Industry-specific intelligent fire management system

1. INTRODUCTION

1.1 Project Overview

Fire Detectors play a very important role in Industries, Shops, Malls, Residential complexes, and parking areas. They help in detecting fire or smoke at an early stage and can help in saving lives. Commercial Fire detecting systems usually have an alarm signaling, with the help of a buzzer or Siren. We have designed an IOT based Fire Alerting System using Temperature and a smoke sensor. This project would not only signal the presence of fire in a particular premise but will also send related information through IOT to the IBM Cloud with cloud object storage using the services like IBM Watson platform and Node-Red services, we get the best result using MIT Inventer application.

The Internet of Things (IoT) is basically the network of 'thingValues' by which physical things can exchange data with the help of sensors, electronics, software, and connectivity. These systems do not require any human interaction. In this Arduino fire alarm system using temperature and smoke sensors using the IOT project, we can send LIVE information like Temperature, Smoke detected by a particular device to the Fire Department.

1.2 Purpose

Fires caused due to electrical and heating equipment are the top-reasons for any residential fire. If the equipment is not properly monitored and maintained, they can undergo electric malfunctions and overheat, resulting in fires. In order to reduce the risk of fires, sensors can be used for performing preventative maintenance on all the household types of equipment like- electrical and fire heating systems. Sensors can be placed on equipment for monitoring the heat signatures and establish a baseline performance to indicate when the equipment has exceeded the prescribed safety norms. The use of sensors could be used for monitoring equipment so that individuals are informed about the unexpected temperature spikes or even equipment misfires.

2. LITERATURE SURVEY

Paper 1:

IOT BASED FIRE DETECTION AND AUTOMATIC WATER SPRINKLER SYSTEM

Published year: 2022

Author name: D Teja, M.Surajkhan, k Jyothi

Journal name:
International journal of engineering applied science and technology.

Summary:
In this paper, Fire detection systems, particularly vision-based systems, identify flames before any loss or destruction occurs. In this model, a novel vision-based technology is created that uses a camera to detect flames over long distances. An immediate alert is generated on android application. The goal of the proposed system is to notify the remote user when a fire accident occurs. By using camera method, the report is automatically generated and delivered to the person immediately following the fire is detected in any part of the frame using Wi-Fi/GSM.

Methodology:
Following the detection of a fire, our technology will take real-time photos of the surrounding area. The flame sensor determines whether or not there is a fire or flame present. A photo transistor is used in this explicit flame detector. The infrared spectral band is used by flame detection systems. Carbon dioxide, which is produced by the combustion of organic compound materials, has a resonance frequency in this range. Put anything that can catch fire in front of the flame sensor. The flame sensor is triggered when it detects a fire or flame. As temperature increases the temperature sensor will detect and it will trigger the buzzer and buzzer will blow. The water pump is connected to a IC. If a flame is detected, IC activates the dc motor and water pump. The sprinklers connected to the pump will sprinkle the water throughout the fire affected area.

Paper 2:

Fire Detection, Monitoring and Alerting System based on IOT

Published year: 2019

Author name : Shreya Gosrani, Abhishek Jadhav, Krutika Lekhak D Chheda

Journal name:
International Journal of Research in Engineering, Science and Management

Summary:
Internet of Things refers to connecting things and people through internet, it has imposed itself as the New business practices in different sectors. To make quick

and efficient response in real time, IoT enhances the way and provides emergency managers with the necessary information and communication to make use of those assets. In this paper it is proposed that a quick response for fire hazards is evaluated and examined by using IoT based model. Fire is one of the major reasons of accidental deaths in the world. To implement this proposed system a low-cost Wi-Fi module, gas detection sensor, Flame detection sensor, buzzer to alert and temperature sensors are used. The sensors detects and alerts the local emergency with the data collected by the system, and alerts organizations like fire departments, police stations and hospitals by sending the exact location to both user and operator through module which all are well connected with.

Methodology:
IoT framework concentrates on public safety and livelihood service sector The fire detecting system with IoT standardized design methods The spark Detection sensor PT333B is used to sense the spark, the Flammable gas sensor MQ-6 is used to detect the gases like LPG/LNG and the GPS module is to obtain device location. These sensors along with Wi-Fi micro-controller are connected via Internet.

Paper 3:

IoT Based Automatic Fire Alarm System

Published year: 2020

Author name: A.Jeevanandham, SivamuruganP

Journal name: Bulletin of scientific research

Summary:
Fire identifiers are utilized to recognize the fire or smoke at a beginning time and can help in sparing lives. Right now, IOT based alarm has been planned utilizing temperature and smoke sensor. By utilizing the temperature sensor, smoke sensor and there is a simple to advanced convertor, which changes over the simple signs got at the sensor end to computerized and afterward transmits them to a smaller scale controller and to the Arduino. The small-scale controller is modified to turn on the ringer, when the temperature and the smoke arrive at an edge esteem. Simultaneously, Arduino sends the information to the Wi-Fi module ESP8266. ESP8266 will then the accompanying information to the IOT site, where, approved individuals can take fitting measure so as to check the fire. The gadget id is the one-of-a-kind id given to a gadget, which would enable the work force to get data identified with the area, where the fire is detected

Methodology:
Iot must be self-contained for searchoperation, decision making based on the real-time data orcurrent condition (object detection), intelligent decision(software program) for the immediate surrounding environment or condition is to perform the task or mission.

Paper4:

Fire detection and alarm system

Publication year: 2019

Author name: Trung Luong

Journal name: HAMK Journal of Electrical and Automation Engineering

Summary:
The central target of this project was to study, analyse and design a fire detection and alarm system. This topic was suitable because it covered a basic and important aspect in our modern life. The objectives of the project were to provide information on fire alarm system in Vietnam and Finland, to show the similarities and differences with systems in both countries. For practical part, Arduino Uno was used as the control unit with other necessary components. Upon completing this project, the author has demonstrated how a fire detection and alarm system works and analysed the system standards in the above-mentioned countries.Moreover,the fire alarm system using the Aurdino Uno was tested and found to work successfully.

Methodology:
The project consist of smoke sensor ,flame sensor, LED and Buzzer, Arduino Uno as its primary components. The fire can be detected by the flame sensor and the smoke sensor also detect fires by sensing small particles in the air.

Paper5:

Fire Detection and Intimation System

Publication year: 2010/2011

Author name: Wambura Makongo

Journal name: DAR ES SALAAM INSTITUTE OF TECHNOLOGY

Summary:
This report designing and implementing Fire detection and intimation system for Dar es salaam Institute of Technology hostel. The institute hostel use Fire extinguishers in hostel but these extinguishers are not totally working. in case of fire outbreak, the Institute has no any system for fire detection which can detect smoke before it outbreak. Also all workshops, laboratories, offices, classes and Library fire
detection systems. Thus, by designing a Fire detection an Intimat

ion system which detects smoke in case of fire outbreak in the hostel will encourage or be a starting point for the Institute.

Methodology:
This project is being done under prototyping-based methodology. In this prototyping –based methodology, the analysis, design, and implementation phases are performed concurrently and all these three phases are performs repeatedly until the system is completed.

2.1. Existing problem

In this present method, substitution is essential for few causes, like failure to meet owner necessities, failure rates of framework requirements, the absence of information, and failure to give substituted fragments on this “fire alarm framework”. In the past framework where the fire might have been identified it would make only the “sprinkler engine and fan ON and also the SMS” will be dispersed to the management. There is a protection problem, which if the management does not exist, then the fire mishap might occur. To deal with these protection problems, the information is saved in the “web server”in that the management will observe from anywhere in the planet.

2.2. Reference

[1]Fire Incidents from 2001-
2014, National Crime Records Bureau (NCRB), Accessed from: https://ncrb.gov.in/.

[2]S.J. Liu, G.Q. Zhu, The Application of GIS And IOT Technology on Building Fire Evacuation, Procedia engineering, 71(2014) 577-582.

[3]S.E. Morris, T.A. Moses, Forest Fire And The Natural Soil Erosion Regime İn The Colorado Front Range, Annals of the association of American geographers, 77 (1987) 245-254 .

[4]A. Alonso-Betanzos, O. Fontenla-Romero, B. Guijarro-Berdiñas, E. Hernández-Pereira, M.I.P.Andrade, E. Jiménez, T. Carballas, An intelligent system for forest fire risk prediction and fire fighting management in Galicia, Expert systems with applications, 25 (2003) 545-554.

[5]B.U. Töreyin, Y. Dedeoğlu, U. Güdükbay, A.E. Cetin, Computer Vision Based Method For Real-Time Fire And Flame Detection, Pattern Recognition Letters, 27 (2006) 49-58.

[6]M.S.A. Azmil, N. Ya'Acob, K.N. Tahar, S.S. Sarnin, (2015) Wireless Fire Detection Monitoring System for Fire and Rescue Application, In 2015 IEEE 11th International Colloquium on Signal Processing & Its Applications (CSPA), IEEE, 84-89.

[7]R.A. Sowah, A.R. Ofoli, S.N. Krakani, S.Y. Fiawoo, Hardware Design and Web-Based Communication Modules of a Real-Time Multisensor Fire Detection and Notification System Using Fuzzy Logic, IEEE Transactions on Industry Applications, 53 (2016) 559-566.

[8]M.F. Othman, K. Shazali, Wireless Sensor Network Applications: A Study İn Environment Monitoring System, Procedia Engineering, 41 (2012) 1204-1210.

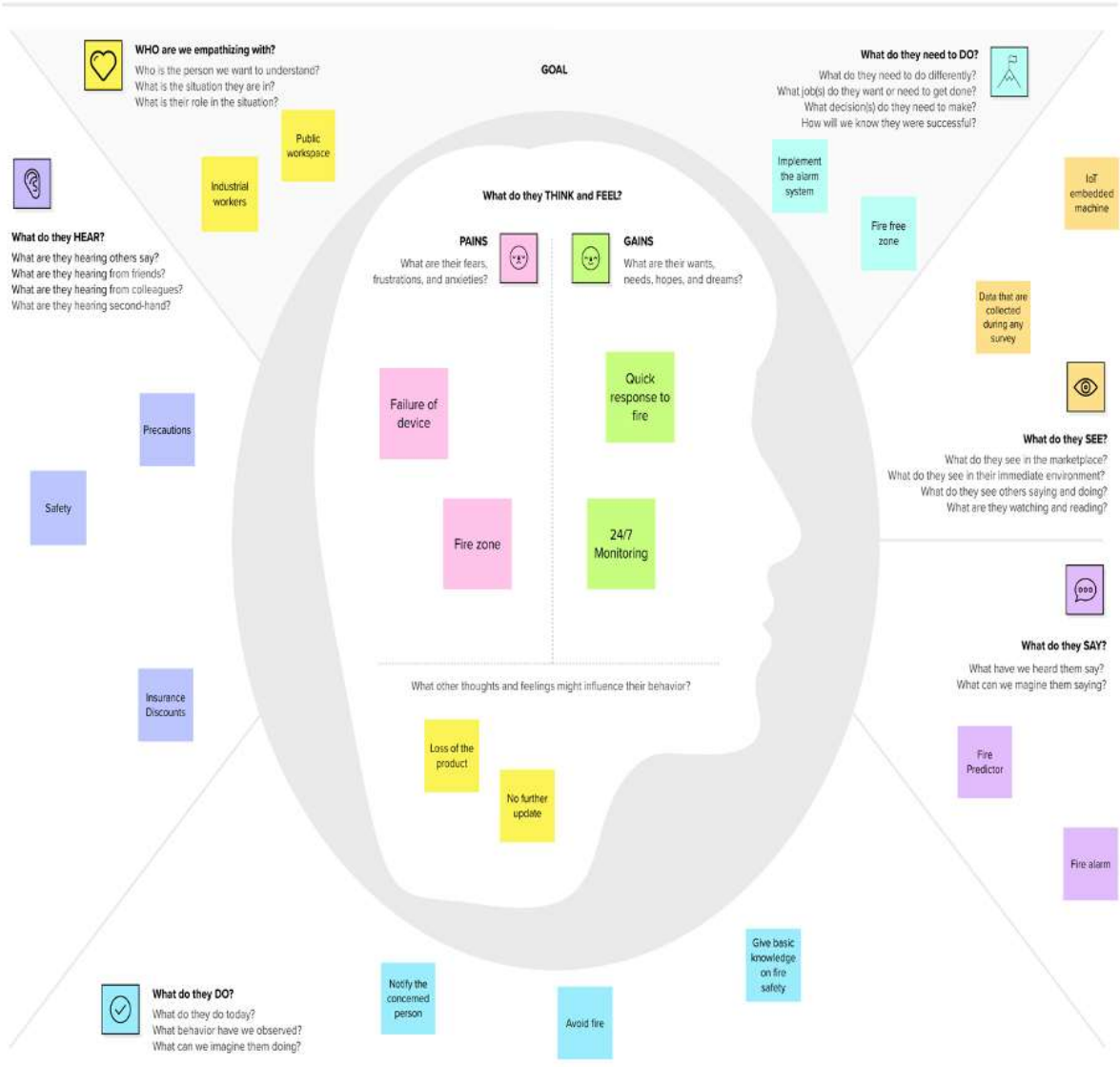
2.3. **Problem Statement Definition**

I am	An Industrialist – a person involved in the ownership and management of industry. They are innovative and willing to work hard to manage people and money.
I'm trying to	Create innovative ideas to and fuel their entire thought to shape up the industry and the safety of the company.
but	They face major problems in the fire safety of the company due to the over usage of generators or leakage of current through wires.
because	This is because of the frequent power cuts and power fluctuations in the industry which may lead to power system failure.
Which makes me feel	This makes the industrialist to greater loss and deep depression.

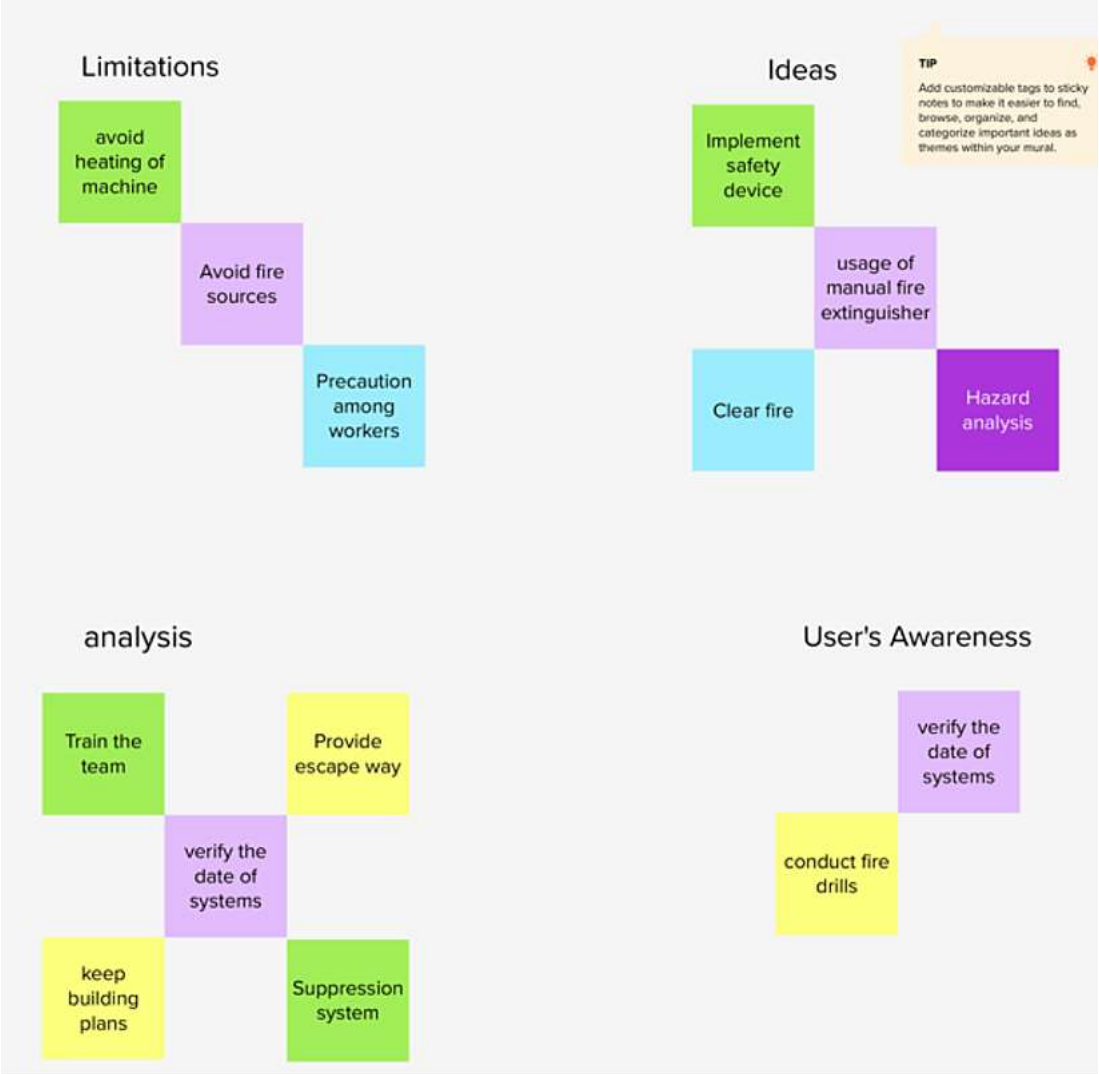
Problem Statement (PS)	I am (Industrialist)	I am trying to	But	Because	Which makes me feel
PS-1	A owner	Safeguard my company	Face fire issues	Not aware of fire safety	Loss and depressed

3. IDEATION & PROPOSED SOLUTION

3.1. Empathy Map Canvas



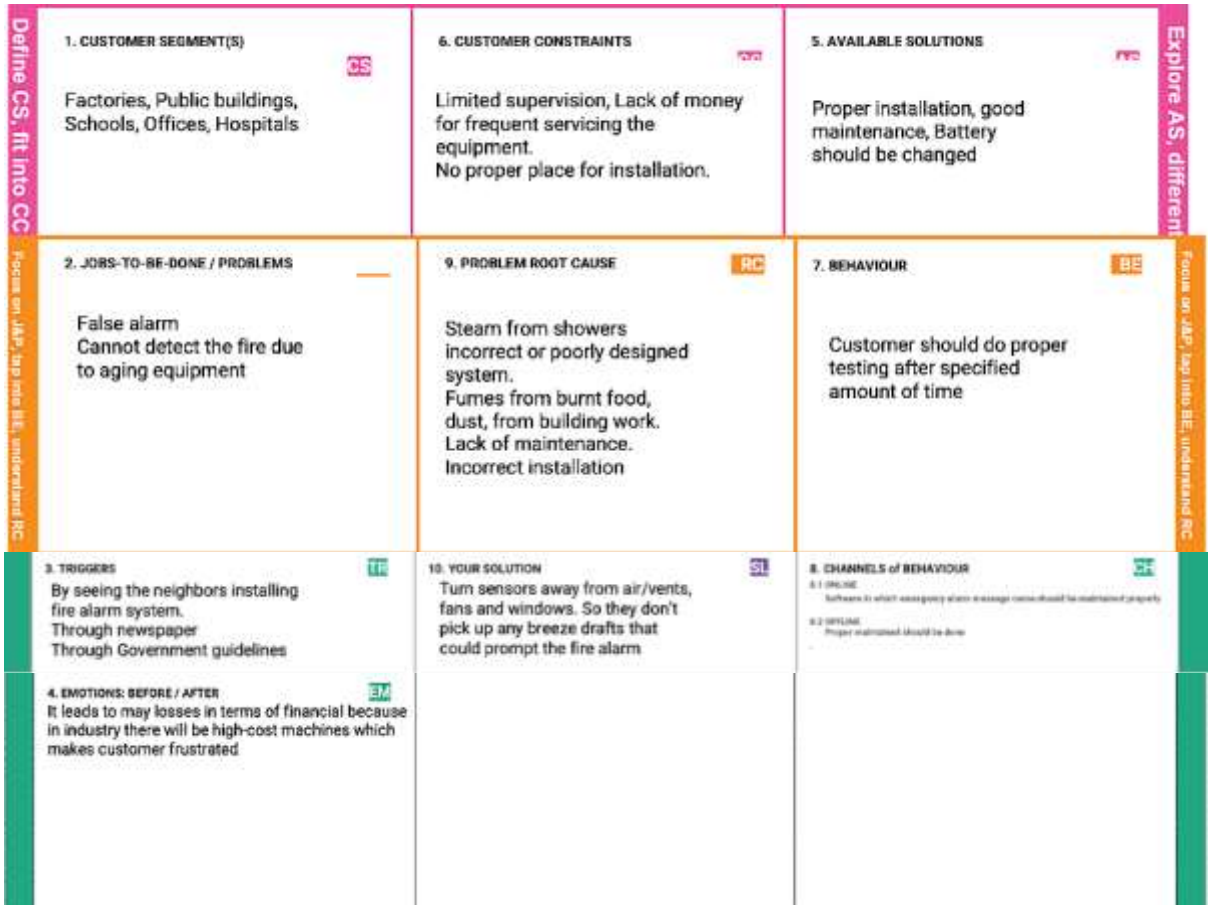
3.2. **Ideation & Brainstorming**



3.3. Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	False alarms are always an important consideration as they are expensive for fire departments majorly in terms wastes cost, time and efforts
2.	Idea / Solution description	By provided drift compensation—the ability to distinguish between subtle changes in the level of smoke detection that occur overtime due to contamination, like dust, and rapid changes resulting from a real fire.
3.	Novelty / Uniqueness	It can send maintain alerts and trouble signals to the control panel to make finding problematic devices and getting them repaired or replaced much faster
4.	Social Impact / Customer Satisfaction	High security in detection of fire and human risk is low.
5.	Business Model (Revenue Model)	1. Low installation cost 2. High secure
6.	Scalability of the Solution	High scalability because of usage of multiple sensors to identify smoke it produce least false alarms.

3.4. Problem Solution fit



4. REQUIREMENT ANALYSIS

4.1. Functional requirement

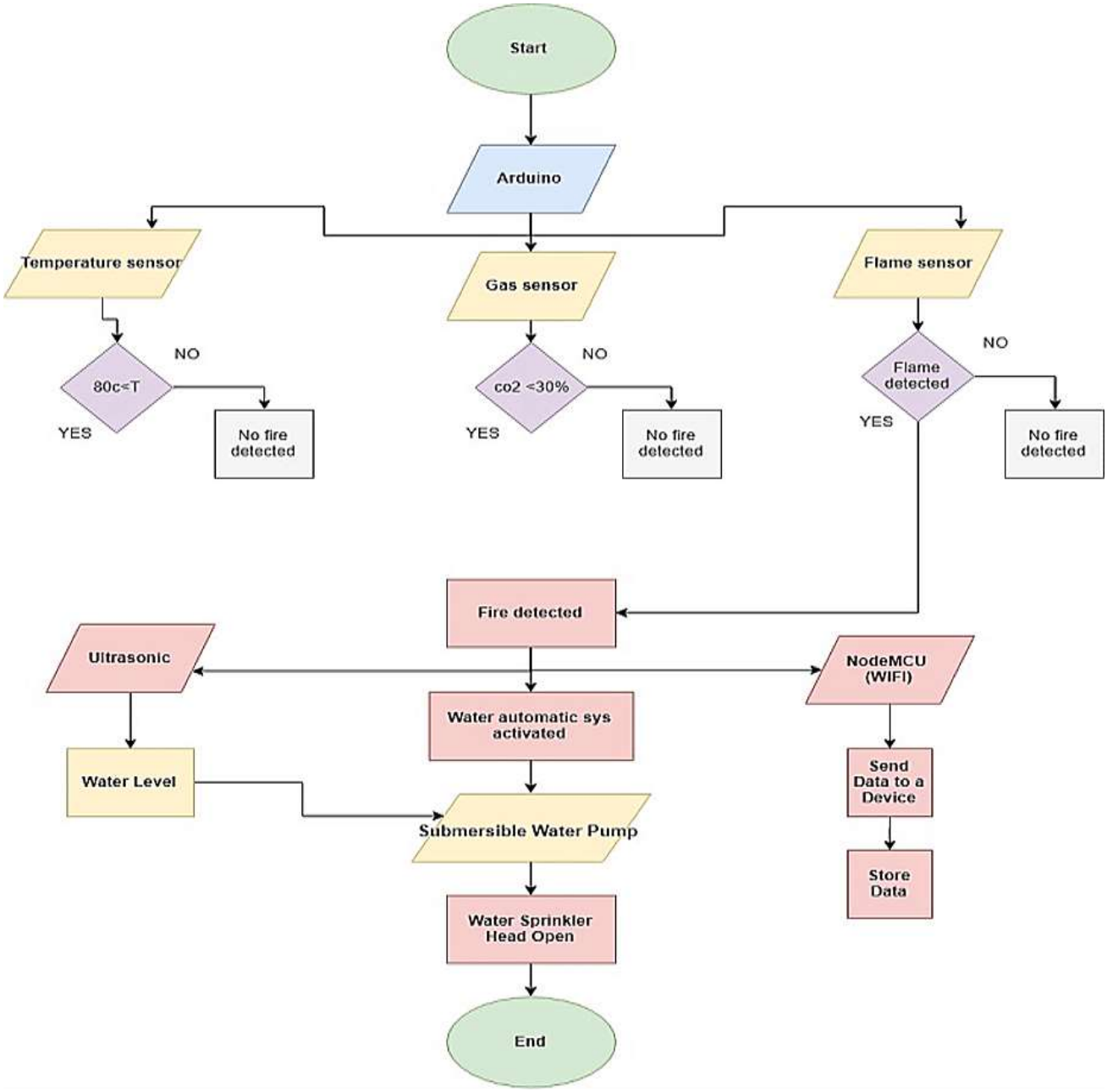
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Emergency alert	Alert through SMS. Alert through audible and visible alarms.
FR-2	Automatic transmission of notification to fire department	Notification through message. Notification through emergency call.
FR-3	Control functions	Activation of duct mounted smoke mounted smoke detector will shut down the heating ventilation and air conditioning equipment to prevent the migration of smoke to non-affected areas of the building.
FR-4	Location notification	Location of fire must be sent to fire department through alarm.

4.2. Non-Functional requirement

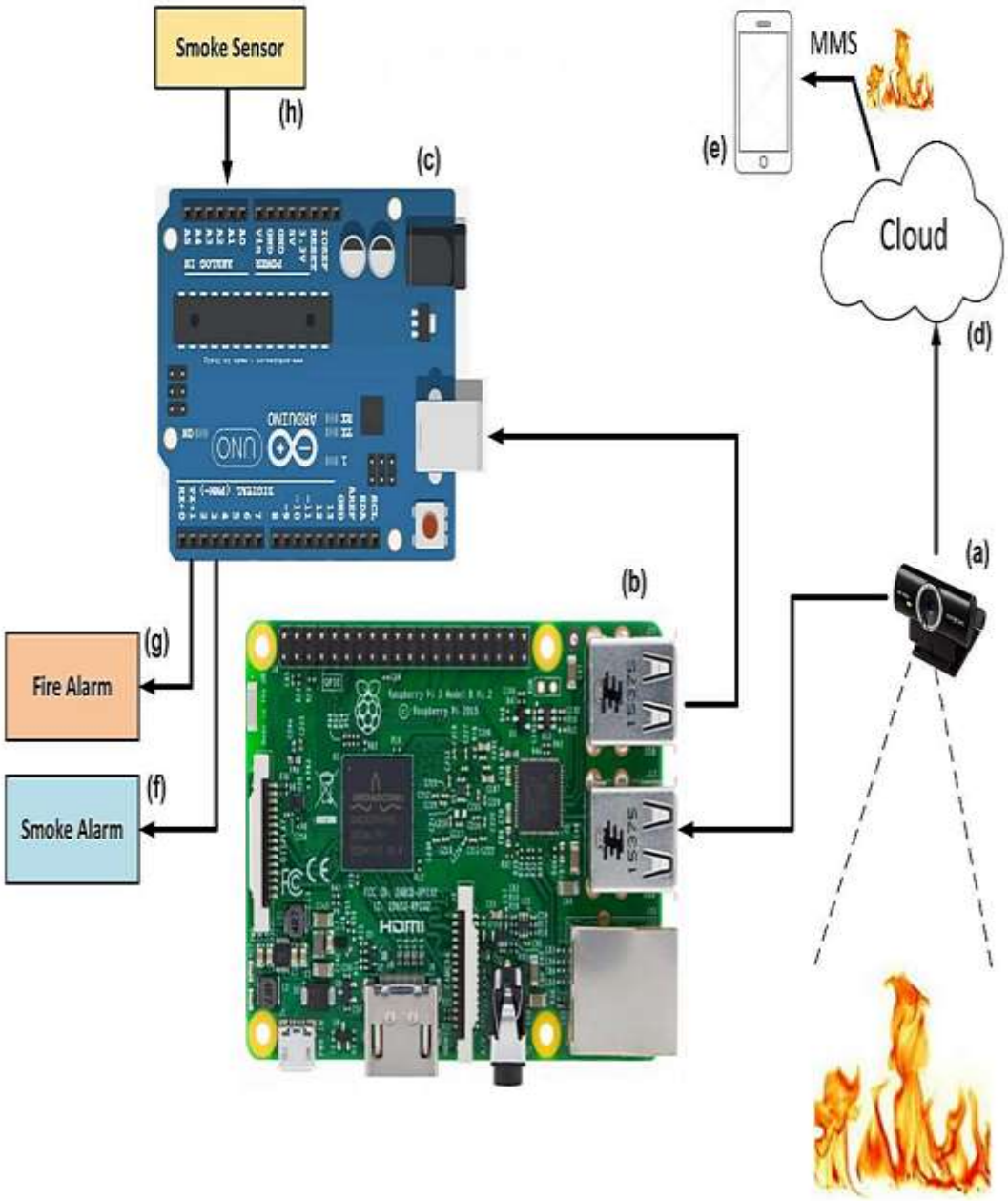
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Feature set consistency .Ease of maintenance and longevity of the system
NFR-2	**Security**	The software remains resilient in the face of attacks
NFR-3	**Reliability**	Response timer will be faster
NFR-4	**Performance**	Throughput/efficiency is high, fewer fire alarms
NFR-5	**Availability**	Ability to use the system for other types of emergency communication
NFR-6	**Scalability**	To accommodate future changes in use and occupancy

5. PROJECT DESIGN

5.1. Data Flow Diagram



5.2. **Solution & Technical Architecture**



5.3. 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

6. PROJECT PLANNING & SCHEDULING

6.1. Sprint Planning & Estimation

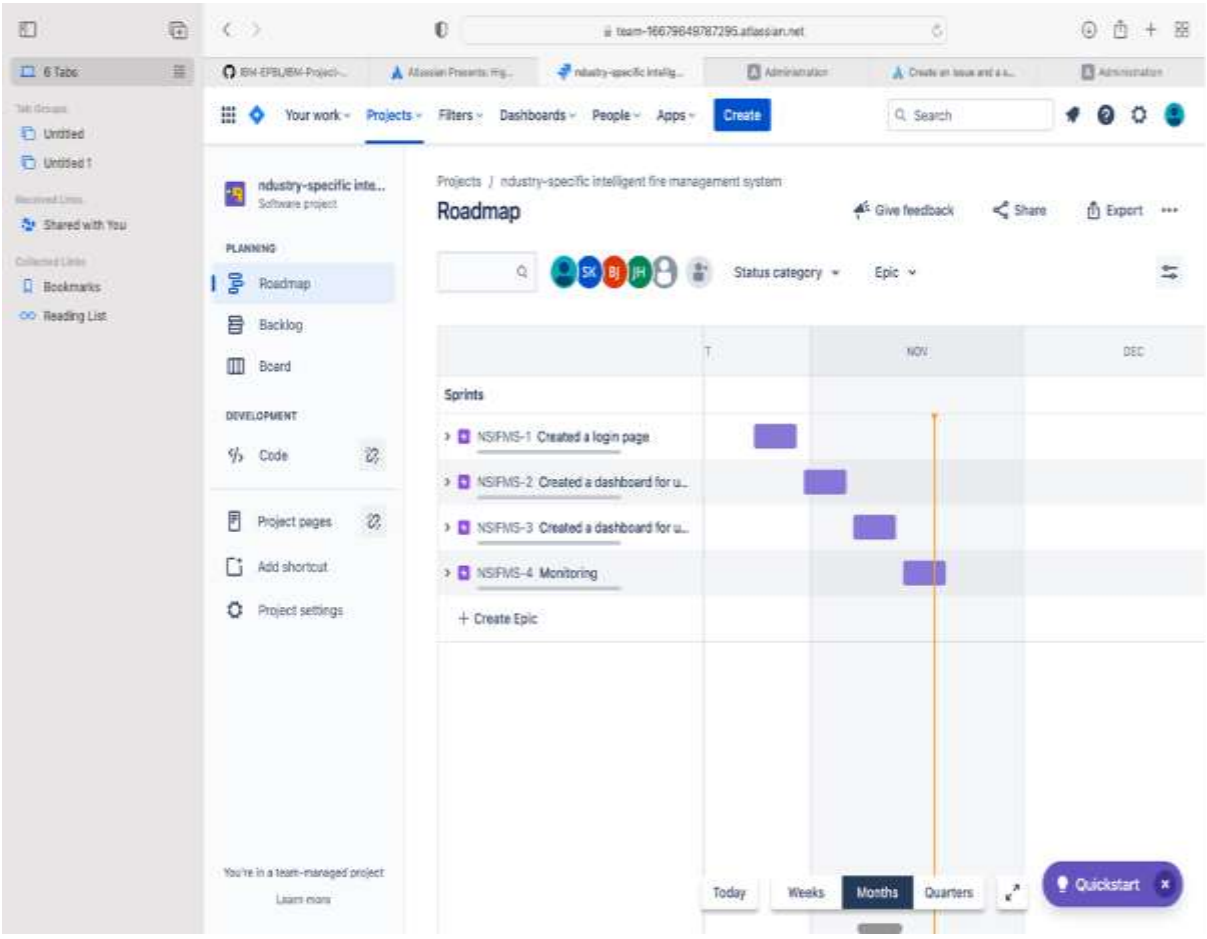
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Login	USN-1	As a customer I might ensure login credential throughgmail ease manner for the purpose of sending alert message to the owner.	2	High	Jerry Hanock T
Sprint-1	Registration	USN-2	As a user, I have registered my details and tools details in a simple and easy manner in case of fire incident, this registered system sends notification to the industrialist.	2	High	Joe Newman J
Sprint-2	Dashboard	USN-3	As a user, in case of fire in the industry I need the sprinkler to spray water on the existing fire automatically.	3	Medium	Bilson J
Sprint-3	Dashboard	USN-4	As a user, I need to safeguard my properties as well as and it will be better to send alert message to the fire department.	2	High	Karan Kumar S
Sprint-3	Dashboard	USN-5	As a user, It is good to have IoT based system to extinguish the fire without human presence	2	High	Jerry Hanock T
Sprint -4	Monitoring the environment	USN 1	User can monitor the situation of the environment from a dashboard that displayssensor information about the environment	2	High	Joe Newman J
Sprint-4	Event Notification	USN 6	Sending an alert SMS to the fire authority incase of fire	2	High	Bilson J

6.2. Sprint Delivery Schedule

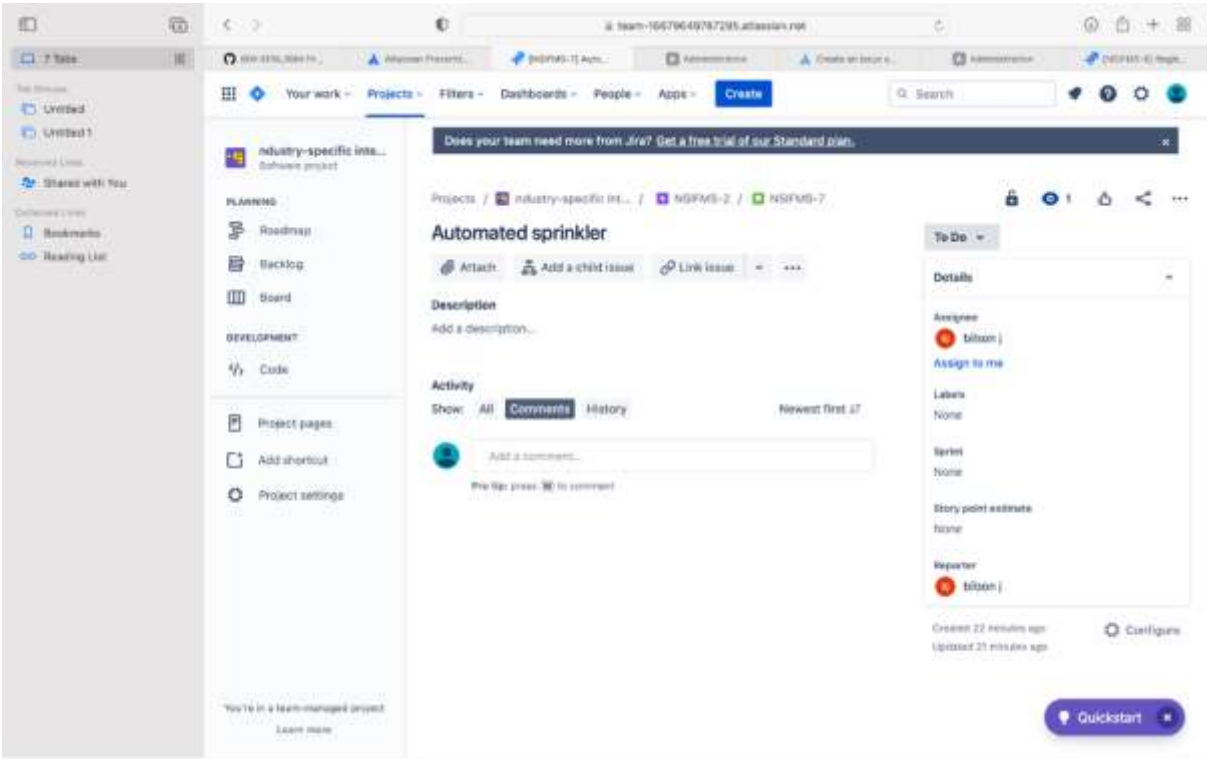
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	4	6 Days	24 Oct 2022	29 Oct 2022	4	29 Oct 2022
Sprint-2	3	6 Days	31 Oct 2022	05 Nov 2022	3	05 Nov 2022
Sprint-3	4	6 Days	07 Nov 2022	12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

6.3. Reports from JIRA

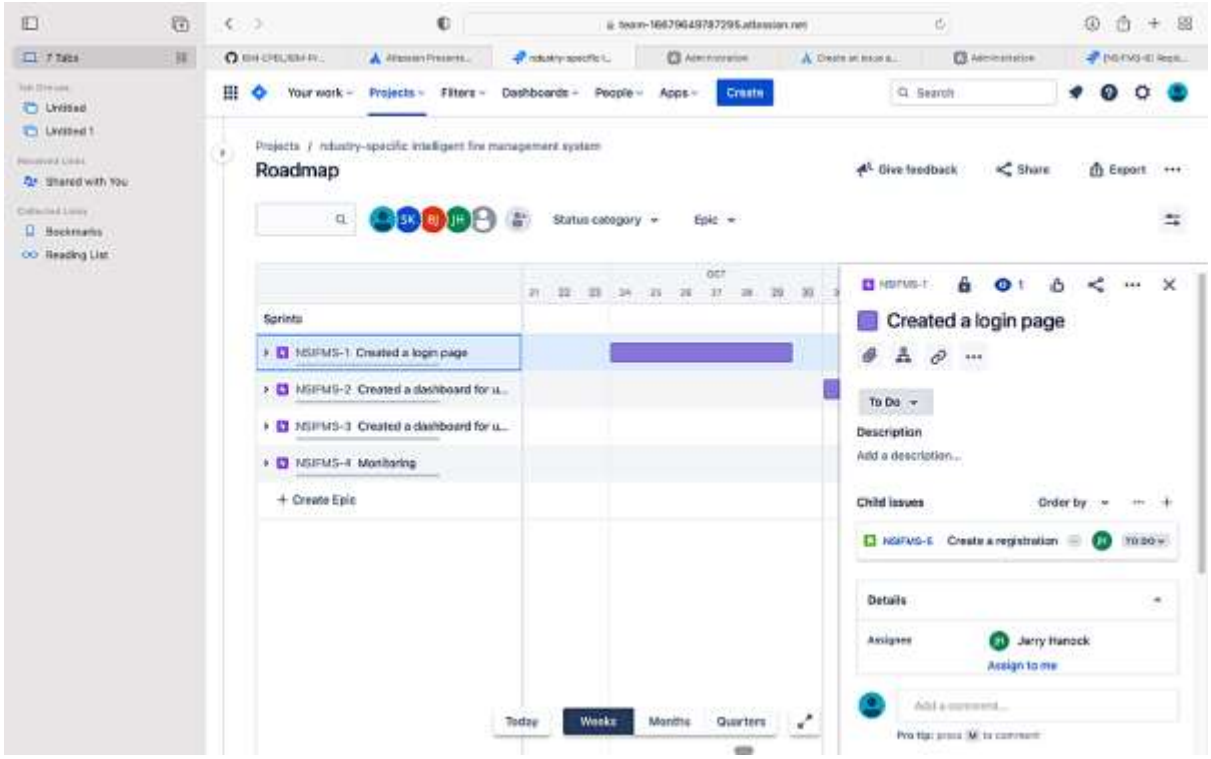
Created Project and added Team Members



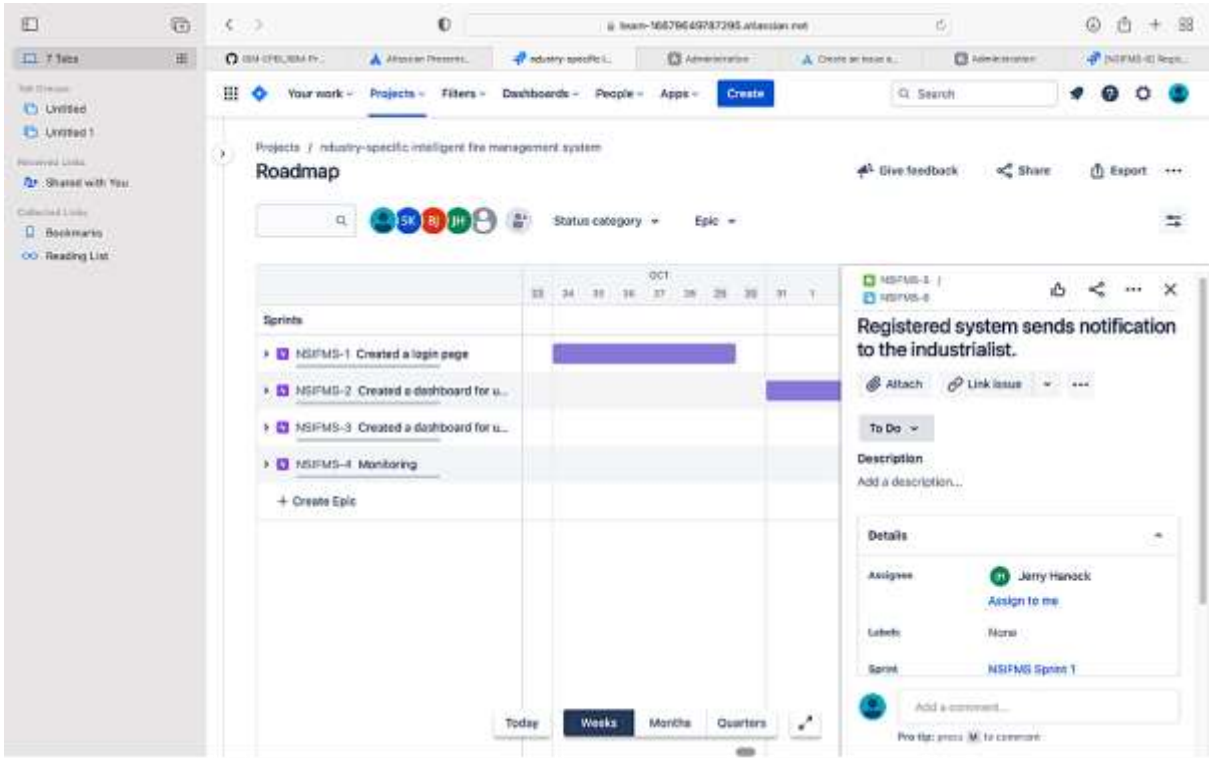
Assigning the Epics to Team Members



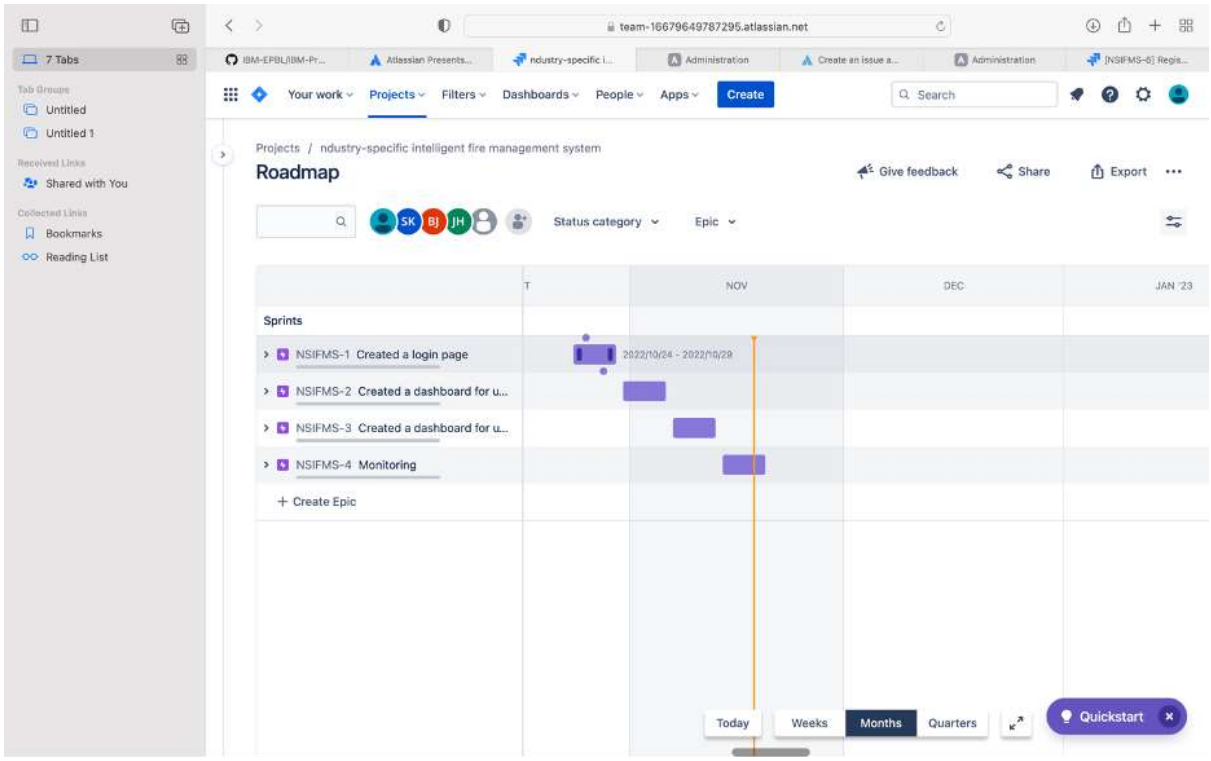
Creating story inside epic and assigning to team members with the timeline



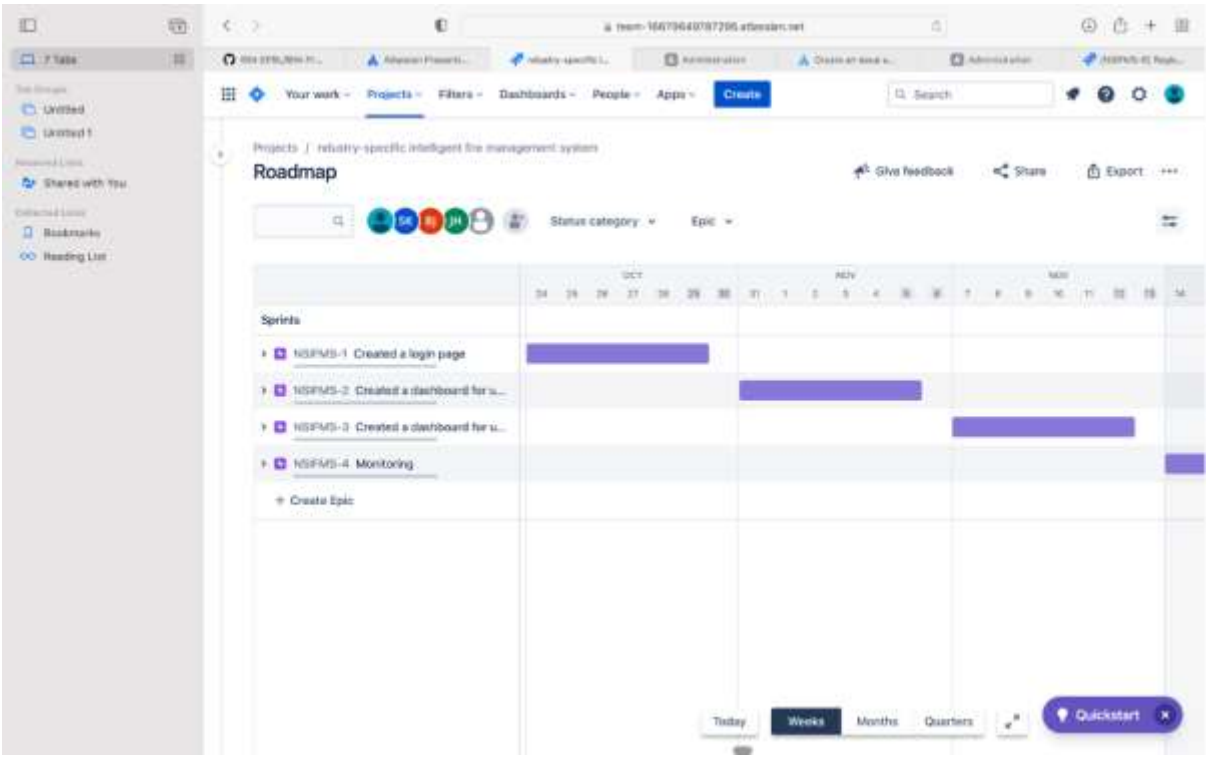
Inside a story creating feature and giving timeline



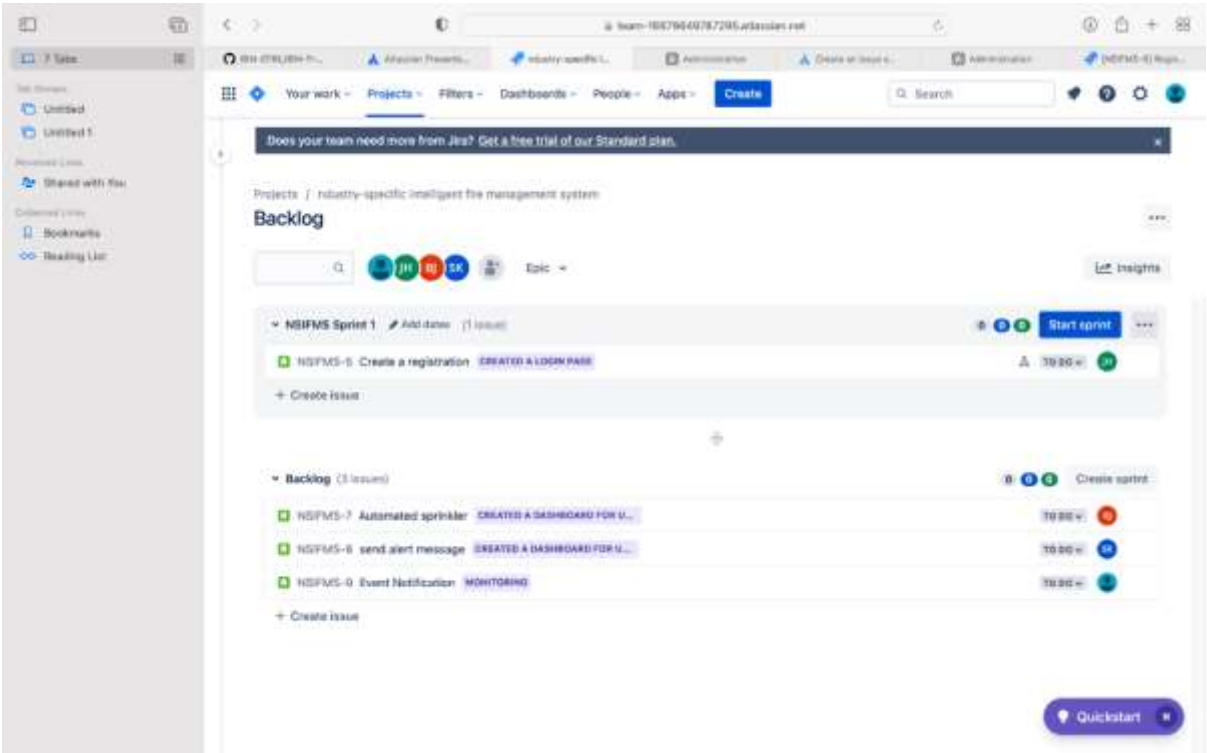
Created Timeline for each epic in roadmap



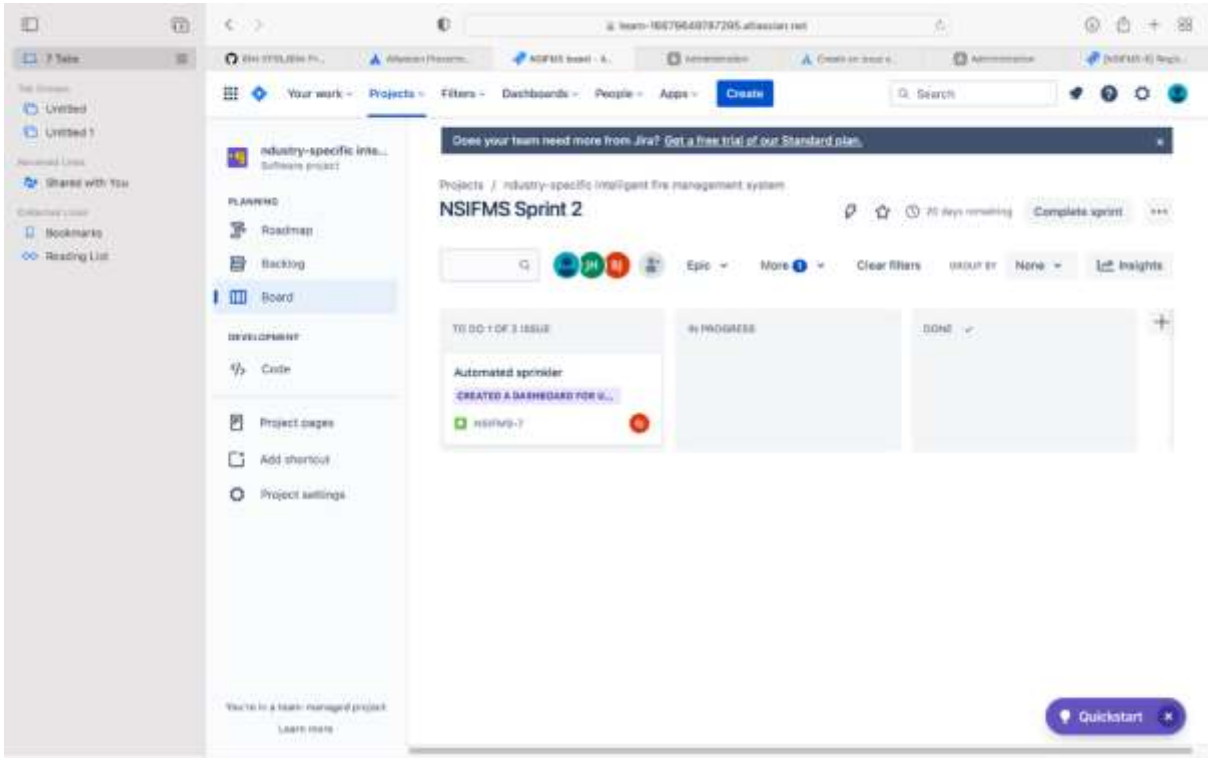
Weekly timeline for epics in roadmap



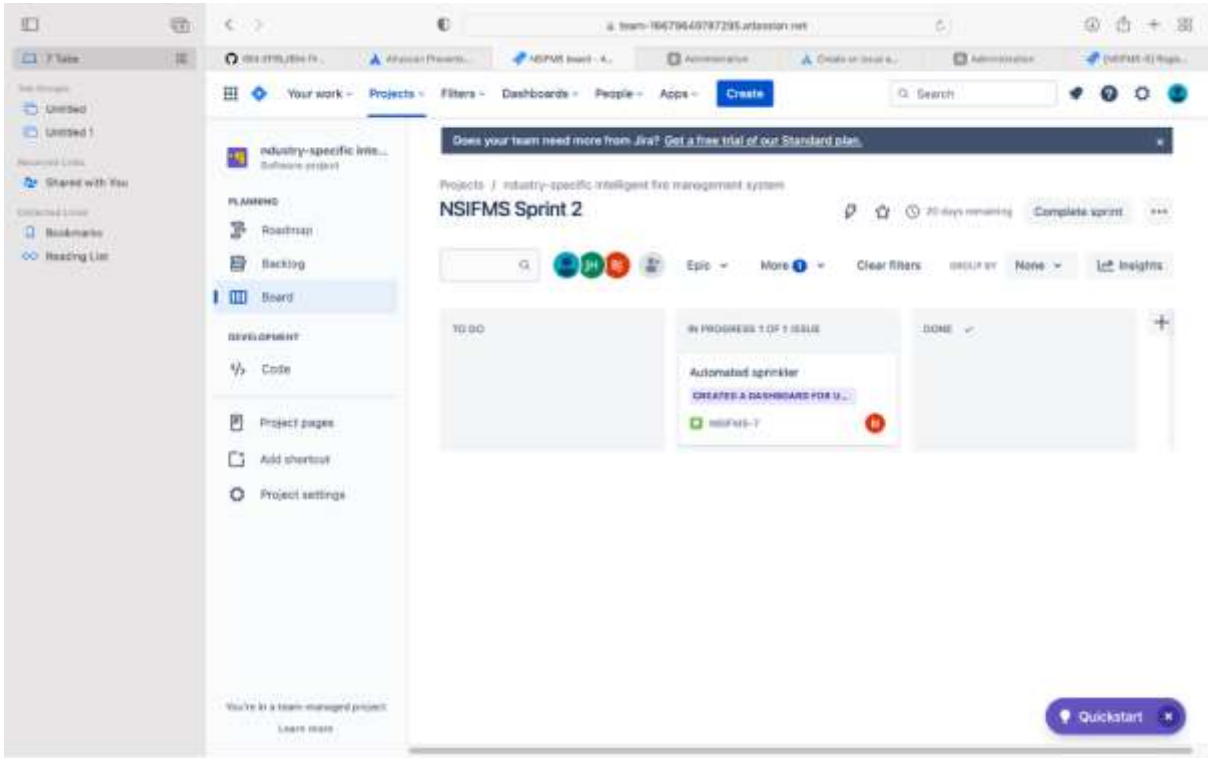
Started Sprint so the remaining in Backlog



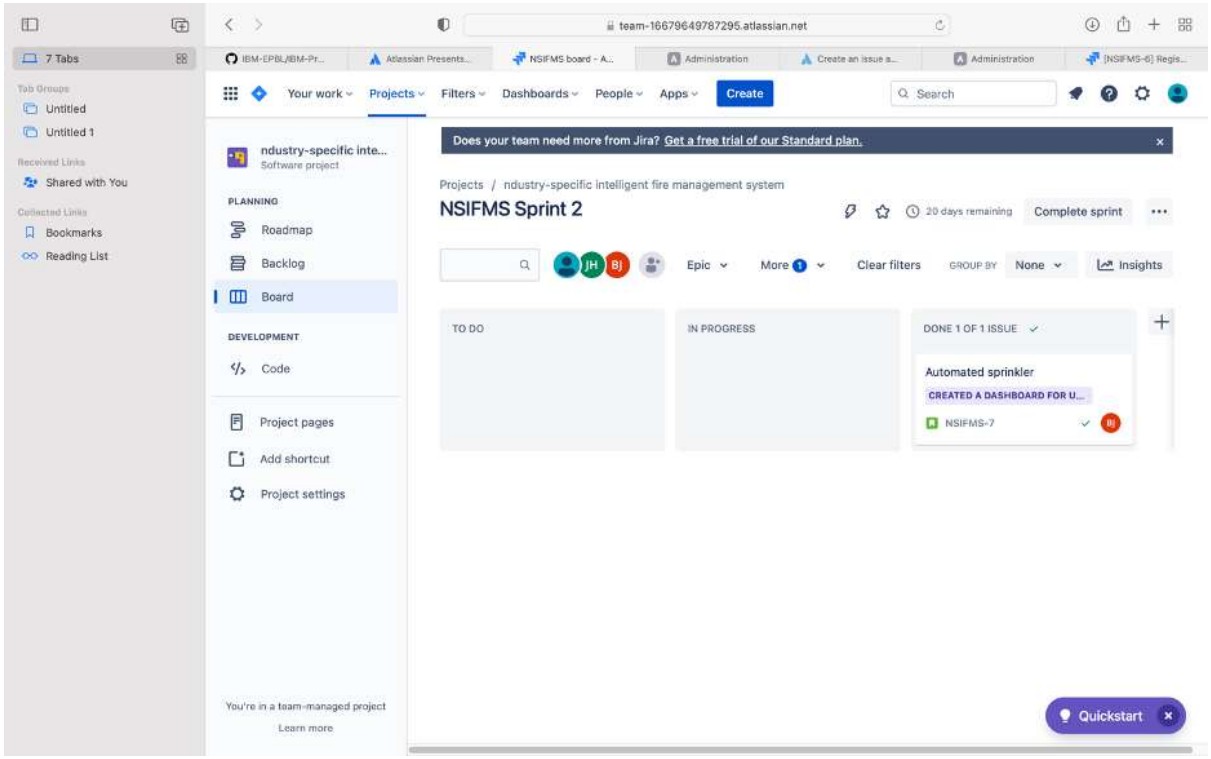
Move the cards to board when sprint is started



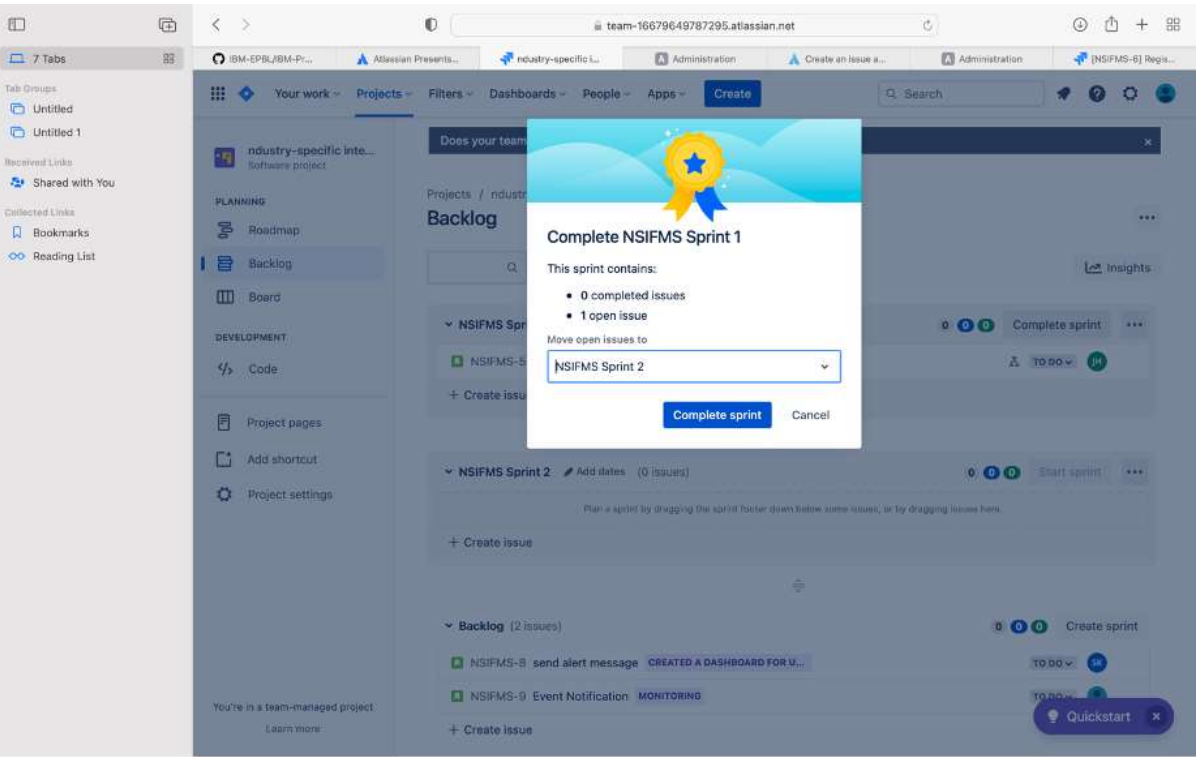
Moving card to in progress



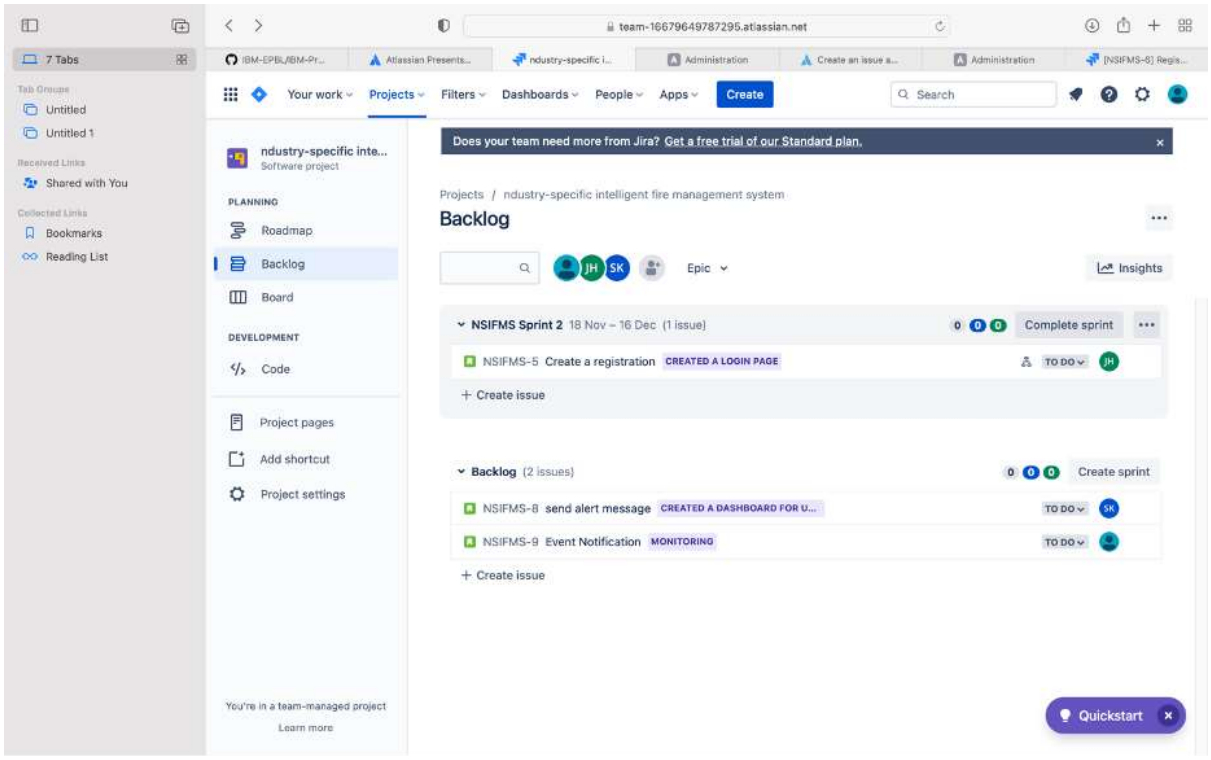
Move the card to done in board



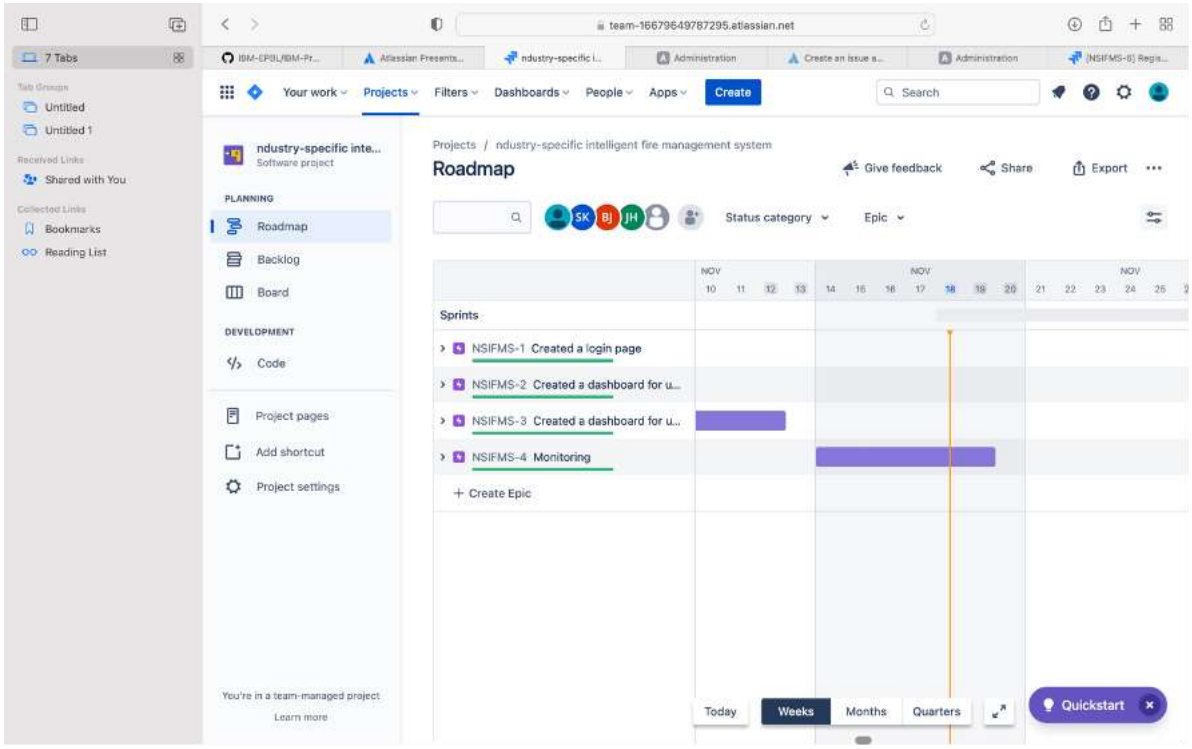
Complete the sprint



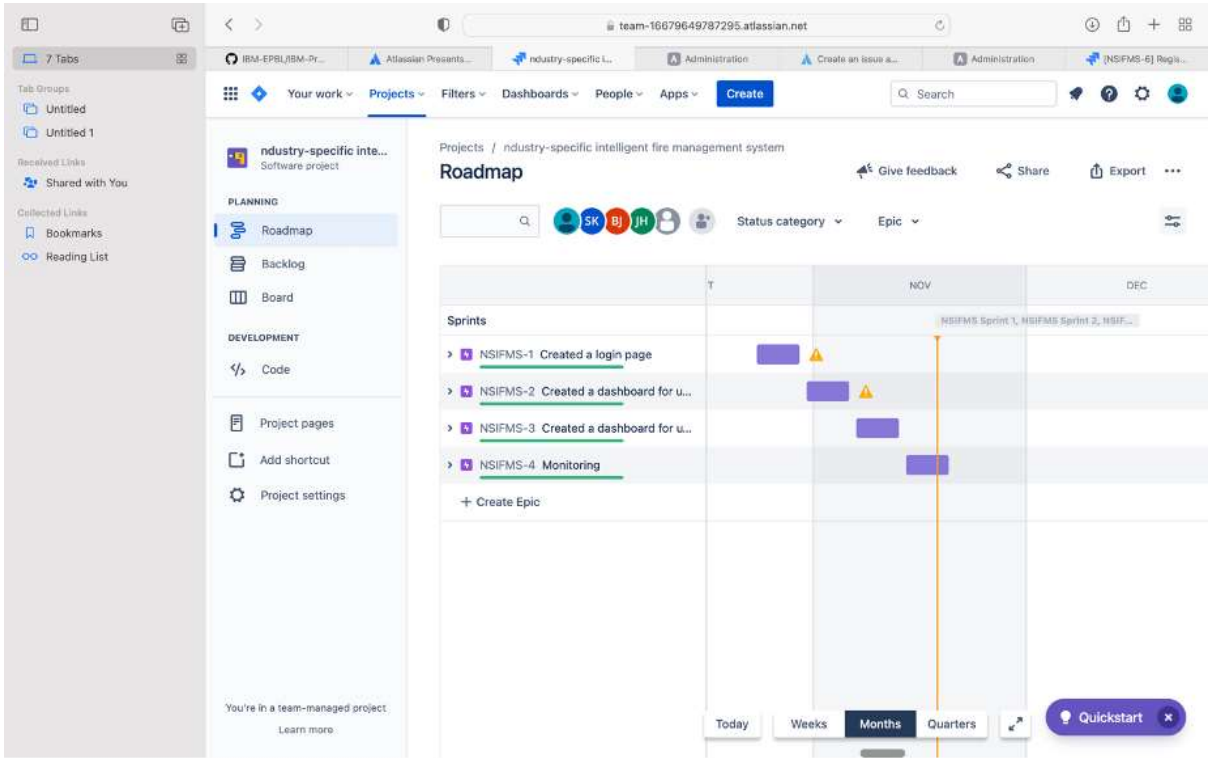
Complete the sprint by changing the status of the project



Complete all sprints in Roadmap



Complete all sprints in Roadmap Monthly



7.CODING & SOLUTIONING

7.1. **Feature 1**

Sketch.ino | diagram.json | libraries.txt | Library Manager

```
#include <WiFi.h>
#include <PubSubClient.h>
#define temp_pin 15
void callback(char* subscribetopic,byte* payload, unsigned int payloadLength);
#define ORG " ugapx5 "
#define DEVICE_TYPE "raspberrypi"
#define DEVICE_ID "12345"
#define TOKEN "12345678"
String data3;

char server[]= ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[]="iot-2/evt/Status1/fmt/json";
char subscribeTopic[]="iot-2/cmd/test/fmt/String";
char authMethod[]="use-token-auth";
char token[]=TOKEN;
char clientID[]="d:"ORG":"DEVICE_TYPE":"DEVICE_ID;

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




 // should match the Beta Coefficient of the thermistor

void setup() {
  Serial.begin(9600);
  analogReadResolution(10);
  pinMode(32,INPUT);
  pinMode(14,OUTPUT);
```

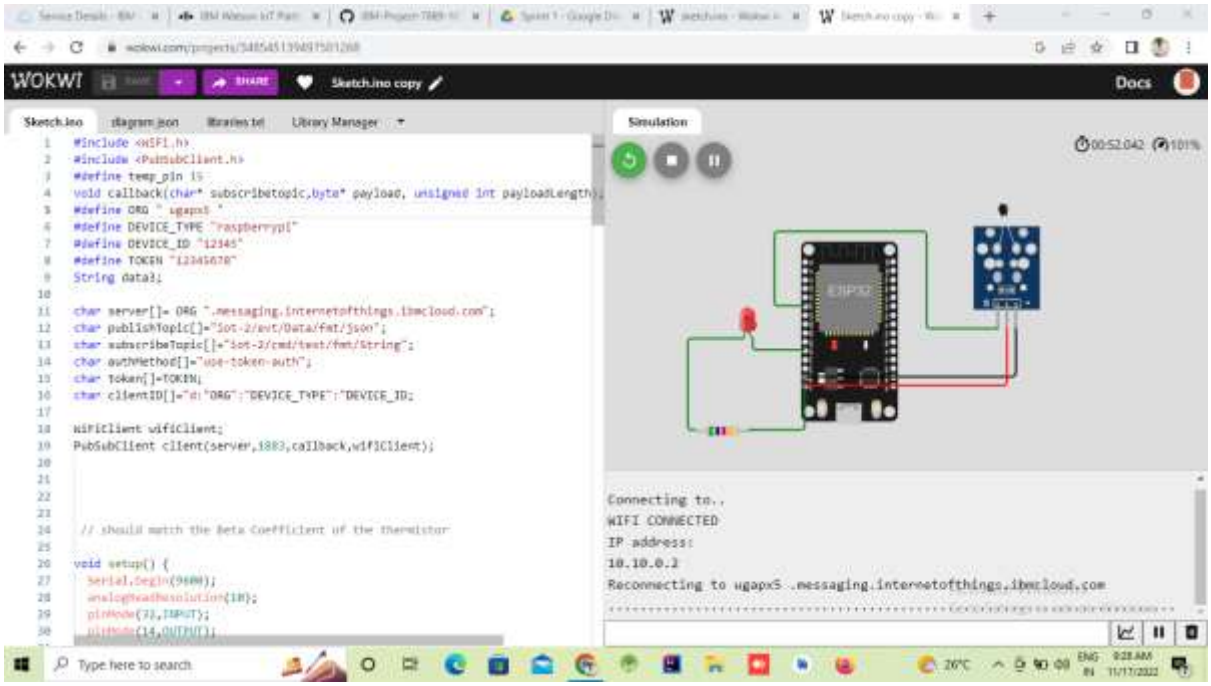
```
  wificonnect();
  mqttconnect();

}
void loop() {
  const float BETA = 3950; // should match the Beta Coefficient of the thermis
int analogValue = analogRead(A4);
float temp = 1 / (log(1 / (1023. / analogValue - 1)) / BETA + 1.0 / 298.15) -
  //float temp = 1 / (log(1 / (1023. / analogValue - 1)) / BETA + 1.0 / 298.15
  Serial.print("Temperature: ");
  Serial.print(temp);
  Serial.println(" °C");
  if(temp>=35){
    PublishData2(temp);
    digitalWrite(14, HIGH);
  }else{
    digitalWrite(14, LOW);
    PublishData1(temp);
}
 delay(1000);
  if(!client.loop()){
    mqttconnect();
  }

  //delay(2000);
}
void PublishData1(float tem){
  mqttconnect();
  String payload= "{\"temp\":";
  payload += tem;
```

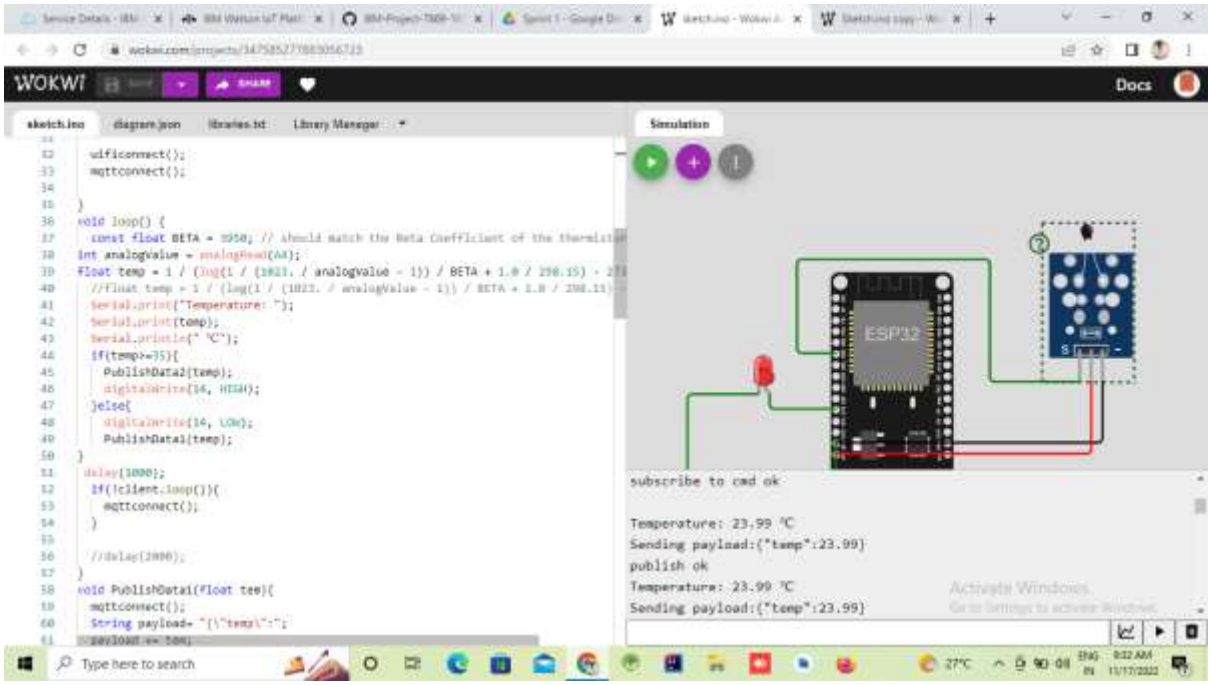
```
  payload+="}";

  Serial.print("Sending payload:");
  Serial.println(payload);

  if(client.publish(publishTopic,(char*)payload.c_str())){
    Serial.println("publish ok");
  } else{
    Serial.println("publish failed");
  }
}
void PublishData2(float tem){
  mqttconnect();
  String payload= "{\"ALERT\":";
  payload += tem;
  payload+="}";

  Serial.print("Sending payload:");
  Serial.println(payload);

  if(client.publish(publishTopic,(char*)payload.c_str())){
    Serial.println("publish ok");
  } else{
    Serial.println("publish failed");
  }
}
void mqttconnect(){
  if(!client.connected()){
    Serial.print("Reconnecting to");
    Serial.println(server);
```

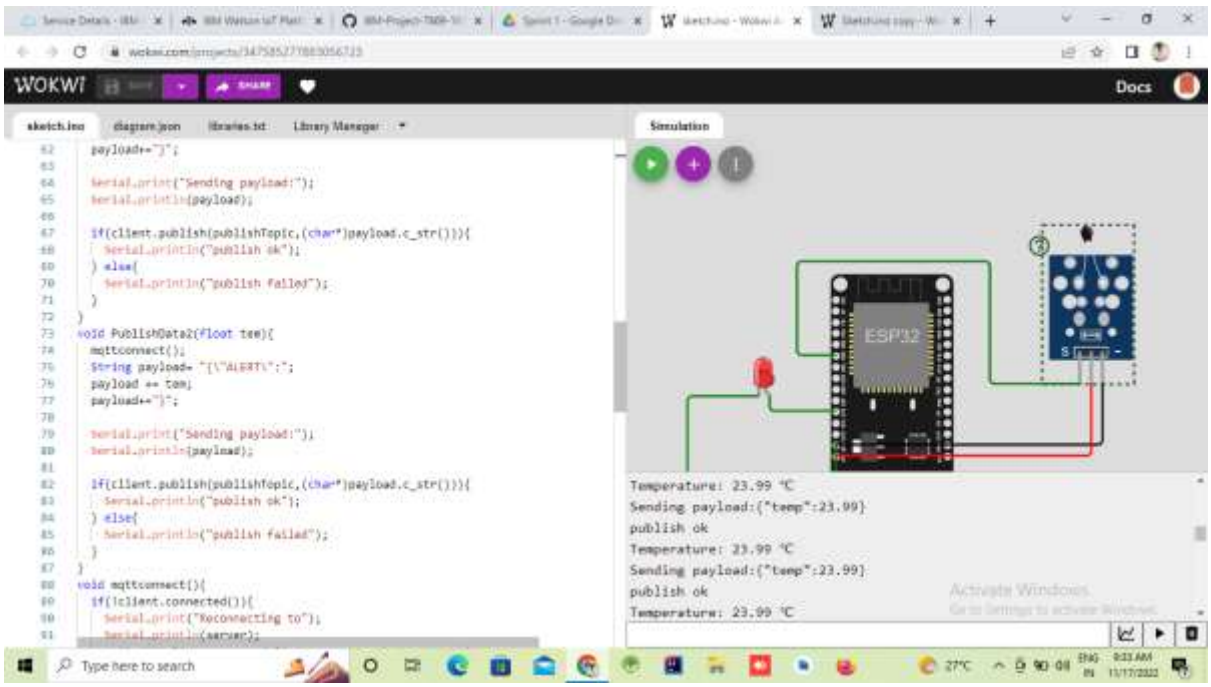
```
    while(!!!client.connect(clientID, authMethod, token)){
      Serial.print(".");
      delay(500);
    }
    initManagedDevice();
    Serial.println();
  }
}

void wificonnect(){
  Serial.println();
  Serial.print("Connecting to");

  WiFi.begin("Wokwi-GUEST","",6);
  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");
```

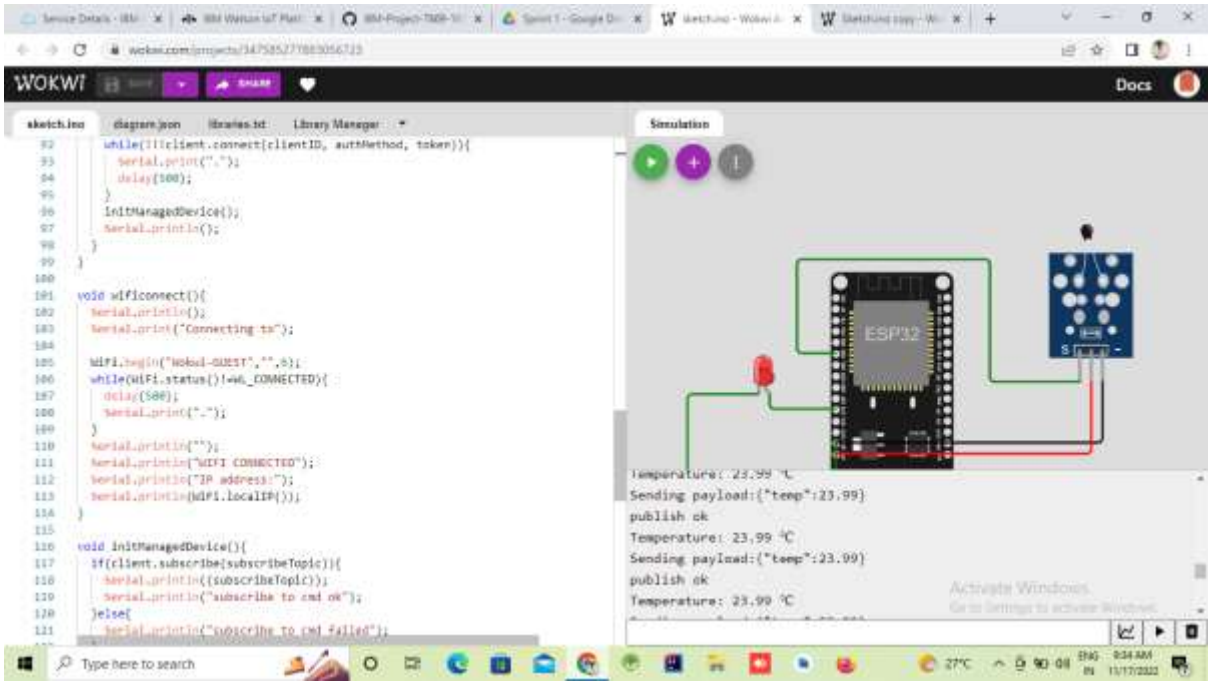
```
    }
  }

  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++){
      data3 += (char)payload[i];
    }
    Serial.println("data:"+ data3);
    if(data3=="lighton"){
      Serial.println(data3);
      digitalWrite(14,HIGH);
    }else{
      Serial.println(data3);
      digitalWrite(14,LOW);
    }
    data3="";
  }
```

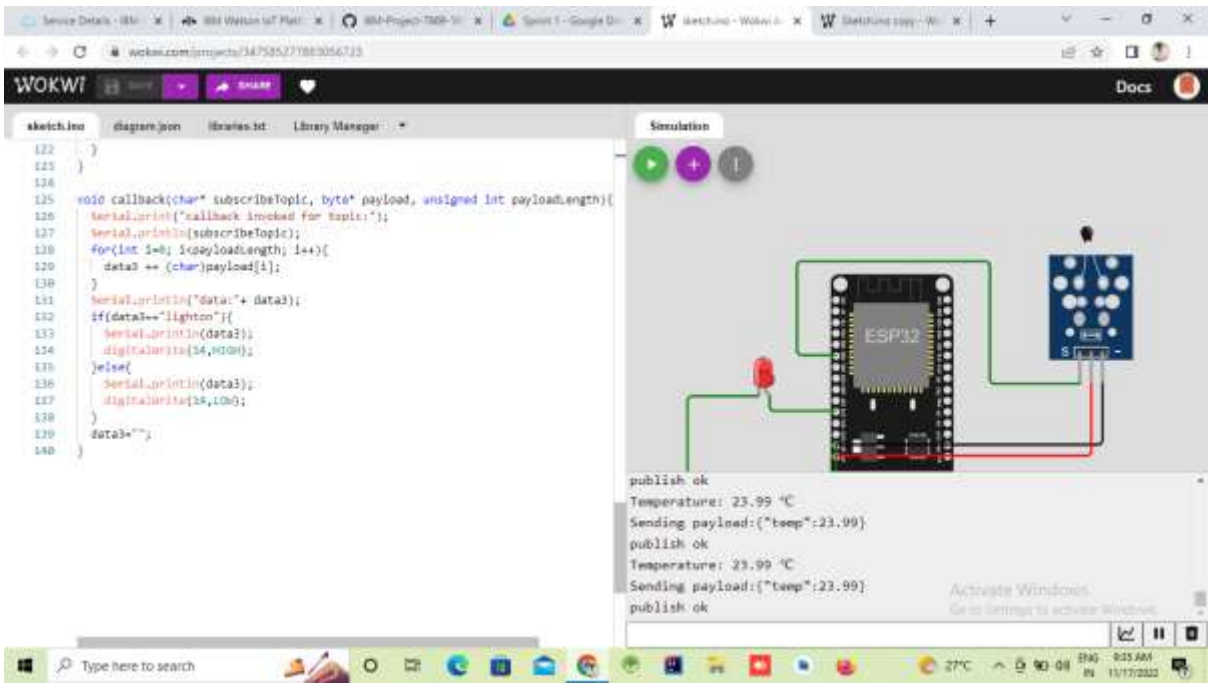
7.2. **Feature 2**




WOKWI
SHARE
Docs
sketch.ino
diagram.json
libraries.txt
Library Manager
Simulation
wificonnect();
mqttconnect();
}
void loop() {
const float BETA = 3950; // should match the Beta Coefficient of the thermistor
int analogValue = analogRead(A0);
float temp = 1 / (log(1 / (1023. / analogValue - 1)) / BETA + 1.0 / 298.15) - 273
//float temp = 1 / (log(1 / (1023. / analogValue - 1)) / BETA + 1.0 / 298.15);
Serial.print("Temperature: ");
Serial.print(temp);
Serial.println(" °C");
if(temp>=35){
PublishData2(temp);
digitalWrite(16, HIGH);
}else{
digitalWrite(16, LOW);
PublishData1(temp);
}
delay(1000);
if(!client.loop()){
mqttconnect();
}
//delay(2000);
}
void PublishData1(float tem){
mqttconnect();
String payload= "{\"temp\":";
payload += tem;
ESP32
subscribe to cmd ok
Temperature: 23.99 °C
Sending payload:{"temp":23.99}
publish ok
Temperature: 23.99 °C
Sending payload:{"temp":23.99}
Activate Windows
Type here to search
27°C


WOKWI
SHARE
Docs
sketch.ino
diagram.json
libraries.txt
Library Manager
Simulation
payload+="}";
Serial.print("Sending payload:");
Serial.println(payload);
if(client.publish(publishTopic,(char*)payload.c_str())){
Serial.println("publish ok");
} else{
Serial.println("publish failed");
}
}
void PublishData2(float tem){
mqttconnect();
String payload= "{\"ALERT\":";
payload += tem;
payload+="}";
Serial.print("Sending payload:");
Serial.println(payload);
if(client.publish(publishTopic,(char*)payload.c_str())){
Serial.println("publish ok");
} else{
Serial.println("publish failed");
}
}
void mqttconnect(){
if(!client.connected()){
Serial.print("Reconnecting to");
Serial.println(server);
ESP32
Temperature: 23.99 °C
Sending payload:{"temp":23.99}
publish ok
Temperature: 23.99 °C
Sending payload:{"temp":23.99}
publish ok
Temperature: 23.99 °C
Activate Windows
Type here to search
27°C


WOKWI
SHARE
Docs
sketch.ino
diagram.json
libraries.txt
Library Manager
Simulation
ESP32
Sending payload:{"temp":23.99}
publish ok
Temperature: 23.99 °C
Activate Windows

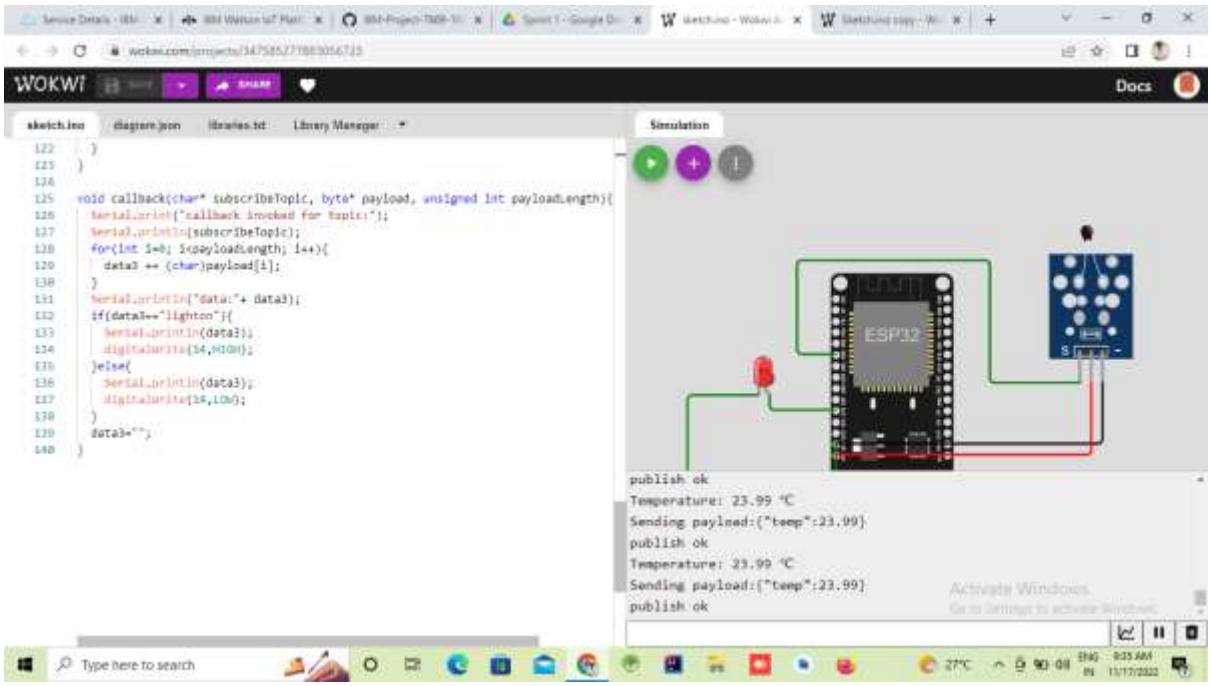

WOKWI
SHARE
Docs
sketch.ino
diagram.json
libraries.txt
Library Manager
Simulation
ESP32
publish ok
Temperature: 23.99 °C
Sending payload:{"temp":23.99}
Activate Windows

8. TESTING

8.1. **Test Cases**

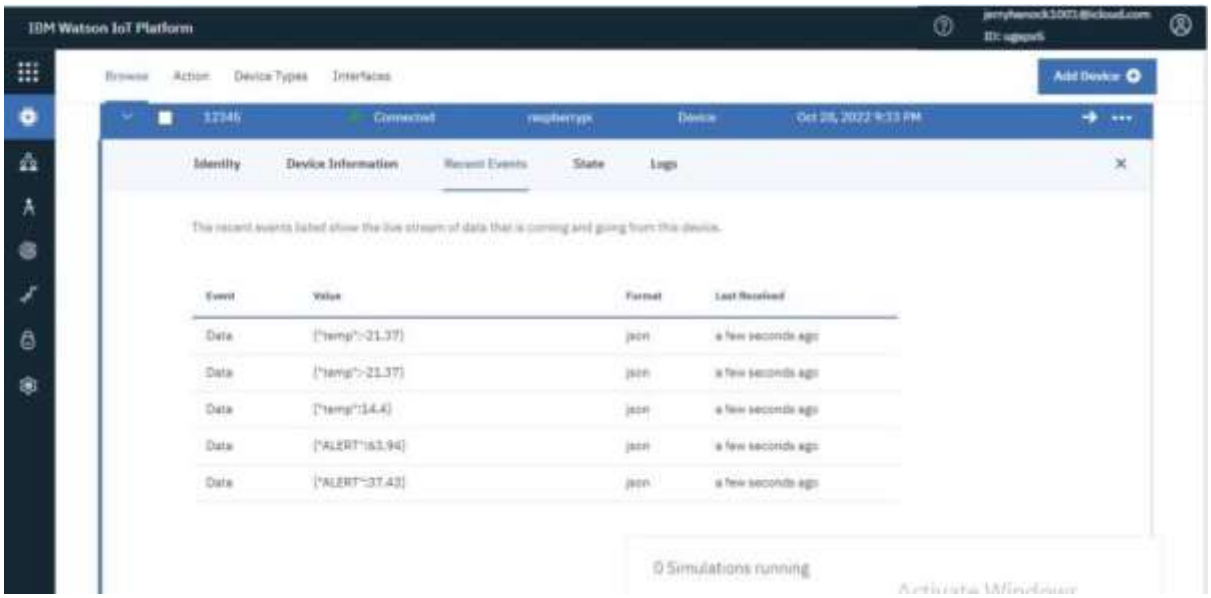
TEST CASE 1:

When the device is deployed it is expected to display the simulation with sensors and devices.



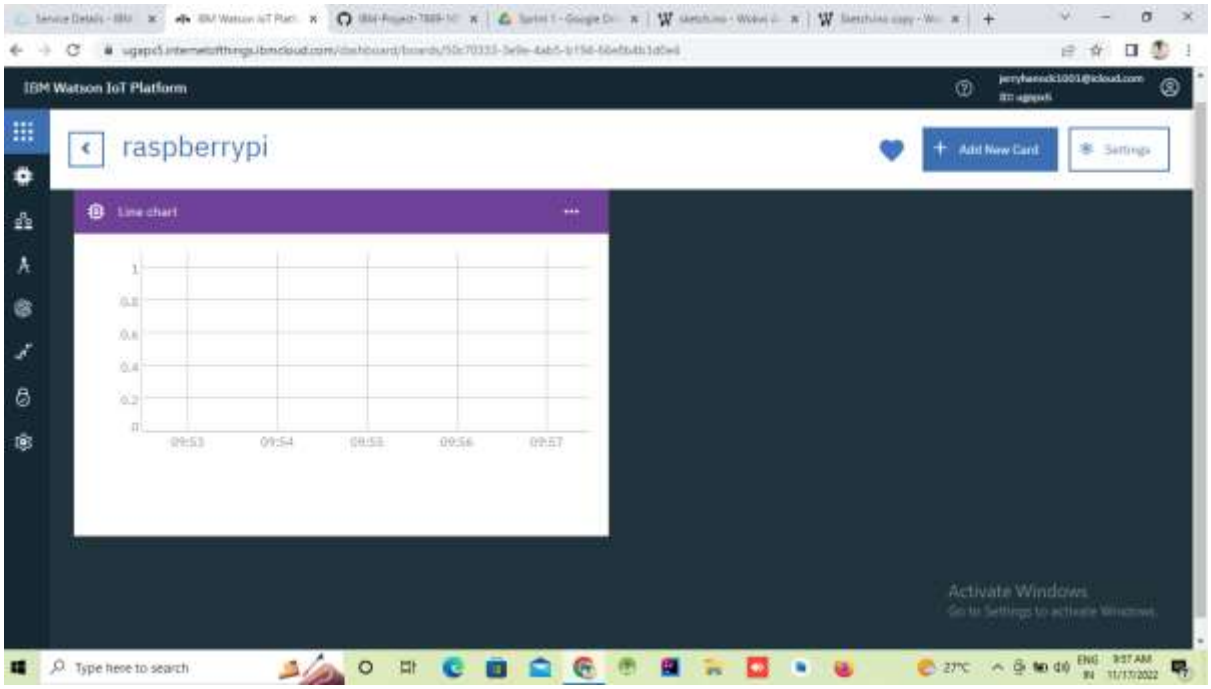
TEST CASE 2:

The simulation of the device is expected to show the output on the IBM Cloud IoT Watson Platform.



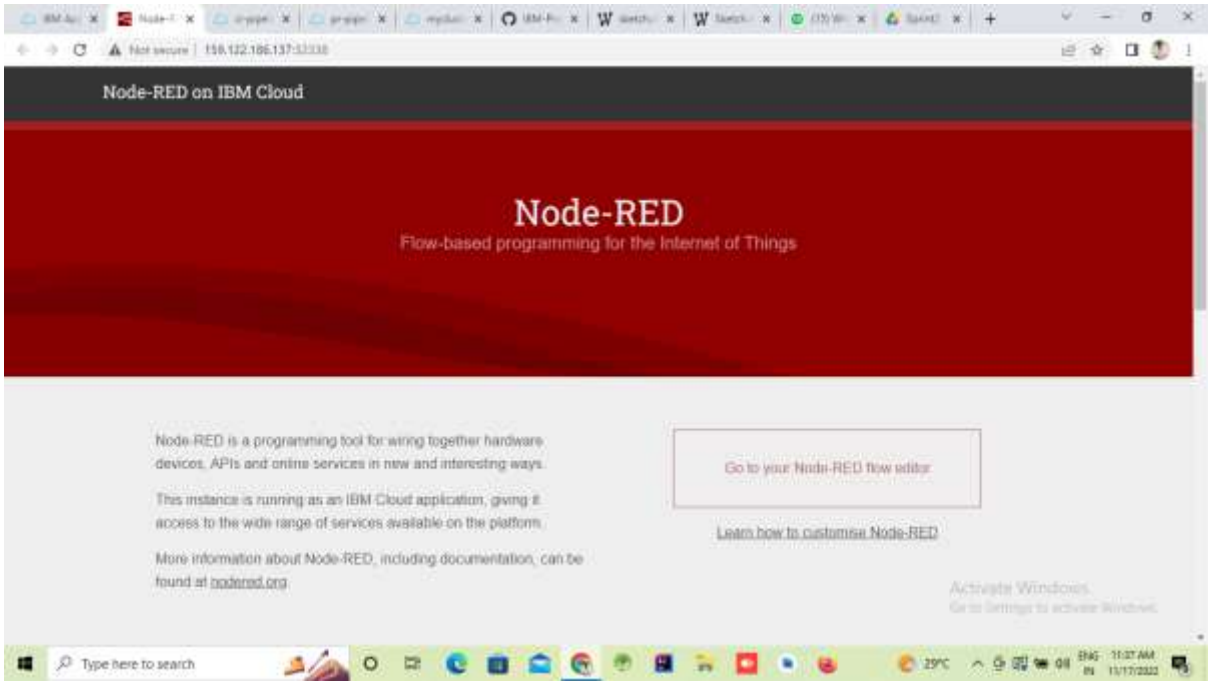
TEST CASE 3:

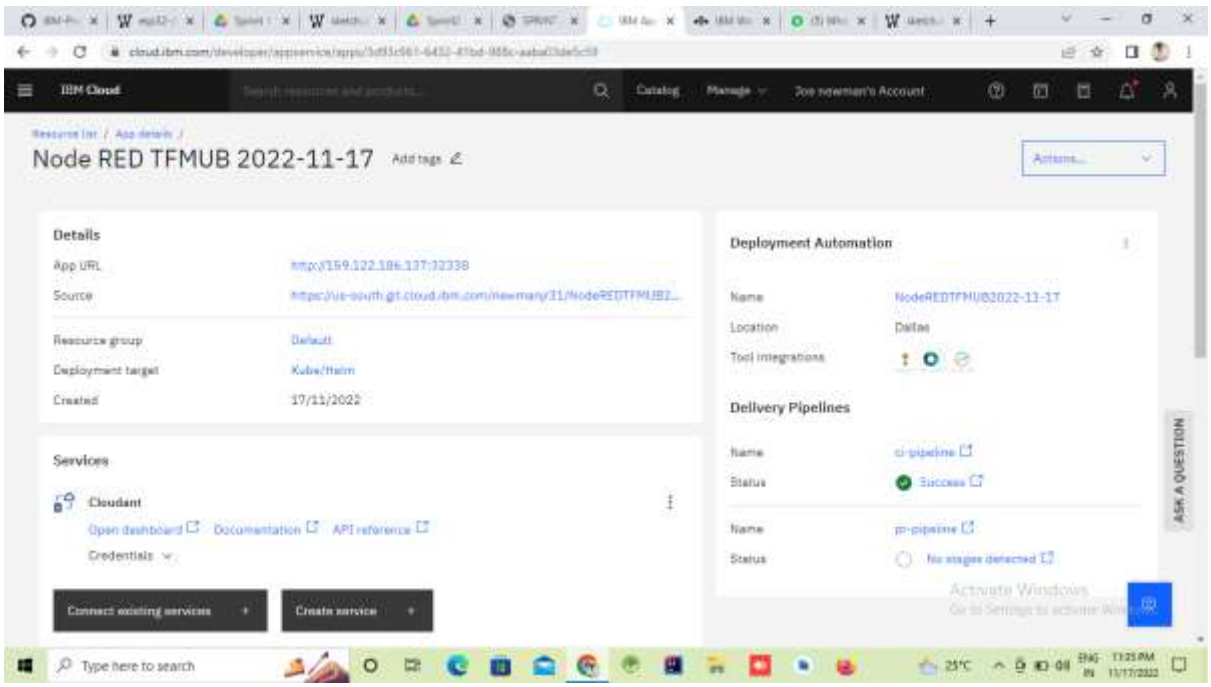
The deployment is expected to show graph on IBM Cloud IoT Watson Platform.

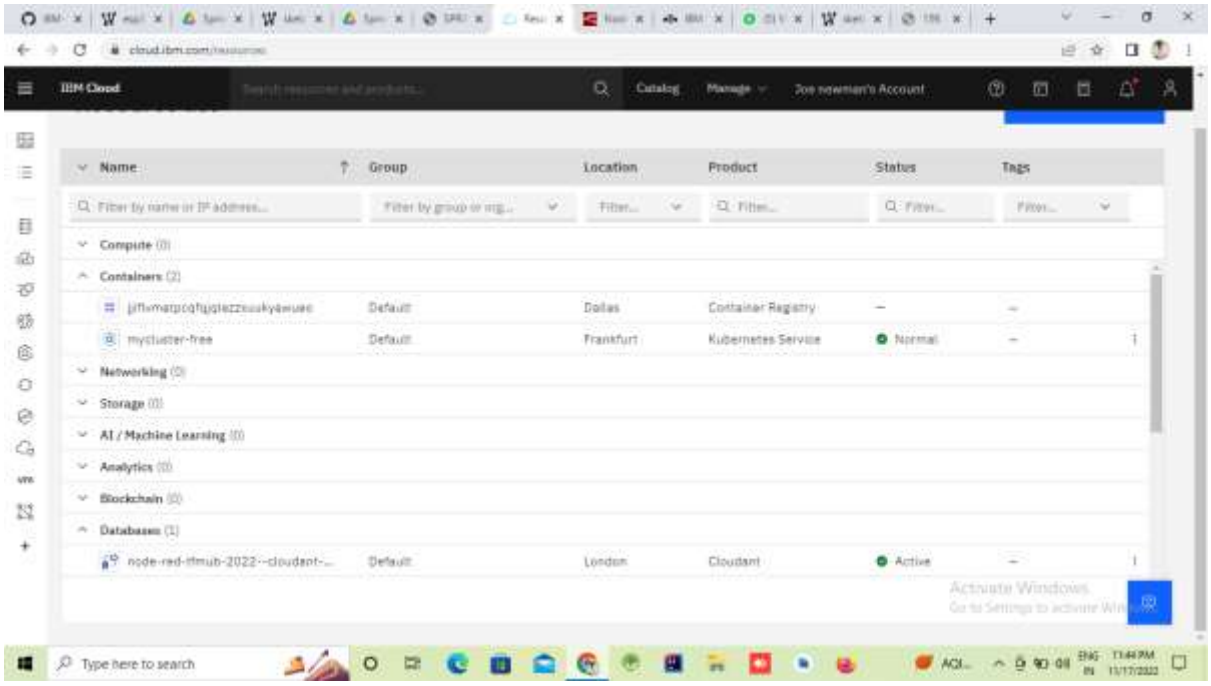


TEST CASE 4:

The expected simulation shows results connected with the Node-Red service on IBM Platform.


Node-RED on IBM Cloud
Node-RED
Flow-based programming for the Internet of Things
Node-RED is a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways.
This instance is running as an IBM Cloud application, giving it access to the wide range of services available on the platform.
More information about Node-RED, including documentation, can be found at nodered.org
Go to your Node-RED flow editor
Learn how to customise Node-RED
Activate Windows
Type here to search

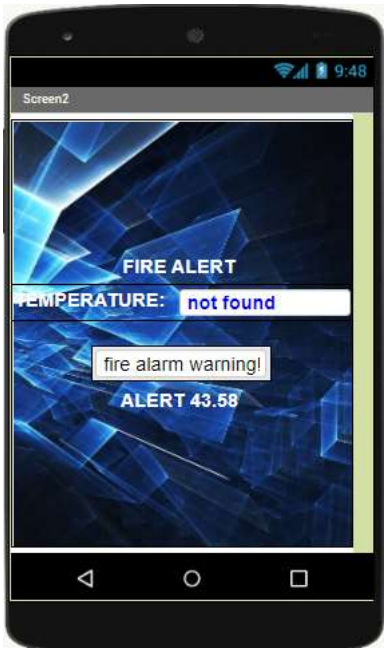

IBM Cloud
Catalog
Manage
Joe newman's Account
Node RED TFMUB 2022-11-17
Add tags
Actions...
Details
App URL
Source
Resource group
Default
Deployment target
Kube/Helm
Created
17/11/2022
Deployment Automation
Name
NodeREDTFMUB2022-11-17
Location
Dallas
Tool integrations
Delivery Pipelines
Name
ci-pipeline
Status
Success
Name
cp-pipeline
Status
No stages detected
Services
Cloudant
Open dashboard
Documentation
API reference
Credentials
Connect existing services
Create service
ASK A QUESTION
Activate Windows
Type here to search

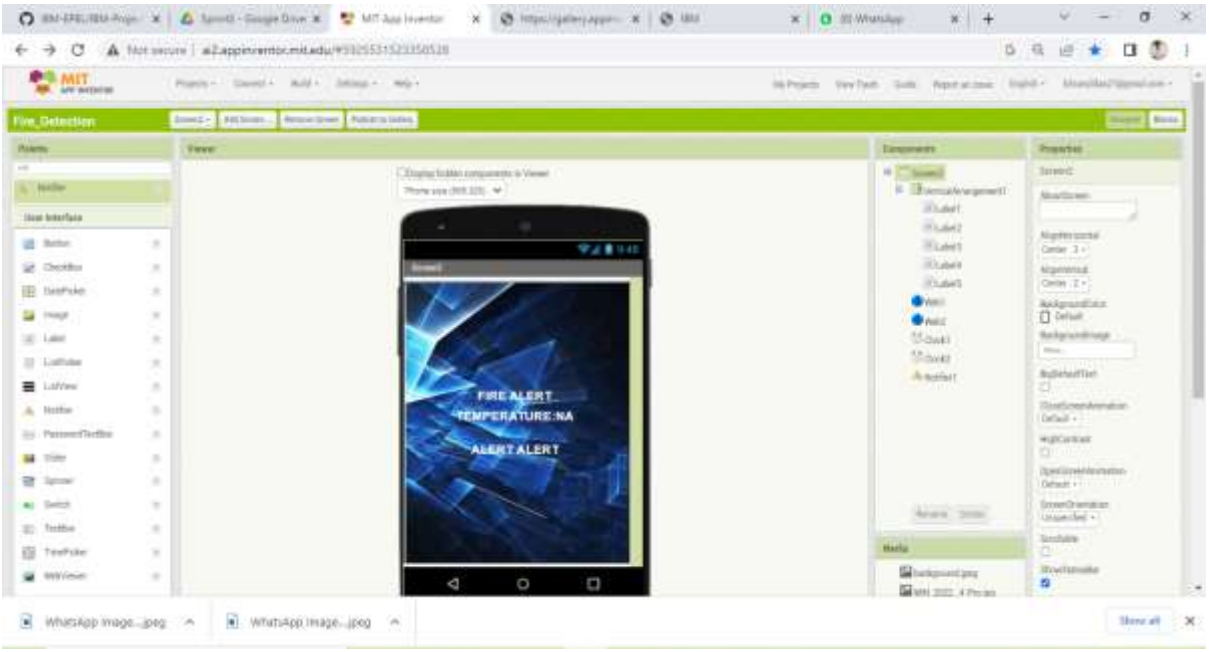


8.2. **User Acceptance Testing**

Case 1:

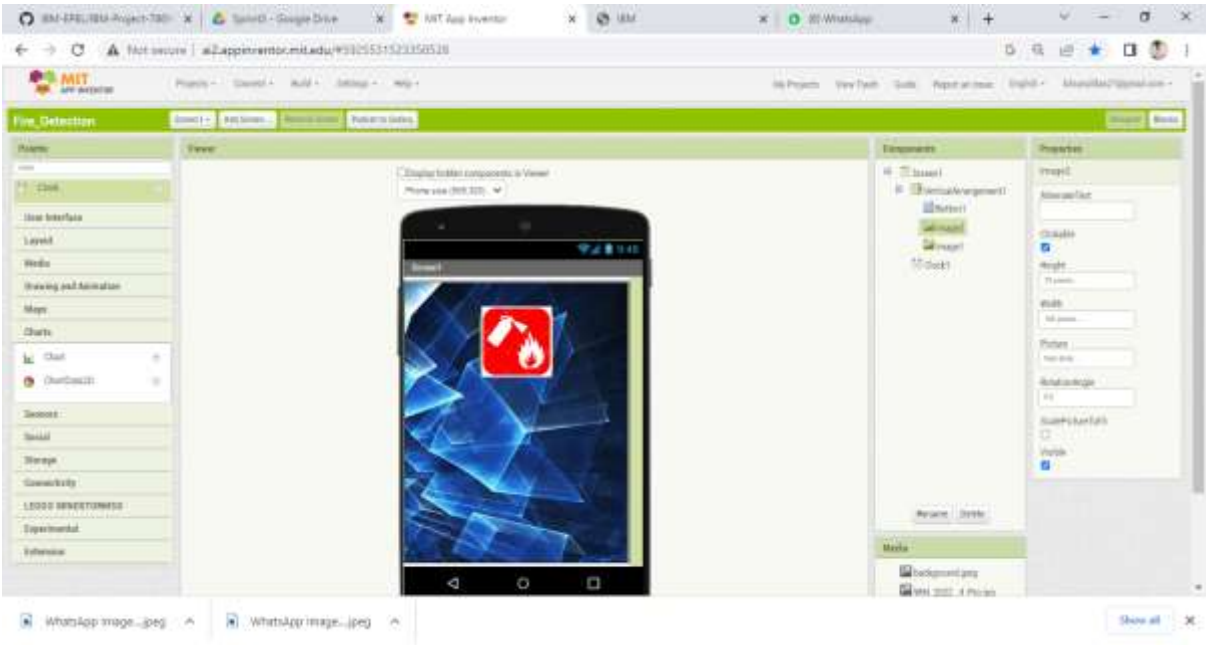
When the device detects the temperature and if it is stable, it will display the result as "NOT FOUND".

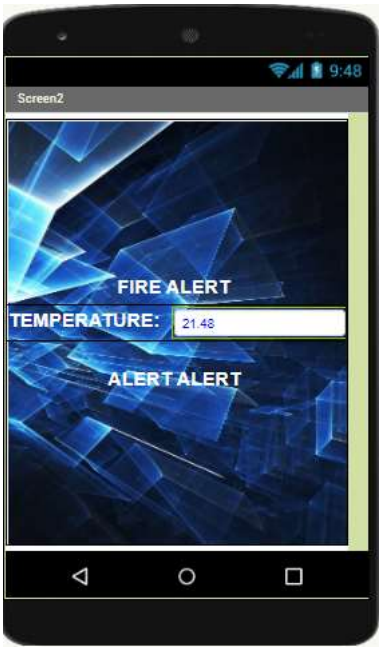




Case 2:

When the device detects the temperature and if it is high, it will display the result as "ALERT!".

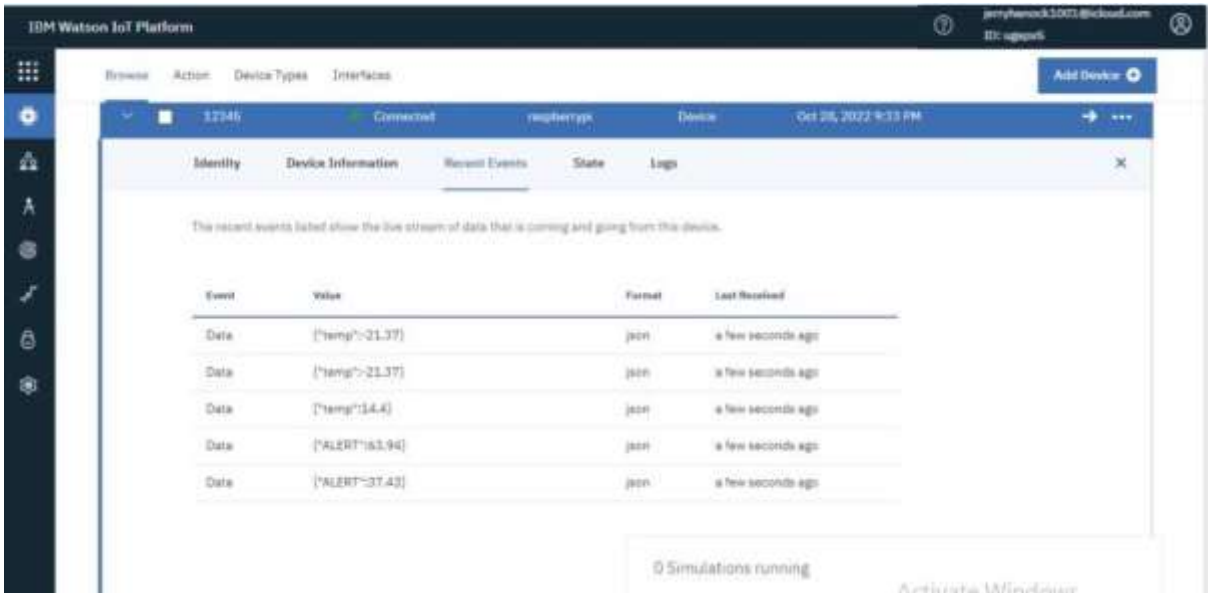




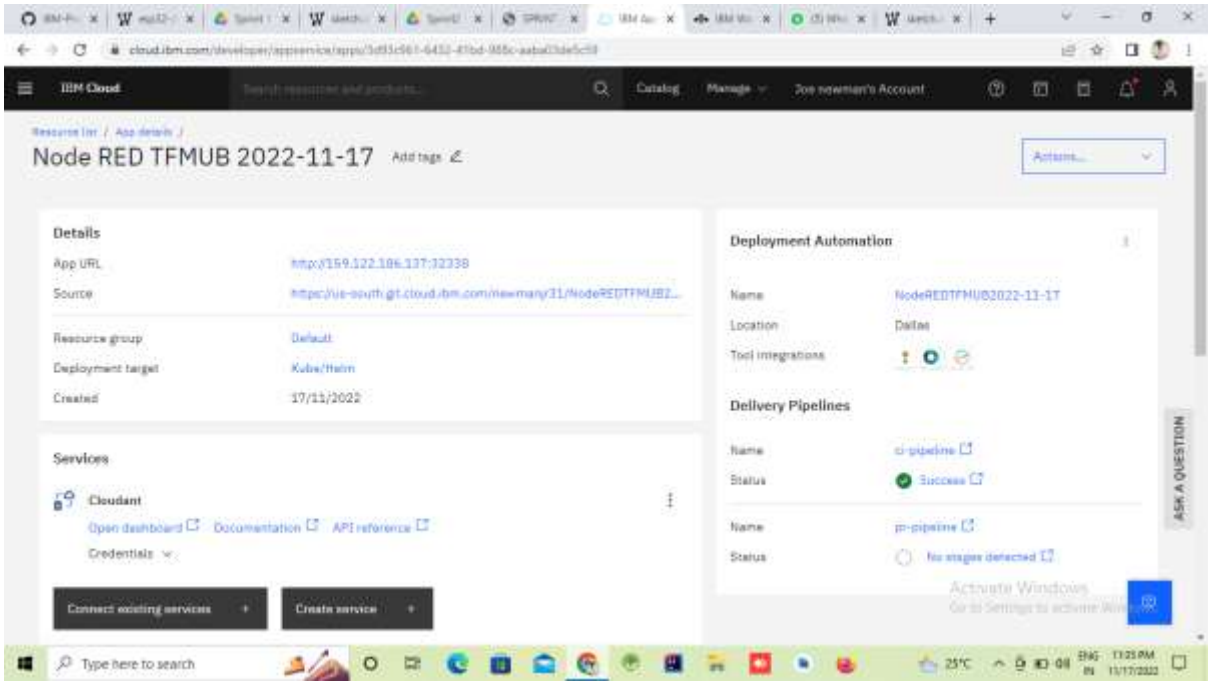
9. RESULTS

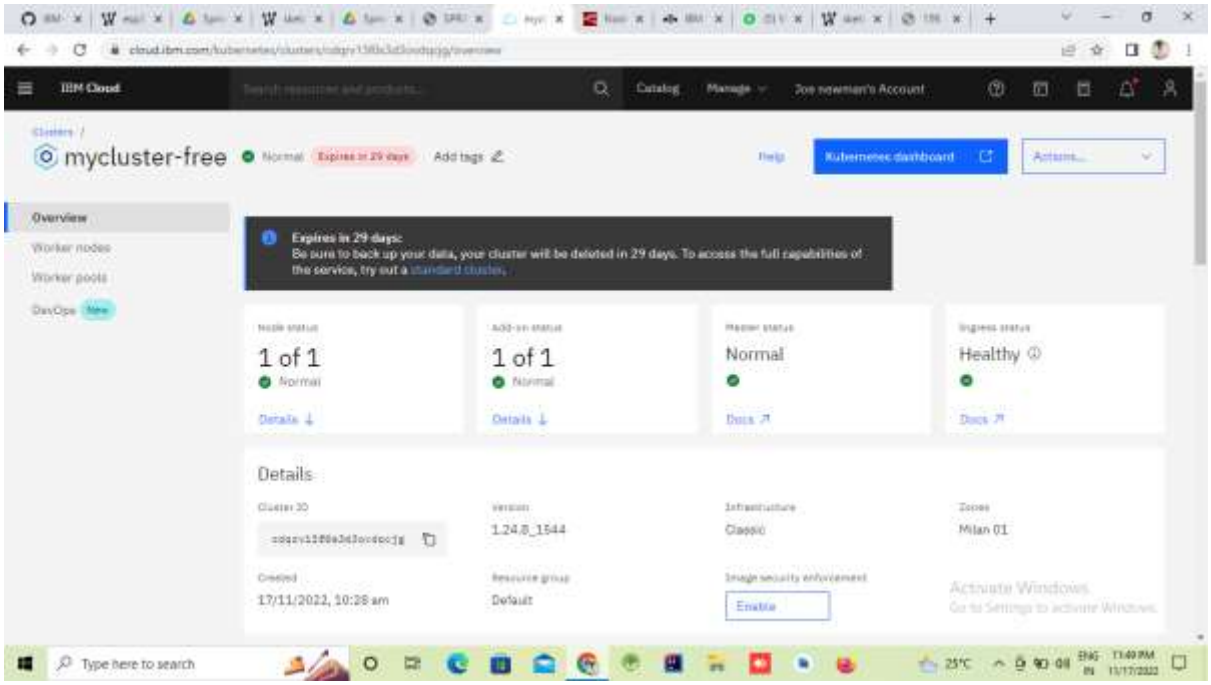
9.1. Performance Metrics

IBM Cloud IoT Watson Platform output



IBM Cloud Node-Red service output





MIT Inventer application output


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

10. ADVANTAGES & DISADVANTAGES

Advantages:

As far as fire alarm installers go, a wireless system is ideal because they are much easier to install. A wireless system essentially involves mounting the devices to the appropriate locations around a building or room, setting up the actual system and syncing it to WiFi. Compare this to a wired system, which requires fire alarm installers to connect the system to power supplies and ensure cables are connected properly.

Another great advantage of a wireless fire alarm system is it operates off of a battery. This frees up a wall outlet and you can feel safe knowing the system will still work in the event of a power outage. And adding a second or subsequent wireless device is easy if you add on to your home or office.

Disadvantages:

The one thing most fire alarm system inspectors caution against with wireless systems is having to replace the battery. 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.

A couple other disadvantages fire alarm system inspectors point out is wireless systems have limited range and don't have centralized monitoring. 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 prevent from the fire accidents in the industries and other hazardous places. The fire can be monitored in various places such as, industries, offices and homes. Prevention is better than cure, as the way by using the

IOT based fire alarm system is suitable for detect the fire accident and save the human life.

12. FUTURE SCOPE

- Creating the application user friendly.
- Increasing the accuracy rate.
- Installing fire proof safety devices.

13. APPENDIX

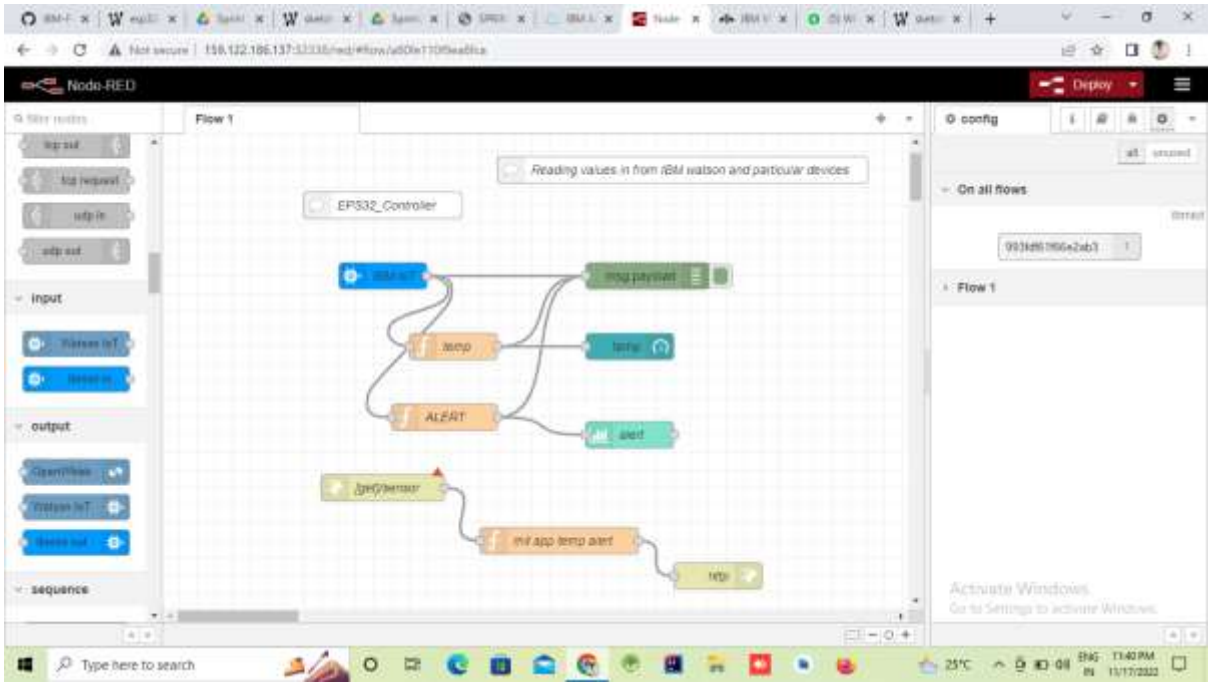
Source code

```
#include <WiFi.h>
#include <PubSubClient.h>
#define temp_pin 15
void callback(char* subscribetopic,byte* payload, unsigned int payloadLength);
#define ORG " ugapx5 "
#define DEVICE_TYPE "raspberrypi"
#define DEVICE_ID "12345"
#define TOKEN "12345678"
String data3;

char server[]= ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[]="iot-2/evt/Status1/fmt/json";
char subscribeTopic[]="iot-2/cmd/test/fmt/String";
char authMethod[]="use-token-auth";
char token[]=TOKEN;
char clientID[]="d:"ORG":"DEVICE_TYPE":"DEVICE_ID;

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



 // should match the Beta Coefficient of the thermistor

void setup() {
  Serial.begin(9600);
  analogReadResolution(10);
  pinMode(32,INPUT);
  pinMode(14,OUTPUT);

  wificonnect();
  mqttconnect();

}
void loop() {
```

```
  const float BETA = 3950; // should match the Beta Coefficient of the thermistor
int analogValue = analogRead(A4);
float temp = 1 / (log(1 / (1023. / analogValue - 1)) / BETA + 1.0 / 298.15) - 273.15;
  //float temp = 1 / (log(1 / (1023. / analogValue - 1)) / BETA + 1.0 / 298.15) - 273.15;
  Serial.print("Temperature: ");
  Serial.print(temp);
  Serial.println(" °C");
  if(temp>=35){
    PublishData2(temp);
    digitalWrite(14, HIGH);
  }else{
    digitalWrite(14, LOW);
    PublishData1(temp);
}
 delay(1000);
  if(!client.loop()){
    mqttconnect();
  }

  //delay(2000);
}
void PublishData1(float tem){
  mqttconnect();
  String payload= "{\"temp\":";
  payload += tem;
  payload+="}";

  Serial.print("Sending payload:");
  Serial.println(payload);

  if(client.publish(publishTopic,(char*)payload.c_str())){
    Serial.println("publish ok");
  } else{
    Serial.println("publish failed");
  }
}
void PublishData2(float tem){
  mqttconnect();
  String payload= "{\"ALERT\":";
  payload += tem;
  payload+="}";

  Serial.print("Sending payload:");
  Serial.println(payload);

  if(client.publish(publishTopic,(char*)payload.c_str())){
    Serial.println("publish ok");
  } else{
    Serial.println("publish failed");
  }
}
void mqttconnect(){
  if(!client.connected()){
```

```
    Serial.print("Reconnecting to");
    Serial.println(server);
    while(!!!client.connect(clientID, authMethod, token)){
      Serial.print(".");
      delay(500);
    }
    initManagedDevice();
    Serial.println();
  }
}

void wificonnect(){
  Serial.println();
  Serial.print("Connecting to");

  WiFi.begin("Wokwi-GUEST","",6);
  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++){
    data3 += (char)payload[i];
  }
  Serial.println("data:"+ data3);
  if(data3=="lighton"){
    Serial.println(data3);
    digitalWrite(14,HIGH);
  }else{
    Serial.println(data3);
    digitalWrite(14,LOW);
  }
  data3="";
}
```

Node-Red Web service



GITHUB DEM0 LINK: https://youtu.be/g1A5sFmhPy8