HAZARDOUS AREA MONITORING FOR INDUSTRIAL PLANT POWERED BY IOT

NALAIYA THIRAN PROJECT BASED LEARNING

ON

PROFESSIONAL READINESS FOR INNOVATION, EMPLOYABILITY AND ENTREPRENEURSHIP

A PROJECT REPORT

S. RITHIK VARSHAN	19106098
K. SENTHIL PRABU	19106104
O. S. SHARMA	19106106
D. VISHVA	19106136

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

HINDUSTHAN COLLEGE OF ENGINEERING AND TECHOLOGY

Approved by AICTE, New Delhi, Accredited with 'A' Grade by NAAC

(An Autonomous Institution, Affiliated to Anna University, Chennai)

COIMBATORE – 641 032

NOVEMBER 2022

Hindusthan College of Engineering And Technology

Approved by AICTE, New Delhi, Accredited with 'A' Grade by NAAC **(An Autonomous Institution, Affiliated to Anna University, Chennai)** Valley Campus, Pollachi Highway, Coimbatore – 641 032

INTERNAL MENTOR:

Dr. P. K. POONGUZHALI

Associate Professor

Department of Electronics and Communication Engineering

Hindusthan College of Engineering and Technology,

Coimbatore - 641 032

INDUSTRY MENTOR

SANTOSHI, BHARADWAJ

IBM

ABSTRACT

The Internet of Things (IoT) is a new sector that aims to connect "things," "people," and "machines" to the internet. Modernization and automation are sweeping the globe, with IoT-based industrial monitoring solutions at the forefront. The importance of assessing the state of the industry is vital to the safety and efficiency of the products. The goal of this study is to create an IoT-based industrial monitoring system with intelligent sensors. Because of the integration of big data, the Blynk app can be used to monitor status from anywhere on the planet. Data analysis has been streamlined, allowing for easier IoT monitoring. The proposed technology could be beneficial to manufacturing industries. Adding technology to any kind of manufacturing industry will assure the safety and well-being of the people as well as prevent accidents. Using automation technology reduces the chances of loss and accidents in the machinery world.

TABLE OF CONTENTS

HAZARDOUS AREA MONITORING FOR INDUSTRIAL PLANT POWERED BY IOT

1. INTRODUCTION

The Internet of Things (IoT) is pervasive across many industries and has an impact on both business operations and everyday lives. IoT stands for the change from computer networks to an object network where each component of daily social and professional life.

Due to accidents, inadequacies, or plain negligence on the part of industry authorities, there have been countless fatalities, severe injuries, and catastrophic damages that have disrupted people's lives suffering' and future generations. To avoid any future catastrophe like this project suggests a cutting-edge checking methodology reliant on the Internet of Things (IoT).

This construction project creates a mechanical observation framework that recognises anomalous concentrations of gases including carbon monoxide, LPG, butane, and hydrogen that could set off an explosion. Additionally, it displays air volume. Along with monitoring the temperature and cleaning up any pollution the company may have humidity levels.

Integration of data from multiple sensors ensures the industry's safety. The system operates reliably and steadily. It is the best and most responsible way to monitor hardware security.

2. OBJECTIVE:

- Monitoring temperature fluctuations is particularly important since different industrial equipment's operations are impacted by temperature variations, which are a physical component of the environment.
- The computer has a microcontroller chip incorporated for managing various settings, and a system keeps track of the real-time data collection. On LCD, values from various parameters are compiled and shown.
- A collection of all the code is burned into the Arduino.
- Each code stands for a certain parameter, such as air, temperature, pressure, or humidity. The systems platform can be used to implement the intelligent industrial remote monitoring of the power system, intelligent furniture monitoring, intelligent warehouse monitoring, etc. This assures the user of the stability and dependability of the system.
- It has good social aspects and is most effective and most economical means of equipment safety monitor.
- It senses changes in temperature, senses smoke, flame etc and sends it to control station by android app.
- In the prototype, installations of sensors in three distinct locations to identify the exact location of fire hazards that have taken place.

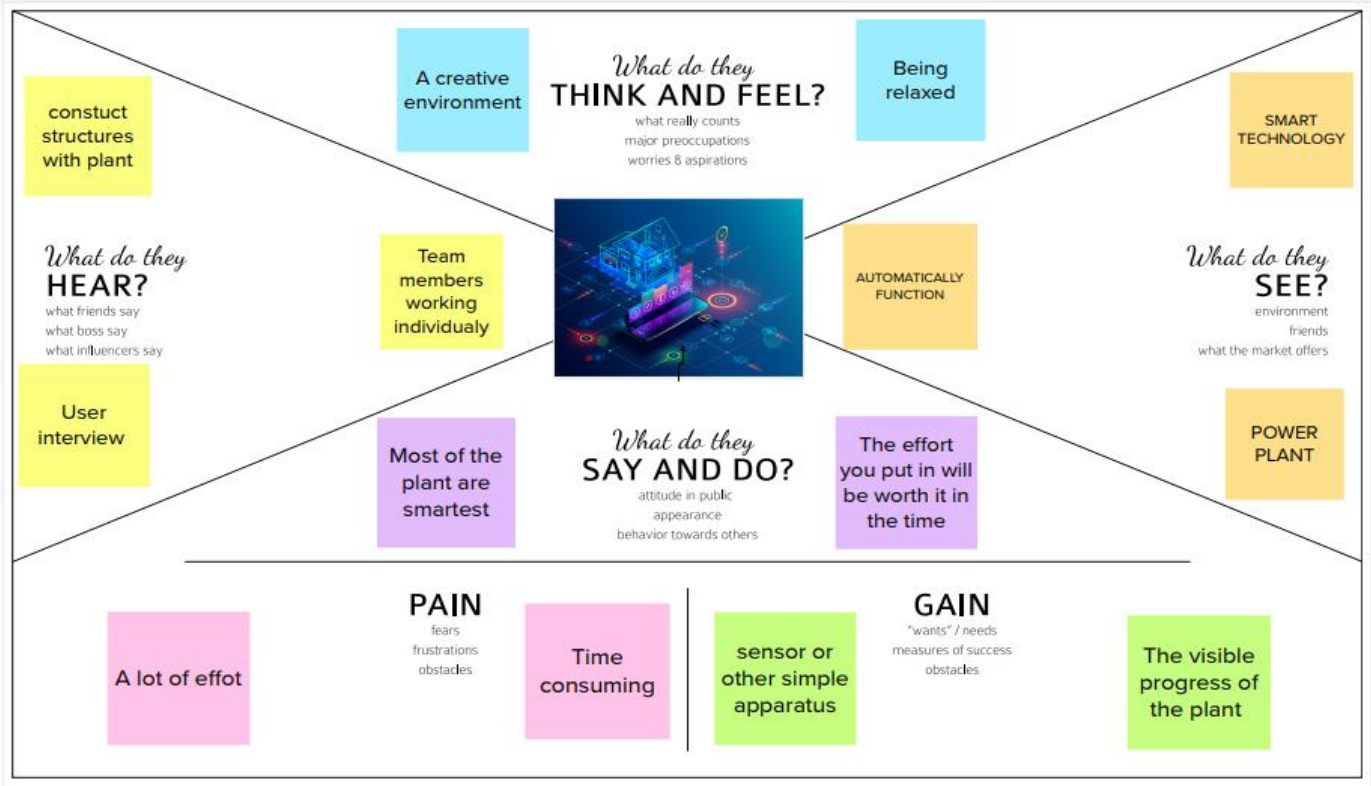
3. IDEATION PHASE

3.1 LITERATURE SURVEY:

The employment of wireless technology is improving people's safety and pleasure in the modern world. IOT, AR, AI, and other wireless technologies are in high demand for adapting to changing lifestyles. Using the same wireless sensors from earlier inventions, we wanted to build a sensor network for the detection and prevention of risks, followed by the eradication of the source that caused the hazard in the first place. The prototype contains sensors for temperature, humidity, fire, and gas.

The variables that can be monitored in advance to stop the occurrence of a major fire include temperature, gas, and humidity. Fire might be avoided if certain parameters are kept under control, and vice versa. We have employed water as an extinguishing agent to put out and put out the fire. A voice module is also included in the prototype. This gadget records audio notes and plays them back to provide an audible alarm of the parameter it has identified. For instance, if a sensor detects a dangerous gas, such as carbon monoxide, in the environment, the speech module will play the audio output "gas detected." As a result, this prototype can be highly helpful for workers in factories, power plants, etc.

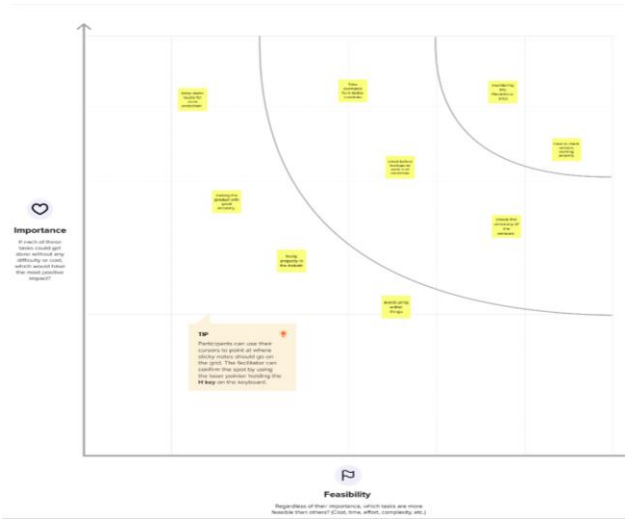
3.2 Empathy Map:



3.3 Ideation :

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	A Employee	**To Monitor the temperature and environment Pressure and some paramters**	It takes long to detect Temperature	Of exploitation to sudden increase of Temperature	Discomfort
PS-2	An Admin	**To view the data to take necessary precaution in time**	It takes more time to view the data	Of less interaction	Dissatisfied

3.4 Brainstorming:



4. PROJECT DESIGN PHASE 1

4.1 Proposed Solution:

S.No.	Parameter	Description
1	Problem Statement (Problem to be solved)	In today's world in spite of security and automation in industrial plants and environments are quiet crucial for machines and humans.
2	Idea / Solution description	This system has been designed to detect dangerous situations like breakdown which is the most important parameter for occurring leakage current in substation and help to avoid them.
3	Novelty / Uniqueness	This IoT system is affordable and suited for several types of industries and it is not complex as the ones that are available in the market.
4	Social Impact / Customer Satisfaction	In Today's world Industrial IoT manufacturers are looking for solutions to boost efficiency and output while maintaining worker safety at work sites in hazardous areas.
5	Business Model (Revenue Model)	This Iot model highly focuses on the safety of the people and it surely makes a fortune for its investors because its flexible and it is not as complicated as the ones in available in the markets.
6	Scalability of the Solution	Today's industrial IoT manufacturers are looking for solutions to boost efficiency and output while maintaining worker safety at work sites working safely in hazardous environments.

4.2 Problem Solution Fit:

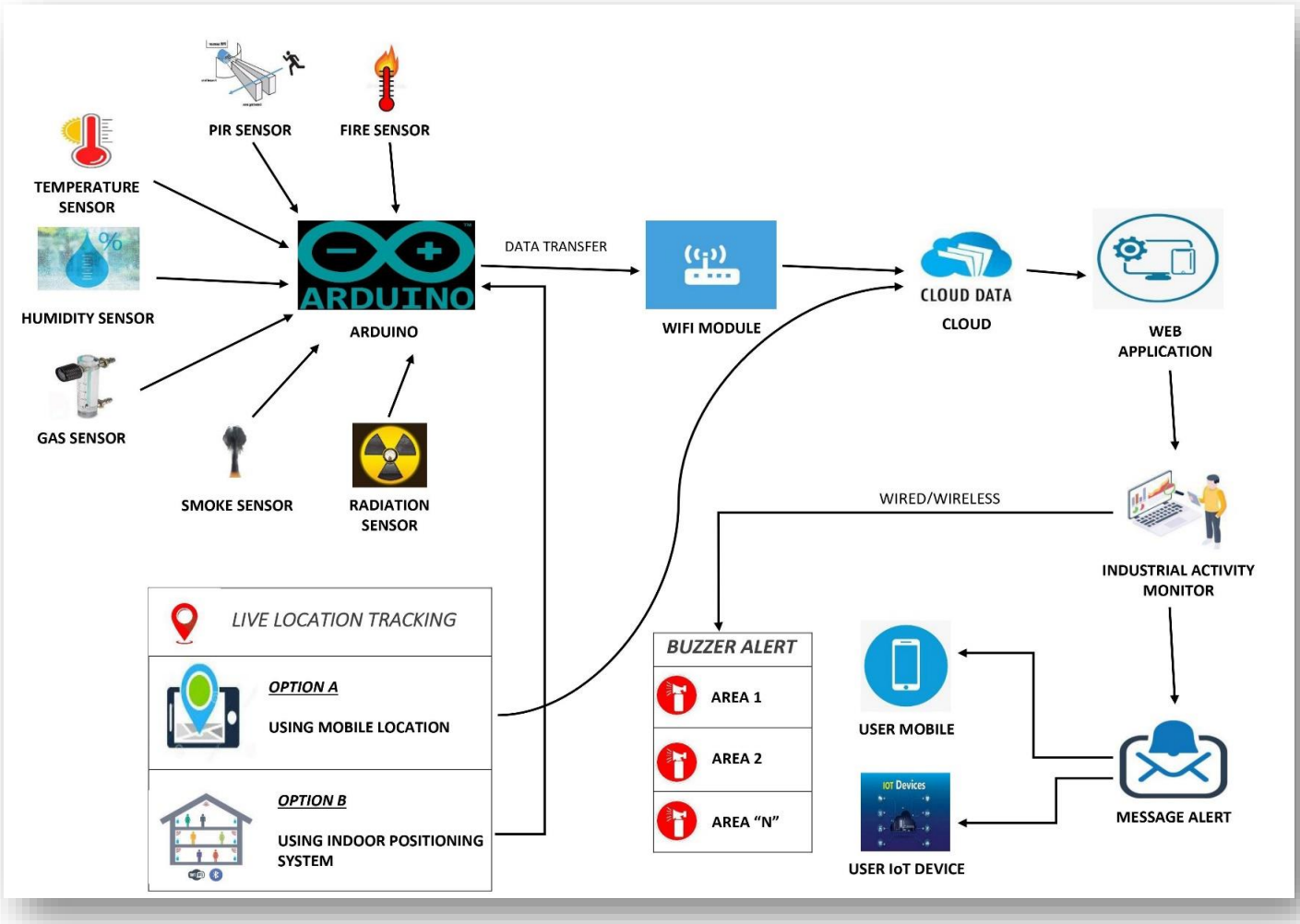
4.3 Solution Architecture:

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.

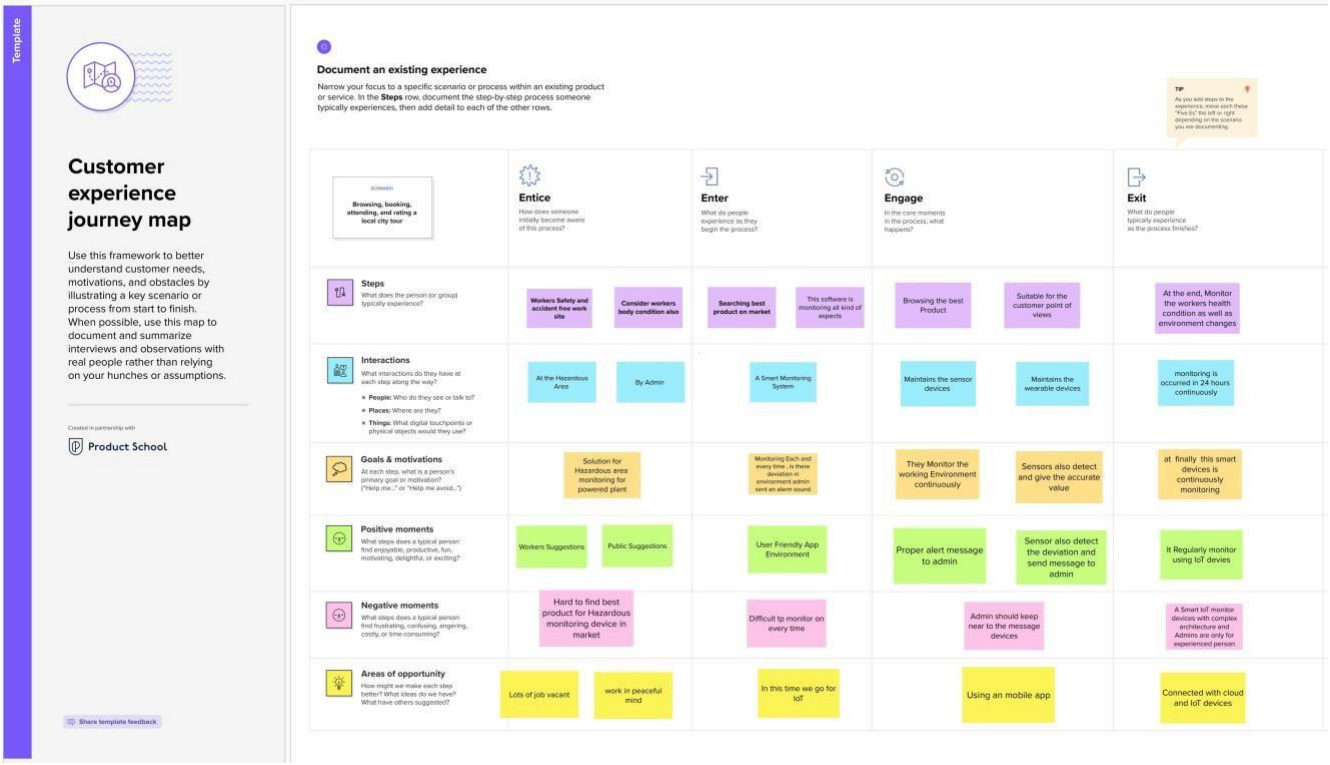
Example - Solution Architecture Diagram:

Hazardous Area Monitoring for Industrial Plant powered by IoT



5. PROJECT DESIGN PHASE 2

5.1 Customer Journey Map:



5.2 Solution Requirements:

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Online Payment for the service
FR-2	User Access	Access the details using web browser Access the details using mobile application
FR-3	User alert	Gets alert as an SMS message Gets alert alarm in the working area.

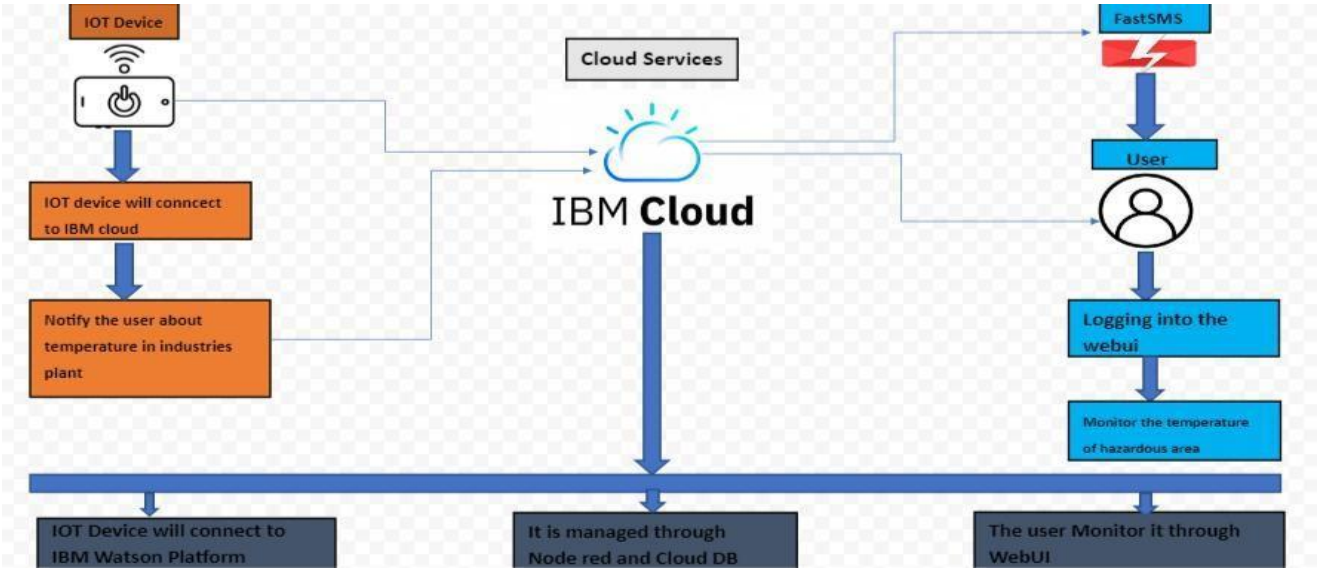
Non-functional Requirements:

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

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The device must be usable by the customer anywhere
NFR-2	**Security**	Data from the sensors are stored securely and away from other data
NFR-3	**Reliability**	Data can be retrieved anytime and no data is discarded without customer knowledge
NFR-4	**Performance**	No performance delay in case of large number of data or more parameters
NFR-5	**Availability**	The device doesn't fail even under harsh conditions. Device continues to send parameters, even after an alert situation.
NFR-6	**Scalability**	Device must be capable of measuring conditions even in a larger industry

5.3 Data Flow Diagrams:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a 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.



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 (Industrial Owner)	Registration	USN-1	As an Industrial Owner, I can register into the application by entering email & password	I can access my account / dashboard	High	Sprint-1
	Data Modules	USN-2	As an Industrial Owner, I can get message about the temperature and humidity	I can receive confirmation email & click confirm	High	Sprint-1
	Login	USN-3	As an industrial Owner, I can login into my account through email and Password	I can access my account	Medium	Sprint-2
	Dashboard	USN-4	As an Industrial Owner, I can monitor of temperature	I can access the dashboard with individual Login id/password	High	Sprint-1
Customer	Registration	USN-1	As an Industrial	I can access my	High	Sprint-1

(Industrial Worker)			Worker, I can register into the application by entering email & password	account / dashboard		
	Data Modules	USN-2	As an Industrial Worker, I can get message about the temperature and humidity	I can receive confirmation email & click confirm	High	Sprint-1
	Login	USN-3	As an industrial Owner, I can login into my account through email and Password	I can access my account	Medium	Sprint-2
	Dashboard	USN-4	As an Industrial Owner, I can get alert high temperature	I can access the dashboard with individual Login id/password	High	Sprint-1

5.4 Technology Stack:

Table-1: Components & Technologies:

S.No	Component	Description
1.	User Interface	Web UI, Mobile App, Chatbot , MIT inventor etc.
2.	Application Logic-1	IoT applications use machine learning algorithms to analyze massive amounts of connected sensor data in the cloud. Using real-time IoT

		dashboards and alerts, you gain visibility into key performance indicators, statistics for mean time between failures, and other information.
3.	Application Logic-2	The internet of things, or IoT, is a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers (UIDs) and the ability to transfer data over a network without requiring human- to-human or human-to-computer interaction.
4.	Application Logic-3	Watson Assistant lets you build conversational interfaces into any application, device, or channel. Add a natural language interface to your application to automate interactions with your end users. Common applications include virtual agents and chat bots that can integrate and communicate on any channel or device.
5.	Database	IoT data comes in three different types, based on the device generating it and the use case. Status data: Status data is basic, raw data that communicates the status of a device or system.
6.	Cloud Database	Cloudant handles software and hardware provisioning, management and scaling, and support
7.	File Storage	IBM Cloud® Block Storage is persistent, high-performance iSCSI storage

		that is provisioned and managed independently of compute instances. iSCSI-based Block Storage LUNs are connected to authorized devices through redundant multi-path I/O (MPIO) connections.
8.	External API-1	Runtime APIs Admin HTTP API This HTTP-based API can be used to remotely administer the runtime. It is used by the Node- RED Editor and command-line admin tool. Hooks The Hooks API provides a way to insert custom code into certain key points of the runtime operation. Storage This API provides a pluggable way to configure where the Node- RED runtime stores data.
9.	External API-2	Editor APIs The APIs available in the editor for nodes and plugins to use. This includes a set set of standard UI widgets that can be used within a node's edit template. Module APIs The APIs provided by npm modules that NodeRED is built from. These can be used to embed Node-RED into existing Node.js applications.

10.	Machine Learning Model	OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products.
11.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	It is an open-source IoT framework. The main purpose of the framework is data collection and device management. Further, it uses IoT protocols like HTTP, MQTT, and CoAP for device connectivity. It is also highly scalable as every type of device easily integrated	Technology of Opensource framework
2.	Security Implementations	Safety The IoT platform should never do something it isn't supposed to do. The principal game changer regarding software in the domain of IoT is safety coupled with accountability and responsibility. Any applied automation through an IoT solution means that we have faith in the system and trust that it will never do harm in the environment. Security The IoT platform must ensure proper device management (via authentication and authorization mechanisms), data privacy, integrity, and confidentiality via secure communication and encryption of data. Security is especially crucial for an IoT platform, as it will rely more on automated security.	e.g. SHA-256, Encryptions, IAM Controls, OWASP etc.
3.	Scalable Architecture	Portability The IoT platform must be portable if it is destined to heterogeneous nodes. This may be achieved by leveraging virtualization technologies (for example, by using the Java Virtual Machine), or packing the	Technology used

		deliverable into host operating system oblivious form (like the Docker image). Adaptability To support an extensive list of devices, and provide more service APIs for integration purposes, it is mandatory to have an adaptable IoT platform. The possible usage scenarios are vast, and cannot be predetermined in advance. Usability To reduce the deployment hassle, and quickly get users up and running with an IoT platform, it must be in a user- friendly form in multiple aspects. This includes the management, supervision, and reporting facilities. Efficiency The IoT platform should ideally have a small footprint, employ advanced data storage technologies, and require adequate hardware resources to be usable in both real time and regular contexts. To move the computation near devices it should run on less capable hardware (for example, inside a smart meter or smartphone).	
4.	Availability	server farm. Load balancers improve application availability and responsiveness and prevent server overload.	Technology used
5	Performance	Fog Computing is a new paradigm and an extension of Cloud Computing. This better performance results justifies the suitability of IoT applications using Fog-Based Cloud Network approach. Imperva and other CDNs can be used to reduce your website's latency, improving overall site performance and UX. Among other methods, this is done through: Content caching – CDNs cache and compress mirror versions of your web pages, which are then stored in strategically placed data centers	Technology used

6. PROJECT PLANNING PHASE

6.1 Prepare Milestone and Activity List:

TITLE	DESCRIPTION	DATE
Literature Survey on TheSelected Project and Information Gathering	A Literature Survey is a compilation summary of research done previouslyin the given topic. Literature survey can be taken from books, research paper online or from any source.	25 September 2022
Prepare Empathy Map	Empathy Map is a visualization tool which can be used to get a betterinsight of the customer	19 September 2022
Ideation-Brainstorming	Brainstorming is a group problemsolving session where ideas are shared, discussed and organized among the team members.	20 September 2022
Define Problem Statement	A Problem Statement is a concise description of the problem or issues aproject seeks to address. The problem statement identifies the current state, the desired future state and any gaps between the two.	17 September 2022
Problem Solution Fit	This helps us to understand the thoughts of the customer their likes, behaviour, emotions etc.	02 October 2022
Proposed Solution	Proposed solution shows the current solution and it helps is going towardsthe desired result until it is achieved.	18 September 2022

Solution Architecture	Solution Architecture is a very complex process I.e. it has a lot of sub-processes and branches. It helps in understanding the components andfeatures to complete our project.	29 September 2022
Customer Journey	It helps us to analyse from the perspective of a customer, who uses our project.	9 October 2022
Functional Requirement	Here functional and non-functional requirements are briefed. It has specific features like usability, security, reliability, performance, availability, and scalability.	16 October 2022
Data Flow Diagrams	Data Flow Diagram is a graphical or visual representation using a standardized set of symbols and notations to describe a business's operations through data movement.	14 October 2022
Technology Architecture	Technology Architecture is a more well defined version of solution architecture. It helps us analyze and understand various technologies that needs to be implemented in the project.	15 October 2022
Prepare Milestone & Activity List	It helps us to understand and evaluate our own progress and accuracy so far.	29 October 2022
Spring Delivery Plan	Sprint planning is an event in scrum that kicks off the sprint. The purpose of sprint planning is to define what can be delivered in the sprint and how that work will be achieved.	14 ovember 2022

6.2 Sprint Delivery Plan:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority
Sprint-1	IBM Cloud services	US-1	Create the IBM Cloud services which are being used in this project.	6	High
Sprint-1	IBM Cloud services	US-2	Configure the IBM Cloud services which are beingused in completing this project.	4	Medium
Sprint-2	IBM Watson IoT platform	US-3	IBM Watson IoT platform acts as the mediator to connect the web application to IoT devices, so create the IBM Watson IoT platform.	5	Medium
Sprint-2	IBM Watson IoT platform	US-4	In order to connect the IoT device to the IBMcloud, create a device in the IBM Watson IoT platform and get the device credentials.	5	High
Sprint-3	IBM Watson IoT platform & Node-REDservice	US-1	Configure the connection security and create APIkeys that are used in the Node-RED service for accessing the IBM IoT Platform.	10	High

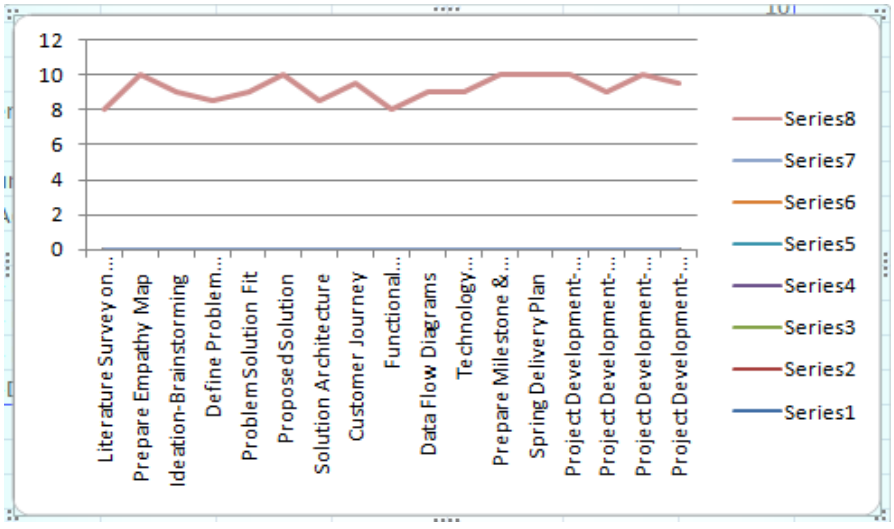
PROJECT TRACKER:

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

Velocity:
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) periteration unit (story points per day)

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

Burndown Chart:
A burndown chart is a graphical representation of work left to do versus time
However, burndown charts can be applied to any project containing measurable progress overtime.



7. PROJECT DEVELOPMENT PHASE

7.1 Project Development - Delivery of Sprint – 1:

CODE:

CODE:

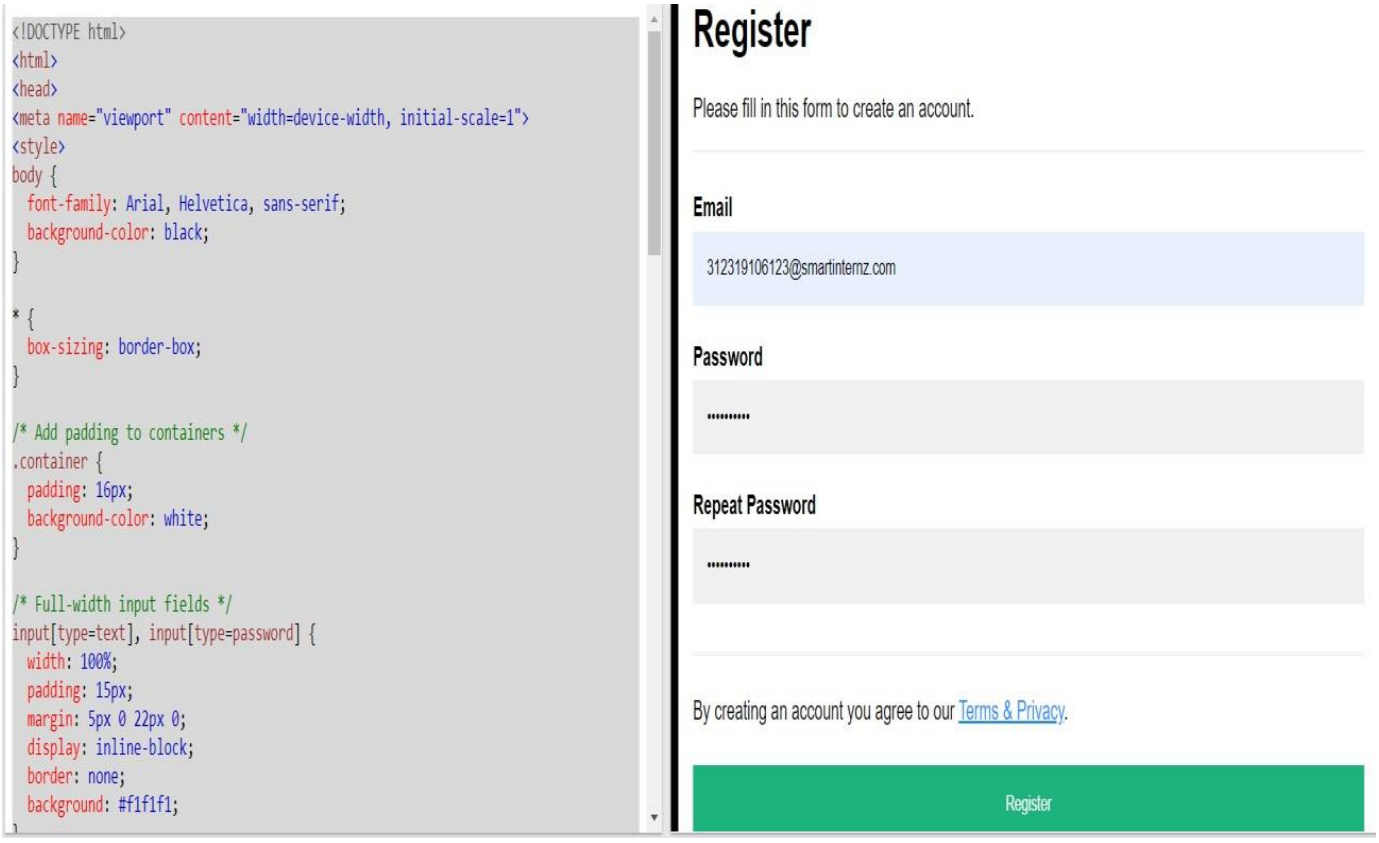
```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style> body {
font-family: Arial, Helvetica, sans-serif; background-color: black;
}

* {
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>


<!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 Project Development - Delivery of Sprint – 2:

ALGORITHM:

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

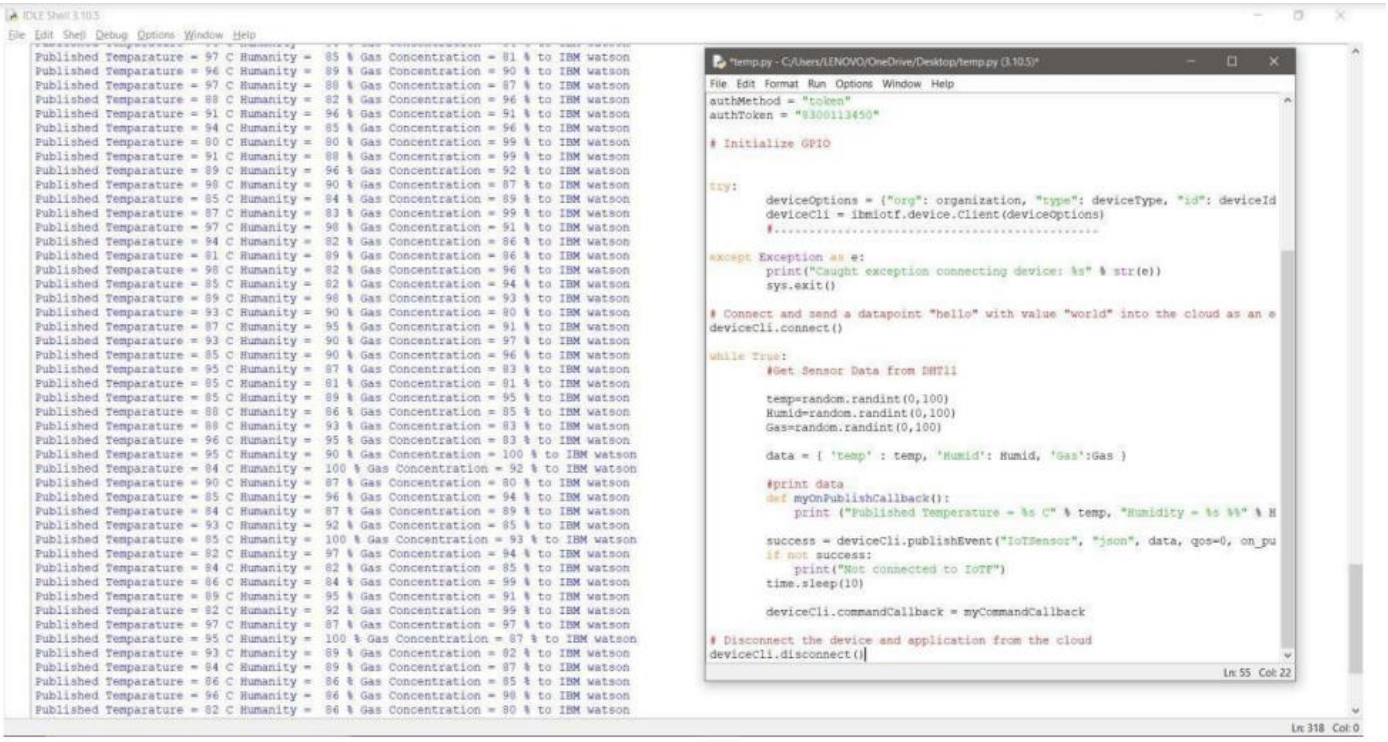
```
#include <dht.h>

#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
}
```

7.3 Project Development - Delivery of 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:

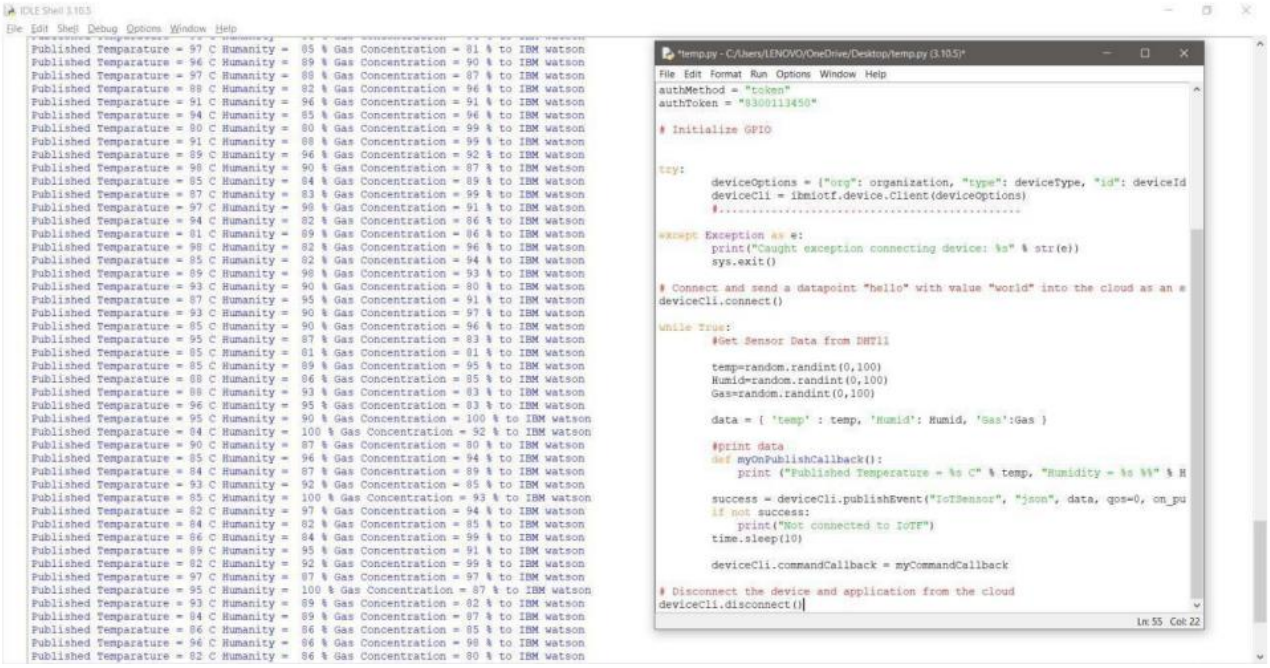


7.4 Project Development - Delivery of Sprint – 4:

```
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 :



8. CONCLUSION

We hope to gain hands-on experience with the trending technologies of "Embedded System" and "Internet of Things" through this project. IoT-enabled industrial monitoring systems have become increasingly popular in a variety of industries because they improve safety standards by providing real-time monitoring of critical parameters such as temperature, humidity, and smoke, as well as alerting officials and workers regularly. The implementation is not only for safety reasons, but it also has the potential to increase industry yields. In our project, the Internet of Things (IoT) is used to collect data and communicate through the internet. We hope that our project will be beneficial enough to be implemented in industries across India, saving lives and property from accidents and risks that are often overlooked by industry personnel and users. Companies in the industrial and logistics sectors can better meet the new era of instant needs by utilizing the Industrial Internet of Things (IoT).

9. REFERENCE

Integrating IoT technologies for an “intelligent” safety management in the process industry Maria Grazia Gnoni , Paolo Angelo Bragatto, Maria Francesca Milazzo, Roberto Setola Procedia manufacturing 42, 511-515, 2020.

IoT sensor-based BIM system for smart safety barriers of hazardous energy in petrochemical construction Lieyun Ding, Weiguang Jiang, Cheng Zhou Frontiers of Engineering Management 9 (1), 1-15, 2022.

Production Plant and Warehouse Automation with IoT and Industry 5.0 Zainab Fatima, Muhammad Hassan Tanveer, Shehnila Zardari, Laviza Falak Naz, Hina Khadim, Noorah Ahmed, Midha Tahir Applied Sciences 12 (4), 2053, 2022.

Explosion protection for industrial wireless networks: Wireless Ethernet in hazardous areas works with explosion-proof enclosures. Kerstin Wolf Control Engineering 67 (10), 24-26, 2020.

IoT enabled environmental toxicology for air pollution monitoring using AI techniques P Asha, LBTJRRGS Natrayan, BT Geetha, J Rene Beulah, R Sumathy, G Varalakshmi, S Neelakandan Environmental Research 205, 112574, 2022.