INDUSTRY SPECIFIC INTELLIGENT FIRE MANAGEMENT SYSTEM

NALAIYA THIRAN PROJECT BASED LEARNING

On

PROFESSIONAL READINESSFOR INNOVATION, EMPLOYABILITY AND ENTREPRENEURSHIP

A PROJECT REPORT

HASAN HARUN	(REG NO:720719104069)
HIRUTHIK	(REG NO: 720719104071)
KALIRAJAN	(REG NO:720719104075)
KARUPPASAMY	(REG NO: 720719104080)

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

HINDUSTHAN COLLEGEOF ENGINEERING AND TECHOLOGY

COIMBATORE – 641 032

NOVEMBER2022

HINDUSTHAN COLLEGE OF ENGINEERING AND TECHNOLOGY

Approved by AICTE,New Delhi, Accredited with 'A' Grade byNAAC**(An Autonomous Institution, Affiliated to Anna University, Chennai)**Valley Campus, PollachiHighway, Coimbatore – 641 032

INTERNAL MENTOR

MRS.B.REENA, M.E., (Ph.D).,

Assistant Professor,

Department of Computer Science and Engineering

Hindusthan Collegeof Engineering

and Technology,Coimbatore-641

032

INDUSTRY MENTOR

Santoshi (IBM)

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

1.1 Project Overview

In this project we will be developing a IoT based device to help in the fire management in the industry. Our device will be using various sensorslike flame sensor,temperature sensor and gas sensor.The project will be highlyefficient in extinguishing the fire in the industry and reduce the damages that is caused by fire.

1.2 Purpose

The main aim of the projectis to provide safe and efficient industry-specific intelligent fire management system. This project will be used in fire prevention, temperature monitoring, gas and flame management.

Chapter 2
LITERATURE

2.1 Existing Problem

The existing mechanismin the fire management systemis ineffiective and expensive. It also requires human interaction to take decision which cannot be feasible at the time of emergency. We will need a spontaneous device which can prevent fire damages before it happens or take appropriate decisions when an emergency occurs

2.2 Reference

1. Stipanicev D., Vuko T., Krstinic D., Stula M., Bodrozic L., Forest Fire Protection by Advanced Video Detection System- Croatian Experiences, Split, Croatia, 2006
2. Losso A., Corgnati L., Perona G., Early Forest Fire Detection:Smoke Identification throughinnovative Image Processing using Commercial Sensors,Environment Including Global Change,Palermo, Italy, 2009
3. KovacsR., Kiss B., Nagy A., Vamos R., Early Forest Fire DetectionSystemFor VegetableFire in the Aggtelek NationalPark Budpest, Hungary,2004
4. Kelha V., RausteY., Buongiorno A., Forest Fire Detection by Satellites forFire Control, European Space Agency, Finland,2000
5. Manyangadze T., Forest Fire Detection for Near Real Time MonitoringusingGeostationary Satellite, International Institute for Geo-information Scienceand Earth Observation, Enschede, Netherland, 2009

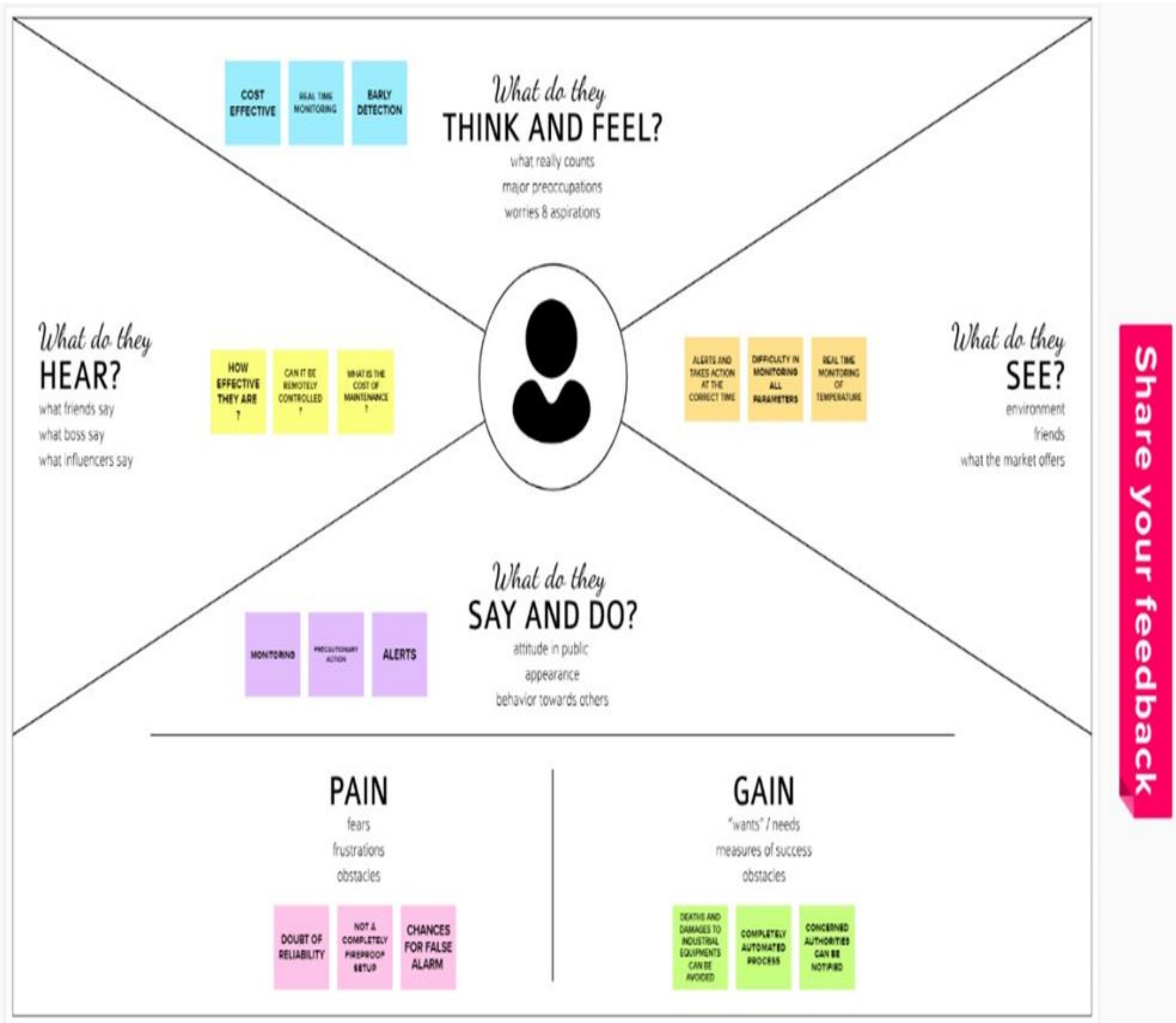
2.3 Problem StatementDefinition

The fire management system in housesand industries are not very reliable,efficient, cost effective and does not have any advance processing and does not have any features like automatic alert system for admin and authorities and in many buildings there are using older fire safety system that doesn’t can even activate the sprinkler system and all of they don't communicate with each other properly to prevent false alarm also monitor the entire system using a applications .

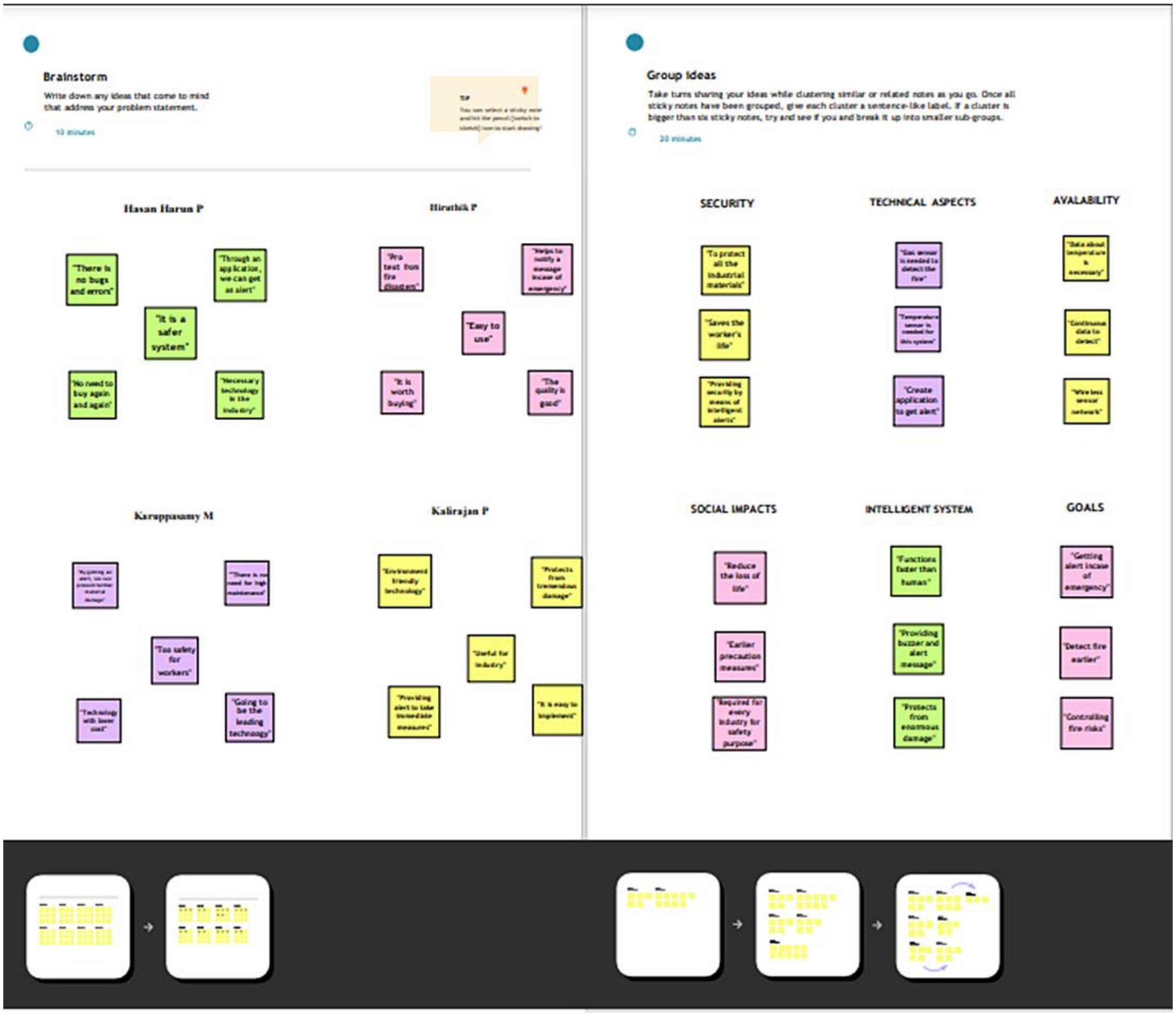
Chapter 3

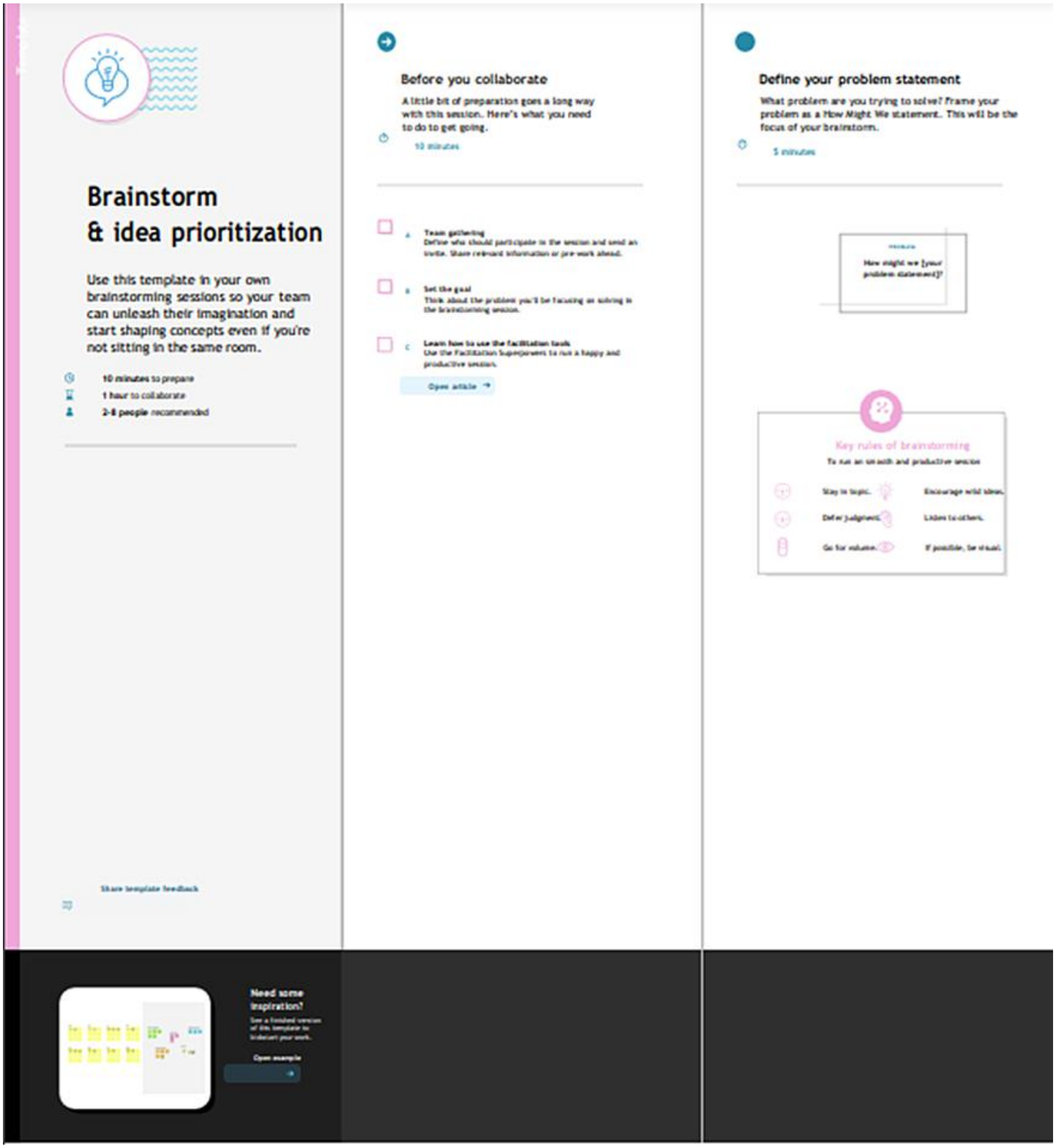
IDEATION & PROPOSED SOLUTION

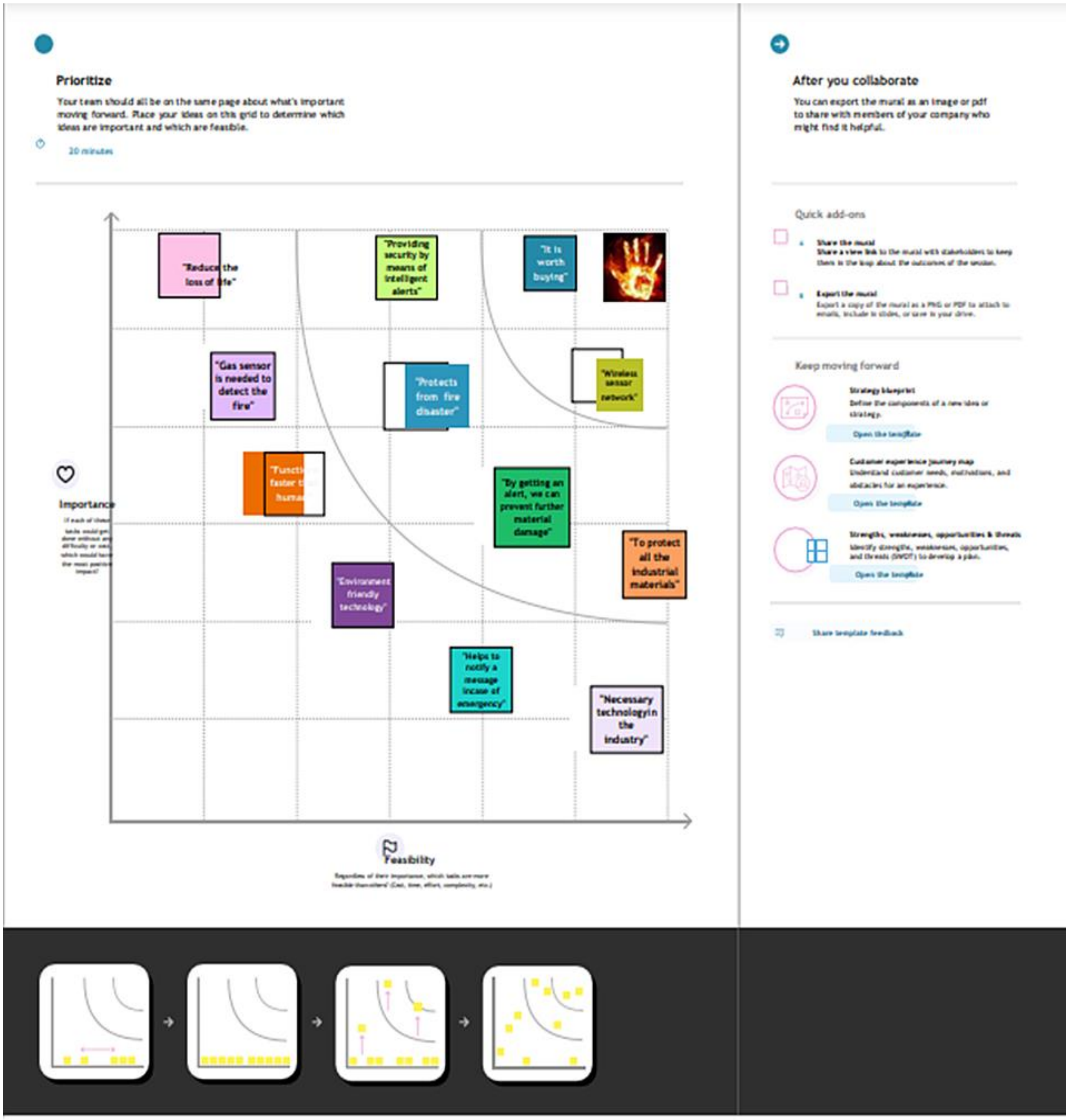
3.1 Empathy Mapping Canvas



3.2 Ideation & Brainstorming


Brainstorm
Write down any ideas that come to mind that address your problem statement.
10 minutes
Hasan Harun P
"There is no bugs and errors"
"Through an application, we can get an alert"
"It is a safer system"
"No need to buy again and again"
"Necessary technology in the industry"
Hiruthik P
"Pro tect from fire disaster"
"Helps to notify a message incase of emergency"
"Easy to use"
"It is worth buying"
"The quality is good"
Karuppasamy M
"There is no need for high maintenance"
"Too safety for workers"
"Technology with lower cost"
"Going to be the leading technology"
Kalirajan P
"Environment friendly technology"
"Protects from tremendous damage"
"Useful for industry"
"Providing alert to take immediate measures"
"It is easy to implement"
Group Ideas
Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.
20 minutes
SECURITY
"To protect all the industrial materials"
"Saves the worker's life"
"Providing security by means of intelligent alerts"
TECHNICAL ASPECTS
"Gas sensor is needed to detect the fire"
"Temperature sensor is needed for this system"
"Create application to get alert"
AVALABILITY
"Data about temperature is necessary"
"Continuous data to detect"
"Wireless sensor network"
SOCIAL IMPACTS
"Reduce the loss of life"
"Earlier precaution measures"
"Required for every industry for safety purpose"
INTELLIGENT SYSTEM
"Functions faster than human"
"Providing buzzer and alert message"
"Protects from enormous damage"
GOALS
"Getting alert incase of emergency"
"Detect fire earlier"
"Controlling fire risks"


Brainstorm
& idea prioritization
Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.
10 minutes to prepare
1 hour to collaborate
2-8 people recommended
Share template feedback
Need some inspiration?
See a finished version of this template to kickstart your work.
Open example
Before you collaborate
A little bit of preparation goes a long way with this session. Here's what you need to do to get going.
10 minutes
Team gathering
Set the goal
Learn how to use the facilitation tools
Open article
Define your problem statement
What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.
5 minutes
How might we [your problem statement]?
Key rules of brainstorming
Stay in topic.
Encourage wild ideas.
Defer judgment.
Listen to others.
Go for volume.
If possible, be visual.



3.3 Proposed Solution

S.No	Parameter	Description
1	Problem Statement (Problem tobe solved)	To improve the safety management system in industries. Improving the safety management systemagainst the fireincidents in industries.
2	Idea / Solution description	To implement the fire safetymanagement 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 trackingsystem
3	Novelty / Uniqueness	Usage of liquid Nitrogen. Liquid nitrogen will immediately vaporize causing a coolingeffect and makes the site deprived of oxygen. Using water sprinklers after liquid nitrogen can put off even intense fire effectively.

4	SocialImpact / Customer Satisfaction	1. Harmful gasses can be purified and released intoatmosphere 2. Causeof impact can be tracedfrom the sensordata that can be analyzed to prevent future accidents.
5	Business Model (Revenue Model)	1. This is used to calculate the probability of ignitionand spread of fires across a landscape. 2. This outcomeallows for a better understanding ofhow changes in one aspects of management can affect otheraspects of management
6	Scalabilityof theSolution	It is trying to execute this technique as we need to introduce an arduinogadget which wasmodified withan Arduino that takes received signals from sensors .Easy reliability and maintenance. Required low timefor maintenance. Cost is reasonable.

3.4 Proposed Solution Fit

Define CS, fit into CC

1. CUSTOMER SEGMENT(S)

CS

Who is your customer?

My customers are those who own organization and industries which can be easily subjected into fire accident.

Examples: oil Industries, cotton industries

6. CUSTOMER CONSTRAINTS

CC

What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.

Fire extinguishers hanging on the walls are weight and the industries or organization has less number of extinguishers so many of them could not use it when occurrence of fire.

5. AVAILABLE SOLUTIONS

AS

Which solutions are available to the customers when they face the problem or need to get the job done? what have they tried in the past?

In most places, fire extinguishers are hung on the walls for safety purposes, when there is an occurrence of fire, extinguishers are make used it by using man power and in some places ,an automatic fire detection system is also there.

Explore AS, differentiate

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

2. JOBS-TO-BE-DONE / PROBLEMS

J&P

Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides

If fire occur in any organization or industries this fire alarm detects the fire and indicates through the alarm and sprinkles the water to extinguish the fire.

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? i.e. customers have to do it because of the change in regulations.

In crackers industries there used chemicals for crackers which are easily inflammable so it can cause fire accident even if a tiny sparks occurs as it happened once in a sivakasi.. Likewise many other problems can also be the root cause of fire

7. BEHAVIOUR

BE

What does your customer do to address the problem and get the job done? i.e. directly related find the right solar panel installer, calculate usage and benefit; ,indirectly associated: customers spend free time on volunteering(i.e. Green peace)

Monitoring the fire with the help of the automatic fire detection system in the respective places to avoid fire accidents ,avoid using old machines and safety precautions are to be followed.

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

	3. TRIGGER What triggers customers toact7 Improving the fire management systemGiving awareness tO Employees	1. **CUSTOMER CONSTRAINTS** What constraints prevent your customers from taking action or limit their choices of **solutions7** a. All industry workers need to have smart devices(smart phones) and also must have a connec	1. **PROBLEM ROOT CAUSE** what is the real reason that this problem exists? What is the back story? a. Failure of Machinery and other equipment. b. Electrical short circuiti	

		ted network. Well monitoring system	ng or overloading c. Chemical Reaction/Runway Reaction.	
	10. YOUR SOLUTION Detecting the presence of fire and Alerting the industrial workers and its presence to thefire officials through the internet. and save the property and life of people. And Usage of automatic sprink1ers along with liquid nitrogen is able to put off alarge fire breakout. In case of toxic gas leakage,they are purified and released into the atmosphere.			

Chapter 4
REQUIREMENT ANALYSIS

a. Functional Requirement

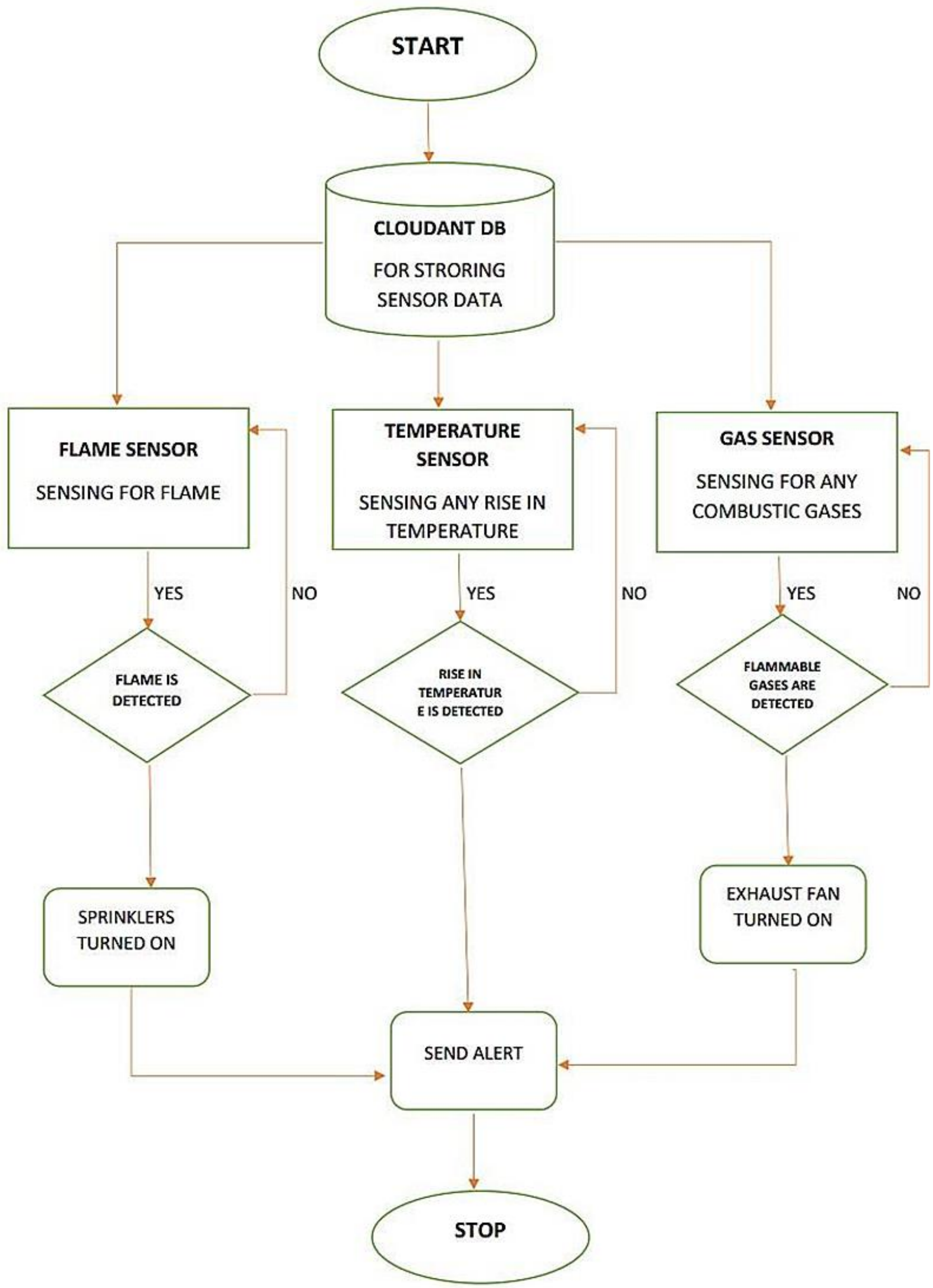
FR No.	Functional Requirement (Epic)	Sub Requirement(Story / Sub-Task)
FR-1	Sensing function	Fire breakout has to be sensed by smoke detectors. Gas leakage has to be sensed by gas sensors.
FR-2	Alerting function	Blaring of alarms
FR-3	Actuation function	Activation of sprinklers. Turning ON the exhaustFan.
FR-4	Notification	Sending SMS to the Authorities. Sending SMS with location to the firestation

b. Non-Functional Requirement

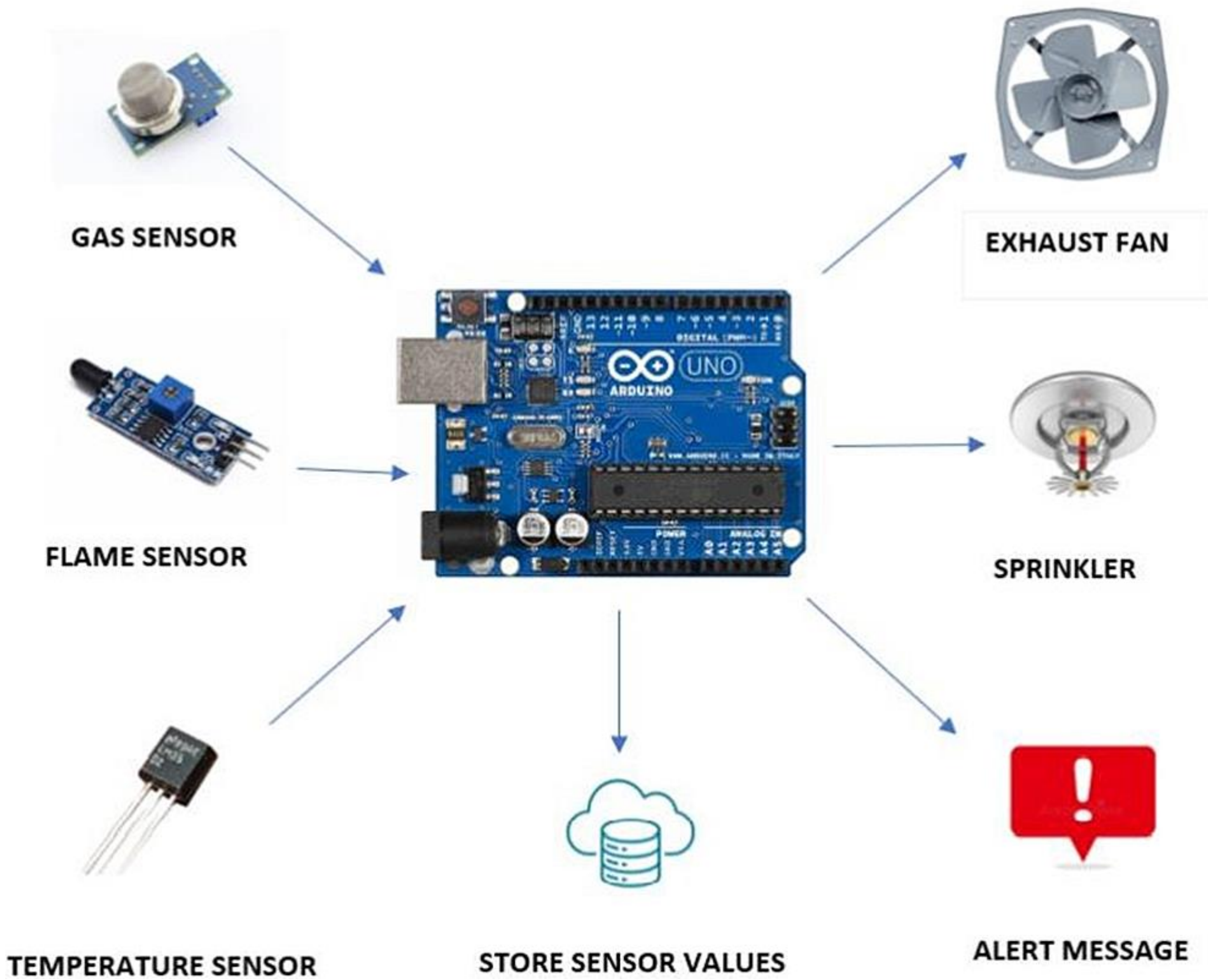
FR No	Non-Functional Requirement	Description
NFR-1	Usability	Ease of use and longevity ofthe system.
NFR-2	Security	Software remains secured inthe face of attacks.
NFR-3	Reliability	High accuracy.
NFR-4	Performance	Faster response.
NFR-5	Availability	Availability of the systems for institutions, restaurants and other publicplaces
NFR-6	Scalability	It accommodates easy modification for variousrequirements

Chapter 5
PROJECT DESIGN

5.1 Data Flow Diagram



a. **5.2 Solution & Technical Architecture**



5.3 User Stories

User Type	Functional Requirement(Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Industry)	Assembling	USN-1	As a user, I must position the sensors in the appropriate location.	I have access to my sensor triggers	High	Sprint-1
		USN-2	As a user, I must test my hardware to ensure that it is operational.	I can use the serial monitor to keep track of all sensor values.	High	Sprint-1
	User Registration	USN-3	As a user, I can create user accounts for the model's essential software.	I can sign up and utilize user Login to access the dashboard.	Medium	Sprint-2
		USN-4	As a user, I can verify that notifications and SMS are delivered properly	I can verify the alerts via Fast2SMS	High	Sprint-1
	Cloud Monitoring	USN-5	As a user, I can keep track on how long data is kept in IBM Cloudant	I can constantly monitor and obtain sensor data.	High	Sprint-1

Chapter 6
PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority
Sprint-1	Wokwi Simulation	USN-1	Making Hardware device or Using Worwi. Connect Temperature,Flame,Gas sensor to Arduino with python script	2	High
Sprint-2	IBM Cloud	USN-2	Create Device in the IBM Watson IOT Platform and link it to Noad-red	2	High
Sprint-3	Node Red service	USN-3	Create a dashboard and link the cloud platform	2	High
Sprint-4	MIT app inverter	USN-4	Link Device, IBM cloud and the developed application. Feed the data to sprinkler, exhaust fan.	2	High

TITLE	DESCRIPTION
	IDEATION PHASE
Literature Survey & Information Gathering	Literature survey on the selected project & gathering information by referring the, technical papers, research publications etc.
Prepare Empathy Map	Prepare Empathy Map Canvas to capture the user Pains & Gains, Prepare list of problem statements.
Problem Statement	List of problems in the project.
Brainstorm And Idea Prioritization	List them by organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility & importance.
	Project Design Phase - I
Proposed Solution	Prepare the proposed solution document, which includes then novelty, feasibility of idea, business model, social impact, scalability of solution, etc.
Problem Solution Fit	Prepare problem - solution fit document.
Solution Architecture	Prepare solution architecture document.

6.2 Sprint Delivery Schedule

Chapter 7
CODING AND SOLUTION

7.1 Feature 1

Generating Random Input and Send it to IBM Watson Cloud

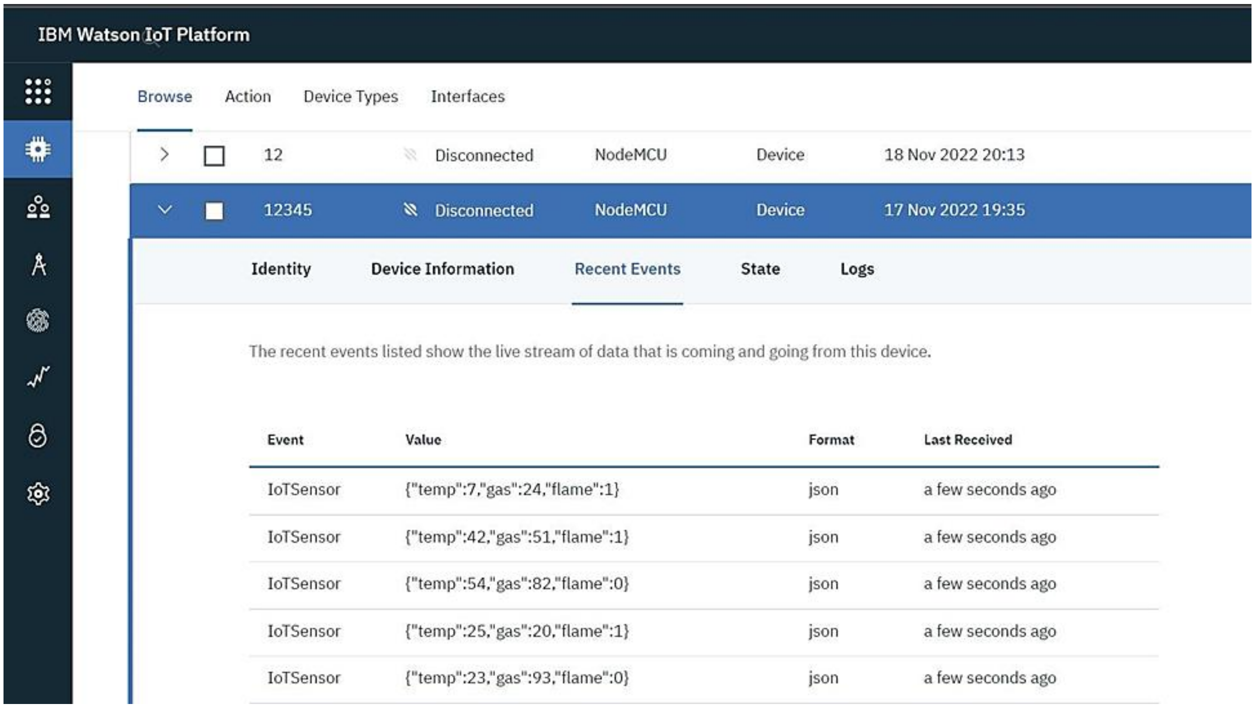
```
importtime
import sys
import ibmiotf.device
importibmiotf.application
import random




#Provide your IBM Watson Device Credentials

organization ="inbee2"
deviceType = "NodeMCU"
deviceId ="12345"
authMethod = "token"
authToken = "12345678"
# Initialize GPIO

def myCommandCallback1(cmd):
print("Commandreceived: %s" % cmd.data['command'])status =
 cmd.data['command']if status == "sprinkleron":
 print("sprinkler is on")
else:
        print("sprinkler is off ") print(cmd)


def myCommandCallback2(cmd):
print("Commandreceived: %s" % cmd.data['command'])status =
 cmd.data['command']if status == "fanon":
   print("fan is on") else:
 print("fan 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 exceptionconnecting device: %s"% str(e)) sys.exit()


# Connect and send a datapoint "hello" with value "world" into the cloud as an eventof type
"greeting" 10 timesdeviceCli.connect()


while True:
```

```
#Get Sensor Data from DHT11

temp=random.randint(0,70)
gas=random.randint(0,100)
flame=random.randint(0,1)

data = { 'temp' : temp, 'gas': gas, 'flame': flame }
#print data
def myOnPublishCallback():
        print ("Published Temperature = %s C" % temp, "Gas = %s %%"% gas, "flame = %s
        %%"
% flame,"to IBM Watson")

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

deviceCli.commandCallback1 = myCommandCallback1
deviceCli.commandCallback2 = myCommandCallback2

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

Python 3.7.4 Shell

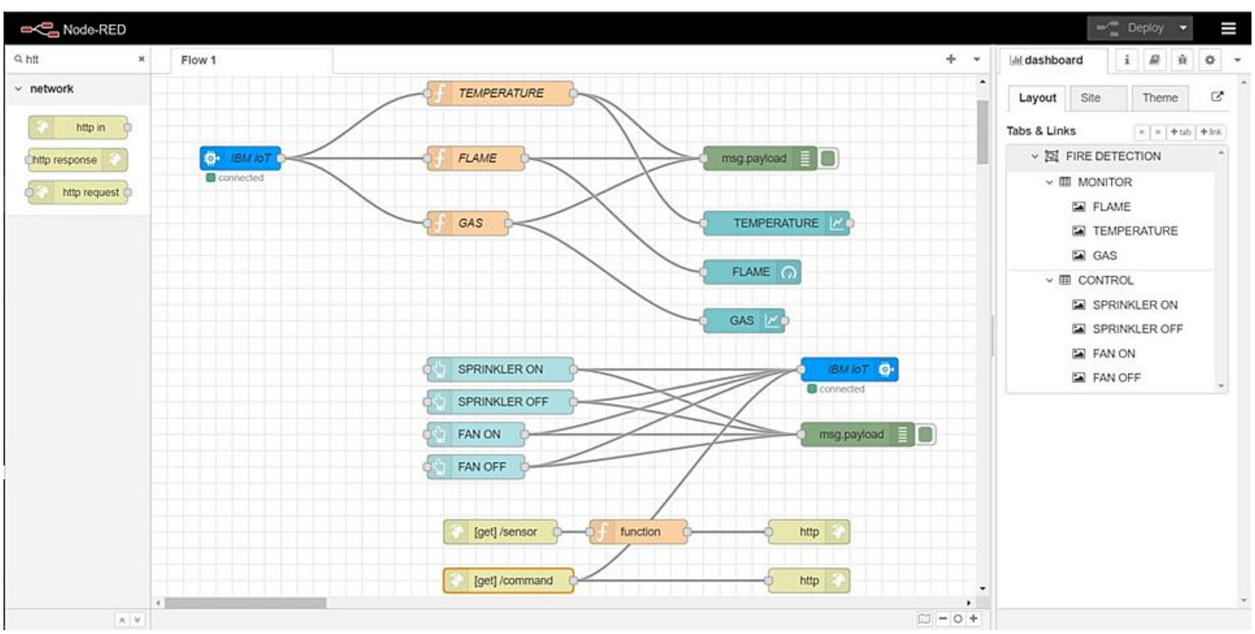
File Edit Shell Debug Options Window Help

```
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul  8 2019, 19:29:22) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========================== RESTART: E:\IBM\IBM.py ==========================
2022-11-19 22:38:24,864   ibmiotf.device.Client      INFO    Connected successfully: d:inbee2:NodeMCU:12345
Published Temperature = 34 C Gas = 0 % flame = 1 % to IBM Watson
Published Temperature = 34 C Gas = 0 % flame = 1 % to IBM Watson
Published Temperature = 34 C Gas = 0 % flame = 1 % to IBM Watson
Published Temperature = 0 C Gas = 77 % flame = 0 % to IBM Watson
Published Temperature = 33 C Gas = 84 % flame = 1 % to IBM Watson
Published Temperature = 29 C Gas = 27 % flame = 1 % to IBM Watson
Published Temperature = 7 C Gas = 1 % flame = 1 % to IBM Watson
Published Temperature = 33 C Gas = 99 % flame = 1 % to IBM Watson
Published Temperature = 5 C Gas = 60 % flame = 1 % to IBM Watson
Published Temperature = 32 C Gas = 25 % flame = 0 % to IBM Watson
Published Temperature = 28 C Gas = 78 % flame = 0 % to IBM Watson
Published Temperature = 23 C Gas = 3 % flame = 0 % to IBM Watson
Published Temperature = 64 C Gas = 70 % flame = 1 % to IBM Watson
Published Temperature = 22 C Gas = 63 % flame = 1 % to IBM Watson
Published Temperature = 17 C Gas = 28 % flame = 1 % to IBM Watson
Published Temperature = 31 C Gas = 52 % flame = 1 % to IBM Watson
Published Temperature = 19 C Gas = 42 % flame = 1 % to IBM Watson
Published Temperature = 10 C Gas = 87 % flame = 1 % to IBM Watson
Published Temperature = 16 C Gas = 30 % flame = 1 % to IBM Watson
Published Temperature = 48 C Gas = 71 % flame = 1 % to IBM Watson
Published Temperature = 47 C Gas = 87 % flame = 1 % to IBM Watson
Published Temperature = 65 C Gas = 7 % flame = 0 % to IBM Watson
Published Temperature = 32 C Gas = 67 % flame = 1 % to IBM Watson
Published Temperature = 36 C Gas = 87 % flame = 0 % to IBM Watson
Published Temperature = 64 C Gas = 83 % flame = 1 % to IBM Watson
Published Temperature = 17 C Gas = 87 % flame = 0 % to IBM Watson
Published Temperature = 69 C Gas = 98 % flame = 1 % to IBM Watson
Published Temperature = 24 C Gas = 80 % flame = 0 % to IBM Watson
Published Temperature = 49 C Gas = 39 % flame = 1 % to IBM Watson
Published Temperature = 69 C Gas = 32 % flame = 1 % to IBM Watson
Published Temperature = 52 C Gas = 65 % flame = 0 % to IBM Watson
Published Temperature = 58 C Gas = 27 % flame = 0 % to IBM Watson
Published Temperature = 37 C Gas = 4 % flame = 0 % to IBM Watson
Published Temperature = 26 C Gas = 2 % flame = 1 % to IBM Watson
Published Temperature = 52 C Gas = 20 % flame = 1 % to IBM Watson
Published Temperature = 17 C Gas = 11 % flame = 0 % to IBM Watson
Published Temperature = 15 C Gas = 31 % flame = 0 % to IBM Watson
Published Temperature = 11 C Gas = 31 % flame = 0 % to IBM Watson
Published Temperature = 49 C Gas = 25 % flame = 1 % to IBM Watson
Published Temperature = 13 C Gas = 30 % flame = 1 % to IBM Watson
Published Temperature = 9 C Gas = 60 % flame = 1 % to IBM Watson
Published Temperature = 65 C Gas = 100 % flame = 0 % to IBM Watson
Published Temperature = 61 C Gas = 3 % flame = 1 % to IBM Watson
Published Temperature = 17 C Gas = 46 % flame = 1 % to IBM Watson
Published Temperature = 19 C Gas = 57 % flame = 1 % to IBM Watson
Published Temperature = 66 C Gas = 94 % flame = 0 % to IBM Watson
Published Temperature = 58 C Gas = 53 % flame = 1 % to IBM Watson
```

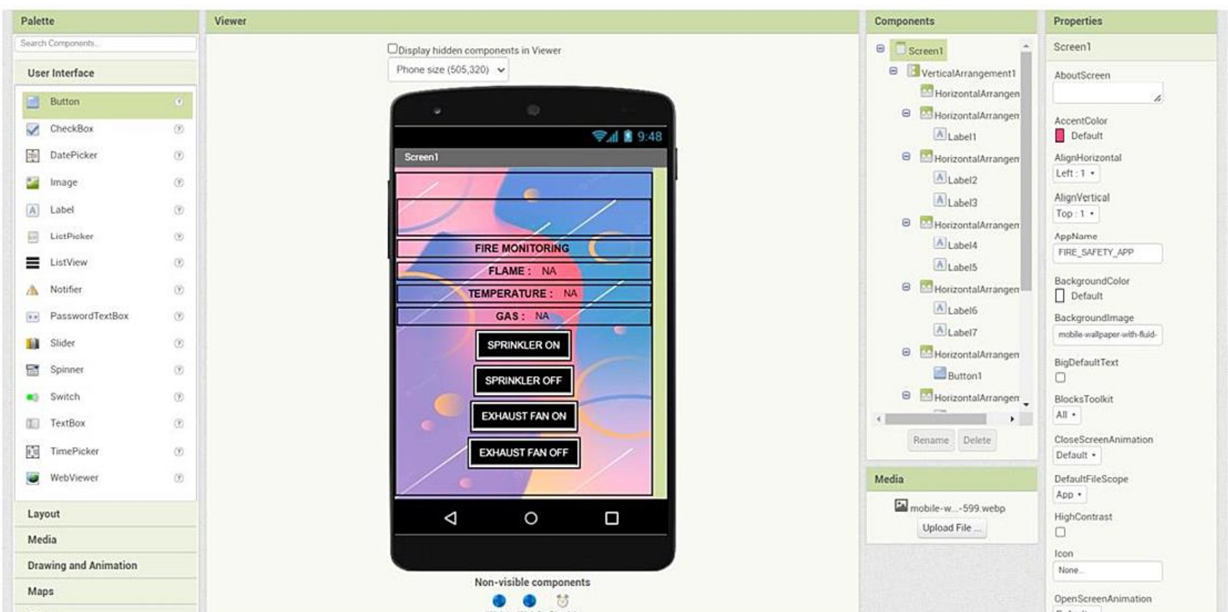


7.2 Feature 2

Send the Data from IBM Watson Cloud to Node-RED andIntegrate MIT App Inventor



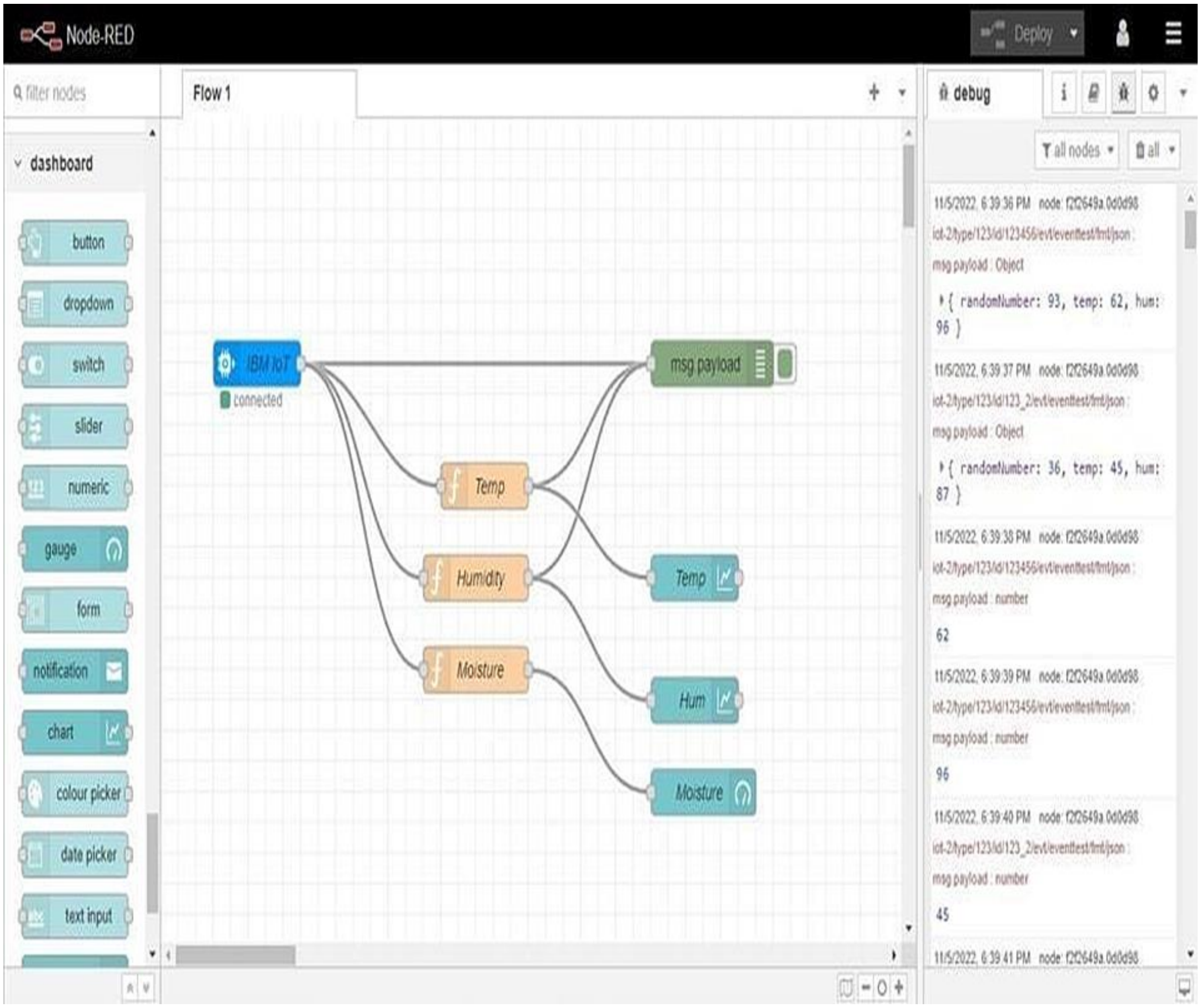

FIRE DETECTION
MONITOR
CONTROL
FLAME
1
SPRINKLER ON
SPRINKLER OFF
FAN ON
FAN OFF
TEMPERATURE
GAS

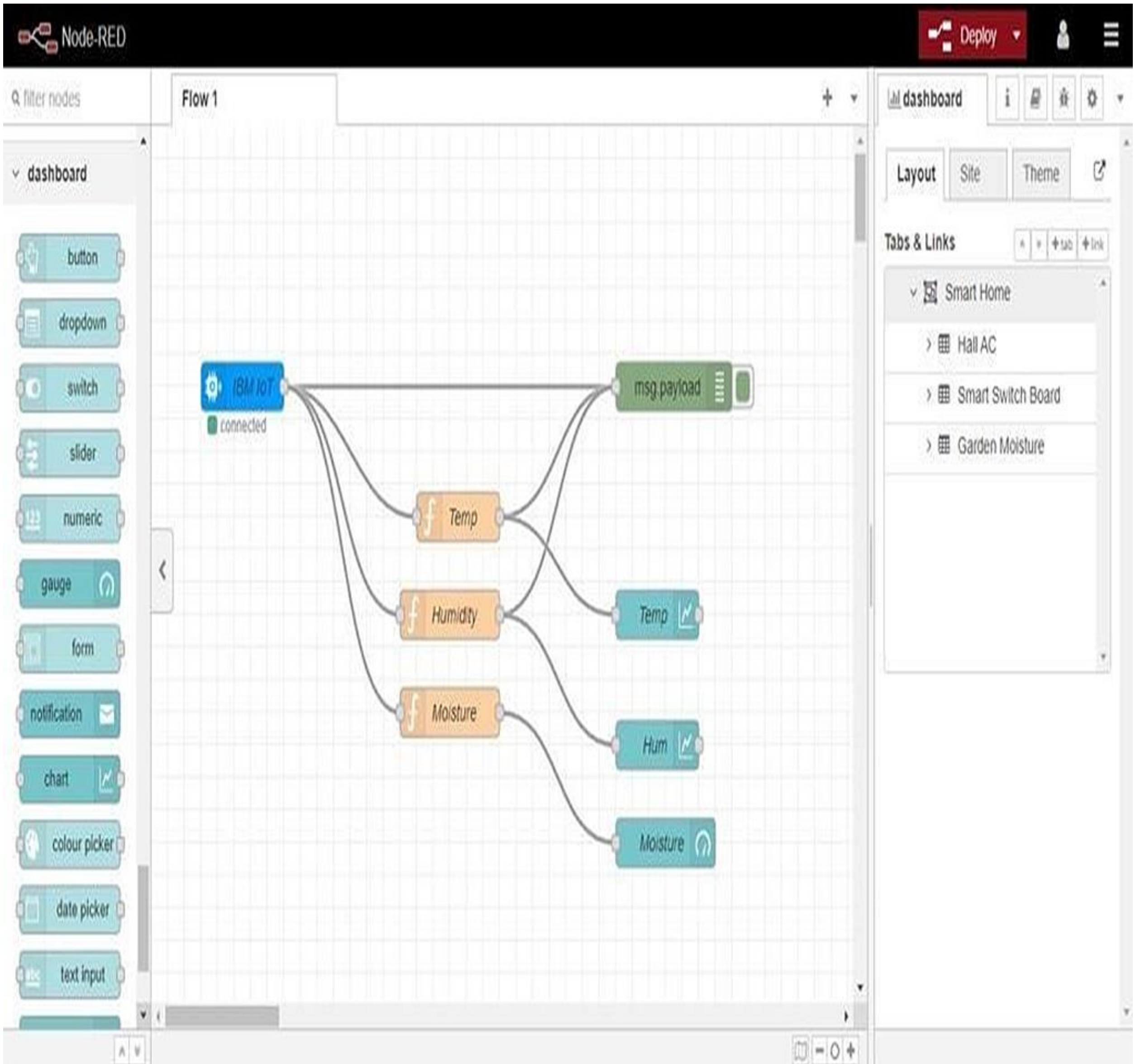

Palette
Viewer
Components
Properties
User Interface
Button
CheckBox
DatePicker
Image
Label
ListPicker
ListView
Notifier
PasswordTextBox
Slider
Spinner
Switch
TextBox
TimePicker
WebViewer
Layout
Media
Drawing and Animation
Maps
Charts
Display hidden components in Viewer
Phone size (505,320)
Screen1
FIRE MONITORING
FLAME : NA
TEMPERATURE : NA
GAS : NA
SPRINKLER ON
SPRINKLER OFF
EXHAUST FAN ON
EXHAUST FAN OFF
Non-visible components
Web1
Web2
Clock1
VerticalArrangement1
Label1
Label2
Label3
Label4
Label5
Label6
Label7
Button1
Rename
Delete
Upload File ...
AboutScreen
AccentColor
AlignHorizontal
AlignVertical
AppName
FIRE_SAFETY_APP
BackgroundColor
BackgroundImage
BigDefaultText
BlocksToolkit
CloseScreenAnimation
DefaultFileScope
HighContrast
Icon
OpenScreenAnimation

Chapter 8

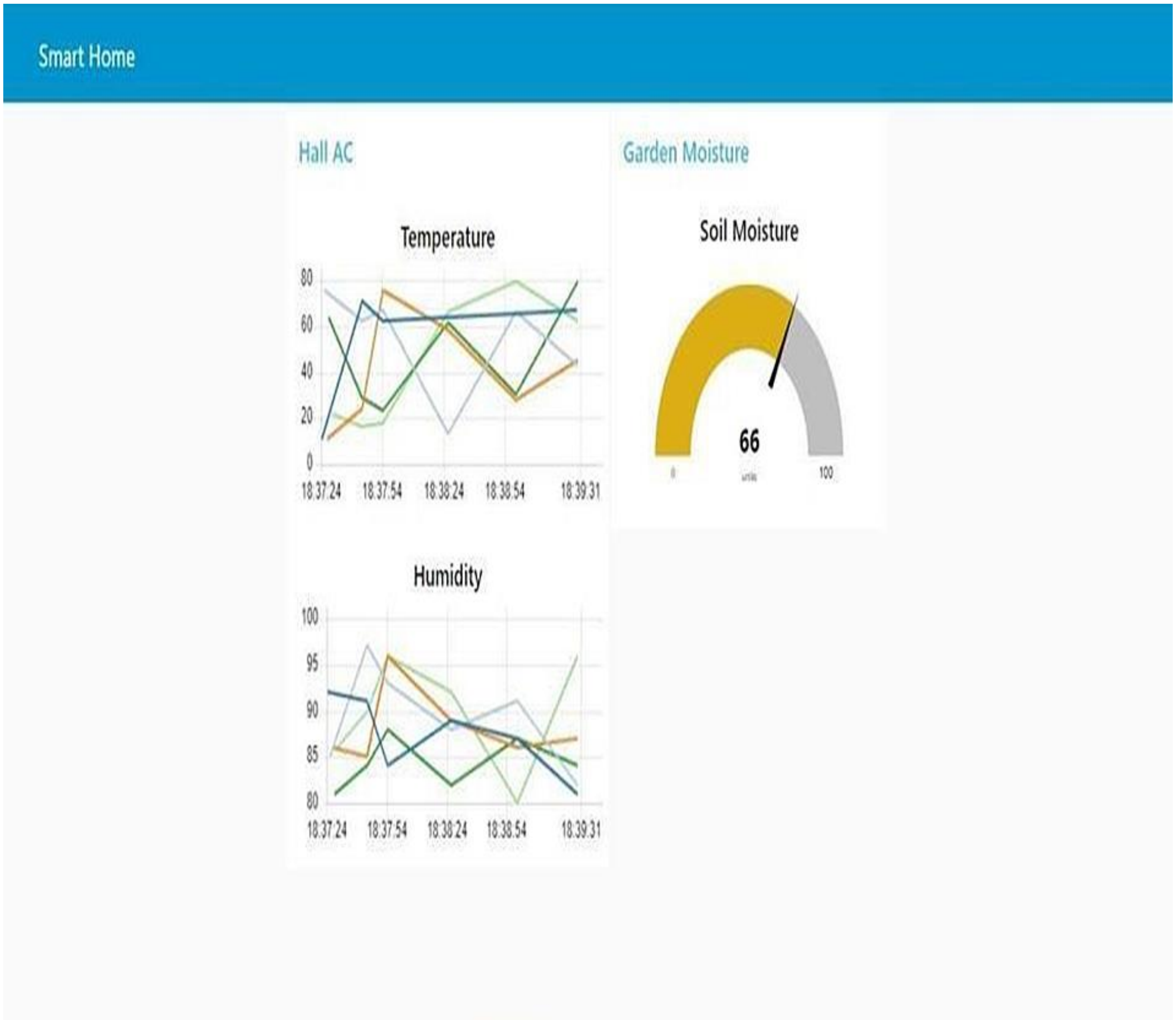
TESTING

8.1 Test Cases


Node-RED
Deploy
filter nodes
Flow 1
debug
all nodes
all
dashboard
button
dropdown
switch
slider
numeric
gauge
form
notification
chart
colour picker
date picker
text input
IBM IoT
connected
msg.payload
Temp
Humidity
Moisture
Temp
Hum
Moisture
msg.payload : Object
{ randomNumber: 93, temp: 62, hum: 96 }
msg.payload : Object
{ randomNumber: 36, temp: 45, hum: 87 }
msg.payload : number
62
msg.payload : number
96
msg.payload : number
45



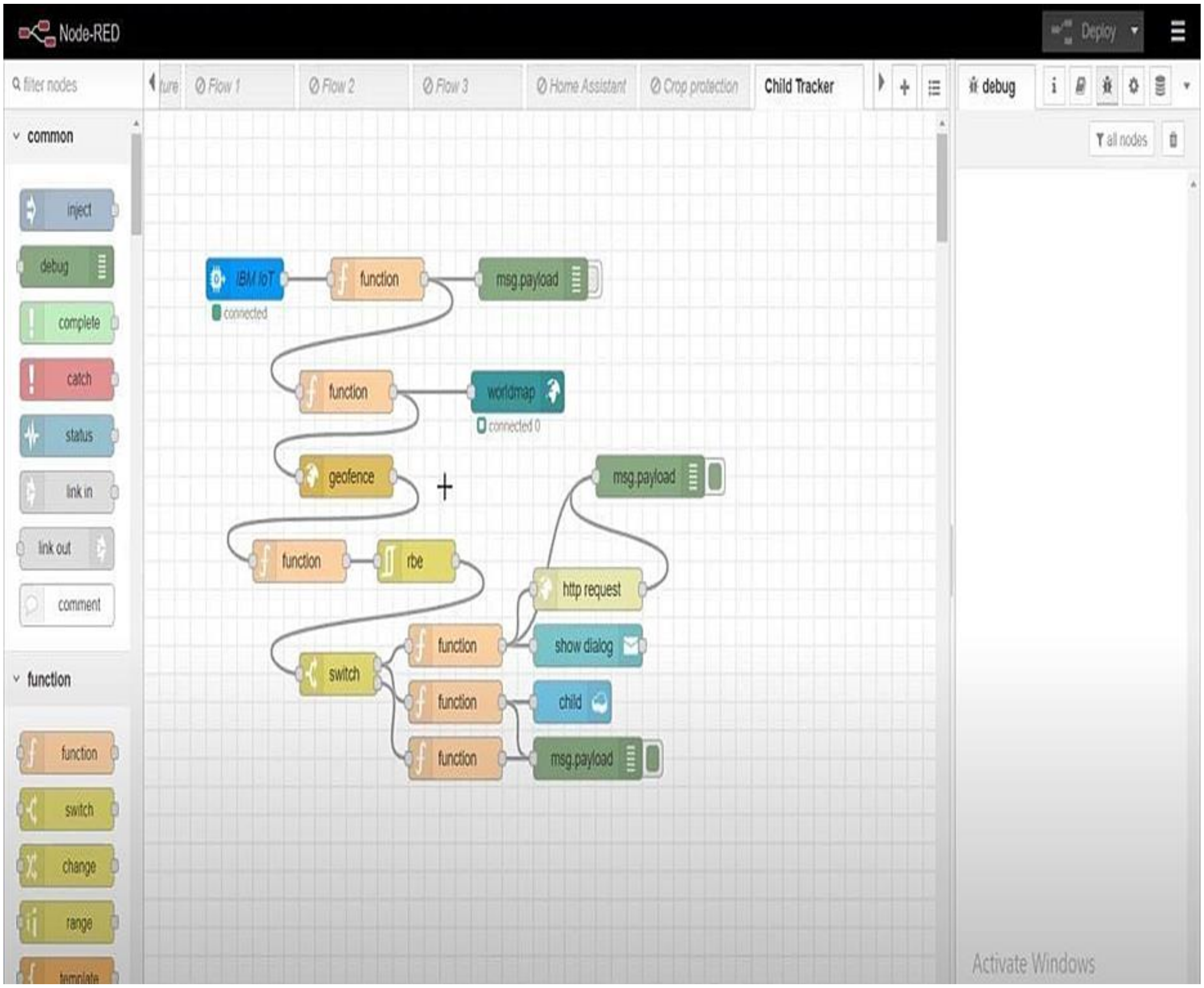
Node-RED
Deploy
filter nodes
Flow 1
dashboard
button
dropdown
switch
slider
numeric
gauge
form
notification
chart
colour picker
date picker
text input
IBM IoT
connected
msg.payload
Temp
Humidity
Moisture
Temp
Hum
Moisture
dashboard
Layout
Site
Theme
Tabs & Links
Smart Home
Hall AC
Smart Switch Board
Garden Moisture



8.2 User Acceptance Testing

Steps Followed:

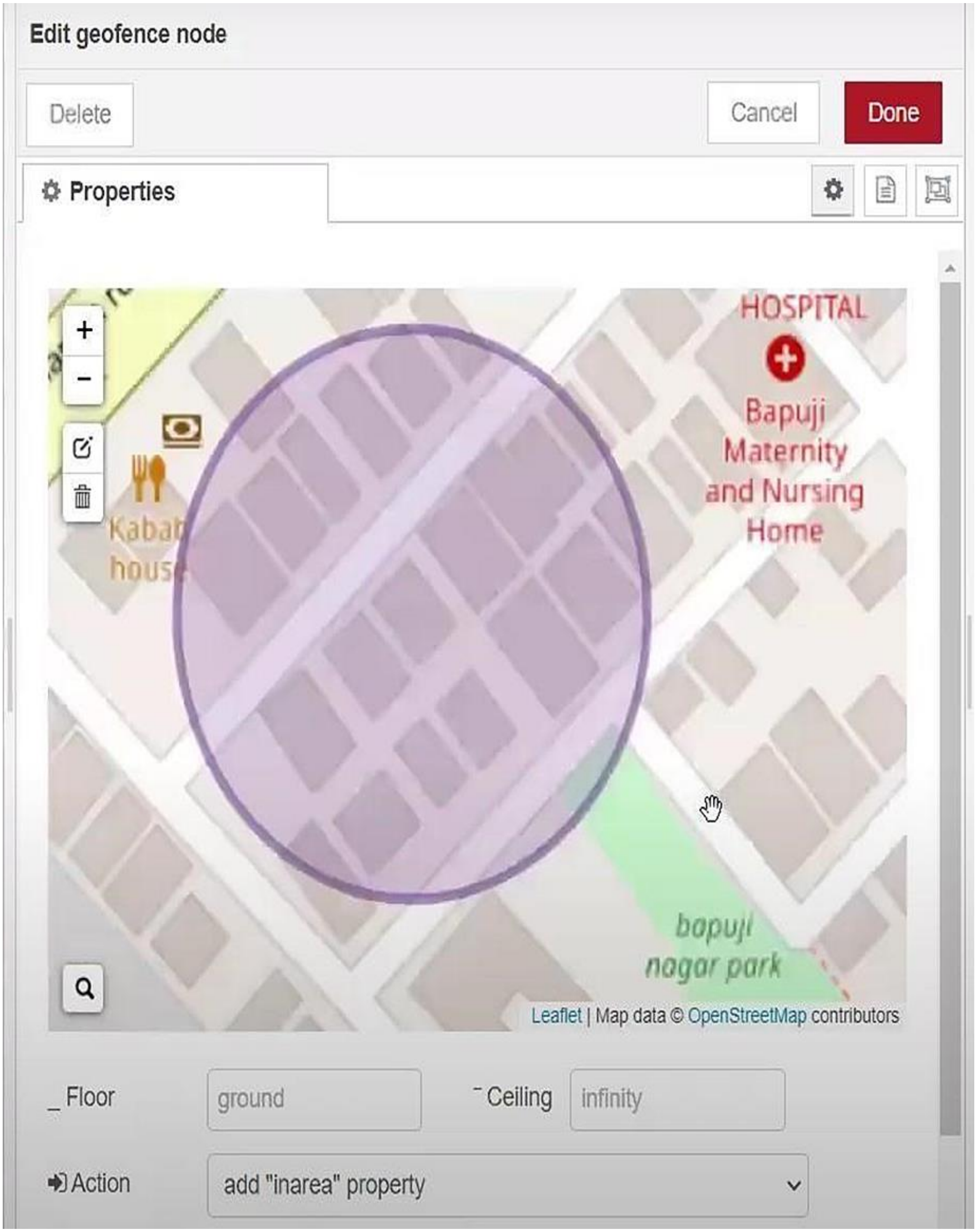
- **Opened a Node-RED project**



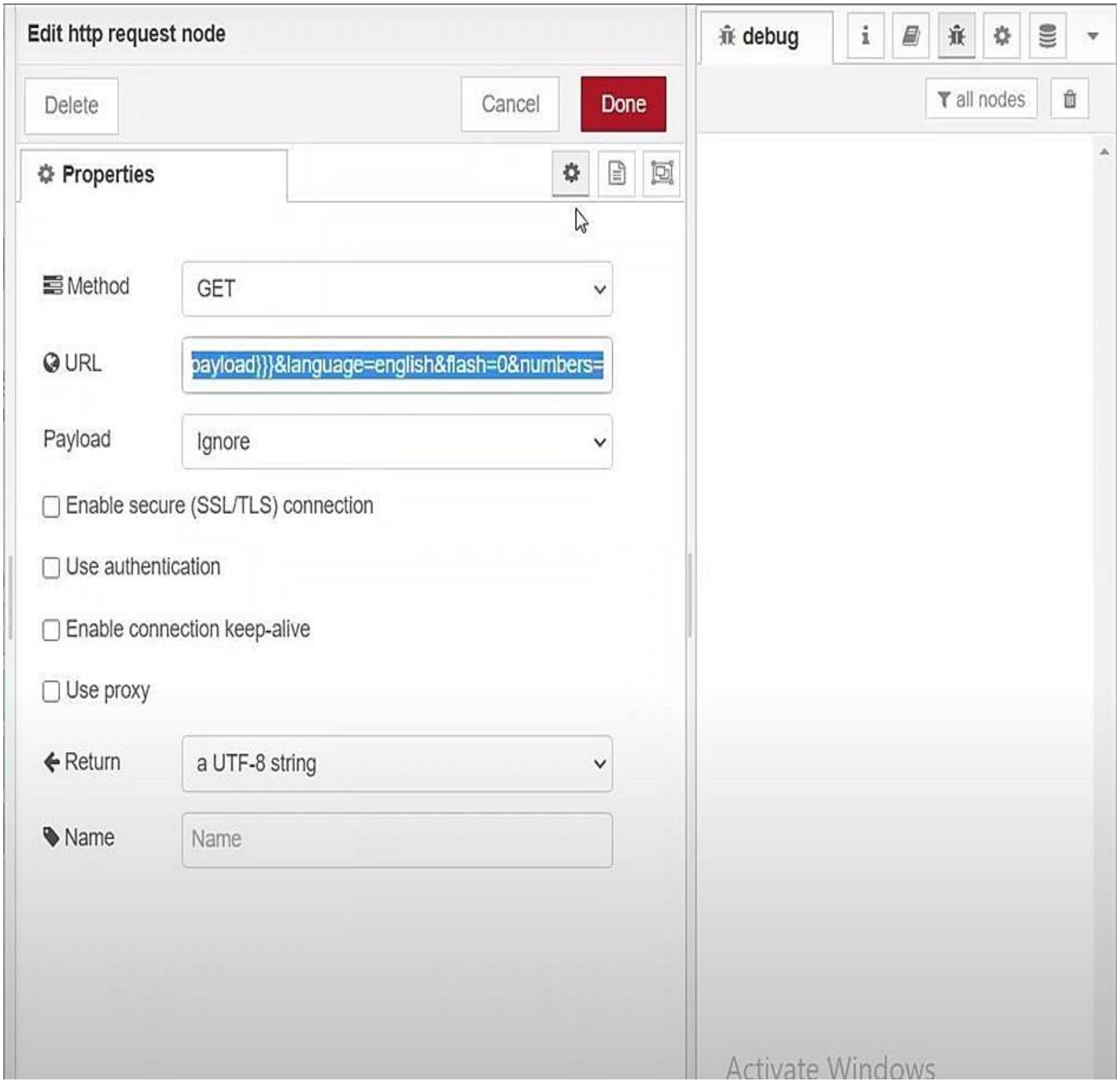
1. **Added code to get child location in python**

```
import json
import wiotp.sdk.device
import time

myConfig = {
    "identity": {
        "orgId": "hj5fmy",
        "typeId": "NodeMCU",
        "deviceId": "12345"
    },
    "auth": {
        "token": "12345678"
    }
}
client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()

while True:
        name= "Smartbridge"
        #in area location

        latitude= 17.4225176
        longitude= 78.5458842

        #out area location

        #latitude= 17.4219272
        #longitude= 78.5488783
        myData={'name': name, 'lat':latitude,'lon':longitude}
        client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
        print("Data published to IBM IoT platfrom: ",myData)
        time.sleep(5)

client.disconnect()
```

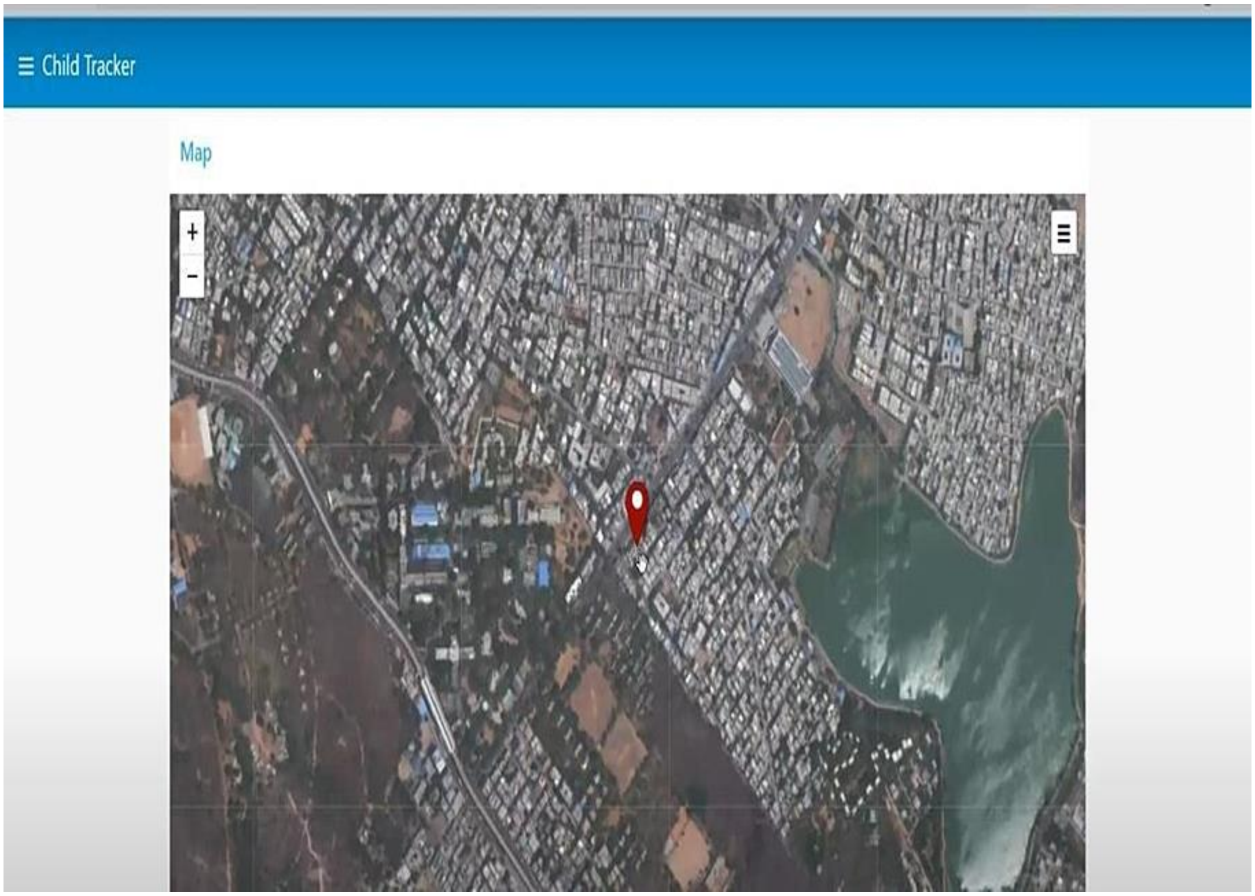
1. **Created the GeoFence**



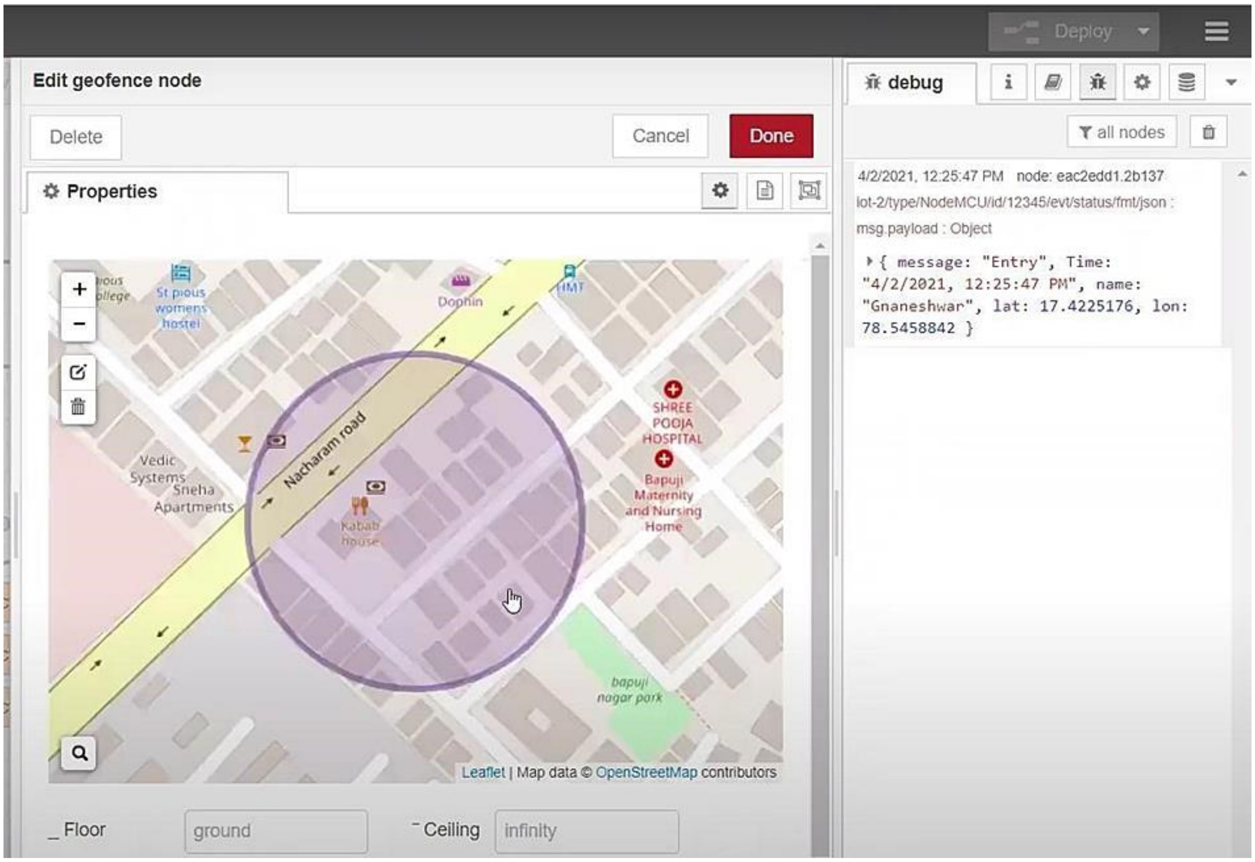
Editing the HTTP Request URL



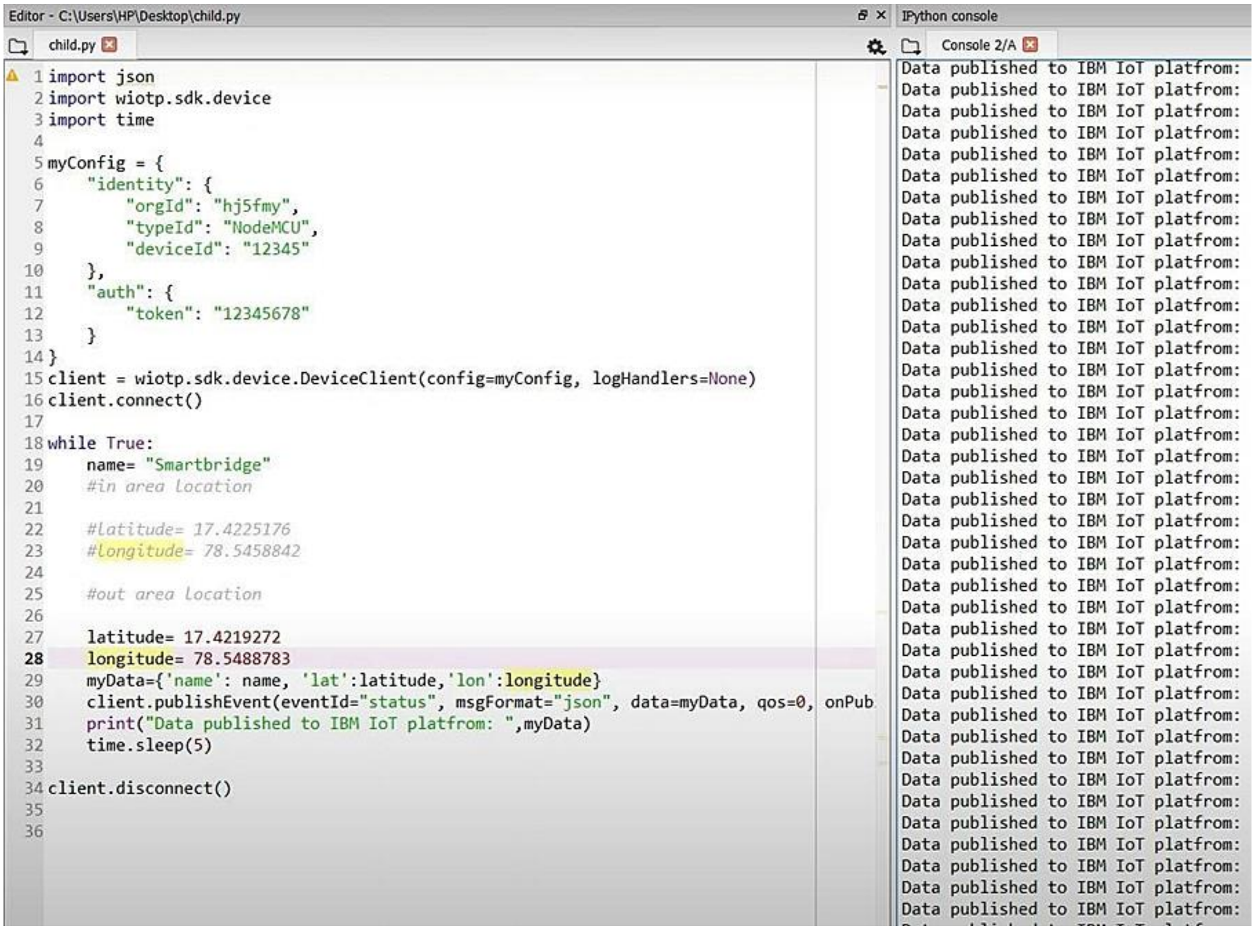
Located the child



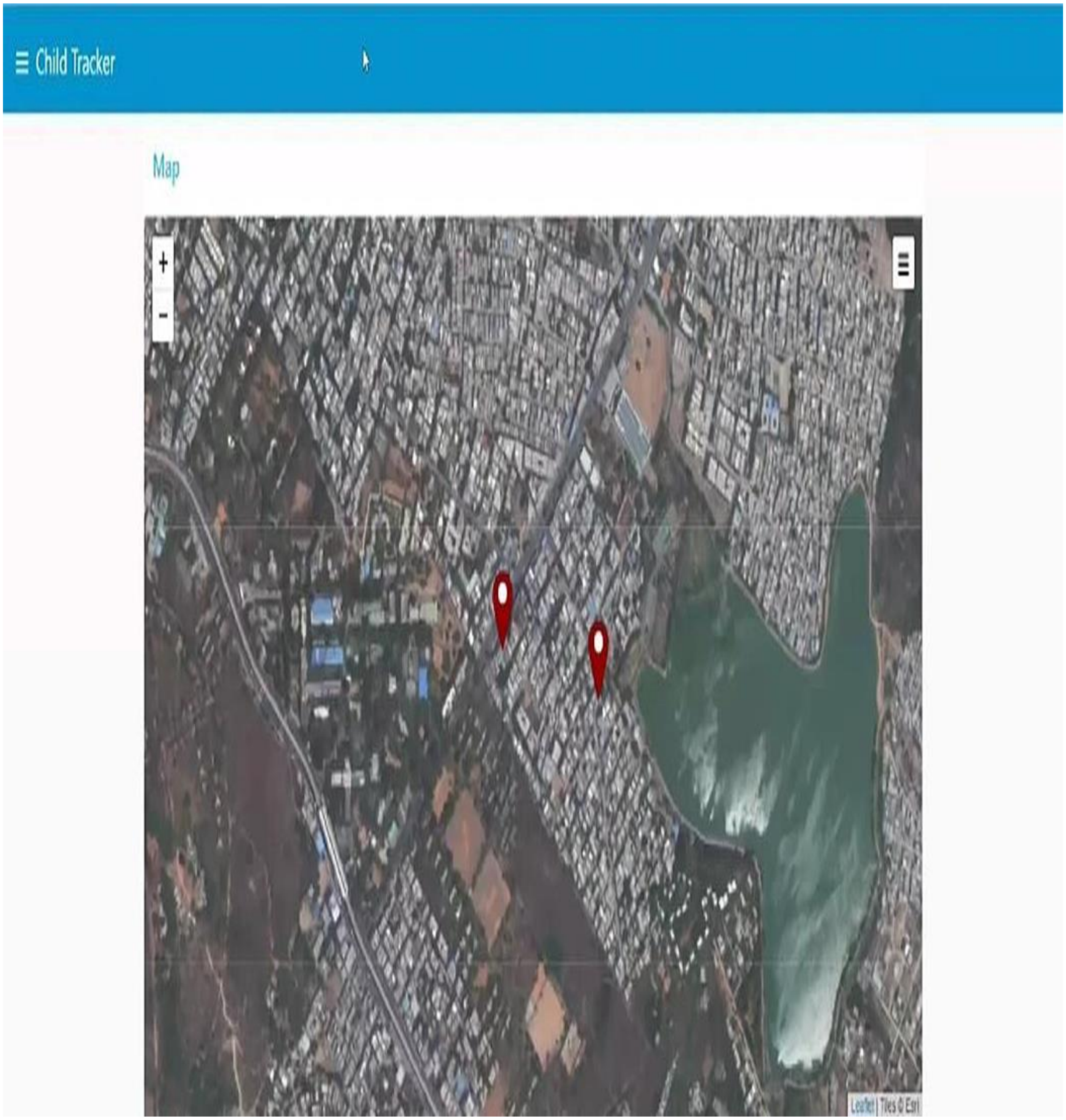
☰ Child Tracker
Map



1. **Python script sending requests to IBM Cloud**

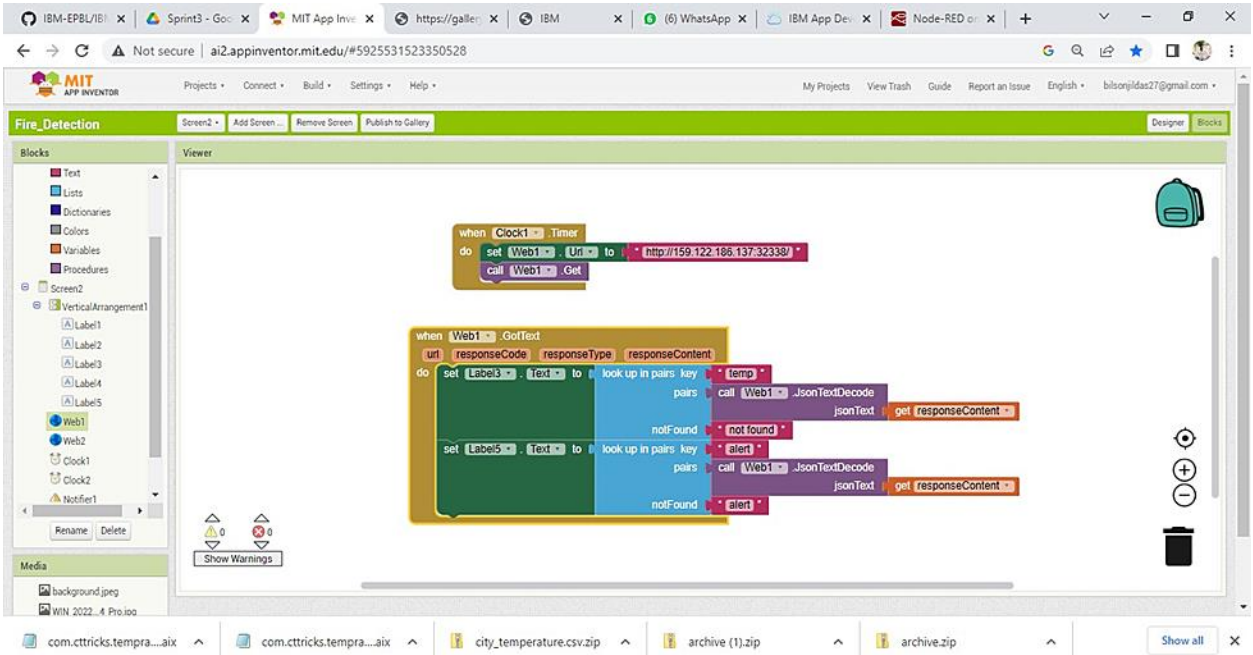


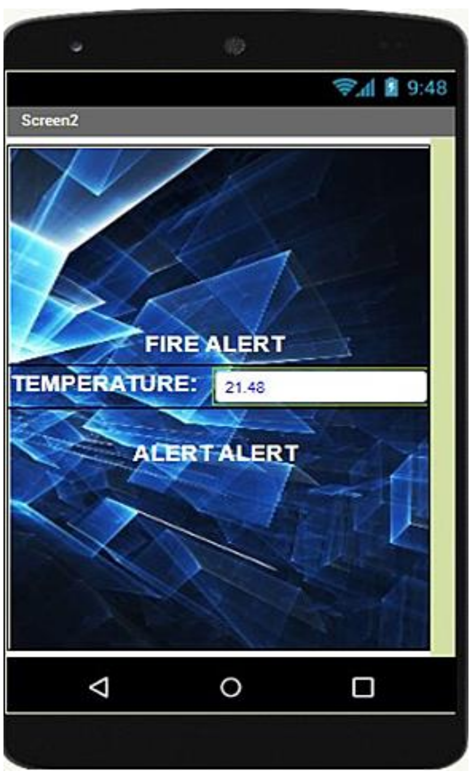
After running the script, the web UI shows “Person is not in the particular area”



Chapter 9
Result

9. 1 Performance Metrics


Fire_Detection
Blocks
Viewer
when Clock1 .Timer
do set Web1 . Url to "http://159.122.186.137:32338/"
call Web1 .Get
when Web1 .GotText
url responseCode responseType responseContent
do set Label3 . Text to look up in pairs key "temp"
pairs call Web1 .JsonTextDecode
jsonText get responseContent
notFound "not found"
set Label5 . Text to look up in pairs key "alert"
pairs call Web1 .JsonTextDecode
jsonText get responseContent
notFound "alert"
Show Warnings


9:48
Screen2
FIRE ALERT
TEMPERATURE:
21.48
ALERT ALERT


9:48
Screen2
FIRE ALERT
TEMPERATURE:
not found
fire alarm warning!
ALERT 43.58


1:45
1:45
Fri, Nov 18
Overcast 19℃
1:45 PM
Fri, Nov 18
1:45 PM
Today
Chennai
fire alarm warning!
3

Chapter 10
Advantages & Disadvantages

Advantages:

1. Better and more accurate fire detection – With an IoT-based system, sensors can be placed in strategic locations around a buildingor area tomore accurately detect a fire
2. Faster responsetimes – An IoT-based systemcan immediately notifythe relevant authorities when a fire is detected, resulting in faster response times.
3. Reduced false alarms – With an IoT-based system,false alarms can bereduced as the system can be programmed to only notify the authorities when certainconditions are met (e.g. a certain level of heat is detected).
4. Greater efficiency – An IoT-based system can automatemany of the tasks associated with fire management, such as dispatching firefightersand keeping track of firefighting resources.

5. Increased Efficiency: By automating tasks and processes, an IoT system

   can help fire departments become more efficientin their operations.
6. Improved Safety: With real-time monitoring and alerts, an IoT system can help keep

   firefighters safe by providingthem with timelyinformation aboutpotential hazards.
7. Enhanced Collaboration: An IoT system can enablebetter collaboration between firefighters and other emergency responders, such as police and ambulance services.
8. Greater Transparency: An IoT system can provide greater transparency into the operations of a fire department, which can be beneficial for boththe department and the public.
9. Reduced Costs: An IoT system can help reduce the costs associated with running a fire department, such as by reducing the need for manuallabor.

Disadvantages:

1. Implementation costs – An IoT-based Intelligent Fire Management System can be expensive to implement, especially if a largenumber ofsensorsare required.
2. Maintenance costs – An IoT-based system can also be expensiveto maintain, as sensors and other hardwarewill need to be regularlycheckedand replaced.

3. Security risks – As IoT-based systems rely on networked devices, theycan be vulnerable to hacking and other securityrisks.
4. Privacy concerns– Some people may be concerned about the privacyimplications of having an IoT-based system, as it will be constantlycollecting dataabout people and their activities
5. Complexity: An IoT system can be complex to set up and maintain,andmay require specialized skills and knowledge.
6. Reliability: An IoT system can be subject to outages and otherdisruptions, whichcould impact the reliability of the system.
7. Security: An IoT system can be vulnerableto security threats,such as

   hacking and data breaches.
8. Privacy: An IoT system can collectand store large amounts of data,which could potentially be used to infringe on the privacy of individuals.

   Cost: An IoT system can be expensive to implement and maintain

Chapter 11
CONCLUSION

The proposedsystem, presents the advancement of Internet technologyin day to day life. The system is suitable for real time small scale industrial process monitoring and controlling applications. proposed module implemented on ESP32, one of the best solutions to implement IoT applications. The module outline was tried, actualized and the accuracy and working of the systemwas verified.

Benefits of use IoT in industry:Elimination of long wiring.

Web based remote monitoring. Immediate action on failures. Ease of maintenance.

Chapter 12
Future Scope

The future scope of IOT based intelligent fire management system is to develop a system that can automatically detect and extinguish fires. The system should be able to identify the type of fire and provide the appropriate response. It should also be able to send alerts to the authoritiesin case of a fire.

The future of IOT based intelligent fire management system looks very promising. With the help of IoT, the system will be able to monitor the fire situation in real time and take appropriate actionaccordingly. The systemwill also be able to automatically detect the fire and send alerts to the concerned authorities.

Chapter 13

APPENDIX

Source code

```
#include "DHTesp.h" constint
DHT_PIN = 15;
DHTespdhtSensor;

void setup(){
  // put your setup code here, to
  run once: Serial.begin(115200);
  Serial.println("Hello,ESP32!");
  dhtSensor.setup(DHT_PIN,DHTesp::DHT22);
}


void loop(){
  TempAndHumidity data =
  dhtSensor.getTempAndHumidity();
```

```
Serial.println("Temp: "+
String(data.temperature,2)+"c");
Serial.println("Humidity:
"+String(data.temperature,1)+"%");
Serial.println("------------------------------
- ");
delay(100); // this speeds up the simulation
}
importtime import sys import ibmiotf.device
import ibmiotf.application import random

#Provide your IBM Watson Device
Credentials organization ="inbee2"
deviceType = "NodeMCU" deviceId
="12345" authMethod = "token"
authToken = "12345678"

# Initialize GPIO

def myCommandCallback1(cmd):
print("Commandreceived: %s" % cmd.data['command'])status =
cmd.data['command']if status == "sprinkleron":
print("sprinkler is on")
else:
print("sprinkler is off ") print(cmd)

def myCommandCallback2(cmd):
print("Commandreceived: %s" % cmd.data['command'])status =
cmd.data['command']if status == "fanon":
print("fan is on") else:
print("fan 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 exceptionconnecting device: %s"% str(e)) sys.exit()
```

```
# Connectand 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


temp=random.randint(0,70)
gas=random.randint(0,100)
flame=random.randint(0,1)


        data = { 'temp' : temp, 'gas': gas,
        'flame': flame } #print datadef
        myOnPublishCallback():
            print ("Published Temperature = %s C"% temp, "Gas = %s %%"% gas, "flame = %s
            %%"
 % flame,"to IBM Watson")


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


        deviceCli.commandCallback1 = myCommandCallback1
        deviceCli.commandCallback2 = myCommandCallback2



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

GitHub & Project Demo Link

https://github.com/IBM-EPBL/IBM-Project-35460-1660284932

https://www.youtube.com/watch?v=r7SJyVuItpI

output

https://wokwi.com/projects/322410731508073042

https://wokwi.com/projects/322410731508073042

https://wokwi.com/projects/322410731508073042

https://wokwi.com/projects/322410731508073042