Project Report Format

1. **INTRODUCTION**
   - 1.1 Project Overview
   - 1.2 Purpose
2. **LITERATURE SURVEY**
   - 2.1 Existing problem
   - 2.2 References
   - 2.3 Problem Statement Definition
3. **IDEATION & PROPOSED SOLUTION**
   - 3.1 Empathy Map Canvas
   - 3.2 Ideation & Brainstorming
   - 3.3 Proposed Solution
   - 3.4 Problem Solution fit
4. **REQUIREMENT ANALYSIS**
   - 4.1 Functional requirement
   - 4.2 Non-Functional requirements
5. **PROJECT DESIGN**
   - 5.1 Data Flow Diagrams
   - 5.2 Solution & Technical Architecture
   - 5.3 User Stories
6. **PROJECT PLANNING & SCHEDULING**
   - 6.1 Sprint Planning & Estimation
   - 6.2 Sprint Delivery Schedule
   - 6.3 Reports from JIRA
7. **CODING & SOLUTIONING (Explain the features added in the project along with code)**
   - 7.1 Feature 1
   - 7.2 Feature 2
   - 7.3 Database Schema (if Applicable)
8. **TESTING**
   - 8.1 Test Cases
   - 8.2 User Acceptance Testing
9. **RESULTS**
   - 9.1 Performance Metrics
10. **ADVANTAGES & DISADVANTAGES**
11. **CONCLUSION**
12. **FUTURE SCOPE**
13. **APPENDIX**
    - Source Code
    - GitHub & Project Demo Link

PROJECT NAME : SPECIFIC INTELLIGENCE SMART FIRE MANAGEMENT SYSTEM

TEAM ID : PNT2022TMID43912

TEAM MEMBERS : SANTHOSH S

RAMAKRISHNA PRASATH S

ARUNKKUMAR K

KIRUBAKARAN D

INDUSTRY MENTOR : SANTOSHI
FACULTY MENTOR : INDHU D

ABSTRACT :

Fire is a very dangerous situation and it is very much necessary to monitor and give warning before anything untoward happens. In many developing countries, houses do not come fitted with fire alarm system as seen in developed countries like Singapore, USA etc. This results in fire being unattended and leading to lot of losses like property, human.This is the IOT (internet of things) based intelligent fire monitoring and controlling system which not only gives the real time information about the situation on the monitor but also takes the corrective action as per the need. In this system the sensors transfer data wirelessly with thehelp of MQTT (message queuing telemetry transport) networking protocol which is designed for constrained with low-bandwidth. MQTT allows us to send commands to control output, read and publish data from sensors nodes and much more. The first concept is the publish and subscribe system. In a publish and subscribe system, a device can publish a message on a topic, or it can be subscribed to a particular topic to receive message. Also it is perfect solution for internet of things application. Due to this all data can be stored in server and this data can be access by the Application program interface which we can display on the monitor and with the help of software the operator can visualize the condition at the time of fire accident.

Keywords—IoT alarm, Node Red, DHT22,Analog Temperature

1.INTRODUCTION

Since the fire causes serious damages, fire detection has been an important study to protect human life and surroundings. If fire happens in these buildings then there will be a bad social impact, major property damage and heavy casualties will be easily caused. So fire should get detected early for can hardly detect fire characteristic parameters like temperature, vapour and flame in the early time. First step to prevent the serious damages from fire is to detect the event properly. Various methods are there to detect fire which uses different properties of fire.

An- other difficulty is to properly identify the characteristics of fire. The fires properties like color, shape, temporal energy, spatial characteristics are identified in different methods. The traditional methods use sensors to detect the fire which cannot be used for early detection. So video based on computer vision based methods are more appropriate for analysis. The fire properties can be identified effectively while analyzing videos. One of the most common problems found in the area of video technology for fire detection is that early identification of fire and its properties. The main causes of fire may be burning things, wildfire, and accidents so on. To detect these fires an efficient methodology is needed as early fire detection system. Fire detection is a key point in security systems. As the system performs, it must detect the fire as early as possible. Early detection of fire in a large area is a difficult task. To identify the properties of fire is also one of the important steps. The fire and fire coloured objects are to be distinguished properly. Thus, the identification may face some difficulties like; it is sensitive to the changes in brightness, presence of shadows or to different tonalities of the red. Another important fact is that smoke identification can be included as early warning system. Smoke is an indication of fire and the proper smoke identification can prevent fire. To differentiate smoke from smoke coloured objects (like fog, cloud, etc.,) is a difficult task. In this paper different method for fire identification is studied. Initially the candidate region is to be identified to reduce the computation. From the analysis candidate region identification canbe done based on background subtraction. The background subtraction gives a better result for identifying the moving object in the scene. Then the next step is to identify the fire region based on the candidate image block. From different methods studied color analysis gives almost true result for identification of fire region.

1.1 Project overview:

In this project we will be discussing about Industry specific intelligent fire management system. Industry specific intelligent fire management system is a system which is specifically designed for the fire safety in industries. This system uses various sensors and detectors to detect the fire and then it takes appropriate action to extinguish the fire. This system is very effective in extinguishing the fire and it also minimizes the damage caused by the fire.

1.2 Purpose:

The purpose of the Industry specific intelligent fire management system is to provide a comprehensive fire management solution for industries. The system will be used to manage fire safety in industries, including fire prevention, fire detection, and fire suppression. The system will also provide industry specific fire safety information and resources.

2.LITERATURE SURVEY

A literature survey was conducted to study the various industry specific intelligent fire management systems. The survey focused on identifying the various features of these systems and the advantages and disadvantages of each system.

(1) Ahmed Imteaj et.al. Studied the problems faced by factory workers in times when fire breaks out. They proposed a system using Raspberry Pi 3 which is capable of detecting fire and providing information about area of fire. The Raspberry Pi controls multiple Arduino boards which are connected with several motors and cameras to capture the fire incident.In this, they discussed about the modern technology that can be used to reduce extremely unfortunate accidents caused by fire. We designed the whole system and calculated its effectiveness.

(2) Ondrej Krejcar proposed a model for location enhancement and personnel tracking using Wi-Fi networks. In this, he has represented the control system concept that is used in handling information of location and control unit operations. The location of the user present in the building, is obtained through Wi-Fi access points [3]. We have studied this to understand the usability of the Wi-Fi networks in live tracking and then have utilized this functionality to track fire and give information about location of fire to various devices intimating people about the mishap.

(3) Authors in have studied the safety features in home and industrial areas. They have designed new model using WSN. Not only have they incorporated temperature and humidity sensors but also included fire and smoke sensors while developing the model. They present a preceding study of WSN is able to detect fire alarm. It is for setting up a wireless sensor network with three sensors. An application was developed for getting home information

(4) Azka Ihsan Nurrahman, Kusprasapta Mutijarsa 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.

(5) Building Fire Emergency Detection and Response. Using Wireless Sensor Networks Yuanyuan Zeng, Seán Óg Murphy, Lanny Sitanayah, Tatiana Maria

Tabirca, Thuy Truong, Ken Brown, Cormac J. Sreenan Department of Computer Science, University College Cork : Wireless sensor networks (WSNs) provide a low cost solution with respect maintenance and installation and in particular, building refurbishment and retrofitting are easily accomplished via wireless technologies. Fire emergency detection and response for building environments is a novel application area for the deployment of wireless sensor networks. In such a critical environment, timely data acquisition, detection and response are needed for successful building automation. This paper presents an overview of our recent research activity in this area. Firstly we explain research on communication protocols that are suitable for this problem. Then we describe work on the use of WSNs to improve fire evacuation and navigation

(6) Avoidance of Fire Accident on Running Train Using ZigBee Wireless Sensor Network R. Pitchai Ramasamy1 ,M. Praveen Kumar1 , S. Sarath Kumar2 and R. Raghu Raman3:-The main objective of our proposed system is to safe guard people's life and government property. This paper will focus on the system that will detect and control the fire accidents on running train. In-house parameters such as temperature and humidity in the each coach can be monitored in real time. From the information collected by the sensor system, decisions for firefighting, alarming, and automatic water sprinkler system can be made more quickly by the relevant system or engine driver. After receiving the signal, the engine driver will stop the train and take necessary action. Key Terms: Fire alarm system, Fire protection systems, Wireless sensor network, Automatic sprinklers, Signal transmission. The trains are moderate vehicles used for transporting people and goods. Mostly, the people prefer the train journey for longer distance as it is cheaper. Since induction of train for public transportation, the fire accidents are not catered seriously by the Indian Railways. The notices showing "Do not smoke", "Do not carry inflammable material" are the only precautionary warnings about the fire in each compartment. However, because of failure in routine maintenance system or by the activities of illegal social elements, the fire accidents in train occur frequently.

2.1 Existing Problem:

The traditional fire management system is very costly and it is not affordable to many industries. Also, the system is not intelligent enough to take decisions on its own. The industry specific intelligent fire management system is a cost effective and intelligent system that can take decisions on its own.

(i) No such system exists.

(ii)There is no industry specific system, however a standard intelligent fire management system exists which can be used in any industry.

After doing the literature survey we have listed some of the features that are existing in the now used fire alarm systems. In the next section of the paper we will propose a new model taking into account what features are being presently used and how they need to improve. The feaatures of the existing system are as uder.Identify status periodically - The system checks for a fire at particular intervals and not continuously or not in real time. This is a drawback as there will possibly be a time lag between the actual fire incident and when the fire will be reported due to periodic identification. Manual operation for transfeering information-Automatic operation is not facilitated in the present systems. Not able to find the pressure point of the building which are likely to catch fire easily. Difficult to sense tructural damage. MEMS are used to get axis of the building block.

2.2 References

Ahmed, A.-K., Kazmi, S. I. A., & Pandey, J. (2018). IoT Based Smart Network for Blood Bank. Paper
presented at the 2018 7th International Conference on Reliability, Infocom Technologies and
Optimization (Trends and Future Directions)(ICRITO).

Al Mamari, A. R. M. H., Al Mamari, H., Kazmi, S. I. A., Pandey, J., & Al Hinai, S. (2019). IoT based
Smart Parking and Traffic Management System for Middle East College. Paper presented at the
2019 4th MEC International Conference on Big Data and Smart City (ICBDSC).

De Luca, G. E., Carnuccio, E. A., Garcia, G. G., & Barillaro, S. (2016). IoT fall detection system for
the elderly using Intel Galileo development boards generation I. Paper presented at the IEEE
CACIDI 2016-IEEE Conference on Computer Sciences.

Ibrahim, O., Hassan, S. M., Abdulkarim, A., Akorede, M. F., & Amuda, S. A. (2019). Design of
Wheatstone Bridge Based Thermistor Signal Conditioning Circuit for Temperature Measurement.
Journal of Engineering Science & Technology Review, 12(1).

Kang, D.-H., Park, M.-S., Kim, H.-S., Kim, D.-y., Kim, S.-H., Son, H.-J., & Lee, S.-G. (2017). Room
temperature control and fire alarm/suppression IoT service using MQTT on AWS. Paper presented
at the 2017 International Conference on Platform Technology and Service (PlatCon).

Lee, I., & Lee, K. (2015). The Internet of Things (IoT):
Applications, investments, and challenges for 8 / 9
Business Horizons, 58(4), 431-440.

Minoli, D. (2013). Building the internet of things with IPv6 and MIPv6: the evolving world of M2m communications: John Wiley & Sons.
Nair, K. K., Abu-Mahfouz, A. M., & Lefophane, S. (2019). Analysis of the Narrow Band Internet of
Things (NB-IoT) Technology. Paper presented at the 2019 Conference on Information
Communications Technology and Society (ICTAS).
Pandey, J., Kazmi, S. I. A., Hayat, M. S., & Ahmed, I. (2017). A study on implementation of smart
library systems using IoT. Paper presented at the 2017 International Conference on Infocom
Technologies and Unmanned Systems (Trends and Future Directions)(ICTUS).
Sengupta, B., Sawant, S., Dhanawade, M., Bhosale, S., & Anushree, M. (2019). Water Quality
Monitoring using IoT.

2.3 Problem Statement Definitions:

Different IoT based applications are used for monitoring and control the systems parameters,
whereas the communication between module and a user is realized via wireless communication techniques such as Wi-Fi, Bluetooth, RF, and ZigBee, programs using SCADA also developed to develop user interface, SCADA are difficult to write and not adaptable to the users because of expensive and excessive libraries, the modules which are implemented using ZigBee, Rf and Bluetooth technologies are widely applicable in easy-to-use applications due to the limited range between the sender and the receiver, and useful for the low data transfer rates and these communication techniques are restricted to simple applications because of their slow communication speeds, distances and low data security because of full range of encryption techniques are not available. Nowadays, timer-controlled systems have been easily replaced with remote controlled systems after the internet became widespread. In these systems, it is known as an important issue to get information about not only the control, but also the conditions of the machines or devices through internet. In accordance with this need, there are some works about implementation of condition monitoring of system through internet and development of internet-based remote controlling or monitoring practices.
Previous implementation “Industrial monitoring system using ESP8266” explains about liquid level monitoring, DC motor speed control, color mixing and energy monitoring using PLCs [3]. “Industrial process monitoring using Raspberrypi” implemented to measure of light intensity, temperature, liquid level monitoring, system and user are connected by using Wi-Fi module [5].

“Industrial automation using Zigbee” Zigbee system is based on wireless sensor network used for regulating and supervising different processes in industries without distracting other processes [7].
paper on home atomization using IoT. IoT is fore part of this system to minimize human efforts. The main aspect of this system to monitor and control the home appliances through mobiles [6]. Previous work on “IoT based Monitoring and Control System for Home Automation” explains automation of home appliances. Using mobile phones how devices are accessed using application tool kits, and electrical appliances are operated remotely with less efforts and automatically provides efficient power consumption [4].
Above mentioned work gives knowledge about industrial parameters monitoring from remote area by inter connecting modules to cloud through Wi-Fi, Bluetooth or Zigbee module. Online supervising of industrial process helps to improve throughput, decreases process time and provides security.
Proposed paper was implemented using ESP32.

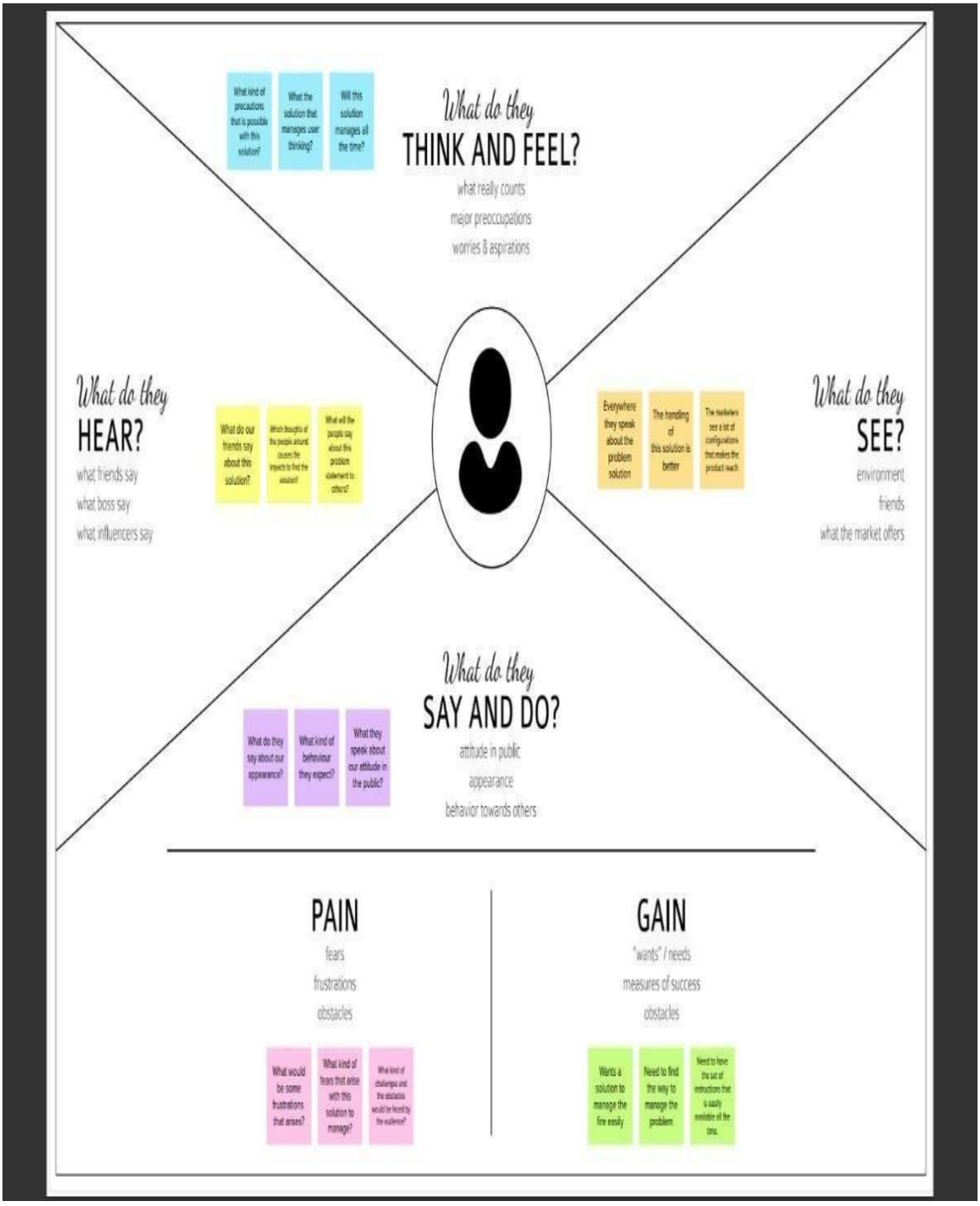
ESP32 is the advanced device suitable for IoT applications in terms of properties and price. ESP32 available in various versions and user-friendly even students can connect it on bread boards easily for laboratory applications. ESP32-Dev kit is perfect solution for implementation of different education application [1].
In the proposed system related work implemented in smarter way with module using ESP32- Dev module which consists of in-built Wi-Fi Blue tooth module no need to connect external board to provide Wi-Fi, and also consists of inbuilt temperature, touch and hall effect sensor industrial process parameters monitored efficiently from remote area using smartphone by downloading Blynk application program in a smartphone.

3.IDEATION AND PROPOSED SOLUTION

This system is implemented with ESP32-DEV module by interconnecting DC motor and distinctive sensors like temperature, smoke, flame sensor is interfaced to cloud by enabling inbuilt Wi-Fi module of ESP32, the system is connected to the cloud as well as android mobile loaded with Blynk application tool kit. The sensor parameter variation is uploaded to the cloud. Through the Cloud all sensors in the industrial applications are monitored easily and efficiently. In this application is Cayenne project builder is used as cloud and user mobile is connected using Blynk, which is android application tool kit. Blynk can be accessed through mobile or Laptop and also used as the cloud. If the sensor parameters limit exceeds than specified limit motor or load which is connected to the load automatically triggered and informed through SMS or electronic mail to the registered users such application very uses full for small scale industry.

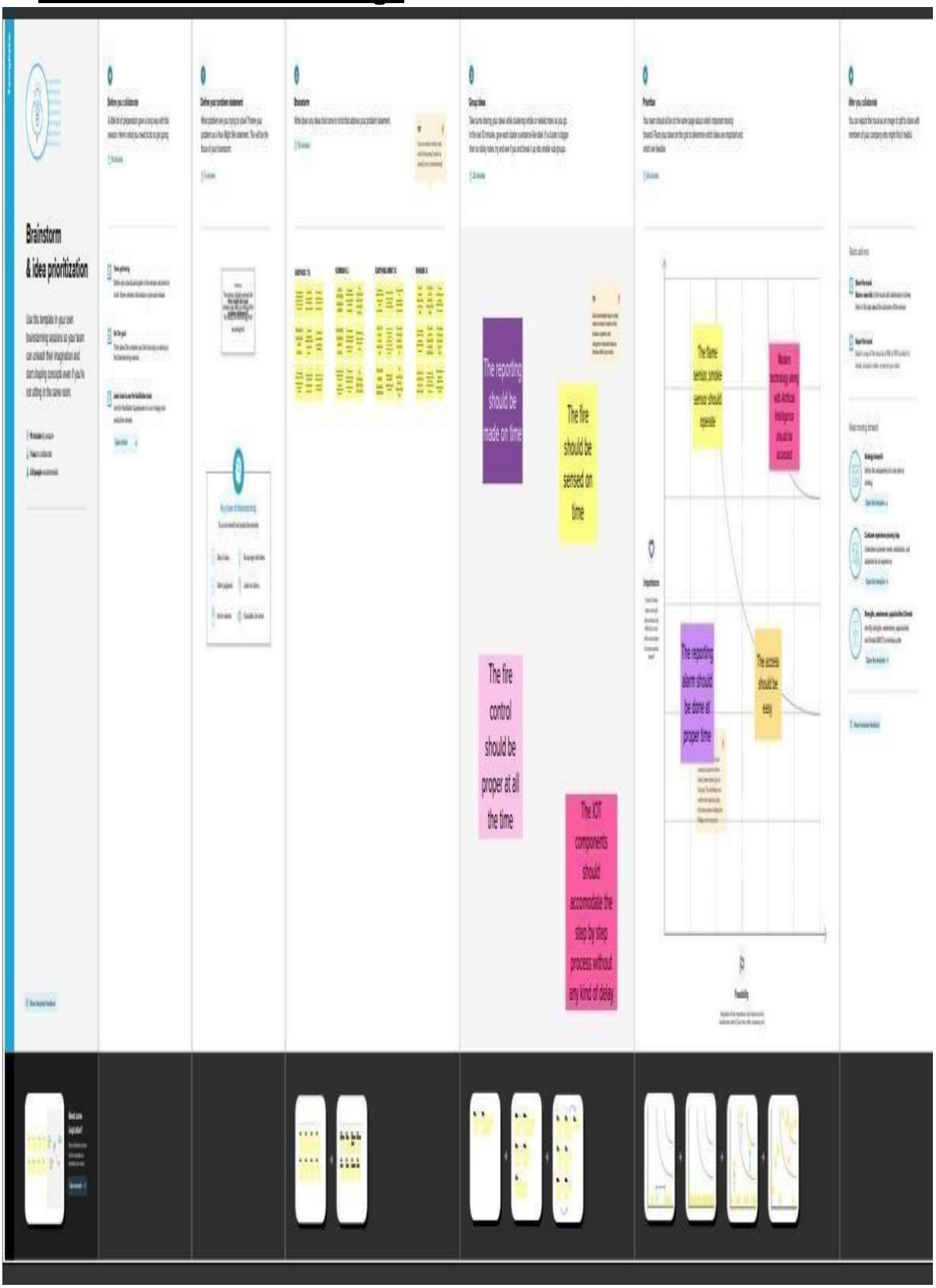
In this system, industrial processes like energy meter monitoring, DC speed control, Temperature, Humidity, Gas levels and Fire accidents if any are monitored through android mobiles, and parameters data can be updated periodically by using cloud.

3.1 Empathy map Canvas:


What kind of precautions that is possible with this solution?
What the solution that manages user thinking?
Will this solution manages all the time?
What do they
THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
What do they
HEAR?
what friends say
what boss say
what influencers say
What do our friends say about this solution?
Which thoughts of the people around causes the impacts to find the solution?
What will the people say about this problem statement to : others?
Everywhere they speak about the problem solution
The handling of this solution is better
The marketers see a lot of configurations that makes the product reach
What do they
SEE?
environment
friends
what the market offers
What do they
SAY AND DO?
attitude in public
appearance
behavior towards others
What do they say about our appearance?
What kind of behaviour they expect?
What they speak about our attitude in the public?
PAIN
fears
frustrations
obstacles
What would be some frustrations that arises?
What kind of fears that arise with this solution to manage?
What kind of challenges and the obstacles would be faced by the audience?
GAIN
"wants" / needs
meesures of success
obstacles
Wants a solution to manage the fire easily
Need to find the way to manage the problem
Need to have the set of instructions that is easily available all the time.

3.2

Ideation and Brainstroming :



Proposed Solution:

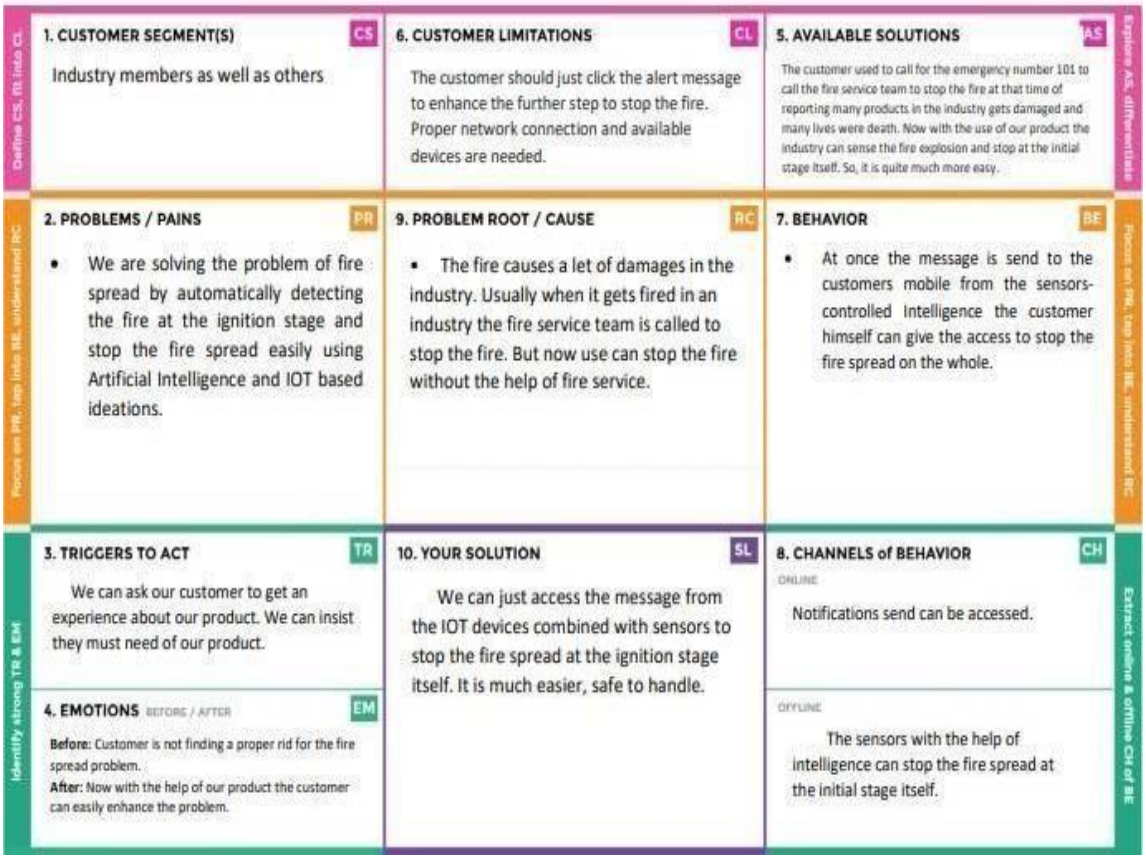
3.3

PROPOSED SOLUTION

S.No	Parameter	Description
1.	Problem Statement (Problem to be solved)	We are going to solve the fire explosion problem
2.	Idea / Solution description	Industry-specific intelligent fire management system. We are going to detect and stop the fire explosion spreading in the industry as well as in the common places.
3.	Novelty / Uniqueness	The usage of the Artificial Intelligence is the uniqueness in our solution for the fire explosion.
4.	Social Impact / Customer Satisfaction	The AI detects and senses the fire using many sensors that we use and it helps the customers to access with the immediate notification and the timely access.
5.	Business Model (Revenue Model)	This model is used to calculate the probability of the ignition and spread across a landscape.
6.	Scalability of the Solution	This is completely modular system makes it easily expandable and business efficient for the customized fire detection, with the significant cost and management.

Problem Solution Fit:

3.4


1. CUSTOMER SEGMENT(S)
CS
Industry members as well as others
6. CUSTOMER LIMITATIONS
CL
The customer should just click the alert message to enhance the further step to stop the fire. Proper network connection and available devices are needed.
5. AVAILABLE SOLUTIONS
AS
The customer used to call for the emergency number 101 to call the fire service team to stop the fire at that time of reporting many products in the industry gets damaged and many lives were death. Now with the use of our product the industry can sense the fire explosion and stop at the initial stage itself. So, it is quite much more easy.
2. PROBLEMS / PAINS
PR
We are solving the problem of fire spread by automatically detecting the fire at the ignition stage and stop the fire spread easily using Artificial Intelligence and IOT based ideations.
9. PROBLEM ROOT / CAUSE
RC
The fire causes a let of damages in the industry. Usually when it gets fired in an industry the fire service team is called to stop the fire. But now use can stop the fire without the help of fire service.
7. BEHAVIOR
BE
At once the message is send to the customers mobile from the sensors-controlled Intelligence the customer himself can give the access to stop the fire spread on the whole.
3. TRIGGERS TO ACT
TR
We can ask our customer to get an experience about our product. We can insist they must need of our product.
4. EMOTIONS BEFORE / AFTER
EM
Before: Customer is not finding a proper rid for the fire spread problem.
After: Now with the help of our product the customer can easily enhance the problem.
10. YOUR SOLUTION
SL
We can just access the message from the IOT devices combined with sensors to stop the fire spread at the ignition stage itself. It is much easier, safe to handle.
8. CHANNELS of BEHAVIOR
CH
ONLINE
Notifications send can be accessed.
OFFLINE
The sensors with the help of intelligence can stop the fire spread at the initial stage itself.

4.REQUIREMENT ANALYSIS

4.REQUIREMENT ANALYSIS

The Intelligent Fire Management System (IFMS) is a comprehensive system designed to manage fire safety in buildings and other structures. The system uses advanced technologies to detect and extinguish fires, and to provide decision-makers with real-time information about the status of fires and the building's occupants.

The system is composed of four main subsystems:
1. The Fire Detection and Suppression System (FDSS)
2. The Building Management System (BMS)
3. The Emergency Communications System (ECS)
4. The Fire Safety Management System (FSMS)

4.1 Functional Requirement:

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through website or application Registration through Social medias (like Instagram, Facebook) Registration through LinkedIN
FR-2	User Confirmation	Verification via Email Verification via OTP
FR-3	User Login	Login through website or App using the respective username and password
FR-4	User Access	Allows the app requirement
FR-5	User Guide	Guides the basic steps of using the application
FR-6	User Upload	User should be able to send the data
FR-7	User Solution	Data report should be generated and delivered to user for per every 24 hours

FR-8	User Data Sync	API interface to increase to invoice system

4.2 Non-Functional Requirements:

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Usability requirements can consider language barriers and localization tasks. Usability can be assessed from the below functions. Efficiency of use. Low perceived workload. Easy and simple UI.
NFR-2	**Security**	Access permissions for the particular system information may only be changed by the system's data administrator.
NFR-3	**Reliability**	The database update process must roll back all related updates when any update fails.
NFR-4	**Performance**	The front-page load time must be no more than 2 seconds for users that access the website using an VoLTE mobile connection.
NFR-5	**Availability**	New module deployment mustn't impact front page, product pages, and check out pages availability and mustn't take longer than one hour. The rest of the pages that may experience problems must display a notification with a timer showing when the system is going to be up again.

NFR-6	**Scalability**	We can increase scalability by adding memory, servers, or disk space. On the other hand, we can compress data, use optimizing algorithms. The website attendance limit must be scalable enough to support 500,000 users at a time.

5.PROJECT DESIGN

Wokwi Software requirement for prototype module implementation:

i. ESP32 ii. DC motor iii. DHT22 sensor iv. Smoke sensor v. Flame sensor vi. Mobile vii. MQTT. viii. project builder.

ESP32: ESP32 is a series of low-power, low-cost system on chip. The microcontrollers with pack of Wi-Fi and dual-mode Bluetooth. The ESP32 series employs a microprocessor of type Tensilica Xtensa LX6 available in both single core and dual core variations and includes in-built sensors, antenna switches, RF balun, low noise, power amplifier receiver amplifier, filters and power-management modules. ESP32 is created and developed by Espressif systems, a Shanghai-based Chinese company, and is manufactured by TSMC using their 40nm process. It is a successor to the ESP32 microcontroller.

Flame sensor:

Flame sensor is sensitive to light intensity, used to identify the flame based on intensity, in industrial applications to detect fire accidents generally flame sensor embedded with alarm purposes. This module can detect flame in 760nm to 1100nm range of light source. The detection distance is up to 100cm. The detection angle is 60 degrees. Sensor gives output as digital or analog signal.

Smoke Sensor:

The smoke sensor is based on the free scale semiconductor MC145012DW smoke detector chip. The IC consists of infrared photoelectric chamber. It detects by sensing scattered light from minute smoke particle or it will take approximately 20seconds for relay to fire.

Temperature sensor:

Temperature sensors tend to measure heat or temperature in the industrial process, and often used in hazardous environment.

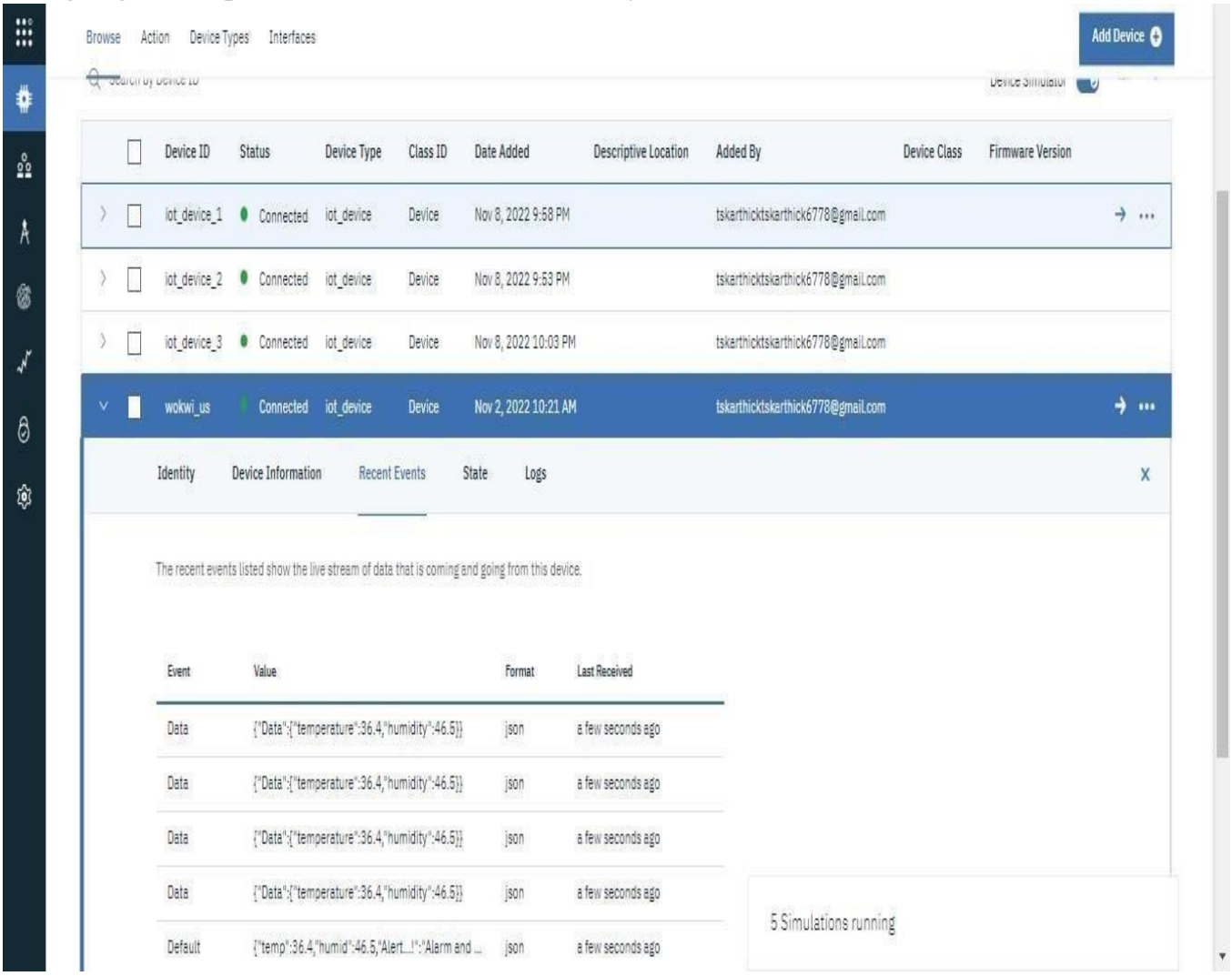
Humidity sensor:

Relative humidity in the environment is measured effectively, they measure both moisture and temperature in the air and gives relative humidity as percentage. Most of the humidity sensors are capacitive measurement sensors, moisture from the air collects on the film and causes changes in the voltage levels between the two plates, change in the voltage then converted in to digital measurement.

IBM Watson Cloud :

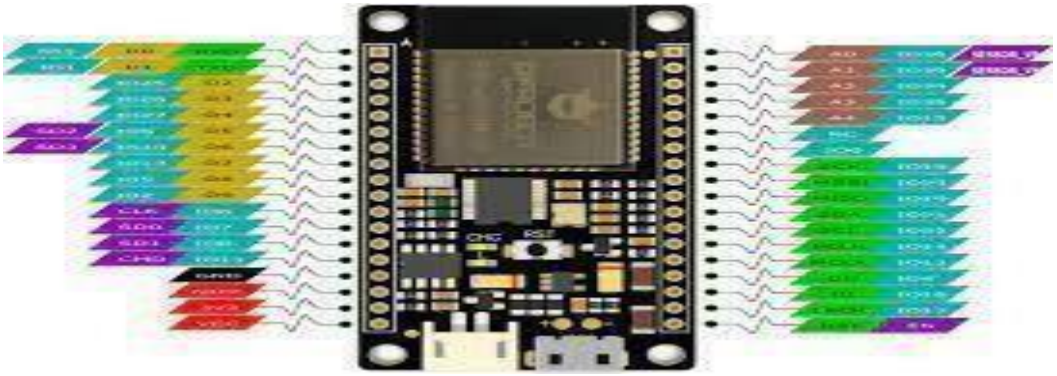
IBM Watson® Studio empowers data scientists, developers and analysts to build, run and manage AI models, and optimize decisions anywhere on IBM Cloud Pak® for Data. Unite teams, automate AI lifecycles and speed time to value on an open multicloud architecture.

Bring together open source frameworks like PyTorch, TensorFlow and scikit-



learn with IBM and its ecosystem tools for code-based and visual data science. Work with Jupyter notebooks, JupyterLab and CLIs — or in languages such as Python, R and Scala.

Programming ESP32:



The ESP32 is a series of Soc's (System on Chip), developed by Espress if Systems. They contain a Tensilica Xtensa LX6 microprocessor, WiFi, and integrated Bluetooth.

It was developed to have low consumption and low cost, and it is a very interesting option for makers or developers of products.

It can be found in some variations, from the chip going through modules and development boards.

To begin, the use of development boards tends to be the best option, since it involves the minimal amount of electronics for maintenance of the ESP32, which (although very small) is already available.

There is also the ease of using USB-Serial converters on these development boards that make the communication process with ESP32 much easier.

Development boards can also be found on a wide range of models, some very similar and others with very specific facilities.

Arduino IDE:

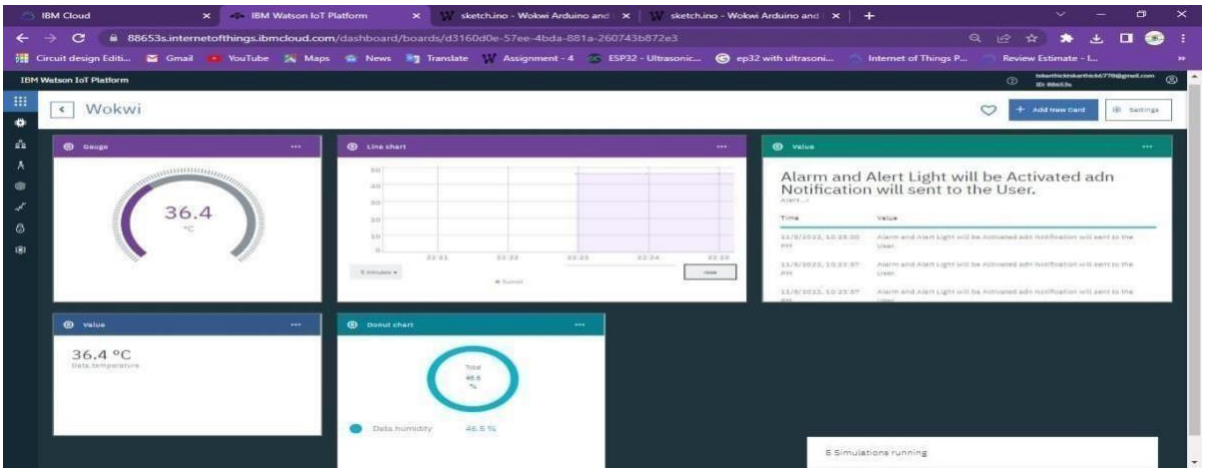
Arduino IDE is integrated development application written in a python programming language it is a cross platform. IDE supports C and C++ languages also. It is a best platform to develop IoT applications.

In this present module TUNIOT is used to generate the code, Tuniot is a visual programming tool used for the ESP32 and ESP8266. ESP32 is a new board library are available under Arduino IDE. Using tuniot it is very easy to write codeby using visual drag and dropping different widgets available on the flat form overall set of the project with different sensors connected ESP32 sensed values are updated in the cloud.

Node_red Project Builder:

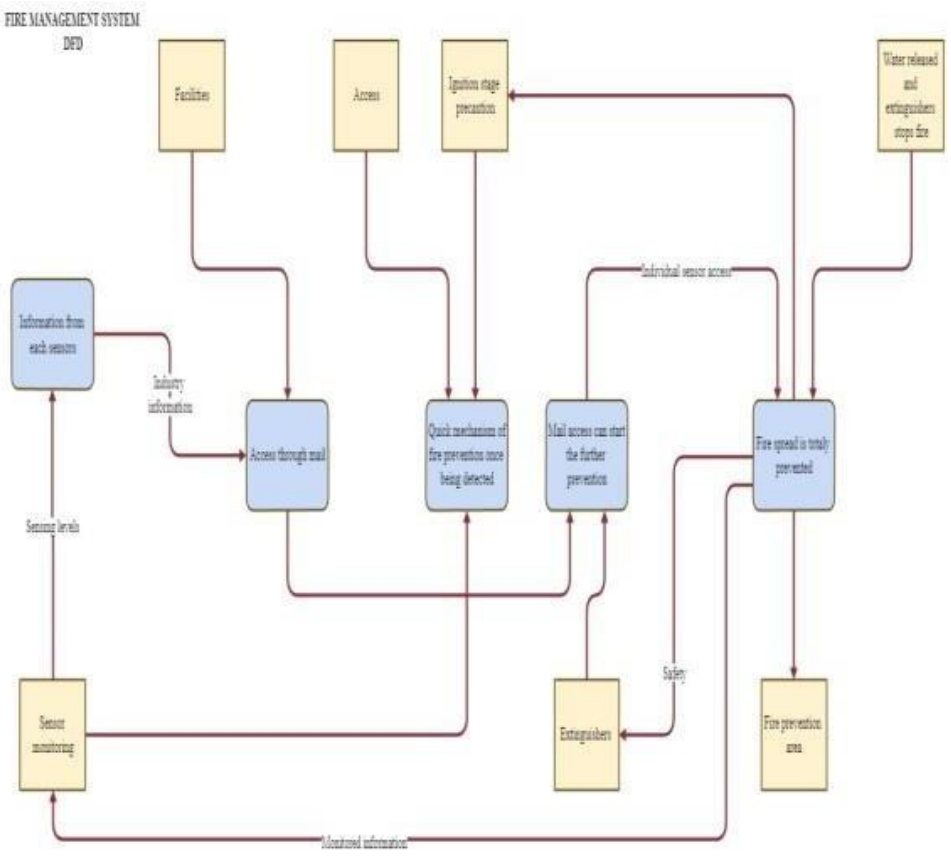
Node red is the world's first drag and drop IoT project builder, using cayenne it is easy to create custom dashboard with drag and drop widgets, cayenne also used as free cloud, Monitoring can be done both from mobiles and Pc, Notifications can be sent, Various widgets are available.

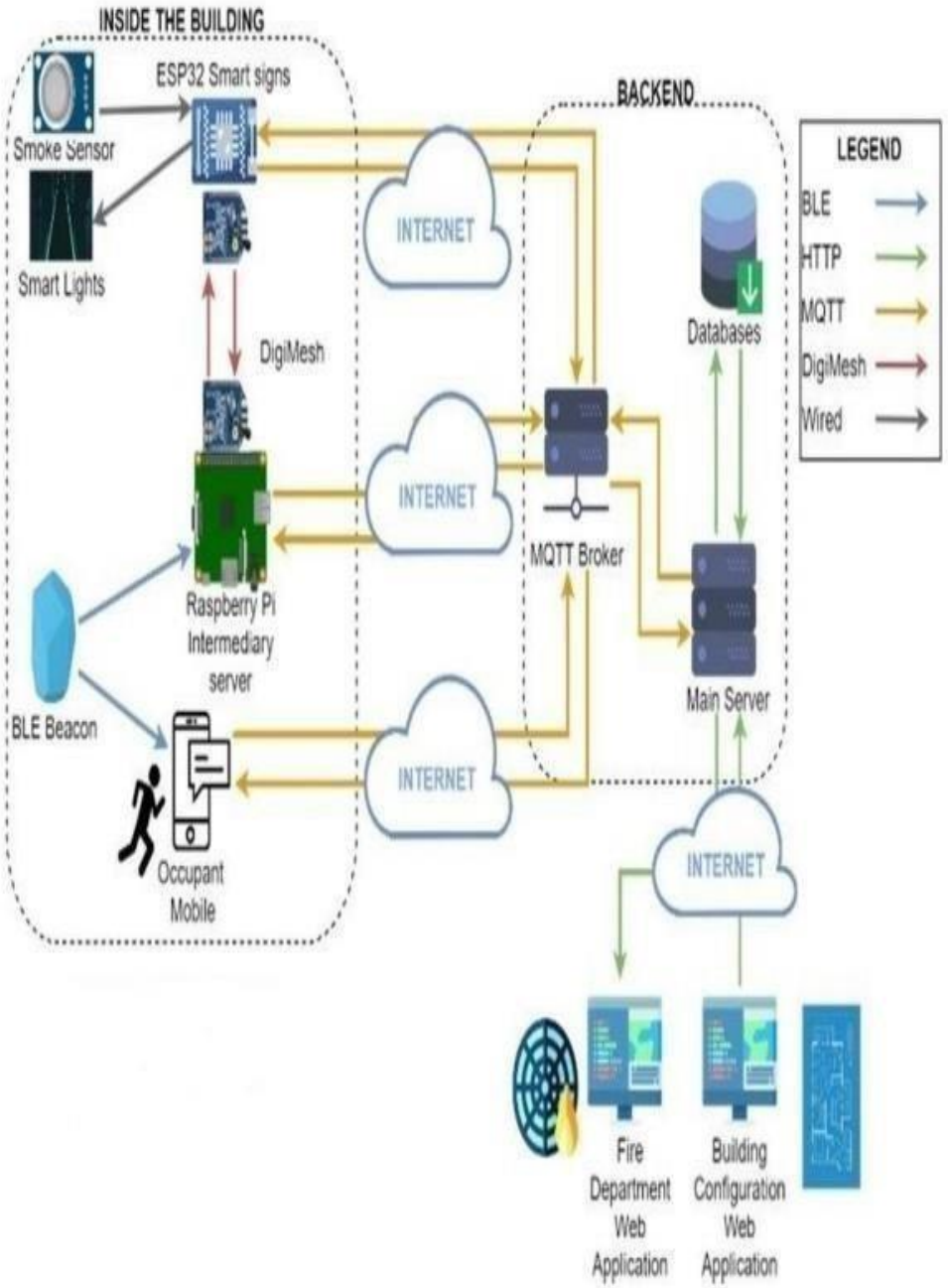
Module Working:



All the components are placed and assembled accordingly. By using the cayenne cloud application, the hardware components are interfaced with cloud. By using proposed system, the changes in the industrial parameters are monitored and controlled using mobile application as well as PC application. Using the triggers in the cloud we primarily set some threshold values to every sensor, when the sensors detect the values greater than the threshold values, alarming devices will alert us by sending email or SMS to mobile devices. Thereby we turn on the Relay switch by mobile or PC application to turn on the cooling fans. The main advantage of this concept is that we can monitor the changes in the industry from anywhere and can operate them easily without requirement of human beings. Proposed module connected to the cloud through in-built Wi-Fi module of ESP32 distinct parameters of the sensors are computed to the cloud, updated values are monitored through smartphone using Blynk application toolkit (BLYNK APP), processed values exceeds the specified limit automatic SMS or electronic mail send to the registered users by the cloud. The updated values are monitored through smartphone using application toolkit (MIT APP INVENTOR) and node red project builder. Proposed module implemented with DC motor and distinct sensors by connecting to the ESP32 Dev at different conditions and displayed and stored in the cloud, monitored and controlled by the industrialist even from remote areas. For example, if industry caught with any fire accident based on the updating values of the sensors (smoke, temperature values exceeds specified limit), machinery or production automatically stopped from the remote area by the responsible person of the industry. These results in the correctness working of the system at every instant of time the values are automatically updated in the cloud.

5.1 Data Flow Diagram:

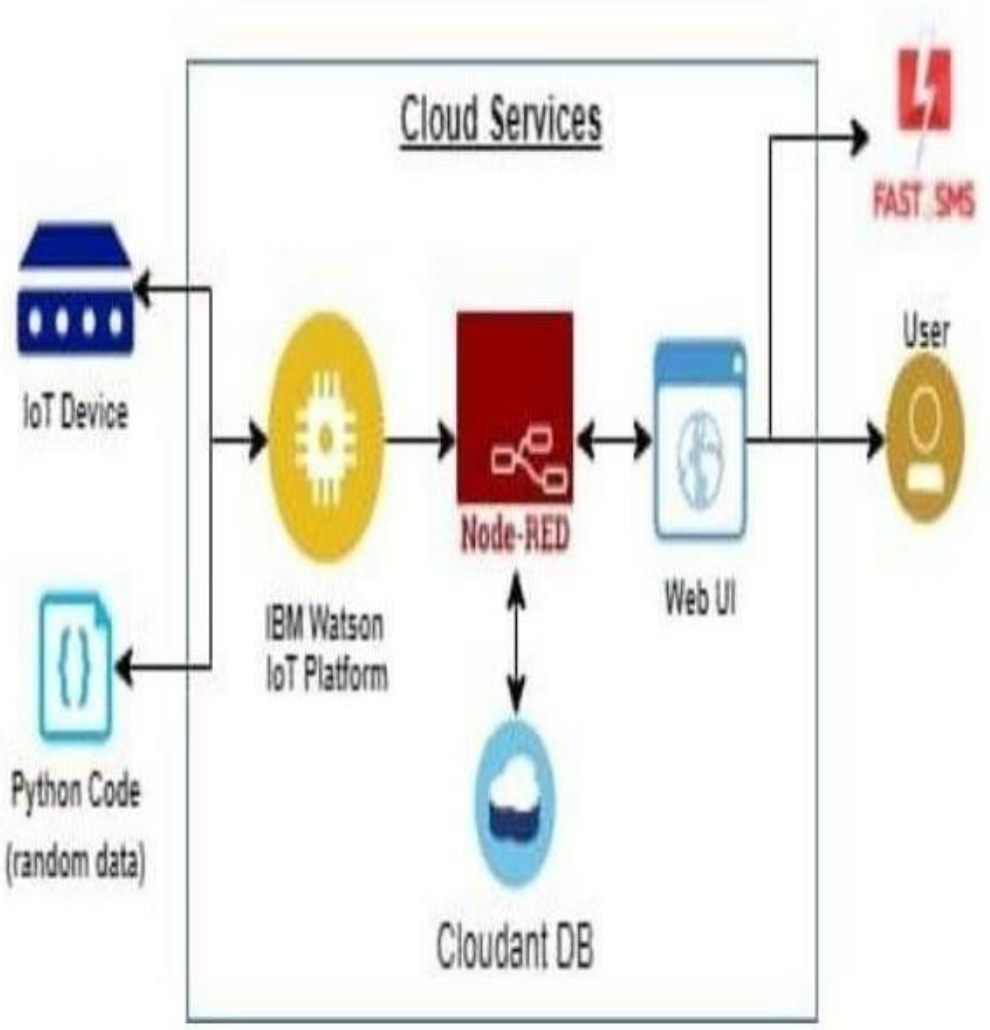

FIRE MANAGEMENT SYSTEM
DFD
Facilities
Access
Ignition stage precautions
Water released and extinguishers stops fire
Individual sensor access
Information from each sensors
Industry information
Access through mail
Quick mechanism off fire prevention once being detected
Mail access can start the further prevention
Fire spread is totaly prevented
Sensing levels
Safety
Sensor monitoring
Extinguishers
Fire prevention area
Monitored information



5.2 Solution and Technical Architecture:

Technical Architecture:

Project: Industry Specific Fire Intelligent Management System



5.3 User Stories:

User Stories

User Type	Functional requirement	User story number	User story/task	Acceptance criteria	Priority	Release
Customer (Mobile user, Web user, Care executive, Administrator)	Registration	USN-1	As a user, I can register for the application by entering my mail, password, and confirming my password	I can access my account/ dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
	Dashboard	USN-3	As a user, I can register for the application through internet	I can register & access the dashboard with Internet login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can confirm the registration in Gmail	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can login with my id and password	High	Sprint-1

6.PROJECT PLANNING AND SCHEDULING

1. Define the project scope, including the goals and objectives of the project. The goal of this project is to develop an IoT-based intelligent fire management system that can detect, monitor, and extinguish fires in real-time. The system will

be composed of a network of sensors and actuators that will be integrated with a central control system.

2. Develop a project schedule that outlines the major milestones and deliverables of the project. The project schedule should include the following milestones:

1. Research and development of the fire detection and extinguishing system.

2. Installation of the fire detection and extinguishing system in the target environment.

3. Testing and evaluation of the system.

The project schedule should also include the following deliverables:

1. A working prototype of the fire detection and extinguishing system.

2. A report detailing the testing and evaluation of the system.

3. Identify the resources required for the project, including manpower, materials, and funding.

The resources required for this project include:

1. Manpower: A team of engineers and technicians for the research and development phase, as well as for the installation and testing phase.

2. Materials: The materials required for the prototype system, as well as for the installation in the target environment.

3. Funding: The funding required for the research and development phase, as well as for the installation and testing phase.

6.1 Sprint Planning and Estimation:

IoT-based intelligent fire management system sprint planning and estimation

The development of an IoT-based intelligent fire management system will require a significant amount of effort and resources. In order to ensure the successful completion of the project, it is essential to plan and estimate the required work accurately.

The first step in planning the development of the system is to identify the specific goals and objectives that the system is intended to achieve. Once these have been

identified, the next step is to determine the specific requirements that the system must meet in order to achieve these goals.

Once the goals and requirements have been identified, the next step is to estimate the amount of work that will be required to develop the system. This estimation should take into account the complexity of the system and the experience of the development team.

Once the estimation has been completed, the next step is to develop a detailed plan for the development of the system. This plan should identify the specific tasks that need to be completed, the resources that will be required, and the timeline for the development.

6.2 Sprint Delivery Schedule:

Development Team

The development of an IoT-based intelligent fire management system will require the involvement of a number of different disciplines. In order to ensure the successful completion of the project, it is essential to assemble a team of experts that have the necessary skills and experience.

The team should include individuals with experience in the following areas:

IoT

Wireless networking

Embedded systems

Software development

Database development

Budget

The development of an IoT-based intelligent fire management system will require a significant amount of financial resources. In order to ensure the successful completion of the project, it is essential to have a clear understanding of the costs associated with the development.

The budget for the project should take into account the following factors:

The costs of the hardware that will be required

The costs of the software that will be required

The costs of the development team

The costs of the testing and deployment of the system

Schedule

The development of an IoT-based intelligent fire management system will require a significant amount of time. In order to ensure the successful completion of the project, it is essential to have a clear understanding of the timeline for the development.

The schedule for the project should take into account the following factors:

The time required for the development of the system

The time required for the testing of the system

The time required for the deployment of the system

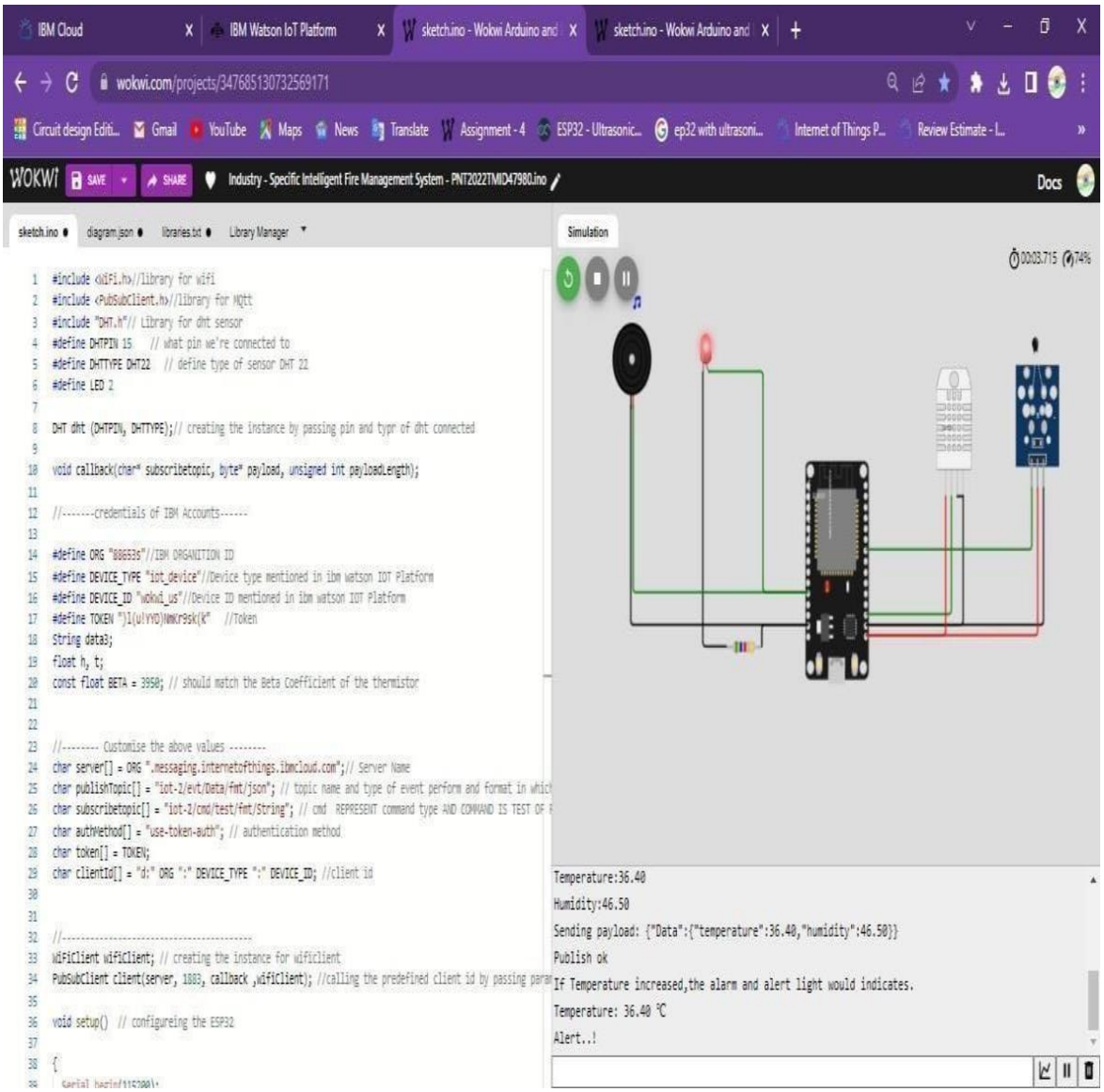
6.3 Reports from JIRA:

The IoT based intelligent fire management system aims to provide a comprehensive solution for fire safety in commercial and industrial buildings. It uses a variety of sensors to detect fire hazards and automatically triggers the fire suppression system. It also has a user interface that allows the user to monitor the status of the system and make changes to the settings.

1. The system should be able to automatically detect and extinguish fires.
2. The system should be able to provide real-time alerts to the authorities in case of a fire.

3. The system should be able to monitor the environment and provide data that can be used to improve fire safety.
4. The system should be able to provide accurate and up-to-date information about the location of fires.
5. The system should be able to provide information about the cause of fires.

7.CODING & SOLUTIONING



7.1 Feature 1:

-Monitoring and detection of fire: The system can constantly monitor the environment for potential fire hazards and provide early warning in the event of a fire.

-Automatic fire suppression: In the event of a fire, the system can automatically deploy fire suppression systems such as sprinklers or fire extinguishers.

-Remote monitoring and control: The system can be monitored and controlled remotely, allowing for quick and effective response to fires.

-Integrated security: The system can be integrated with security systems to provide additional protection against fire hazards.

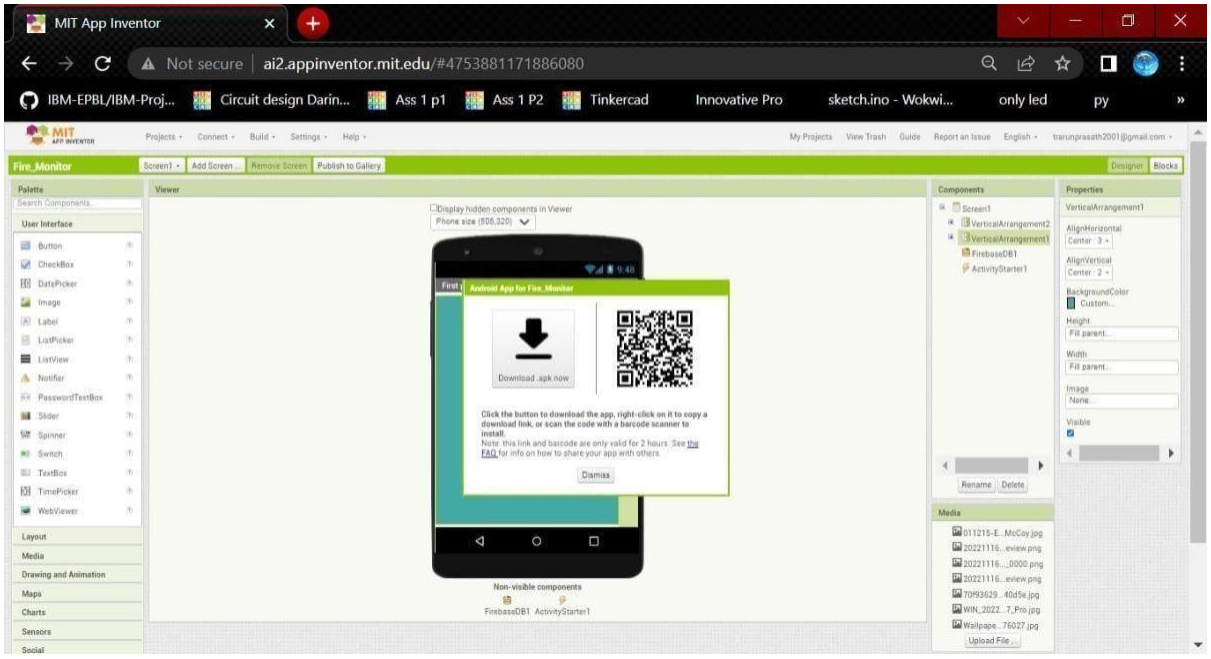
7.2 Feature 2:

-The cloud platform enables the iot based intelligent fire management system to remotely monitor and manage fire safety devices and systems in real-time. It also provides data analysis and reporting capabilities to help improve fire safety.

-The fire detection and suppression system is fully automated and cloud based. It uses advanced sensors to detect fire and notify the concerned personnel. The system is also equipped with intelligent video analytics that can identify the fire and its location. The fire management system is also equipped with a fire suppression system that can automatically extinguish the fire.

MIT App Inventor is an intuitive, visual programming environment that allows everyone even children to build fully functional apps for smartphones and tablets. Those new to MIT App Inventor can have a simple first app up and running in less than 30 minutes. And what's more, our blocks-based tool facilitates the creation of complex, high-impact apps in significantly less time than traditional programming environments. The MIT App Inventor project seeks to democratize software development by empowering all people, especially young people, to move from technology consumption to technology creation

Blocks-based coding programs inspire intellectual and creative empowerment. MIT App Inventor goes beyond this to provide real empowerment for kids to make a difference -- a way to achieve social impact of immeasurable value to their communities.


MIT App Inventor
Not secure
ai2.appinventor.mit.edu/#4753881171886080
IBM-EPBL/IBM-Proj...
Circuit design Darin...
Ass 1 p1
Ass 1 P2
Tinkercad
Innovative Pro
sketch.ino - Wokwi...
only led
py
Fire_Monitor
Screen1
Add Screen ...
Remove Screen
Publish to Gallery
Designer
Blocks
Palette
Viewer
Components
Properties
Android App for Fire_Monitor
Download .apk now
Click the button to download the app, right-click on it to copy a download link, or scan the code with a barcode scanner to install.
Dismiss
Non-visible components
FirebaseDB1
ActivityStarter1


Reset Your password
Please enter your email address and we will send you a password reset link.
Email Address
Send Reset Link

HTML (the Hypertext Markup Language) and CSS (Cascading Style Sheets) are two of the core technologies for building Web pages. HTML provides the structure of the page, CSS the (visual and aural) layout, for a variety of devices.

HTML is the language for describing the structure of Web pages. HTML gives authors the means to:

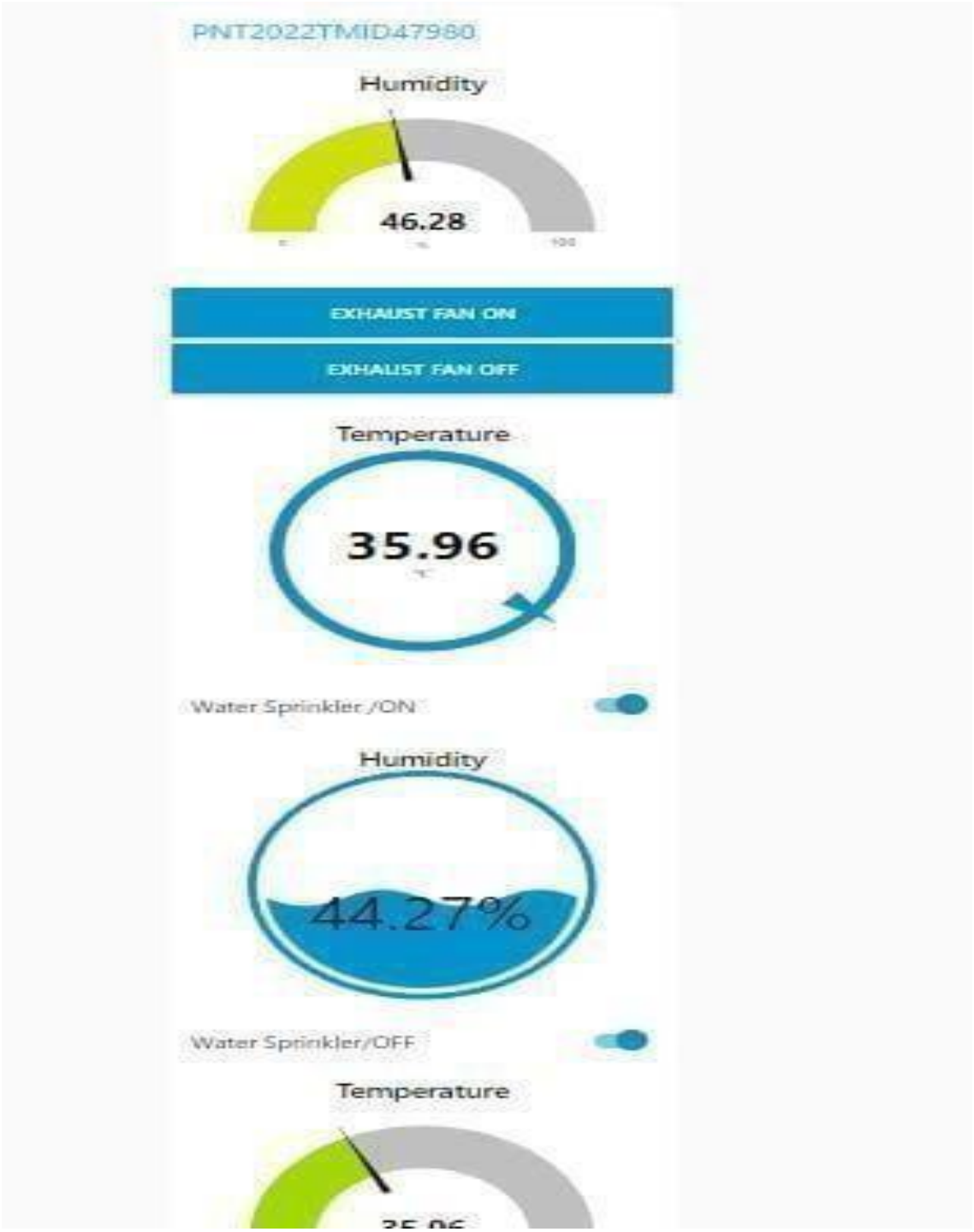
Publish online documents with headings, text, tables, lists, photos, etc.

Retrieve online information via hypertext links, at the click of a button.

Design forms for conducting transactions with remote services, for use in searching for information, making reservations, ordering products, etc.

Include spread-sheets, video clips, sound clips, and other applications directly in their documents.

CSS is the language for describing the presentation of Web pages, including colours, layout, and fonts. It allows one to adapt the presentation to different types of devices, such as large screens, small screens, or printers. CSS is independent of HTML and can be used with any XML-based markup language. The separation of HTML from CSS makes it easier to maintain sites, share style sheets across pages, and tailor pages to different environments. This is referred to as the separation of structure (or: content) from presentation.



Node-RED provides a web browser-based flow editor, which can be used to create JavaScript functions. Elements of applications can be saved or shared for re-use. The runtime is built on Node.js. The flows created in Node-RED are stored using JSON. Since version 0.14, MQTT nodes can make properly configured TLS connections.

Node-RED is a flow-based programming tool, originally developed by IBM's Emerging Technology Services team and now a part of the OpenJS Foundation.

7.3 Data Schema:

The database schema of the Iot based intelligent fire management system includes a table for storing information about fire incidents, a table for storing information about fire alarms, and a table for storing information about fire extinguishers.

8.TESTING

8.1 Test Cases:

1. The system is able to automatically detect and extinguish fires in the vicinity.
2. The system is able to automatically detect and report the presence of smoke in the area.
3. The system is able to automatically detect and report the presence of flames in the area.
4. The system is able to automatically shut off all gas and electrical supplies in the event of a fire.
5. The system is able to automatically notify the fire department in the event of a fire.
6. The system is able to automatically notify the occupants of the building in the event of a fire.
7. The system is able to automatically evacuate the building in the event of a fire.

8.2 User Acceptance Testing:

The IoT based Intelligent Fire Management System using ESP32 is tested by the user to check if it is working as expected. The user should be able to see the following:

-The system should be able to detect a fire and send an alert to the user.

-The system should be able to turn on the sprinklers automatically when a fire is detected.

-The system should be able to track the location of the fire and provide updates to the user.

-The system should be able to provide information about the intensity of the fire.

9.RESULTS

Fire alarms are prime necessity in modern buildings and architecture, especially in banks data centres and gas stations. They detects the fire in ambience at very early stage By sensing smoke or slash and heat and raise an alarm which warns people About the fire and furnish sufficient time to take preventive measures it not only prevents a big losses caused by deadly fire but sometime proves to be live savers. Fire alarm is a device that detects the presence of fire and atmospheric changes relating to smoke. The fire alarm operates to alert people to evacuate a location in which a fire or smoke accumulation is present. When functioning properly, if fire alarm will sound too naughty five people on and immediate fire emergency. The distinct sound exist to allow the notification to be hard the fire alarm constructed by this project Is reliable at low-cost. The IoT based intelligentfire management system can help reduce the number of fire incidents, as well as the damage caused by them. It can also help improve the efficiency of firefighting operations.

9.1 Performance Matrices:

There are many performance matrices that can be used to evaluate the performance of an IoT-based intelligent fire management system. Some of the most important performance matrices include:

1. Response time: This is the time taken for the system to detect a fire and activate the fire suppression system.
2. Accuracy: This is the percentage of fires that are accurately detected by the system.
3. False positive rate: This is the percentage of times that the system incorrectly detects a fire.
4. False negative rate: This is the percentage of times that the system fails to detect a fire.
5. system availability: This is the percentage of time that the system is operational.
6. MTBF: This is the mean time between failure of the system.
7. Maintainability: This is the ease with which the system can be maintained.

10.ADVANTAGES AND DISADVANTAGES:

There are several advantages and disadvantages to using an IoT-based Intelligent Fire Management System:

Advantages:

1. Better and more accurate fire detection – With an IoT-based system, sensors can be placed in strategic locations around a building or area to more accurately detect a fire.

2. Faster response times – An IoT-based system can immediately notify the relevant authorities when a fire is detected, resulting in faster response times.

3. Reduced false alarms – With an IoT-based system, false alarms can be reduced as the system can be programmed to only notify the authorities when certain conditions are met (e.g. a certain level of heat is detected).

4. Greater efficiency – An IoT-based system can automate many of the tasks associated with fire management, such as dispatching firefighters and keeping track of firefighting resources.

5. Increased Efficiency: By automating tasks and processes, an IoT system can help fire departments become more efficient in their operations.

6. Improved Safety: With real-time monitoring and alerts, an IoT system can help keep firefighters safe by providing them with timely information about potential hazards.

7. Enhanced Collaboration: An IoT system can enable better 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 both the 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 manual labor.

<u>Disadvantages:</u>

1. Implementation costs – An IoT-based Intelligent Fire Management System can be expensive to implement, especially if a large number of sensors are required.

2. Maintenance costs – An IoT-based system can also be expensive to maintain, as sensors and other hardware will need to be regularly checked and replaced.

3. Security risks – As IoT-based systems rely on networked devices, they can be vulnerable to hacking and other security risks.

4. Privacy concerns – Some people may be concerned about the privacy implications of having an IoT-based system, as it will be constantly collecting data about people and their activities.

5. Complexity: An IoT system can be complex to set up and maintain, and may require specialized skills and knowledge.
6. Reliability: An IoT system can be subject to outages and other disruptions, which could impact the reliability of the system.
7. Security: An IoT system can be vulnerable to security threats, such as hacking and data breaches.
8. Privacy: An IoT system can collect and store large amounts of data, which could potentially be used to infringe on the privacy of individuals.
9. Cost: An IoT system can be expensive to implement and maintain.

11. CONCLUSION:

The proposed system, presents the advancement of Internet technology in 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 system was verified.

There is a general agreement over the fire and protection segments that at 220,000 for every annum the degrees of bogus and undesirable alarms radiating from fire alarm and discovery frameworks is excessively high. Bogus and undesirable alarms squander fire and salvage administration assets; cause superfluous and costly interruption to end-clients which can bring about the loss of trust in frameworks and has seen a few frameworks turned off. As fire alarm and identifications frameworks are so firmly inserted into the clearing systems and strategies created to meet the necessities of

Building Regulations and Fire Safety Law their utilization is far reaching and there are entrenched outsider accreditation plans for producers and installers. The item measures and testing systems anyway stay quiet on the reasons for bogus alarms.

Benefits of use IoT in industry:

Elimination of long wiring.

Web based remote monitoring.

Immediate action on failures.

Ease of maintenance.

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 authorities in case of a fire.There is no one-size-fits-all answer to this question, as the scope of an IoT-based intelligent fire management system will vary depending on the specific needs of the organization deploying it. However, some potential applications of such a system include early detection and notification of fires, automatic fire suppression, and remote monitoring and control of fire safety systems.

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 action accordingly. The system will also be able to automatically detect the fire and send alerts to the concerned authorities.

13. APPENDIX Source

Code:

//.....Credentials For IBM .....

Organization ID

73497z

Device

Type

iot_device

Device ID

1234

Authentication Method

use-token-auth

Authentication Token

12345678

```
//.....Project Source Link on Wokwi.....

Wokwi Link - https://wokwi.com/projects/347685130732569171

#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt
#include "DHT.h"// Library for dht sensor
#define DHTPIN 15       // what pin we're connected to
#define DHTTYPE DHT22 // define type of sensor DHT 22 #define
LED 2

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

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

//-------credentials of IBM Accounts------

#define ORG "88653s"//IBM ORGANITION ID
#define DEVICE_TYPE "iot_device"//Device type mentioned in ibm watson IOT
Platform
#define DEVICE_ID "1234"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "12345678"                    //Token
String data3; float h, t; const float BETA = 3950; // should match the Beta
Coefficient of the thermistor //-------- Customise the above values --------

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Name
```

```
char publishTopic[] = "iot-2/evt/Data/fmt/json"; // topic name and type of
event perform and format in which data to be send char subscribetopic[] = "iot-
2/cmd/test/fmt/String"; // cmd REPRESENT command type AND
COMMAND IS TEST OF FORMAT STRING

char authMethod[] = "use-token-auth"; // authentication method char

token[] = TOKEN;

char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; //client id



// _____

WiFiClient wifiClient; // creating the instance for wificlient

PubSubClient client(server, 1883, callback ,wifiClient); //calling the predefined
client id by passing parameter like server id,portand wificredential


void setup() // configureing the ESP32



{

  Serial.begin(115200);

  dht.begin();

  delay(10);

  Serial.println();

  wificonnect();

  mqttconnect();


  Serial.begin(9600);

  analogReadResolution(10);

  pinMode(18,INPUT); pinMode(14,OUTPUT);

  pinMode(12,OUTPUT);
```

```
}


void loop() // Recursive Function
{

  h = dht.readHumidity(); t =
  dht.readTemperature();
  Serial.print("Temperature:
  ");
  Serial.println(t);
  Serial.print("Humidity:");
  Serial.println(h);

  PublishData(t, h);
  delay(1000);      if
  (!client.loop()) {
  mqttconnect();
  }

//.............Analog Temperature Sensor.....................

  int analogValue = analogRead(18); float celsius  = 1 / (log(1  / (1023. /
  analogValue - 1)) / BETA + 1.0 / 298.15) +
36.4;
  Serial.print("Temperature: ");
  Serial.print(celsius);
  Serial.println(" ℃");
```

```
  Serial.print("Alert..!");
if(celsius >= 35)
  digitalWrite(14,
  HIGH); else
  digitalWrite(14, LOW);
  delay(1000);

 }

 /*.....retrieving to Cloud. ..... */

 void PublishData(float temp, float humid) {
  mqttconnect(); //function call for connecting to ibm

  /* creating the String in in form JSon to update the data to ibm cloud

  */

  String payload = "{\"Data\":{\"temperature\":";
  payload += temp; payload += ","
  "\"humidity\":"; payload += humid; payload +=
  "}}";
  Serial.print("Sending payload: ");
  Serial.println(payload);
```

```
  if (client.publish(publishTopic, (char*) payload.c_str())) {
    Serial.println("Publish ok"); // if it sucessfully upload data on the cloud then it will print publish ok in Serial monitor or else it will print publish failed
    Serial.println("If Temperature increased,the alarm and alert light would indicates. ");
  } else {
    Serial.println("Publish failed");
  }


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


     initManagedDevice();
     Serial.println();
  } } void wificonnect() //function defination for
wificonnect {
  Serial.println();
  Serial.print("Connecting to ");


  WiFi.begin("Wokwi-GUEST", "", 6);//passing the wifi credentials to establish the connection while (WiFi.status() != WL_CONNECTED) {
    delay(500);
```

```
    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");


  }
}


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++) {
  Serial.print((char)payload[i]); data3
    += (char)payload[i];
  }
```

```
  Serial.println("data: "+ data3); if(data3=="lighton")
  {
Serial.println(data3);
digitalWrite(LED,HIGH);

  }

  else
  {
Serial.println(data3);
digitalWrite(LED,LOW);

  } data3="";

}
```

//.....Python Script for Random Outputs of Temperature and Humidity.....

```
import time import sys import ibmiotf.application import ibmiotf.device import
random


#Provide your IBM Watson Device
Credentials organization = "bxobbs"
deviceType = "b5ibm" deviceId = "b5device"
authMethod = "token" authToken =
"b55m1eibm"
```

```
# Initialize GPIO


def myCommandCallback(cmd): print("Command
 received: %s" % cmd.data['command'])
 status=cmd.data['command'] if status=="lighton":
        print ("led is on")
    else :
        print ("led is off")

    #print(cmd)




 try:
        deviceOptions = {"org": organization, "type": deviceType, "id": deviceId,
"auth-method": authMethod, "auth-token": authToken} deviceCli
        = ibmiotf.device.Client(deviceOptions)
        #.....
except Exception as e:
        print("Caught exception connecting device: %s" % str(e)) sys.exit()


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

```
while True:
        #Get Sensor Data from DHT11


        temp=random.randint(0,100)
        Humid=random.randint(0,100)
data = { 'temp' : temp, 'Humid': Humid }
        #print          data          def
        myOnPublishCallback():
        print                ("Published
        Temperature  =  %s  C"  %
        temp, "Humidity = %s %%"
% Humid, "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.commandCallback = myCommandCallback


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

Github and Project Demo link:

//.....Project Source Link on Wokwi.....

Wokwi Link - https://wokwi.com/projects/347685130732569171

//.....Project Data in json Format ...../


```
{
  "version": 1,
  "author": "JAISAKTHI",
  "editor": "wokwi",
  "parts": [
    { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": 10, "left": -60.67, "attrs": {} },
    {
      "type": "wokwi-led",
      "id": "led1",
      "top": -109,
      "left": -244.4,
      "attrs": { "color": "red" }
    },
    {
      "type": "wokwi-dht22",
      "id": "dht1", "top":
      -70.9,
      "left": 157.2,
      "attrs": { "temperature": "36.4", "humidity": "46.5" }
    },
    {

      "type": "wokwi-ntc-temperature-sensor",
      "id": "ntc1",
```

```
  "top": -69.55,
  "left": 253.55,
  "rotate": 90,
  "attrs": {}
},
{

  "type": "wokwi-resistor",
  "id": "r1",
  "top": 169.5,
  "left": -190.59,
  "attrs": { "value": "5600" }
  },
  {

    "type": "wokwi-buzzer",
    "id": "bz1",
    "top": -118.83,
    "left": -378.64,
    "attrs": { "volume": "0.1" }
  }
],
"connections": [
  [ "esp:TX0", "$serialMonitor:RX", "", [] ],
  [ "esp:RX0", "$serialMonitor:TX", "", [] ],
  [ "dht1:GND", "esp:GND.1", "black", [ "v0" ] ],
  [ "dht1:SDA", "esp:D15", "green", [ "v0" ] ],
  [ "ntc1:GND", "esp:GND.1", "black", [ "v0" ] ],
```

```
    [ "ntc1:VCC", "esp:3V3", "red", [ "v0" ] ],
    [ "led1:C", "r1:1", "black", [ "v0" ] ],
    [ "r1:2", "esp:GND.2", "black", [ "v0" ] ],
    [ "led1:A", "esp:D14", "green", [ "v-0.86", "h89.56", "v199.46" ] ],
    [ "ntc1:OUT", "esp:D18", "green", [ "v0" ] ],
    [ "bz1:1", "esp:GND.2", "black", [ "v0" ] ],
    [ "bz1:2", "esp:D14", "green", [ "v0" ] ],
    [ "dht1:VCC", "esp:3V3", "red", [ "v0" ] ],
    [ "dht1:NC", "dht1:GND", "black", [ "v0" ] ]

  ]
}
```

GITHUB LINK : https://github.com/santhoshS19