PROJECT REPORT

DATE	18 November 2022
TEAM ID	PNT2022TMID10955
PROJECT NAME	Project - Hazardous Area Monitoring for Industrial Plantpowered by IoT

ABSTRACT:

The proper operation of the industrial process depends in large part on the core area of industrial safety.The manufacturing industry is one of the most dangerous in terms of work safety. Among different kinds of accidents in industrial environments, the biggest causes involving humans are related to machines and equipment used for manufacturing. Although there are standards and regulations for machines' safe operation, some specific criteria could only be identified by specialists and managers in Environment, Health, and Safety (EHS). The internet of things (IoT) is a fundamental technology for Industry 4.0, bringing many benefits for automation and process control.

INTRODUCTION:

The Industrial Monitoring System project uses the Internet of Things (IoT) as its foundation. Arduino is utilised to control a variety of sensors (using smoke and temperature sensors), giving the industry total control. This project uses the Internet of Things (IoT) to give users access to data. The Internet of Things (IoT) is a system of interconnected "things" that enables physical objects to exchange data via sensors, electronics, software, and networking. These autonomous systems can function without human interaction. The continuous monitoring sensors alert the in charge of the area by sending an alert message as well as with an alarm. The sensors continuously monitor the temperature, leakage of gases and humidity and take necessary measures to ensure the safety of the workers in the industry.

LITERATURE SURVEY:

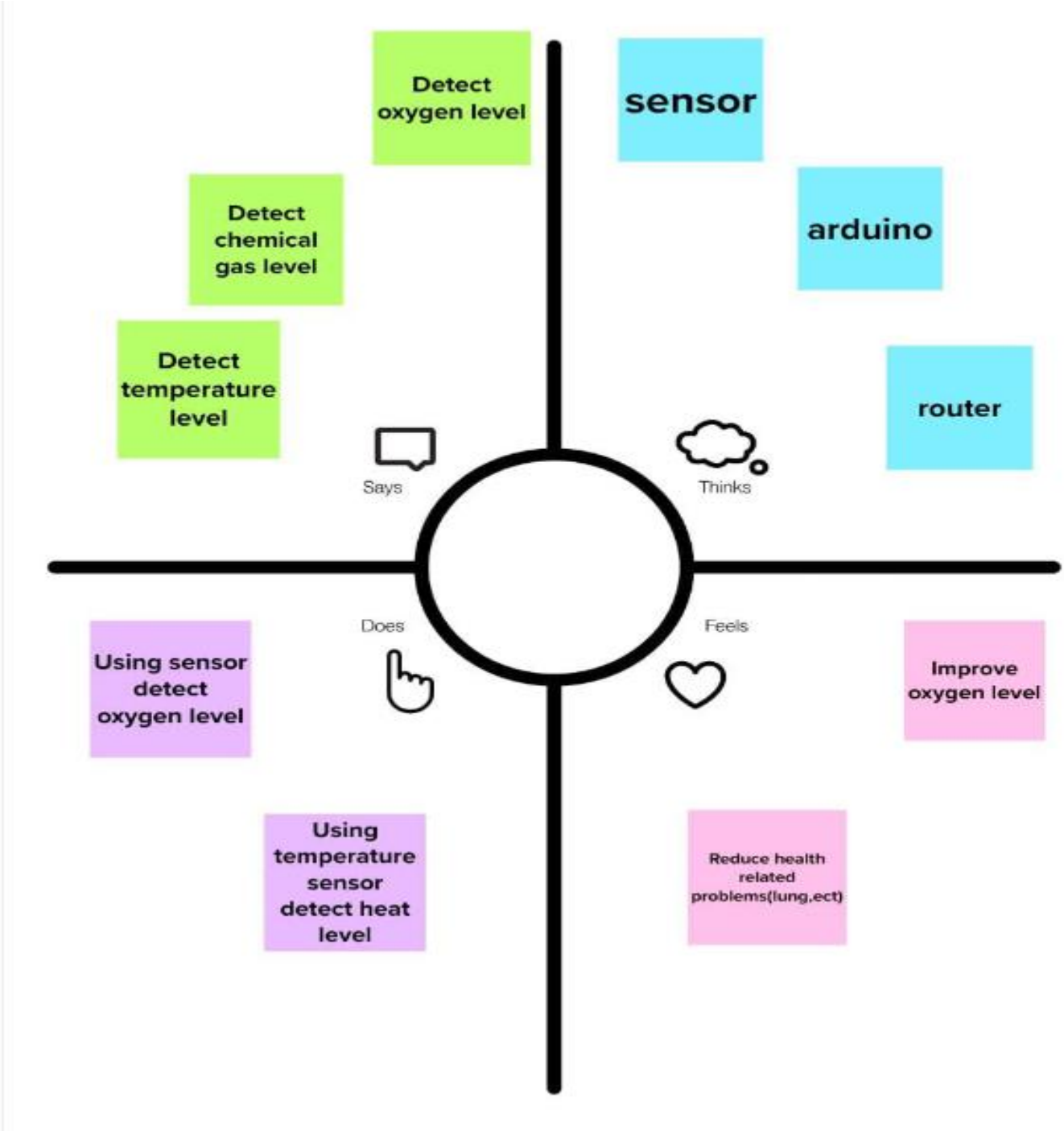
V S D Rekha published on " International Conference on Electronics and Renewable Systems (ICEARS) 2022".IM Everyday life is undergoing technological developments for a higher standard of living as well as for safety and security. IoT is a wireless communication development. The concept of gas leakage detection with an alert system using the internet of things has been shown in this research. Numerous gases that are dangerous to life are all around us yet may go unnoticed. If there is a gas leak over the places that are important to households, industrial plant personnel, and oil refinery employees, and the gases are flammable, there will be a loss of life. Therefore, this study suggests a configuration that can notify us anytime a gas release occurs. The Arduino Nano is utilised in this essay. Gases like LPG are detected using the MQ-2 Gas Sensor. The Arduino Nano is wired to monitor things like carbon monoxide. The 16x2 LCD panel and GSM module on the Arduino Nano are interfaced. The GSM module calls and sends an alarm SMS to the individual. When a gas leak is discovered, the system quickly provides an alert signal by buzzing, lighting the LED connected to the MCU, and displaying a warning message on the LCD. Gas leaking is signalled by green and red LEDs.

Michael Frey Published on "IEEE 5th World Forum on Internet of Things (WF-IoT) 2019".Sensors are typically used in industrial production plants to monitor or record operations, and actuators are used to enable corrective actions in the event of errors, failures, or harmful situations. Embedded controllers connect these "things" to local networks, which are now made possible by the Internet of Things (IoT). These local networks are frequently wireless low-power networks that connect to a cloud via the global Internet. Under the industrial IoT, interconnected sensors and actuators form a crucial subsystem that typically operates in challenging circumstances. How to interconnect vital industrial components in a secure and safe way is now up for discussion. In this study, we examine ICN's potential to offer limited controllers in industrial safety systems a secure and reliable networking solution. Hazardous gas sensing is demonstrated here. Compare with IP-based techniques like CoAP and MQTT in common industrial settings, such as refineries. Based on our research, information centric networking should be implemented in a safety-critical industrial IoT due to the content-centered security model and improved DoS resistance. Evaluation of the RIOT operating system's crypto efforts for content security reveals their viability in typical deployment settings.

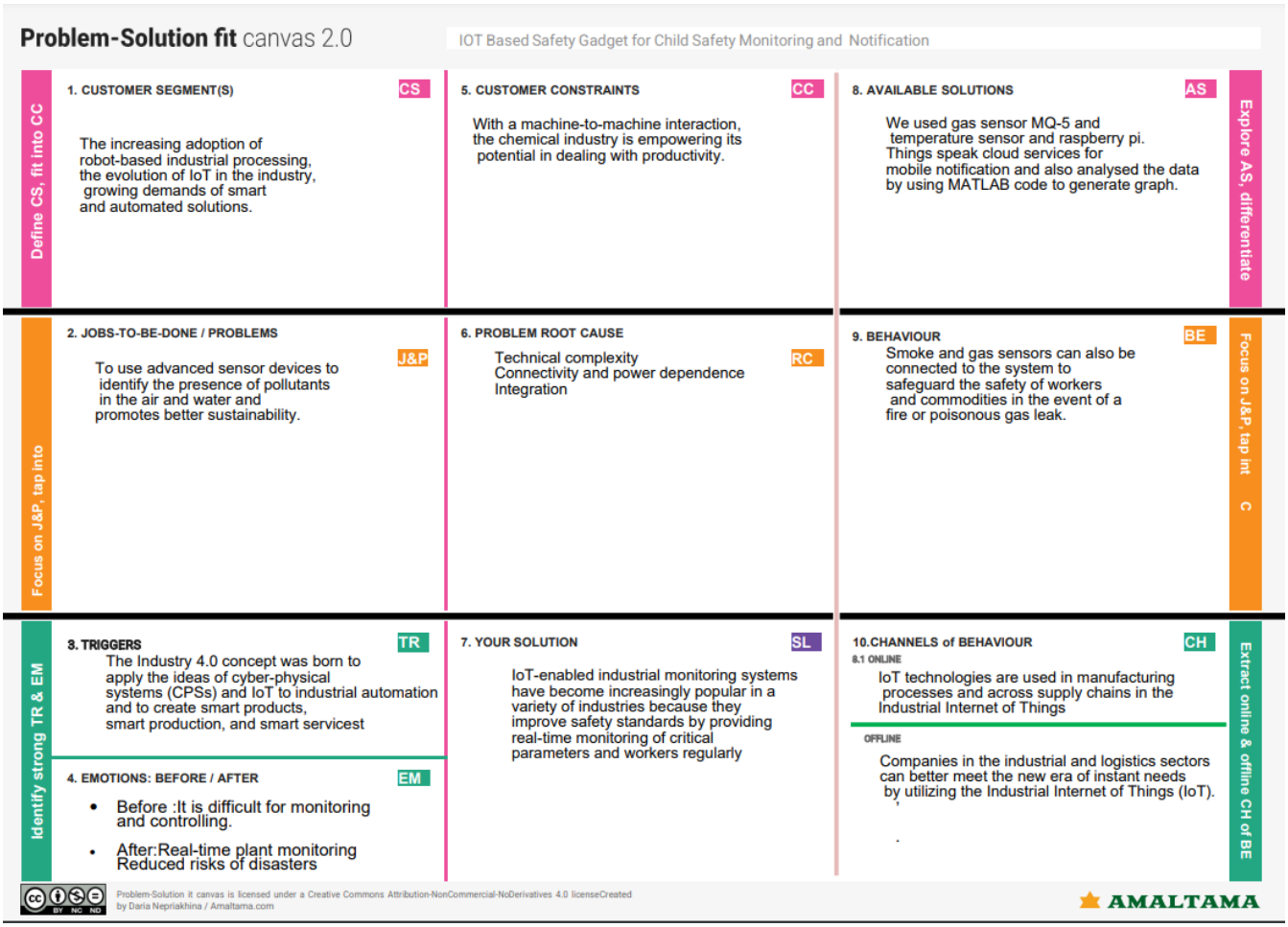
SUDIP MISRA Published on "IEEE International conference 2022" .The Industrial Internet of Things (IIoT) connects all of the actors who are involved in an industrial environment in order to increase operational and management efficiencies. Data can travel over a communication network that is frequently complicated and heterogeneous thanks to this bridging. It allows for prompt decision-making that has an impact on a variety of organisational areas, including business, operations, maintenance, safety, stock, and logistics. Despite the abundance of works in the IIoT field addressing the aforementioned aspects, very few works address safety in industries. Industrial safety is a crucial area that has room for improvement in the context of IIoT-based solutions for industrial safety management, especially whenever it is linked to human safety. We give a thorough overview of through this examination of of the industrial safety problems that are common. The safety aspects of several IIoT application domains, including healthcare, transportation, manufacturing, and mining, are then categorised and thoroughly examined. Finally, we review the research gaps in several fields and suggest new lines of investigation. To secure people's safety and reduce hazards, we explore a variety of technologies, prototypes, systems, models, methodologies, and applications. This research's main goal is to investigate, synthesise, and acknowledge the applicability of previous studies to safety management using the IIoT.

IDEATION AND PROPOSED SOLUTION:

1. Empathy Map



2. Proposed Solution Fit:



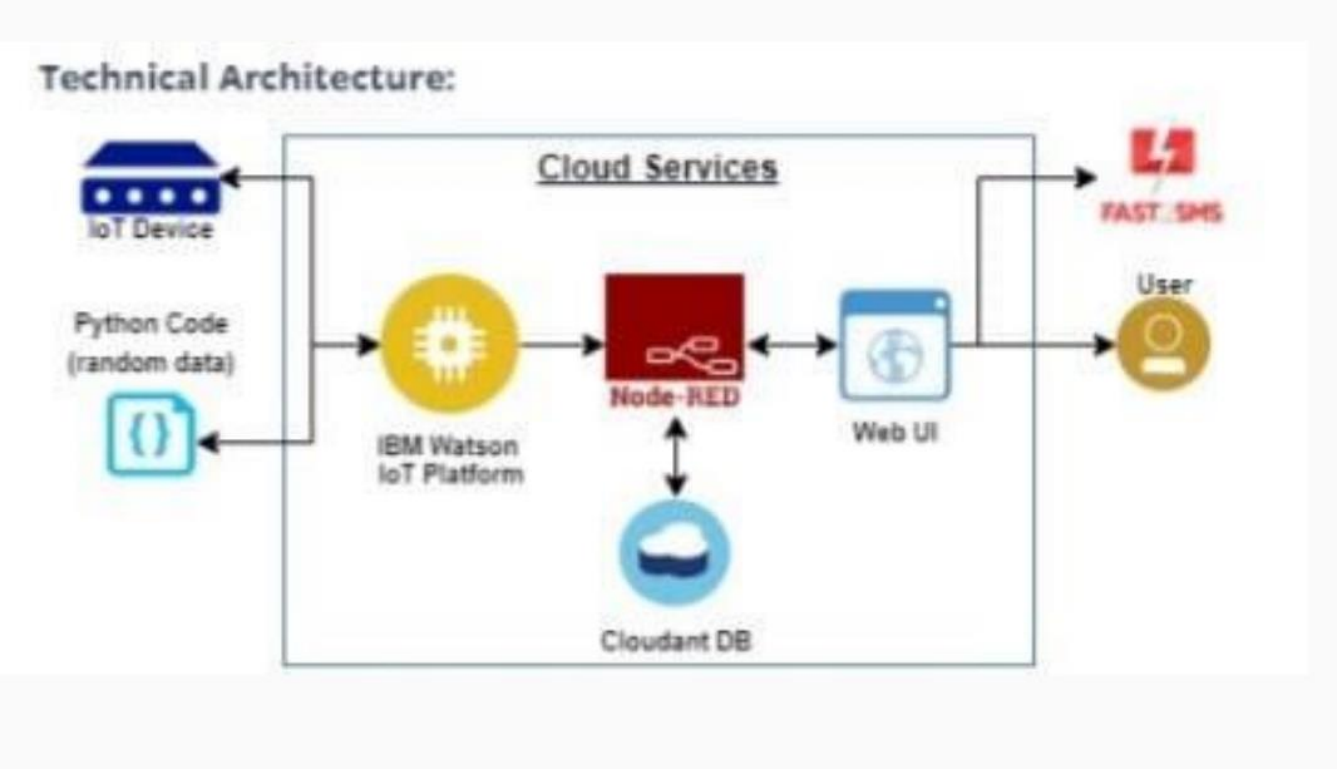
3. Proposed Solution

S.No.	Parameter	Description
•	Problem Statement (Problem to be solved)	To monitoring industrial parameter monitoring and power consumption control.
•	Idea / Solution description	The industrial devices are controlled using cloud server which alerts the admin about uneven conditions using Bluetooth.In short automation has become an important term, whether at home or the industries.
•	Novelty / Uniqueness	Relays can switch AC and DC, transistors can only switch DC. Relays can switch high voltages, transistors cannot.
•	Social Impact / Customer Satisfaction	Relays are bulkier than transistors for switching small currents. Relays cannot switch rapidly (except reed relays), transistors can switch many times per second. Relays use more power due to the current flowing through their coil.
•	Business Model (Revenue Model)	Industrial parameters are temperature, humidity, gas and fire. These parameters are continuously

		monitored by ARM7 and send values to the android app using WIFI module.If any value goes above the threshold then it send SMS to supervisor and start the output devices.
•	Scalability of the Solution	We implement a GSM module based Embedded System concept. This Arduino based embedded system concept is to monitor and control industrial parameter such as (temperature, gas, fire,humidity) and inform to responsible person.

SOLUTION ARCHITECTURE:

Solution architecture includes sensors, cloud services, apps, databases and hardware modules.



These components include a Temperature and humidity sensor, IBM Watson platform, Rasberry pi, Cloudant DB, Web Browser, SMS services and mobile applications.

Functional Requirements:

Following are the functional requirements of the proposed solution.

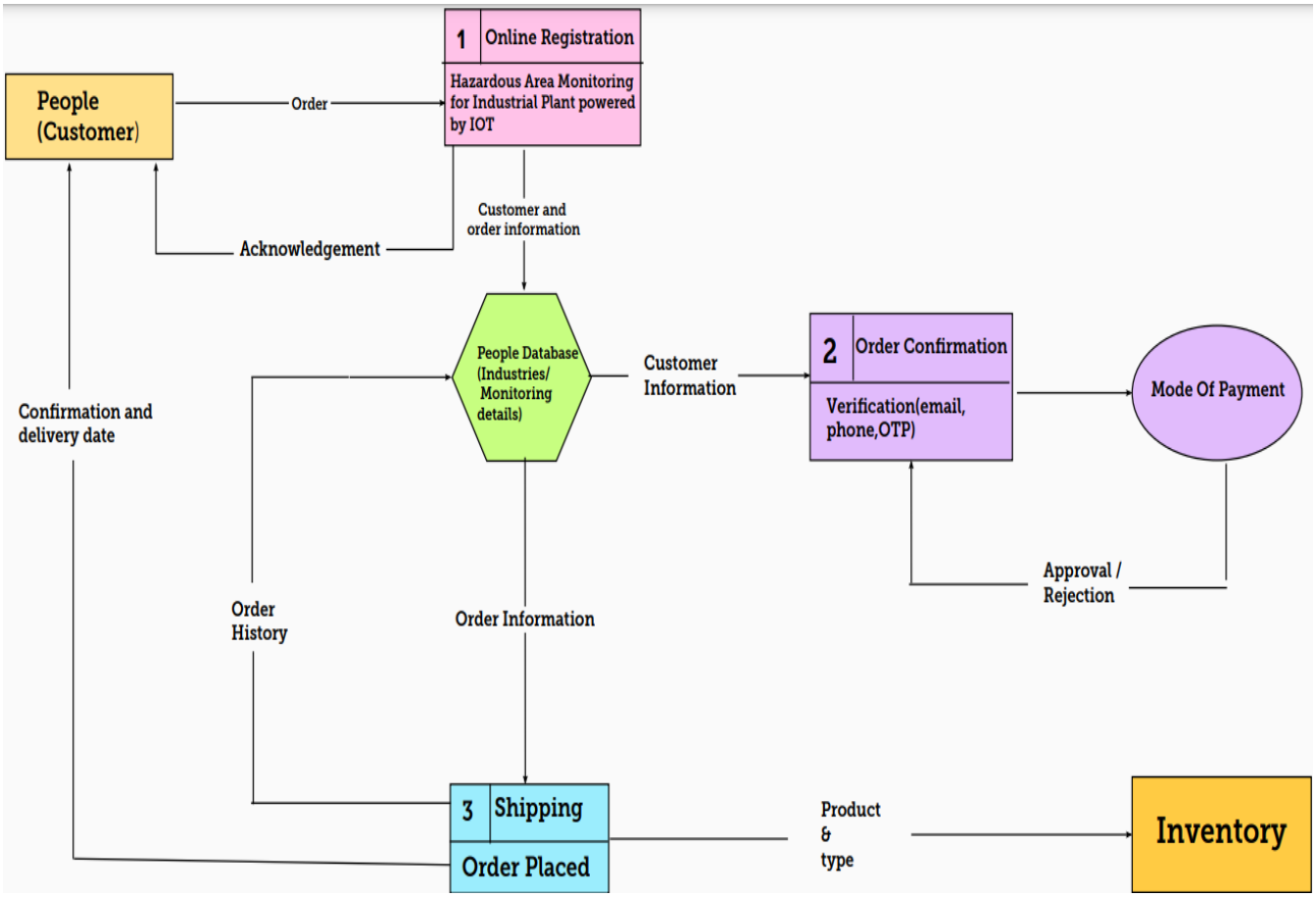
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	**User Requirements**	Air protection pH ,Humidity and Temperature
FR-2	**User Registration**	Manual Registration Registration through webpage Registration through Form Registration through Gmail
FR-3	**User Confirmation**	Confirmation Via Email Confirmation Via OTP Confirmation Via Phone
FR-4	**Payment Options**	Cash on Delivery Net Banking/UPI Credit/Debit/ATM Card
FR-5	**Product Delivery and Installation**	Door Step delivery Free Installation
FR-6	**Product Feedback**	Through GoogleForms Through Phone calls Through Webpage

Non-functional Requirements:

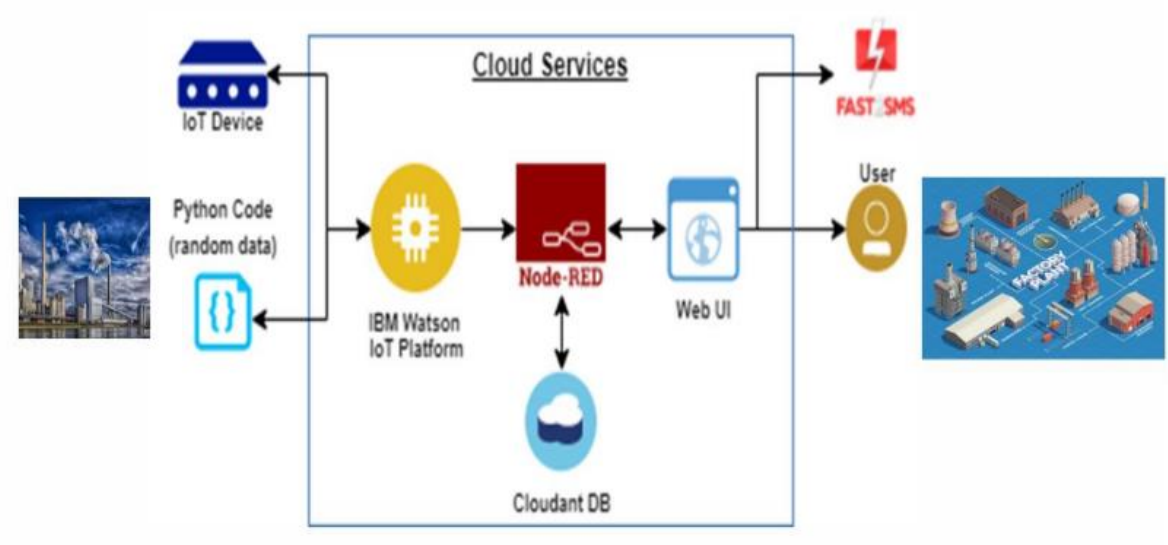
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Have a clear and self-explanatory manual. Easier to use Even an illiterate people have to use the product without any difficulties
NFR-2	**Security**	Application has to be secured with 2 step authorisation Passwords and passkeys will be assigned as per the users need.
NFR-3	**Reliability**	Hardware requires a regular checking and service Software may be updated periodically Immediate alert is provided in case of any system failure
NFR-4	**Performance**	The application must have a good user interface It should have a minimal energy requirement It has to save air and environment
NFR-5	**Availability**	All the features will be available when the user requires. It depends on the need of the people and the customization the user has done.
NFR-6	**Scalability**	The product has to cover all the space of industrial area irrespective of the size or area of a field.

Dataflow Diagram:



TECHNOLOGY (Stack and Architecture):



Hazardous area monitoring for industrial plant.

Table-1: Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Web UI, Mobile App	Angular Js / React Js etc.
2.	Application Logic-1	Logic for a process in the application	Python
3.	Application Logic-2	Logic for a process in the application	IBM Watson STT service
4.	Application Logic-3	Logic for a process in the application	IBM Watson Assistant
5.	Cloud Database	Database Service on Cloud	IBM Cloudant etc.

6.	File Storage	File storage requirements	IBM Block Storage
7.	Machine Learning Model	Purpose of Machine Learning Model	Object Recognition Model, etc.
8.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	Local, Cloud Foundry, Kubernetes, etc.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	The open-source frameworks used	Chrome
2.	Security Implementations	The security / access controls implemented, use of firewalls etc.	IBM cloud Encryptions
3.	Scalable Architecture	The scalability of architecture (3 – tier, Micro-services)	IBM cloud architecture
4.	Availability	The availability of application (e.g. use of load balancers, distributed servers etc.)	Web application can even be used by the workers in the industry
5.	Performance	The performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	Since the web application is high efficient, it can be used by the workers irrespective of time.

Sprint Schedule and Estimation

Sprint	Functional Requirement (Epic)
Sprint-1	Registration
Sprint-1	
Sprint-1	Login
Sprint-2	Post Job
Sprint-2	Job Search
Sprint-3	Apply

Sprint-3	Send Confirmation	USN-7	Confirmation mail is sent from the respected company	4	High	Ahamed J Harish K, Infant Maria Roshan V, Gokulakrishnan R
Sprint-4	Dashboard	USN-8	As a user, I need to maintain the my actions in an application	6	High	Ahamed J Harish K, Infant Maria Roshan V, Gokulakrishnan R
Sprint-4	Recruiter Review	USN -9	As a recruiter, I must make the reviews appear onthe candidate's profile	3	High	Ahamed J Harish K, Infant Maria Roshan V, Gokulakrishnan R
Sprint-4	Chatbot	USN-10	As a user, I can interact with Watson Assistant to resolve my queries on skills to be learnt	1	Low	Ahamed J Harish K, Infant Maria Roshan V, Gokulakrishnan R

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

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	15	7 Days	24 Oct 2022	31 Oct 2022	15	31 Oct 2022
Sprint-2	15	7 Days	1 Nov 2022	07 Nov 2022	15	07 Nov 2022
Sprint-3	10	6 Days	08 Nov 2022	13 Nov 2022	10	13 Nov 2022
Sprint-4	10	6 Days	14 Nov 2022	20 Nov 2022	10	20 Nov 2022

Velocity:

Sprint-1 and Sprint-2

$$AV = \frac{Sprint\ duration}{Velocity} = \frac{15}{7} = 2.14$$

Sprint-3 and Sprint-4

$$AV = \frac{Sprint\ duration}{Velocity} = \frac{10}{6} = 1.6$$

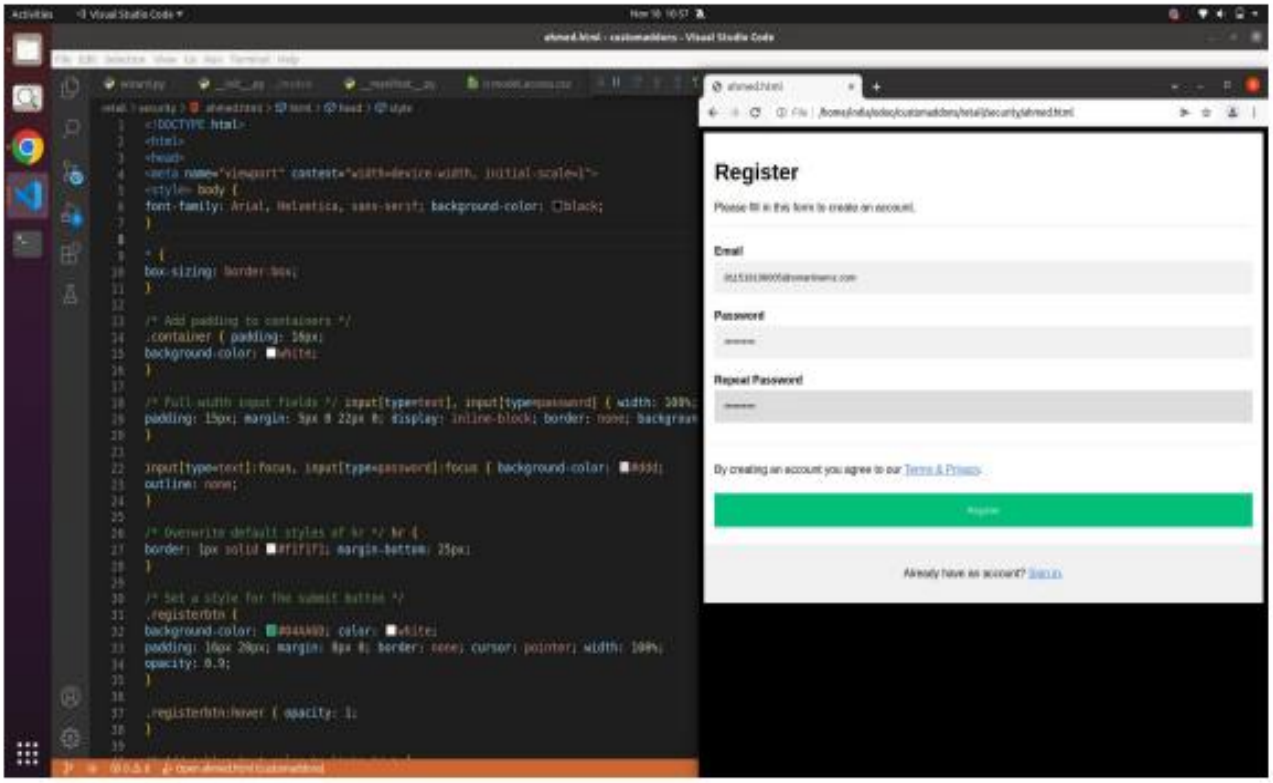
Project development phase:

CODE:

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {
  font-family: Arial, Helvetica, sans-serif;
  background-color: black;
}

* {
  box-sizing: border-box;
}

/* Add padding to containers */
.container
  { padding:
  16px;
  background-color: white;
}

/* Full-width input fields */
input[type=text], input[type=password]
{width: 100%;
  padding: 15px;
  margin: 5px 0 22px 0;
  display: inline-block;
  border: none;
  background: #f1f1f1;
}

input[type=text]:focus, input[type=password]:focus
  {background-color: #ddd;
  outline: none;
}
```

```
/* Overwrite default styles of hr */
hr {
  border: 1px solid #f1f1f1;
  margin-bottom: 25px;
}

/* Set a style for the submit button */
.registerbtn {
  background-color: #04AA6D;
  color: white;
  padding: 16px 20px;
  margin: 8px 0;
  border: none;
  cursor: pointer;
  width: 100%;
  opacity: 0.9;
}

.registerbtn:hover
  {opacity: 1;
}

/* Add a blue text color to links */
a {
  color: dodgerblue;
}

/* Set a grey background color and center the text of the "sign in" section */
.signin {
  background-color: #f1f1f1;
  text-align: center;
}
</style>
</head>
<body>

<form action="/action_page.php">
  <div class="container">
    <h1>Register</h1>
    <p>Please fill in this form to create an account.</p>
    <hr>

    <label for="email"><b>Email</b></label>
    <input type="text" placeholder="Enter Email" name="email" id="email" required>

    <label for="psw"><b>Password</b></label>
    <input type="password" placeholder="Enter Password" name="psw" id="psw"
required>
```

```
    <label for="psw-repeat"><b>Repeat Password</b></label>
    <input type="password" placeholder="Repeat Password" name="psw-repeat"
id="psw-repeat" required>
    <hr>
    <p>By creating an account you agree to our <a href="#">Terms &
Privacy</a>.</p>

    <button type="submit" class="registerbtn">Register</button>
  </div>

  <div class="container signin">
    <p>Already have an account? <a href="#">Sign in</a>.</p>
  </div>
</form>

</body>
</html>
```



Sprint 2:

CODE:

```
#IBM Watson IOT Platform
#pip install wiotp-sdk
import wiotp.sdk.device
import time
import random
myConfig = {
    "identity": {
        "orgId": "hj5fmy",
        "typeId": "NodeMCU",
        "deviceId":"12345"
    },
    "auth": {
        "token": "12345678"
    }
}

def myCommandCallback(cmd):
    print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
    m=cmd.data['command']

client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()

while True:
    temp=random.randint(-25,135)
```

```
    hum=random.randint(0,100)
    myData={'temperature':temp, 'humidity':hum}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,
onPublish=None)
    print("Published data Successfully: %s", myData)
    client.commandCallback = myCommandCallback
    time.sleep(2)
client.disconnect()
```

SENSOR CODE:

```
#include <dht.h>

#define dht_apin A0                    // Analog Pin 0 is connected to DHT sensor
#define mqt_apin A1                    // Analog Pin 1 is connected to MQT 135 sensor
dht DHT;
int sensorValue;

void setup(){

  Serial.begin(9600);                  //Serial port to communicate with Python code
Serial1.begin(9600);                  //Serial port to communicate with Wearable
device through Bluetooth (HC-05)

  delay(500);                                  //Delay to let system boot
}

void loop(){

    DHT.read11(dht_apin);                      // read analog input pin 0(DHT11)
    sensorValue = analogRead(mqt_apin);        // read analog input pin 1(MQ135)

    //Send Humidity status to Python Code

    Serial.print("Current humidity = ");
    Serial.print(DHT.humidity);
    Serial.print("%  ");

    //Send Temperature status to Python Code

    Serial.print("temperature = ");
    Serial.print(DHT.temperature);
    Serial.println("C  ");

    //Send AirQuality sensor value to Python code

    Serial.print("AirQua=");
    Serial.print(sensorValue, DEC);
    Serial.println(" PPM");
```

```
    //Send signals to the Wearable

    Serial1.println("H T A");
    Serial1.println(DHT.humidity);
    Serial1.println(DHT.temperature);
    Serial1.println(sensorValue, DEC);

    delay(100);                                  // wait 100 milliseconds for next reading
}
```

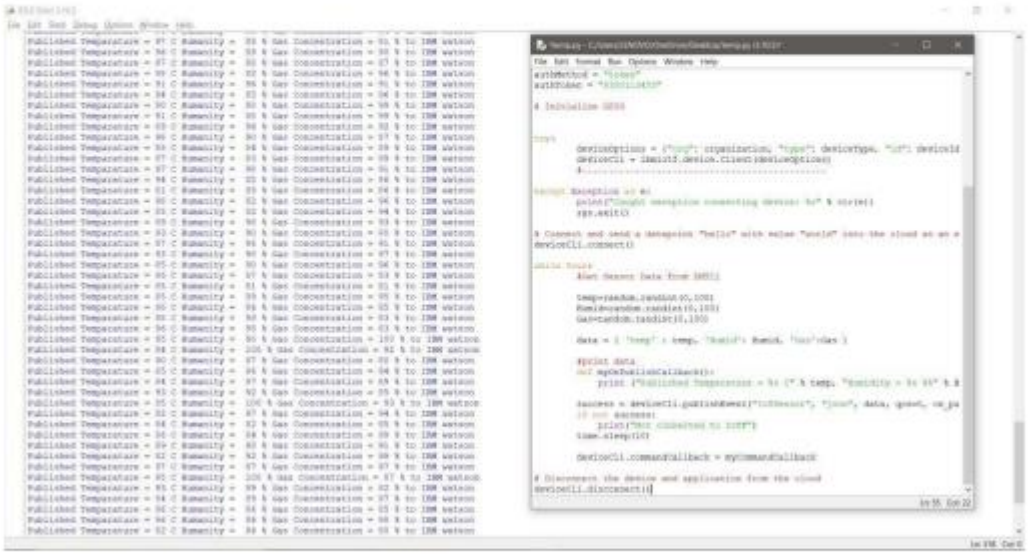
Sprint 3:

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


#Provide your IBM Watson Device Credentials
organization = "lcft5g" deviceType = "Final"
deviceId = "Hello" authMethod = "token"
authToken = "8300113450"



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)

      Gas=random.randint(0,100)


      data = { 'temp' : temp, 'Humid': Humid,'Gas':gas }

      #print        data        def

      myOnPublishCallback():

        print ("Published Temperature = %s C" % temp, "Humidity = %s %%" %
Humid, "Gas Concentration = %s"%Gas"to IBM Watson")


      success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,
on_publish=myOnPublishCallback) if

      not success:

        print("Not connected to IoTF")

      time.sleep(10)
```

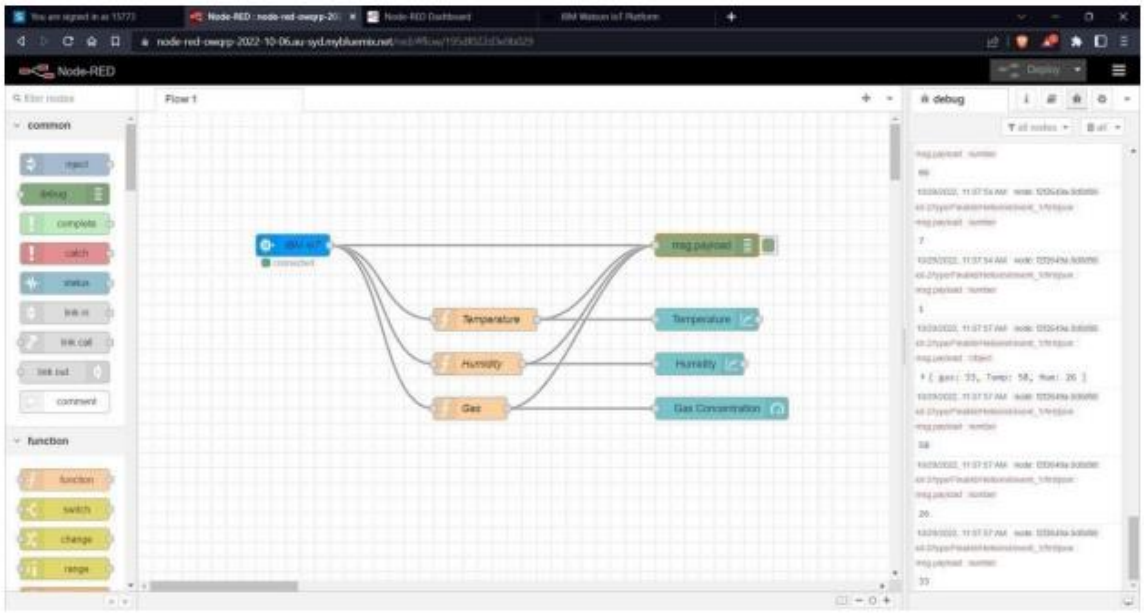
```
    deviceCli.commandCallback = myCommandCallback
# Disconnect the device and application from the cloud
deviceCli.disconnect()
```

Output:

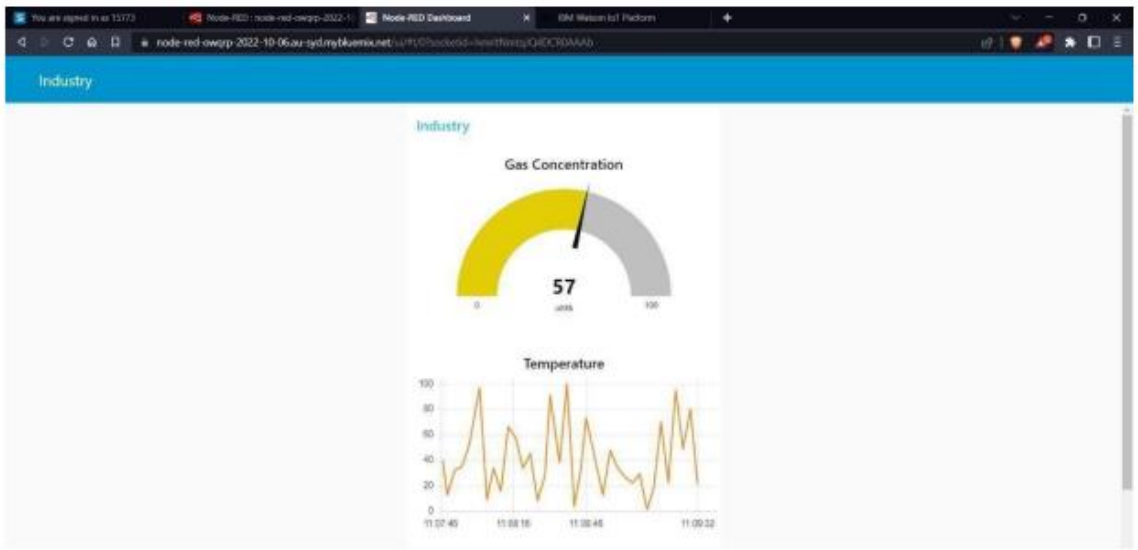


Sprint 4:

1. Data is brought to Node-RED



2. Data is displayed in Dashboard



DEVELOP THE PYTHON SCRIPT:

```
import wiotp.sdk.device
import time
import random
myConfig =
{ "identity":
{ "orgId": "6yafic",
"typeId": "Sprint1",
  "deviceId":"SprintID"
  },
 "auth": {
  "token": "sW(iQhEK*t)4!jgrjD"
 }
}

def myCommandCallback(cmd):
 print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
  m=cmd.data['command']

client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()
```

while True:

temp=random.randint(0,50)

heart=random.randint(60,100)

myData={'temperature':temp, 'heartrate':heart}

client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)

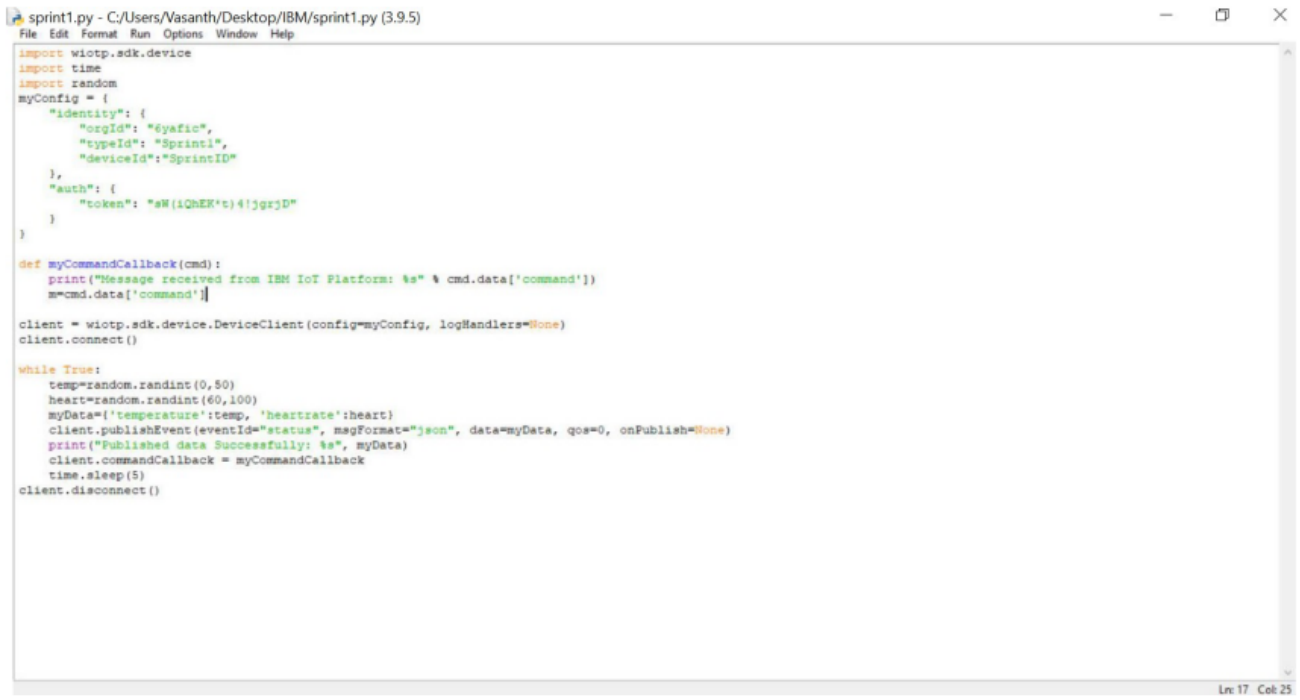
print("Published data Successfully: %s", myData)

client.commandCallback = myCommandCallback

time.sleep(5)

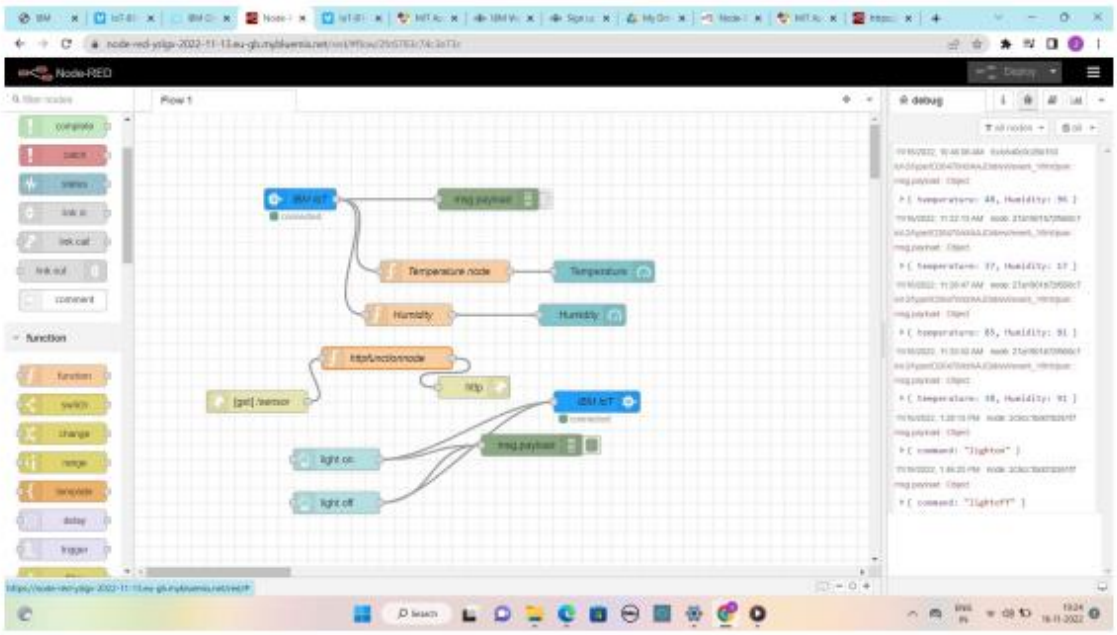
client.disconnect()

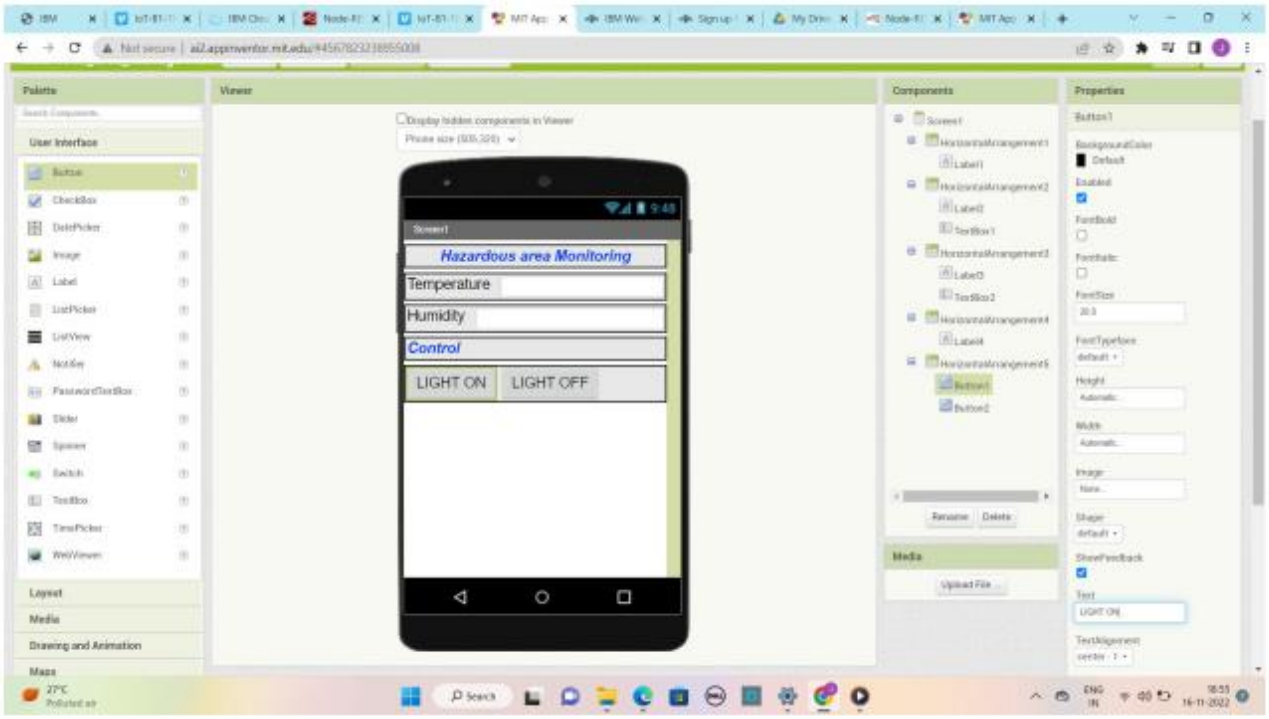
Publishing the python script

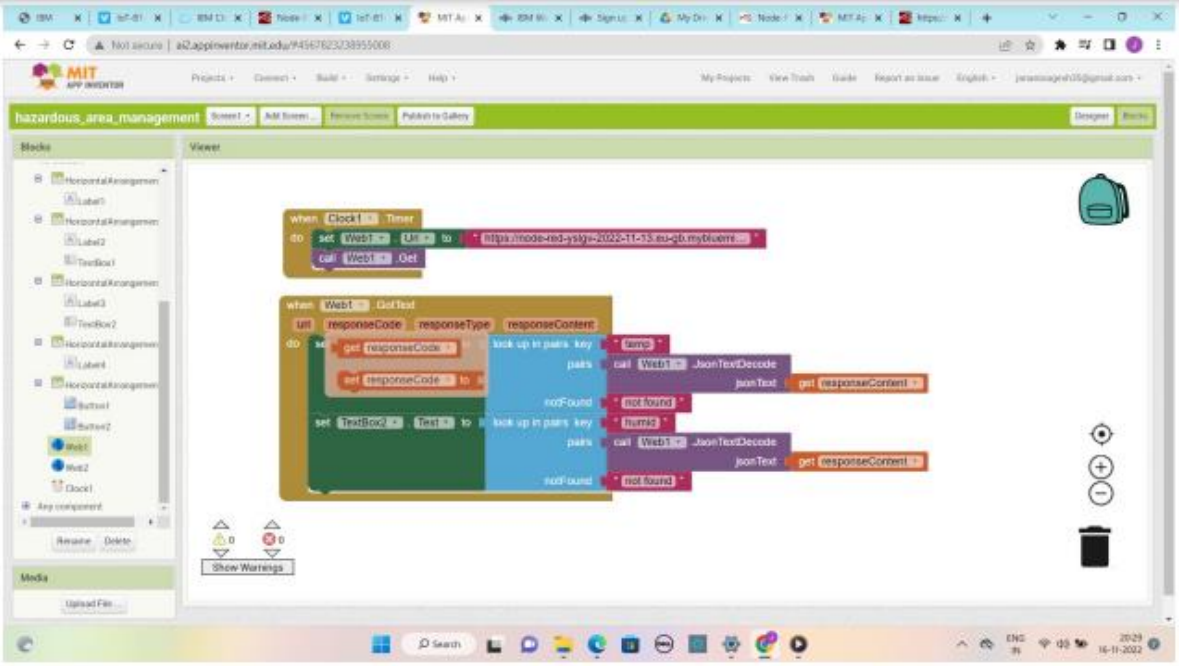


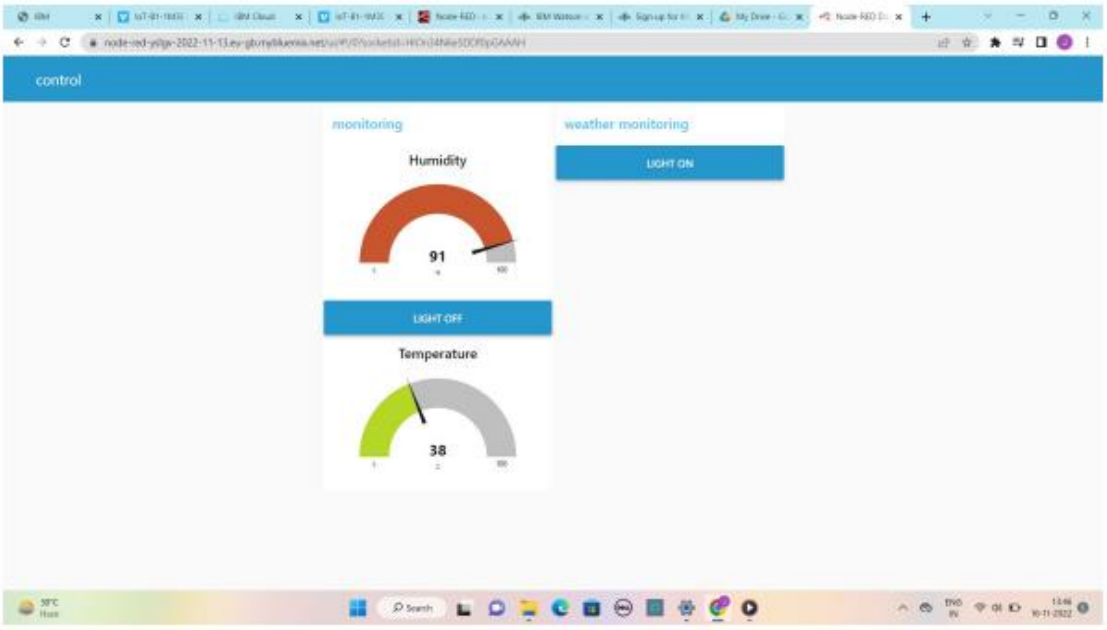
Develop a Web Application Using Node red Service:

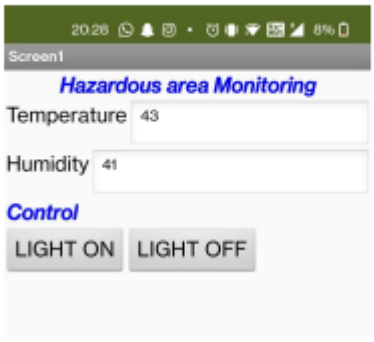
Create HTTP Request to communicate with Mobile App





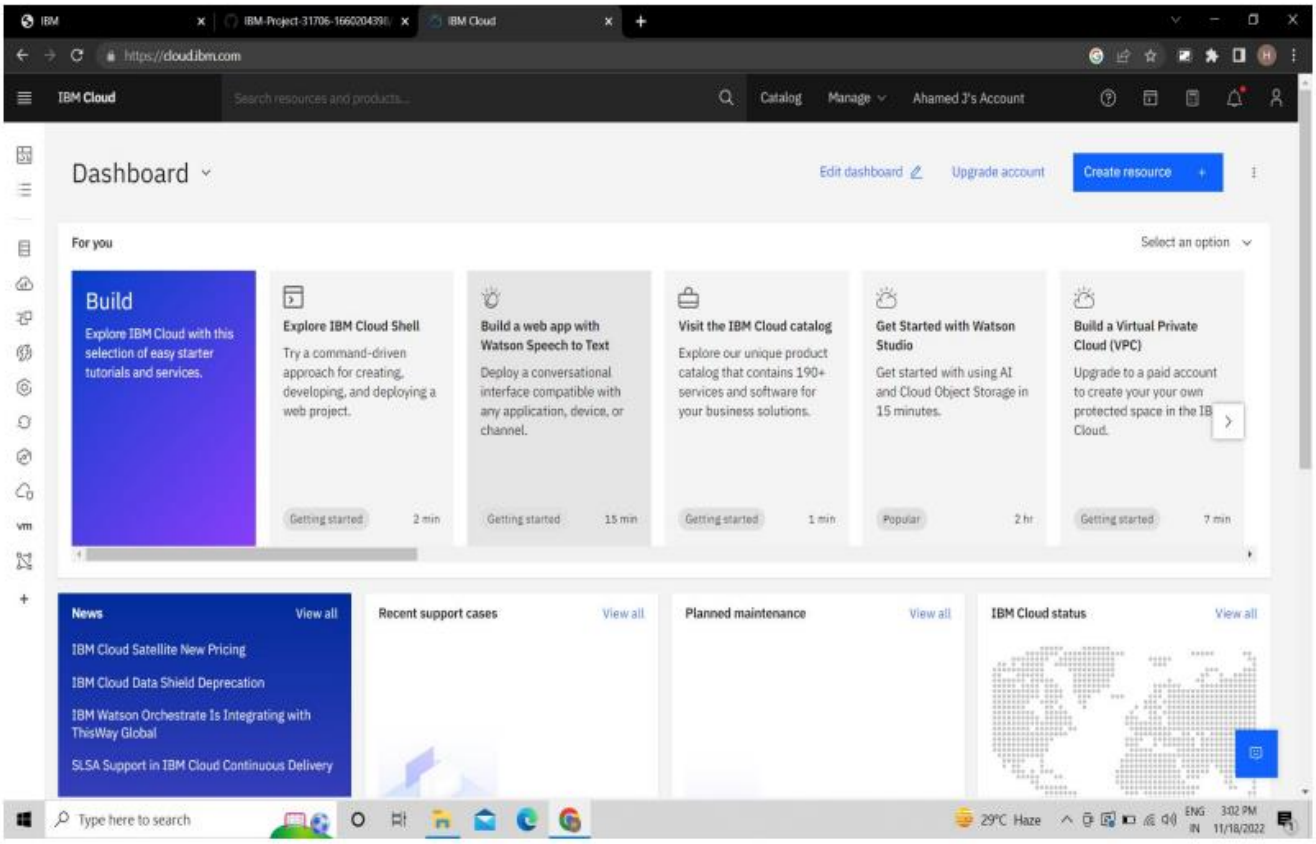

hazardous_area_management
Blocks
Viewer
Media
Show Warnings

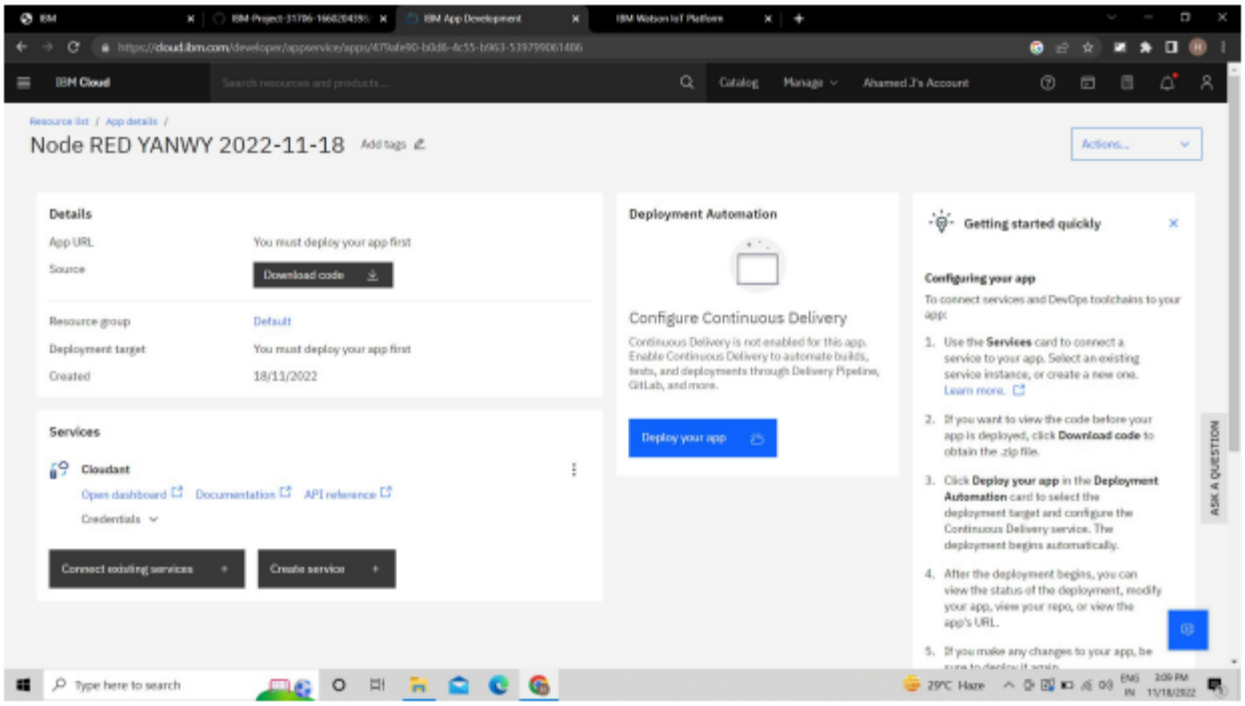

control
monitoring
Humidity
91
LIGHT OFF
Temperature
38
weather monitoring
LIGHT ON


Screen1
Hazardous area Monitoring
Temperature
43
Humidity
41
Control
LIGHT ON
LIGHT OFF

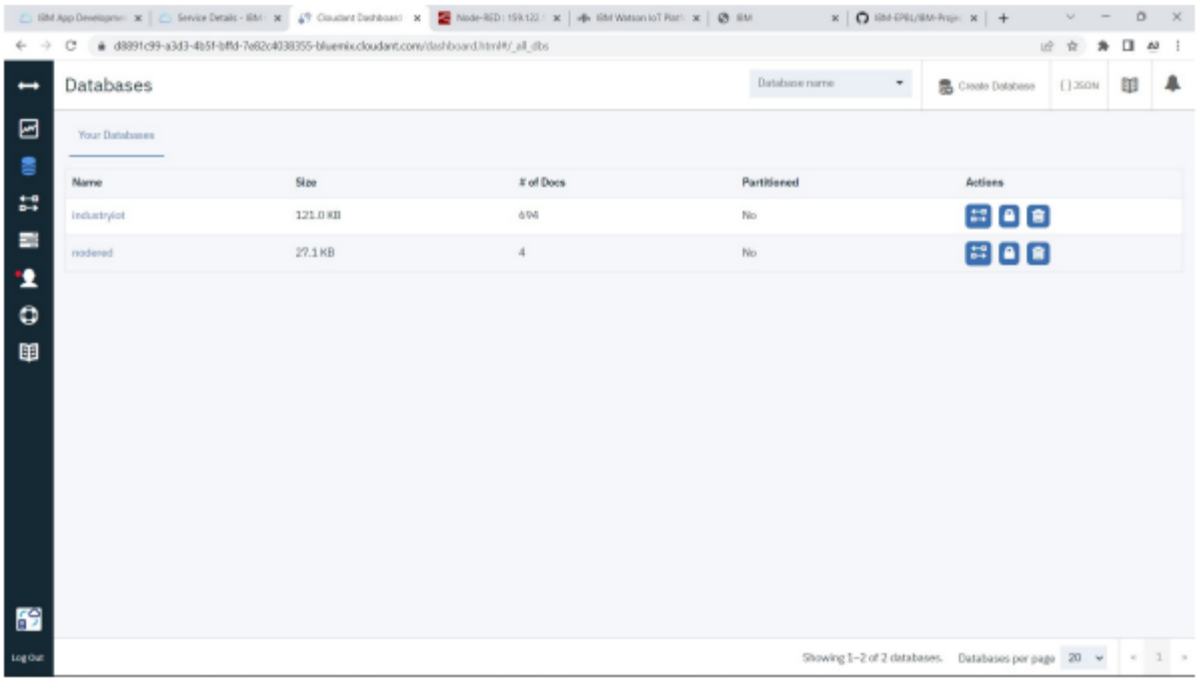
Prerequisites:

IBM IoT Watson Platform

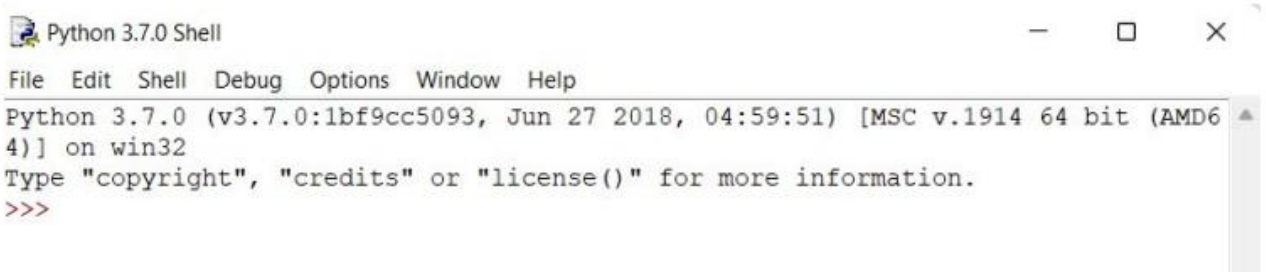




Cloudant DB



Prerequisites-software:



ADVANTAGES:

- IoT technology provides the most economic and budgeted solutions for the users.
- IoT is a fostering innovation.
- IoT powered level will be an improved supply chain.

DISADVANTAGES:

- Misuse of privacy and data.
- Expense.

.

CONCLUSION:

IoT is currently prevalent and gaining ground in many industries, with industrial applications being one of the most significant. The Internet of Things (IoT) offers a plethora of opportunities for enterprises to enhance productivity, lower costs, and boost sector growth. However, in industrial regions, danger monitoring and mitigation are frequently disregarded.

Therefore, the purpose of this project is to use IoT to actively monitor and analyse different aspects of a typical heavy industrial zone, such as temperature and levels of environmental gases. If the aforementioned parameters are more than the advised safe values. The system is able to monitor repeated issue alerts. Additionally, the data produced in real-time can offer crucial information about how well the work is progressing in various zones.

This system can be used in a variety of industrial settings, including heavy part production lines, underground factories, metal refineries, automaticwelding factories, and mining. It will assist in creating a secure and effective working environment in these locations while also opening up new avenues for enhancing their security measures.

FUTURE SCOPE:

There are the top applications of IIoT that aid the future Scope of Industrial

Monitoring:

1. Firstly, providing service engineers and manage remote access to industrial machines.

2. Secondly, allowing web-based virtual network connection to manage and observe HMI

functions on the IOT platform.

3. In addition, it offers predictive analytics for maintaining machines and identifying potential

Problems.

4. Most importantly, it controls, monitors, and manages data from multiple systems in various

locations. Simultaneously storing the collected data at a central cloud application. Hence, realtime machine data and analysis are easily accessible using industrial communication networks.

TEAM ID : PNT2022TMID10955

Github link: https://github.com/IBM-EPBL/IBM-Project-6173-1658824225