A PROJECT REPORT

Hazardous Area Monitoring for Industrial Plant powered by IoT

Domain : Internet of Things.

Team ID : PNT2022TMID48690

College Name : NPR
College of Engineering and
Technology

G. Naga vishwa (920819106039)
Department of Electronics and Communication Engineering

V. Divesh (920819106016)
Department of Electronics and Communication Engineering

R. P. Ramvignhesh (920819106048)

Department of Electronics and Communication Engineering

M. Dhinesh(920819106015)
Department of Electronics and Communication Engineering

TABLE OF CONTENTS

13. APPENDIX

Source Code
GitHub & Project Demo Link

1. INTRODUCTION

1.1 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.

1.2 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

2.1 Existing problem

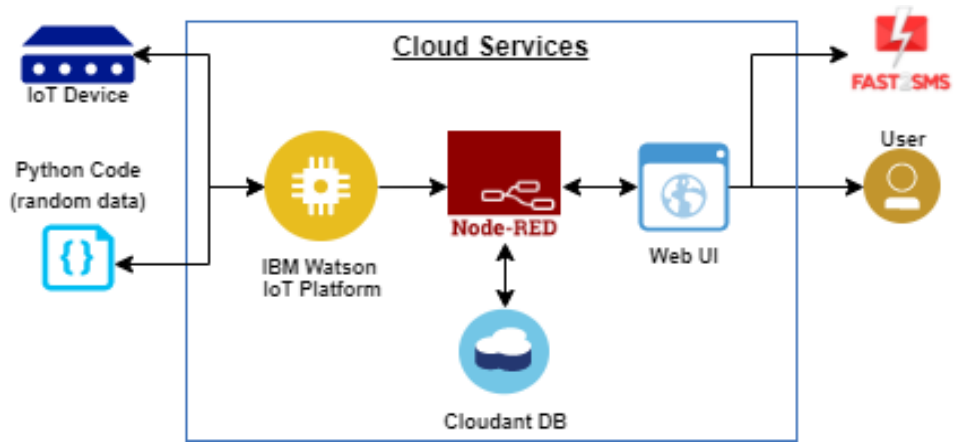
Internet of Things reception in conventional and slow changing modern plants, for example, power, water, oil-gas and substance has demonstrated to be helpful in giving business esteem by changing how information is used in direction and perception. Regular modern IoT use cases include securing information 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

2.2 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.

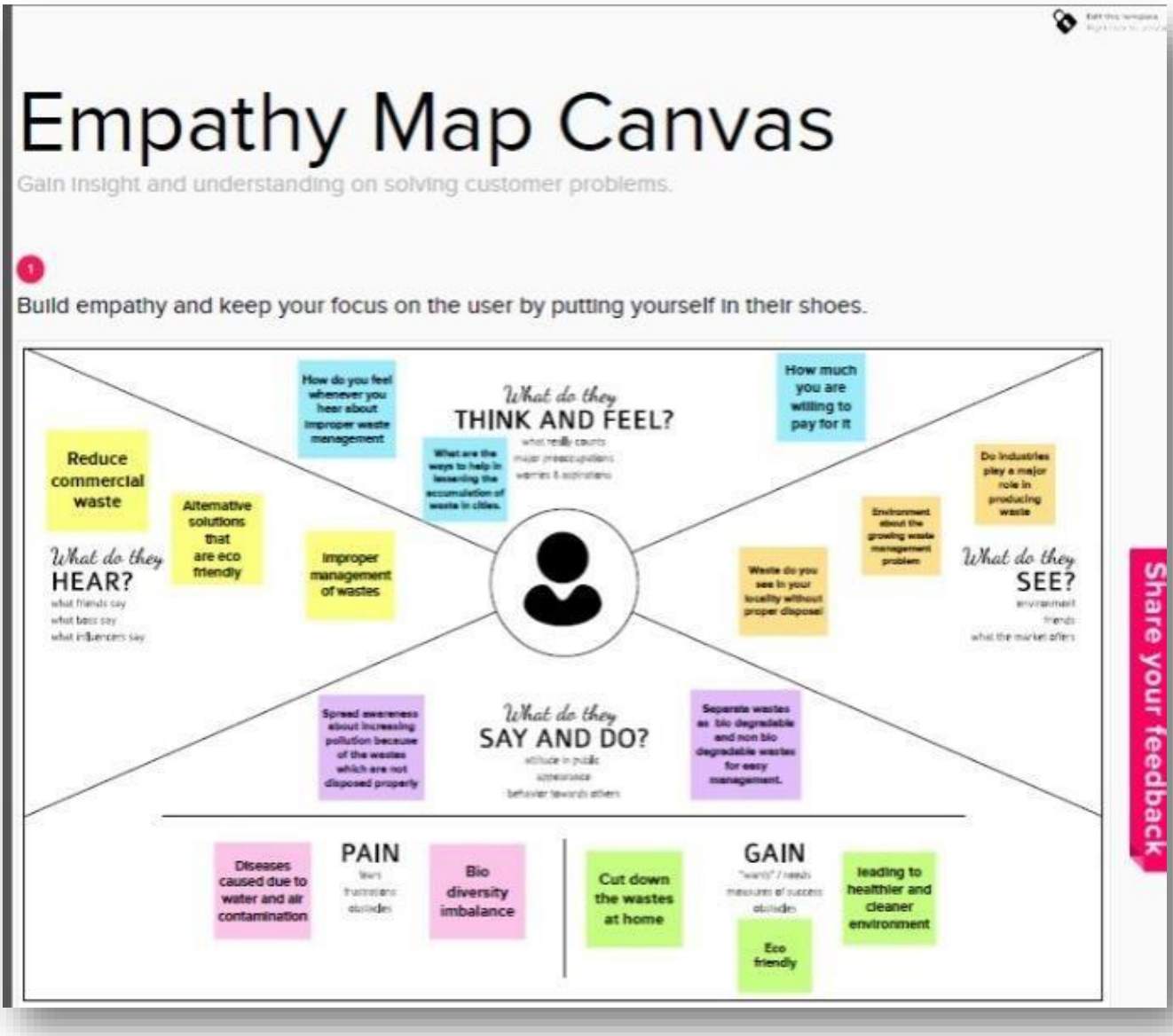
2.3Problem Statement Definition

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.

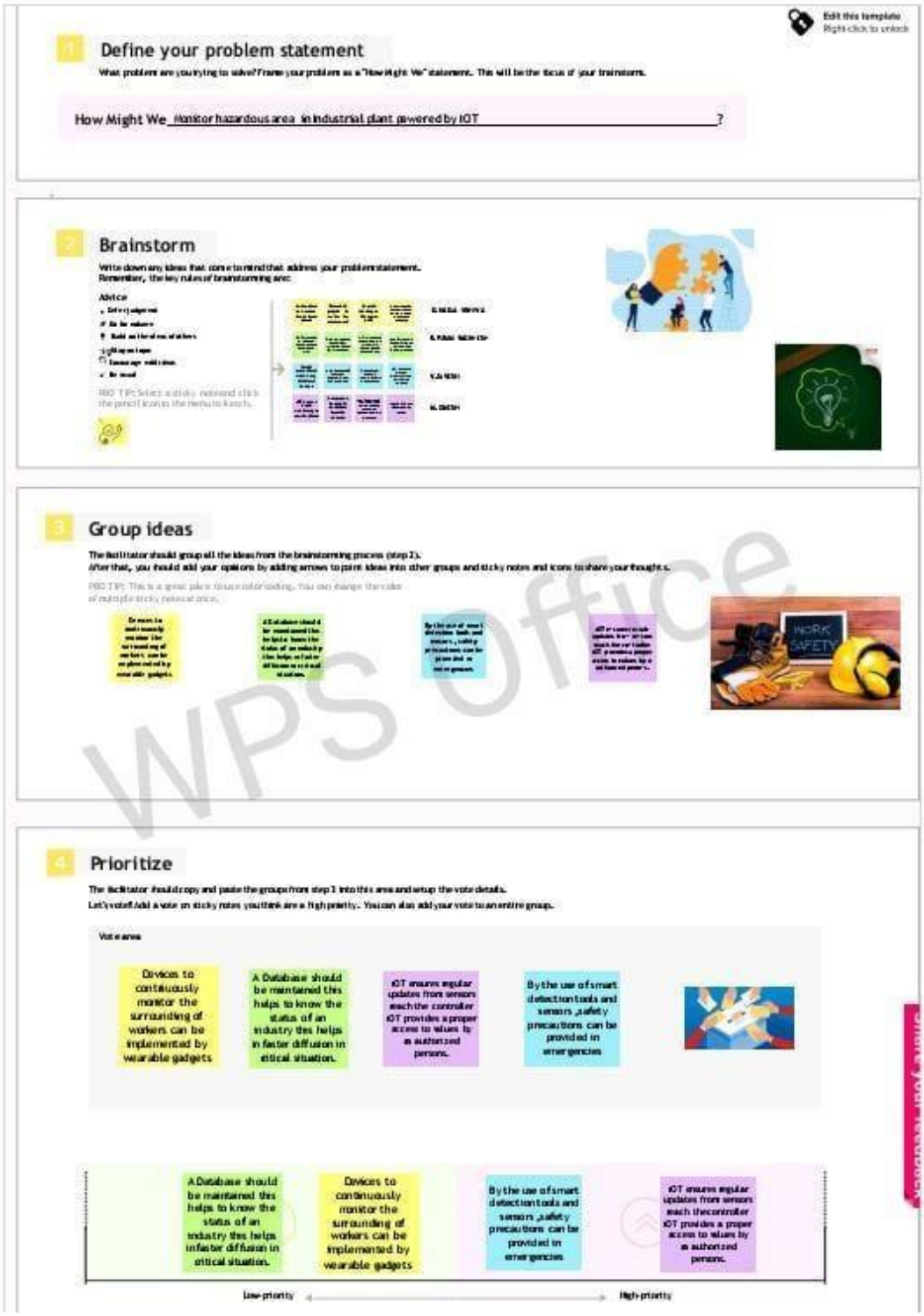


3. IDEATION & PROPOSEDSOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming



3.3 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 (Revenue Model)**

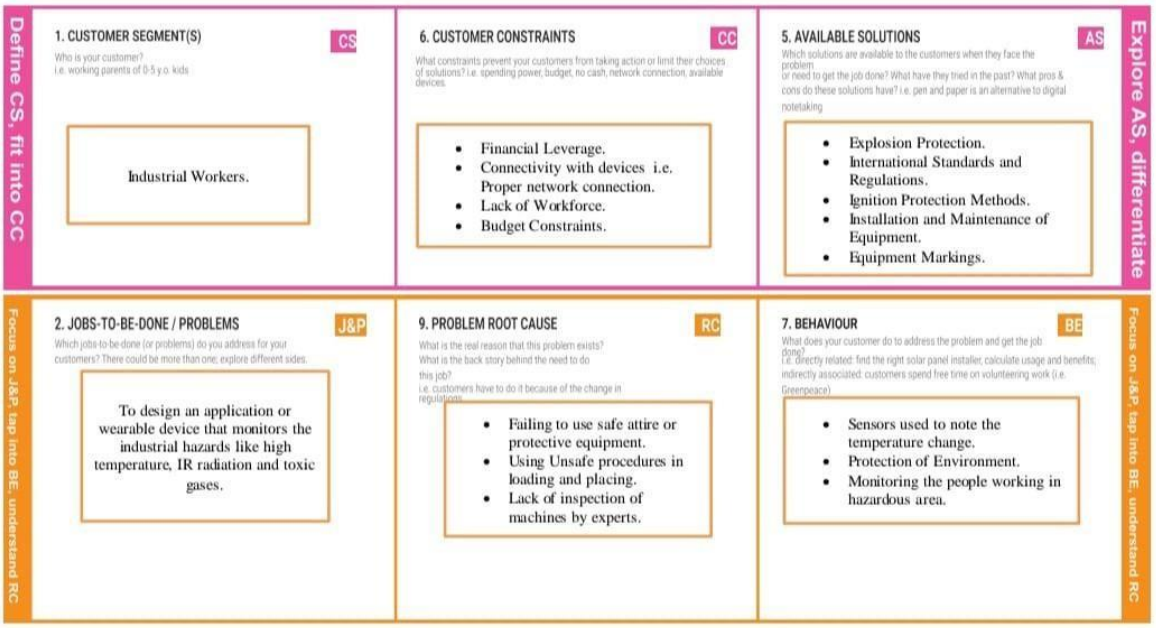
- Through our mobile application the revenue can be made in the form of pop-up advertisements, overlay ads from third party services.
- Wearable devices can be priced and sold by the industry to the workers.Can be implemented in different hazardous areas.

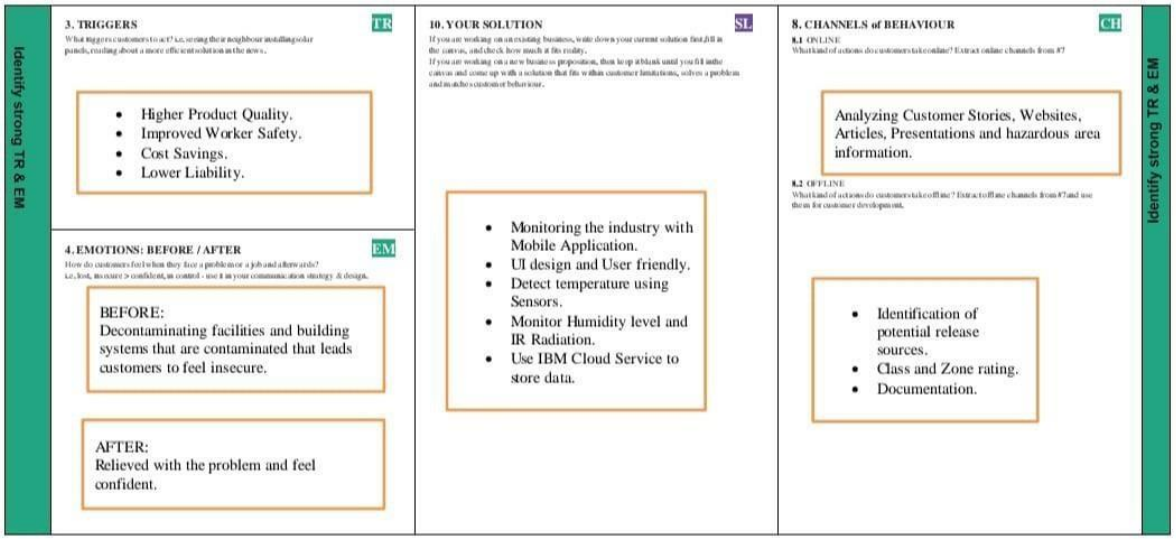
6. **Scalability of the Solution**

- It ensures the safety of each and every worker working in harmful gases and high temperature environment.
- By increasing the number of devices, this can be implemented in acommercial level.

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

3.4 Problem Solution fit





4. REQUIREMENT ANALYSIS

4.1 Functional requirement

FR No.	Functional Requirement (Epic)	Sub Requirement(Story/Sub-Task)
FR-1	**Data Gathering**	The smart beacon must be able to accurately determine the temperature of a specific area.
FR-2	**Location Detection**	A wearable device entering a dangerous region must be recognised by the smart beacon.
FR-3	**Beacon Data Syncing**	The smart beacon must be able to share its stored data with both the wearable device and admin dashboard through the cloud.
FR-4	**Wearable Device Display**	The temperature of the location where the worker is present must be displayed by the wearable device.
FR-5	**SMS Notification**	The workers should be informed through SMS to their phone that they need to leave the location if it is determined that the temperature has reached dangerous levels.
FR-6	**Admin Dashboard**	The admin is notified via the dashboard if the temperature is found to have reached dangerous levels, and they must then take the appropriate safety measures.

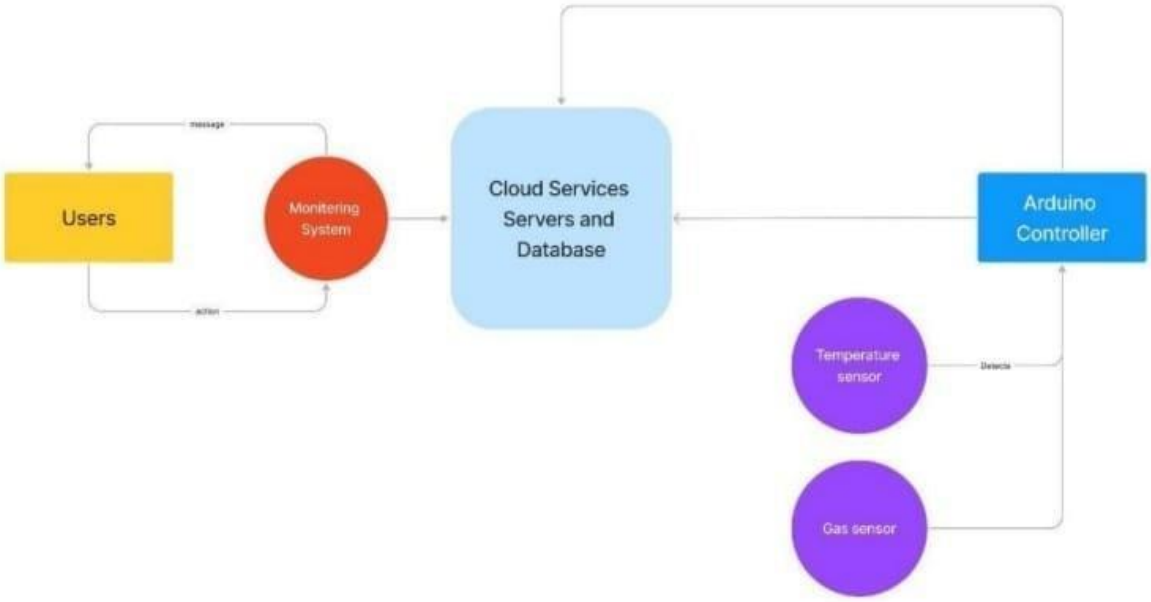
4.2 Non-Functional requirements

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The wearable device should be slim and shouldn't irritate or distract the workers who are wearing it. They should also clearly notify the workers when danger is identified and show the temperature consistently without significant delays.
NFR-2	**Security**	The connection of the beacons to the cloud and wearable devices should be secure. The security of the database that stores all of the temperature data needs to be strengthened.
NFR-3	**Reliability**	Even at harmful temperatures, the wearable device should be able to operate without any issues. If a problem is found, it should alert the user and the admin so it can be fixed immediately. For reliability, the beacons should also get routine maintenance.

NFR-4	**Performance**	High end sensors and processors are needed for the device to update temperature values in real time. It is important to minimise the time it takes for data to be sent to the cloud and other devices.
NFR-5	**Availability**	Regardless of where they are or what time it is, the user should be able to monitor the temperature of the region. In order to guarantee that safety measures may be taken when danger is recognised, the dashboard should be enabled at all times.
NFR-6	**Scalability**	Installing more smart beacon devices and connecting them to the same system as the existing beacons is all that is required to expand the area that needs to be watched. It is also extremely scalable because it may be repeated in several plants with various variables to be tracked.

5. PROJECT DESIGN

5.1 Data Flow Diagrams



5.2 Solution & Technical Architecture

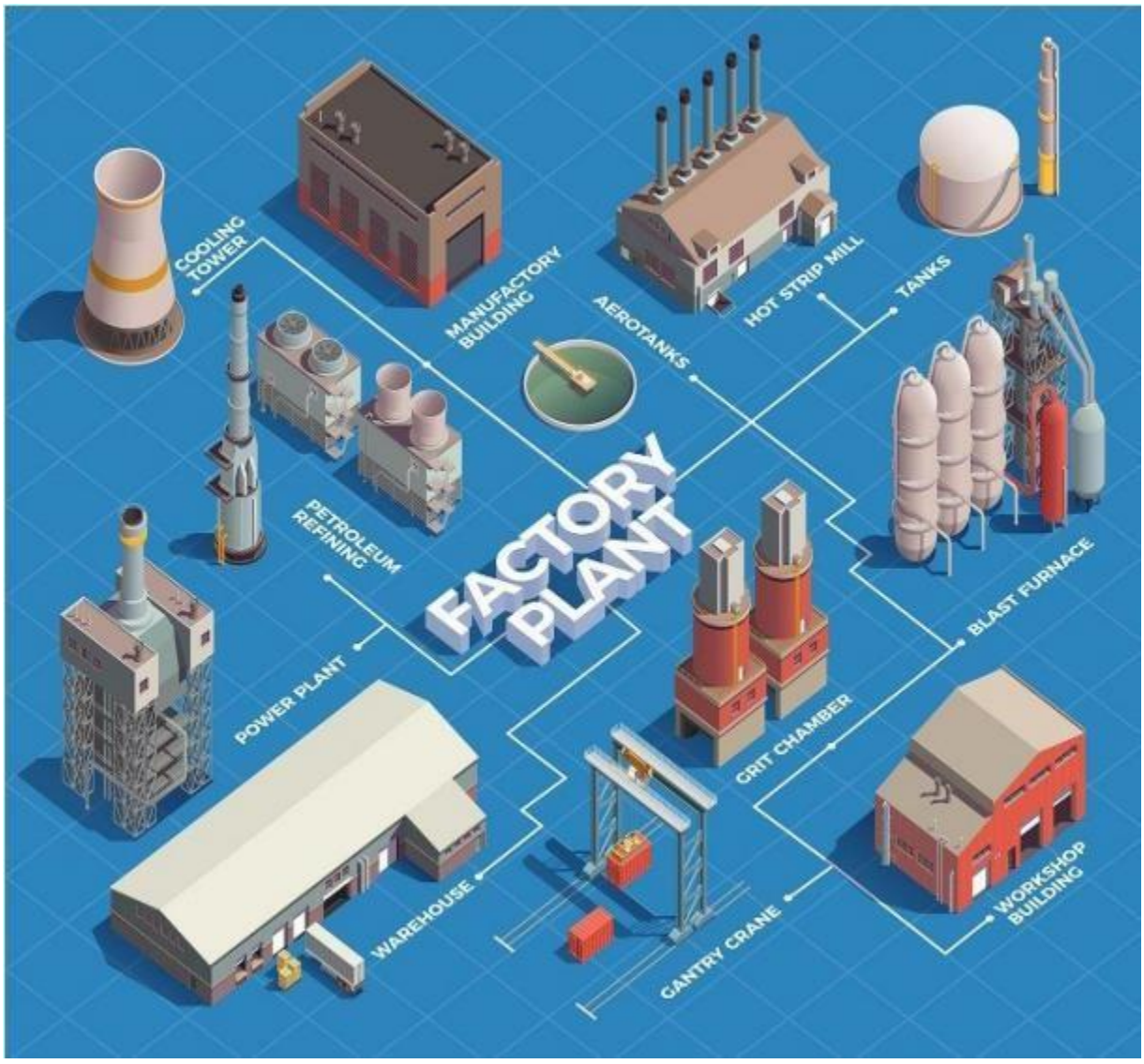
Solution Architecture

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

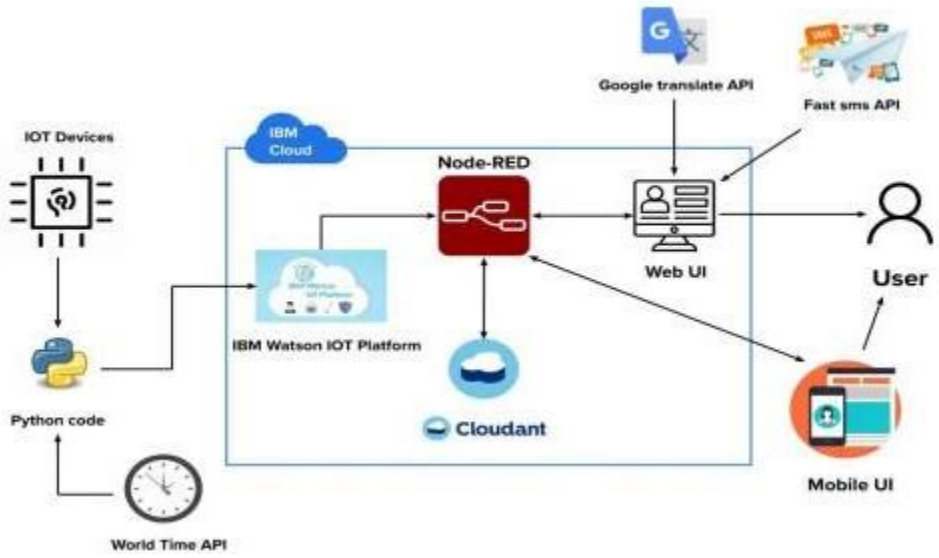
Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behaviour, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed, and delivered.

Example - Solution Architecture Diagram: HAZARDOUS AREA FOR MONITORING INDUSTRIAL POWER PLANT USING IOT.


FACTORY PLANT
COOLING TOWER
MANUFACTORY BUILDING
AEROTANKS
HOT STRIP MILL
TANKS
PETROLEUM REFINING
POWER PLANT
BLAST FURNACE
GRIT CHAMBER
WAREHOUSE
GANTRY CRANE
WORKSHOP BUILDING

Technical Architecture



1:Components&Technologies:

S.No	Component	Description	Technology
1.	UserInterface	Web UI, Mobile App, SMS service and Wearabledevices	Node-RED, Fast sms and MIT Appinventor
2.	ApplicationLogic-1	Gettinginputfromsmartbeacons	EmbeddedCandPython
3.	ApplicationLogic-2	Processdataincloud	IBMWatsonIOTplatform,Cloudant DBandNode-RED
4.	ApplicationLogic-3	Displaydatatotheuser	WebUI,FastsmsandMobile application
5.	Database	Realtimedatabase	CloudantDB
6.	CloudDatabase	DatabaseServiceonCloud	IBMCloudant
7.	ExternalAPI-1	Tosendsmstouser	FastsmsAPI
8.	ExternalAPI-2	Languageforthewebsiteiswrittentobedynamic	Google translateAPI
9.	ExternalAPI-3	Toaccesstime	WorldtimeAPI
10.	SmartBeacon	Tomonitortheareaandupdatethestatsinthecloud	NodeMCUandSensors
11.	Infrastructure(Server/Cloud)	ApplicationDeploymentonCloud	IBMCloud

Table-2:ApplicationCharacteristics:

S.No	Characteristics	Description	Technology
1.	Open-SourceFrameworks	TheNode-REDopensourceframeworksareusedto build the web application as well as tocommunicate with the mobile application and tohandlealertsms	Node-REDframework
2.	ScalableArchitecture	The3–tierarchitectureusedwithaseparateuserinterface, application tier and data tier makes iteasilyscalable	IBMWatsonStudio
3.	Availability	Thewebapplicationishighlyavailableasitisdeployedincloud	IBMCloud
4.	Performance	Theperformanceofthewebsiteisimprovedwith cachingandsecurity	IBMCloudInternetServices

5.3 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
		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

6.1 Sprint Planning & Estimation

Product Backlog, Sprint Schedule, and Estimation

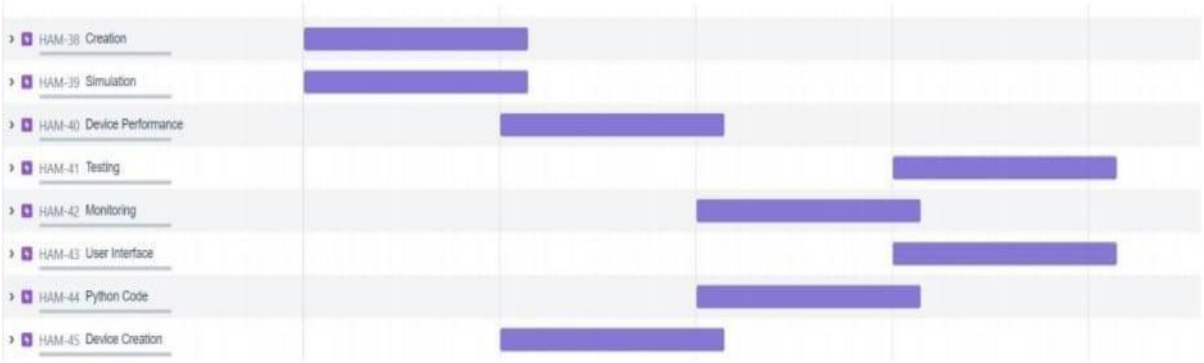
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Creation	USN-1	Creating a code for connecting sensor and Arduino.	4	High	Ram vignhesh
Sprint-1	Simulation	USN-2	Simulating the code.	4	Medium	Ram vignhesh
Sprint-2	Device Creation	USN-3	Device creation using IOT Watson platform with credentials	4	High	Dhinesh
Sprint-2	Device performance	USN-4	Required performance of device using local node red platform.	4	Medium	Dhinesh
Sprint-3	Python code	USN-5	Python code for the temperature alert and humidity check.	4	Medium	Divesh
Sprint-3	User interface	USN-6	Creation of web UI (user interface) connected to the software.	4	High	Divesh
Sprint-4	Monitoring	USN-7	Design an application for the project using MIT app inventor.	4	Low	Naga vishwa
Sprint-4	Testing	USN-8	Test the application with required MIT AI2 Companion code.	4	High.	Naga vishwa

6.2 Sprint Delivery Schedule

Project Tracker, Velocity & Burndown Chart:

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	12 NOVEMBER 2022	17 NOVEMBER 2022	20	17 NOVEMBER 2022
Sprint-2	20	6 Days	12 NOVEMBER 2022	17 NOVEMBER 2022	20	17 NOVEMBER 2022
Sprint-3	20	6 Days	12 NOVEMBER 2022	17 NOVEMBER 2022	20	17 NOVEMBER 2022
Sprint-4	20	6 Days	12 NOVEMBER 2022	17 NOVEMBER 2022	20	17 NOVEMBER 2022

6.3 Reports from JIRA



7. CODING & SOLUTIONING:

7.1 Feature 1:

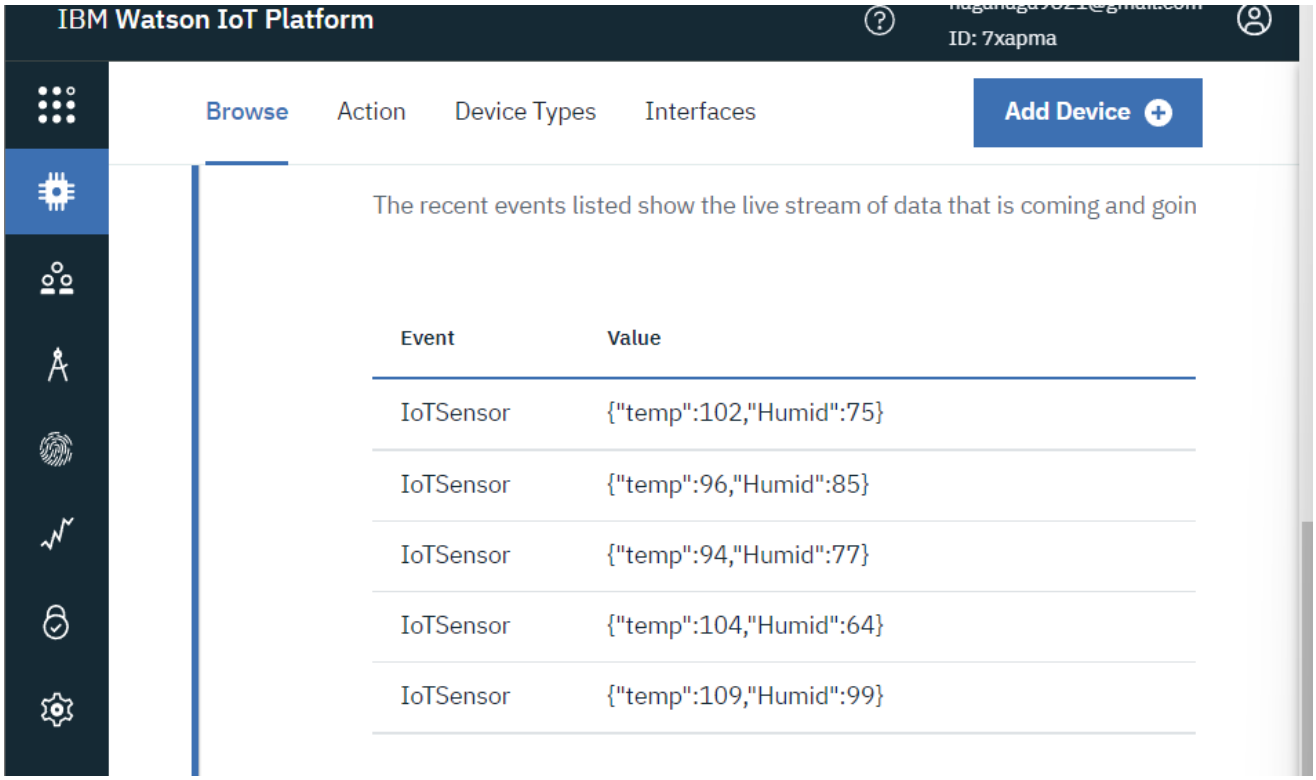
Hazardous Area Monitoring for Industrial Plant powered by IoT

Language : C++, Python

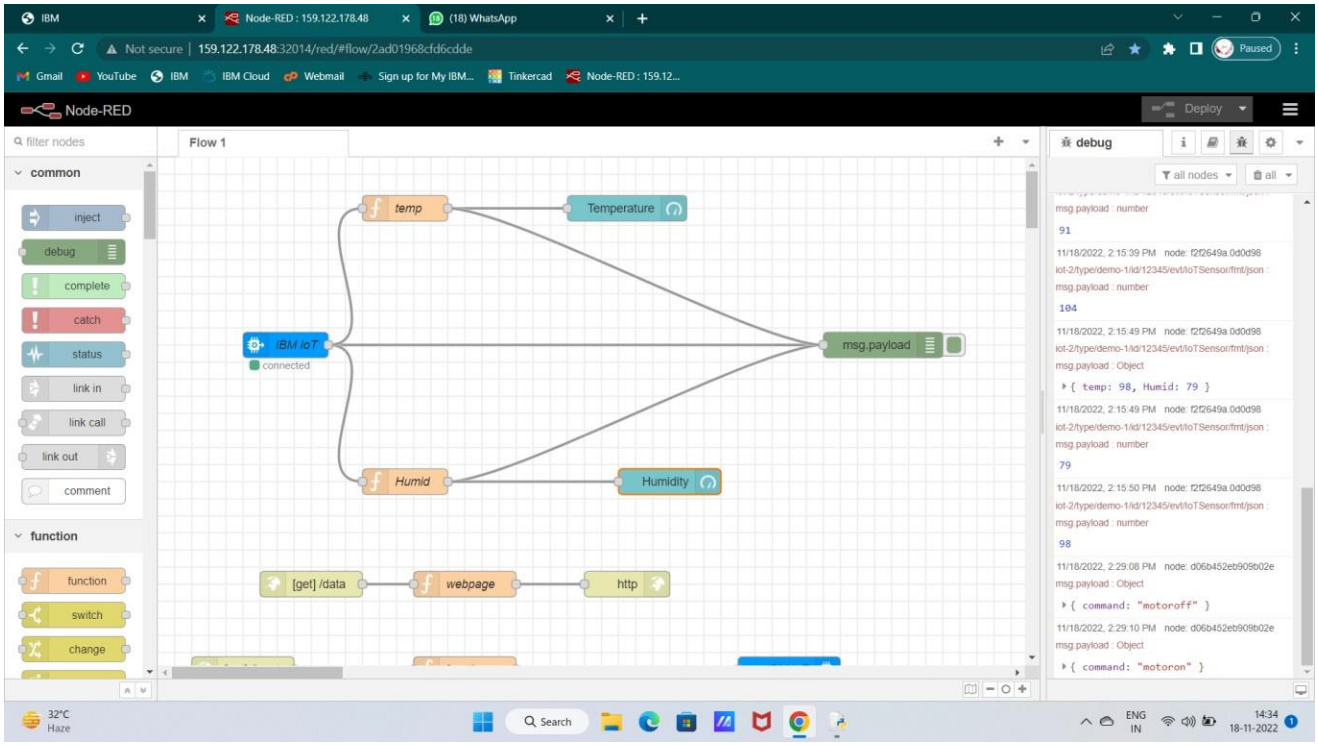
Tools/IDE : WOKWi, IBM Watson, IBM Node-RED, IBM Cloudant DB, Python 3.7.0, MIT Invertor.

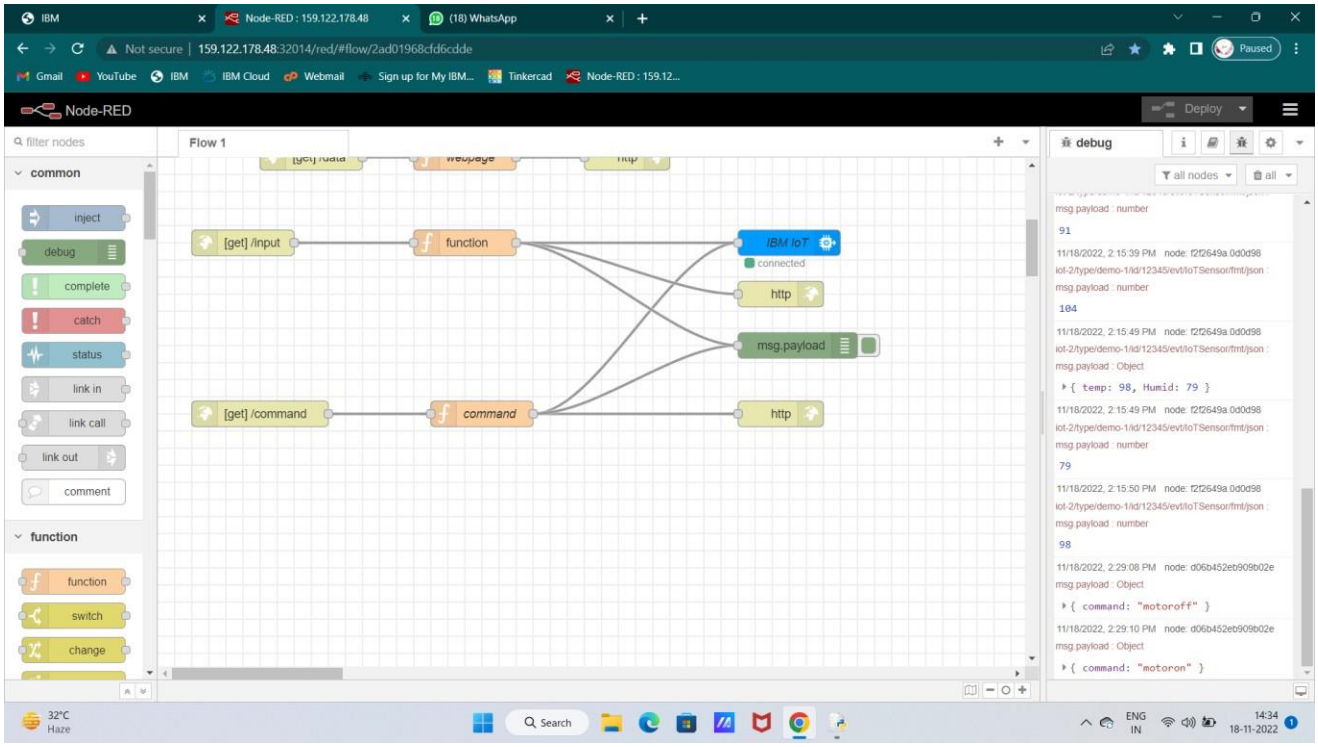
7.2 Feature 2:

Index



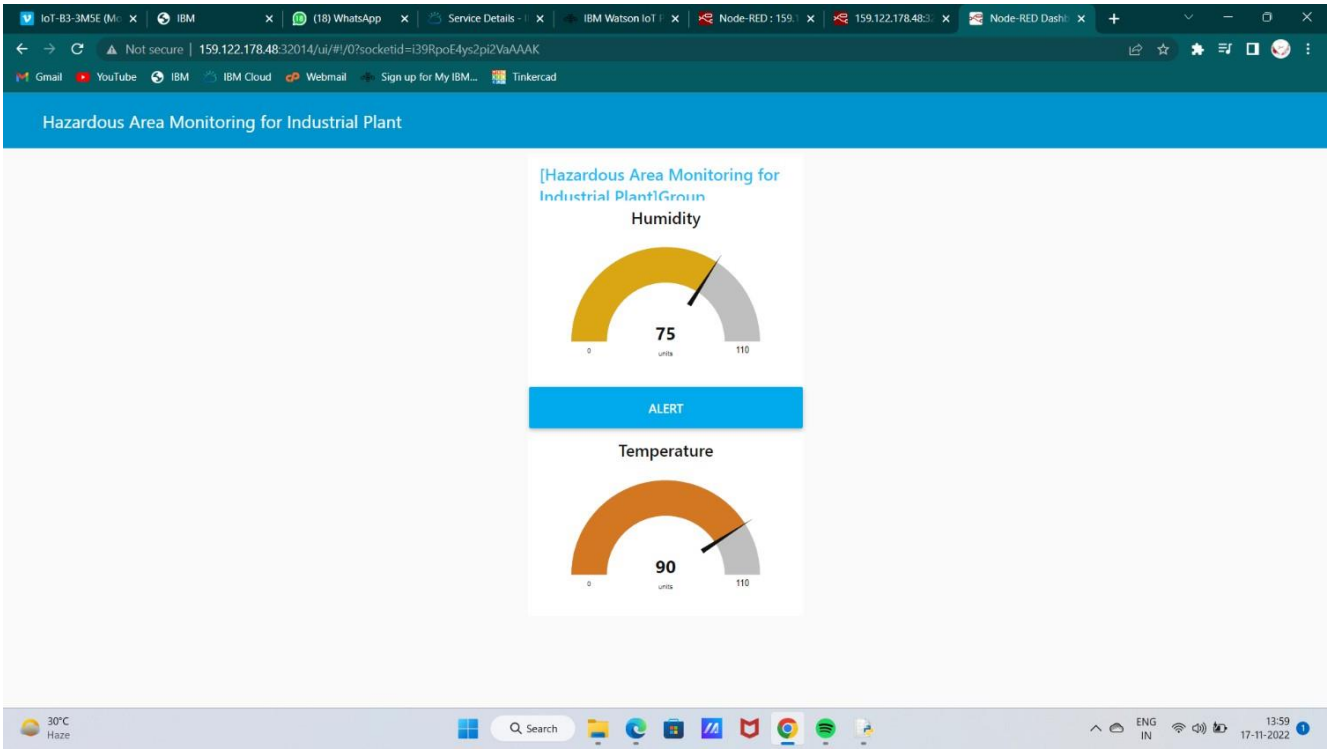
7.3 Database Schema



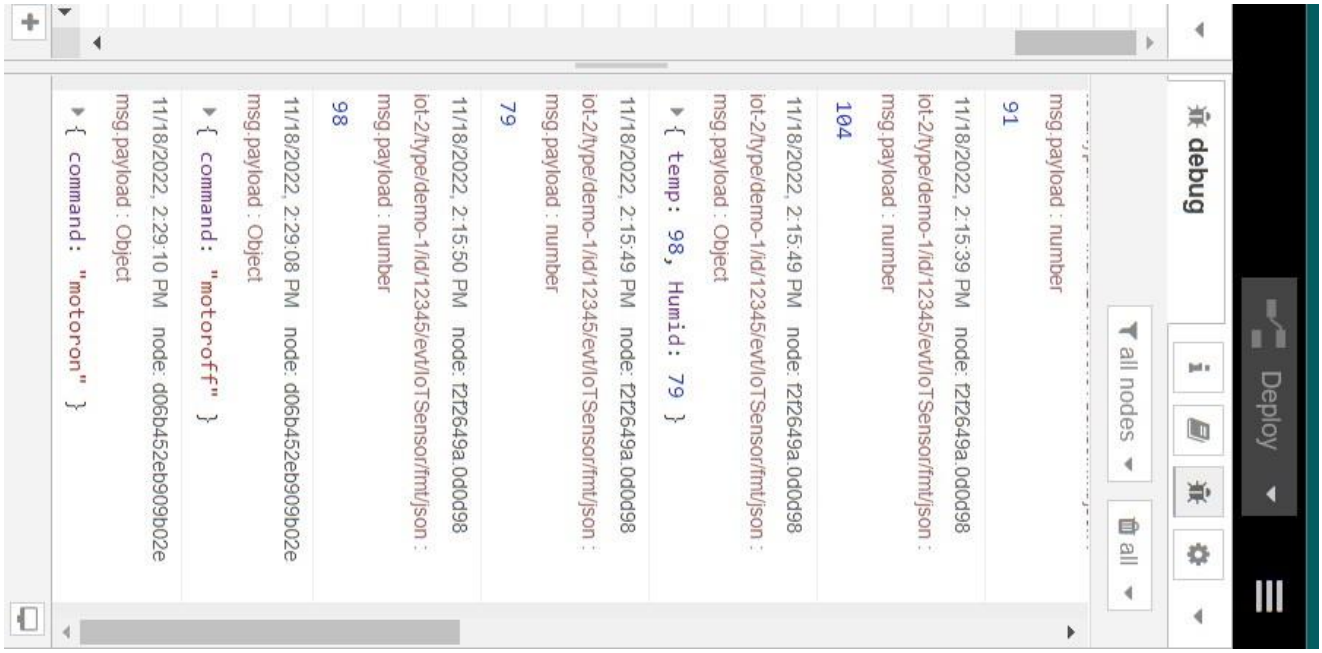


8. TESTING

8.1 Test Cases

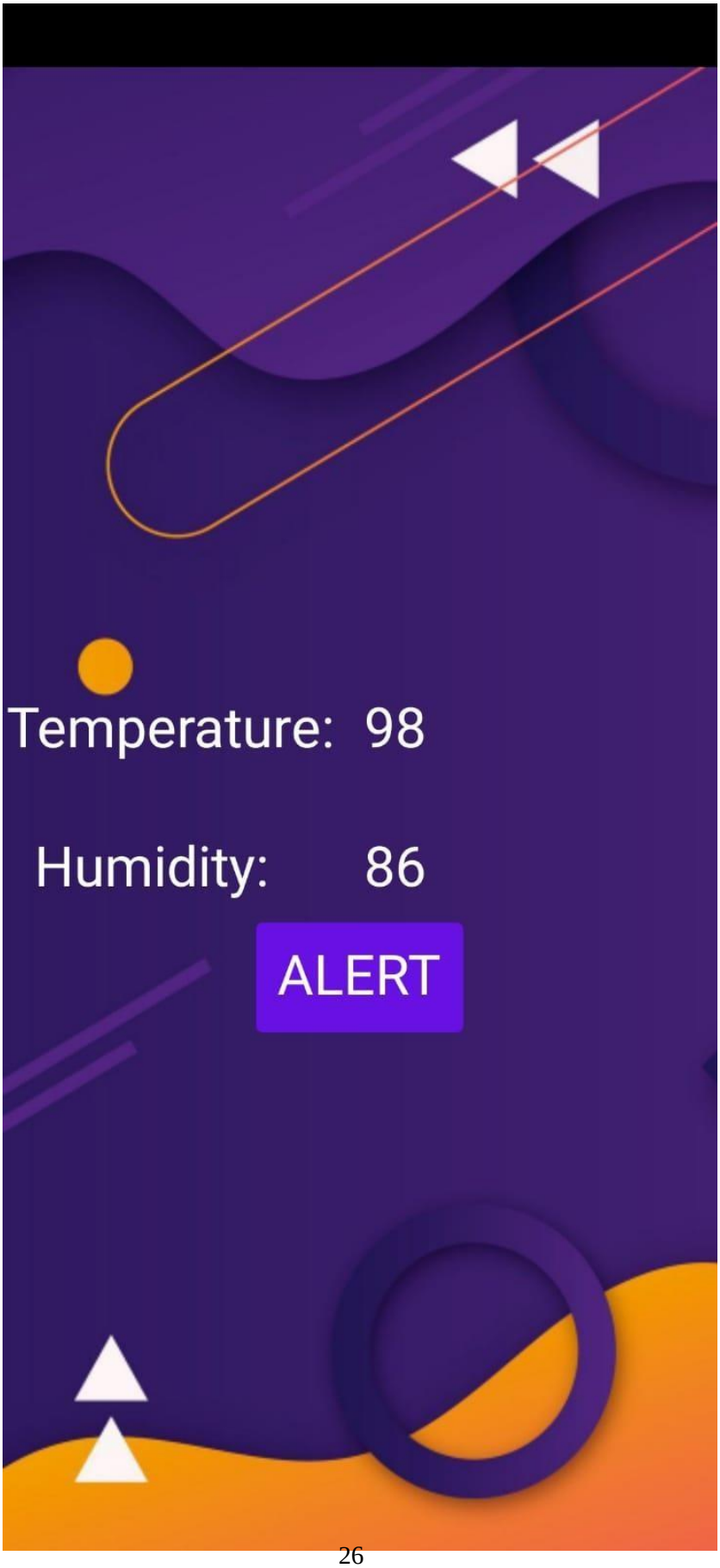


8.2 User Acceptance Testing



9. RESULTS

9.1 Performance Metrics


Temperature: 98
Humidity: 86
ALERT

10. ADVANTAGES and

DISADVANTAGESADVANTAGES:

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

DISADVANTAGES

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

11. 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.

12. 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.

13. APPENDIX

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

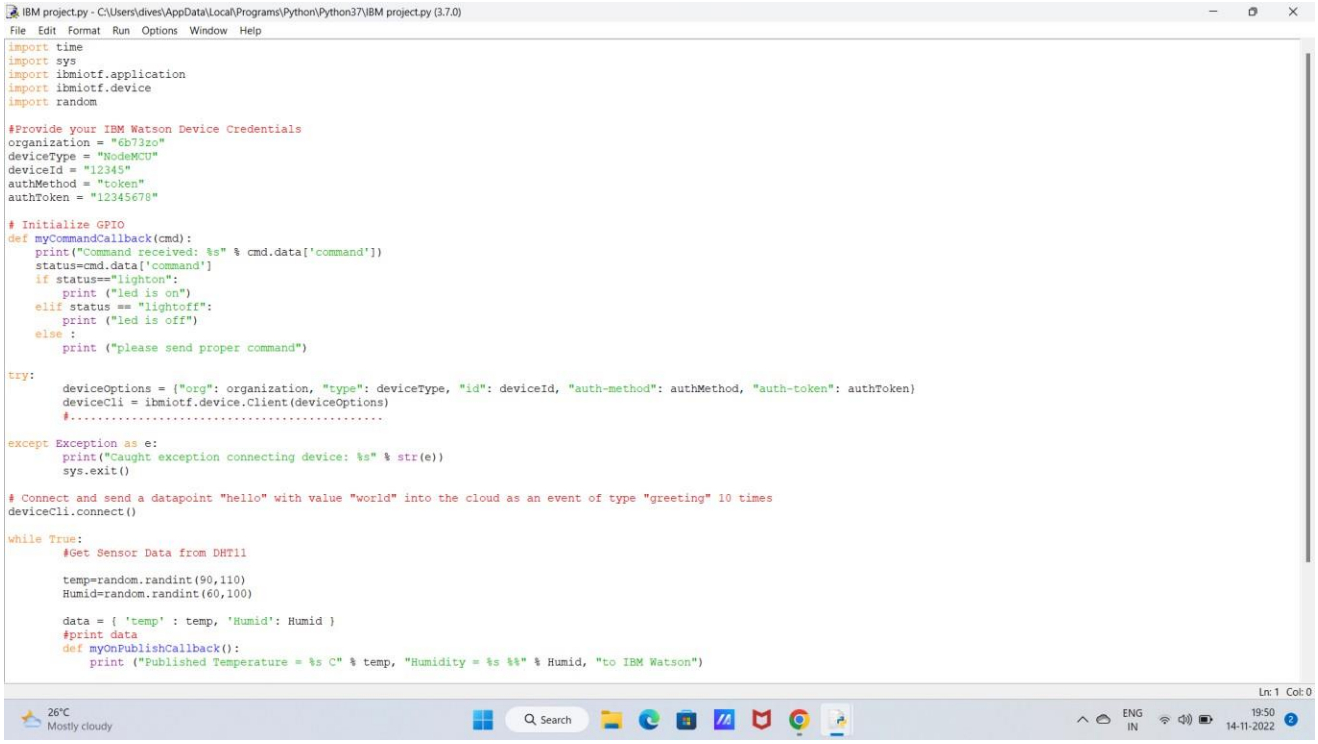
```
{

  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:



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

#Provide your IBM Watson Device Credentials
organization = "6b73zo"
deviceType = "NodeMCU"
deviceId = "12345"
authMethod = "token"
authToken = "12345678"

# Initialize GPIO
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="lighton":
        print ("led is on")
    elif status == "lightoff":
        print ("led is off")
    else :
        print ("please send proper command")

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(90,110)
        Humid=random.randint(60,100)

        data = { 'temp' : temp, 'Humid': Humid }
        #print data
        def myOnPublishCallback():
            print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % Humid, "to IBM Watson")
```

GitHub Link : https://github.com/IBM-EPBL/IBM-Project-42227-1660656579

Project Demo Link : https://github.com/IBM-EPBL/IBM-Project-42227-1660656579/tree/main/project%20video