INDUSTRY - SPECIFIC INTELLIGENT FIRE MANAGEMENT SYSTEM

TEAM ID	PNT2022TMID21245
PROJECT NAME	INDUSTRY-SPECIFIC INTELLIGENT FIRE MANAGEMENT SYSTEM

TEAM LEADER : LAKSHMI SREE S

TEAM MEMBER 1: BHAGYALAKSHMI T

TEAM MEMBER 2: HARSINI A.M

TEAM MEMBER 3: MADHUMITHA P.R

1.INTRODUCTION:

1.1 PROJECT OVERVIEW:

In this project, we are going to develop a smart fire management system which includes a Gas sensor, Flame sensor and temperature sensors to detect any changes in the environment. Based on the temperature readings and if any Gases are present the exhaust fans are powered ON. If any flame is detected the sprinklers will be switched on automatically. Emergency alerts are notified to the authorities and Fire station.

1.2 PURPOSE:

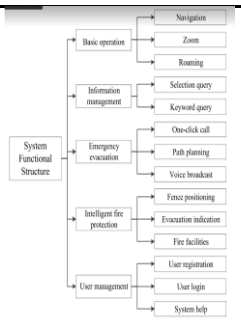
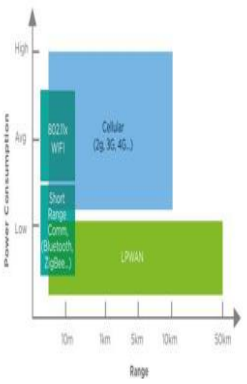
- To detect fire out break in industries and warn the industry workers.
- To save the life of industry workers.
- To protect the machines and other industry properties from damages.
- To gain knowledge of IoT platform.
- Connecting IoT devices to the Watson IoT platform and exchanging the sensor data.
- To gain knowledge on Cloudant DB.
- Creating a web application through which the user interacts with the device.

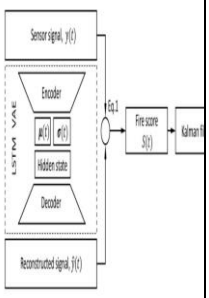
2. LITERATURE REVIEW:

2.1 EXISTING PROBLEM:

- The existing method is that, it will only detect any fire blast out in that industry. But it will not react to it by sending any messages to the admin.
- The inconsistencies are also related to the poor performance of the active and passive protection system, which in most cases fails to function in accordance with fire safety standards.
- The existing methods will not be continuously sending the environment parameters relating to the fire accidents like temperature, humidity, smoke level, carbon di oxide and other values to the respective authorites.

2.2 REFERENCES:

TITLE	YEAR OF PUBLICATION	AUTHORS(s)	METHODOLOGY	ALGORITHM USED	MODEL/FLOW DIAGRAM
1.MOBILE FIRE EVACUATION SYSTEM FOR LARGE PUBLIC BUILDINGS BASED ON ARTIFICIAL INTELLIGENCE AND IOT	2019	HUIXIAN JIANG	According to an engineering example of a shopping mall, a grid environment model is established, and the best evacuation route is planned by analyzing three different stages of fire with improved ant colony algorithm. Finally, the intelligent evacuation indicator is dynamically displayed.	Ant Colony optimization algorithm	
2. LPWAN BASED IOT SURRVEILLANCE SYSTEM FOR OUTDOOR FIRE DETECTION	2020	Vladimir Sanchez Padilla	This system presents a low-cost Internet of Things (IoT) prototype for fire detection in outdoor environments based on sensors and Low Power Wide Area	Arduino programming	

			Network (LPWAN). The system needs to set off the alarm at the moment the LM35 sensor records a threshold temperature. The system uses the SigFox back-end for sending the warning email and SMS through a callback done by the Ubidots platform.		
3. ADVANCES TOWARD THE NEXT GENERATION FIRE DETECTION: DEEP LSTM VARIATIONAL AUTOENCODER FOR IMPROVED SENSITIVITY AND RELIABILITY	2021	Zhaoyi Xu	The LSTM-VAE fire detection consists of (1) a variational autoencoder with deep LSTM networks for both the encoder and decoder, and (2) a denoising Kalman filter. We use the LSTM-VAE neural network to encode the input signal. The encoder learns the most salient features or properties of the input signal. Next, the decoder rebuilds the input signal based on the previously learned lower	Long Short-Term Memory(LSTM) Neural Network. Variational Autoencoder. Kalman Filter-Algorithm. Alarm criteria for the smoothed fire score.	 FIGURE 1. Architecture of the proposed LSTM-VAE fire detection.

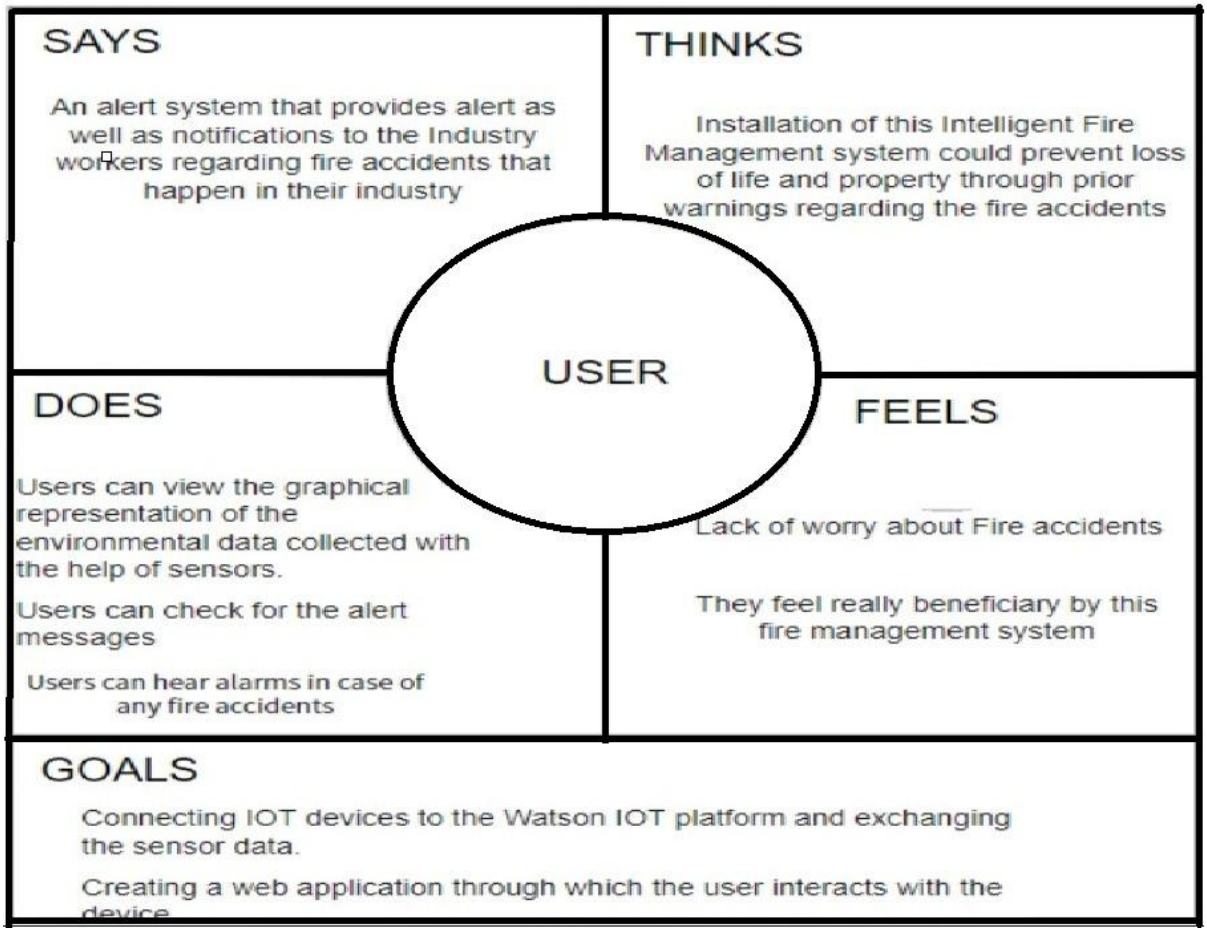
			dimensional representation of the signal. It might appear otiose to copy an input signal to a neural network and reproduce it as its output.		

2.3 PROBLEM STATEMENT DEFINATION:

Design of Intelligent fire management system that generates alerts regarding fire management system. The problem is that it will affect the industry workers who suddenly meet with the fire accidents. The issue is due to sudden occurrence of fire accidents at the industries, without any fire management system, we may face loss of life and property. This issue occurs only when sudden fire accidents break out in the industries.

3. IDEATION AND PROPOSED SOLUTION:

3.1 EMPATHY MAP CANVAS:



3.2 IDEA AND BRAINSTORMING:

• *A voice alert system that announces danger with a pre-recorded message.*

Our solution for Industry-specific fire management system includes

• The smart fire management system includes a Gas sensor, Flame sensor and temperature sensors to detect the current environmental parameters and to know any deviations from the pre-determined threshold.

• Based on the temperature readings and smoke level, if any dangerous gases are detected, the exhaust fans are powered ON. (Exhaust fans – actuators)

• If any flame is detected the sprinklers will be switched on automatically.

• Emergency alerts are notified to the authorities and Fire station.

• *A central alarm system where all the detectors can be connected to a central controller and send the signal directly to this controller.*

• *An IOT Based system with arduino(microcontroller),smoke sensor and temperature sensor to monitor the temperature and gas levels.*

• *Remote monitoring: cloud uploading of information can be set up to offer alarms on-site to collect sensor details.*

3.3 PROPOSED SOLUTION:

By grouping the ideas of the team members, we have proposed a solution to the problem.

- The smart fire management system includes a Gas sensor, Flame sensor and temperature sensors to detect any changes in the environment.
- Based on the temperature readings and if any Gases are present the exhaust fans are powered ON.
- If any flame is detected the sprinklers should be switched on.

Novelty / Uniqueness Preceding system's objectives:

- In the prior systems, the main objective is to switch on the alarms and sprinklers if the environmental readings from the sensors cross the pre-determined threshold.
- If the admin wants to know about the sensor readings, he/she needs to log in to the cloud.

Proposed system's objective:

- We will be designing a Web application through which admin can know if the read values cross the threshold values

3.4 PROBLEM SOLUTION FIT:

1. CUSTOMER SEGMENT(S) Who is your customer?

Industry people – Industries would be using this Intelligent Fire Management system for warning the fire accidents.

2. JOBS-TO-BE-DONE / PROBLEMS Which jobs-to-be-done (or problems) do you address for your customers?

Fire Management system for industries which is to be designed using IOT and cloud.

3. TRIGGERS TR What triggers customers to act?

When industries decide to protect their life and products from destruction due to sudden fire accidents, they would be triggered to buy this designed IOT solution.

4. EMOTIONS: BEFORE / AFTER EM How do customers feel when they face a problem or a job and afterwards?

Emotions before: Prior to the solution, they will be worrying about the losses that occur due to fire accidents.

Emotions after: If they face a sudden fire accident, they feel that they can control the current situation (because Temperature and smoke values read from sensor can be viewed by the admin through the web application at any time) and also, they know how to rectify it (alarms and sprinklers would be activated automatically).

5. AVAILABLE SOLUTIONS Which solutions are available to the customers when they face problem?

In the prior systems, the main objective is to switch on the alarms and sprinklers if the environmental readings from the sensors cross the pre-determined threshold. If the admin wants to know about the see about the status, he/she needs to log in to cloud.

6. CUSTOMER CONSTRAINTS What constraints prevent your customers from taking action or limit their choices of solutions?

All industry workers need to have a smart device (smart phone) and also must have connected network.

7. BEHAVIOUR What does your customer do to address the problem and get the job done?

The industry workers need to check the web app periodically and if there is any fire accident, they will get alert message. so, that they can take mandatory actions.

8. CHANNELS of BEHAVIOUR

8.1 ONLINE What kind of actions do customers take online? Extract online channels from #7

If there are any fire accidents, they can get alert message from the web app.

8.2 OFFLINE What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development. If the currently collected sensor value deviate from the threshold value, they get alarms and warnings.

9. PROBLEM ROOT BECAUSE What is the real reason that this problem exists? What is the back story behind the need to do this job?

If there is a sudden outbreak of fire accidents in industry, they cannot predict the current situation before they arise. It may lead to loss of life and property. So, this is the main reason for the designing of this solution.

10. YOUR SOLUTION

The smart fire management system includes a Gas sensor, Flame sensor and temperature sensors to detect the current environmental parameters and to know any deviations from the pre-determined threshold. Based on the temperature readings and smoke level, if any dangerous gases are detected, the exhaust fans are powered ON. (Exhaust fans – actuators) If any flame is detected the sprinklers will be switched on automatically. Emergency alerts are notified to the authorities automatically.

4. REQUIREMENT ANALYSIS:

4.1 FUNCTIONAL REQUIREMENTS:

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Admin should be able to login to his IBM cloud through his root account detail.
FR-2	Displaying sensor details	Data collected from sensors should be displayed in form of charts(smoke level, co level, Temperature level).
FR-3	Creation of Mobile Application	Sensor values generated for IBM cloud can be viewed by the Admin via the developed mobile Application
FR-4	Switching on Alarms and sprinklers	Based on the displayed sensor values necessary actions should be taken by the admin.

4.2 NON - FUNCTIONAL REQUIREMENTS:

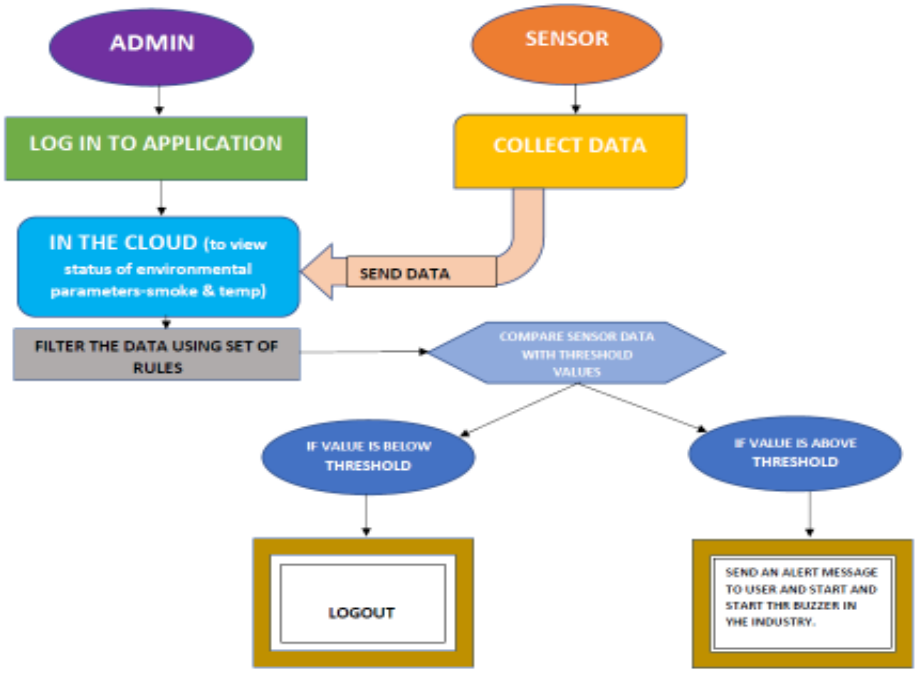
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The system should be easy for the customers to use.

NFR-2	**Security**	No should have access to modify the data sent from the sensors.
NFR-3	**Reliability**	The system should be made fail-safe operation using redundancy. If one module fails then the parallel module should take over the operation.
NFR-4	**Performance**	All the modules should work without any connection interruption. Quickly Responsive.
NFR-5	**Availability**	The system should be continuously monitoring the environment 24/7.
NFR-6	**Scalability**	The system should have the capacity to accommodate a greater amount of usage.

5. PROJECT DESIGN:

5.1 DATA FLOW DIAGRAMS:

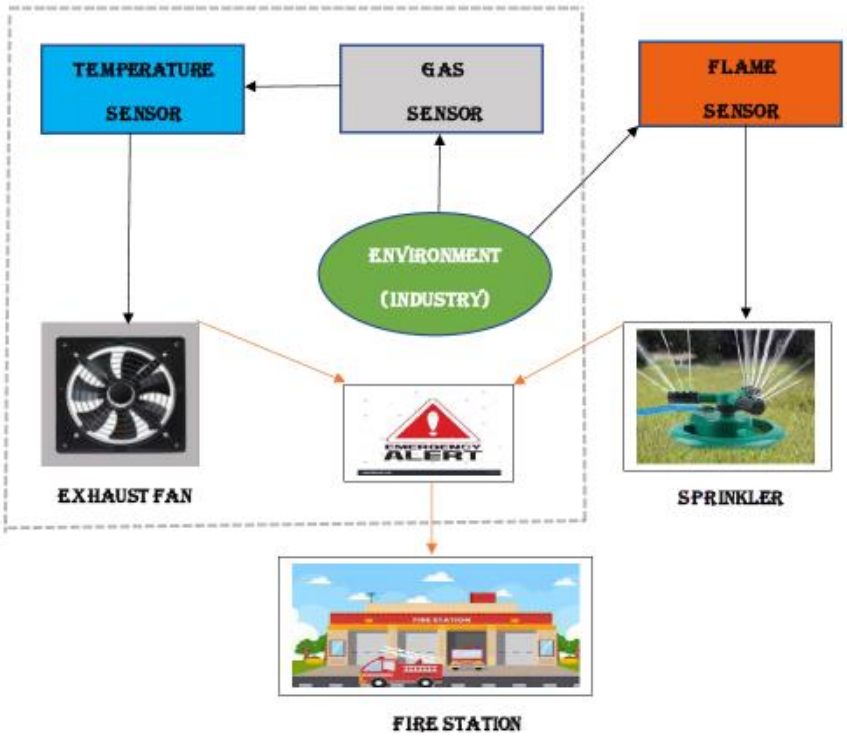
1. User configure credentials for Watson IOT hub service and starts app.
2. IOT devices connected.
3. Design of data flow in Node-Red application.
4. Parsing data from Node-Red to Watson IOT hub.
5. Creating of things in IBM Watson IOT hub.
6. Data gets published in cloud.
7. Sensor data collected from IOT devices to be displayed in applications.
8. Ringing alarms in case of deviation in the sensor load values.



5.2 SOLUTION AND TECHNICAL ARCHITECTURE:

• The smart fire management system includes a Gas sensor, Flame sensor and temperature sensors to detect any changes in the environment.
• Based on the temperature readings and if any Gases are present the exhaust fans are powered ON.
• If any flame is detected the sprinklers will be switched on automatically.
• Emergency alerts are notified to the authorities and Fire station.

Solution Architecture Diagram:



TECHNOLOGY STACK:

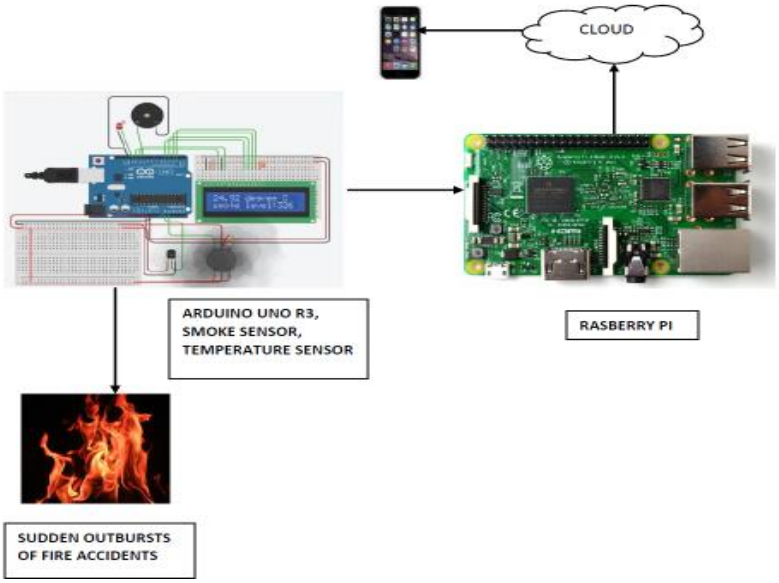


Table 1: components and Technologies

Components	Description	Technology
User interface	User interface with the equipment using mobile app ,web UI	Node-red and MIT App Inventor
Application logic 1	Logic for generating data	Python script and IBM Watson IOT platform

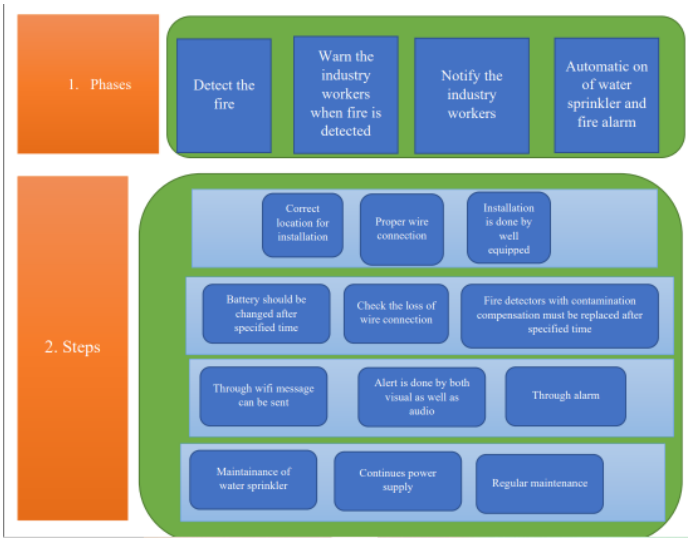
Application logic 2	Logic for detecting the condition of the switches	Python script
Application logic 3	Logic for controlling process for the fire	Raspberry pi
Cloud data base	Database Service on Cloud	IBM DB2, IBM Cloudant, IBM Watson, Node red service
File storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem

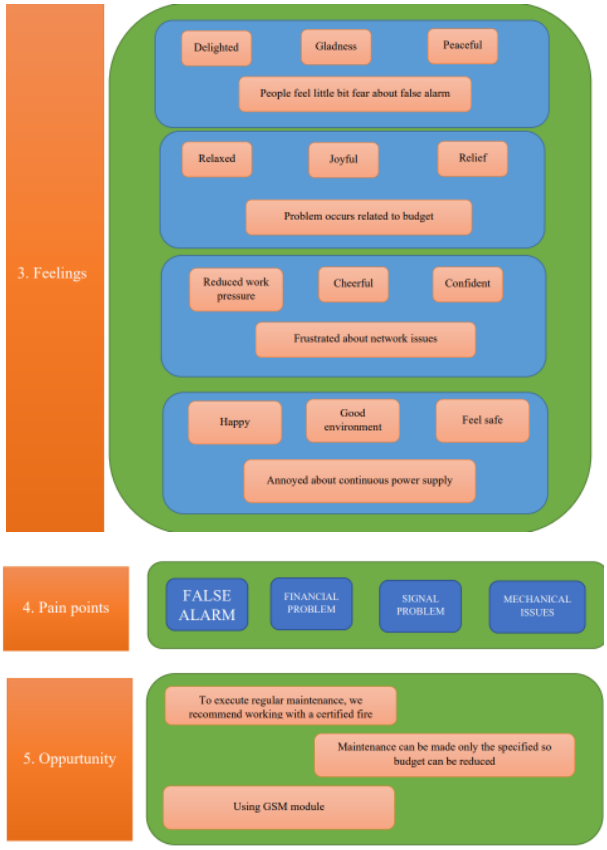
Table-2: Application Characteristics

Characteristics	**Description**	**Technology**
Throughout	High Efficiency is achieved	Using IOT
Scalability	To accommodate future changes in use and occupancy	Update can be made easily using mobile app

5.3 USER STORIES:

A user journey map (also known as a customer journey map) is a diagram that visually illustrates the user flow through your site, starting with initial contact or discovery, and continuing through the process of engagement into long-term loyalty and advocacy.





6. PROJECT PLANNING AND SCHEDULING:

6.1 SPRINT PLANNING AND ESTIMATION:

Literature Survey & Information Gathering
Literature survey on the selected project & gathering information by referring the, technical papers, research publications etc.
20 SEP 2022

Prepare Empathy Map
To prepare a Empathy Map Canvas in order to capture the user Pains & Gains, Prepare list of problem statements.
23 SEP 2022

Ideation
List the by organizing the brainstorming session and prioritize the top 4 ideas based on the feasibility & importance.
25 SEP 2022

Proposed Solution
Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.
5 OCTOBER 2022

Problem Solution Fit
Prepare problem - solution fit document.

10 OCTOBER 2022

Solution Architecture
To prepare a solution architecture document.
10 OCTOBER 2022

Customer Journey
To prepare the customer journey maps to understand the user interactions & experiences with the application (entry to exit).
15 OCTOBER 2022

Functional Requirement
To prepare the functional requirement document.
15 OCTOBER 2022

Data Flow Diagrams
To draw the data flow diagrams and submit for review.
20 OCTOBER 2022

Technology Architecture
To prepare the technology architecture diagram.
25 OCTOBER 2022

Prepare Milestone & Activity List
To prepare the milestones & activity list of the project.
1 NOVEMBER 2022

Project Development - Delivery of Sprint-1, 2, 3 & 4
To develop & submit the developed code by testing it.
12 NOVEMBER 2022

6.2 SPRINT DELIVARY SCHEDULE:

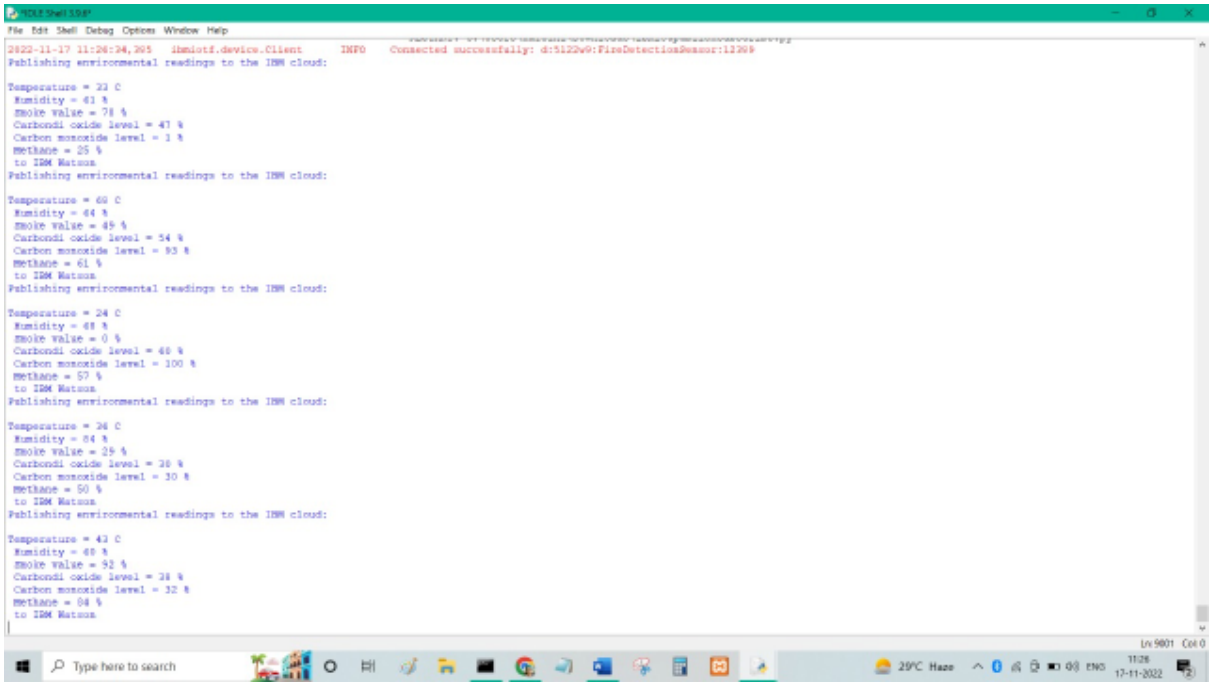
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Cloud software	USN-1	Ibm cloud creation, creating IBM Iot Watson, Installing Node-red	20	High	Lakshmi Sree, Harsini, Madhumitha Bhagyalakshmi,
Sprint-2	Simulation Software	USN-2	Device creation in IBM Iot watson, creating simulation for that device type and generating	20	High	Lakshmi Sree, Harsini, Madhumitha Bhagyalakshmi,

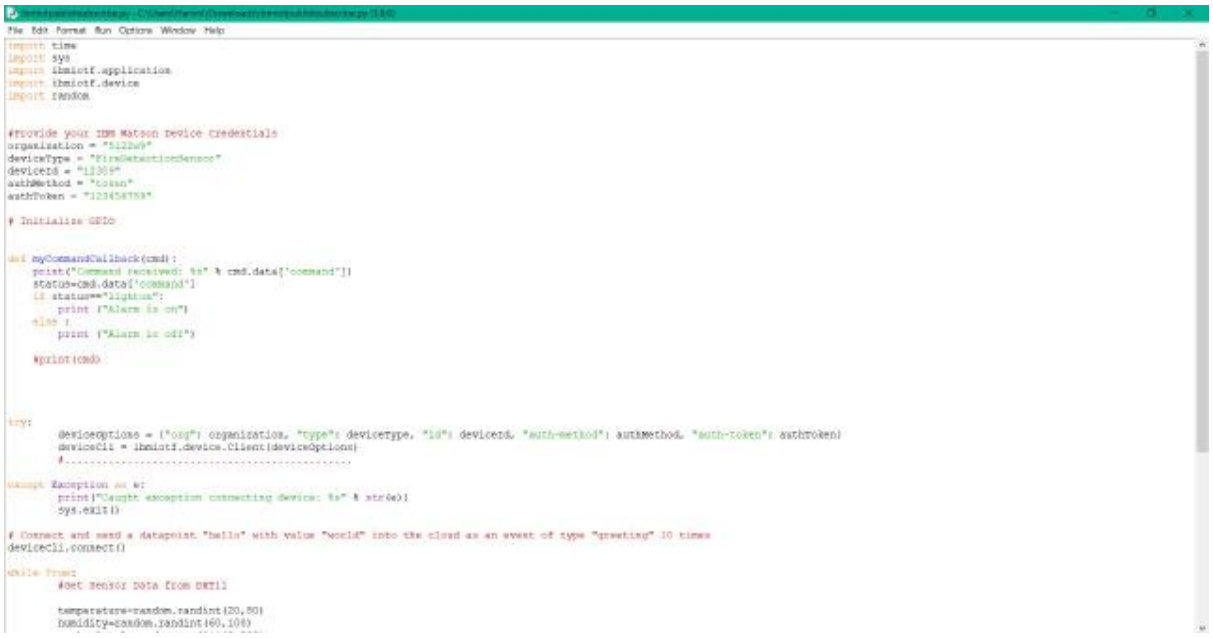
			data, creating flows in Node-red that collects data form IBM.			
Sprint-3	Node-red connection, MIT App Inventor	USN-3	Developing a web application using Node-red service, Developing mobile application for Industry specific Intelligent Fire Management system, Developing python script for publishing values to IBM Watson Iot platform	20	High	Lakshmi Sree, Harsini, Madhumitha Bhagyalakshmi,
Sprint-4	Connecting	USN-4	CONNECTING NODE-RED UI, IBM WATSON IOT PLATFORM, MOBILE APP – 'INDUSTRY SPECIFIC INTELLIGENT FIRE MANAGEMENT SYSTEM' AND PYTHON SCRIPT.	10	High	Lakshmi Sree, Harsini, Madhumitha Bhagyalakshmi,

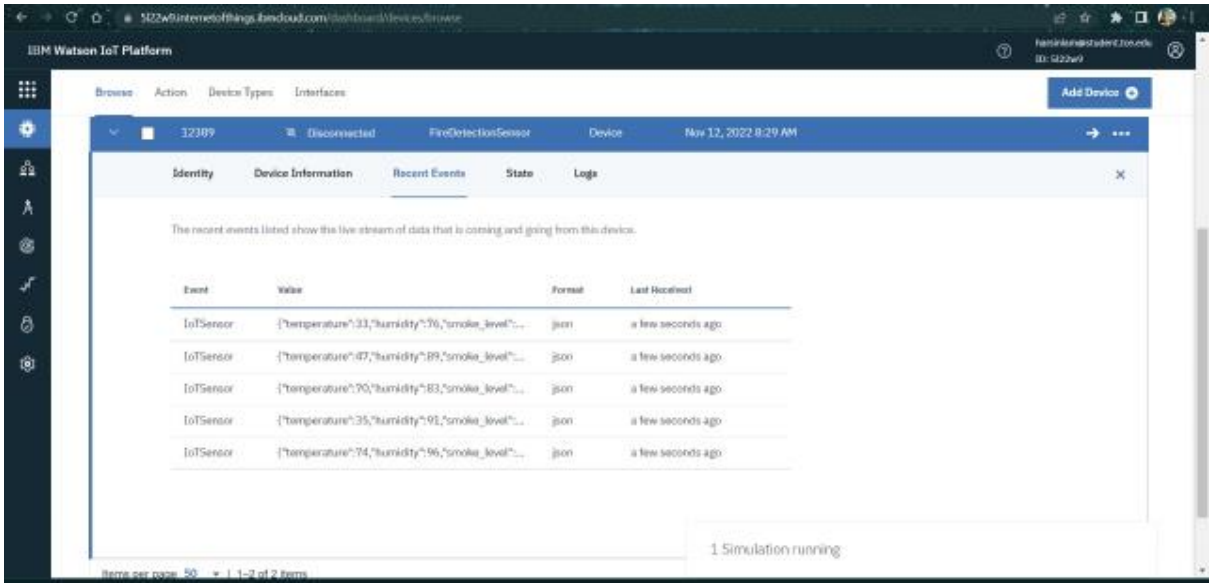
7. CODING AND SOLUTIONING:

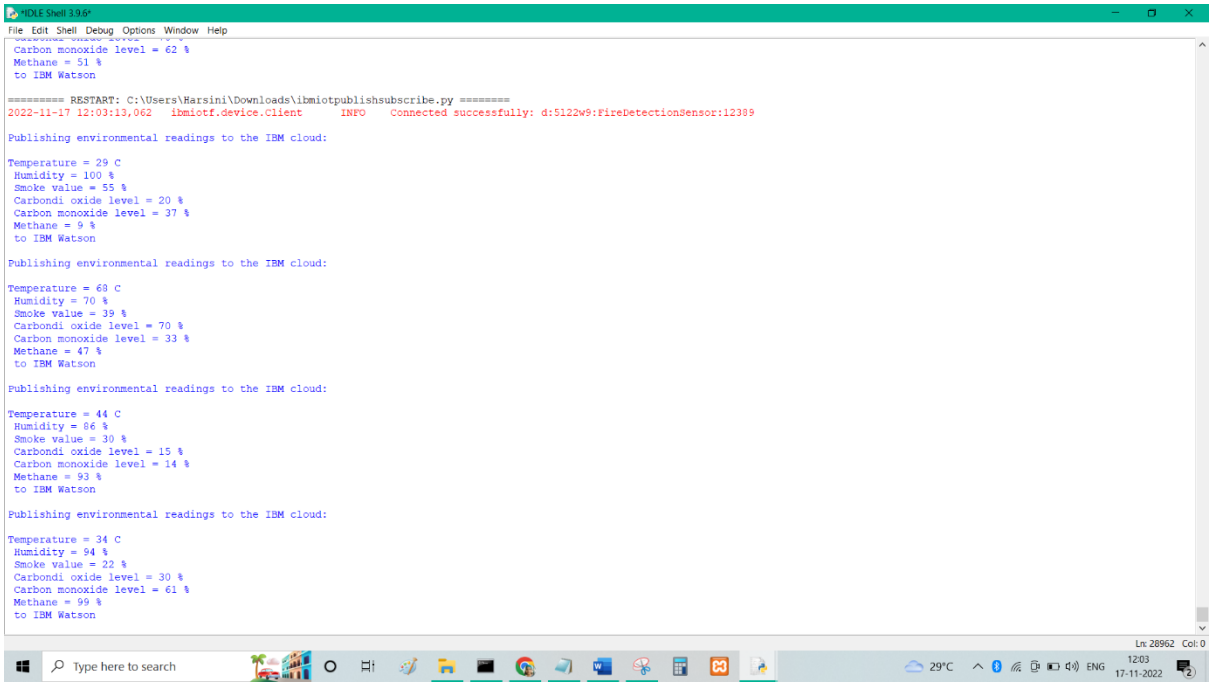
CONNECTING NODE-RED UI, IBM WATSON IOT PLATFORM, MOBILE APP – 'INDUSTRY SPECIFIC INTELLIGENT FIRE MANAGEMENT SYSTEM' AND PYTHON SCRIPT.

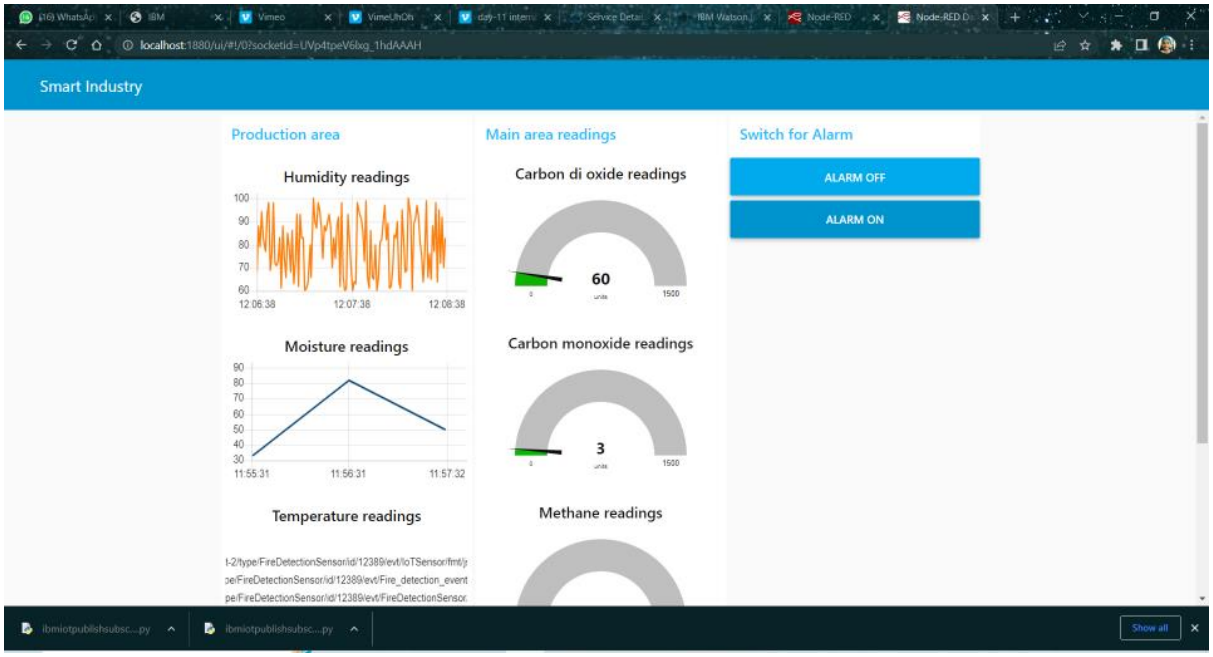
7.1 FEATURE 1:

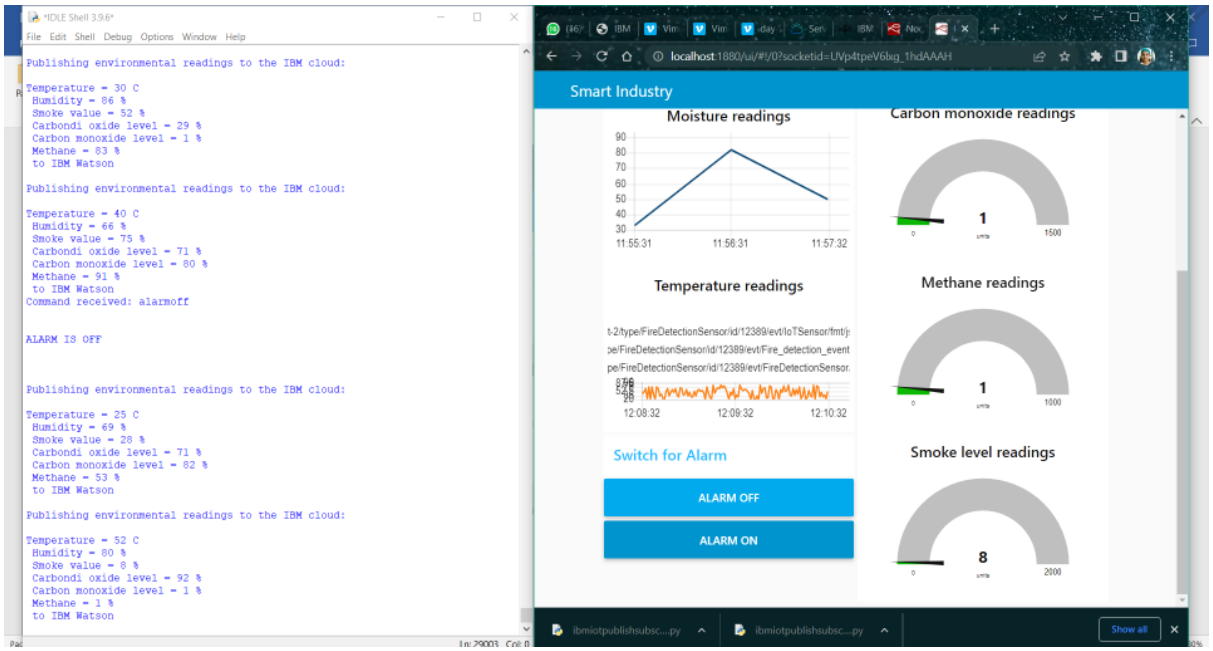

IDLE Shell 3.9.0
File Edit Shell Debug Options Window Help
2022-11-17 11:26:34,265 ibmiotf.device.Client INFO Connected successfully: d:5122v9:FireDetectionSensor:12309
Publishing environmental readings to the IBM cloud:
Temperature = 22 C
Humidity = 41 %
Smoke value = 78 %
Carbondi oxide level = 47 %
Carbon monoxide level = 1 %
Methane = 25 %
to IBM Watson
Publishing environmental readings to the IBM cloud:
Temperature = 60 C
Humidity = 44 %
Smoke value = 45 %
Carbondi oxide level = 54 %
Carbon monoxide level = 93 %
Methane = 61 %
to IBM Watson
Publishing environmental readings to the IBM cloud:
Temperature = 24 C
Humidity = 48 %
Smoke value = 0 %
Carbondi oxide level = 46 %
Carbon monoxide level = 100 %
Methane = 57 %
to IBM Watson
Publishing environmental readings to the IBM cloud:
Temperature = 26 C
Humidity = 54 %
Smoke value = 29 %
Carbondi oxide level = 38 %
Carbon monoxide level = 30 %
Methane = 50 %
to IBM Watson
Publishing environmental readings to the IBM cloud:
Temperature = 42 C
Humidity = 48 %
Smoke value = 92 %
Carbondi oxide level = 28 %
Carbon monoxide level = 32 %
Methane = 04 %
to IBM Watson
Ln: 9801 Col: 0
Type here to search
29°C Haze
ENG
11:28
17-11-2022


File Edit Format Run Options Window Help
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
#Provide your IBM Watson Device Credentials
organization = "5122v9"
deviceType = "FireDetectionSensor"
deviceId = "12309"
authMethod = "token"
authToken = "12345678"
# Initialize GPIO
def myCommandCallback(cmd):
print("Command received: %s" % cmd.data['command'])
status=cmd.data['command']
if status=="lighton":
print ("Alarm is on")
else :
print ("Alarm is off")
#print(cmd)
try:
deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method": authMethod, "auth-token": authToken}
deviceCli = ibmiotf.device.Client(deviceOptions)
#..
except Exception as e:
print("Caught exception connecting device: %s" % str(e))
sys.exit()
# Connect and send a datapoint "hello" with value "world" into the cloud as an event of type "greeting" 10 times
deviceCli.connect()
while True:
#Get Sensor data from DHT11
temperature=random.randint(20,80)
humidity=random.randint(60,100)

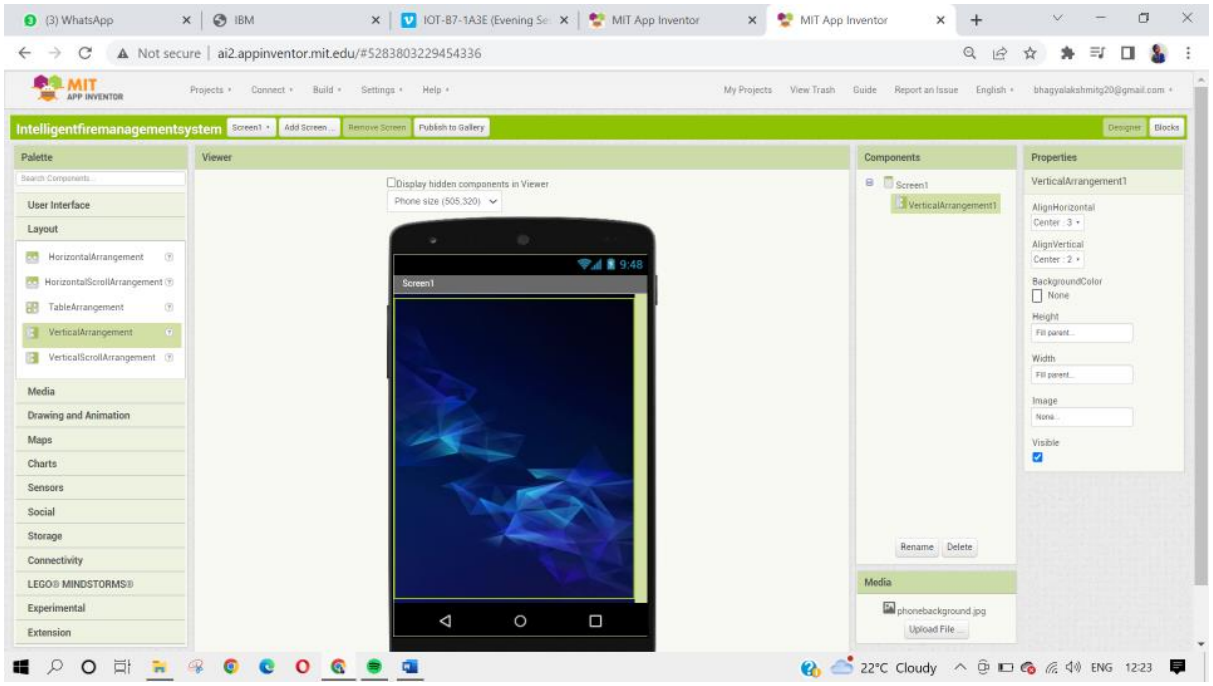

IBM Watson IoT Platform
Browse
Action
Device Types
Interfaces
Add Device
12389
Disconnected
FireDetectionSensor
Device
Nov 12, 2022 8:29 AM
Identity
Device Information
Recent Events
State
Logs
The recent events listed show the live stream of data that is coming and going from this device.
Event
Value
Format
Last Received
IoTSensor
json
a few seconds ago
1 Simulation running

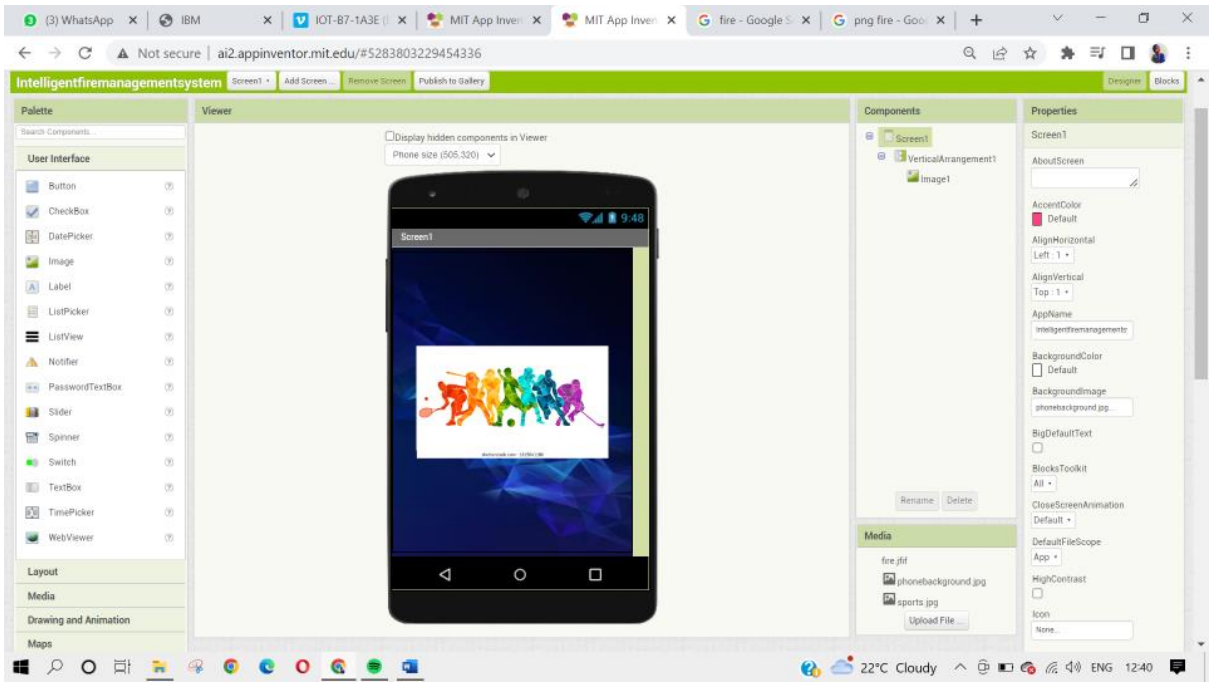

IDLE Shell 3.9.6
File Edit Shell Debug Options Window Help
Carbon monoxide level = 62 %
Methane = 51 %
to IBM Watson
========== RESTART: C:\Users\Harsini\Downloads\ibmiotpublishsubscribe.py ========
2022-11-17 12:03:13,062 ibmiotf.device.Client INFO Connected successfully: d:5122w9:FireDetectionSensor:12389
Publishing environmental readings to the IBM cloud:
Temperature = 29 C
Humidity = 100 %
Smoke value = 55 %
Carbondi oxide level = 20 %
Carbon monoxide level = 37 %
Methane = 9 %
to IBM Watson
Publishing environmental readings to the IBM cloud:
Temperature = 68 C
Humidity = 70 %
Smoke value = 39 %
Carbondi oxide level = 70 %
Carbon monoxide level = 33 %
Methane = 47 %
to IBM Watson
Publishing environmental readings to the IBM cloud:
Temperature = 44 C
Humidity = 86 %
Smoke value = 30 %
Carbondi oxide level = 15 %
Carbon monoxide level = 14 %
Methane = 93 %
to IBM Watson
Publishing environmental readings to the IBM cloud:
Temperature = 34 C
Humidity = 94 %
Smoke value = 22 %
Carbondi oxide level = 30 %
Carbon monoxide level = 61 %
Methane = 99 %
to IBM Watson
Ln: 28962 Col: 0
Type here to search
29°C
ENG
12:03
17-11-2022

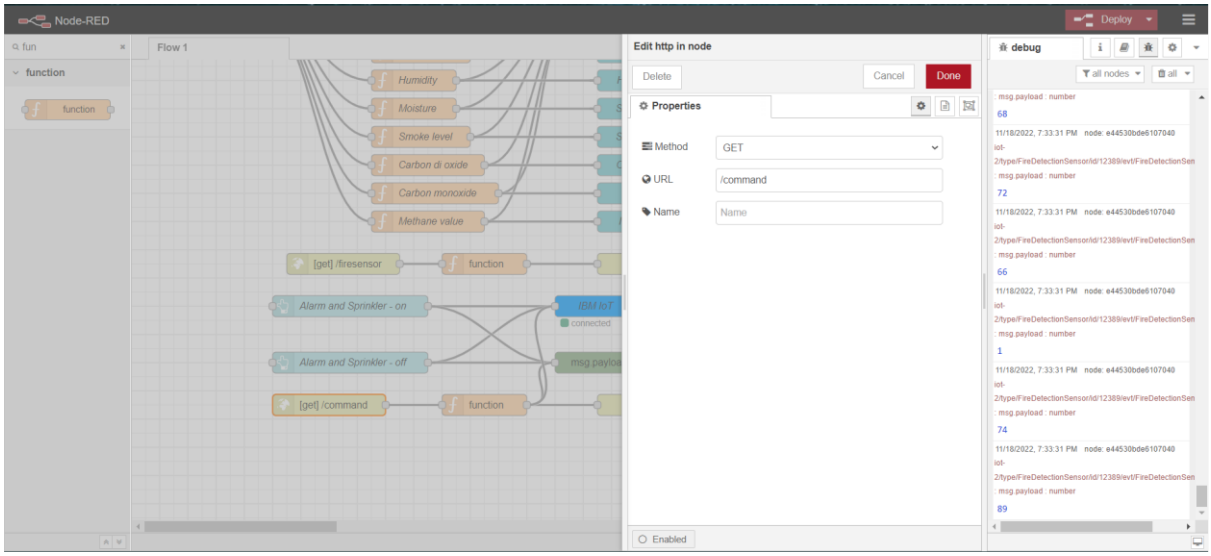

Smart Industry
Production area
Humidity readings
Moisture readings
Temperature readings
Main area readings
Carbon di oxide readings
60
Carbon monoxide readings
3
Methane readings
Switch for Alarm
ALARM OFF
ALARM ON

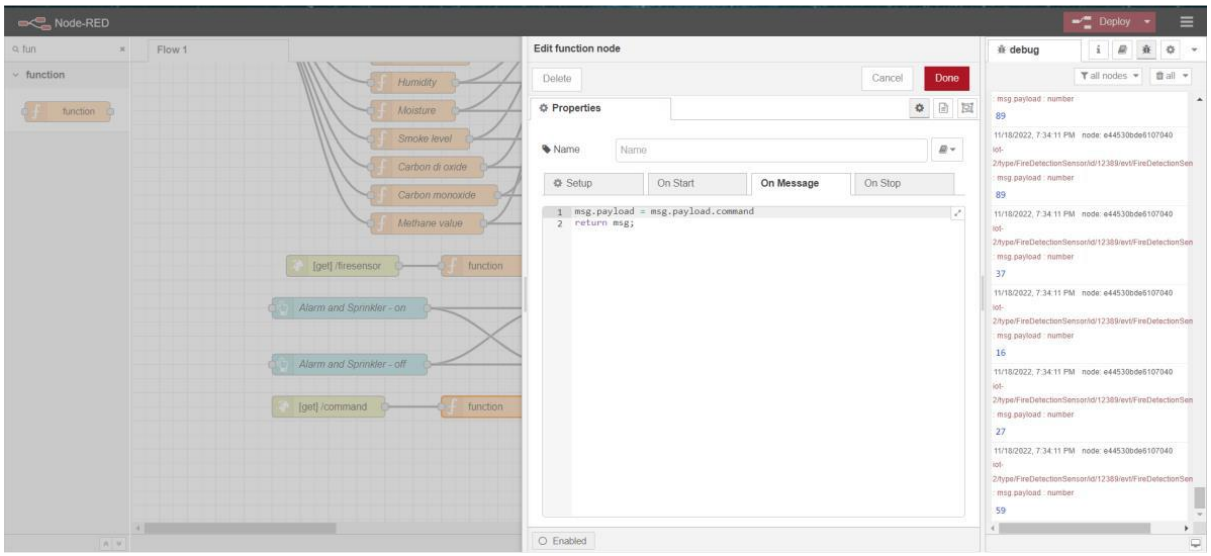

Publishing environmental readings to the IBM cloud:
Temperature = 30 C
Humidity = 86 %
Smoke value = 52 %
Carbondi oxide level = 29 %
Carbon monoxide level = 1 %
Methane = 83 %
to IBM Watson
Publishing environmental readings to the IBM cloud:
Temperature = 40 C
Humidity = 66 %
Smoke value = 75 %
Carbondi oxide level = 71 %
Carbon monoxide level = 80 %
Methane = 91 %
to IBM Watson
Command received: alarmoff
ALARM IS OFF
Publishing environmental readings to the IBM cloud:
Temperature = 25 C
Humidity = 69 %
Smoke value = 28 %
Carbondi oxide level = 71 %
Carbon monoxide level = 82 %
Methane = 53 %
to IBM Watson
Publishing environmental readings to the IBM cloud:
Temperature = 52 C
Humidity = 80 %
Smoke value = 8 %
Carbondi oxide level = 92 %
Carbon monoxide level = 1 %
Methane = 1 %
to IBM Watson
Smart Industry
Moisture readings
Carbon monoxide readings
Temperature readings
Methane readings
Switch for Alarm
ALARM OFF
ALARM ON
Smoke level readings

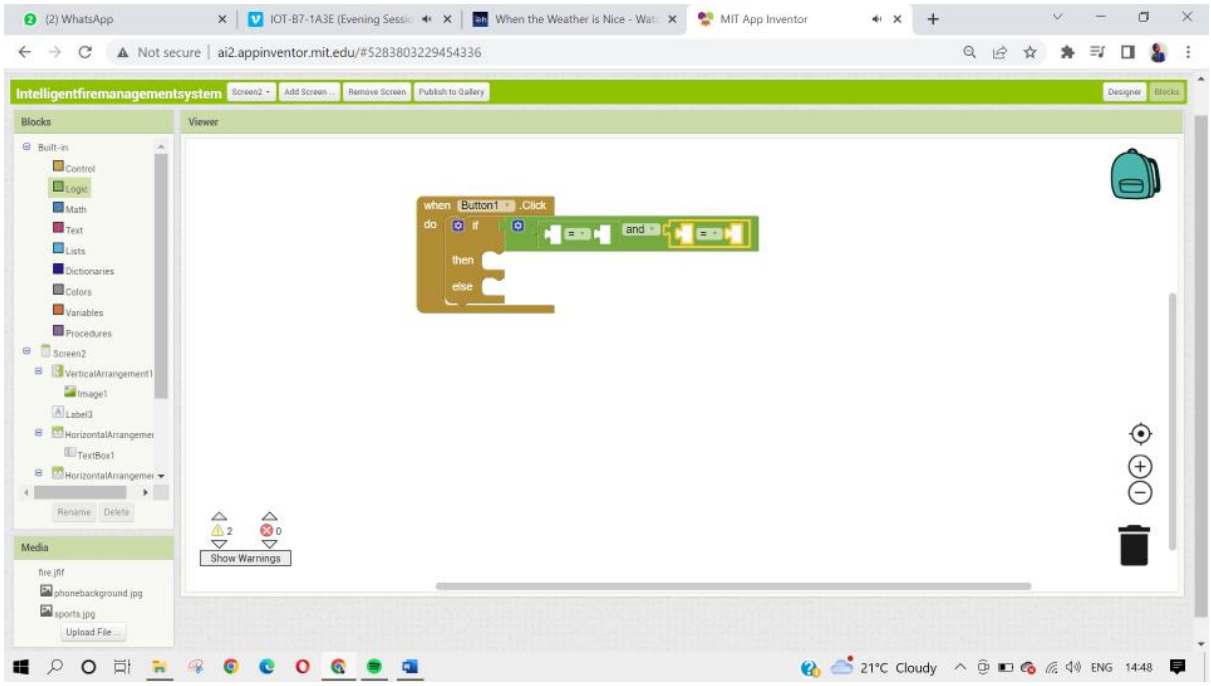
7.2 FEATURE 2:

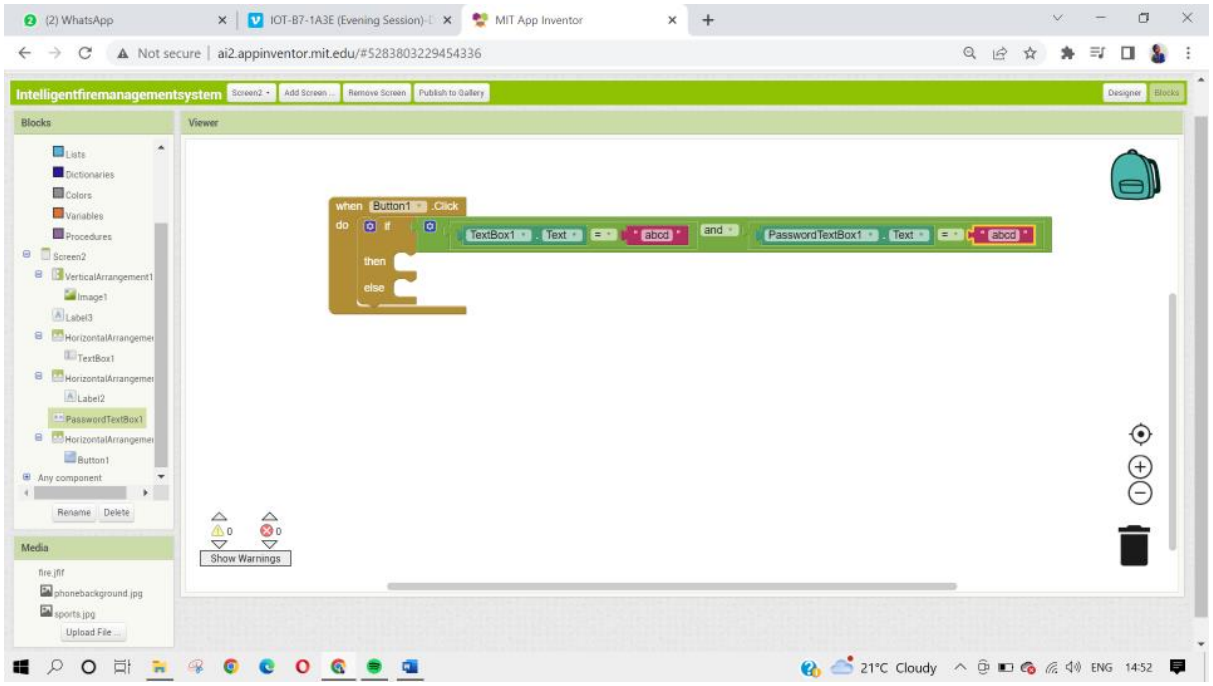

(3) WhatsApp
IBM
IOT-B7-1A3E (Evening Se
MIT App Inventor
MIT App Inventor
Not secure | ai2.appinventor.mit.edu/#5283803229454336
MIT APP INVENTOR
Projects
Connect
Build
Settings
Help
My Projects
View Trash
Guide
Report an Issue
English
bhagyalakshmitg20@gmail.com
Intelligentfiremanagementsystem
Screen1
Add Screen ...
Remove Screen
Publish to Gallery
Designer
Blocks
Palette
Search Components...
User Interface
Layout
HorizontalArrangement
HorizontalScrollArrangement
TableArrangement
VerticalArrangement
VerticalScrollArrangement
Media
Drawing and Animation
Maps
Charts
Sensors
Social
Storage
Connectivity
LEGO® MINDSTORMS®
Experimental
Extension
Viewer
Display hidden components in Viewer
Phone size (505,320)
9:48
Screen1
Components
Screen1
VerticalArrangement1
Rename
Delete
Media
phonebackground.jpg
Upload File ...
Properties
VerticalArrangement1
AlignHorizontal
Center : 3
AlignVertical
Center : 2
BackgroundColor
None
Height
Fill parent...
Width
Fill parent...
Image
None...
Visible
22°C Cloudy
ENG
12:23

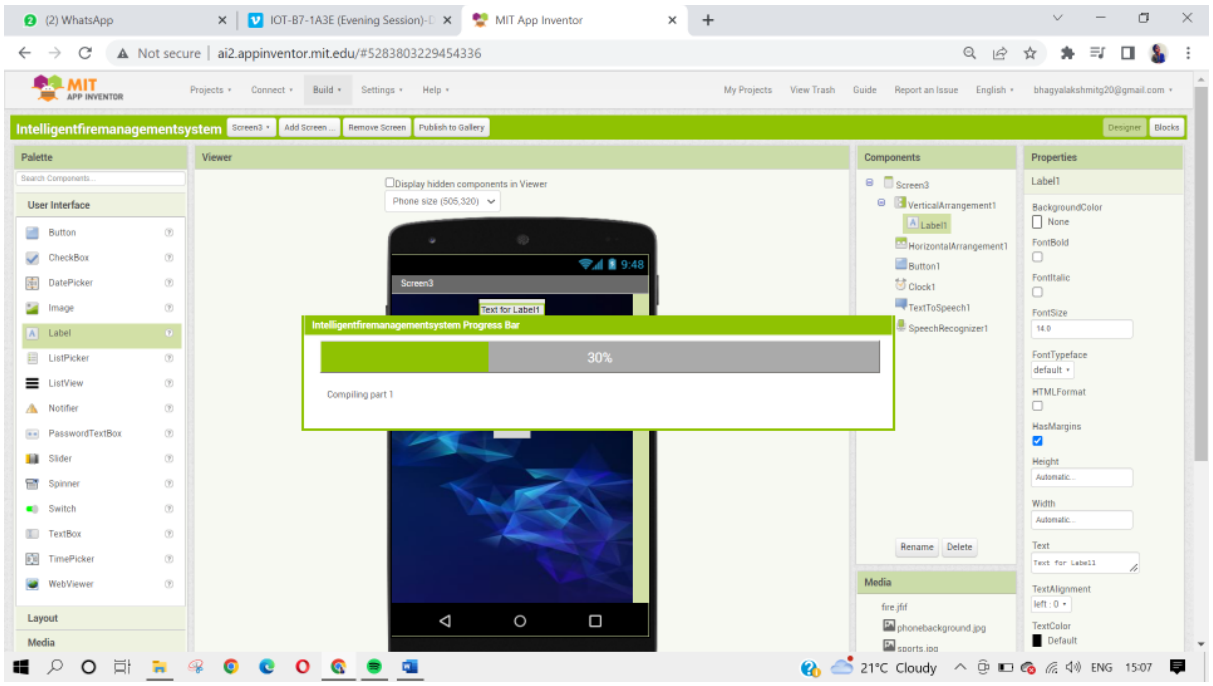

(3) WhatsApp
IBM
IOT-B7-1A3E (
MIT App Inven
MIT App Inven
fire - Google S
png fire - Goo
Not secure | ai2.appinventor.mit.edu/#5283803229454336
Intelligentfiremanagementsystem
Screen1
Add Screen ...
Remove Screen
Publish to Gallery
Designer
Blocks
Palette
Search Components...
User Interface
Button
CheckBox
DatePicker
Image
Label
ListPicker
ListView
Notifier
PasswordTextBox
Slider
Spinner
Switch
TextBox
TimePicker
WebViewer
Layout
Media
Drawing and Animation
Maps
Viewer
Display hidden components in Viewer
Phone size (505,320)
9:48
Screen1
Components
Screen1
VerticalArrangement1
Image1
Rename
Delete
Media
fire.jfif
phonebackground.jpg
sports.jpg
Upload File ...
Properties
Screen1
AboutScreen
AccentColor
Default
AlignHorizontal
Left : 1
AlignVertical
Top : 1
AppName
intelligentfiremanagementsy
BackgroundColor
Default
BackgroundImage
phonebackground.jpg...
BigDefaultText
BlocksToolkit
All
CloseScreenAnimation
Default
DefaultFileScope
App
HighContrast
Icon
None...
22°C Cloudy
ENG
12:40


Node-RED
Deploy
Flow 1
Humidity
Moisture
Smoke level
Carbon di oxide
Carbon monoxide
Methane value
[get] /firesensor
function
Alarm and Sprinkler - on
Alarm and Sprinkler - off
[get] /command
IBM IoT
connected
Edit http in node
Delete
Cancel
Done
Properties
Method
GET
URL
/command
Name
Enabled
debug
all nodes
68
72
66
1
74
89


Node-RED
Deploy
Flow 1
Edit function node
Delete
Cancel
Done
Properties
Name
Setup
On Start
On Message
On Stop
1 msg.payload = msg.payload.command
2 return msg;
Enabled
debug
89
37
16
27
59


(2) WhatsApp
IOT-B7-1A3E (Evening Sessio
When the Weather is Nice - Wat
MIT App Inventor
Not secure | ai2.appinventor.mit.edu/#5283803229454336
Intelligentfiremanagementsystem
Screen2
Add Screen ...
Remove Screen
Publish to Gallery
Designer
Blocks
Viewer
Built-in
Control
Logic
Math
Text
Lists
Dictionaries
Colors
Variables
Procedures
Screen2
VerticalArrangement1
Image1
Label3
TextBox1
Rename
Delete
Media
fire.jfif
phonebackground.jpg
sports.jpg
Upload File ...
when Button1 .Click
do if and
then
else
Show Warnings
21°C Cloudy
ENG
14:48

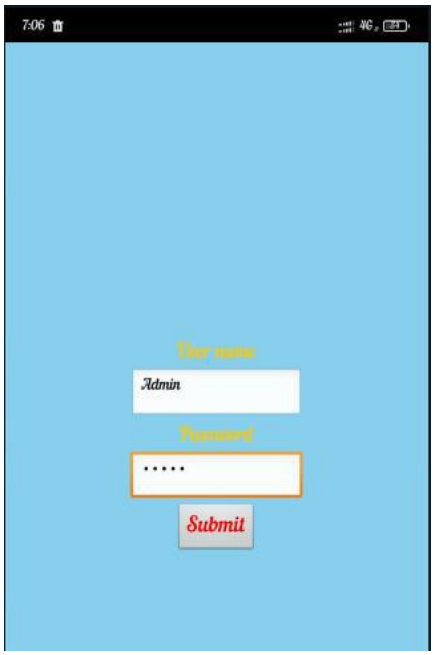




8. TESTING:

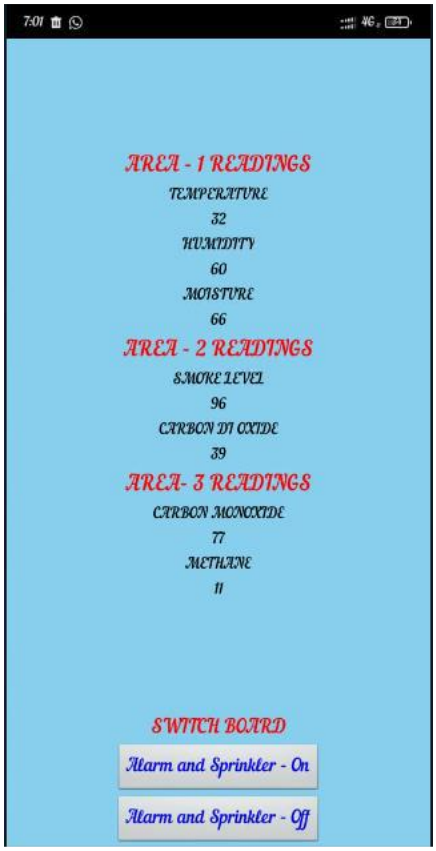
MIT APP
Home screen

Log in Page:

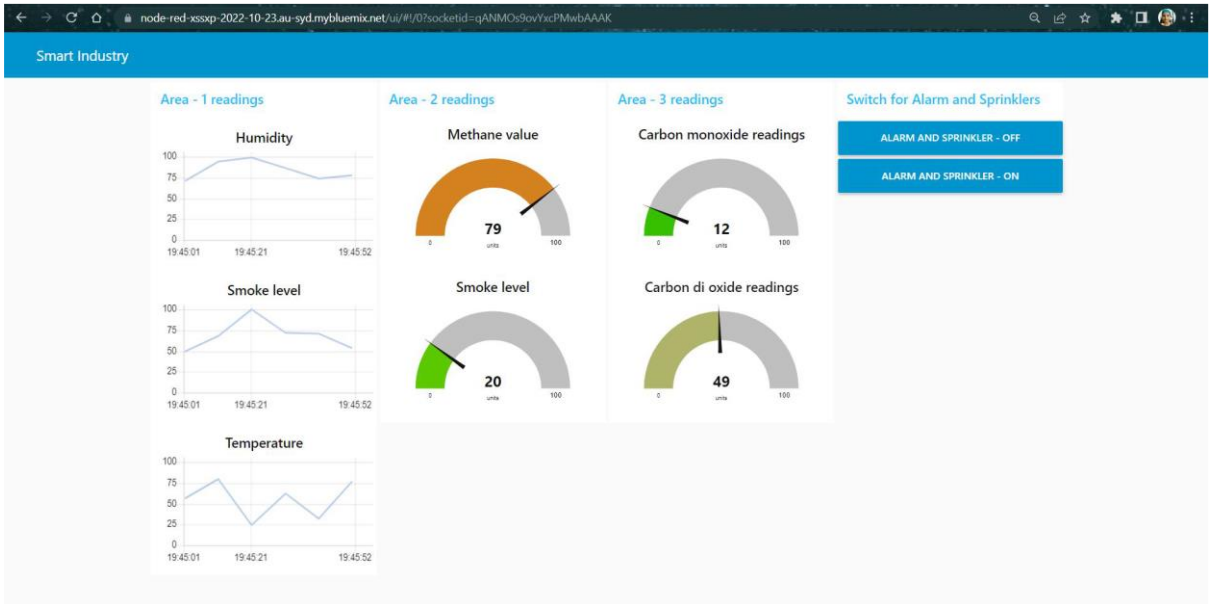


Main screen:

Industry environmental readings generated from IBM cloud displayed in both Node-red and MIT app



NODE-RED UI



9. RESULTS:

Thus, we have achieved our goal by creating and mobile application.

- Sending random fire and temperature values will be sent to the IBM IoT platform
- Sensors values can be viewed in the Web Application
- Notifies the admin the random values cross the threshold value.
- All these have been done.

10. ADVANTAGES:

1. To ensure the proper response to any potential hazards, an alert message will be sent to the industry management.

2.The mobile application is designed in an user-friendly manner.

3. Connecting this application to the actual sensors in the industries, give the accurate sensor readings.

11. CONCLUSION:

Thus, our proposed system here aims at helping out the industries in the event of any fire outbreak by alerting the industry workers by a mobile application. By using this application, the industry management can continuously monitor the temperature, smoke levels in the environment. This model is made user friendly so anybody can view and maintain his/her account. This application will help many industries to progress form traditional to user-friendly frameworks.

12. FUTURE SCOPE:

In the future, this mobile application can be made available to the main users. Based on the prevailing condition they will be able to access the actuators with the help of the buttons set in the app. Now, that the randomly generating the values are being displayed in the application, in the future, the values generated by the actual sensors deployed in the industries may be displayed in the application. Therefore, based on the displayed values the environment can be monitored and if there is any hazardous situation, automatic turn of alarms and sprinklers can also be done.

13. APPENDIX:

SOURCE CODE:

```
import time

import sys

import ibmiotf.application

import ibmiotf.device

import random
```

```
#Provide your IBM Watson Device Credentials
organization = "5l22w9"
deviceType = "FireDetectionSensor"
deviceId = "12389"
authMethod = "token"
authToken = "123456789"

# Initialize GPIO



def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="alarmon":
        print ("\n\nALARM IS ON\n\n")
    else :
        print ("\n\nALARM IS OFF\n\n")

    #print(cmd)




try:
        deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-
method": authMethod, "auth-token": authToken}
        deviceCli = ibmiotf.device.Client(deviceOptions)
        #.....

except Exception as e:
        print("Caught exception connecting device: %s" % str(e))
```

```
        sys.exit()


# Connect and send a datapoint "hello" with value "world" into the cloud as an event of type "greeting" 10 times
deviceCli.connect()


while True:
    #Get Sensor Data from DHT11


    temperature=random.randint(20,80)
    humidity=random.randint(60,100)
    smoke_level=random.randint(0,100)
    co2_level=random.randint(0,100)
    co_level=random.randint(0,100)
    methane=random.randint(0,100)


    data = { 'temperature' : temperature, 'humidity': humidity, 'smoke_level' : smoke_level, 'co2_level' : co2_level, 'co_level': co_level, 'methane': methane}
    #print data
    def myOnPublishCallback():
        print ("\nPublishing environmental readings to the IBM cloud:\n")
        print ("Temperature = %s C\n" % temperature, "Humidity = %s %%\n" % humidity,"Smoke value = %s %%\n" % smoke_level,"Carbondi oxide level = %s %%\n" % co2_level,"Carbon monoxide level = %s %%\n" % co_level,"Methane = %s %%\n" % methane, "to IBM Watson")


    success = deviceCli.publishEvent("FireDetectionSensor", "json", data, qos=0, on_publish=myOnPublishCallback)
    if not success:
        print("Not connected to IoT Watson")
    time.sleep(1)
```

```
    deviceCli.commandCallback = myCommandCallback

# Disconnect the device and application from the cloud
deviceCli.disconnect()
```

GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-26965-1660041740

PROJECT DEMO LINK:

https://drive.google.com/drive/folders/1fI30EM7jHUGfha0nTif7wjf33lDZRX_j