A PROJECT REPORT ON

Hazardous Area Monitoring for Industrial Plant powered by IoT

Domain : Internet of Things.

Team ID : PNT2022TMID41855

College Name : Salem College of Engineering and Technology

Manikandan.B (622419104028)

Sabaresan.S (622419104036)

Nizraahemad.J (622419104304)

Sarathkumar.E (622419104305)

TABLE OF CONTENTS

1. INTRODUCTION

a. Project Overview

- ✔ The Industrial Internet of things or IoT has gained recognition due to the advancement it has made in communication technology. Industrial IoT is an application of IoT that enables control of industries over the Internet using smart devices and sensors. The two main entity which ensures effectiveness in any field is monitoring and control.
- ✔ It is the Gain knowledge of Watson IoT Platform.
- ✔ Connecting IoT devices to the Watson IoT platform and exchanging the sensor data.
- ✔ Gain knowledge on Cloudant DB.Creating a Web Application through which the user interacts with the device.

b. Purpose

Through this, we can monitor the temperature parameters of the hazardous areas in industrial plants.The area is integrated with smart beacon devices which will be broadcasting the temperature of that particular area.Every person working in those areas will be given smart wearable devices which will be acting as beacon scanners.Whenever the person goes near the beacon scanners, he can view the temperature on his wearable device and if the temperature is high, he will receive the alerts to the mobile through SMS using API.Through this wearable device, the data is sent to the cloud and through the dashboard, the admins of that particular plant can view the data and take necessary precautions if required.

2. LITERATURE SURVEY

a. Existing problem

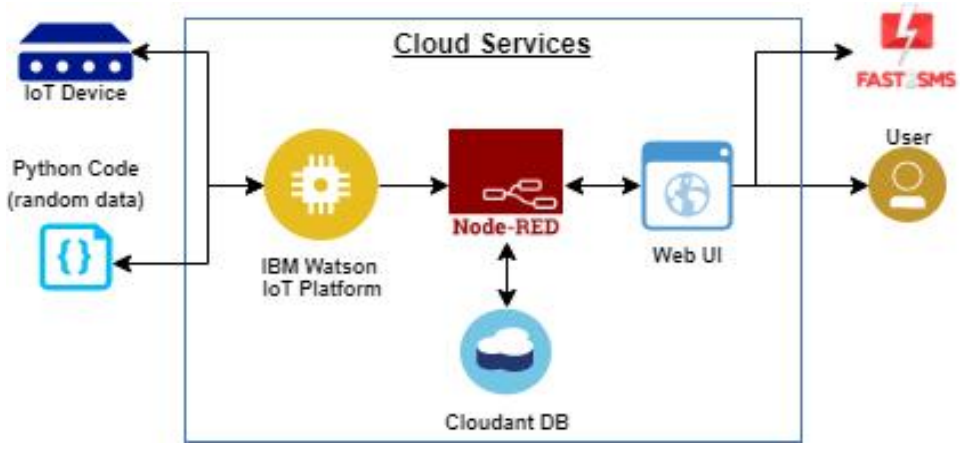
Internet of Things reception in conventional and slow changing modern plants, for example, power, water, oil-gas and substance has demonstrated to be helpful in giving business esteem by changing how information is used in direction and perception. Regular modern IoT use cases include securing information from sensor gadgets in plant and imparting something very similar to web for neighbourhoods or remote observing and control. The sensor information procurement in a modern plant hence becomes central as similar obtained information is utilized for drawing out the hidden information on framework. IoT regularly requires a nearby, low power remote correspondence to secure information from sensor gadgets and neighbourhood's entryway that is associated with web for neighbourhoods or remote checking and control. This paper portrays how Bluetooth low energy (BLE) innovation can be utilized to associate sensor hubs to Web based administrations and applications involving passage in a modern plant. It likewise examines the exhibition of BLE innovation as a neighbourhood's correspondence for sensor gadget observing

b. References

1. Anitha Varghese, Rahul N. Gore, Himashri Kour, Mihit Gandhi, Deepaknath Tandur.
2. Apeksha Rane, Bhushan Vidhale, Priyanka Hemant Kale, Ganesh Khekare.
3. Mehedi Hasan, Abdul Hasib Siddique, Farida Habib Semantha, Fahad Faisal, Mohammed Rezwanul Islam, Mosfiqun Nahid Hassan.
4. S. Drakshayani, Y. LaksmiManjusha, P. Ramadevi,V. Madhuravani, K. Rama Sugun.
5. K. Krishna Kishore, M.H. Sai Kumar, M. B. S. Murthy.

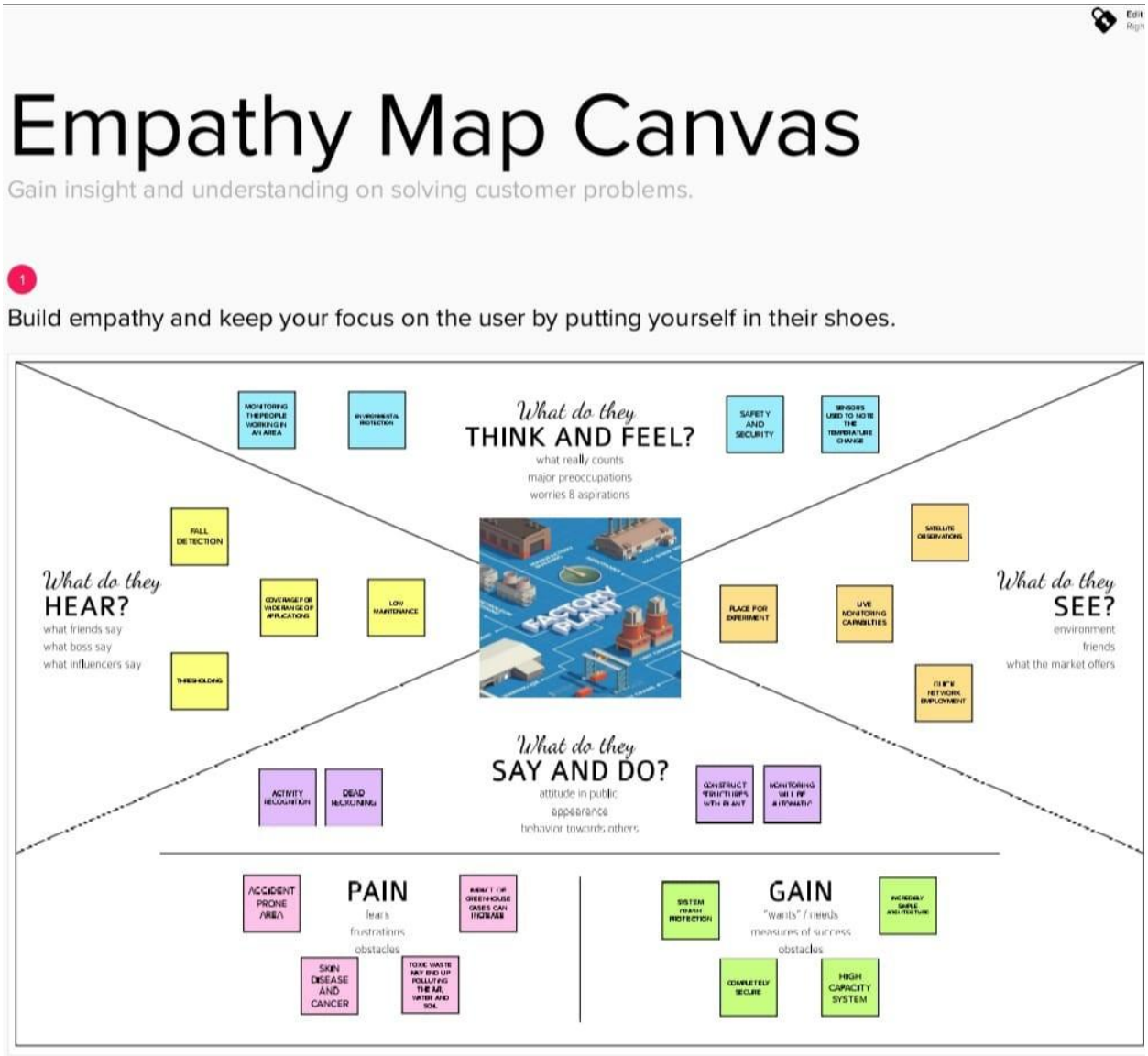
c. Problem Statement Definition

As we can see in today's world only some devices like PC's and mobiles are connected to internet. Now-a-days world is fully overtaken by the internet and internet of things. Internet is use for basic need of all human beings. The Internet of Things (IOT) is the network of physical objects. It simply means to monitor a physical device or machine or it is inter-networking of physical devices which is embedded with electronics, sensors, software and network connectivity to enable it to achieve greater value and services by exchanging data with the manufacturer IOT permits objects to be sensed or controlled remotely across the network infrastructure. The result improves accuracy, economic benefits, efficiency and reduces intervention of human. In this paper we are going to deal with basic and important concepts of IOT and its scope in upcoming future. This paper studies the need of IOT in day-to-day life for different applications and gives brief information about IOT. IOT contributes significantly toward revolutionary farming methods. So, we are trying to demonstrate IOT in Automatic watering system. Automatic watering system monitors and maintain the approximate moisture content in industry.

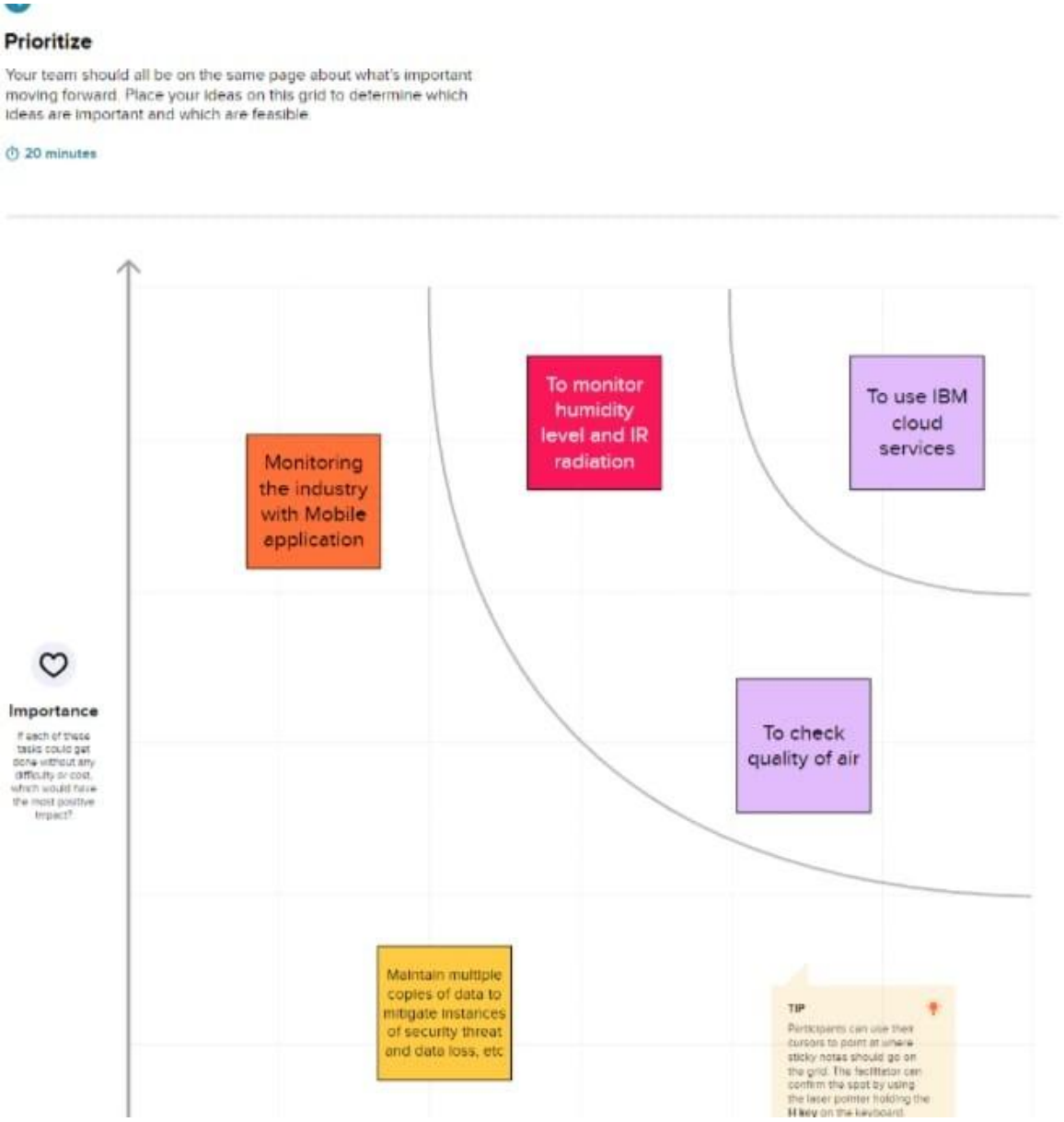


3. IDEATION & PROPOSEDSOLUTION

a. Empathy Map Canvas



b. Ideation & Brainstorming



c. Proposed Solution

1. Problem Statement (Problemto be solved)

To monitor and alert the industrial workers the riskof toxic or hazardous gases within the area of an industry, ensuring the safety of the workers.Difficulty in continuous manual monitoring of temperature and communication in hazardous areas.

2. Idea / Solutiondescription

The hazardous area is integrated with smart temperature beacons which will be sensing and broadcasting the temperature of that particular area. Every person working in those areas will be given smart wearable devices which will be acting as beacon scanners. Whenever the person goes near the beacons, he can view the temperature on his wearable device and if the temperature is high, he will receive the alertsto the mobile through SMS using API.

3. Novelty / Uniqueness

- Makes it easier to know the temperature (or) any hazardous gases present in the area withoutthe worker havingto constantly doing manual checks.

- Smart wearable devices are used.

- Alerts via SMS to mobiles of the workers when high temperature is detected.

- Alerts on both the wearable device and mobile application occurs simultaneously to prevent the worker from entering into hazardous

areas

4. **Social Impact / Customer Satisfaction**

- Ensures safety.
- Comfortable & User-friendly.
- Simple and reliable.
- More focus on work without any fear.

5. **Business Model (RevenueModel)**

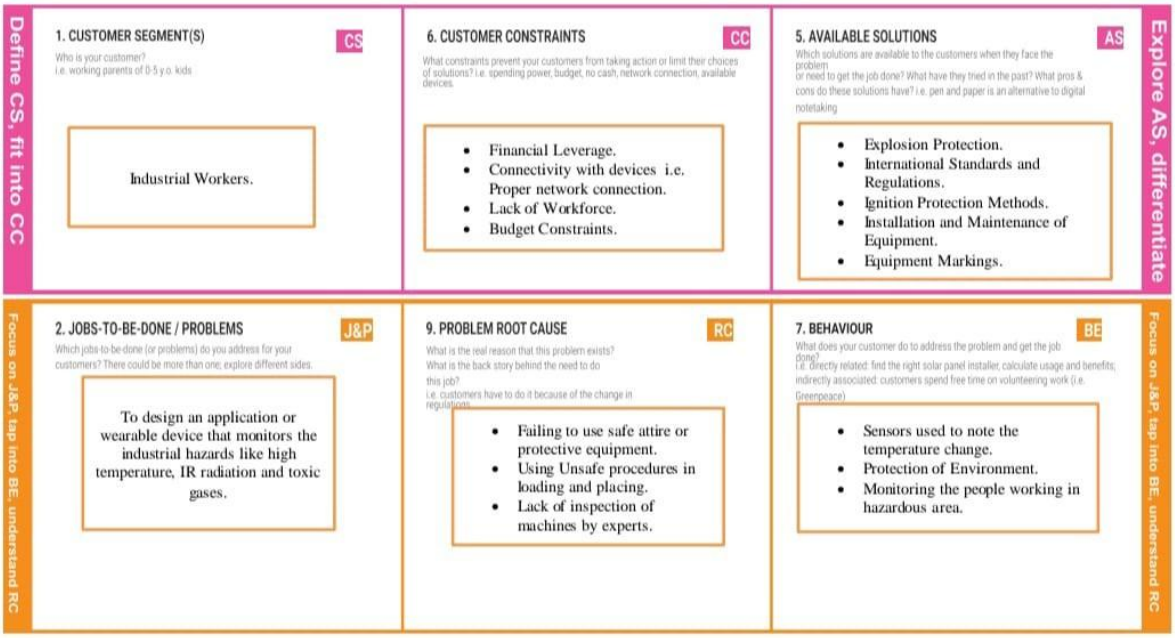
- Through our mobile application the revenue can be made in the form of pop-up advertisements, overlay ads from third party services.
- Wearable devices can be priced and sold by the industryto the workers. Can be implemented in different hazardousareas.

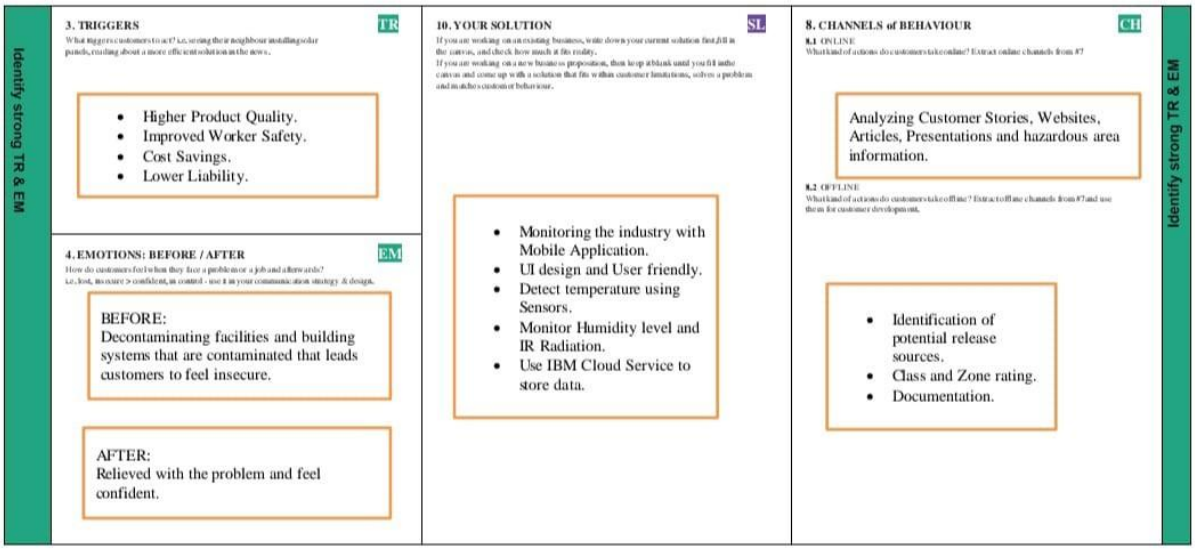
6. **Scalability of the Solution**

- It ensures the safety of each and every worker working in harmful gases and hightemperature environment.
- By increasing the numberof devices, thiscan be implemented in a commercial level.

In future, other elements like radiationand gases can also be monitored.

d. Problem Solution fit





4. REQUIREMENT ANALYSIS

a. Functional requirement

FR No.	Functional Requirement (Epic)	Sub Requirement (Story/ Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation viaOTP
FR-3	User Details	• Arduino Controller • Temperature Sensor • Gas Sensor • Humidity Sensor • Internet • Server • Cloud Platform
FR-4	User Requirements	• The two main entitywhich ensures effectivenessinany fieldis monitoring andcontrol. • To design a low-cost, low-power Wi-Fi based industrial monitoring system that controls and monitors the remotemanufacturing plants and industries using a web application. • There are certain sensors used to keep a check on temperature , humidity, gas leakage , pressure etc. in the workenvironment to ensurethe workers safety.

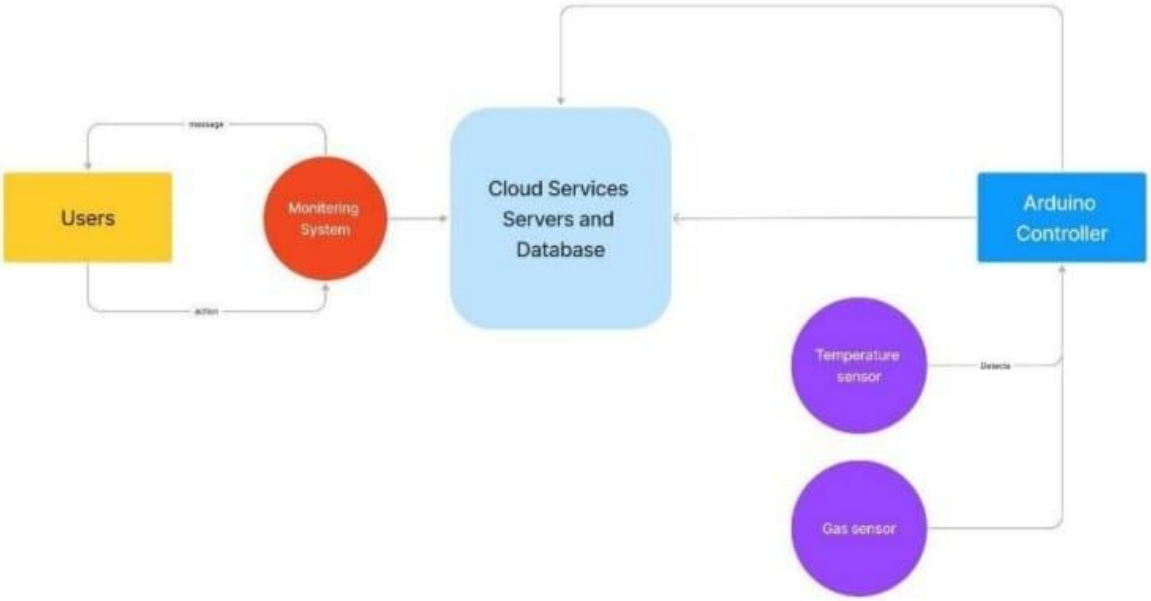
b. Non-Functional requirements

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Incredibly simple Architecture : The system requires just3 core hardware componentsto run, and runs with minimal space and resource requirements. Cost Effective: Cheaper to implement than other options and it cover alarge area and user-base with low maintenance costs.
NFR-2	**Security**	When developing industrial systems that will be situated in hazardous areas, engineers need to include functional safety aspects into their thought process. Thiscannot be leftuntil thatlast minute; it should be a key consideration throughout the development cycle.
NFR-3	**Reliability**	The System uses encrypted notifications for sensitive notifications and communication with cloud service is also based on

		unique API keys. The platform also offers increased fleet visibility remotely, allowing engineers to monitor assetsfrom anywhere in the world.
NFR-4	**Performance**	It is configured in such a way that it recovers and reconnects itself after a crash/power cut, and can resume working immediately
NFR-5	**Availability**	It can serve alerts than a million unique users and morenew users can be added by just adding their API key tothe system.
NFR-6	**Scalability**	Notification parameters and user access control can beadjusted to suit requirements. Periodic data collection gives a morethorough overview of asset condition, removing the need for maintenance teams tocarry out observations and collect data manually.

5. PROJECT DESIGN

a. Data Flow Diagrams



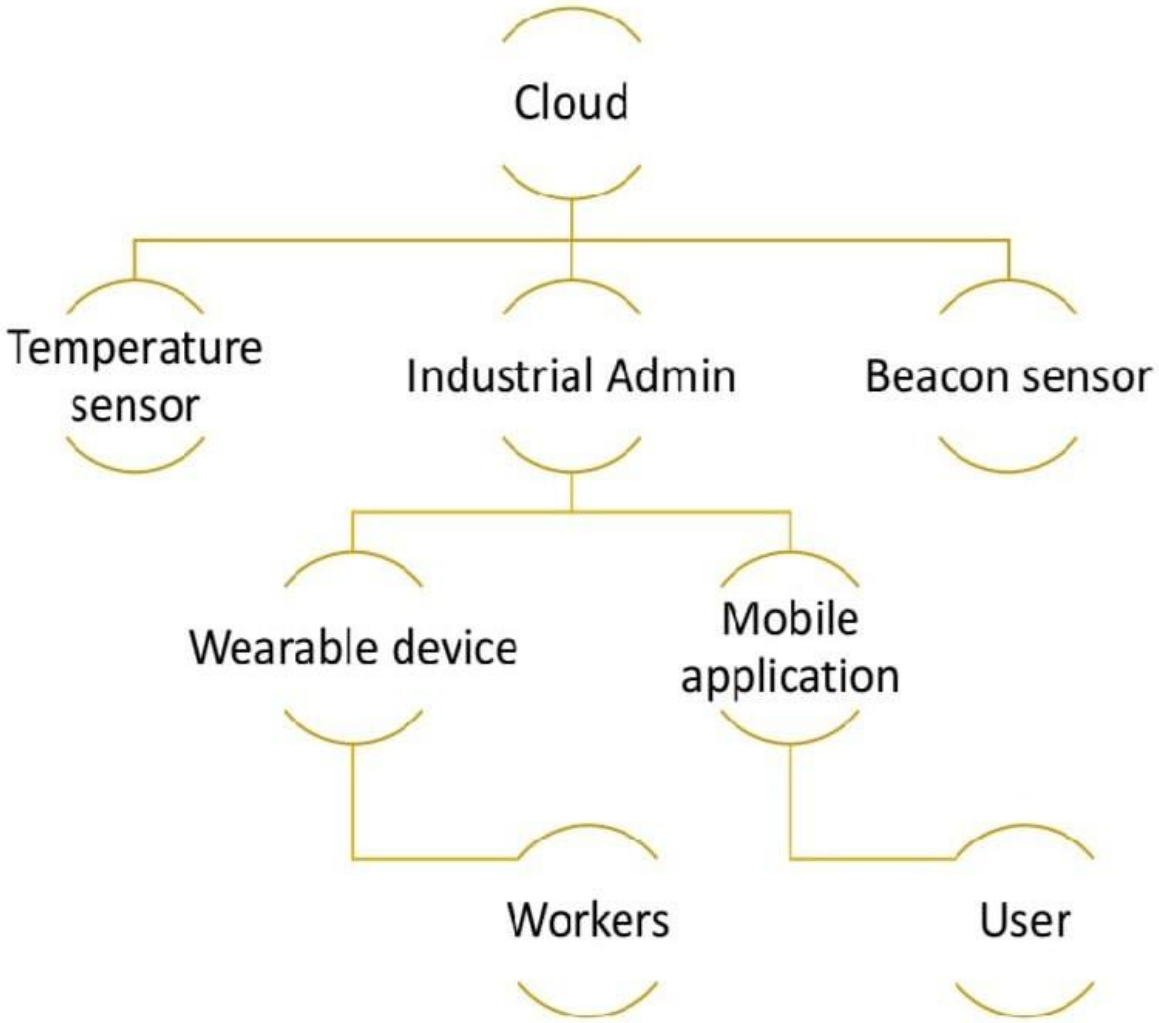
b. Solution & Technical Architecture

Solution Architecture

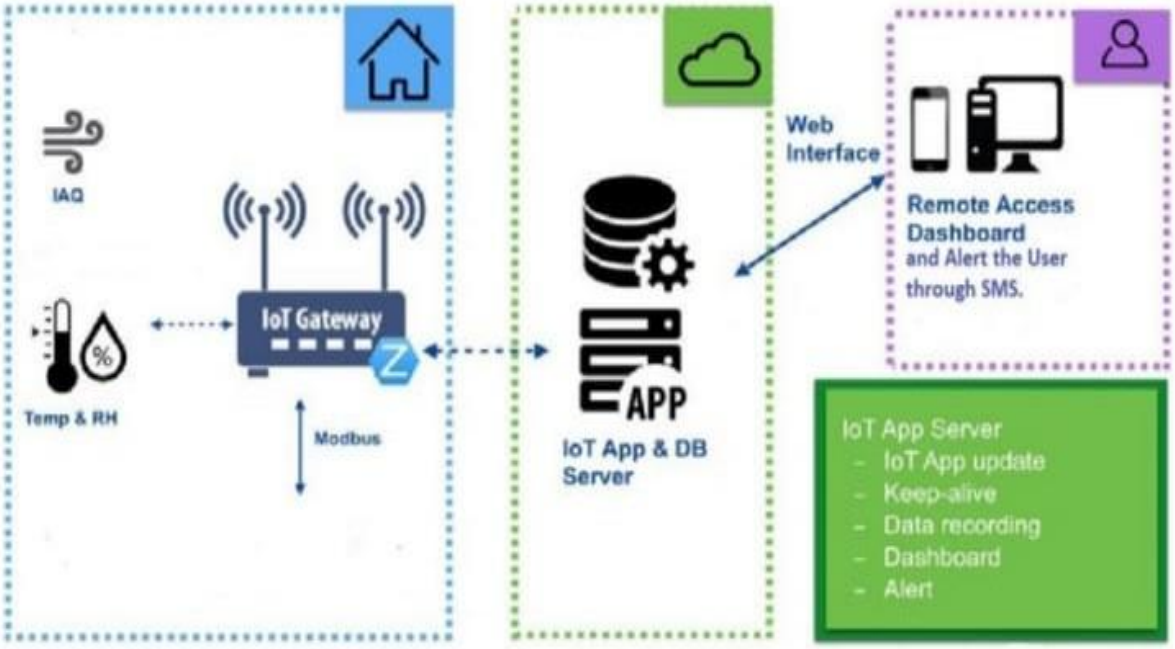
In Some industrials plants, there are some area which are to be monitored time to time . Sometimes the condition may becomecritical which may lead to loss of propertyand human loss.

To monitor the conditions we can integrate the smart devices in theareas which are needed to be monitored. Every device will be actingas a beacon and it is connected to temperature sensors. We can broadcast the temperature data along with the location of that particular area through beacons.

The persons who generally monitor these places will be given a wearabledevice which will be acting as a beacon scanner.Wheneverthe person enters the desired area then he can view the requiredparameters and can be alerted, these are send to cloud.


Cloud
Temperature sensor
Industrial Admin
Beacon sensor
Wearable device
Mobile application
Workers
User

Technical Architecture



c. User Stories

User Stories

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can register & access the dashboard with Gmail Login	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can access the dashboard	High	Sprint-1
	Dashboard	USN-6	As a user, I can enter the scores.	I can find eligibility	High	Sprint-2
Customer (Web user)	Registration	USN-7	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-8	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
		USN-9	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
		USN-10	As a user, I can register for the application through Gmail	I can register & access the dashboard with Gmail Login	Medium	Sprint-1
	Login	USN-11	As a user, I can log into the application by entering email & password	I can access the dashboard	High	Sprint-1
	Dashboard	USN-12	As a user, I can enter the scores.	I can find eligibility	High	Sprint-2
Customer Care Executive	Support	USN-13	As a Customer Care Executive, responding to queries via telephone, live chat etc.	Immediate response is sent.	Medium	Sprint-3
		USN-14	As a Customer Care Executive, Ask for and act on customer feedback	Thank you for your valuable feedback	High	Sprint-2
		USN-15	As a Customer Care Executive, analyse customer data and communication to adjust customer care strategies.	We'll look into that issue soon & try to rectify it	Low	Sprint-3
Administrator	Administrative functions	USN-16	As an Administrator, design, develop, maintain and troubleshoot websites.	No issues are found	High	Sprint-3
		USN-17	As an Administrator, view and manage user permissions in an application.	Allows the user to manage permissions	Low	Sprint-4
		USN-18	As an Administrator, implementing user protocols & creating backups.	Data is synced & later recovered	Medium	Sprint-4
		USN-19	As an Administrator, resolving software problems & updating new features.	Update to new features	High	Sprint-4

6. PROJECT PLANNING & SCHEDULING

a.Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Creation	USN-1	Creating a code for connecting sensor and Arduino.	4	High	Manikandan B Sabaresan S Nizrarhamed J Sarathkumar E
Sprint-1	Simulation	USN-2	Simulating the code.	4	Medium	Manikandan B Sabaresan S
Sprint-2	Device Creation	USN-3	Device creation using IOT Watson platform with credentials	4	High	Sabaresan S Nizrarhamed J
Sprint-2	Device performance	USN-4	Required performance of device using local node red platform.	4	Medium	Manikandan B Sabaresan S Nizrarhamed J Sarathkumar E
Sprint-3	Python code	USN-5	Python code for the temperature alert and humidity check.	4	Medium	Manikandan B Nizrarhamed J Sarathkumar E
Sprint-3	Monitoring	USN-6	Design an application for the project using MIT app inventor.	4	High	Sabaresan S Sarathkumar E
Sprint-4	Testing	USN-7	Test the application with required MIT AI2 Companion code.	4	Low	Manikandan B Sabaresan S Nizrarhamed J Sarathkumar E

b.Sprint Delivery Schedule

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

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

c.Reports from JIRA

7. CODING & SOLUTIONING:

a.Feature 1:

Hazardous Area Monitoring for Industrial Plant powered by IoT

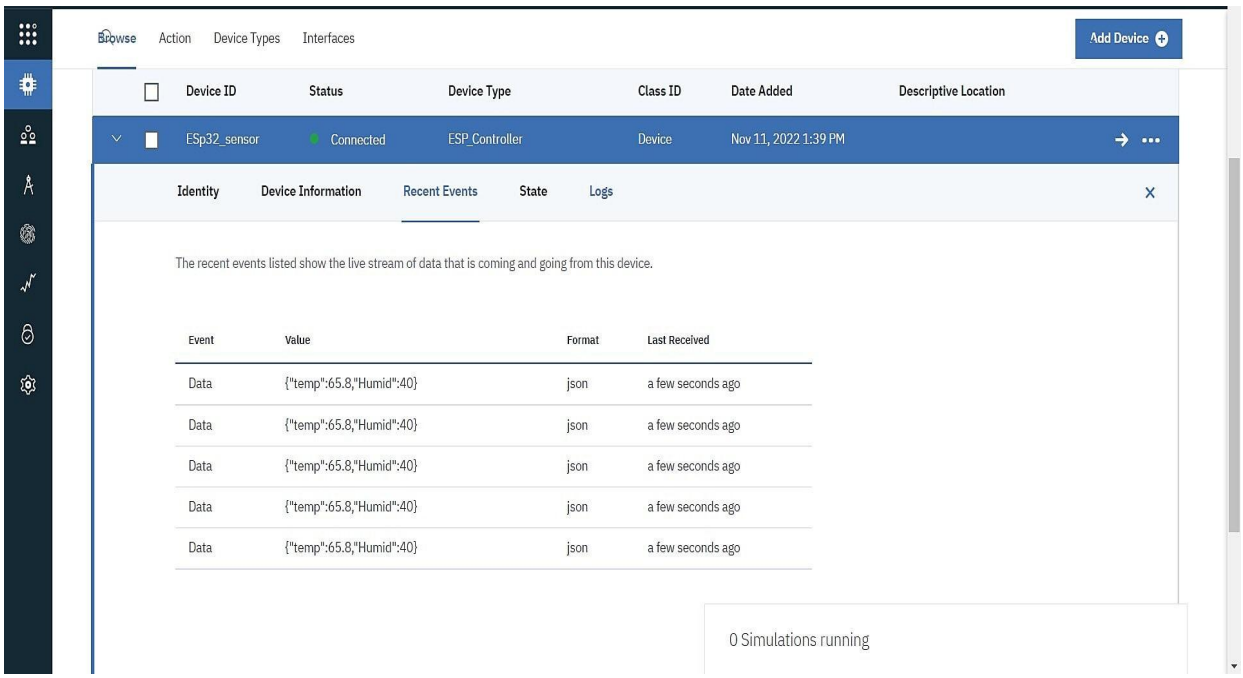
Languages :C++, Python

Tools/IDE :WOKWi, IBM Watson, Node-RED, IBM Cloudant DB,

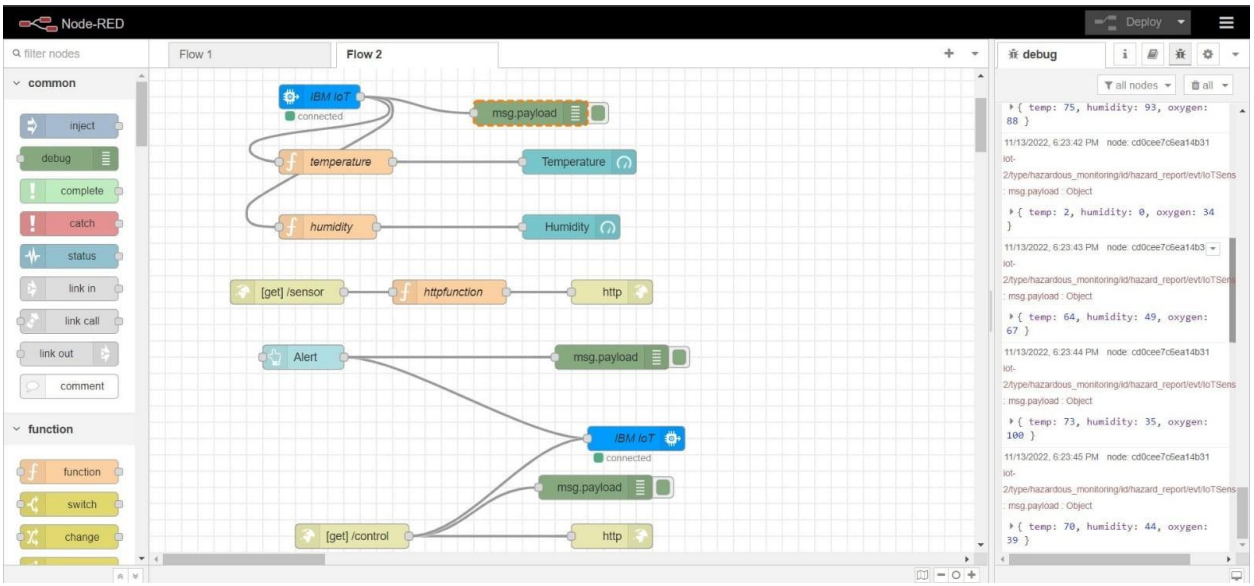
Python 3.7.4, MIT Invertor.

b.Feature 2:

Index

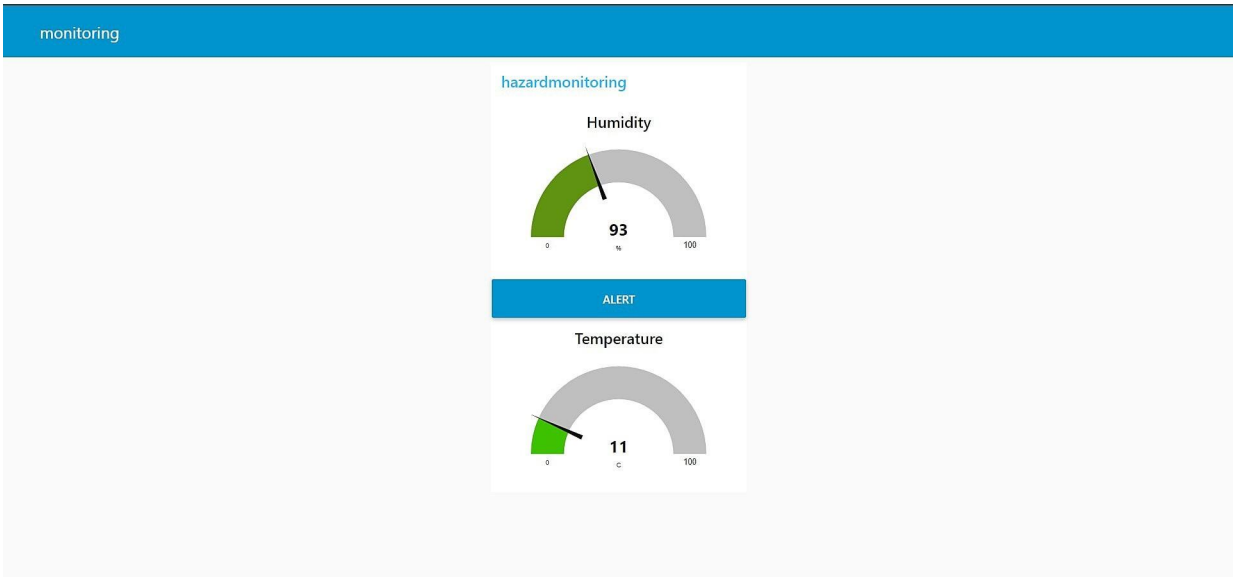


c.Database Schema

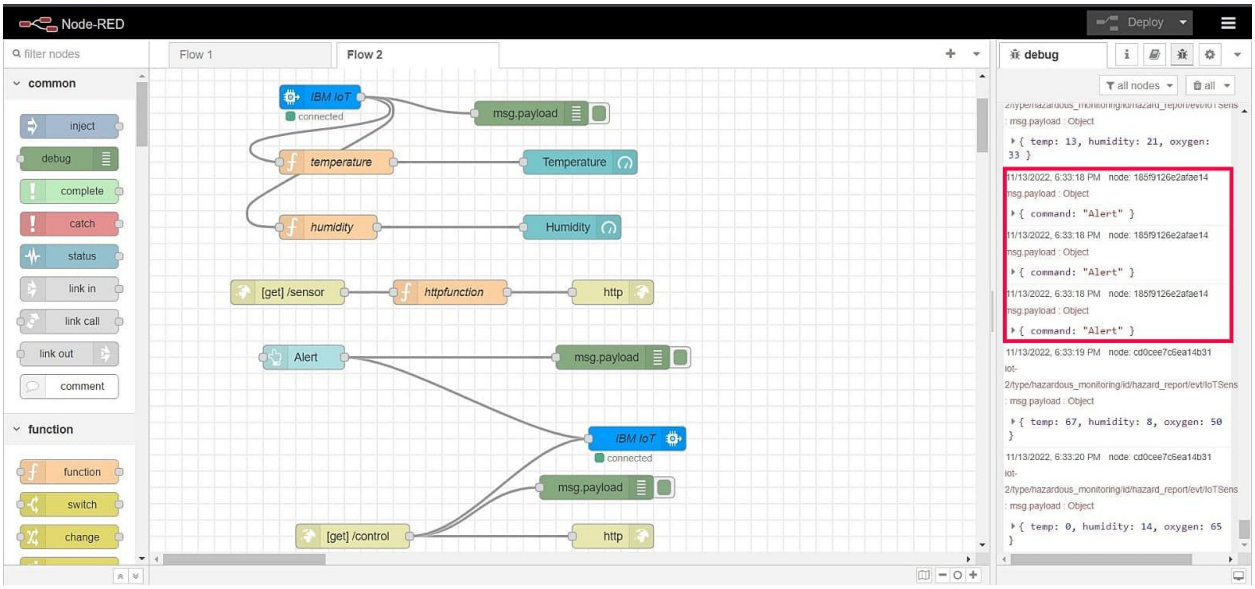


8. TESTING

a.Test Cases

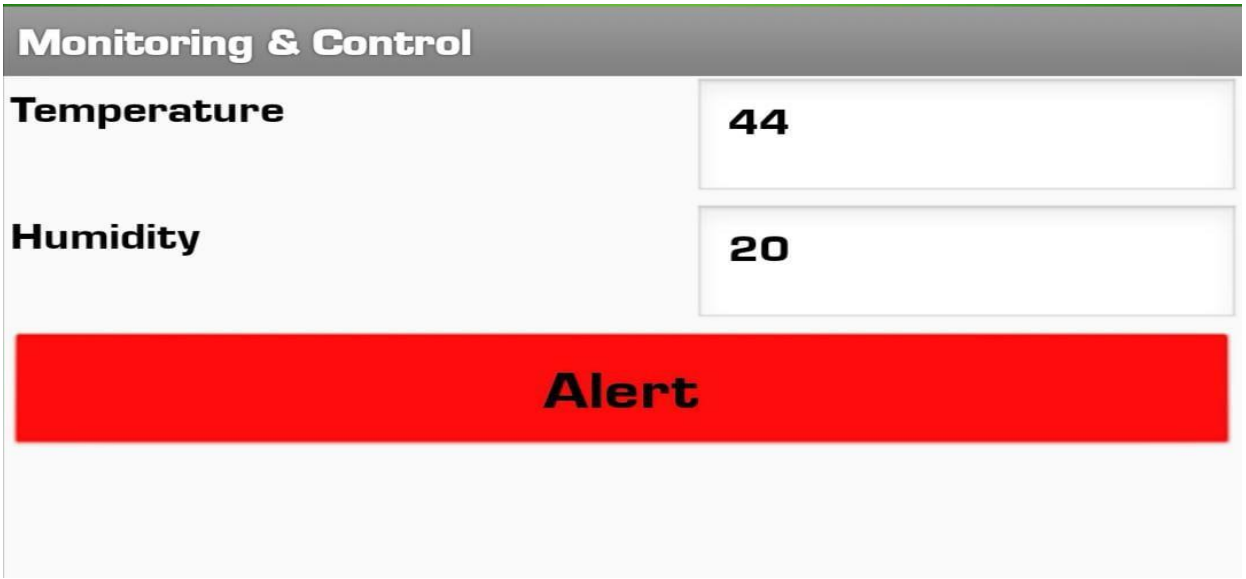


b.User Acceptance Testing

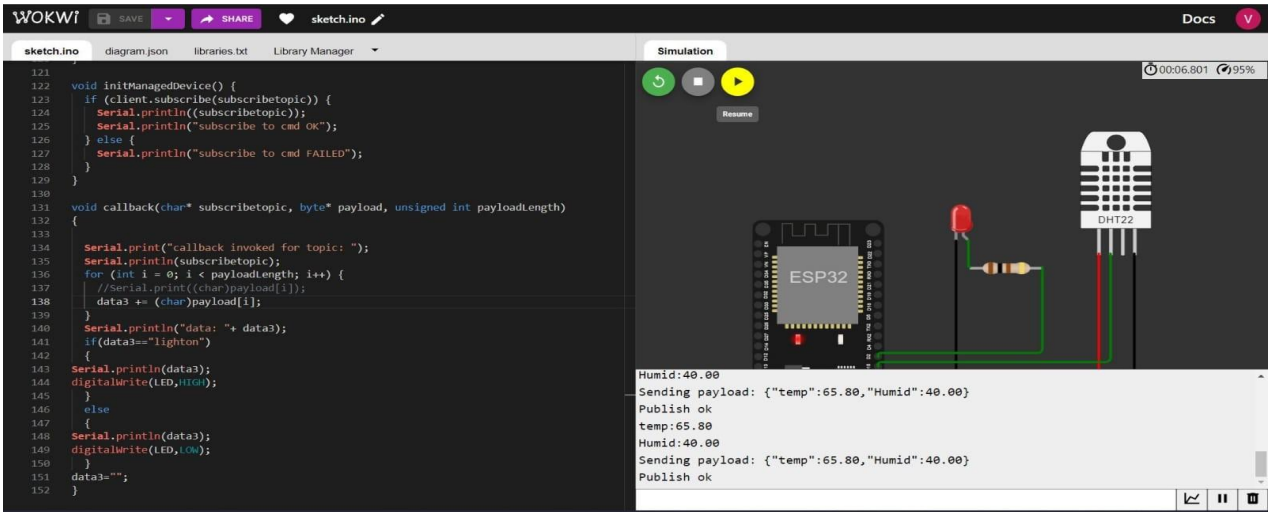


9. RESULTS

a.Performance Metrics



Sensor Performance:



10.ADVANTAGESandDISADVANTAGES

11.ADVANTAGES:

- Quickly Finding Any Issue In Production Line.
- Keeping Records Of Raw Materials & Accuracy.
- Predict what problem might occur.
- Decrease the deaths in Accidents.
- Ensuring saefty and comfort.
- No Need For Routine Survey.

DISADVANTAGES

- Misuse of privacy and data.
- Expense.
- Communication channel disconnection occurs oftern.
- Complex uses.

12.CONCLUSION:

The Internet of Things has a broad perspective in shaping tomorrow's world. Even though the IoT system has some demerits, its merits like saving consumer's time and money outstand its cons. It is predicted that soon IoT applications will be installed and used equally in both domestic and industrial areas. Companies are working hard to shoot back IoT disadvantages and making this futuristic technology more beneficial for the betterment of humanity.

13. FUTURE SCOPE:

IoT is bound to be an effective technology in the future, and IoT enabled devices are likely to be all-pervasive, from industry to households. The **future scope of IoT** is bright and varied, and it is only a matter of time before the above applications of the technology are realized.

While wearable technology allows patients to self monitor their health in real-time, the sensors and variants used in the healthcare industry are significantly more sophisticated. As sensors' accuracy and precision based on IoT increases, the share of manual errors in taking medical readings will decrease.

14. APPENDIX

a.Source Code

Code for connecting sensor and IBM Cloud:

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

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

"DHT.h"/  Library for dht11

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

DHTTYPE DHT22 / define type of sensor DHT 11#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 "0vbvyp"/ IBM ORGANITION ID

#define DEVICE_TYPE "ESP_Controller"/ Device type mentioned in ibm watson IOT Platform

#define DEVICE_ID "ESp32_sensor"/ Device ID mentioned in ibm watson IOT Platform #define

TOKEN "Q6w0Y9DwRfU1DWMJry"

String data3;

float h, t;



/ -------- 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 andformat in which data to be send

char subscribetopic[] = "iot-2/cmd/command/fmt/String";/ cmd REPRESENT command typeAND 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();

 pinMode(LED,OUTPUT);

 delay(10); Serial.println();

 wificonnect();

 mqttconnect();

}


void loop()/ Recursive Function
```

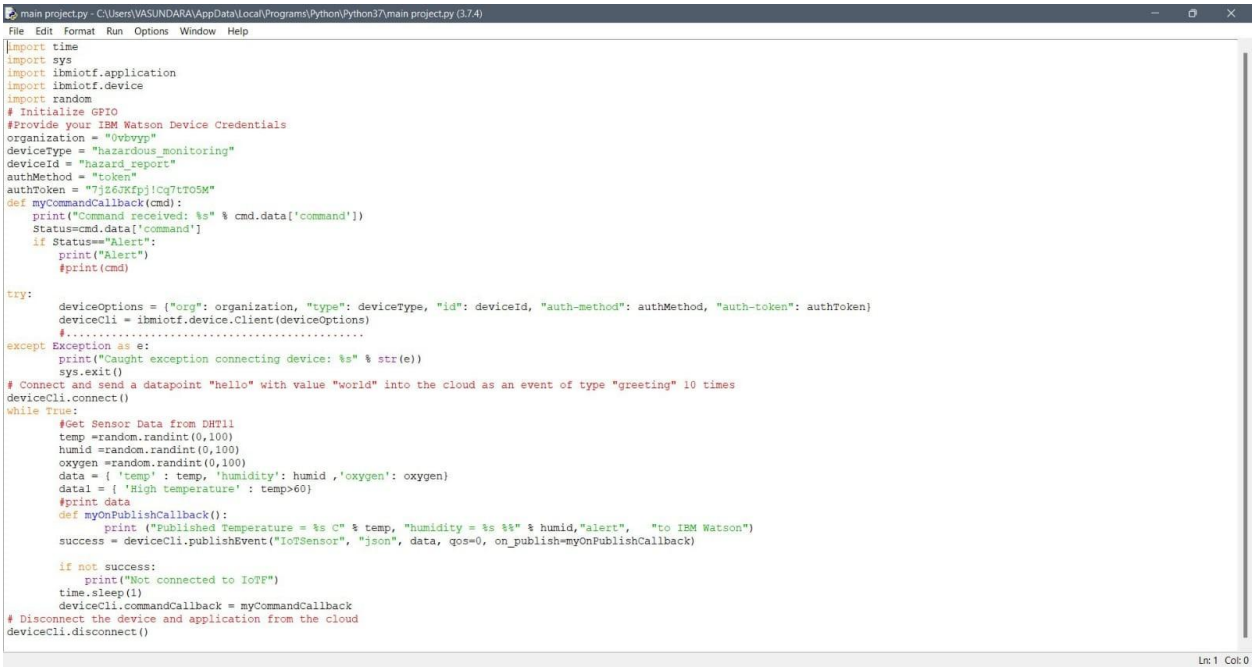
```
{

  h = dht.readHumidity();
  t = dht.readTemperature();
  Serial.print("temp:");
  Serial.println(t);
  Serial.print("Humid:");
  Serial.println(h);

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




/.....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 = "{\"temp\":";
```

```
  payload += temp;
  payload += "," "\"Humid\":";
  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
  } 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 Code for the Temperature Alert and Humidity check:



```
main project.py - C:\Users\VASUNDARA\AppData\Local\Programs\Python\Python37\main project.py (3.7.4)
File  Edit  Format  Run  Options  Window  Help
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
# Initialize GPIO
#Provide your IBM Watson Device Credentials
organization = "0vbvyp"
deviceType = "hazardous_monitoring"
deviceId = "hazard_report"
authMethod = "token"
authToken = "7jZ6JKfpj!Cq7tTO5M"
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    Status=cmd.data['command']
    if Status=="Alert":
        print("Alert")
        #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)
        oxygen =random.randint(0,100)
        data = { 'temp' : temp, 'humidity': humid ,'oxygen': oxygen}
        data1 = { 'High temperature' : temp>60}
        #print data
        def myOnPublishCallback():
              print ("Published Temperature = %s C" % temp, "humidity = %s %%" % humid,"alert",   "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()
                                                                                                                    Ln: 1  Col: 0
```

c. **Github &Project :** IBM-EPBL/**IBM-Project-26087-1660010747**

d. **Demo Link : https://uploadnow.io/f/Fl6xmT3**