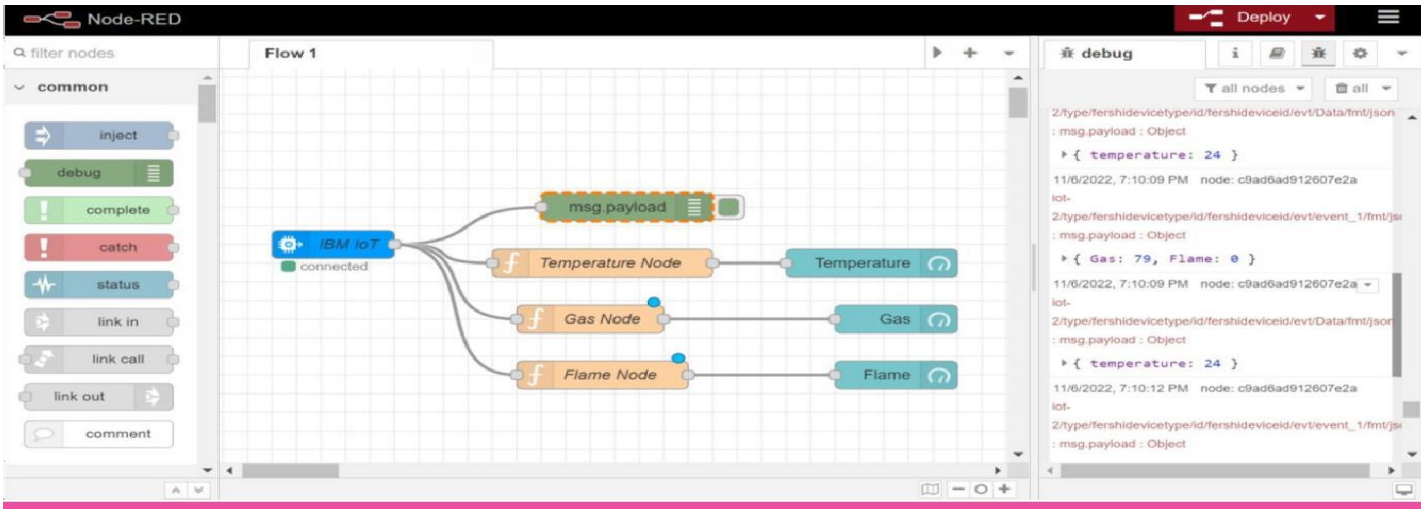


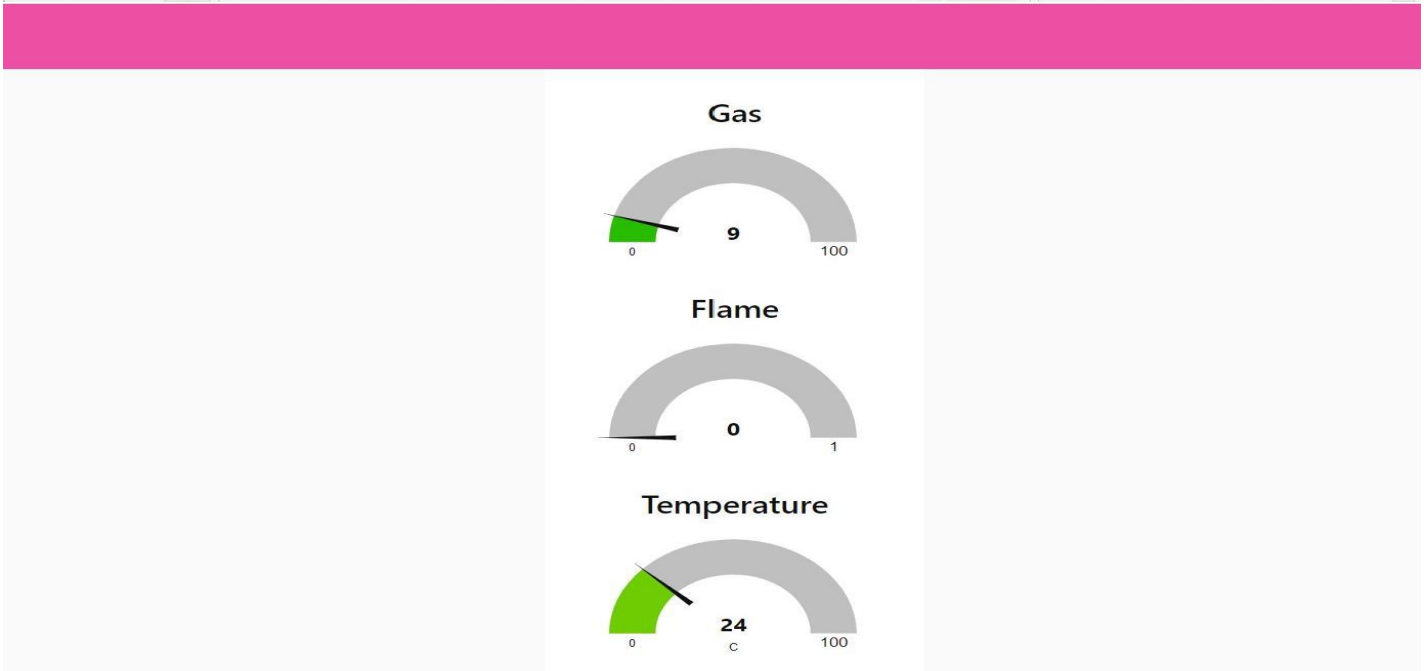
Displaying flame sensor values:





Displaying gas sensor values:





Sprint 3:

Code :

```
#include <WiFi.h>
 #include <Wire.h>
 #include <SPI.h>
#include "ThingSpeak.h"
 #include<WiFiClient.h>

unsigned long my Channel Number = 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[] = "RATHIDEVI";
char pass[]= "RATHIDEVI";
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()
{
 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 sensorif (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();
}
```

Sprint 4:

Code:

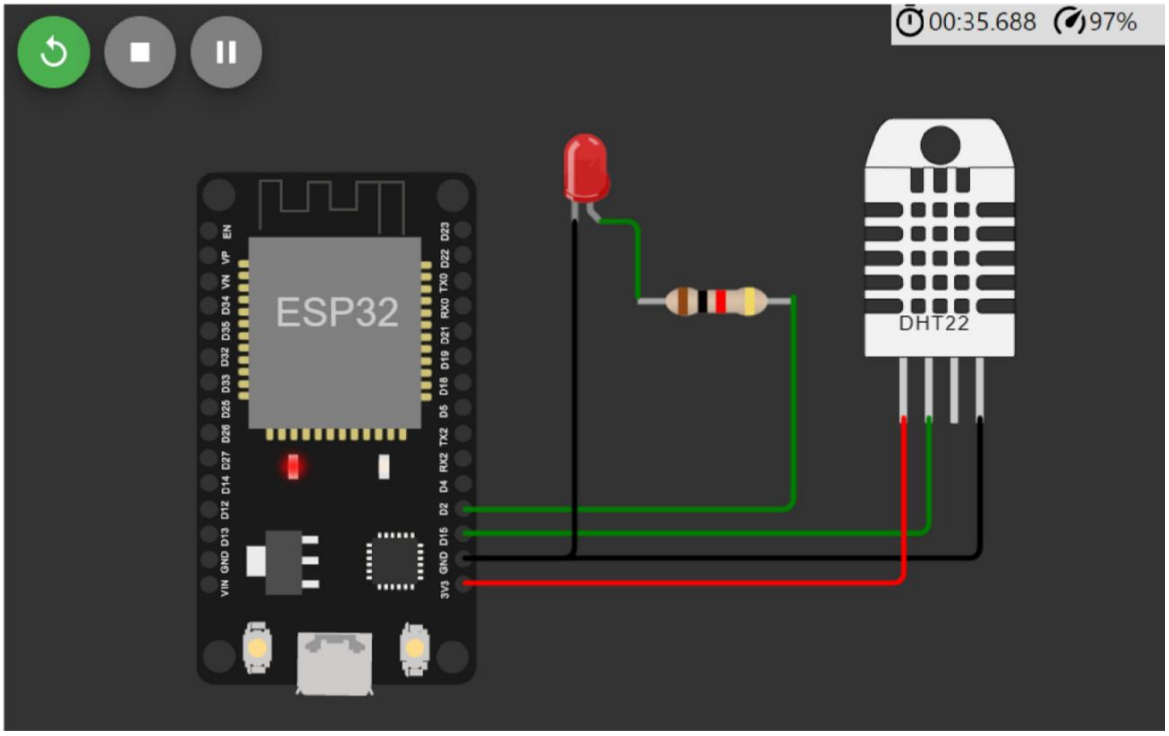
```
#include <WiFi.h>
 #include <Wire.h>
 #include <SPI.h>
#include "ThingSpeak.h"
 #include<WiFiClient.h>

unsigned long my Channel Number = 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[] = "RATHIDEVI";
char pass[]= "RATHIDEVI";
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()
{
 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 sensorif (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();
}
```



Milestone and Activity List:

TITLE	DESCRIPTION	DATE
Literature Survey& Information Gathering	A literature review is a comprehensive summary of previous researches on the topic. The literature review surveys scholarly articles, books, and other sources relevant to a particular area of research.	12 October 2022

Prepare Empathy Map	An empathy map is a collaborative tool teams can use to gain a deeper insight into their customers. It helps us to understand the customer's pain, gain and difficulties from their point of view.	12 October 2022
Ideation - Brainstorming	problem-solving method that helped us to gather and organize various ideas and thoughts from Team members.	13 October 2022

Problem Solution Fit	It helped us understand and analyze all the thoughts of our customer,their choice of options, problems, root cause, behavior and emotions.	14 October 2022
Proposed solution	It helped us analyze and examine our solution more in the grounds of uniqueness, social impact, business model, scalability etc.	15 October 2022
Solution Architecture	Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. It helped us understand the features and components used to complete the project.	16 October 2022
Customer journey	It helped to analyse the various steps, interactions, goals and motivation, positives, negatives and opportunities.	16 October 2022

Functinal requirements	It briefs about functional and non-functional requirements. It involvesthe various steps in the entire process. It also specifies features usability, security, reliability, performance, availability and scalability.	17 October 2022
Technology Architecture	A tech stack is the combination of technologies a company uses to build and run an application or project. Ithelps us analyse and understand various technologies that needs to be implemented in the project.	17 October 2022
Data flow Diagrams	A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neatand clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes theinformation, and wheredata is stored.	18 October 2022

Sprint Delivery plan	Sprint Planning is an event in scrum that defines what can be delivered in the upcoming sprint and how that work will be achieved. It helps us to organize and complete the work effectively and efficiently.	05 November 2022
Prepare milestone and activity list	Helps us understand and evaluate our progress and accuracy so far.	07 November 2022
Project Development - Delivery of Sprint-1	Develop and submit the developed code by testing it.	In progress

Coding And Solutions:

Develop a python code for publishing random sensor data

(Fire &Temperature if required humidity) to the IBM IOT Platform.

Feature 1:

```
import
wiotp.sdk.device
import time import
random myConfig = {
"identity": {
        "orgId": "0bm892",
        "typeId": "NodeMcu",
        "deviceId":"1234"
    },
    "auth": {
        "token": "12345678"
    } }  def
myCommandCallback(cmd):
    print("Message received from IBM IoT Platform: %s"
% cmd.data['command'])     m=cmd.data['command']

client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()  while True:
    temp=random.randint(-20,125)
hum=random.randint(0,100)
    myData={'temperature':temp, 'humidity':hum}
client.publishEvent(eventId="status", msgFormat="json", data=myData,
qos=0, onPublish=None)
    print("Published data Successfully: %s",
myData)     client.commandCallback =
myCommandCallback     time.sleep(2)
client.disconnect()
```

Feature 2:

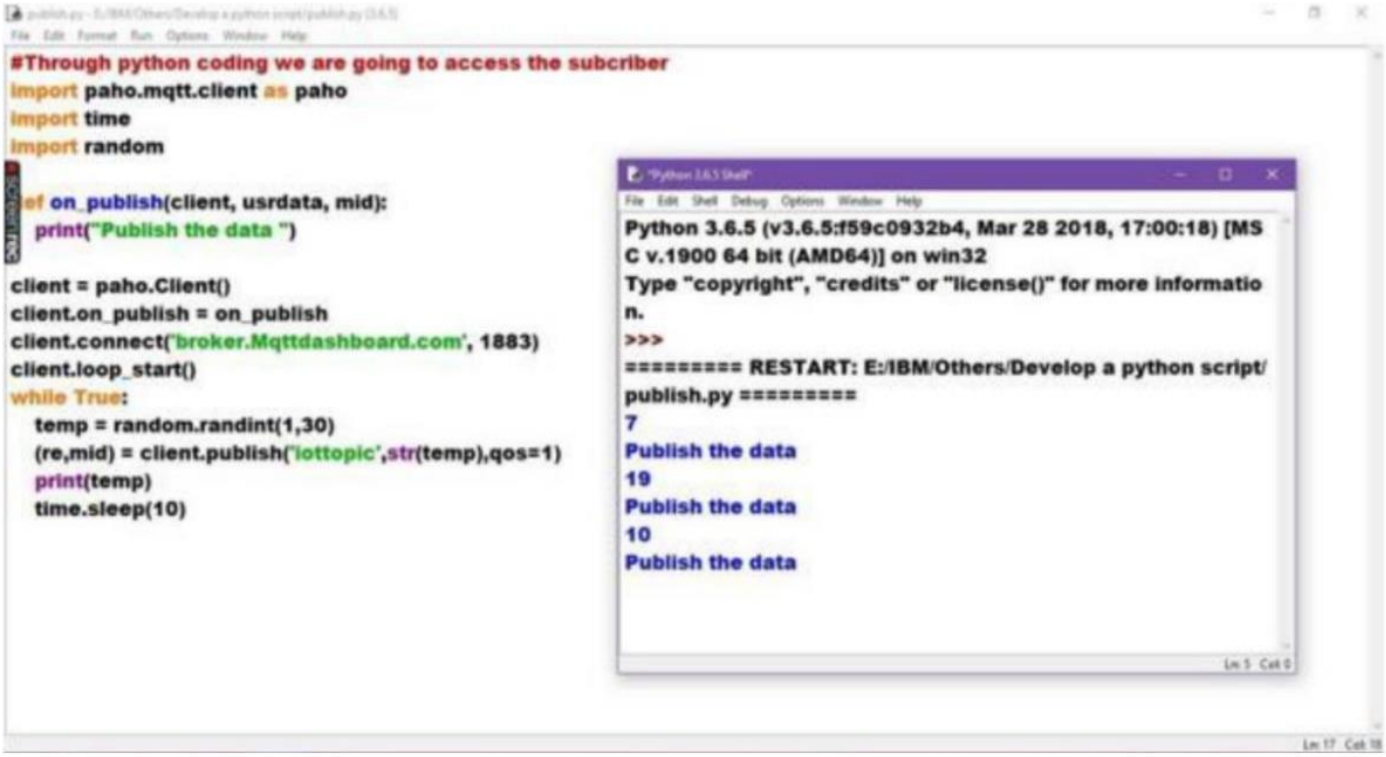
PROGRAM:

```
#IBM Watson IOT
Platform #pip install
wiotp-sdk import
wiotp.sdk.device
import time import
random
myConfig = {
 "identity": {
 "orgId": "kojkab",
 "typeId": "1234",
 "deviceId":"lee123"
 },
 "auth": {
 "token": "987456321"
 }
}
def myCommandCallback(cmd):
 print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
m=cmd.data['command']
client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()
while True:
```

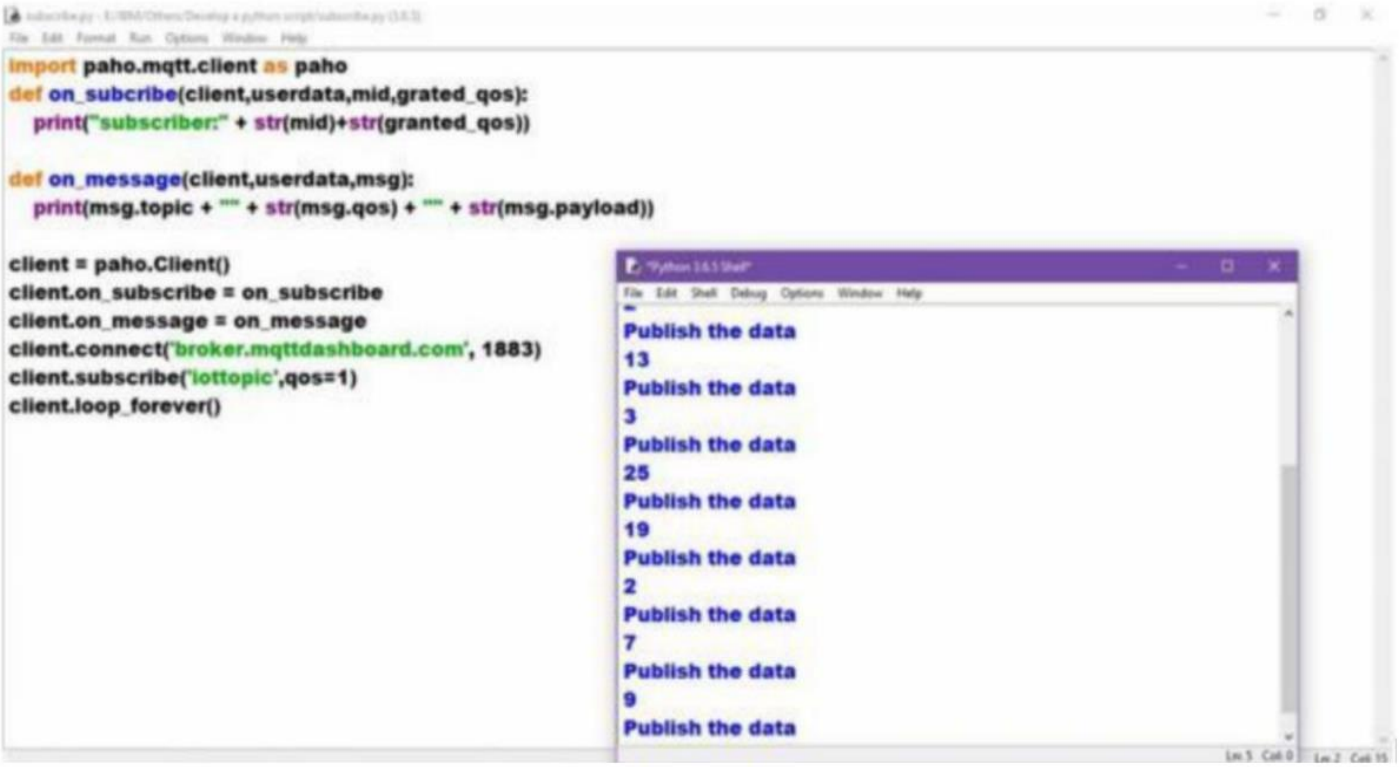
```
 temp=random.randint(-20,125)
hum=random.randint(0,100)
myData={'temperature':temp, 'humidity':hum}
 client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,
onPublish=None)
 print("Published data Successfully: %s", myData)
client.commandCallback = myCommandCallback
 time.sleep(2)
client.disconnect()
```

Screen Shots:

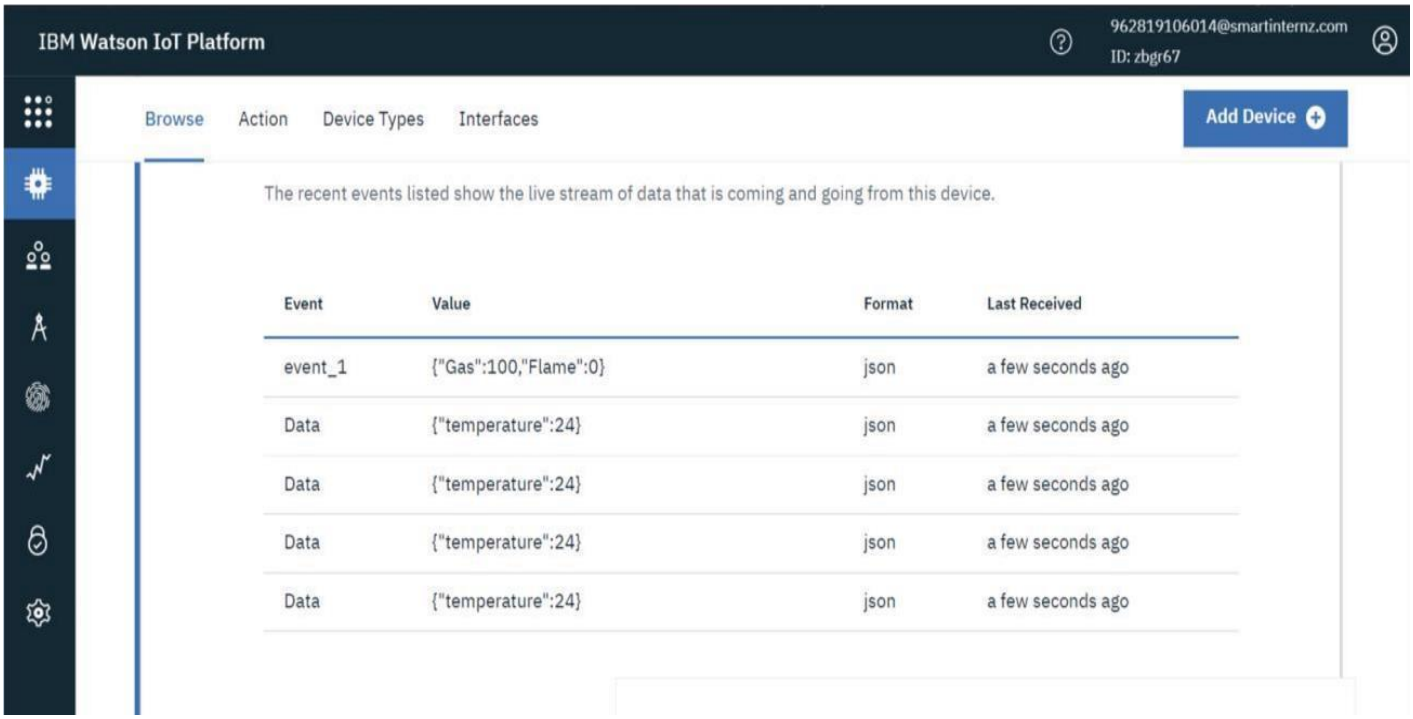
Screen Shot 1:



Screen Shot 2:



Screen Shot 3:



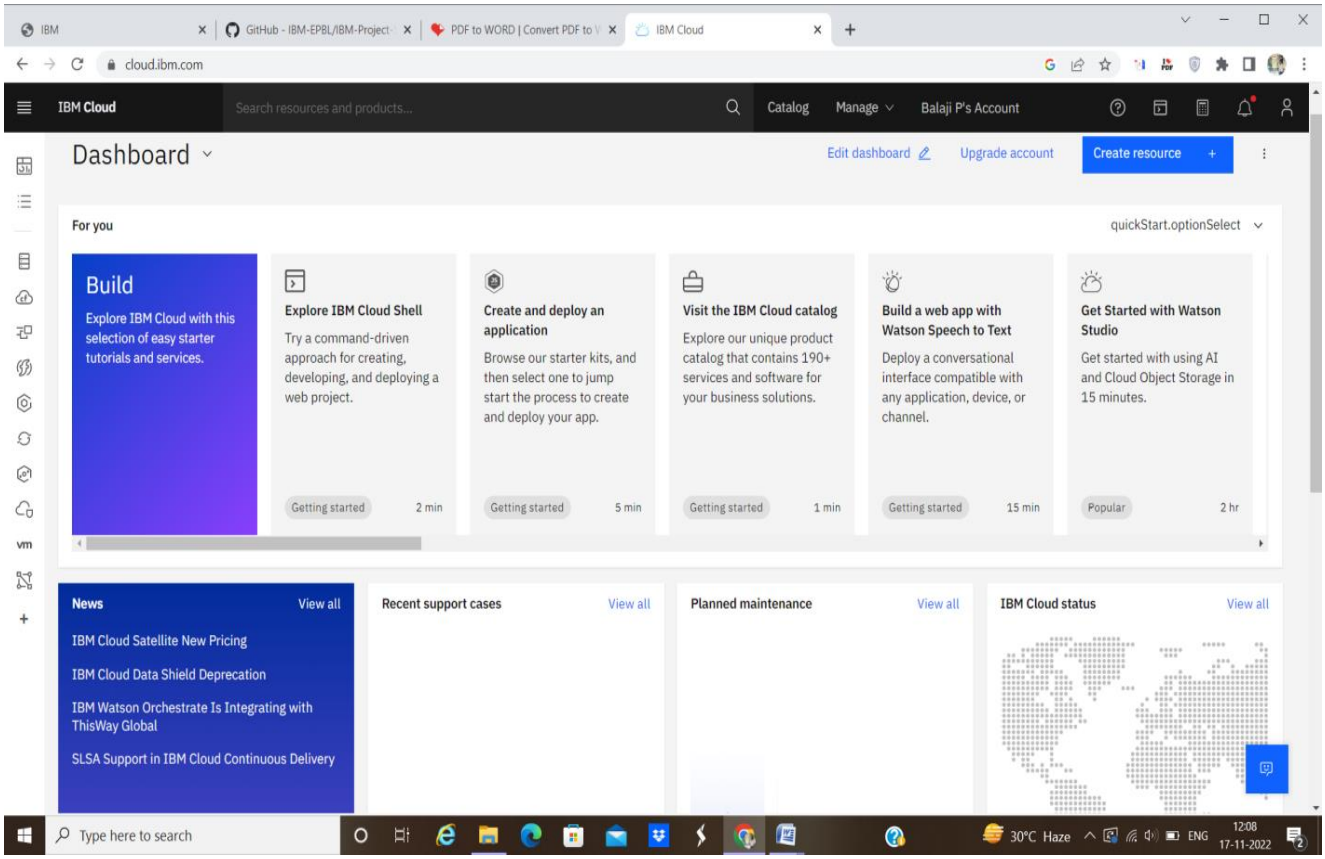
Prerequisites:

To complete this project, the prerequisites are

- IBM Cloud Services
  * IBM Watson IoT Platform
  * Node-RED Service
  * Cloudant DB

- Software
  * Python

IBM Cloud Services

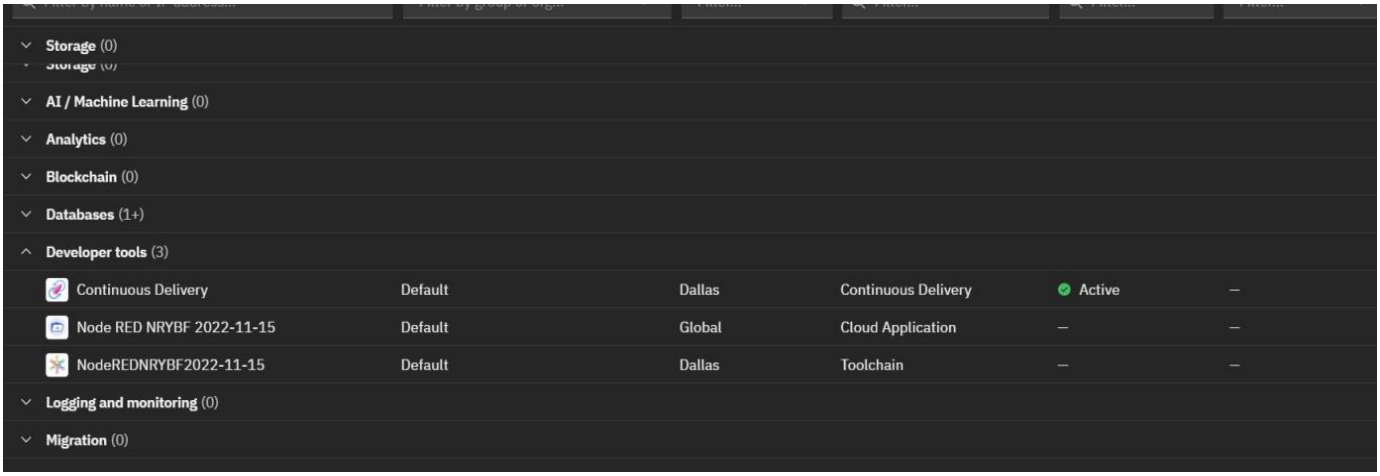


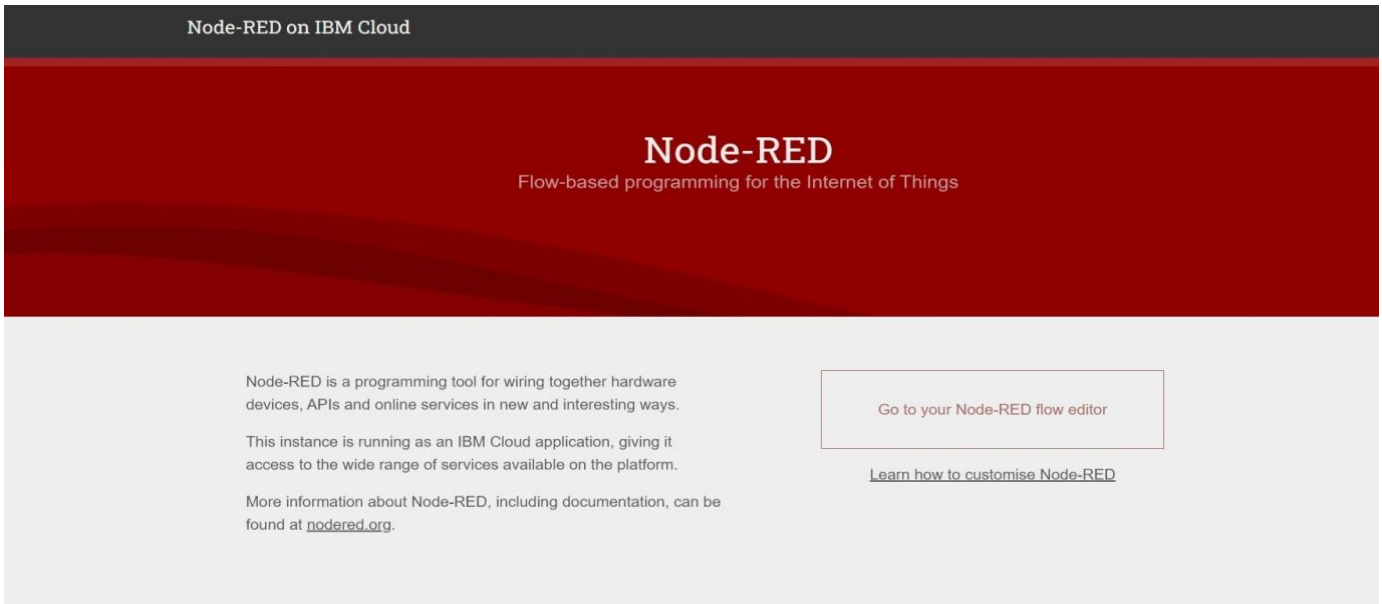
IBM Watson IoT Platform

IBM Watson IoT platform acts as the mediator to connect the web application to IoT device, so create the IBM Watson IoT platform

Node-Red

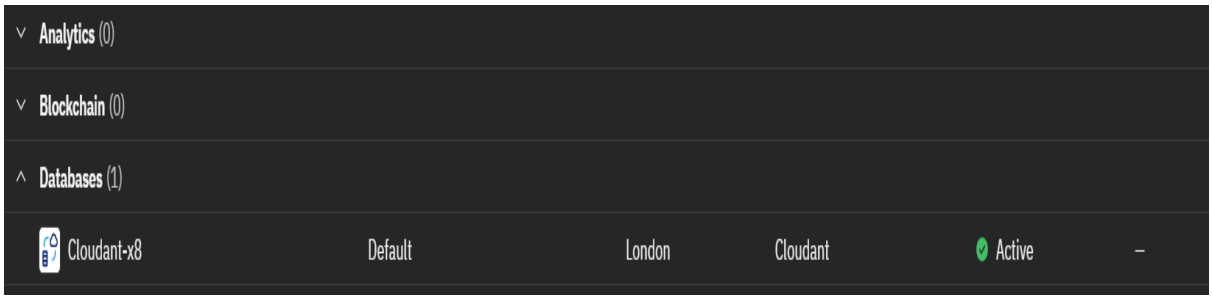
To create a web application, we need to create a Node-RED service.





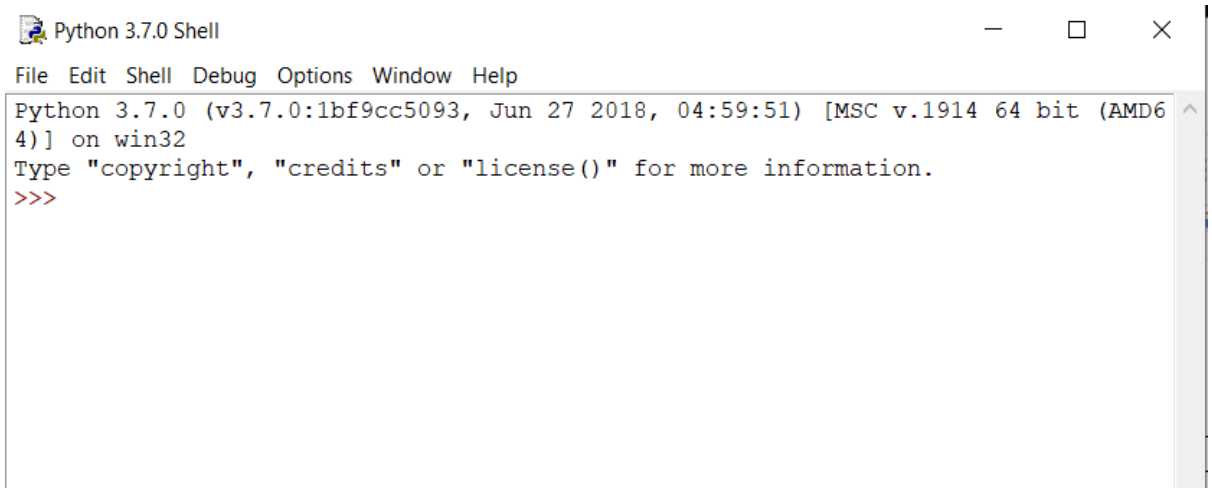
Cloudant DB

To store the sensor values, Cloud db is created.



Python

To Develop a python script to publish random sensor data to the IBM IoT platform, we need to install Python 3.7.0 Shell



Advantages & Disadvantages:

Advantages:

- Avoid Smoke & Fire Inhalation.
- Early Detection.
- Insurance Discounts.
- Environmentally friendly - no chemicals used.
- Proactive and permanent fire protection to secure business processes and valuable goods.
- Protecting multiple hazards with just one system.
- Fire protection without any interruption - no refilling or replacement needed.

Disadvantages:

- They can be costly to install and maintain, because they need more maintenance and testing.
- The wiring required can be complex, and the control panel may need to be replaced if it becomes damaged**.**
- Their initial cost can be higher than for traditional smoke detectors.
- They, also, require more maintenance than other types of detectors, as the filter system needs to be regularly cleaned or replaced.

Conclusion:

Hence electronic circuits can be designed for the fire based alarms and they provide very high efficiency and can be used for the security reasons.

Early fire detection is best achieved by the installation and maintenance of fire detection equipment in all rooms and areas of the house or building.

The performance of the project met the original technical problem, which was to build a circuit that would sound an alarm when the heat in the atmosphere reach a hazardous temperature.

Also the project was well under the overall project cost projected, making the project a good product since the application was successfully demonstrated and the circuit price was reasonable.

Future Scope:

Sensor-Assisted Fire Fighting

The way firefighters put out fires in a burning building changes once there are smart sensors installed inside.

Connected to the internet, these sensors allow firefighters to get a live feed into the progress of the fire, thereby helping them strategize the best way to handle the situation.

Using building schematics and rendered computer models from the sensor technology, firefighters are much more prepared to act effectively and safely.

Wireless Devices:

Perhaps most applicable to dealers looking to grow their RMR, wireless devices provide mobile capabilities to homeowners looking to install themselves, or even to take with them when relocating.

According to firesystemsltd.co.uk,

"Some of the systems on the market are using mesh network for the first time in wireless fire detection technology. The detectors are connected to each other and are using different frequencies on different bandwidths."

For those who look for something truly reliable in any situation, many devices can be connected in wired and non-wired formats. This dual connectivity provides unprecedented coverage and ultimate reliability.

Yet, for buildings that are difficult to wire, or consumers who want something simple, wire-free systems will take the market by storm.

Looking To The Future:

As new technologies emerge, dealers should be sure to leverage both timeless and emerging technologies to target more customers.

Devices are becoming more and more capable, and regardless their application, they are all evolving to connect to the internet and cooperate with other devices.

Consumers look for smart home technology along with commercial products that work in a cohesive environment.

Appendix:

Source Code:

```
// Chage These Credentials with your Blynk Template credentials
```


```
// Chage These Credentials with your Blynk Template credentials
```


```
#define BLYNK_TEMPLATE_ID "TMPLqCSC89Q2"

#define BLYNK_DEVICE_NAME "Fire Detection"

#define BLYNK_AUTH_TOKEN "PxJ7MvV-hMXaEwKe39Lip9vLqZRNSCOX"
```

```
#define BLYNK_PRINT Serial

#include <ESP8266WiFi.h>

#include<OneWire.h>

#include<DallasTemperature.h>

#include <BlynkSimpleEsp8266.h>

char auth[] = BLYNK_AUTH_TOKEN;

char ssid[] = "praveen";

 // Change your Wifi/ Hotspot Name

char pass[] = "24092001";

// Change your Wifi/ Hotspot Password

BlynkTimer timer;

#define fire D2

#define smoke A0 ONE_WIRE_BUS GREEN RED D6D5 D4

#define buzzer D7

int fire_Val = 0;

int data = 0;

OneWire oneWire(ONE_WIRE_BUS);

DallasTemperature DS18B20(&oneWire);

float temp = 0;

WidgetLED led(V1);

void setup() //Setup function - only function that is run in deep sleep mode

{

Serial.begin(9600); //Start the serial output at 9600 baud

pinMode(GREEN, OUTPUT);

pinMode(smoke,INPUT);

pinMode(buzzer,OUTPUT);

pinMode(fire, INPUT);

pinMode(RED, OUTPUT);
```

```
pinMode(buzzer, OUTPUT);
pinMode(ONE_WIRE_BUS, INPUT);
Blynk.begin(auth, ssid, pass);//Splash screen delay
delay(2000);
timer.setInterval(500L, mySensor);
}
void loop() //Loop function
{
Blynk.run();
timer.run();
}
void mySensor()
{
fire_Val = digitalRead(fire);
data = DS18B20 BlynkanalogRead .virtualWrite requestTemperatures (smoke (V2,data ); ();
temp = DS18B20.getTempCByIndex(0);
Blynk.virtualWrite(V3,temp);
if ((fire_Val == HIGH)||(data > 500)||(temp > 35))
{
Blynk.logEvent("fire_alert");
digitalWrite(GREEN, LOW);
digitalWrite(RED, HIGH);
tone(buzzer, 1000);
Blynk.virtualWrite(V0, 1);
Serial.print("fIRE Level: ");
Serial.println(fire_Val);
Serial.write("fire detected");
led.on();
```

```
}
else
{
digitalWrite(GREEN, HIGH);
digitalWrite(RED, LOW);
noTone(buzzer);
Blynk.virtualWrite(V0, 0);
Serial.print("fIRE Level: ");
Serial.println(fire_Val);
led.off();
Serial.write("no fire detected");
Serial.println(data);
Serial.println(temp);
}
}
```

OUTPUT DEMO LINK:

https://drive.google.com/file/d/1I__e9U4vEg9BEf0xCuQFw-ARb-PpQ-ML/view?usp=sharing

Git-Up Link:

https://github.com/IBM-EPBL/IBM-Project-30420-1660146408