Industry-specific intelligent fire management system

Team ID	: **PNT2022TMID33754**
Team	:**VINOTH K - TEAM LEAD**
	NITHISH P
	NIVASH B
	PARAMESWARAN S

TABLE OF CONTENTS

INTRODUCTION:

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

1.2 Purpose

The purpose of this project is to report and industry safety.

2. LITERATURE SURVEY

2.1 Existing problem

- Using the internet of things (IoT) to connect things, service, and people for intelligent operations has been discussed and deployed in many industry domains such as smart city, smart energy, healthcare, food and water tracking, logistics and retail, and transportation.
- However, scarce information is available for IoT usage in industrial automation domain for reliable and collaborative automation with respect to e.g., enabling scalable collaboration between heterogeneous devices and systems, offering predictable and fault-tolerant real-time closed-loop control, and inclusion of intelligent service features from edge devices to the cloud.
- In this paper, we will clarify the specific quality attribute constraints within industrial automation, present specific industrial IoT challenges due to these constraints, and discuss the potentials of utilizing some technical solutions to cope with these challenges.

2.2 References :

[1] Choi, S. H., Bae, B. G., and Lee, B. R. (2015), “The sensing model of disaster issues based on relevance to disaster from social big data.” Proceedings of Korea Institute of Information Scientists and Engineers, Korea, Vol. 2015,

[2] J. Gubbi, R. Buyya, S. Marusic, M. Palaniswami, Internet of things (IoT): a vision, architectural elements, and future directions. Futur. Gener. Comput. Syst. 29, 1645–1660 (2013)

[3] "Internet of things and its application in the electrical power industry", Electric Technology, 2016. "Industrial Internet of Things: Unleashing the Potential of Connected Products and Services", World economic Forum Industry Agenda, January 2015.

[4] Y. Chen and H. Hu, "Internet of Intelligent Things and Robot as a Service", Journal Simulation Modelling Practice and Theory, 2013. "Industrial Internet Insights Report for 2015", Accenture. nternet of Things - From Research and Innovation to Market Deployment, Editors:Ovidiu Vermesan, Peter Friess, River Publishers Series in Communication.

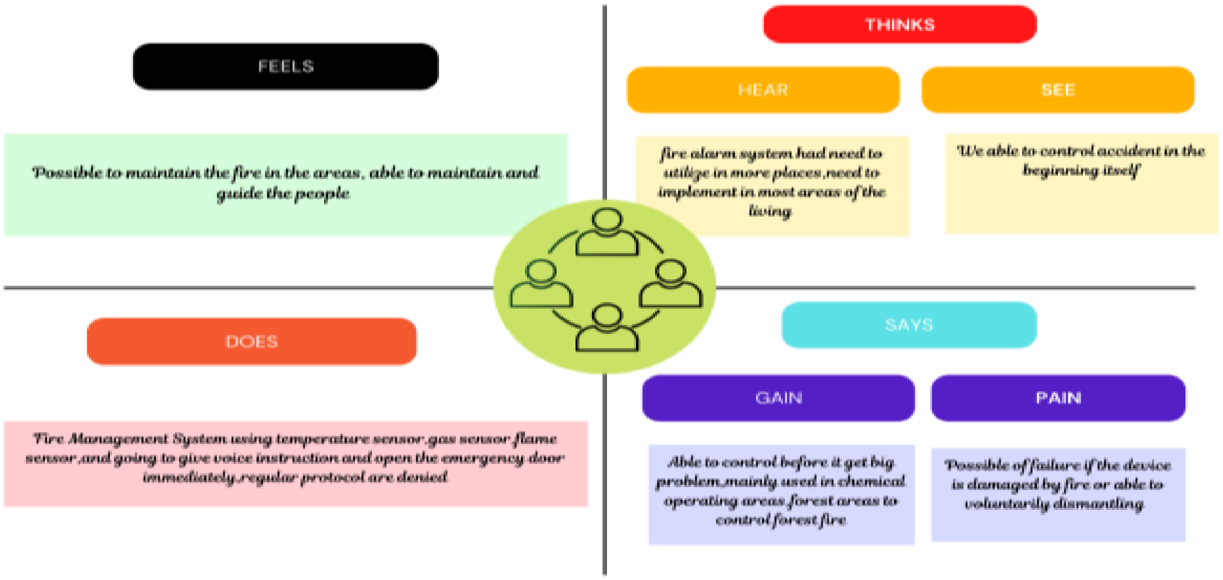
[5] avis J, Edgar T, Porter J, Bernaden J, Sarli M (2012) Smart manufacturing, manufacturing intelligence and demand-dynamic performance

2.3 Problem Statement Definition

The industry is using smart fire management systems. This smart fire management system uses a GAS Sensor, Flame Sensor, and Temperature Sensor. The gas sensor is used to detect any gas leakage and unwanted gasses in closed areas. If gas is detected in the surroundings sensors are automatically activated. A flame sensor is used to capture the shape of the flame RGB color model to identify the fire. Temperature Sensor, to check the amount of heat that is present in the surroundings. These sensors detect the fire, and when it is identified, then suddenly it forwards the alarm to alert the workers.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



Possible to maintain the fire in the areas, able to maintain and guide the people

fire alarm system had need to utilize in more places ,need to implement in most areas of the living

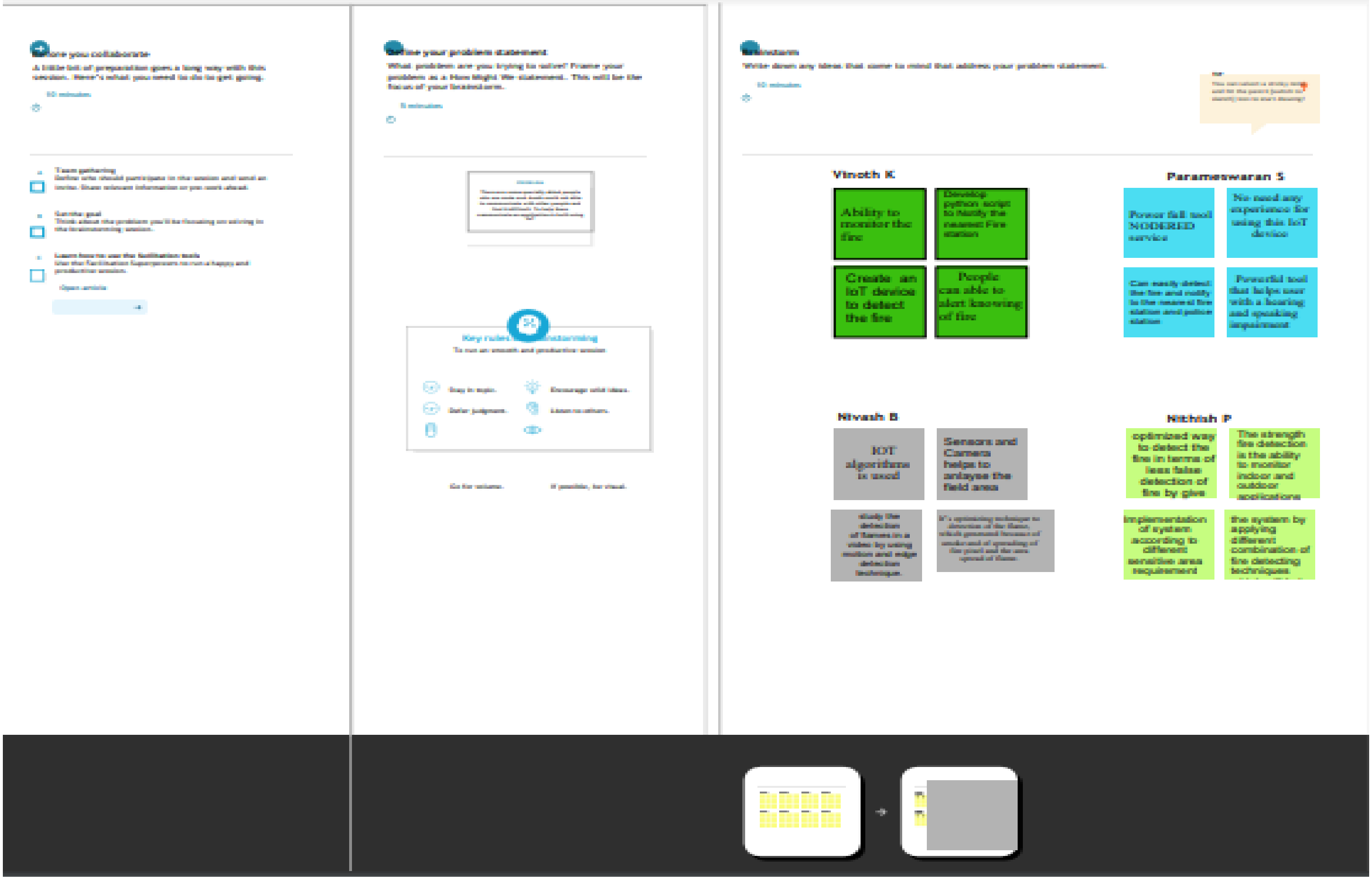
We able to control accident in the beginning itself

Fire Management System using temperature sensor,gas sensor,flame sensor,and going to give voice instruction and open the emergency door immediately,regular protocol are denied

Able to control before it get big problem,mainly used in chemical operating areas,forest areas to control forest fire

Possible of failure if the device is damaged by fire or able to voluntarily dismantling

3.2 Ideation & Brainstorming



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
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

Set the goal
Think about the problem you'll be focusing on solving in the brainstorming session.

Learn how to use the facilitation tools
Use the Facilitation Superpowers to run a happy and productive session.

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

Key rules of brainstorming
To run an smooth and productive session

Stay in topic. Encourage wild ideas.
Defer judgment. Listen to others.
Go for volume. If possible, be visual.

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

Vinoth K

- Ability to monitor the fire
- Develop python script to Notify the nearest Fire station
- Create an IoT device to detect the fire
- People can able to alert knowing of fire

Parameswaran S

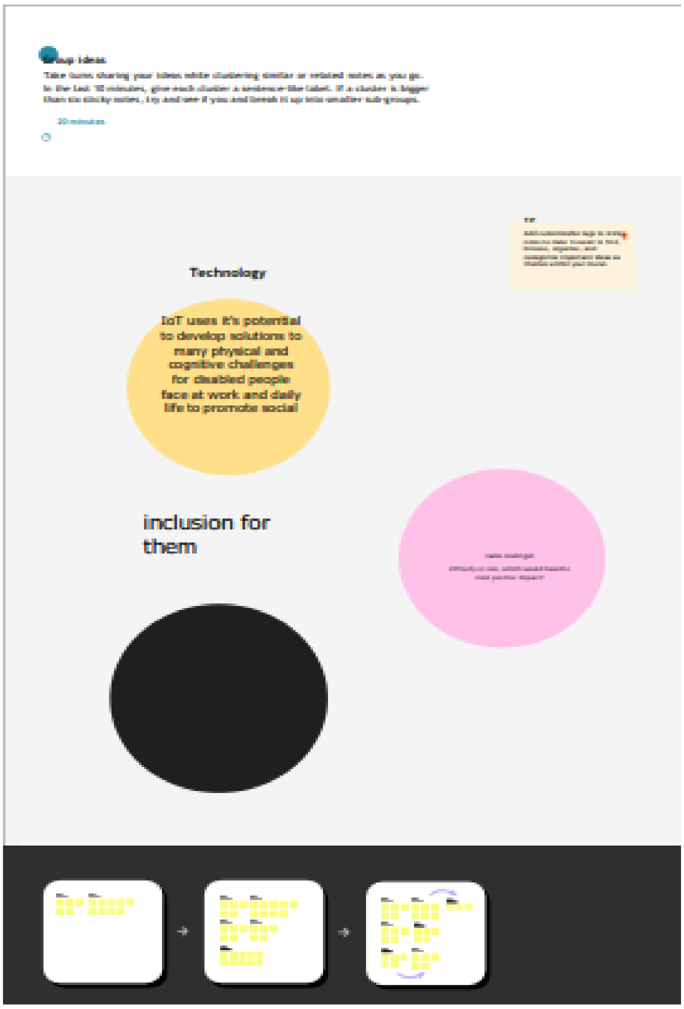
- Power full tool NODERED service
- No need any experience for using this IoT device
- Can easily detect the fire and notify to the nearest fire station and police station
- Powerful tool that helps user with a hearing and speaking impairment

Nivash B

- IOT algorithms is used
- Sensors and Camera helps to anlayse the field area
- study the detection of flames in a video by using motion and edge detection technique.
- It's optimizing technique to detection of the flame, which generated because of smoke and of spreading of the pixel and the area spread of flame.

Nithish P

- optimized way to detect the fire in terms of less false detection of fire by give
- The strength fire detection is the ability to monitor indoor and outdoor applications
- implementation of system according to different sensitive area requirement
- the system by applying different combination of fire detecting techniques



Group ideas

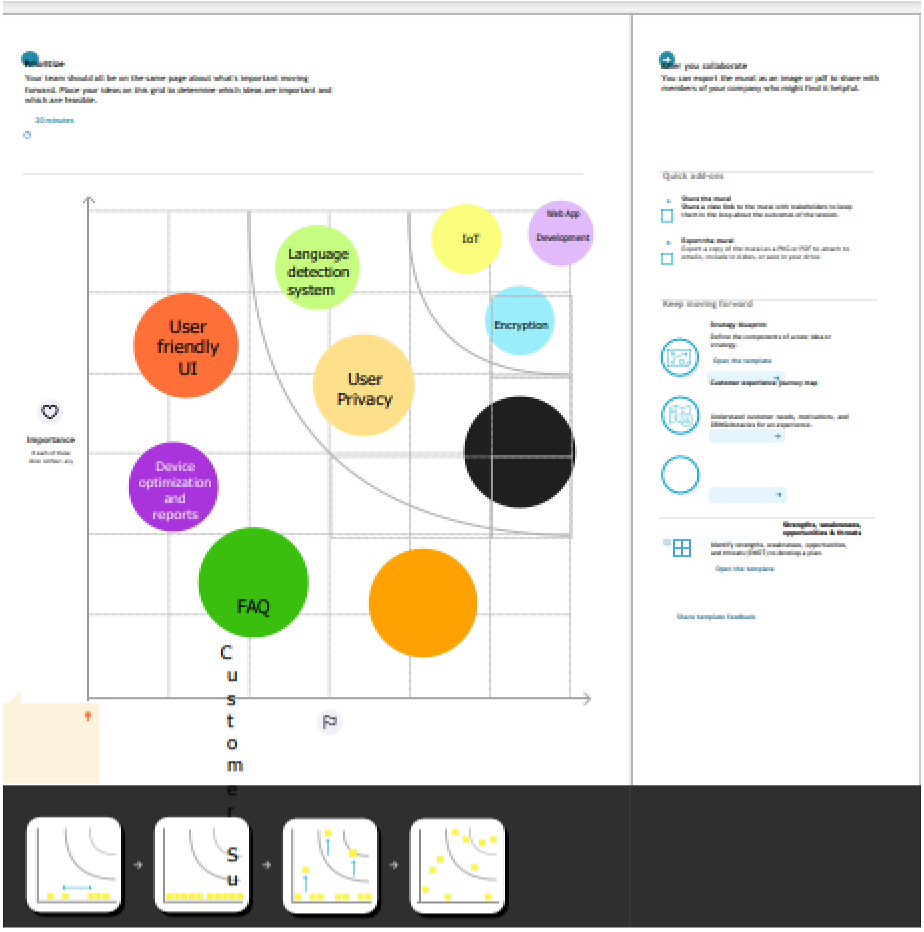
Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

Technology

IoT uses it's potential to develop solutions to many physical and cognitive challenges for disabled people face at work and daily life to promote social

inclusion for them



Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

Quick add-ons

□ Share the mural
Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.

□ Export the mural
Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.

Keep moving forward

Strategy blueprint
Define the components of a new idea or strategy.
Open the template

Customer experience journey map
Understand customer needs, motivations, and obstacles for an experience.

Strengths, weaknesses, opportunities & threats
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.
Open the template

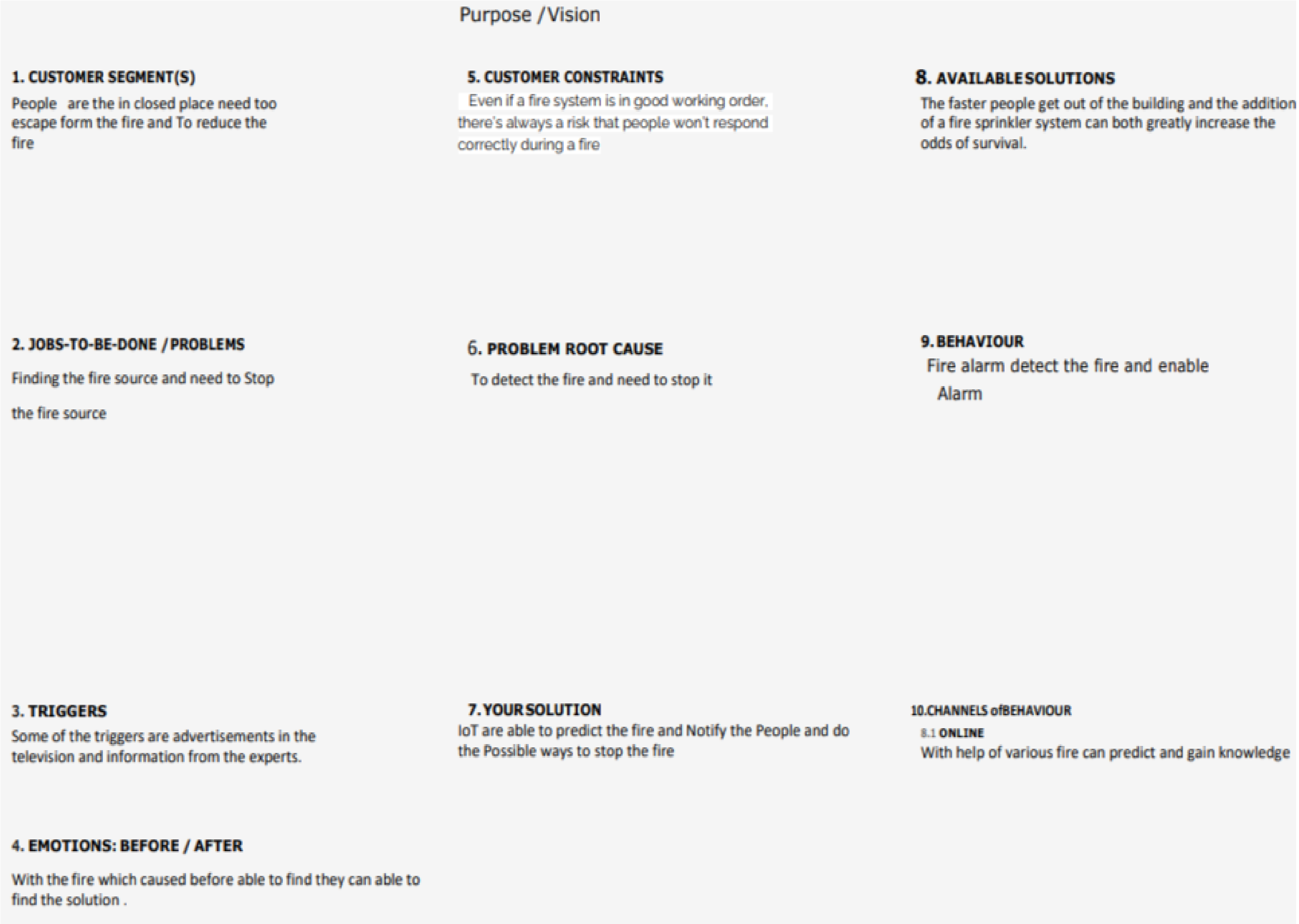
Share template feedback

3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Nowadays industry work is hard and produces more accidents. IoT technology uses automation functions to control accidents and disasters. The industry is using smart fire management systems. This smart fire management system uses a GAS Sensor, Flame Sensor, and Temperature Sensor. The gas sensor is used to detect any gas leakage and unwanted gasses in closed areas. If gas is detected in the surroundings sensors are automatically activated. A flame sensor is used to capture the shape of the flame RGB color model to identify the fire. Temperature Sensor, to check the amount of heat that is present in the surroundings. These sensors detect the fire, and when it is identified, then suddenly it forwards the alarm to alert the workers. When the alarm sound is received by the protocol, it releases all the doors in the industry and also alerts those members to get out of work from the industries. The sprinklers are activated and the water spears all the places of fire. This causes the workers to not panic when a flame is caught in the industries. It is very useful for workers and prevents the industries with a short period. When these sensors are not present in the industries, it is very hazardous to all workers and sometimes it creates severe injuries and even death. Emergency alerts are notified to the authorities and the Fire station. Through the smoke and gaseous substances, it can easily be detected by the sensor, due to this, the exhaust fan is turned on.
2.	Idea / Solution description	IOT used to analyze fire management industry specific intelligence. Automatic fire detection

		, Temperature , Gas leakages also finding automatic safety protection.
3.	Novelty / Uniqueness	Industries are having only many kinds of solutions. This method avoid the human deaths and does not create panic situations. • Avoid panic situation instruct to the Voice instruction • Critical time removal of the protocols are automatically (ex: Doors, Emergency exits ,etc..)
4.	Social Impact / Customer Satisfaction	• Customer hope to testing the project and medium cost • Using protocols easily • Manages the sensors in one control
5.	Business Model (Revenue Model)	● Medium cost • Is 50k less than the cost price • Get more profit
6.	Scalability of the Solution	• Scalability of the solution is 90% working model of the this project • Working possible to create and procedure to handle

3.4 Problem Solution :



Purpose / Vision

1. CUSTOMER SEGMENT(S)

People are the in closed place need too escape form the fire and To reduce the fire

2. JOBS-TO-BE-DONE / PROBLEMS

Finding the fire source and need to Stop

the fire source

3. TRIGGERS

Some of the triggers are advertisements in the television and information from the experts.

4. EMOTIONS: BEFORE / AFTER

With the fire which caused before able to find they can able to find the solution .

5. CUSTOMER CONSTRAINTS

Even if a fire system is in good working order, there's always a risk that people won't respond correctly during a fire

6. PROBLEM ROOT CAUSE

To detect the fire and need to stop it

7. YOUR SOLUTION

IoT are able to predict the fire and Notify the People and do the Possible ways to stop the fire

8. AVAILABLE SOLUTIONS

The faster people get out of the building and the addition of a fire sprinkler system can both greatly increase the odds of survival.

9. BEHAVIOUR

Fire alarm detect the fire and enable Alarm

10.CHANNELS ofBEHAVIOUR

8.1 **ONLINE**

With help of various fire can predict and gain knowledge

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	No registration requirement
FR-2	User Confirmation	No confirmation requirement

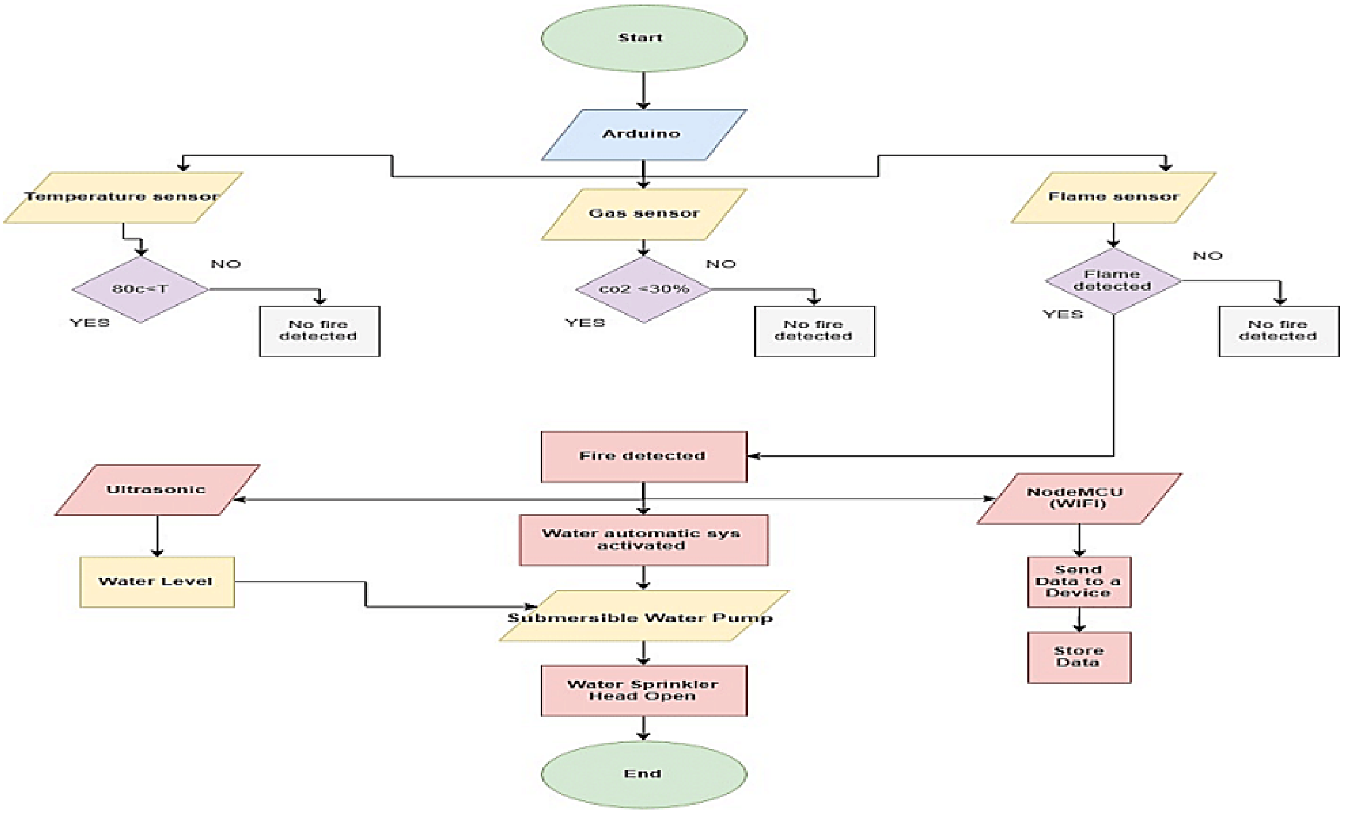
4.2 Non-Functional requirement

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Used to detect the fire, smoke, gasses using IoT devices
NFR-	**Security**	Easy to save the lives of the worker

2		
NFR-3	**Reliability**	People can able to feel free about the fire in the industry
NFR-4	**Performance**	High performance to detect the fire
NFR-5	**Availability**	Can fix where ever need
NFR-6	**Scalability**	Improve the security and spraying the water fire extinguisher

5. PROJECT DESIGN

5.1 Data Flow Diagrams



5.2 Solution Architecture

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

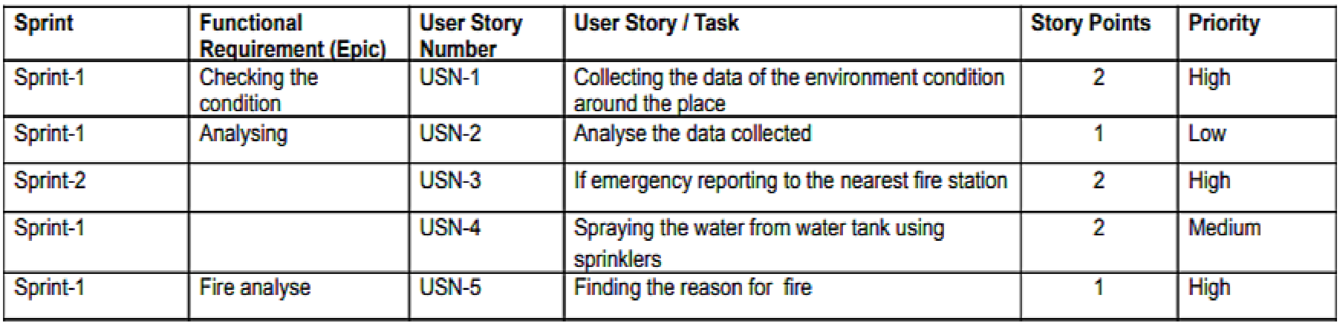
•Find the best tech solution to solve existing business problems.

•Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.

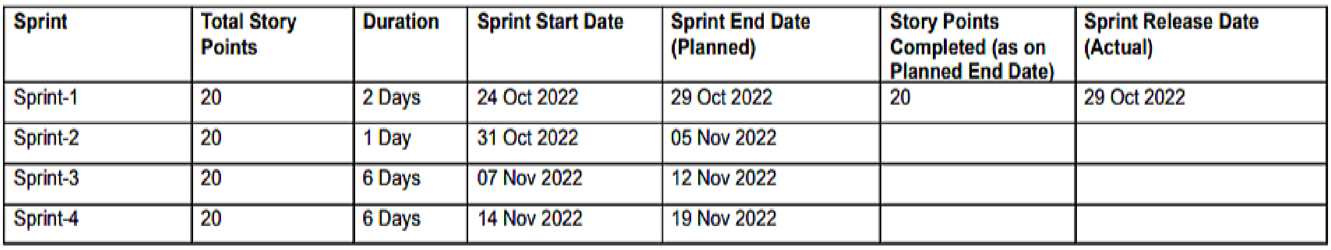
•Define features, development phases, and solution requirements.

• Provide specifications according to which the solution is defined, managed, and delivered.

5.3 User Stories



Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority
Sprint-1	Checking the condition	USN-1	Collecting the data of the environment condition around the place	2	High
Sprint-1	Analysing	USN-2	Analyse the data collected	1	Low
Sprint-2		USN-3	If emergency reporting to the nearest fire station	2	High
Sprint-1		USN-4	Spraying the water from water tank using sprinklers	2	Medium
Sprint-1	Fire analyse	USN-5	Finding the reason for fire	1	High



Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	2 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	1 Day	31 Oct 2022	05 Nov 2022		
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022		
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022		

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

STEP 1 Identify the fire

STEP 2 Prepare an abstract, problemstatement

STEP 3 List required objects needed

STEP 4 Create a code and run it

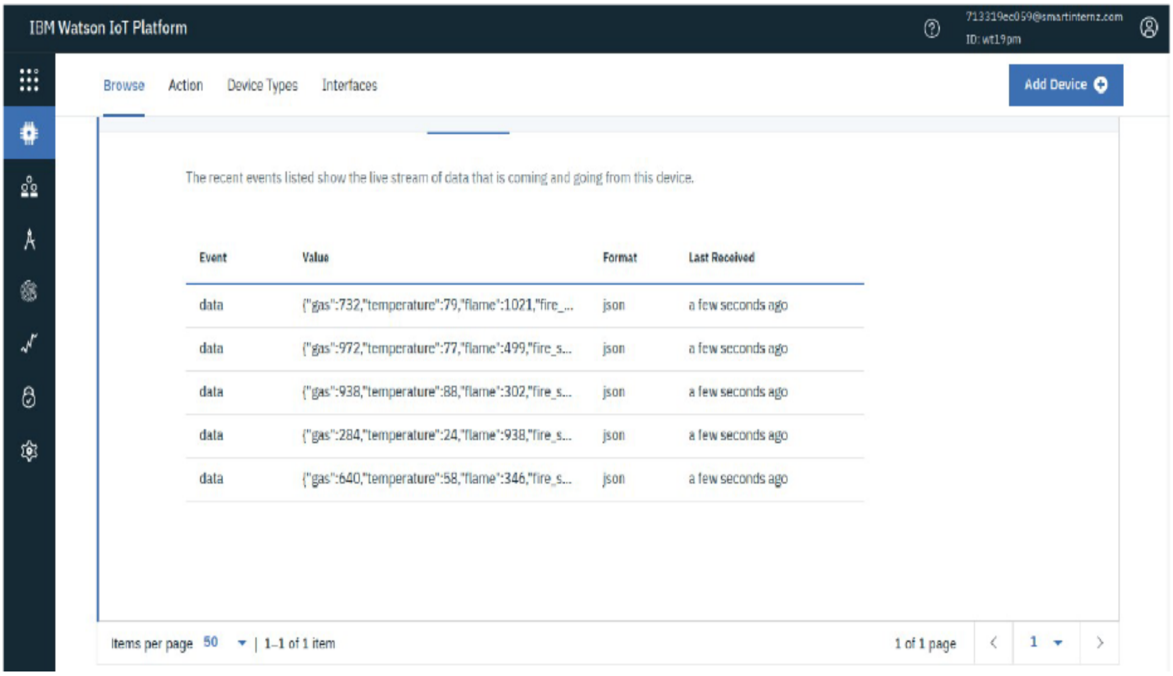
STEP 5 Make a prototype

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

STEP 7 Solution for the problem is found

6.2 Reports from JIRA

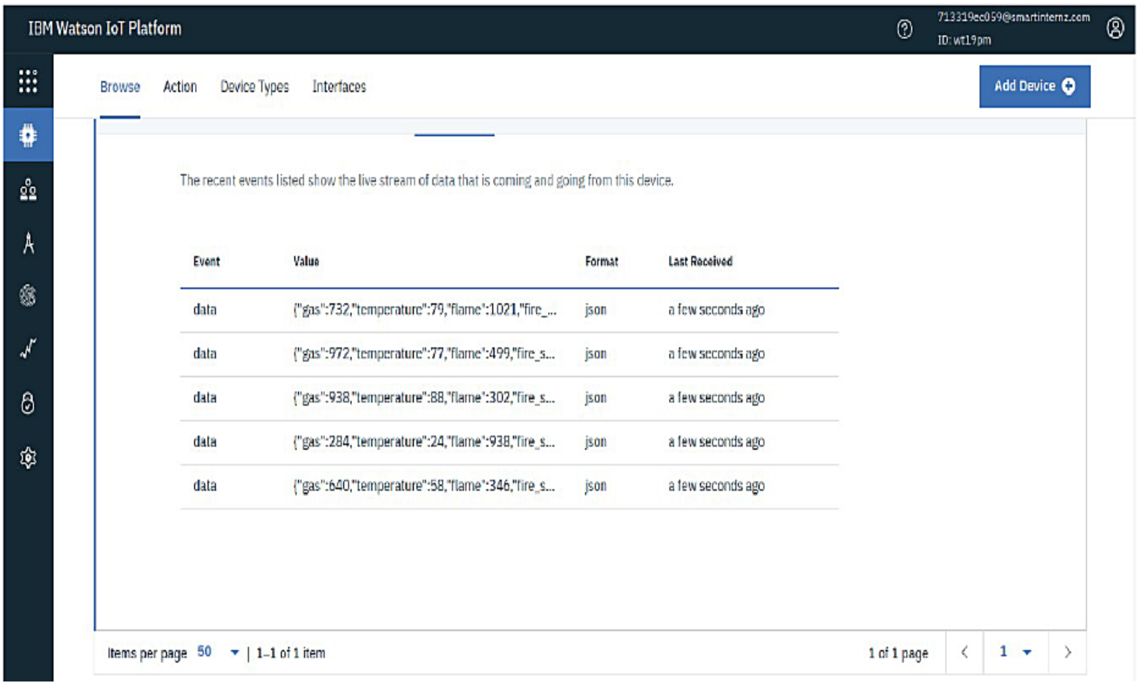
SPRINT 1



The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
data	{"gas":732,"temperature":79,"flame":1021,"fire_...	json	a few seconds ago
data	{"gas":972,"temperature":77,"flame":499,"fire_s...	json	a few seconds ago
data	{"gas":938,"temperature":88,"flame":302,"fire_s...	json	a few seconds ago
data	{"gas":284,"temperature":24,"flame":938,"fire_s...	json	a few seconds ago
data	{"gas":640,"temperature":58,"flame":346,"fire_s...	json	a few seconds ago

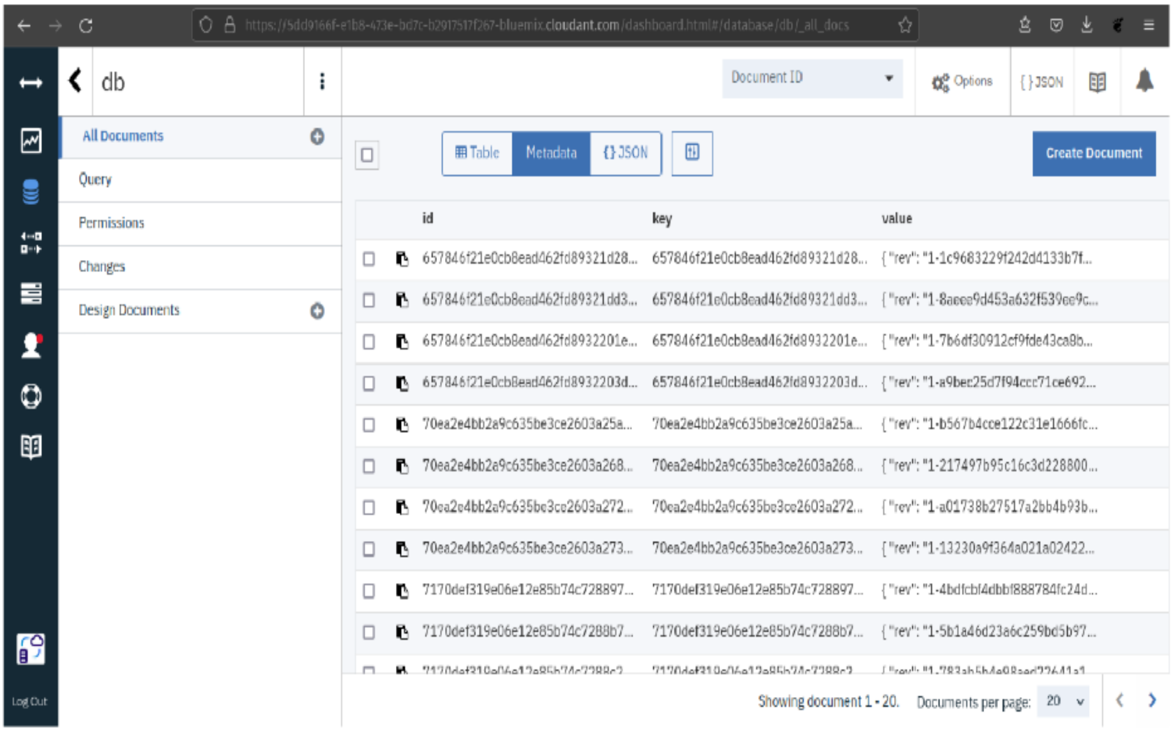
SPRINT 2



The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
data	{"gas":732,"temperature":79,"flame":1021,"fire_...	json	a few seconds ago
data	{"gas":972,"temperature":77,"flame":499,"fire_s...	json	a few seconds ago
data	{"gas":938,"temperature":88,"flame":302,"fire_s...	json	a few seconds ago
data	{"gas":284,"temperature":24,"flame":938,"fire_s...	json	a few seconds ago
data	{"gas":640,"temperature":58,"flame":346,"fire_s...	json	a few seconds ago

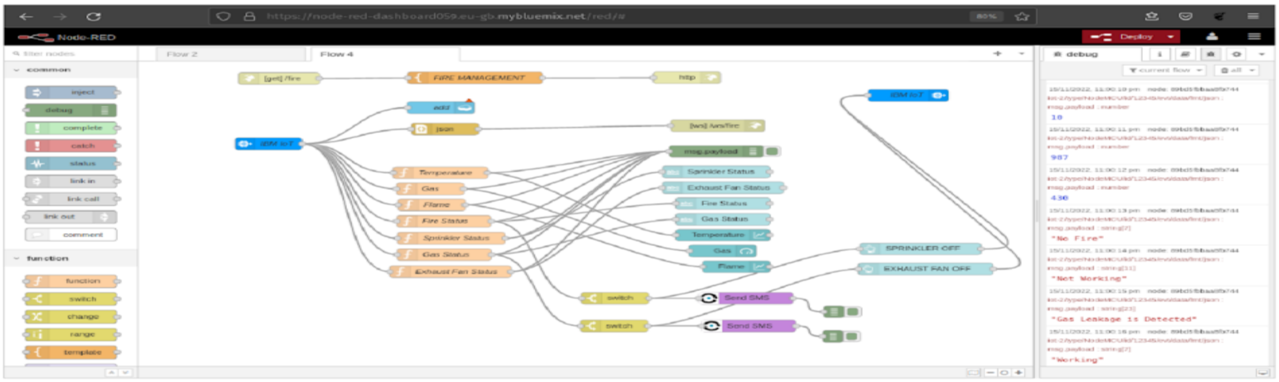
SPRINT 3



	id	key	value
□	657846f21e0cb8ead462fd89321d28...	657846f21e0cb8ead462fd89321d28...	{"rev": "1-1c9683229f242d4133b7f...
□	657846f21e0cb8ead462fd89321dd3...	657846f21e0cb8ead462fd89321dd3...	{"rev": "1-8aeee9d453a632f539ee9c...
□	657846f21e0cb8ead462fd8932201e...	657846f21e0cb8ead462fd8932201e...	{"rev": "1-7b6df30912cf9fde43ca8b...
□	657846f21e0cb8ead462fd8932203d...	657846f21e0cb8ead462fd8932203d...	{"rev": "1-a9bec25d7f94ccc71ce692...
□	70ea2e4bb2a9c635be3ce2603a25a...	70ea2e4bb2a9c635be3ce2603a25a...	{"rev": "1-b567b4cce122c31e1666fc...
□	70ea2e4bb2a9c635be3ce2603a268...	70ea2e4bb2a9c635be3ce2603a268...	{"rev": "1-217497b95c16c3d228800...
□	70ea2e4bb2a9c635be3ce2603a272...	70ea2e4bb2a9c635be3ce2603a272...	{"rev": "1-a01738b27517a2bb4b93b...
□	70ea2e4bb2a9c635be3ce2603a273...	70ea2e4bb2a9c635be3ce2603a273...	{"rev": "1-13230a9f364a021a02422...
□	7170def319e06e12e85b74c728897...	7170def319e06e12e85b74c728897...	{"rev": "1-4bdfcbf4dbbf888784fc24d...
□	7170def319e06e12e85b74c7288b7...	7170def319e06e12e85b74c7288b7...	{"rev": "1-5b1a46d23a6c259bd5b97...
□	7170def319e06e12e85b74c7288c2	7170def319e06e12e85b74c7288c2	{"rev": "1-783ab5b4e08aed72641a1

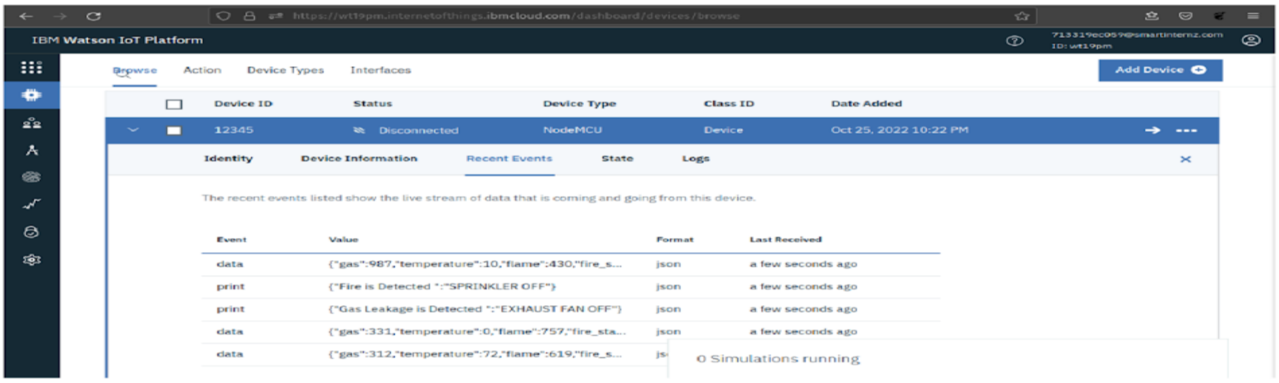
Showing document 1 - 20. Documents per page: 20

SPRINT 4



```
msg.payload : number
10
msg.payload : number
987
msg.payload : number
430
msg.payload : string[7]
"No Fire"
msg.payload : string[11]
"Not Working"
msg.payload : string[23]
"Gas Leakage is Detected"
msg.payload : string[7]
"Working"
```

IBM WATSON IOT:



https://wt19pm.internetofthings.ibmcloud.com/dashboard/devices/browse
IBM Watson IoT Platform
713319ec059@smartinternz.com
ID: wt19pm
Browse
Action
Device Types
Interfaces
Add Device

□	Device ID	Status	Device Type	Class ID	Date Added
□	12345	Disconnected	NodeMCU	Device	Oct 25, 2022 10:22 PM

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
data	{"gas":987,"temperature":10,"flame":430,"fire_s...	json	a few seconds ago
print	{"Fire is Detected ":"SPRINKLER OFF"}	json	a few seconds ago
print	{"Gas Leakage is Detected ":"EXHAUST FAN OFF"}	json	a few seconds ago
data	{"gas":331,"temperature":0,"flame":757,"fire_sta...	json	a few seconds ago
data	{"gas":312,"temperature":72,"flame":619,"fire_s...	js	

0 Simulations running

7. CODING & SOLUTIONING

7.1 Feature 1

· IoT device

· IBM Watson Platform

· Node red

· Cloudant DB

· Web UI

· MIT App Inventor

· Python code

7.2 Feature 2

· Login

· Verification

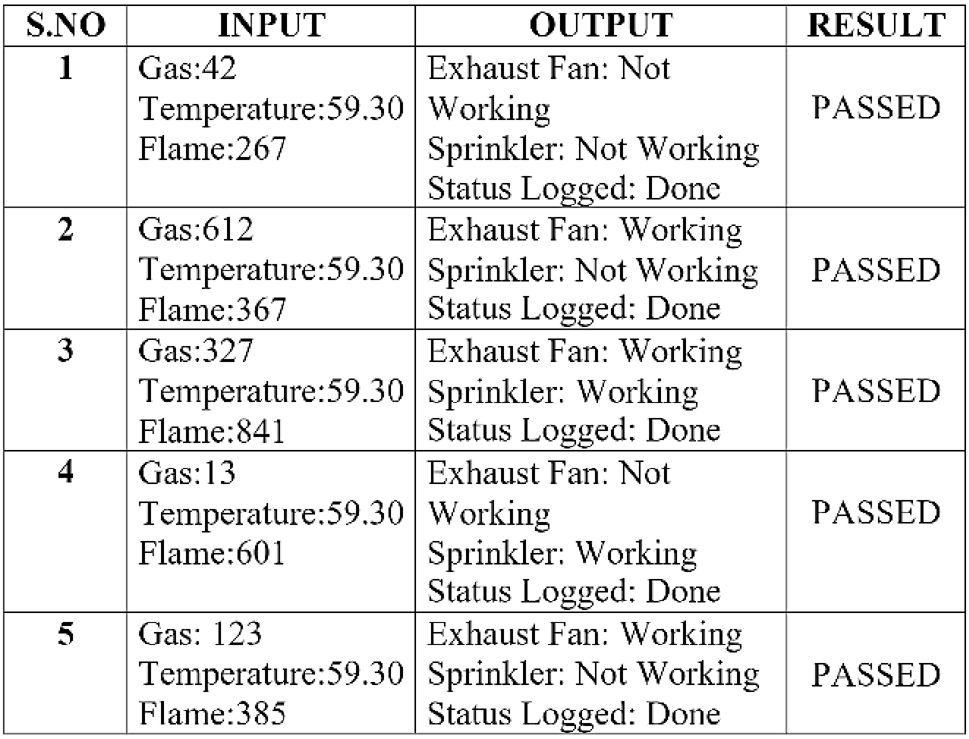
· Ticket Booking

· Adding rating

8. TESTING AND RESULTS

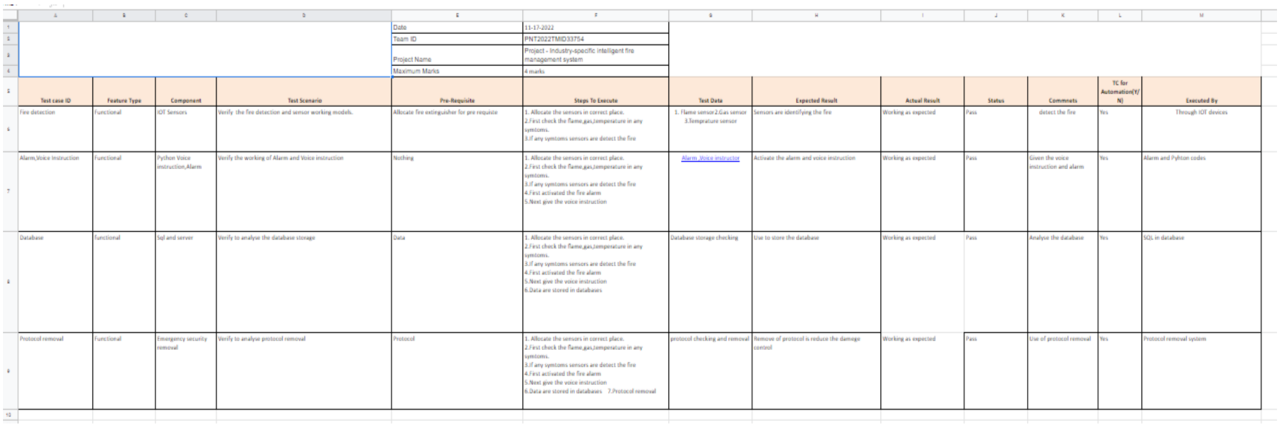
8.1 Test Cases

test case 1:



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

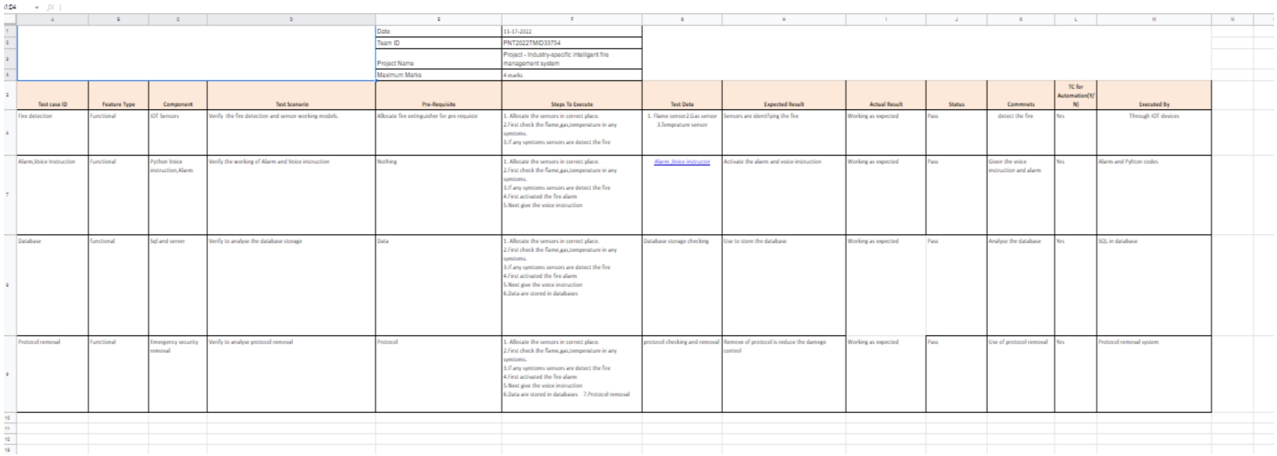
Test case 2



Date	11-17-2022
Team ID	PNT2022TMID33754
Project Name	Project - Industry-specific intelligent fire management system
Maximum Marks	4 marks

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	TC for Automation(Y/N)	Executed By
Fire detection	Functional	IOT Sensors	Verify the fire detection and sensor working models.	Allocate fire extinguisher for pre requiste	1. Allocate the sensors in correct place. 2.First check the flame,gas,temperature in any symtoms. 3.If any symtoms sensors are detect the fire	1. Flame sensor2.Gas sensor 3.Temprature sensor	Sensors are identifying the fire	Working as expected	Pass	detect the fire	Yes	Through IOT devices
Alarm,Voice Instruction	Functional	Python Voice instruction,Alarm	Verify the working of Alarm and Voice instruction	Nothing	1. Allocate the sensors in correct place. 2.First check the flame,gas,temperature in any symtoms. 3.If any symtoms sensors are detect the fire 4.First activated the fire alarm 5.Next give the voice instruction	Alarm ,Voice instructor	Activate the alarm and voice instruction	Working as expected	Pass	Given the voice instruction and alarm	Yes	Alarm and Pyhton codes
Database	functional	Sql and server	Verify to analyse the database storage	Data	1. Allocate the sensors in correct place. 2.First check the flame,gas,temperature in any symtoms. 3.If any symtoms sensors are detect the fire 4.First activated the fire alarm 5.Next give the voice instruction 6.Data are stored in databases	Database storage checking	Use to store the database	Working as expected	Pass	Analyse the database	Yes	SQL in database
Protocol removal	Functional	Emergency security removal	Verify to analyse protocol removal	Protocol	1. Allocate the sensors in correct place. 2.First check the flame,gas,temperature in any symtoms. 3.If any symtoms sensors are detect the fire 4.First activated the fire alarm 5.Next give the voice instruction 6.Data are stored in databases 7.Protocol removal	protocol checking and removal	Remove of protocol is reduce the damege control	Working as expected	Pass	Use of protocol removal	Yes	Protocol removal system

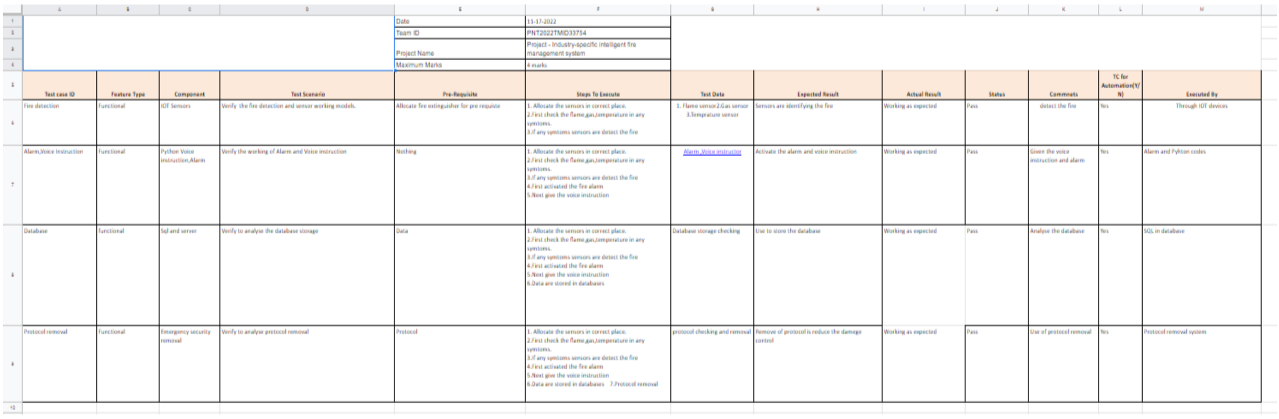
Test case 3



Date	11-17-2022
Team ID	PNT2022TMID33754
Project Name	Project - Industry-specific intelligent fire management system
Maximum Marks	4 marks

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	TC for Automation(Y/N)	Executed By
Fire detection	Functional	IOT Sensors	Verify the fire detection and sensor working models.	Allocate fire extinguisher for pre requiste	1. Allocate the sensors in correct place. 2.First check the flame,gas,temperature in any symtoms. 3.If any symtoms sensors are detect the fire	1. Flame sensor2.Gas sensor 3.Temprature sensor	Sensors are identifying the fire	Working as expected	Pass	detect the fire	Yes	Through IOT devices
Alarm,Voice Instruction	Functional	Python Voice instruction,Alarm	Verify the working of Alarm and Voice instruction	Nothing	1. Allocate the sensors in correct place. 2.First check the flame,gas,temperature in any symtoms. 3.If any symtoms sensors are detect the fire 4.First activated the fire alarm 5.Next give the voice instruction	Alarm ,Voice instructor	Activate the alarm and voice instruction	Working as expected	Pass	Given the voice instruction and alarm	Yes	Alarm and Pyhton codes
Database	functional	Sql and server	Verify to analyse the database storage	Data	1. Allocate the sensors in correct place. 2.First check the flame,gas,temperature in any symtoms. 3.If any symtoms sensors are detect the fire 4.First activated the fire alarm 5.Next give the voice instruction 6.Data are stored in databases	Database storage checking	Use to store the database	Working as expected	Pass	Analyse the database	Yes	SQL in database
Protocol removal	Functional	Emergency security removal	Verify to analyse protocol removal	Protocol	1. Allocate the sensors in correct place. 2.First check the flame,gas,temperature in any symtoms. 3.If any symtoms sensors are detect the fire 4.First activated the fire alarm 5.Next give the voice instruction 6.Data are stored in databases 7.Protocol removal	protocol checking and removal	Remove of protocol is reduce the damege control	Working as expected	Pass	Use of protocol removal	Yes	Protocol removal system

Test case 4



Date	11-17-2022
Team ID	PNT2022TMID33754
Project Name	Project - Industry-specific intelligent fire management system
Maximum Marks	4 marks

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	TC for Automation(Y/N)	Executed By
Fire detection	Functional	IOT Sensors	Verify the fire detection and sensor working models.	Allocate fire extinguisher for pre requiste	1. Allocate the sensors in correct place. 2.First check the flame,gas,temperature in any symtoms. 3.If any symtoms sensors are detect the fire	1. Flame sensor2.Gas sensor 3.Temprature sensor	Sensors are identifying the fire	Working as expected	Pass	detect the fire	Yes	Through IOT devices
Alarm,Voice Instruction	Functional	Python Voice instruction,Alarm	Verify the working of Alarm and Voice instruction	Nothing	1. Allocate the sensors in correct place. 2.First check the flame,gas,temperature in any symtoms. 3.If any symtoms sensors are detect the fire 4.First activated the fire alarm 5.Next give the voice instruction	Alarm ,Voice instructor	Activate the alarm and voice instruction	Working as expected	Pass	Given the voice instruction and alarm	Yes	Alarm and Pyhton codes
Database	functional	Sql and server	Verify to analyse the database storage	Data	1. Allocate the sensors in correct place. 2.First check the flame,gas,temperature in any symtoms. 3.If any symtoms sensors are detect the fire 4.First activated the fire alarm 5.Next give the voice instruction 6.Data are stored in databases	Database storage checking	Use to store the database	Working as expected	Pass	Analyse the database	Yes	SQL in database
Protocol removal	Functional	Emergency security removal	Verify to analyse protocol removal	Protocol	1. Allocate the sensors in correct place. 2.First check the flame,gas,temperature in any symtoms. 3.If any symtoms sensors are detect the fire 4.First activated the fire alarm 5.Next give the voice instruction 6.Data are stored in databases 7.Protocol removal	protocol checking and removal	Remove of protocol is reduce the damege control	Working as expected	Pass	Use of protocol removal	Yes	Protocol removal system

9. ADVANTAGES

- Easy to manage
- Medium cost
- Manage the large database

10. DISADVANTAGES

- If anyone is sensor is a damage the total experiment failure\

11. CONCLUSION

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

12. FUTURE SCOPE

This application is ensured for safety for the passengers while they are traveling alone as well as when they travel with their family or friends In future, this application may also be used by passengers who travel by bus. By further enhancement of the application the passengers can explore more features regarding their safety.

13. APPENDIX

13.1 Source Code :

```
<!DOCTYPE html>

<html>

<head>

 <title>Login</title>

 <meta charset="utf-8">

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

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

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

 <link href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-
```

```
awesome.min.css" rel="stylesheet">
 <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
</head>
<body>
 <div class="container-fluid">
        <div class="row ">
                <!-- IMAGE CONTAINER BEGIN -->
                <div class="col-lg-6 col-md-6 d-none d-md-block infinity-image-
container"></div>
                <!-- IMAGE CONTAINER END -->

                <!-- FORM CONTAINER BEGIN -->
                <div class="col-lg-6 col-md-6 infinity-form-container">
                        <!-- <h4 class = "test" style="color:#fa6903;">FIRE
MANAGEMENT</h4> -->
                        <div class="col-lg-9 col-md-12 col-sm-9 col-xs-12 infinity-form">
                                <!-- Company Logo -->
                                <div class="text-center mb-3 mt-5">
                                        <h4 class = "test" style="color:#fa6903;">FIRE
MANAGEMENT</h4>
                                </div>
                                <div class="text-center mb-4">
                    <h4>Login into your account</h4>
                   </div>
                   <!-- Form -->
                                <form class="px-3">
                                        <!-- Input Box -->
                                        <div class="form-input">
                                                <span><i class="fa fa-envelope-o"></i></span>
                                                <input type="email" name=""
placeholder="Email Address" tabindex="10"required>
                                        </div>
```

```
                         <div class="form-input">
                              <span><i class="fa fa-lock"></i></span>
                              <input type="password" name="" placeholder="Password" required>
                         </div>
                         <div class="row mb-3">
                         <!-- Remember Checkbox -->
              <div class="col-auto d-flex align-items-center">
               <div class="custom-control custom-checkbox">
                <input type="checkbox" class="custom-control-input" id="cb1">
                <label class="custom-control-label text-black" for="cb1" style = >Remember me</label>
               </div>
              </div>
                  </div>
                  <!-- Login Button -->
             <div class="mb-3">
                              <button type="submit" class="btn btn-block">Login</button>
                         </div>
                         <div class="text-right ">
              <a href="reset.html" class="forget-link">Forgot password?</a>
             </div>
                    </form>
               </div>
          </div>
          <!-- FORM CONTAINER END -->
     </div>
 </div>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
 <title>Reset</title>
 <meta charset="utf-8">
 <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
 <link href="style.css" rel="stylesheet" type="text/css">
 <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css">
 <link href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css" rel="stylesheet">
 <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
</head>
<body>
 <div class="container-fluid">
        <div class="row">
                <!-- IMAGE CONTAINER BEGIN -->
                <div class="col-lg-6 col-md-6 d-none d-md-block infinity-image-container"></div>
                <!-- IMAGE CONTAINER END -->


                <!-- FORM CONTAINER BEGIN -->
                <div class="col-lg-6 col-md-6 infinity-form-container">
                        <div class="col-lg-8 col-md-12 col-sm-8 col-xs-12 infinity-form">
                                <div class="text-center mb-3 mt-5">
                                        <img src="ps.webp" width="150px">
                                </div>
        <div class="reset-form d-block">
                           <form class="reset-password-form px-3">
                            <h4 class="mb-3" style="color:#fa6903">Reset Your password</h4>
```

```
                               <p class="mb-3 text-black">
                       Please enter your email address and we will send you a password reset link.
                      </p>
                      <div class="form-input">
                                           <span><i class="fa fa-envelope"></i></span>
                                           <input type="email" name="" placeholder="Email Address" tabindex="10"required>
                                     </div>
                      <div class="mb-3">
                                           <button type="submit" class="btn" >Send Reset Link</button>
                                     </div>
                     </form>
                    </div>

                    <div class="reset-confirmation d-none px-3">
                         <div class="mb-4">
                               <h4 class="mb-3">Link was sent</h4>
                         <h6 class="text-white">Please, check your inbox for a password reset link.</h6>
                         </div>

                         <a href="login.html">
                               <button type="submit" class="btn">Login Now</button>
                               </a>
                         <div>
                   </div>
             </div>
             <!-- FORM CONTAINER END -->
```

```
           </div>
 </div>

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

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

</body>
</html>
#include <time.h>
#include <WiFi.h>
#include <PubSubClient.h>

#define ORG "wt19pm"
#define DEVICE_TYPE "NodeMCU"
#define DEVICE_ID "12345"
#define TOKEN "12345678"

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/data/fmt/json";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;

WiFiClient wifiClient;
PubSubClient client(server, 1883, wifiClient);

float temperature  = 0;
int gas = 0;
int flame = 0;

String flame_status = "";
String Gas_status = "";
String exhaust_fan_status = "";
String sprinkler_status = "";


void setup() {
  Serial.begin(99900);
   wifiConnect();
   mqttConnect();
}

void loop() {

  srand(time(0));

    //initial variables and random generated data

    temperature = random(-20,125);
    gas = random(0,1000);
    int flamereading = random(200,1024);
    flame = map(flamereading,200,1024,0,2);
```

```
//set a flame status

switch (flame) {
case 0:
   flame_status = "No Fire";
   break;
case 1:
   flame_status = "Fire is Detected";
   break;
}

//send the sprinkler status

if(flame==1){
   sprinkler_status = "Working";
}
else{
   sprinkler_status = "Not Working";

}

//toggle the fan according to gas reading

if(gas > 100){
   Gas_status = "Gas Leakage is Detected";
   exhaust_fan_status = "Working";

}
else{
   Gas_status = "No Gas Leakage is Detected";
   exhaust_fan_status = "Not Working";
}
```

```
//json format for IBM Watson

String payload = "{";
payload+="\"gas\":";
payload+=gas;
payload+=",";
payload+="\"temperature\":";
payload+=(int)temperature;
payload+=",";
payload+="\"flame\":";
payload+=flamereading;
payload+=",";
payload+="\"fire_status\":\""+flame_status+"\",";
payload+="\"sprinkler_status\":\""+sprinkler_status+"\",";
payload+="\"Gas_status\":\""+Gas_status+"\",";
payload+="\"exhaust_fan_status\":\""+exhaust_fan_status+"\"}";

if(client.publish(publishTopic, (char*) payload.c_str()))
{
   Serial.println("Publish OK");
}
else{
   Serial.println("Publish failed");
}
delay(1000);


if (!client.loop())
{
  mqttConnect();
  }
}


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

}


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

```
{
  "version": 1,
  "author": "PNT2022TMID51903",
  "editor": "wokwi",
  "parts": [ { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": -110.32, "left": 3.84, "attrs": {} } ],
  "connections": [ [ "esp:TX0", "$serialMonitor:RX", "", [] ], [ "esp:RX0", "$serialMonitor:TX", "", [] ] ]
}
```

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

#define ORG "wt19pm"
#define DEVICE_TYPE "NodeMCU"
#define DEVICE_ID "12345"
#define TOKEN "12345678"

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/data/fmt/json";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;

WiFiClient wifiClient;
PubSubClient client(server, 1883, wifiClient);

float temperature  = 0;
```

```
int gas = 0;
int flame = 0;

String flame_status = "";
String Gas_status = "";
String exhaust_fan_status = "";
String sprinkler_status = "";


void setup() {
  Serial.begin(99900);
   wifiConnect();
   mqttConnect();
}

void loop() {

  srand(time(0));

    //initial variables and random generated data

    temperature = random(-20,125);
    gas = random(0,1000);
    int flamereading = random(200,1024);
    flame = map(flamereading,200,1024,0,2);

    //set a flame status

    switch (flame) {
    case 0:
        flame_status = "No Fire";
        break;
    case 1:
```

```
   flame_status = "Fire is Detected";
   break;
}

//send the sprinkler status

if(flame==1){
   sprinkler_status = "Working";
}
else{
   sprinkler_status = "Not Working";

}

//toggle the fan according to gas reading

if(gas > 100){
   Gas_status = "Gas Leakage is Detected";
   exhaust_fan_status = "Working";

}
else{
   Gas_status = "No Gas Leakage is Detected";
   exhaust_fan_status = "Not Working";
}

//json format for IBM Watson

String payload = "{";
payload+="\"gas\":";
payload+=gas;
payload+=",";
payload+="\"temperature\":";
```

```
  payload+=(int)temperature;
  payload+=",";
  payload+="\"flame\":";
  payload+=flamereading;
  payload+=",";
  payload+="\"fire_status\":\""+flame_status+"\",";
  payload+="\"sprinkler_status\":\""+sprinkler_status+"\",";
  payload+="\"Gas_status\":\""+Gas_status+"\",";
  payload+="\"exhaust_fan_status\":\""+exhaust_fan_status+"\"}";

  if(client.publish(publishTopic, (char*) payload.c_str()))
  {
    Serial.println("Publish OK");
  }
  else{
    Serial.println("Publish failed");
  }
  delay(1000);


  if (!client.loop())
  {
   mqttConnect();
  }
}

void wifiConnect()
{
 Serial.print("Connecting to ");
 Serial.print("Wifi");
```

```
  WiFi.begin("Wokwi-GUEST", "", 6);
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  }
  Serial.print("WiFi connected, IP address: ");
  Serial.println(WiFi.localIP());


}


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

    Serial.println();
  }
}
include <time.h>
#include <WiFi.h>
#include <PubSubClient.h>
```

```
#define ORG "wt19pm"
#define DEVICE_TYPE "NodeMCU"
#define DEVICE_ID "12345"
#define TOKEN "12345678"

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/data/fmt/json";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;

WiFiClient wifiClient;
PubSubClient client(server, 1883, wifiClient);

float temperature  = 0;
int gas = 0;
int flame = 0;

String flame_status = "";
String Gas_status = "";
String exhaust_fan_status = "";
String sprinkler_status = "";


void setup() {
  Serial.begin(99900);
   wifiConnect();
   mqttConnect();
}

void loop() {

  srand(time(0));
```

```
//initial variables and random generated data

temperature = random(-20,125);
gas = random(0,1000);
int flamereading = random(200,1024);
flame = map(flamereading,200,1024,0,2);

//set a flame status

switch (flame) {
case 0:
   flame_status = "No Fire";
   break;
case 1:
   flame_status = "Fire is Detected";
   break;
}

//send the sprinkler status

if(flame==1){
   sprinkler_status = "Working";
}
else{
   sprinkler_status = "Not Working";

}

//toggle the fan according to gas reading

if(gas > 100){
   Gas_status = "Gas Leakage is Detected";
   exhaust_fan_status = "Working";

}
else{
   Gas_status = "No Gas Leakage is Detected";
   exhaust_fan_status = "Not Working";
}

//json format for IBM Watson

String payload = "{";
payload+="\"gas\":";
payload+=gas;
payload+=",";
payload+="\"temperature\":";
payload+=(int)temperature;
payload+=",";
payload+="\"flame\":";
payload+=flamereading;
payload+=",";
payload+="\"fire_status\":\""+flame_status+"\",";
payload+="\"sprinkler_status\":\""+sprinkler_status+"\",";
payload+="\"Gas_status\":\""+Gas_status+"\",";
payload+="\"exhaust_fan_status\":\""+exhaust_fan_status+"\"}";

if(client.publish(publishTopic, (char*) payload.c_str()))
{
   Serial.println("Publish OK");
}
else{
   Serial.println("Publish failed");
```

```
    }
    delay(1000);



    if (!client.loop())
    {
      mqttConnect();
    }
}


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

}


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

    Serial.println();
  }
}
```

13.2 GitHub

DEMO VIDEO LINK:

https://drive.google.com/file/d/148qOYty0_DJcV-lVpbxWOZgXhFTLjEvj/view

GitHub link:

https://github.com/IBM-EPBL/IBM-Project-40815-1660635725