Industry-specific intelligent fire management system
Project Report

Team ID : PNT2022TMID45391

Team Size : 4

Team Leader : ARUL B

Team member : AHAMED HABEEB ANSARI V U

Team member : ARUN KUMAR R

Team member : YUVARAJA E

1. INTRODUCTION:

1.1 Project Overview:

Nowadays, fire incidents have become a critical issue , which must be dealt with on time without any unnecessary delay to avoid the loss in lives and belongings . It is considered a fire situation when the monitored temperature exceeds 50o C. In critical places such as hospitals, schools, and banks, personnel's arrival time to come for help in fire hazards is around 15 minutes . The statistics show that there are 475,500 structural fires annually in the United States, causing 2,950 civilian deaths, 12,775 civilian injuries, and $7.9 billion in property damage . According to the National Fire Protection Association (NFPA), two-third of U.S. household fires occur in premises with no working smoke alarms, alarms with no proper maintenance, or misplaced alarms . The appropriate allocation of fire alarms with a proactive warning could save lives and reduce property losses . Particularly, there are many types of fire alarms as heat detectors and smoke detectors ; studying these types helps to decide which type is more suitable for home or store. For instance, heat detectors are classic options when the temperature reaches a certain level. Thus, it is more suitable for applications that rapid response is not required or in an environment where smoke detectors cannot be placed like frozen areas. Heat detectors have a lower false alarm rate but still slower in response because the temperature rises slowly . With all these lacks, smoke detectors remain better than heat detectors Smoke alarms will more likely detect fires before it really starts. Smoke comes when the energy of an object is consumed due to the loss of Carbon Dioxide (CO2) from heat .

1.2 Purpose:

Smoke detectors are classified into three types: ionization, photoelectric, and combination. All these types can be studied further in instruments and measurement books. In this study, we will highlight a brief description of ionization. In brief, ionization is a radioactive material that receives radiation from the fire. It enters the ionization chamber, which is an air-filled space between two electrodes and permits a small, constant current between the electrodes. This type is the best for fast fires or fires caused by bombs or accidents the system analyses the collected data using Ubidots platform which results in a faster response. Thus, the highlighted four points make the proposed system superior in terms of affordability, effectiveness, and responsiveness.

2. LITERATURE SURVEY:

2.1 Existing problem:

They have proposed a prototype for a centralized management system for homes or offices which helps better in managing the safety features. In this, home management system is required. This system controls the room lights by turning on and off automatically, it keeps the record of use of electronic device status, turning on and off the ac regulator automatically, it displays the room temperature in home. If fire is detected in the house, it turn on sprinkler at home, it supervises at home via surveillance cameras, take photos and store them including recordings of surveillance at home, it detects the movements of people at home, and provide notification when someone enters the house

2.2 References:

1."Intelligent home management system prototype design and development", International Conference on Information Technology Systems and Innovation (ICITSI), IEEE, 2015..
Azka Ihsan Nurrahman

2.Developing a Fire Monitoring and Control System Based on IoT", Advances in Intelligent Systems Research..**Jianjun Yi2**

3.IoT Based Fire Alarm and Monitoring System," International Journal of Innovations ,September 2017..**Saumya Tiwari**

2.3 Problem Statement Definition:

There are only traditional fire alarms located in some pints inside the building which can be very useful to keep the students' lives but still not enough because it is now have become an old system and can made mistakes sometimes and that can be not save or will make bushing for nothing and it may cause study lateness.
Don't know the location of the fire when it happens One of the disadvantages to use some un improved system is that it will not show you all the aspects some them will not appear like needed information of the exact location of the starting point of fire. It is very important to have a device provides you the location otherwise no one will nowhere to go or how they should act against this situation they will simply try to exit from the exit doors but the way to the exit door must be clear and save to all people in the building.

3. IDEATION &PROPOSED SOLUTION:

3.1 Empathy Map Canvas:

Ideationphase Empathize&Discover

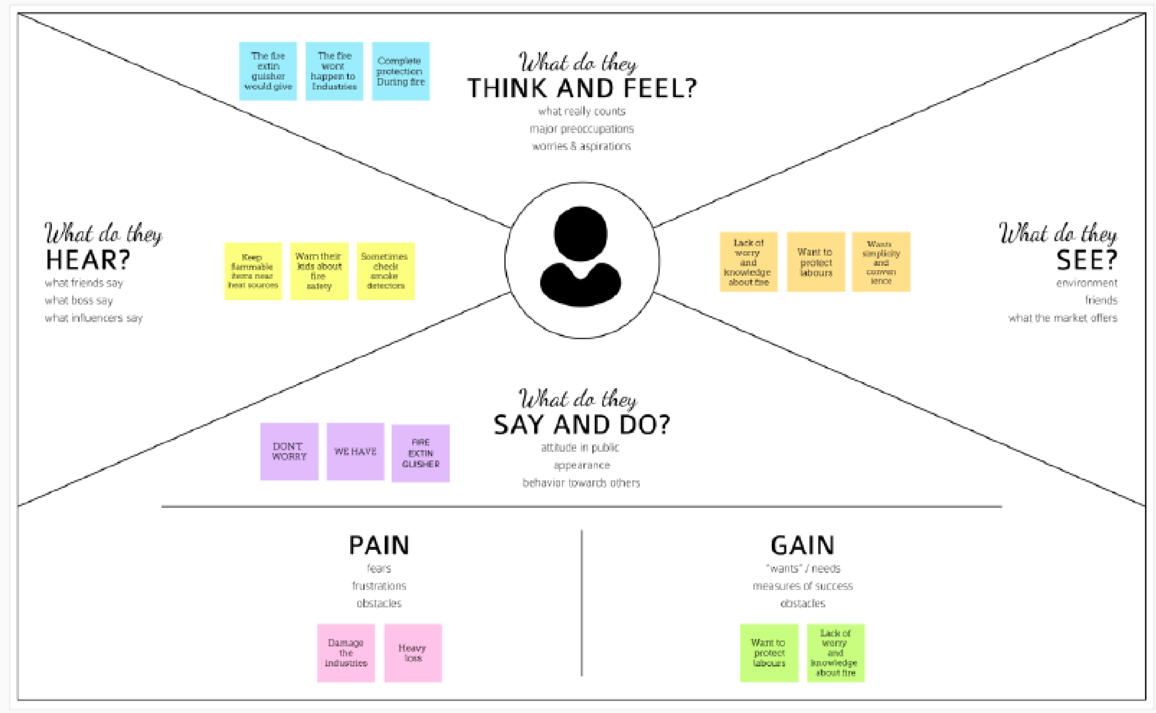
Date	19September 2022
Team ID	PNT2022TMID45391

Project Name	Project-Industry-specific intelligent firemanagement system
Maximum Marks	4Marks

Empathy Map Canvas: Industry-specific intelligent fire management system

EMPATHYMAP CANVAS

EMPATHY MAP CANVAS



3.2 Ideation &Brainstorming:

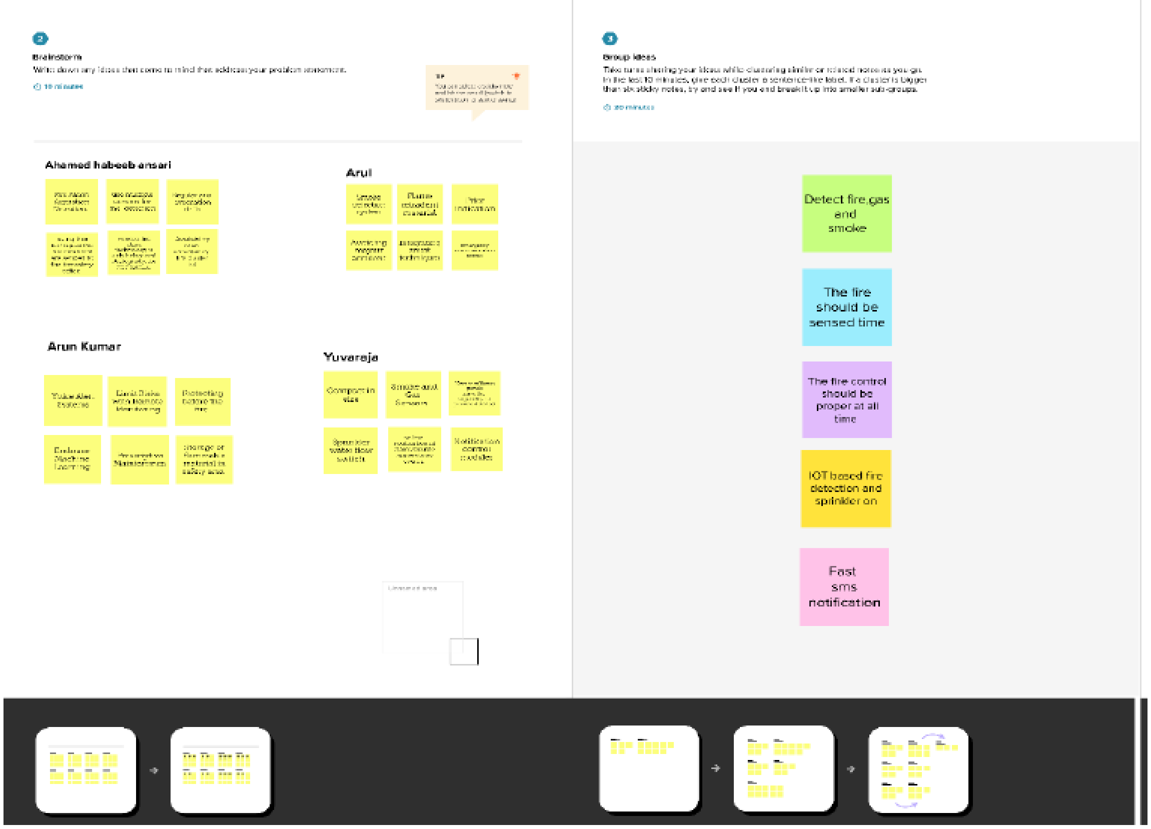
Brainstorm & Idea Prioritization

Date	19September 2022
Team ID	PNT2022TM1045391

Project Name	Project-Industry-specific intelligent fire management system
Maximum Marks	4Marks

Brainstorm & Idea Prioritization Template:

Step-1: Team Gathering, Collaboration and Select the Problem Statement


2
Brainstorm
Write down any ideas that come to mind that address your problem statement.
10 minutes
Ahamed habeeb ansari
Arul
Arun Kumar
Voice Alert Systems
Protecting before the fire
Enhance Machine Learning
Storage of flammable material in safety area.
Yuvaraja
Compact in size
Smoke and Gas Sensors
Sprinkler water flow switch
Notification control modules
3
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 can break it up into smaller sub-groups.
20 minutes
Detect fire,gas and smoke
The fire should be sensed time
The fire control should be proper at all time
IOT based fire detection and sprinkler on
Fast sms notification

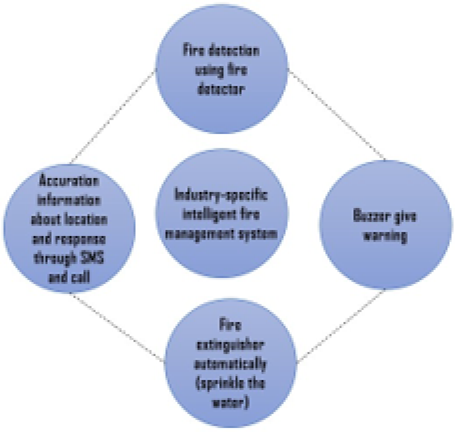
3.3 Proposed Solution:

Proposed Solution

Date	19 September 2022
Team ID	PNT2022TM1045391
Project Name	Project-Industry-specific intelligent fire management system

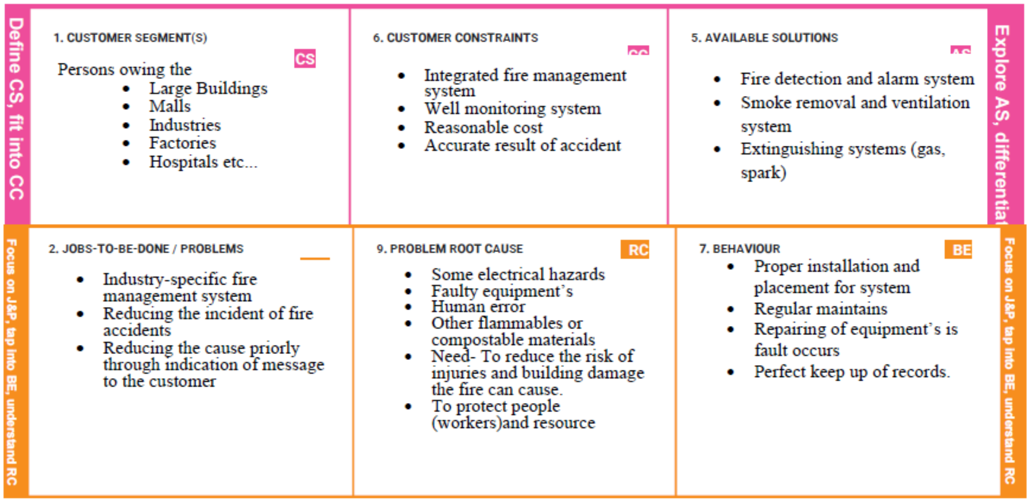
Maximum Marks	2 Marks

Proposed Solution Template:

S.No.	**Parameter**	**Description**
1.	Problem Statement (Problem to besolved)	To improvethe safety management system in industries. Improving the safety management system against the fire incidents in industries.
2.	Idea / Solution description	To implement the fire safety management in industry based on IOT using Arduino uno board with fire detection and fire extinguisher system. And using some sensors (Humidity sensor, Flame sensor, smoke sensor) with GPS tracking system.
3.	Novelty / Uniqueness	An integrated system of temperature monitoring, gas monitoring, fire detection automatically fire extinguisher with accusation of information about locations andresponse through SMS notification andcall.
4.	Social Impact/ Customer Satisfaction	■ It early prevents the accident cost byfire in industries. ■ Nearby locations so maximum extendmore accurate reliability. ■ Compatibility designintegrated system.
5.	Business Model(Revenue Model)	 Fire detection using fire detector Accuration information about location and response through SMS and call Industry-specific intelligent fire management system Buzzer give warning Fire extinguisher automatically (sprinkle the water)

6.	Scalability of the Solution	■ This project can be used more efficientlywith accurate information requiring. ■ Easy operability and maintenance. ■ Required low time for maintain ■ Cost is reasonable value.

3.4 Problem Solution fit:



Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Persons owing the

- Large Buildings
- Malls
- Industries
- Factories
- Hospitals etc...

6. CUSTOMER CONSTRAINTS CC

- Integrated fire management system
- Well monitoring system
- Reasonable cost
- Accurate result of accident

5. AVAILABLE SOLUTIONS AS

- Fire detection and alarm system
- Smoke removal and ventilation system
- Extinguishing systems (gas, spark)

Explore AS, differentia

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

2. JOBS-TO-BE-DONE / PROBLEMS

- Industry-specific fire management system
- Reducing the incident of fire accidents
- Reducing the cause priorly through indication of message to the customer

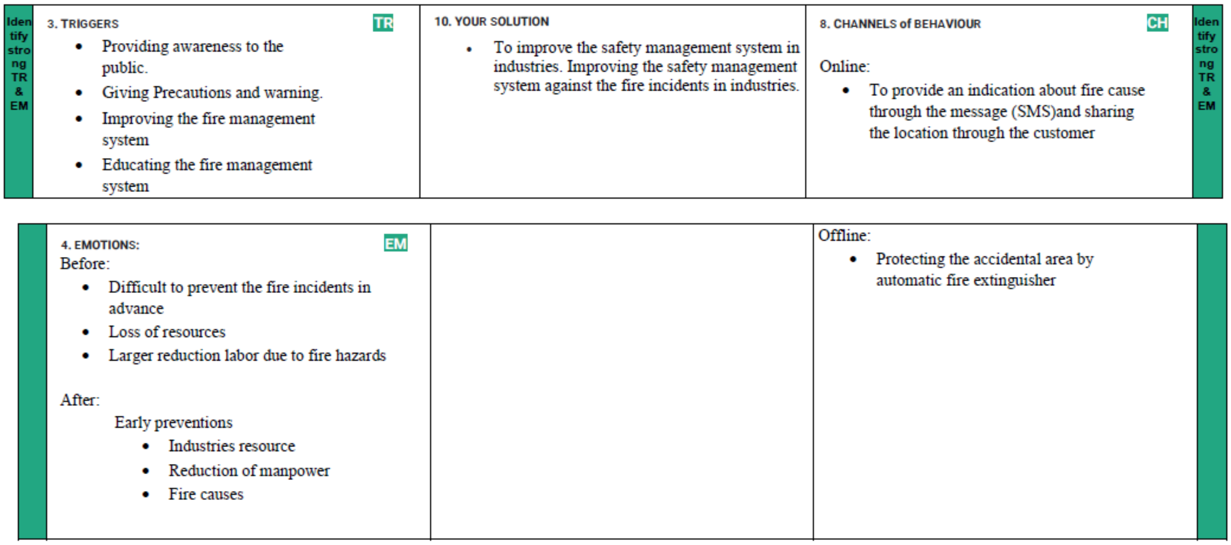
9. PROBLEM ROOT CAUSE RC

- Some electrical hazards
- Faulty equipment's
- Human error
- Other flammables or compostable materials
- Need- To reduce the risk of injuries and building damage the fire can cause.
- To protect people (workers)and resource

7. BEHAVIOUR BE

- Proper installation and placement for system
- Regular maintains
- Repairing of equipment's is fault occurs
- Perfect keep up of records.

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



Identify strong TR & EM

3. TRIGGERS TR

- Providing awareness to the public.
- Giving Precautions and warning.
- Improving the fire management system
- Educating the fire management system

10. YOUR SOLUTION

- To improve the safety management system in industries. Improving the safety management system against the fire incidents in industries.

8. CHANNELS of BEHAVIOUR CH

Online:

- To provide an indication about fire cause through the message (SMS)and sharing the location through the customer

Identify strong TR & EM

4. EMOTIONS: EM

Before:

- Difficult to prevent the fire incidents in advance
- Loss of resources
- Larger reduction labor due to fire hazards

After:

Early preventions

- Industries resource
- Reduction of manpower
- Fire causes

Offline:

- Protecting the accidental area by automatic fire extinguisher

4. REQUIREMENT ANALYSIS:

4.1 Functional requirement:

Solution Requirements (Functional & Non-functional)

Date	03 October 2022
Team ID	PNT2022TM1045391
Project Name	Project -Industry-specific intelligent firemanagement system
Maximum Marks	4 Marks

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story/ Sub-Task)
FR-1	User Registration	● Registration through Form ● Registration through Gmail ● Registration through LinkedIn
FR-2	User Confirmation	● Confirmation via Email ● Confirmation via OTP
FR-3	User Loginwith respectiveID	● Login through website ● App using the respective username andpassword
FR-4	User Access	● Access the app requirements
FR-5	User Upload	● User should be able to upload the data
FR-6	User Solution	● Data report should be generated ● Delivered to userfor every 24 hours
FR-7	User Data Sync	● API interface to increase to invoicesystem
FR-8	User Guide	● Guides the basic steps of usingtheapplication

4.2 Non-Functional requirements:

Non-functional Requirements:

Following are the non-functional requirements of the proposedsolution.

FR No.	**Non-Functional Requirement**	**Description**
NFR-1	**Usability**	● Low perceived workload. ● Easy and simpleUI. ● Usability requirements canconsiderlanguage barriers and localization tasks. ● Usability can be assessed fromthebelow functions. ● Efficiency of use.
NFR-2	**Security**	● Access permissions for the system information may onlybe changed by the system's data administrator.
NFR-3	**Reliability**	● The database updateprocess must rollback all related updates when any update fails.
NFR-4	**Performance**	● The front-page loadtime must be nomore than 2 seconds for users that access the website using an VoLTE mobileconnection.

NFR-5	**Availability**	● The rest of the pages that may experience problems must display anotification with a timer showing when the system is going to be up again. ● New module deployment mustn’t impactfront page, product pages, andcheckout pages availability and mustn’t take longerthan one hour.
NFR-6	**Scalability**	● We can increase scalability by addingmemory, servers, or disk space. On the other hand,we can compress data,use optimizing algorithms. ● The website attendance limit mustbe scalable enough to support 500,000users at a time.

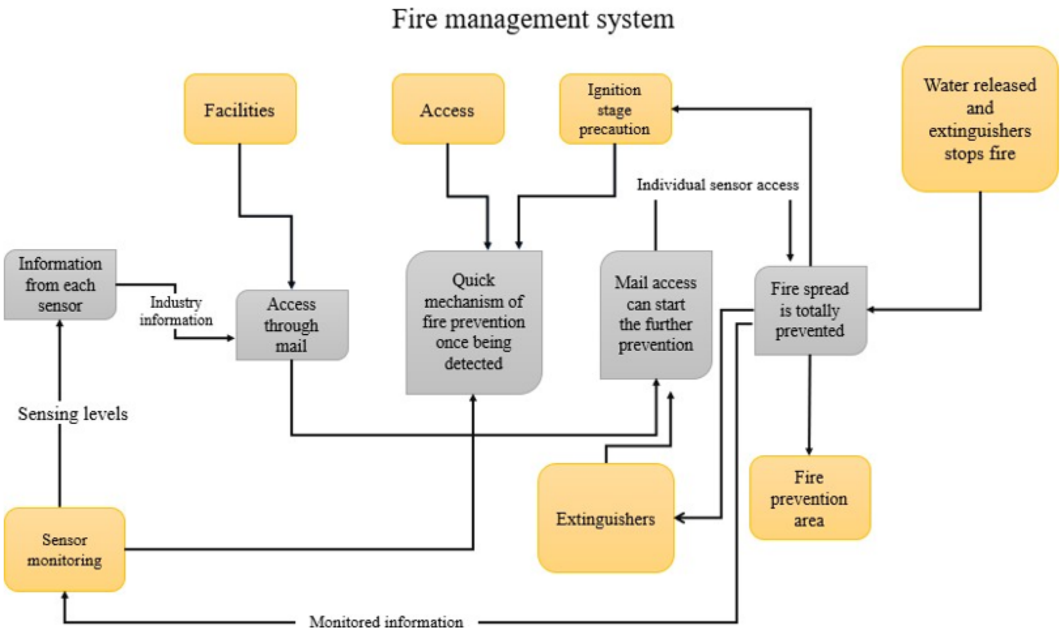
5. PROJECT DESIGN:

5.1 Data Flow Diagrams:

Data Flow Diagram & User Stories

Date	03October 2022
Team ID	PNT2022TM1045391
Project Name	Project - Industry-specific intelligent firemanagement system
Maximum Marks	4Marks

Data Flow Diagrams:

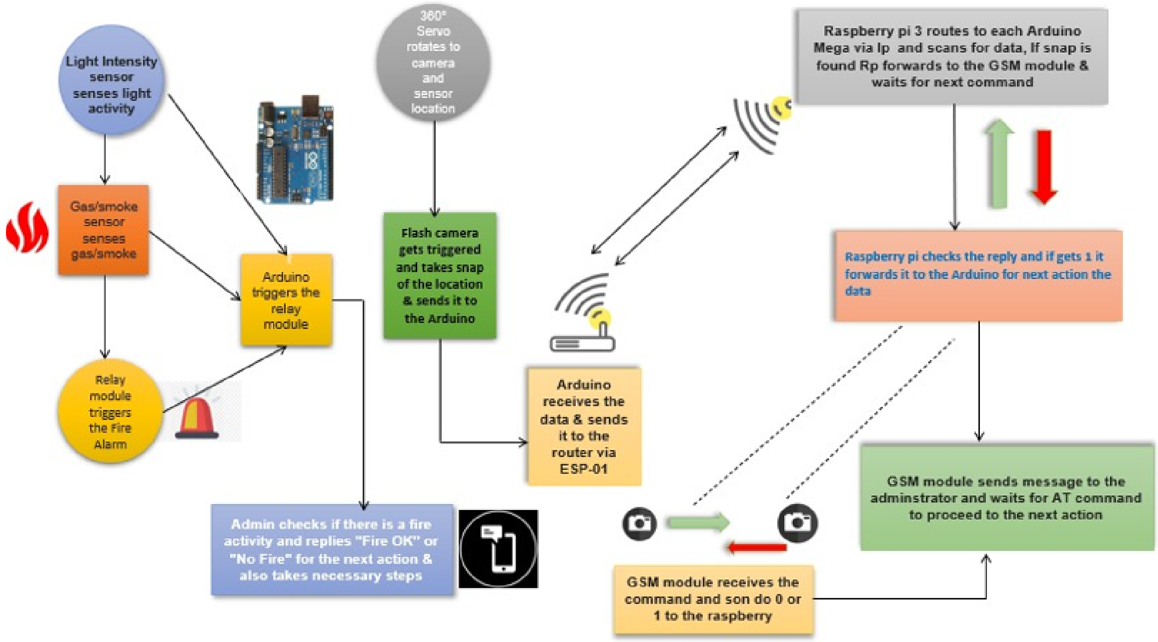

Fire management system
Facilities
Access
Ignition stage precaution
Water released and extinguishers stops fire
Individual sensor access
Information from each sensor
Industry information
Access through mail
Quick mechanism of fire prevention once being detected
Mail access can start the further prevention
Fire spread is totally prevented
Sensing levels
Extinguishers
Fire prevention area
Sensor monitoring
Monitored information

5.2 Solution & Technical Architecture:

Solution Architecture

Date	23September 2022
Team ID	PNT2022TM1045391
Project Name	Industry-specific intelligent firemanagement system
Maximum Marks	4Marks

Solution Architecture Diagram:


Light Intensity sensor senses light activity
360° Servo rotates to camera and sensor location
Raspberry pi 3 routes to each Arduino Mega via Ip and scans for data, If snap is found Rp forwards to the GSM module & waits for next command
Gas/smoke sensor senses gas/smoke
Arduino triggers the relay module
Flash camera gets triggered and takes snap of the location & sends it to the Arduino
Raspberry pi checks the reply and if gets 1 it forwards it to the Arduino for next action the data
Relay module triggers the Fire Alarm
Arduino receives the data & sends it to the router via ESP-01
GSM module sends message to the adminstrator and waits for AT command to proceed to the next action
Admin checks if there is a fire activity and replies "Fire OK" or "No Fire" for the next action & also takes necessary steps
GSM module receives the command and son do 0 or 1 to the raspberry

5.3 User Stories:

UserStories

Use the below templateto list all the user stories for the product.

User Type	Functional Requirement (Epic)	UserStory Number	User Story/ Task	Acceptance criteria	Priority	Release

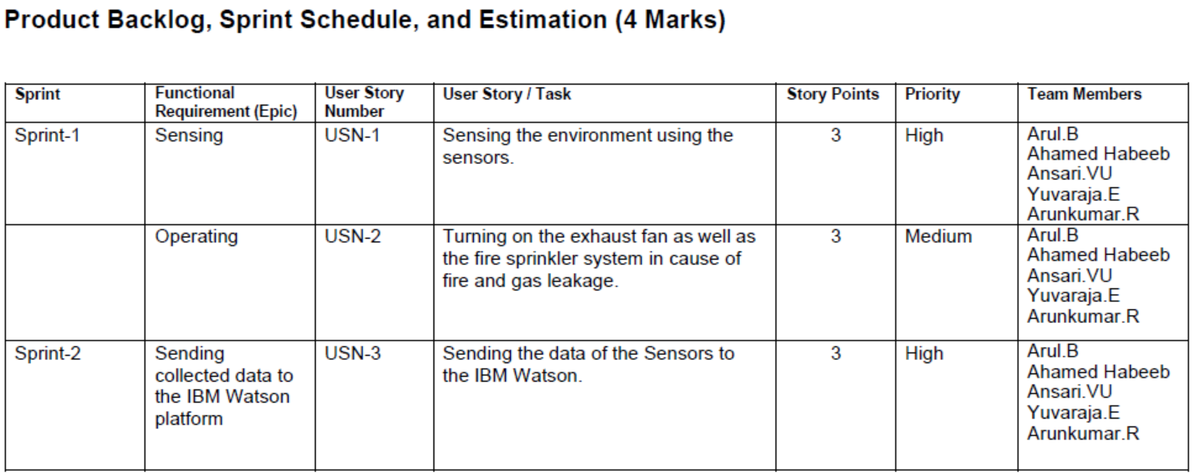
• Customer(Mobile user, Web user) • CustomerCare • Executive Administrator	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account/ dashboard	High	Sprint-1
		USN-2	As a user, I willreceive confirmation email once I have registered for the application	I can receive confirmation email & clickconfirm	High	Sprint-1

		USN-3	As a user, I can register for the application through Facebook	I can register &accessthe dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I canregister for the application through Gmail		Medium	Sprint-1
	Login	USN-5	As a user, I canlog into the application by entering email & password		High	Sprint-1

6. PROJECT PLANNING SCHEDULING:

6.1 Sprint Planning & Estimation:

Date	17 November 2022
Team ID	PNT2022TMID45391
Project Name	Project - INDUSTRY-SPECIFIC INTELLIGENT FIRE MANAGEMENT SYSTEM
Maximum Marks	8 Marks



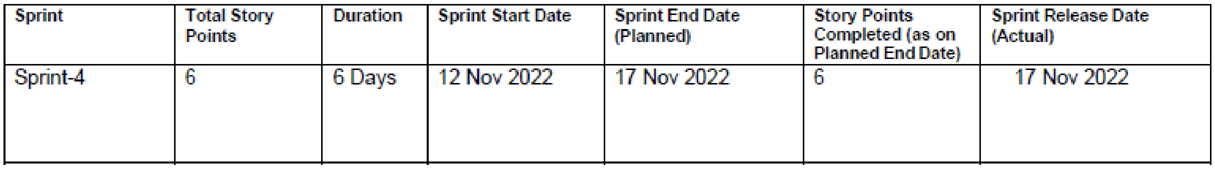
Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Sensing	USN-1	Sensing the environment using the sensors.	3	High	Arul.B Ahamed Habeeb Ansari.VU Yuvaraja.E Arunkumar.R
	Operating	USN-2	Turning on the exhaust fan as well as the fire sprinkler system in cause of fire and gas leakage.	3	Medium	Arul.B Ahamed Habeeb Ansari.VU Yuvaraja.E Arunkumar.R
Sprint-2	Sending collected data to the IBM Watson platform	USN-3	Sending the data of the Sensors to the IBM Watson.	3	High	Arul.B Ahamed Habeeb Ansari.VU Yuvaraja.E Arunkumar.R

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
	Node red	USN-4	Sending the data from the IBM Watson to the Node red.	3	High	Arul.B Ahamed Habeeb Ansari.VU Yuvaraja.E Arunkumar.R
Sprint-3	Storing of sensor data	USN-5	Storing in Cloudant database.	2	Medium	Arul.B Ahamed Habeeb Ansari.VU Yuvaraja.E Arunkumar.R
	Registration	USN-6	Entering my email and password to verify authentication process.	1	Medium	Arul.B Ahamed Habeeb Ansari.VU Yuvaraja.E Arunkumar.R
	Web UI	USN-7	Monitors the situation of the environment which displays sensor information.	3	High	Arul.B Ahamed Habeeb Ansari.VU Yuvaraja.E Arunkumar.R
Sprint-4	Fast SMS Service	USN-8	Use Fast SMS to Send alert message once the parameters like temperature, flame and gas sensor readings goes beyond the threshold value.	3	High	Arul.B Ahamed Habeeb Ansari.VU Yuvaraja.E Arunkumar.R
	Turn ON/OFF the actuators	USN-9	User can turn off the Exhaust fan as well as the sprinkler system If need in that Situation.	2	Medium	Arul.B Ahamed Habeeb Ansari.VU Yuvaraja.E Arunkumar.R
	Testing	USN-10	Testing of project and Final Deliverables.	1	Low	Arul.B Ahamed Habeeb Ansari.VU Yuvaraja.E Arunkumar.R

Project Tracker, Velocity & Burndown Chart: (4 Marks)

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	6	6 Days	25 SEPT 2022	30 SEPT 2022	6	30 SEPT 2022
Sprint-2	6	6 Days	03 Oct 2022	09 OCT 2022	6	09 OCT 2022
Sprint-3	6	6 Days	15 OCT 2022	30 OCT 2022	6	30 OCT 2022



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-4	6	6 Days	12 Nov 2022	17 Nov 2022	6	17 Nov 2022

6.2 Sprint Delivery Schedule:

Milestone and Activity List

Date	24 October 2022
Team ID	PNT2022TM1045391
Project Name	Project - Industry-specific intelligent fire management system
Maximum Marks	4 Marks

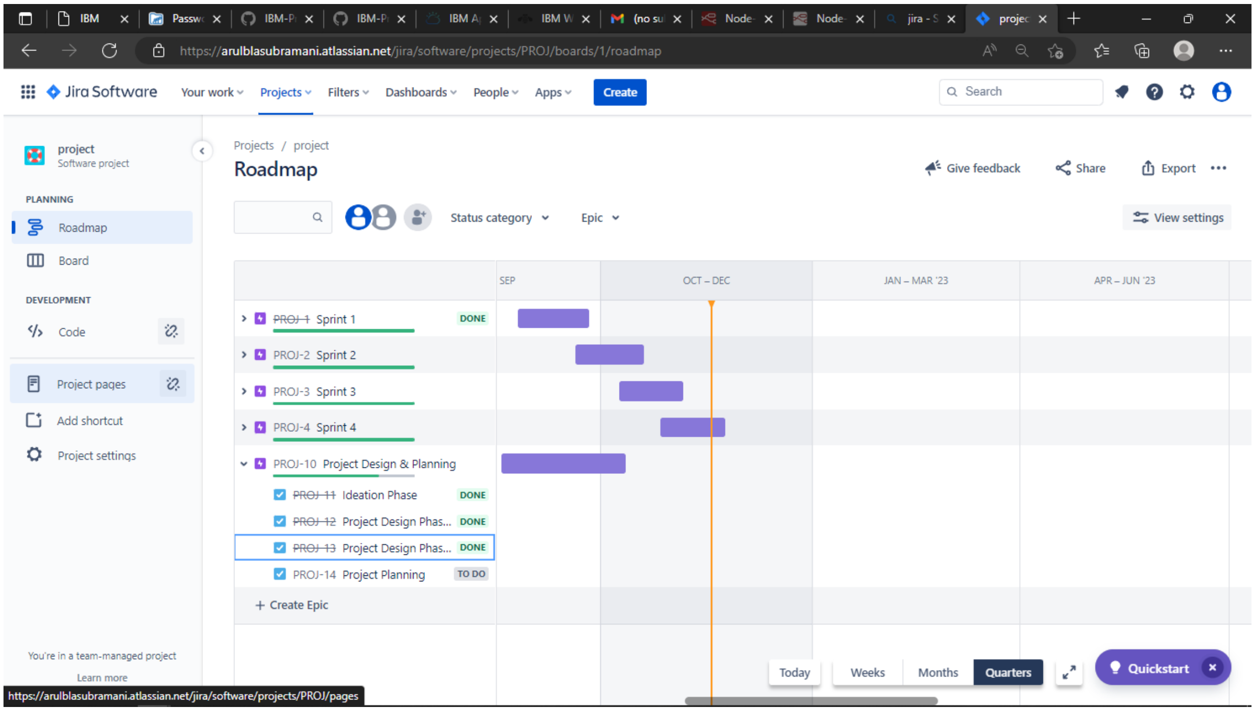
TITLE	DISCRIPTION	DATE
Literature Survey& Information Gathering	Literature survey on the selected project & gathering information by referring the, technical papers, research publications etc.	**19 SEPT 2022**
Prepare Empathy Map	Prepare Empathy Map Canvas to capture the user Pains & Gains, Prepare list of problem statements	**19 SEPT 2022**
Problem Statement	Prepare Problem statement of Industry-specific intelligent fire management system	**19 SEPT 2022**

Ideation	List the by organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility & importance.	**19 SEPT 2022**
Proposed Solution	Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.	**03 OCT 2022**
Problem Solution Fit	Prepare problem - solution fit document	**03 OCT 2022**
Solution Architecture	Prepare solution architecture document	**09 OCT 2022**
Customer Journey	Prepare the customer journey maps to understand the user interactions & experiences with the application (entry to exit).	**16 OCT 2022**
Functional Requirement	Prepare the functional requirement document.	**16 OCT 2022**
Data Flow Diagrams	Draw the data flow diagrams and submit for review	**16 OCT 2022**
Technology Architecture	Prepare the technology architecture diagram.	**16 OCT 2022**

Prepare Milestone & Activity List	Prepare the milestones & activity list of the project.	**10 NOV 2022**
Sprint Schedules	Prepare the sprint plan and divided tasks according to agile method	**10 NOV 2022**
Project DevelopmentDelivery Sprint - 1	Develop & submit the developed code by testing it.	**ON PROCESS**
Sprint - 2	Develop & submit the developed code by testing it	**ON PROCESS**
Sprint – 3	Develop & submit the developed code by testing it	**ON PROCESS**
Sprint – 4	Develop & submit the developed code by testing it	**ON PROCESS**

6.3 Reports from JIRA:

Team ID	PNT2022TMID45391
Project Name	Project -Industry-specific intelligent firemanagement system



https://arulblasubramani.atlassian.net/jira/software/projects/PROJ/boards/1/roadmap

Jira Software · Your work · Projects · Filters · Dashboards · People · Apps · Create · Search

project
Software project

PLANNING
- Roadmap
- Board

DEVELOPMENT
- Code
- Project pages
- Add shortcut
- Project settings

You're in a team-managed project
Learn more

Projects / project

Roadmap

Give feedback · Share · Export

Status category · Epic · View settings

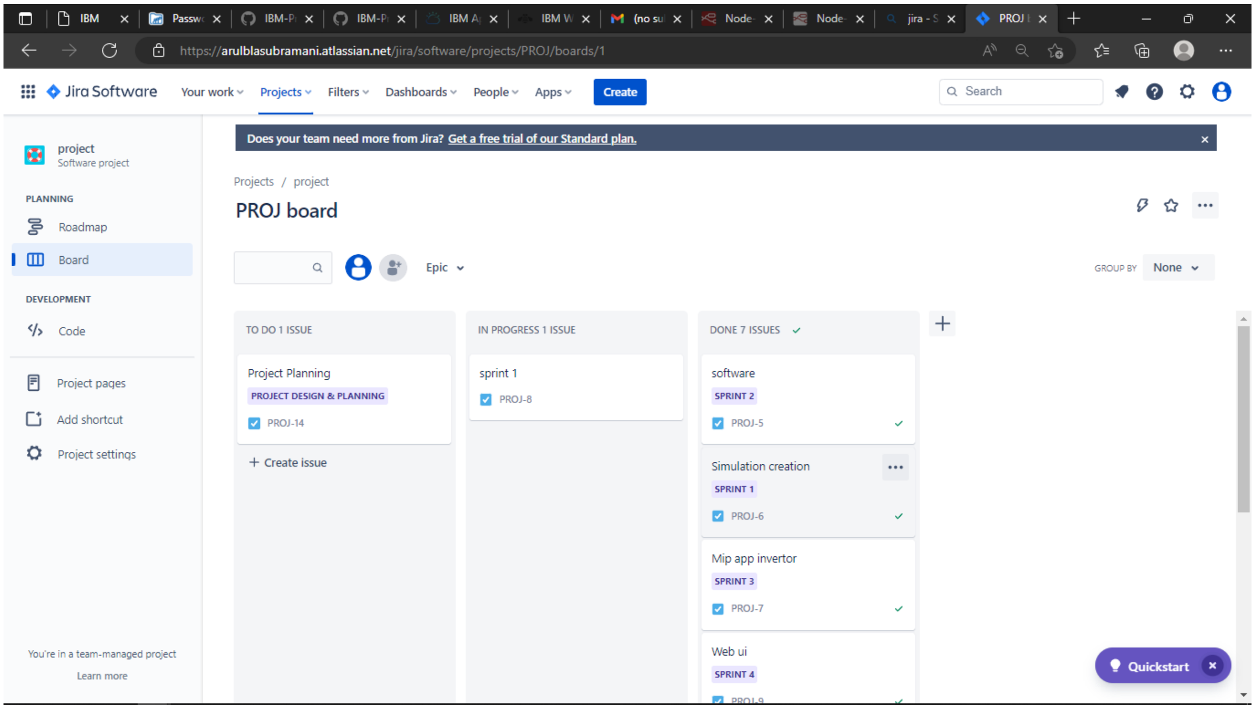
SEP · OCT – DEC · JAN – MAR '23 · APR – JUN '23

Issue	Summary	Status
~~PROJ-1~~	Sprint 1	DONE
PROJ-2	Sprint 2	
PROJ-3	Sprint 3	
PROJ-4	Sprint 4	
PROJ-10	Project Design & Planning	
~~PROJ-11~~	Ideation Phase	DONE
~~PROJ-12~~	Project Design Phas...	DONE
~~PROJ-13~~	Project Design Phas...	DONE
PROJ-14	Project Planning	TO DO

+ Create Epic

Today · Weeks · Months · Quarters · Quickstart

https://arulblasubramani.atlassian.net/jira/software/projects/PROJ/pages



https://arulblasubramani.atlassian.net/jira/software/projects/PROJ/boards/1

Jira Software · Your work · Projects · Filters · Dashboards · People · Apps · Create · Search

Does your team need more from Jira? Get a free trial of our Standard plan.

project
Software project

PLANNING
- Roadmap
- Board

DEVELOPMENT
- Code
- Project pages
- Add shortcut
- Project settings

You're in a team-managed project
Learn more

Projects / project

PROJ board

Epic · GROUP BY None

TO DO 1 ISSUE	IN PROGRESS 1 ISSUE	DONE 7 ISSUES ✓
Project Planning PROJECT DESIGN & PLANNING PROJ-14	sprint 1 PROJ-8	software SPRINT 2 PROJ-5 ✓
+ Create issue		Simulation creation SPRINT 1 PROJ-6 ✓
		Mip app invertor SPRINT 3 PROJ-7 ✓
		Web ui SPRINT 4 PROJ-9 ✓

Quickstart

7.CODING & SOLUTIONING (Explain the features added in the project along with code)

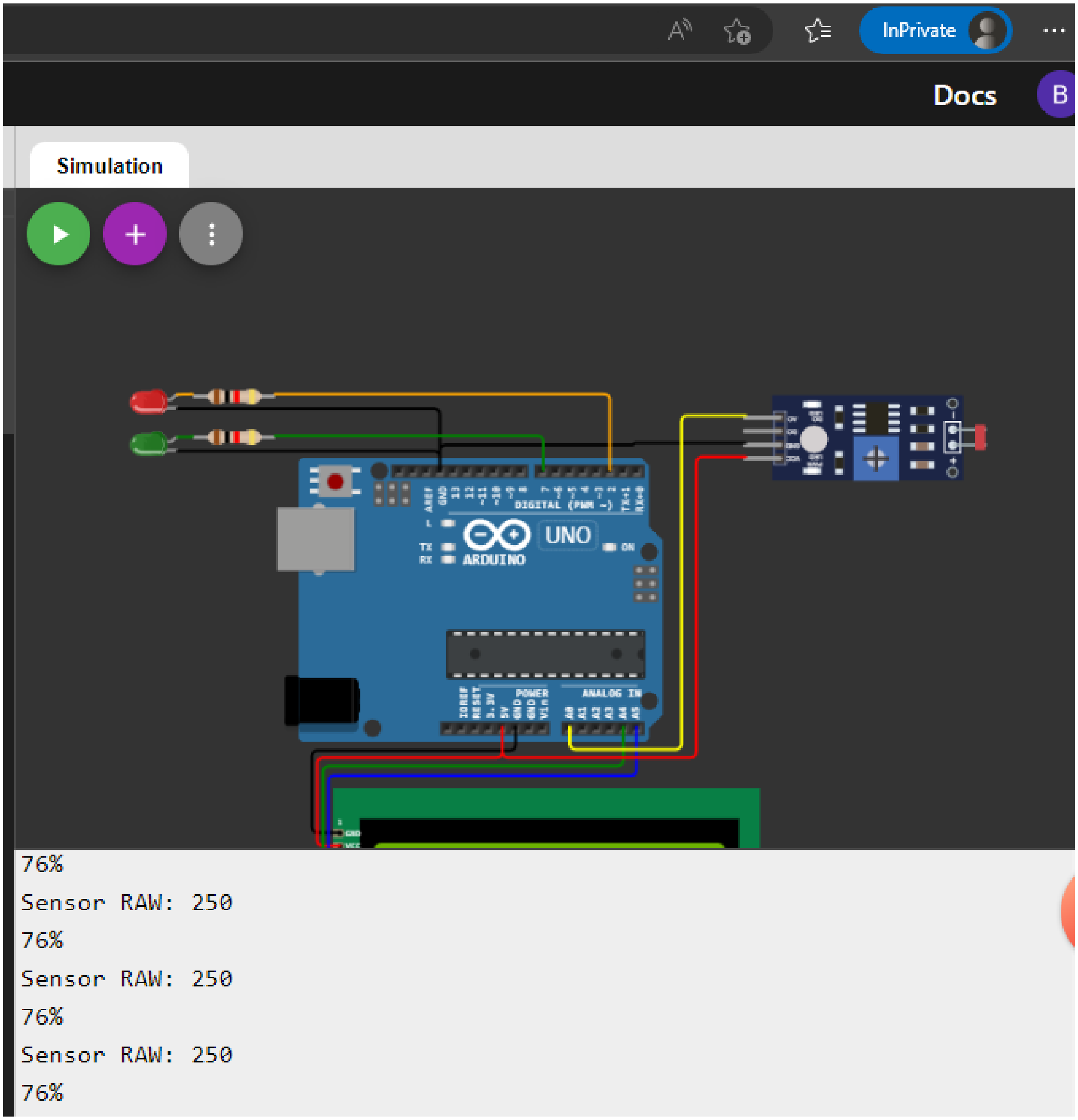
7.1Feature 1:

```
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);  //define I2C address 0x27, 16 column and 2 rows

float flamelevel = 0; // mapped and inverted % of sensor range
#define greenLED 7 // fire okay or firing
#define redLED 2 // warning or flashing alert

// empirically relate flame % to actual fire condition (TBD)
const int minSurvive = 15; // minimum level for idle, below is outfire
const int idleLow = 20; // lowest reading for healthy idle
const int idleTarget = 30; // target reading for resting idle
const int firingLow = 70; // lowest reading for actively firing
const int firingHigh = 90; // reading for full firing



void setup() {
  lcd.init();
  lcd.clear();
  lcd.backlight();
  Serial.begin(9600);

  pinMode(greenLED, OUTPUT); // set green pin led as output
  digitalWrite(greenLED, LOW); // turn off green led

  pinMode(redLED, OUTPUT); // set red led pin as output
  digitalWrite(redLED, LOW); // turn off red led
}

void loop() {
  float analogValue = analogRead(A0);
  Serial.print("Sensor RAW: ");
  Serial.println(analogValue, 0);
  flamelevel = map(analogValue, 0, 1024, 100, 0);
  Serial.print(flamelevel, 0);
  Serial.println("%");
```

```
// disabling the lcd commands makes serial print work
lcd.setCursor(0, 0);
lcd.print(F("Flame: "));

if (flamelevel >= firingHigh) { // stoker is fully firing
  lcd.print("Full Fire");
  digitalWrite(greenLED, HIGH); // turn on green led
  digitalWrite(redLED, LOW); // turn off red led
  delay(300);
  digitalWrite(greenLED, LOW); // turn off green led for flash
}

if ((flamelevel >= firingLow) && (flamelevel < firingHigh)) { // stoker is firing
  lcd.print("Firing   ");
  digitalWrite(greenLED, HIGH); // turn on green led
  digitalWrite(redLED, LOW); // turn off red led
}

if ((flamelevel < firingLow) && (flamelevel > idleLow) ) {  // idle fire
  lcd.print("Idle fire ");
  digitalWrite(greenLED, HIGH); // turn on green led
  digitalWrite(redLED, HIGH); // turn off red led
}

if ((flamelevel <= idleLow) && (flamelevel >= minSurvive) ) {  // low fire
  lcd.print("Low fire  ");
  digitalWrite(greenLED, LOW); // turn on green led
  digitalWrite(redLED, HIGH); // turn off red led
  // trigger stoker run timer = 2 mins?
}

if (flamelevel < minSurvive) {  // fire out
  lcd.print("FIRE OUT! ");
  digitalWrite(greenLED, LOW); // turn on green led
  digitalWrite(redLED, HIGH); // turn off red redLED
  delay(300);
  digitalWrite(redLED, LOW); // turn off red led for flash
  // send alert
}

lcd.setCursor(0, 1);
```

```
  lcd.print("   Level: ");
  lcd.print(flamelevel, 0);
  lcd.print("%    ");

  delay(200);
}
```

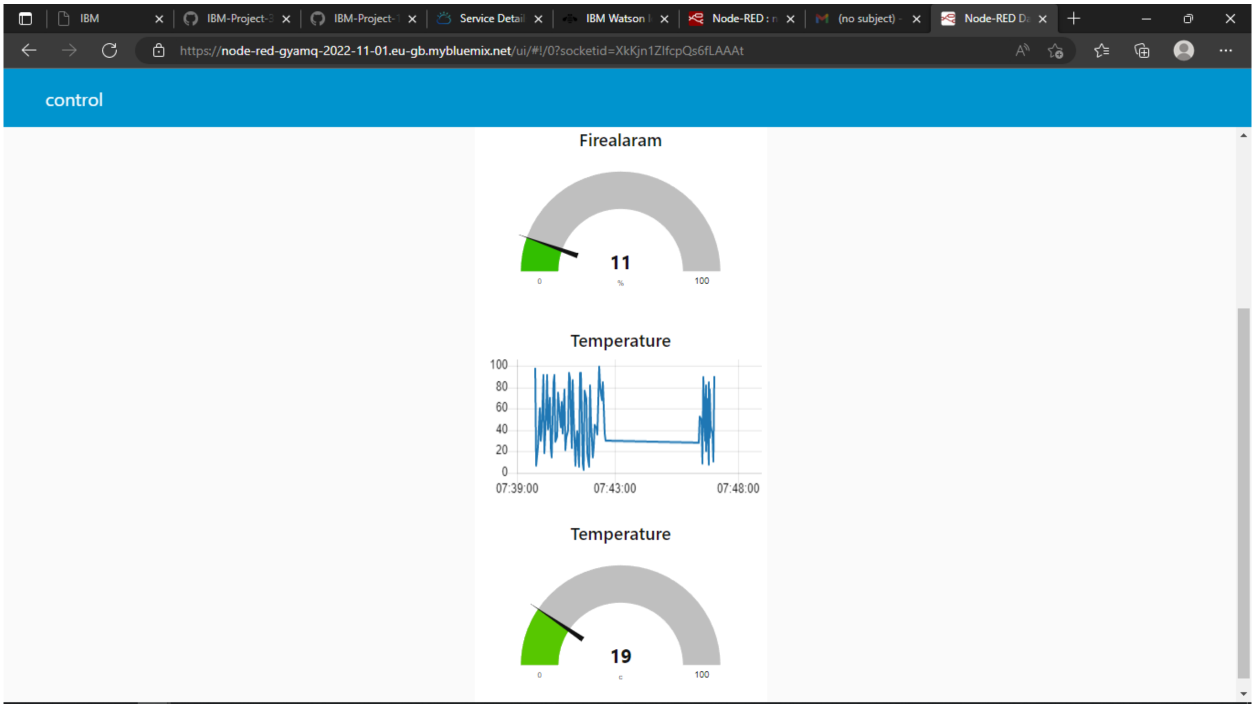
7.2:Feature 2:

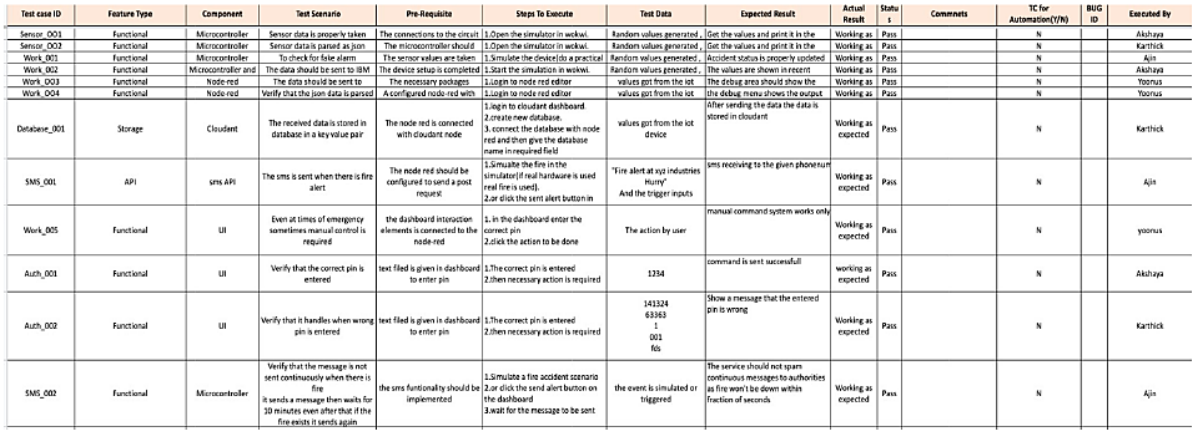


```
76%
Sensor RAW: 250
76%
Sensor RAW: 250
76%
Sensor RAW: 250
76%
```

8.TESTING:

8.1 Test Cases

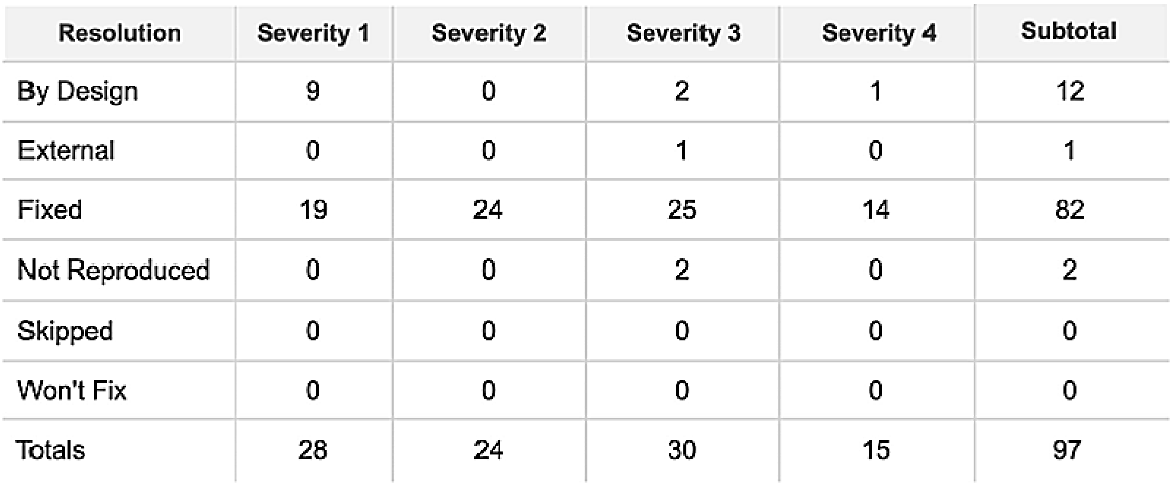




Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	TC for Automation(Y/N)	BUG ID	Executed By
Sensor_OO1	Functional	Microcontroller	Sensor data is properly taken	The connections to the circuit	1.Open the simulator in wokwi.	Random values generated ,	Get the values and print it in the	Working as	Pass		N		Akshaya
Sensor_OO2	Functional	Microcontroller	Sensor data is parsed as json	The microcontroller should	1.Open the simulator in wokwi.	Random values generated ,	Get the values and print it in the	Working as	Pass		N		Karthick
Work_001	Functional	Microcontroller	To check for fake alarm	The sensor values are taken	1.Simulate the device(do a practical	Random values generated ,	Accident status is properly updated	Working as	Pass		N		Ajin
Work_002	Functional	Microcontroller and	The data should be sent to IBM	The device setup is completed	1.Start the simulation in wokwi.	Random values generated ,	The values are shown in recent	Working as	Pass		N		Akshaya
Work_OO3	Functional	Node-red	The data should be sent to	The necessary packages	1.Login to node red editor	values got from the iot	The debug area should show the	Working as	Pass		N		Yoonus
Work_OO4	Functional	Node-red	Verify that the json data is parsed	A configured node-red with	1.Login to node red editor	values got from the iot	the debug menu shows the output	Working as	Pass		N		Yoonus
Database_001	Storage	Cloudant	The received data is stored in database in a key value pair	The node red is connected with cloudant node	1.login to cloudant dashboard. 2.create new database. 3. connect the database with node red and then give the database name in required field	values got from the iot device	After sending the data the data is stored in cloudant	Working as expected	Pass		N		Karthick
SMS_001	API	sms API	The sms is sent when there is fire alert	The node red should be configured to send a post request	1.Simualte the fire in the simulator(if real hardware is used real fire is used). 2.or click the sent alert button in	"Fire alert at xyz industries Hurry" And the trigger inputs	sms receiving to the given phonenum	Working as expected	Pass		N		Ajin
Work_005	Functional	UI	Even at times of emergency sometimes manual control is required	the dashboard interaction elements is connected to the node-red	1. in the dashboard enter the correct pin 2.click the action to be done	The action by user	manual command system works only	Working as expected	Pass		N		yoonus
Auth_001	Functional	UI	Verify that the correct pin is entered	text filed is given in dashboard to enter pin	1.The correct pin is entered 2.then necessary action is required	1234	command is sent successfull	working as expected	Pass		N		Akshaya
Auth_002	Functional	UI	Verify that it handles when wrong pin is entered	text filed is given in dashboard to enter pin	1.The correct pin is entered 2.then necessary action is required	141324 63363 1 001 fds	Show a message that the entered pin is wrong	Working as expected	Pass		N		Karthick
SMS_002	Functional	Microcontroller	Verify that the message is not sent continuously when there is fire it sends a message then waits for 10 minutes even after that if the fire exists it sends again	the sms funtionality should be implemented	1.Simulate a fire accident scenario 2.or click the send alert button on the dashboard 3.wait for the message to be sent	the event is simulated or triggered	The service should not spam continuous messages to authorities as fire won't be down within fraction of seconds	Working as expected	Pass		N		Ajin

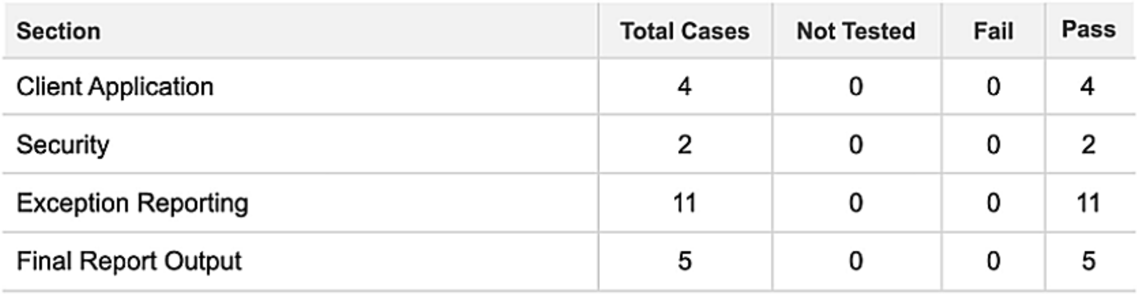
8.2User Acceptance Testing

Defect Analysis



Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	9	0	2	1	12
External	0	0	1	0	1
Fixed	19	24	25	14	82
Not Reproduced	0	0	2	0	2
Skipped	0	0	0	0	0
Won't Fix	0	0	0	0	0
Totals	28	24	30	15	97

Test Case Analysis



Section	Total Cases	Not Tested	Fail	Pass
Client Application	4	0	0	4
Security	2	0	0	2
Exception Reporting	11	0	0	11
Final Report Output	5	0	0	5

9.RESULTS:

9.1Performance Metrics

CPU Usage:

The micro version of C++ makes the most efficient use of the CPU. The program runs in O(1) time for each loop, ignoring the network and communication. To improve communication with MQTT, the program sleeps every 1 second. Because the program runs in O(1) time and the compiler optimizes it during compilation, there is less CPU load per cycle. The following instructions are stored on the stack memory and can be popped after execution.

Memory Usage:

The sensor values and networking data are saved in the ESP32's sram. It's a lot of information because the ESP32 only has 520 KB of memory. To save memory and ensure optimal program execution, the exact addresses for each memory cycle are overwritten with new values.

Error Rates:

The error rates are very low because the backend and dashboard are handled with node-red. Exceptions are handled properly so that the system's usability is not affected.

Latency and Respose Time:

The DOM handling of the received data is optimal and latency is low .After the DOM is loaded the entire site is loaded to the browser.

10.ADVANTAGES & DISADVANTAGES:

ADVANTAGES :

- Safety Benefits
- Quicker Response Times Minimisation of Costs
- Integration of Systems
- Minimisation of Unnecessary Business Interruptions
- Self Monitoring System

DISADVANTAGES:

- The system is essentially useless if the batteries aren’t charged, since it won’t work properly.
- There is a bit of a burden to homeowners or business owners to always remember to keep the batteries fresh so the system operates properly when you need it most
- Range can be a problem for large offices or homes, since a weak wireless connection may cause the system to not operate reliably.
- Wireless fire alarm systems also don’t connect directly to the telephone lines,
- which are linked to the fire departments, so the response to an emergency could be slower as a result.

11.CONCLUSION:

In this project an attempt has been made for making IOT based fire alarm system to identify fire accident and pass on the alert message to the authenticated user in the industry. By analysing the literature review of various authors, this system is efficient to So we conclude that, our problem premise is solved using IoT devices by developing a smart management system that solves many inherent problems in traditional fire management systems, such asactively monitoring for fire breakouts and gas leakage and sending SMS alertsadministrators and fire authorities.

12.FUTURE SCOPE:

1.We implemented smoke detect, but we want to thief detect before any stolen.
2.To make the circuit complex free and less expensive.
3.To implement the circuit by applying latest version of the renown software.
4.Minimising the transistor.
5.To protect our home, office, University, industry etc. automatically without any person when fire occurred.

13.APPENDIX:

DHT22 - Temperature & Humidity Sensor:

The DHT22 is a simple and inexpensive digital temperature and humidity sensor. It measures the surrounding air with a capacitive humidity sensor and a thermistor and outputs a digital signal on the datapin (no analog input pins needed).

Flow Sensors:

A flow sensor (also known as a "flow meter") is an electronic device that measures or controls the flow rate of liquids and gases through pipes and tubes.

13.1 Source Code:

```
#include <WiFi.h>//library for wifi

#include <PubSubClient.h>//library for MQtt

#include "DHT.h"// Library for dht11

#include <cstdlib>

#include <time.h>

#include <mjson.h>

#define DHTPIN 15 // what pin we're connected to

#define DHTTYPE DHT22 // define type of sensor DHT 11

DHT dht (DHTPIN, DHTTYPE); // creating the instance by passing pin and typr of dht connected

void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);
```

```
//-------credentials of IBM Accounts------
#define ORG "4sm1u8"
#define DEVICE_TYPE "arul0906"
#define DEVICE_ID "traingingid"
#define TOKEN "AQ6gu0dvFLO+If!FKW"
String data3 = "";
String accidentstatus ="";
String sprinkstatus = "";
float temp =0;
bool isfanon = false;
bool issprinkon = false;
bool cansprinkoperate = true;
bool canfanoperate = true;
```

29

```
bool cansentalert = false;
int gas = 0;
int flame = 0;
int flow = 0;
long int cooldown= 600;
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/data/fmt/json";
char subscribetopic[] = "iot-2/cmd/command/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
//-----------------------------------------
WiFiClient wifiClient; // creating the instance for wificlient
```

```
PubSubClient client(server, 1883, callback ,wifiClient); //calling the predefined client id by passing
parameter like server id,portand wificredential
void setup()// configureing the ESP32
{
Serial.begin(115200); dht.begin();
//if real gas sensor is used make sure the senor is heated up for acurate readings
/*
- Here random values for readings and stdout were used to show the
working of the devices as physical or simulated devices are not available.
30
*/ delay(10);
Serial.println();
wificonnect();
mqttconnect();
}
void loop()
{
temp = dht.readTemperature();
//setting a random seed (only for random values not in real life scenarios)
srand(time(0));
//initial variable activities like declaring , assigning gas = rand()%400;
int flamereading = rand()%1024;
flame = map(flamereading,0,1024,0,1024);
int flow = ((rand()%100)>50?1:0); //find the accident status 'cause fake alert may be caused by
some mischief
activities
if(temp < 45 ){ if(flame > 650 ){
```

```
accidentstatus = "Need Auditing";

if(canfanoperate)

isfanon = true; else

isfanon = false;

issprinkon = false;

}

else if(flame <= 10){

31

accidentstatus = "nothing happened";

isfanon = false;

issprinkon = false;

}

}else if(temp >= 45 && temp <= 55 )

{

if(flame <=650 && flame >100 )

{

if(cansprinkoperate)

issprinkon = true; else

issprinkon = false;

accidentstatus = "moderate";

if(gas > 160 && canfanoperate )

{

isfanon = true;

}

else{

isfanon = false;

}
```

```
}
else if(flame <= 100 && flame > 10)
{
if(cansprinkoperate)
32
issprinkon = true; else
issprinkon = false;
isfanon = false;
accidentstatus = "moderate";
}
}
else if(temp > 55)
{
if(flame > 650)
{
gas = 500 + rand()%500;
accidentstatus = "severe";
if(cansprinkoperate)
issprinkon = true; else
issprinkon = false;
if(canfanoperate)
isfanon = true; else
isfanon = false;
}
else if(flame < 650 && flame > 400 )
{
gas = 300 + rand()%500;
```

```
accidentstatus = "severe";
if(cansprinkoperate)
issprinkon = true; else
issprinkon = false;
if(canfanoperate)
isfanon = true;
else
isfanon = false;
33
}
}
else {
accidentstatus = "Need moderate Auditing";
isfanon = false;
issprinkon = false;
}
if(issprinkon){ if(flow)
{
sprinkstatus = "working";
}
else{
sprinkstatus = "not working";
}
}
else if(!issprinkon)
{
sprinkstatus = "ready";
```

```
}
else {
sprinkstatus = "something's wrong";
34
}
PublishData(temp,gas,flame,flow,isfanon,issprinkon);
//a cooldown period is set as the values and situations are random in real life sceanarios the time
can be
reduced or neclected
if(accidentstatus=="severe" && cooldown >= 600)
{
cooldown = 0;
sendalert();
PublishData(temp,gas,flame,flow,isfanon,issprinkon);
cansentalert = false;
}
if(cooldown > 999999)
{
cooldown = 601;
}
delay(1000);
++cooldown;
if (!client.loop())
{
mqttconnect();
}
}
```

```
/*.....................................retrieving to
Cloud. */
void PublishData(float temp, int gas ,int flame ,int flow,bool isfanon,bool issprinkon) {
mqttconnect(); //function call for connecting to ibm
/*
creating the String in in form JSon to update the data to ibm cloud
*/
String payload = "{\"temp\":";
payload += temp;
payload += "," "\"gas\":";
payload += gas;
payload += "," "\"flame\":";
payload += flame;
payload += "," "\"flow\":";
payload += ((flow)?"true":"false");
payload += "," "\"isfanon\":";
payload += ((isfanon)?"true":"false");
payload += "," "\"issprinkon\":";
payload += ((issprinkon)?"true":"false");
payload += "," "\"cansentalert\":";
payload += ((cansentalert)?"true":"false");
payload += "," "\"accidentstatus\":";
payload += "\""+accidentstatus+"\"";
payload += "," "\"sprinkstatus\":";
payload += "\""+sprinkstatus+"\"";
payload += "}";
```

```
if (client.publish(publishTopic, (char*) payload.c_str())) {
36
Serial.println("Publish ok"); // if it sucessfully upload data on the cloud then it will print publish ok
in Serial
monitor or else it will print publish failed
} else {
Serial.println("Publish failed");
}
}
void mqttconnect() {
if (!client.connected())
{
Serial.print("Reconnecting client to ");
Serial.println(server);
while (!!!client.connect(clientId, authMethod, token))
{
Serial.print(".");
delay(500);
}
initManagedDevice();
Serial.println();
}
}
void wificonnect() //function defination for wificonnect
{
37
Serial.println();
Serial.print("Connecting to ");
WiFi.begin("Wokwi-GUEST", "", 6);
while (WiFi.status() != WL_CONNECTED)
{
delay(100);
Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
}
void initManagedDevice() {
if (client.subscribe(subscribetopic))
{
Serial.println((subscribetopic));
Serial.println("subscribe to cmd OK");
} else {
Serial.println("subscribe to cmd FAILED");
}
}
//handles commands from user side
38
void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
{
Serial.print("callback invoked for topic: ");
Serial.println(subscribetopic);
```

```
for (int i = 0; i < payloadLength; i++)
{
data3 += (char)payload[i];
}
Serial.println("data: "+ data3);
const char *s =(char*) data3.c_str();
double pincode = 0;
if(mjson_get_number(s, strlen(s), "$.pin", &pincode))
{
if(((int)pincode)==137153)
{
const char *buf; int len;
if (mjson_find(s, strlen(s), "$.command", &buf, &len)) // And print it
{
String command(buf,len);
if(command=="\"cantfan\""){
//this works when there is gas sensor reads high value and if there should be a
//manual trigger else it will be automate canfanoperate = !canfanoperate;
}
else if(command=="\"cantsprink\"")
{
cansprinkoperate = !cansprinkoperate;
}else if(command=="\"sentalert\""){
```

```
//this works when there is accident status is severe and if there should be a
//manual trigger else it will be automate resetcooldown();
}
```

```
}
}
}
data3="";
}
void resetcooldown()
{
cooldown = 0;
}
//sent alert request to node-red void sendalert(){
cansentalert = true; cooldown = 0;
}
```

GitHub link: IBM-EPBL/IBM-Project-40534-1660630862

Project Demo Link: https://vimeo.com/772758470

wowki: FIre alaram - Wokwi Arduino and ESP32 Simulator