PROJECT BASED EXPERIENTIAL LEARNING PROGRAM (NALAIYA THIRAN)

HAZARDOUS AREA MONITORING FOR INDUSTRIAL POWER PLANT BY IOT

A PROJECT REPORT

Submitted by

KEERTHANA P (111719106070)

KAYALVIZHI V (111719106068)

KONDRAGUNTA JAHNAVI (111719106076)

KEESARAPALLI KEERTHI (111719106071)

TEAM ID :PNT2022TMID16047

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

R.M.K.ENGINEERING COLLEGE

(An Autonomous Institution)

R.S.M. Nagar, Kavaraipettai-601 206

NOVEMBER 2022

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

One of the most important parameters in industrial areas is temperature. In our project we are monitoring the temperature of the specified surrounding industrial areas with the help of the temperature sensors. We will implant the beacon devices all over the surroundings of the industrial area.

These beacon devices internally consist of the temperature sensor which is used for monitoring the temperature of that area. All these data from the beacon devices are collected by the IBM Cloud Watson Platform. All the workers, who are working in that industrial area will be provided with a wearable device.

In this wearable device the temperature of that particular area is displayed. The administrator will also be provided with a dashboard in which the temperature will be displayed. If the workers who were wearing a wearable device come near the beacon device, temperature will be shown to their respective wearable device.

If there is any abnormality in the temperature of the surroundings in that industrial area then a SMS will be sent to every worker mobile as an alert. The administrator will immediately take action in a faster way when he observes the abnormality of the temperature. Here takes action in the sense he will immediately move the workers to a safe place through a safe exit.

The temperature abnormality is a very critical condition which may lead to industrial explosions. In our project we are saving the lives of the workers from the sudden explosions due to the temperature.

1.2 PURPOSE

Hazardous area Monitoring describes the monitoring of machinery/equipment in classified areas to prevent catastrophic events from occurring. Electro-Sensors, Inc. hazard monitoring systems are designed to provide safety across all Industries.

This system provides safety by monitoring Temperature. This benchmark protection helps facilities closely monitor their most important assets, and stay ahead of maintenance projects. These systems have improved the safety and productivity of facilities and plants.

Oftentimes safety concerns arise after a catastrophic event occurs, the temperature sensors in this system can help you prevent these occurrences by proactively monitoring your machinery. Safety is reliant on employees and equipment, and hazard monitoring helps your employees stay safe by providing the confidence in your machinery that is needed in modern industrial locations.

Our sensors will simplify the safety process for your facility and provide the monitoring that has been accepted in industries. Whether you are in the Grain, Petrol, Mining, Processing, or another industry we will do our best to provide the monitoring that you and your facility deserve.

CHAPTER 2

LITERATURE SURVEY

2.1 EXISTING PROBLEM

Reporting systems about any accidents or incidents should be available all time and also be quicker.Manually monitoring the condition of the environment to detect any danger is time consuming.

Temperature of the industrial areas should be monitored for the sake of workers. High or low temperature causes many health issues to the workers or employees

People working in the hazardous areas are affected by many health problems.It is not safe to work continuously in those areas. Continuously monitoring the industrial areas temperature is difficult by manual. Evacuating all the workers at a time is difficult.

2.2 REFERENCES

[1]MohdFauzi, Othman, KhairunnisaShazali. “Wireless Sensor Network Applications’ study in environment monitoring system” 2012.

[2]Majid Bahrepour, NirvanaMeratnia, Paul Havinga. “Automatic re detection-a Survey from wireless sensor network perspective” 2008.

[3] U. Arun Ganesh, M. Anand, S. Arun, M. Gunaseelan and R. Karthik. ”Forest Fire Detection Using Optimized Solar – Powered Zigbee Wireless Sensor Networks” 2013.

[4] FalohunA.S., Oke A.O, Abolaji B.M., Oladejo O.E. "Dangerous Gas Detection using an Integrated Circuit and MQ-9" 2016.

[5] A.M.patki, Anjali V. Patil. "Raspberry Pi based industrial process monitoring by using wireless communication" 2017.

[6] Mr. Bharath, MrsSurvaMubeen. "Wireless industrial parameter monitoring using Raspberry pi 3". 2016.

[7] Mrs.Poonam, Prof. Yusuf Mulge. “Remote Temperature Monitoring using LM35”. 2013.

[8] Ganga, D., & Ramachandran, V. (2018). IoT-based vibration analytics of Electrical Machines. IEEE Internet of Things Journal, 5(6), 4538–4549. https://doi.org/10.1109/jiot.2018.2835724

[9] Dai, B. (2019). Design of a complex wind power generation parameter control system based on embedded control combined with the internet of things. Web Intelligence, 17(2), 131–139. https://doi.org/10.3233/web-190407

[10] Wang, X., & Cai, S. (2020). An efficient named-data-networking-based IOT Cloud Framework. IEEE Internet of Things Journal, 7(4), 3453–3461. https://doi.org/10.1109/jiot.2020.2971009

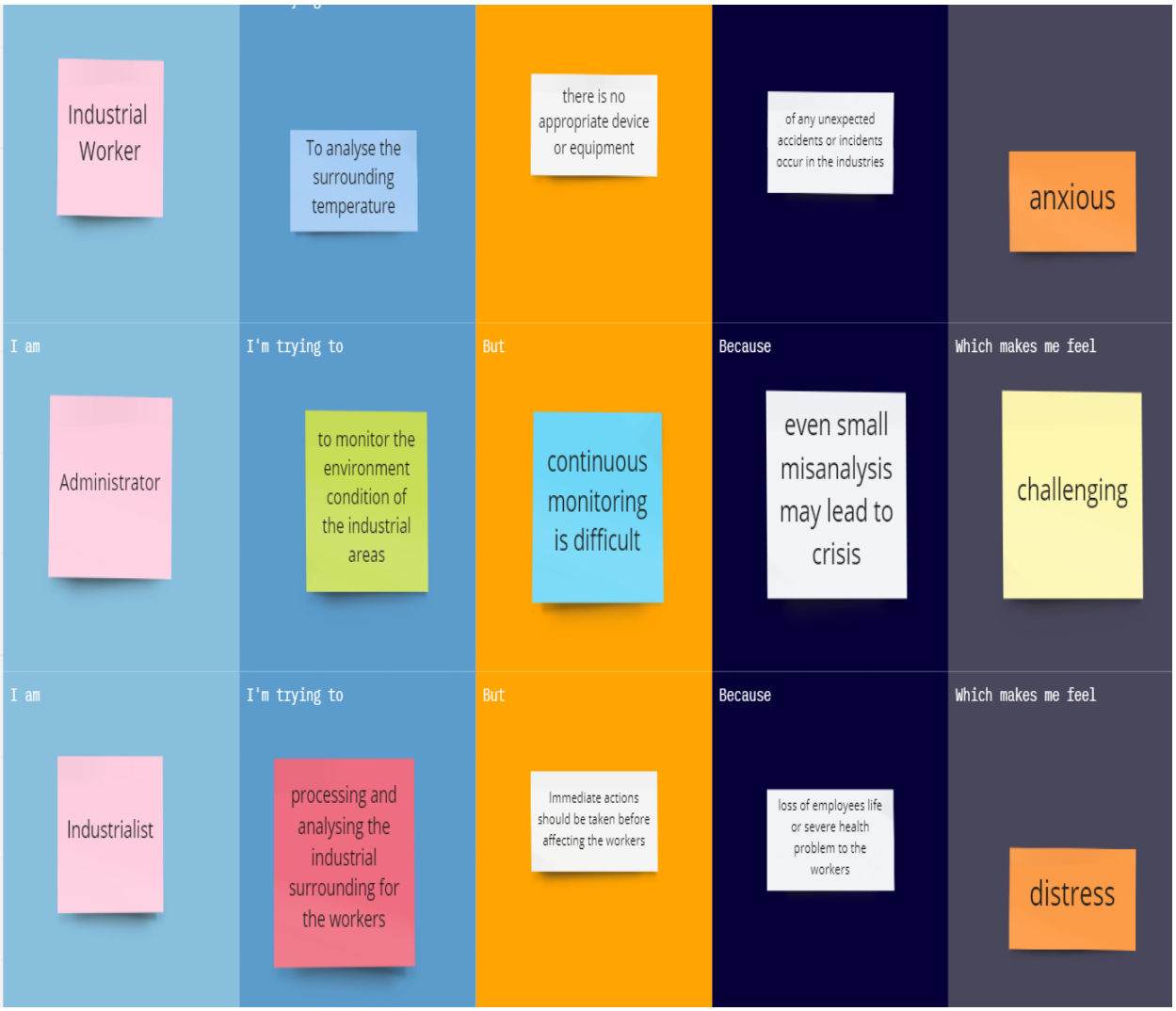
[11] Saha, S., & Majumdar, A. (2017). Data Centre temperature monitoring with ESP8266 based wireless sensor network and cloud based dashboard with Real Time Alert System. 2017 Devices for Integrated Circuit (DevIC). https://doi.org/10.1109/devic.2017.8073958

[12] Chawla, Y. P. (2022). Wi-Fi Computing Network empowers Wi-Fi Electrical Power Network. Cloud Computing Enabled Big-Data Analytics in Wireless Ad-Hoc Networks, 49–64. https://doi.org/10.1201/9781003206453-4

[13] Lee, C.-H., Lee, H.-S., & Kim, S.-K. (2017). A study on response characteristics of photoelectric type smoke detector chamber due to dust and wind velocity. Fire Science and Engineering, 31(1), 50–57. https://doi.org/10.7731/kifse.2017.31.1.050

[14] Luampon, R., & Charmongkolpradit, S. (2019). Temperature and relative humidity effect on equilibrium moisture content of cassava pulp. Research in Agricultural Engineering, 65(No. 1), 13–19. https://doi.org/10.17221/112/2017-rae

2.3 PROBLEM STATEMENT DEFINITION



PROBLEM STATEMENT (PS)	I AM (CUSTOMER)	I AM TRYING TO	BUT	BECAUSE	WHICH MAKES ME FEEL
PS-1	Industrial worker	Analyze the Surrounding temperature	There is no appropriate device or equipment	Of any unexpected accidents or incidents occur in the industries	anxious
PS-2	Administrator	To monitor the environment condition of the industrial areas	Continuous monitoring is difficult	Even small misanalysis may lead to crisis	challenging
PS-3	Industrialist	Processing and analysing the industrial surroundings for the workers	Immediate actions should be taken before affecting the workers	Loss of employee's life or severe health problem to the workers	distress

CHAPTER 3

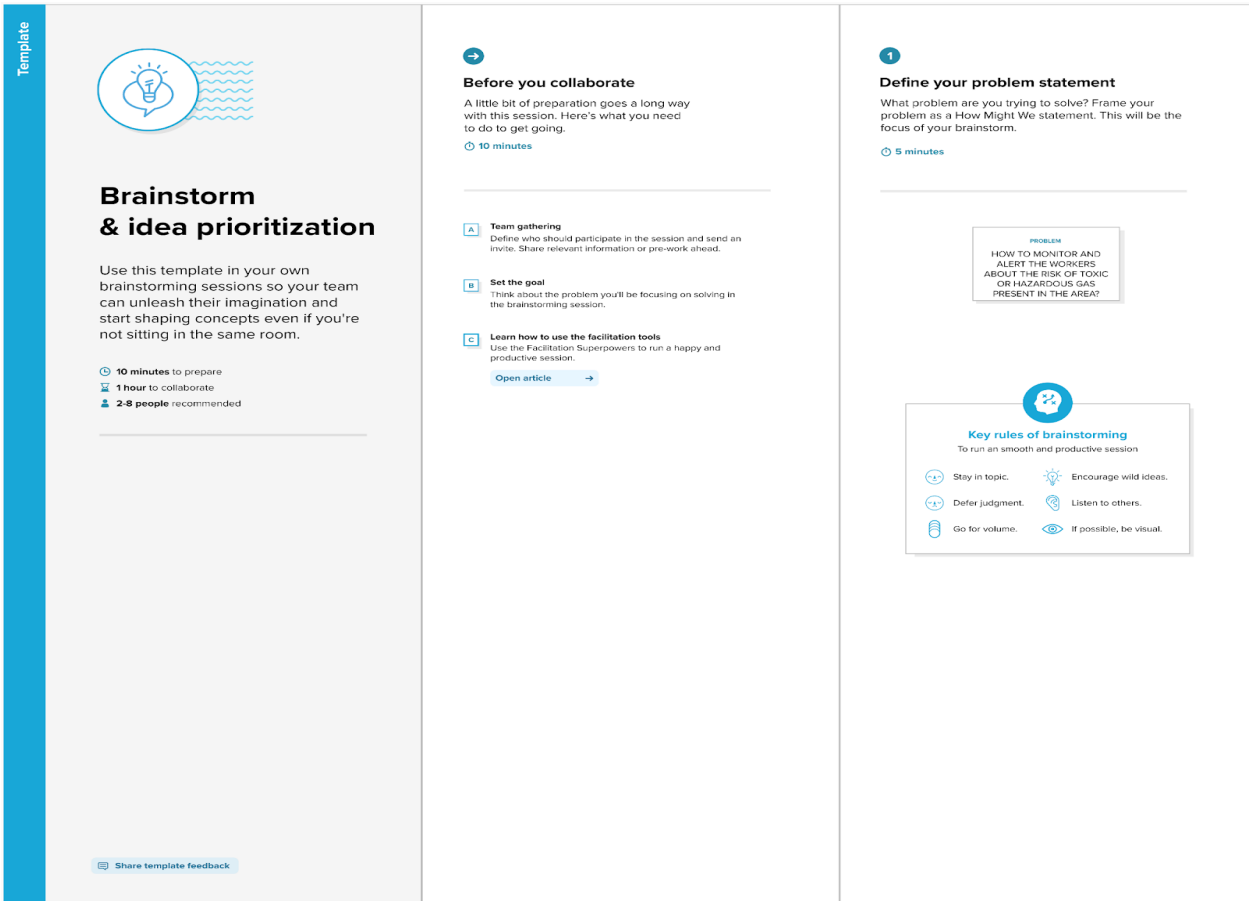
IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

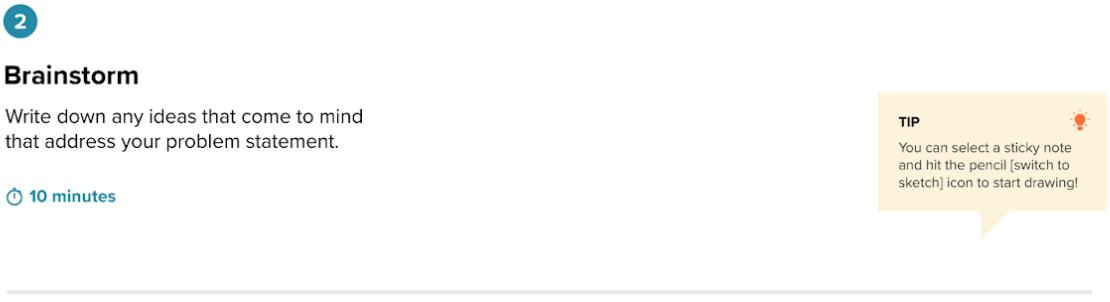
3.2 IDEATION AND BRAINSTORMING

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate,helping each other develop a rich amount of creative solutions.

STEP 1: Team Gathering, Collaboration and Select the Problem



STEP 2: Brainstorm, Idea Listing



KEERTHANA P

- USING SMART DEVICES TO MONITOR THE AREA
- WEB APP FOR MANAGING AND DISPLAYING DATA
- CONSTANT MONITORING 24/7
- ALERT SHOULD BE GIVEN PROMPTLY TO EVERYONE

KAYAL VIZHI V

- WATERPROOF WEARABLE DEVICE
- COMFORTBLE TO WEAR
- ADDITION OF BLUETOOTH BASED MONITORING
- PROVISION OF ALERT WHEN THE VALUE REACHES BEYOND THE THRESHOLD

KONDRAGUNTA JAHNAVI

- MONITORING AND MAINTENANCE CAN BE DONE BY THE ADMIN
- PROVISION OF SAFETY AND THE SECURITY
- PROVISION OF THE USER FRIENDLY PROCESS
- SENSORS WITH OPTIMAL SENSITIVITY

KEESARAPALLI KEERTHI

- DATABASE SHOULD BE MAINTAINED SECURELY
- COST EFFECTIVE OPERATION
- IMPROVED CUSTOMER SERVICE AND RETENTION
- DETECTION OF THE LOCATION SHOULD BE PRECISE

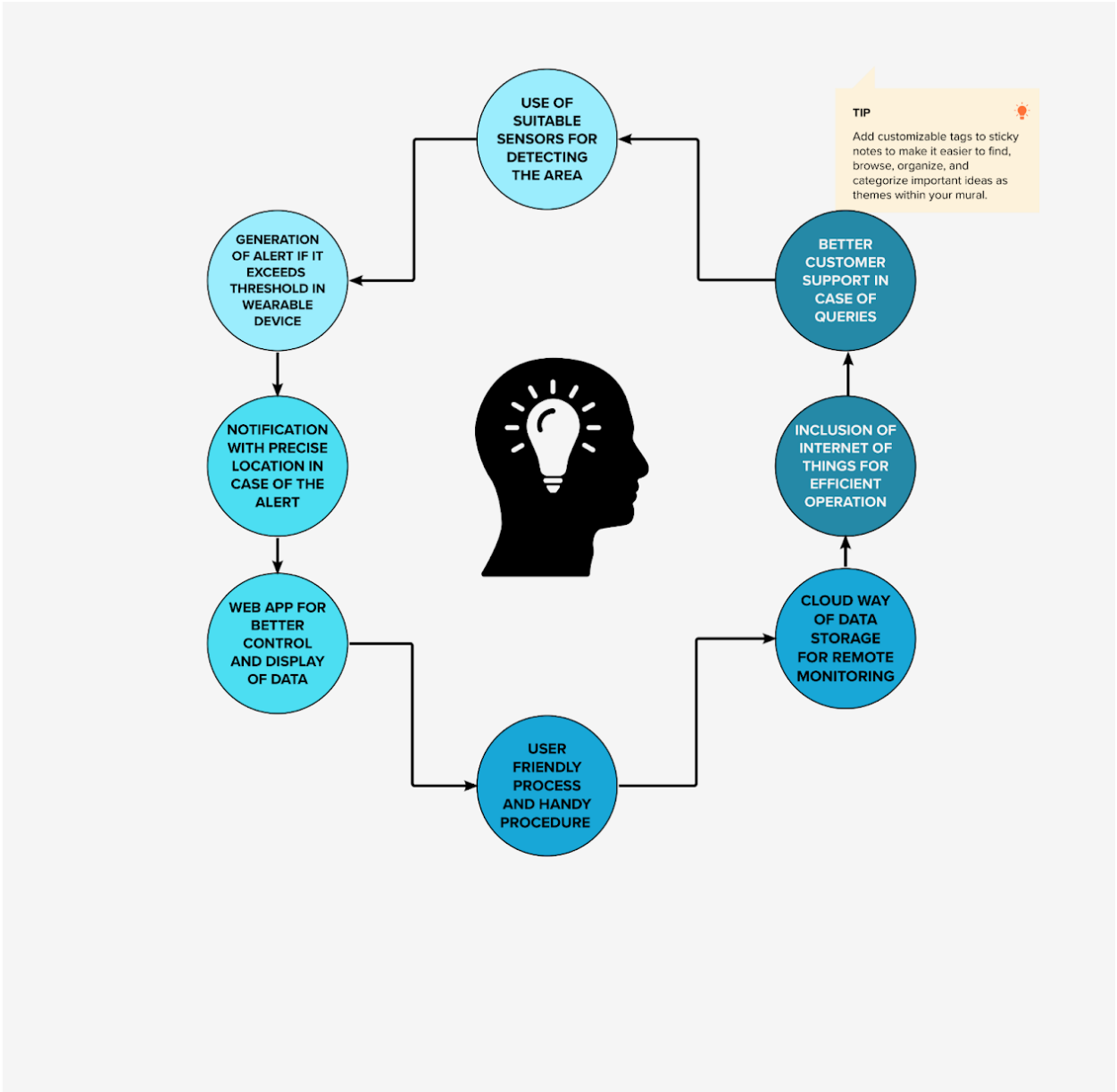
STEP 3: Grouping

3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes



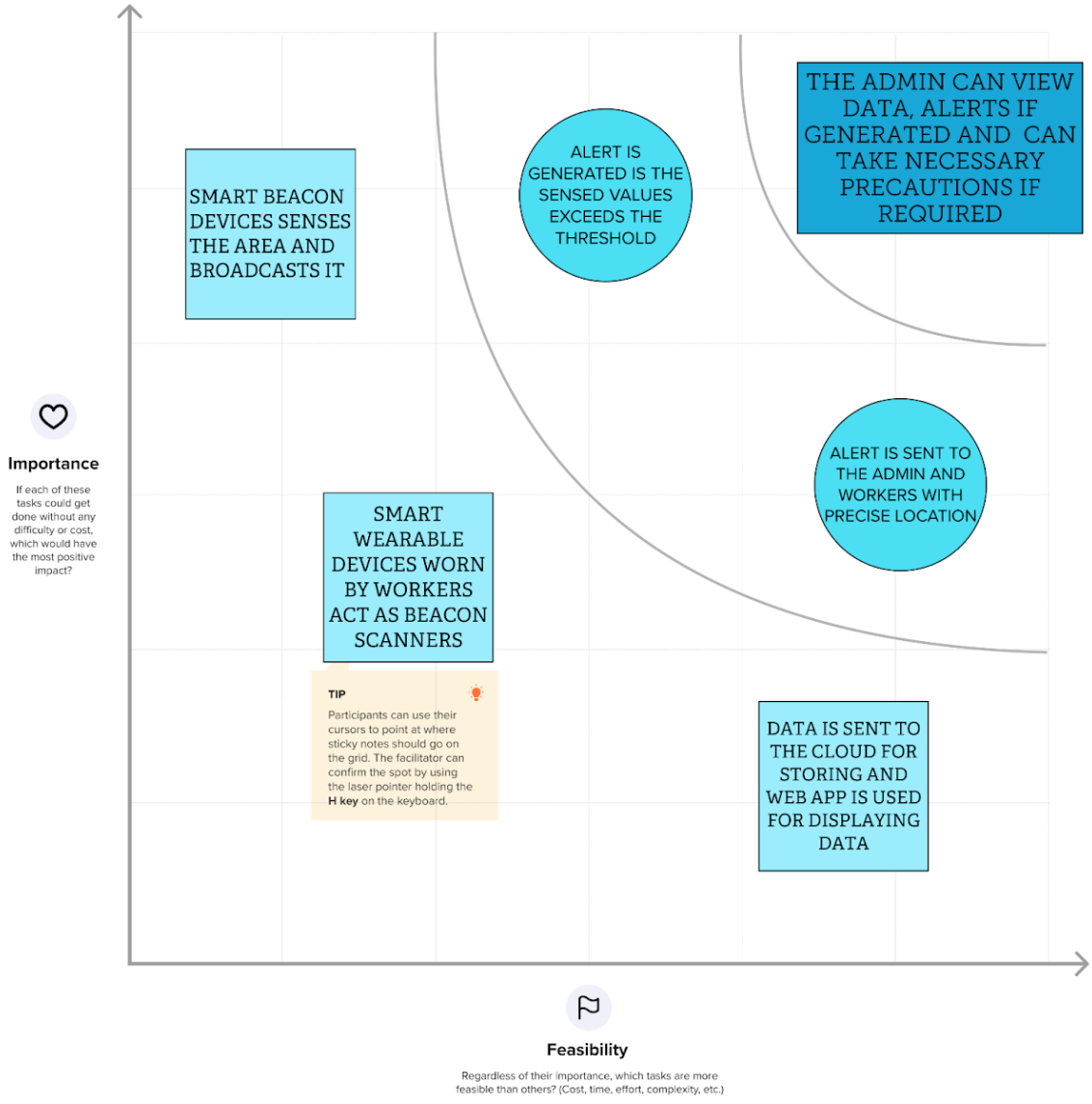
STEP 4: Idea Prioritization

4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

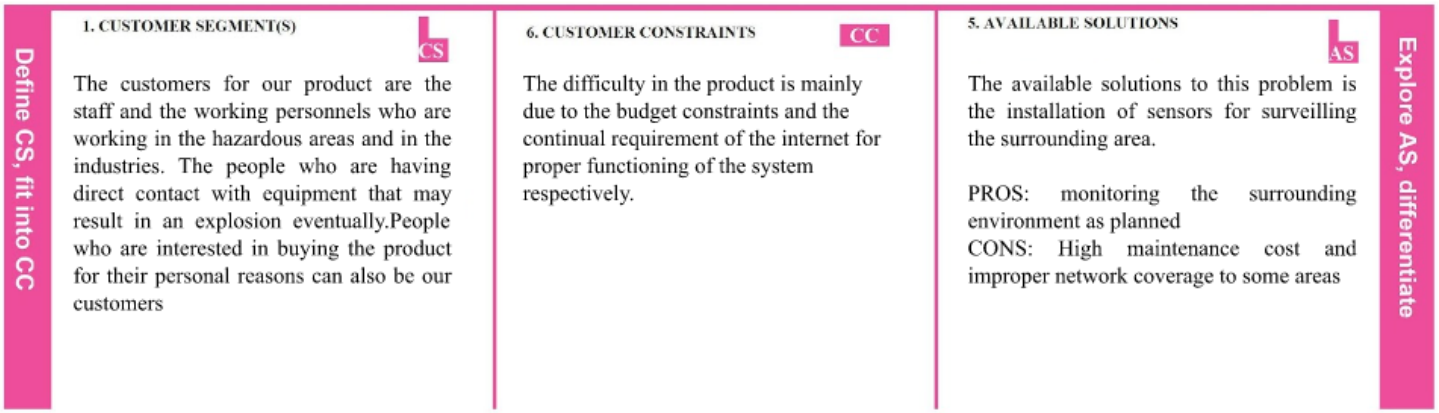


3.3 PROPOSED SOLUTION

S.NO	PARAMETER	DESCRIPTION
1	**Problem Statement (Problem to be solved)**	To monitor the surrounding area and measure the hazardous gas level and alert the working personnel in case of dangerous level of toxicity
2	**Idea / Solution description**	Implementation of a wearable device which can collect and store the data for future use. An alert is also sent when the temperature or the toxicity level is high thereby preventing the workers from the dangerous situations.
3	**Novelty / Uniqueness**	Our solution does not need the involvement of manual labor **1.** Preventing workers from getting exposed to the hazardous surroundings all the time **2.** Monitoring the surrounding all the time using different types of sensors **3.** Alerts are sent to both the workers and the admin promptly to prevent unnecessary situations **4.** Data is collected and stored in the cloud platform for future use simultaneously
4	**Social Impact / Customer Satisfaction**	**1.** Prevention of environmental and property damage **2.** Halting fatalities and injuries to working personnel **3.** Ensuring safety of the workers as well as people

		4. Comfortable and simple wearable device **5.** User-friendly solution
5	**Business Model (Revenue Model)**	**1.** It is an advanced technique where we can prevent the lives of the workers at low cost with minimum human intervention **2.** Its accuracy will be high since it involves measurement using machines and hence maximum prevention can be achieved
6	**Scalability of the Solution**	**1.** Since our product is a wearable device, it can be supplied to as many people working in the surrounding environment **2.** It is a solution involving simple concepts hence we can change the system as per our requirements **3.** Our system is flexible and can adapt to any type of the environments **4.** We can have many devices at comparatively lower cost than other systems

3.4 PROBLEM SOLUTION FIT



Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS — J&P

The job that needs to be done in order to produce our product is that we need to sense and obtain the values of the various surrounding parameters and then process it to check the danger level. We also need to alert the workers and admin in case of any emergency as soon as possible and store the data in cloud storage

9. PROBLEM ROOT CAUSE — RC

Unexpected changes in the composition of the materials in the hazardous area leading to fire explosions can be a root cause of our problem Manual monitoring can also cause issues due to negligence.

7. BEHAVIOUR — BE

The workers and admins will be provided with information about the surroundings all the time.They can check it using their smart device at any time.The concerned person will take necessary actions in case of emergency.They can take a look at the history of data using cloud storage with the help of a web application.

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS — TR

Fear of human loss and financial loss. The reason that wokers cannot predict the cause of explosions in the industries beforehand.

4. EMOTIONS: BEFORE / AFTER — EM

BEFORE: Absence of the awareness on the danger ahead—>approximate calculations and decisions—>Endangering their lives

AFTER: Knowledge on the various surrounding parameters—->accurate precision with decisions—>prevention of the lives

10. YOUR SOLUTION — SL

The hazardous area is integrated with smart beacon devices. All workers will be given smart wearable devices which will be acting as beacon scanners. Whenever a person goes near the beacon scanners he can view the various parameters on his device and if the temperature is high, he and admin will receive the alerts and the data is sent to the cloud.

8. CHANNELS OF BEHAVIOUR — CH

ONLINE: Acquiring online support from the company people Getting clearance on their queries.

OFFLINE: Customers will get the assistance in person and can see the resolving procedure in real time. They can also get to know more in offline mode.

Extract online & offline CH of BE

CHAPTER 4

REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

Following are functional requirements of the proposed solution

FR NO	FUNCTIONAL REQUIREMENT (Epic)	SUB REQUIREMENT (Story/Sub-Task)
FR-1	**Registration**	Registration of the user using their credentials
FR-2	**Confirmation**	Confirmation of the details using the verification link
FR-3	**User guidelines**	General guidelines for the user useful for the process initialization
FR-4	**Sensing parameters**	The smart beacon devices must be able to sense the parameters of the area
FR-5	**Location Identification**	The smart devices must be able to detect the location of the workers in the area precisely.
FR-6	**Displaying of data**	The wearable display should display the temperature of the working area to the concerned workers

FR-7	**SMS Intimation**	If the observed data for an area is found to be risky for the workers then they shall be notified via SMS
FR-8	**Data Sync**	The data has to be shared and synced to both the workers and the admin through cloud
FR-9	**Admin**	The admin should be informed about the alert through the dashboard presented to him

4.2 NON-FUNCTIONAL REQUIREMENTS

Following are non-functional requirement of the proposed solution

NFR NO	NON FUNCTIONAL REQUIREMENTS	DESCRIPTION
NFR-1	**Versatility**	The device has to be comfortable to wear and the sensors should sense and intimate the people promptly
NFR-2	**Dependability**	The device should function properly for a predetermined lifetime and should notify people in case of any fault

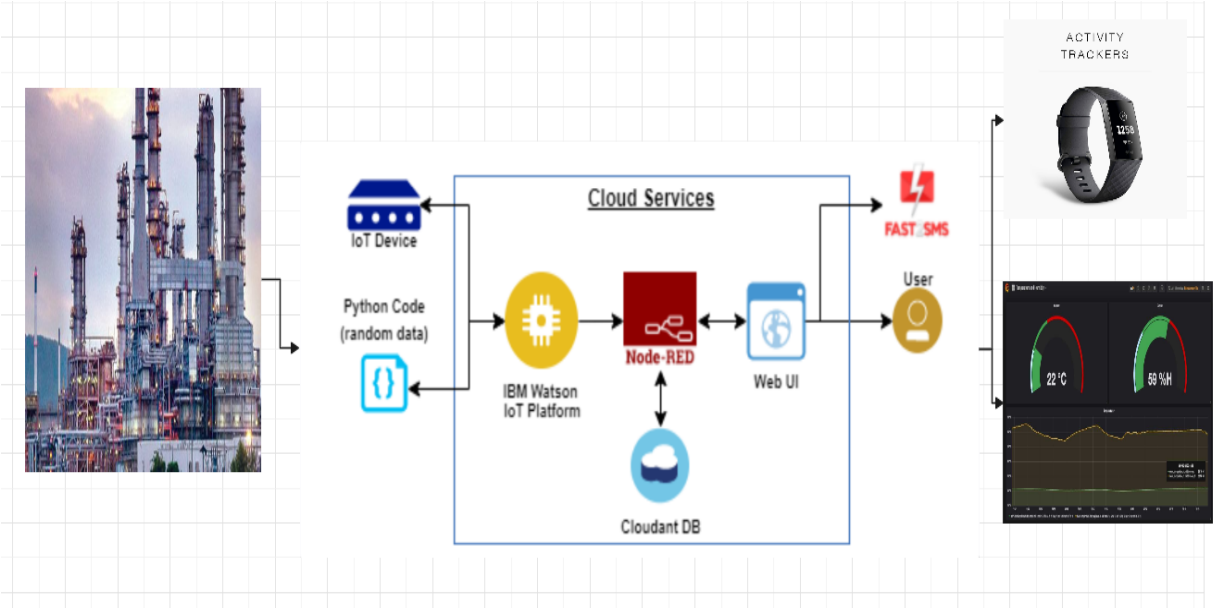
NFR-3	**Security**	The data that is shared and stored in the cloud should be secured from various attacks and malwares.
NFR-4	**Productivity**	The sensitivity of the products should be optimal and should update data spontaneously
NFR-5	**Accessibility**	The system should meet all possible demands of the user with ease of access for better functioning of the system
NFR-6	**Adaptability**	The product should entertain easy modification of the system if required as per user demands

CHAPTER 5

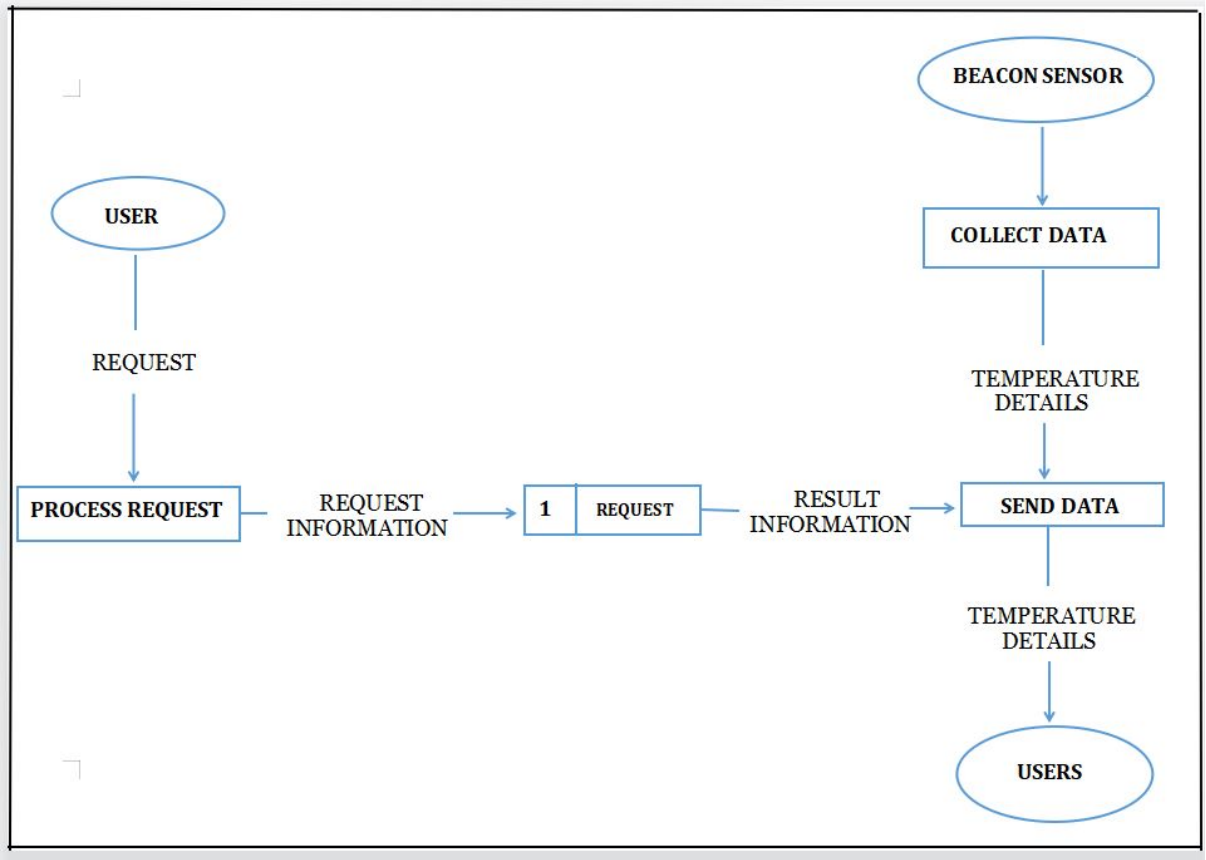
PROJECT DESIGN

5.1 DATA FLOW DIAGRAM

A data flow diagram(DFD) is a traditional visual representation of the information flows within the system.A neat and clear DFD can depict the right amount of the system requirement graphically.It shows how data enters and leaves the system,what changes the information,and where data is stored respectively.



DFD LEVEL 0 (INDUSTRY STANDARD)

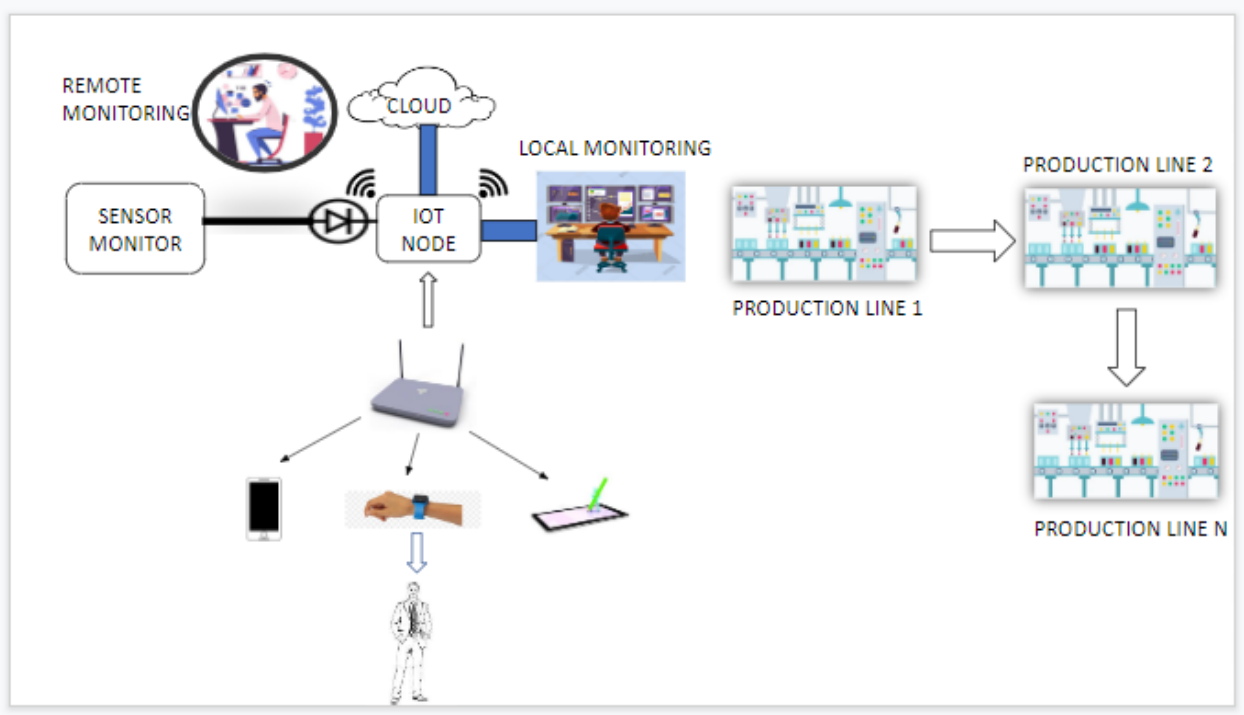


EXPLANATION:

- ❖ Write an appropriate python code for the designated IOT system
- ❖ IOT system is integrated with Watson cloud service for data collection
- ❖ IBM Watson IOT platform provides many services and connected to node red
- ❖ With the help of cloud service,temperature is displayed on the wearable device's dashboard
- ❖ Then the incidents in the industries can be avoided.

5.2 SOLUTION & TECHNICAL ARCHITECTURE:

Solution architecture is the process of developing a solution based on predefined processes, guidelines, and best practices with the objective that the developed solution.It helps us to track the problems easily and find solutions to it



- In this design, we develop an IOT based hazardous area monitoring system in industrial areas with the help of the environmental parameters since the environmental condition determines the living ability

- This project helps the employees in the industries to monitor the suitability condition of the environment to work peacefully without any concerns

- To initialize the project, first beacon devices are installed around the industrial areas, which contains sensor to monitor the temperature of the surrounding

- Temperature helps us determine the hazardous condition of the environment to avoid any dangerous incidents.Beacon devices records, process and analyzes the temperature of the surrounding
- These records are collected and stored in the cloud.The cloud services of IBM Watson Platform.The employees or workers are provided with wearable devices.Administrators are also present for remote monitoring
- The data in the cloud is sent to the wearable devices and the dashboard of the administrator.The data can be viewed if the workers move near to the beacon devices
- If the temperature exceeds the threshold temperature level,an alert message is sent to each worker through SMS and displayed in the dashboard.With the help of this,they can evacuate the areas before the occurrence of any incidents.

TECHNICAL ARCHITECTURE:

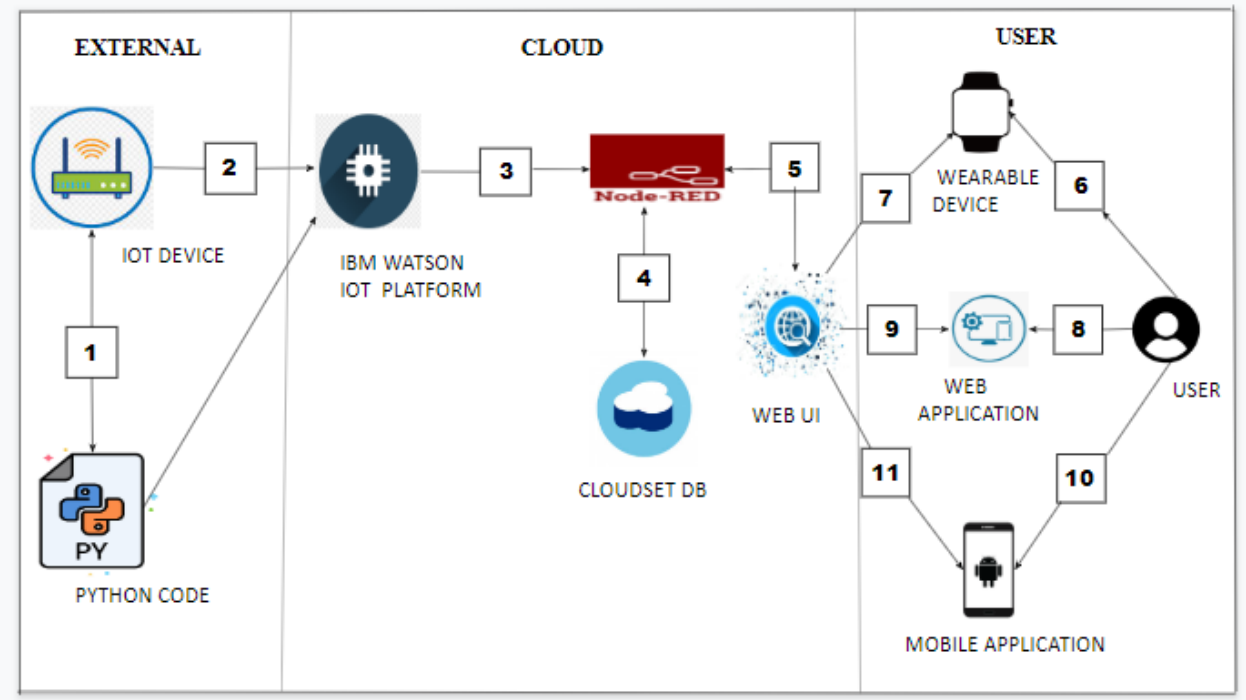


TABLE-1: COMPONENTS & TECHNOLOGIES:

S.No	COMPONENTS	DESCRIPTION	TECHNOLOGY
1.	**User Interface**	Using WEB UI,Mobile App,SMS service and wearable devices user can interact	Node -RED,Fast SMS,MIT App Inventor,HTML, CSS, Java,Python code
2.	**Application Logic-1**	Collecting input from smart beacons	C and Python
3.	**Application Logic-2**	Computing the input data to the cloud	IBM Watson IOT platform,Cloudant DB and Node-RED
4.	**Application Logic-3**	Exhibit the data to the user	WEB UI,Fast SMS and Mobile application
5.	**Database**	Real Time database	Cloudant DB
6.	**Cloud DataBase**	Database service built and accessed through a cloud platform.	IBM Cloudant
7.	**File Storage**	Storage Service	IBM Block Storage
8.	**External API-1**	To convey SMS to user	Fast SMS
9.	**External API-2**	Code for tasks can be composed for the working of smart beacon devices.	Python and C Modules
10.	**External API-3**	To access time	World Time APL
11.	**Smart Beacon**	To detect the area and update the data in the cloud.	NodeMCU and Sensors
12.	**Infrastructure(Server /Cloud)**	Establishing application on cloud	IBM Cloud Services

TABLE-2:APPLICATION CHARACTERISTICS:

S.N	CHARACTERISTICS	DESCRIPTION	TECHNOLOGY
1.	**Open-Source Frameworks**	To build web application,mobile application and circuit designing using Node-RED open source frameworks	App Inventor and Node-Red Framework
2.	**Security Implementations**	Unique login credentials should be given to the users	Email and respective password
3.	**Scalable Architecture**	The 3 – tier architecture used in the project has a separate user interface, application tier and the data tier makes the process easy	IBM WATSON
4.	**Availability**	The web application is highly available as it is deployed in cloud	IBM Cloud
5.	**Performance**	The performance of the web UI is improved using cache, security services	IBM cloud services

5.3 USER STORIES

USER TYPE	FUNCTIONAL REQUIREMENTS	USER STORY NUMBER	USER STORY/TASK	ACCEPTANCE CRITERIA	PRIORITY	RELEASE
INITIALIZATION	Registration	USN-1	Registration of the user using their credentials	Registration should be easily available to all the workers in industries	High	Sprint-1

	User Confirmation	USN-2	Confirm the user by sending a verification link and OTP to the mobile number.	The link should be working perfectly and OTP should be sent and within one minute	Medium	Sprint-2
	Rule and Regulations	USN-3	Share the guidelines to be followed during the initialization process	Guidelines help even ordinary people to be aware of the installment and working	Medium	Sprint-3
MONITOR THE ENVIRONMENT	Installation	USN-4	The beacon devices should be installed all around the industrial places	The smart beacon devices should cover the entire industries with some distance between them	High	Sprint-1
	Collection of data	USN-5	The ability of the beacon devices is to monitor	The temperature parameter is the important	High	Sprint-1

			the temperature of the industrial areas	parameter to identify the environment condition		
	Catalog data	USN-6	The temperature of the industrial area is stored in IBM cloud services and in wearable devices and monitors.	Data should be synchronized between cloud and the wearable devices	High	Sprint-1
EMPLOYEES	Wearable devices	USN-7	The wearable devices display the temperature of the industrial area when they go near the beacon devices	The devices should be available to all workers and be worn when they enter the industrial area	High	Sprint-1

	Wearable device customization	USN-8	Devices systemized based on their ability of knowledge such as language , font, size, etc.	Customization help the workers to have a better understanding and act according to it	Medium	Sprint-2
	SMS Intimation	USN-9	If the observed data for an area is found to be risky for the workers, then they shall be notified via SMS	The workers is notified through the SMS if the beacon device identify any rise in temperature in the environment	High	Sprint-1
ADMIN	Monitor	USN-10	The temperature absorbed by the beacon devices will be displayed in the monitor through the cloud	The temperature changes are analyzed and monitored	High	Sprint-1

	Monitor Systemization	USN-11	The dashboard can be customized based on the administrato r such as alert button, message to the help counter	The admin can systemize the UI based on their needs	Medium	Sprint-2

CHAPTER 6

PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION

SPRINT	FUNCTIONAL REQUIREMEN T(EPIC)	USER STORY NUMBER	USER STORY/TASK	STORY POINTS	PRIORITY	TEAM MEMBERS
SPRINT-1	Registration (Industrial Owner)	**USN - 1**	As a owner, registration into the application through email and password	5	High	Keerthana, Jahnavi
SPRINT-1	Registration (Industrial Worker)	**USN - 2**	As an employee, registration into the	2	High	Keerthi, Kayalvizhi

			application through email and password			
SPRINT-1	Data Modules (Industrial Owner)	**USN - 3**	As a owner, environmental temperature and humidity are received	5	High	Keerthana, Kayalvizhi
SPRINT-1	Data Modules (Industrial Worker)	**USN - 4**	As an employee, environmental temperature and humidity are received	2	High	Jahnavi, Keerthi
SPRINT-1	Login (Industrial Owner)	**USN - 5**	As a owner,login into the account by email and password	3	Medium	Keerthi, Keerthana
SPRINT-1	Login (Industrial Worker)	**USN - 6**	As an employee, login into the account by email and password	1	Medium	Jahnavi, Kayalvizhi

SPRINT-2	IOT Dashboard Interfacing	**USN - 7**	As an employee, interfacing data and internet can be done	8	High	Kayalvizhi, Keerthana
SPRINT-3	Web UI	**USN - 8**	As an employee, accessing data through website	3	High	Jahnavi, Keerthi
SPRINT-4	Mobile UI	**USN - 9**	As an employee, datacan be viewed through mobile application	2	Medium	Keerthana, Keerthi

6.2 SPRINT DELIVERY SCHEDULE

PROJECT TRACKER,VELOCITY & BURN DOWN CHART(4)

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	4 Days	14 NOV 2022	19 NOV 2022	20	19 NOV 2022

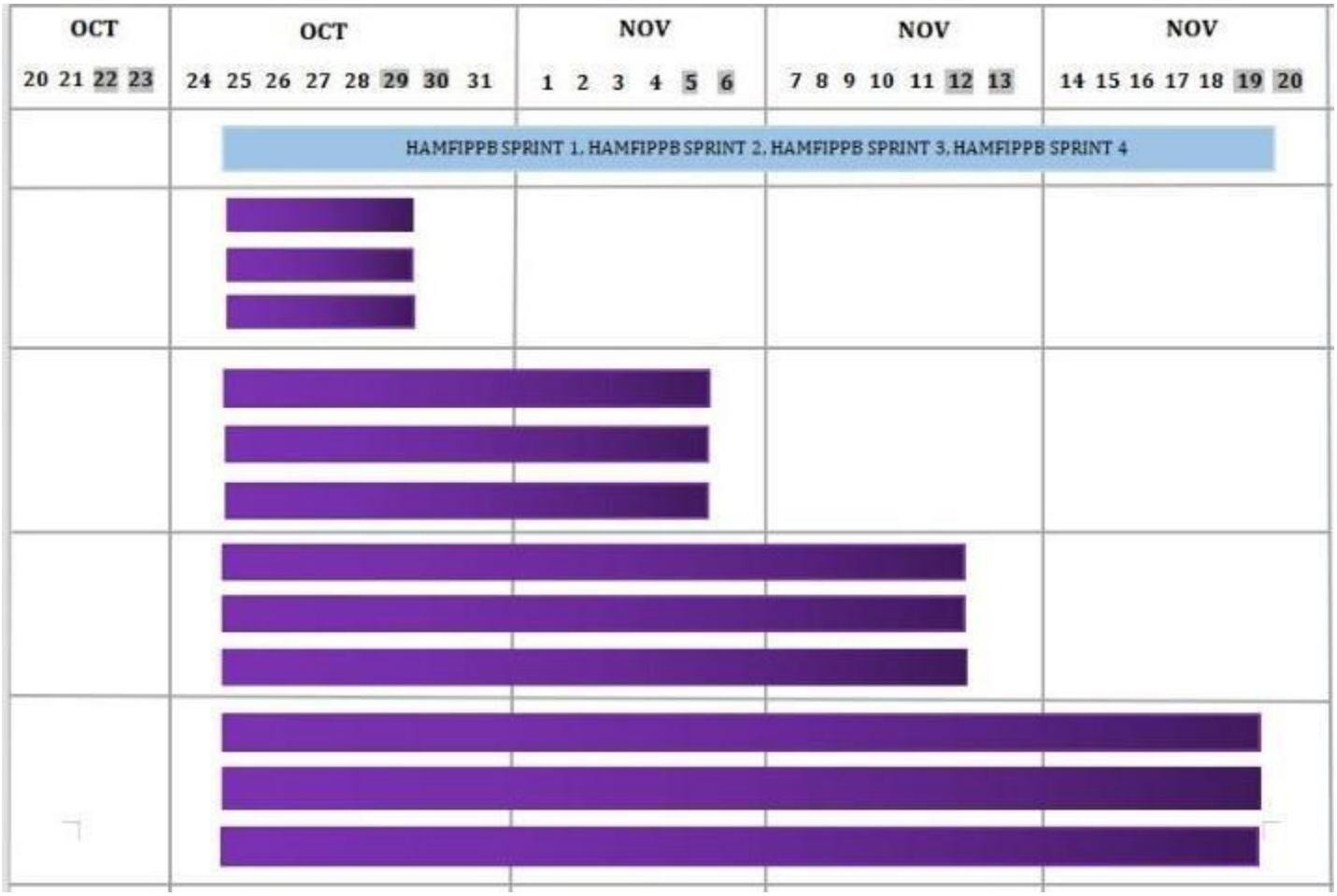
VELOCITY:

Let us consider 10 days as sprint duration and the velocity of the team is 20(points per sprint).Let us consider the team's average velocity(Avg V) per iteration unit(Story points per day)

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

BURNDOWN CHART:

A burndown chart shows the amount of work that has been completed in an epic or sprint, and the total work remaining. Burndown charts are used to predict your team's likelihood of completing their work in the time available. They are also great for keeping the team aware of any scope creep that occurs.


OCT
20 21 22 23
OCT
24 25 26 27 28 29 30 31
NOV
1 2 3 4 5 6
NOV
7 8 9 10 11 12 13
NOV
14 15 16 17 18 19 20
HAMFIPPB SPRINT 1. HAMFIPPB SPRINT 2. HAMFIPPB SPRINT 3. HAMFIPPB SPRINT 4

CHAPTER 7

CODING AND SOLUTIONING

7.1 FEATURE 1

Registration of all the wearable devices with the cloud is effortless
Accessibility of the cloud is smooth and available at any time

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>
```

7.2 FEATURE 2

The temperature of the industrial areas is monitored and alert is given through SMS to all the workers.Temperature is also displayed in the dashboard of the administrator

ALGORITHM:

- ❖ Import the Packages
- ❖ Create ‘myConfig’ location
- ❖ Implement the wiotp.sdk.device.DeviceClient
- ❖ Run a while Loop
- ❖ Get temperature and humidity sensor readings
- ❖ Display the data

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(- 20,125)
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
}
```

CHAPTER 8

TESTING

8.1 TEST CASES

pyth.py - C:\Users\91949\AppData\Local\Programs\Python\Python37\pyth.py (3.7.0)

File Edit Format Run Options Window Help

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

#Provide your IBM Watson Device Credentials
organization = "m22k3n"
deviceType = "abcd"
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)
```

vimeo.com is sharing your screen. Stop sharing Hide

Ln: 15 Col: 0

28°C Sunny | Search | ENG IN | 12:10 19-11-2022

Python 3.7.0 Shell

File Edit Shell Debug Options Window Help

```
Published Temperature = 102 C Humidity = 99 % to IBM Watson
Published Temperature = 97 C Humidity = 79 % to IBM Watson
Published Temperature = 93 C Humidity = 91 % to IBM Watson
Published Temperature = 101 C Humidity = 63 % to IBM Watson
Published Temperature = 94 C Humidity = 88 % to IBM Watson
Published Temperature = 95 C Humidity = 84 % to IBM Watson
Published Temperature = 105 C Humidity = 73 % to IBM Watson
Published Temperature = 105 C Humidity = 72 % to IBM Watson
Published Temperature = 97 C Humidity = 68 % to IBM Watson
Published Temperature = 92 C Humidity = 63 % to IBM Watson
Published Temperature = 98 C Humidity = 94 % to IBM Watson
Published Temperature = 109 C Humidity = 73 % to IBM Watson
Published Temperature = 109 C Humidity = 61 % to IBM Watson
Published Temperature = 93 C Humidity = 83 % to IBM Watson

=== RESTART: C:\Users\91949\AppData\Local\Programs\Python\Python37\pyth.py ===
2022-11-19 12:07:45,348   ibmiotf.device.Client      INFO    Connected successfully: d:m22k3n:abcd:12345
Published Temperature = 97 C Humidity = 82 % to IBM Watson
Published Temperature = 90 C Humidity = 62 % to IBM Watson
Published Temperature = 109 C Humidity = 62 % to IBM Watson
Published Temperature = 97 C Humidity = 90 % to IBM Watson
Published Temperature = 103 C Humidity = 70 % to IBM Watson
Published Temperature = 97 C Humidity = 83 % to IBM Watson
Published Temperature = 110 C Humidity = 91 % to IBM Watson
Published Temperature = 102 C Humidity = 99 % to IBM Watson
Published Temperature = 90 C Humidity = 95 % to IBM Watson
Published Temperature = 106 C Humidity = 78 % to IBM Watson
Published Temperature = 95 C Humidity = 60 % to IBM Watson
Published Temperature = 97 C Humidity = 74 % to IBM Watson
Published Temperature = 91 C Humidity = 61 % to IBM Watson
Published Temperature = 103 C Humidity = 85 % to IBM Watson
Published Temperature = 102 C Humidity = 76 % to IBM Watson
Published Temperature = 102 C Humidity = 94 % to IBM Watson
Published Temperature = 106 C Humidity = 83 % to IBM Watson
Published Temperature = 108 C Humidity = 62 % to IBM Watson
Published Temperature = 103 C Humidity = 95 % to IBM Watson

=== RESTART: C:\Users\91949\AppData\Local\Programs\Python\Python37\pyth.py ===
2022-11-19 12:10:51,914   ibmiotf.device.Client      INFO    Connected successfully: d:m22k3n:abcd:12345
Published Temperature = 108 C Humidity = 74 % to IBM Watson
```

vimeo.com is sharing your screen. Stop sharing Hide

Ln: 1230 Col: 0

28°C Sunny | Search | ENG IN | 12:10 19-11-2022

8.2 USER ACCEPTANCE TESTING

CODE:

```
#include <DHT.h> WiFiClient wifiClient; String data3;
#define DHTTYPE DHT11 #define DHTPIN 4
#define MQTPIN 34
DHT dht(DHTPIN, DHTTYPE);
#define ORG "22h49t"
#define DEVICE_TYPE "NodeMCU" #define DEVICE_ID
"NodeMCU"
#define TOKEN "12345678" #define speed 0.034 void callback(char*
topic, byte* playload, unsigned int payloadLength); char server[] =
ORG ".messaging.internetofthings.ibmcloud.com"; char
publishTopic[]
= "iot-2/evt/Data/fmt/json"; char topic[] = "iot-
2/cmd/test/fmt/String"; char authMethod[] = "use-token-auth"; char
token[] = TOKEN; char clientId[] = "d:" ORG ":" DEVICE_TYPE ":"
DEVICE_ID; PubSubClient client(server, 1883, callback ,
wifiClient);
void publishData(); String command;

String data = ""; long duration; float dist;
void setup()
{
Serial.begin(115200); dht.begin(); wifiConnect(); mqttConnect();
}
```

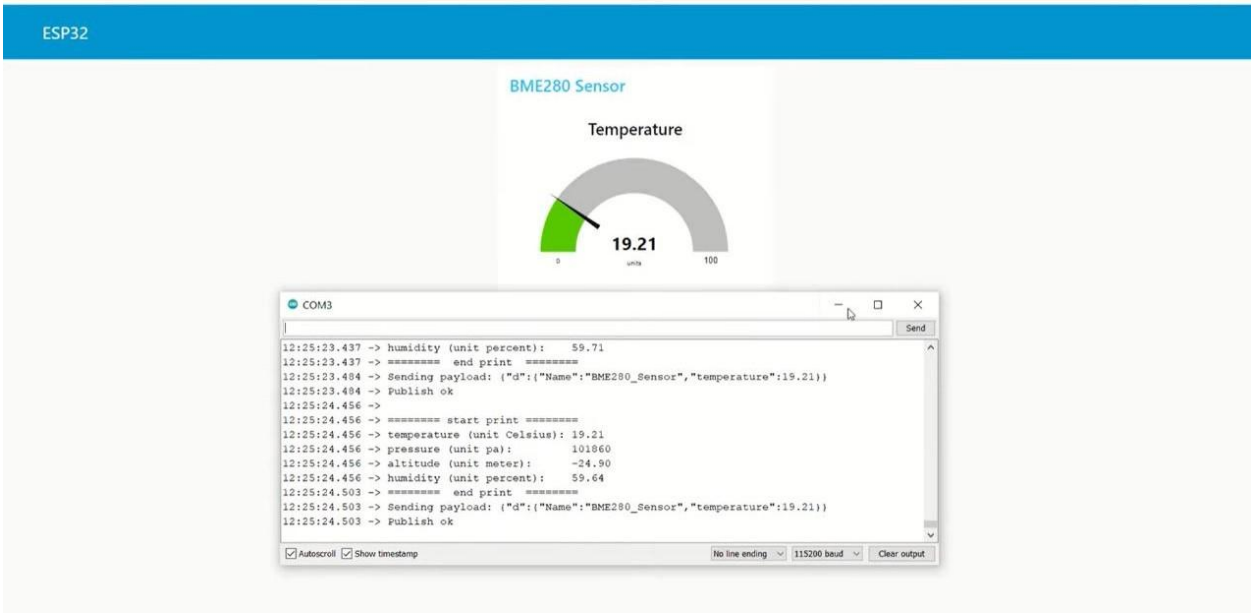
```
void loop() { publishData(); delay(500); if (!client.loop()) {
mqttConnect();
}
}
void wifiConnect() {
Serial.print("Connecting to "); Serial.print("Wifi");
WiFi.begin("JerroldWi-Fi","75779901"); while (WiFi.status() !=
WL_CONNECTED) { delay(500); Serial.print(".");
}
Serial.print("WiFi connected, IP address: ");
Serial.println(WiFi.localIP());
}
void mqttConnect() {
if (!client.connected()) {
Serial.print("Reconnecting MQTT client to "); Serial.println(server);
while (!client.connect(clientId, authMethod, token)) {
Serial.print("."); delay(500);
}
initManagedDevice(); Serial.println();

}
void initManagedDevice() { if (client.subscribe(topic)) {
Serial.println("IBM subscribe to cmd OK");
} else {
Serial.println("subscribe to cmd FAILED");
}
}
```

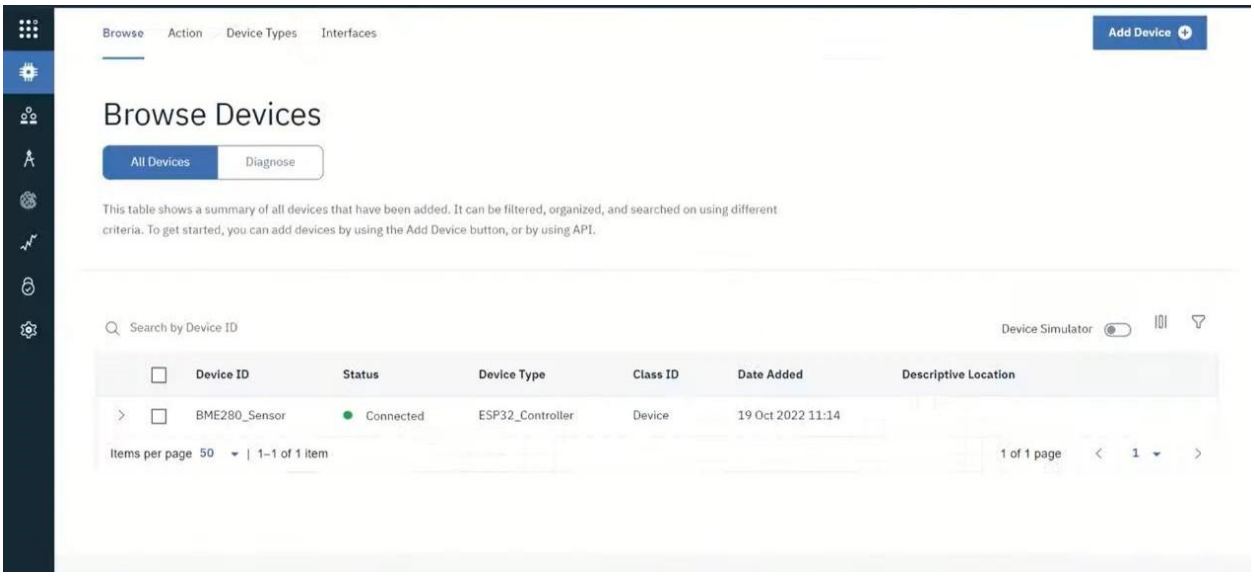
```
void publishData()
{
int sensorValue = analogRead(MQTPIN); //MQT 135 connected to
GPIO 34 (Analog ADC1_CH6)
Serial.print("AirQua=");
Serial.print(sensorValue, DEC); Serial.println(" PPM"); float humid =
dht.readHumidity(); float temp = dht.readTemperature(true); String
payload = "{\"Humidity\":"; payload += humid; payload += "}"; if
(client.publish(publishTopic, (char*) payload.c_str())) {
Serial.println("Publish OK");
}
payload = "{\"Temperature\":"; payload += temp; payload += "}"; if
(client.publish(publishTopic, (char*) payload.c_str())) {
Serial.println("Publish OK");
}
payload = "{\"AirQuality\":"; payload += String(sensorValue);
payload += "}"; if (client.publish(publishTopic, (char*)
payload.c_str())) { Serial.println("Publish OK");
}

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++) {
dist += (char)payload[i];
}
Serial.println("data:" + data3); if (data3 == "lighton") {
Serial.println(data3);
```

```
}
data3 = "";
}
```

CHAPTER 9

PERFORMANCE METRICS





CHAPTER 10

ADVANTAGES

- ➔ It is highly reliable and economical.
- ➔ It stores data that can be used for future needs.
- ➔ Many devices can be included in future extensions.
- ➔ It integrates, delivers and customizes the best solution in the market.
- ➔ It is efficient and boosts the business with the help of data stored in the system.
- ➔ It is an open solution system and it is easily integrable with external developments at any level.
- ➔ It also reduces human effort.
- ➔ It reduces the level of destruction.
- ➔ Automation of sensors leads to better monitoring of devices.
- ➔ It has robust and simple construction.

CHAPTER 11

CONCLUSION

Hazardous area monitoring for industrial areas is done in many ways but IOT is a well renowned method to integrate, monitor, process in an easy way. The IOT based industrial monitoring system is highly efficient and provides real time monitoring of many environmental parameters and helps to increase their yields.

In this project, We interfaced embedded systems and IOT to obtain an effortless monitoring of the industrial areas. Many beacon devices are

installed to monitor the condition and alert the workers. If the temperature of the environment exceeds, an alert is sent to all workers and immediate actions are taken, Which makes this project even more efficient.

This can be installed in many hazardous industries such as metal refineries, underground industries, mining and other heavy parts manufacturing factories. This shows the significant role of IOT in monitoring industries.

CHAPTER 12

FUTURE SCOPE

The IoT-based study can be enhanced further by offering extra functionality to industry personnel to improve industry control and monitoring. Temperature sensors can also be connected to the system to safeguard the safety of workers and commodities in the event of a fire.

Data can be used to minimize industrial dangers in high-profile factories, track yield in power plants, assure safety in fast-paced industries, and assess nuclear safety levels, among other things.

Time can be saved if the info is delivered quickly. It is reliable for damage and fault detection and real-time monitoring systems. An unlimited number of devices can be included in future extensions. It provides open solutions and it is easily integrable with external developments at any level.

CHAPTER 13

APPENDIX

CODE

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

#Provide your IBM Watson Device Credentials
organization = "i3869j"
deviceType = "abcd"
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")

        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()
```

DEMO LINK:

https://www.kapwing.com/videos/6378fdbc13328901b81d9176

GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-33373-1660219282