**Project Report Format**

|  |  |
| --- | --- |
| Date | 20 May 2023 |
| Team ID | NM2023TMID09251 |
| Project Name | Industrial Workers Health and Safety System based on Internet of Things |

1. **INTRODUCTION**

1.1 Project Overview

1.2 Purpose

2. **IDEATION & PROPOSED SOLUTION**

2.1 Problem Statement Definition

2.2 Empathy Map Canvas

2.3 Ideation & Brainstorming

2.4 Proposed Solution

3. **REQUIREMENT ANALYSIS**

3.1 Functional requirement

3.2 Non-Functional requirements

**4. PROJECT DESIGN**

4.1 Data Flow Diagrams

4.2 Solution & Technical Architecture

4.3 User Stories

**5. CODING & SOLUTIONING (Explain the features added in the project along with code)**

5.1 Feature 1

5.2 Feature 2

**6. RESULTS**

6.1 Performance Metrics

**7. ADVANTAGES & DISADVANTAGES**

**8. CONCLUSION**

**9. FUTURE SCOPE**

**10. APPENDIX**

Source Code

1. **INTRODUCTION :**

**1.1** Project Overview:

With the help of sensors attached to the shoes of the workers, the information such as the temperature measure, altitude parameters, the total distance walked is recorded and sent to the cloud for storage. This data will be visualised in the mobile application and through a web application,  the authorities can check every worker's status. If someone is working on the higher altitudes, important and required precautions are sent to them, thus increasing their safety.

**1.2** Purpose:

In some industrial areas and coal mines there are many things which are need to be monitored in order to take some safety measures when there are some critical conditions.These parameters are sent to the available beacon scanners in ever place through BLE technology.From beacon scanners the entire data is sent to the cloud. This data will be visualized in the mobile application and through web app the authorities can check every workers status and if someone is working at higher altitudes he may alert him with necessary precaution

1. **IDEATION & PROPOSED SOLUTION**:

2.1 Problem Statement Definition:

Create a problem statement to understand your customer's point of view. The Customer Problem Statement template helps you focus on what matters to create experiences people will love.

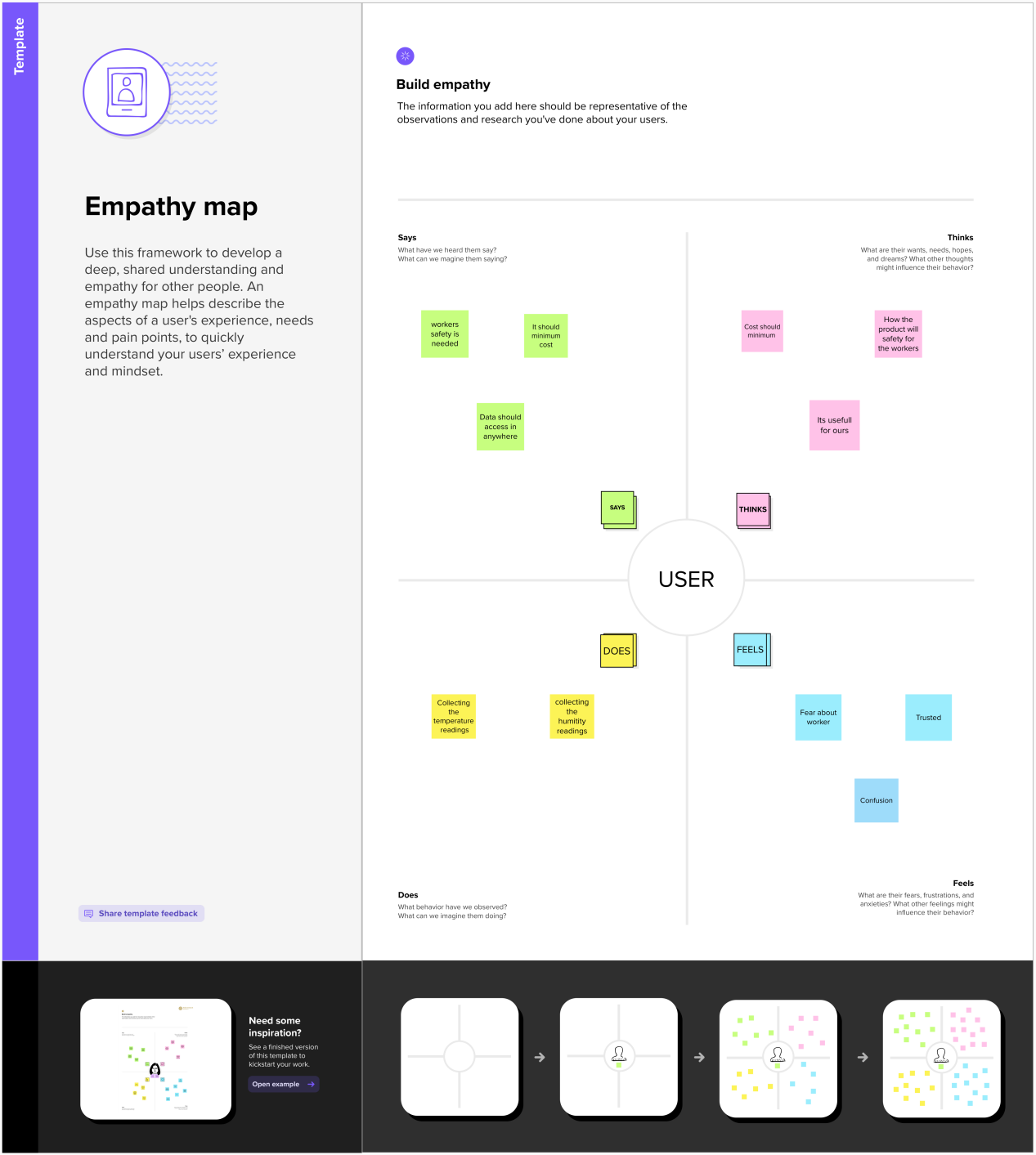
A well-articulated customer problem statement allows you and your team to find the ideal solution for the challenges your customers face. Throughout the process, you’ll also be able to empathize with your customers, which helps you better understand how they perceive your product or service.

Graphical user interface, text, application, email

Description automatically generated

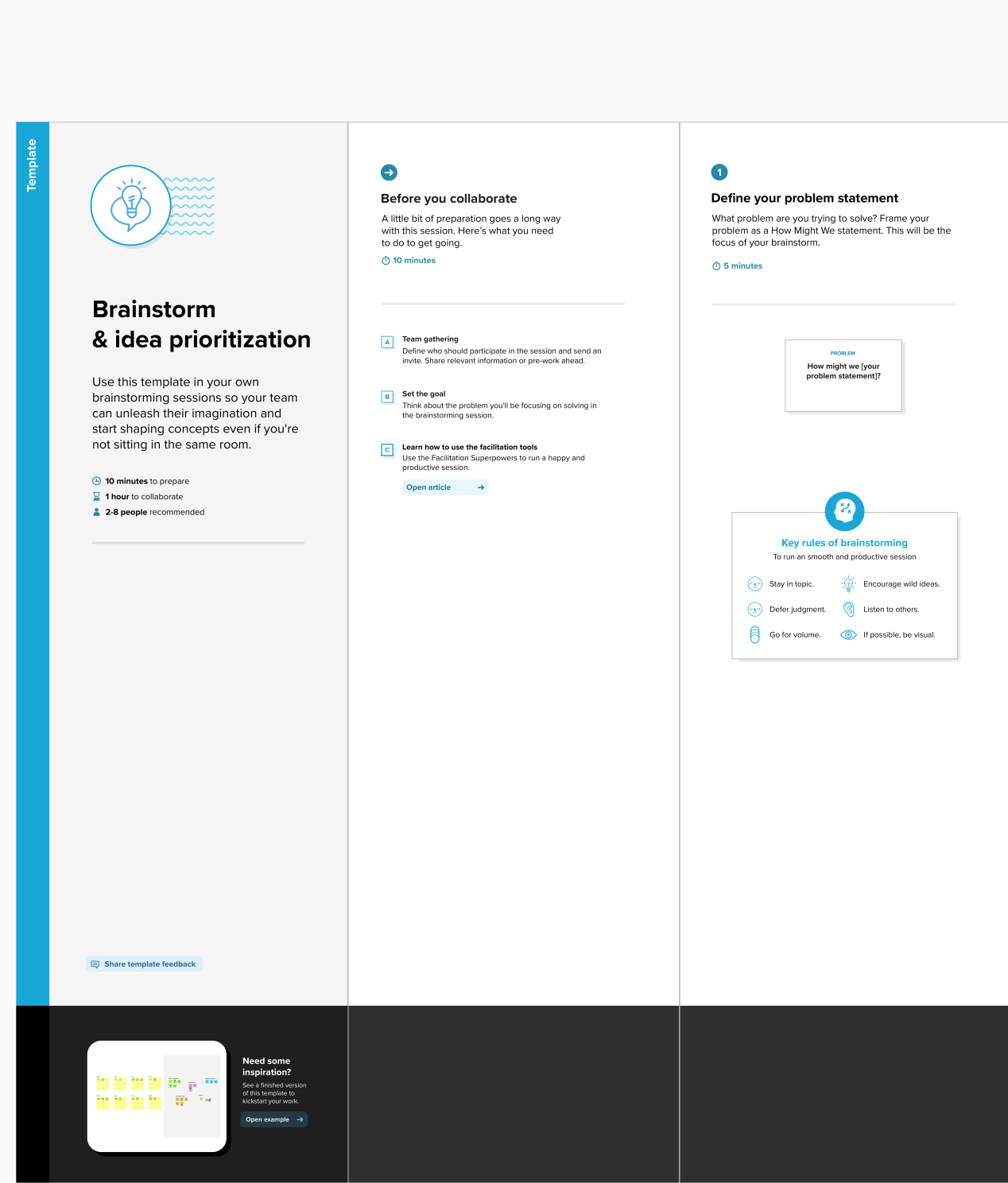
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Problem Statement (PS)** | **I am (Customer)** | **I’m trying to** | **But** | **Because** | **Which makes me feel** |
| PS-1 | owner | To monitor the workers safety | I need devices to monitor the workers safety | I can easly monitor in anywhere at 24/7 hours | fear |

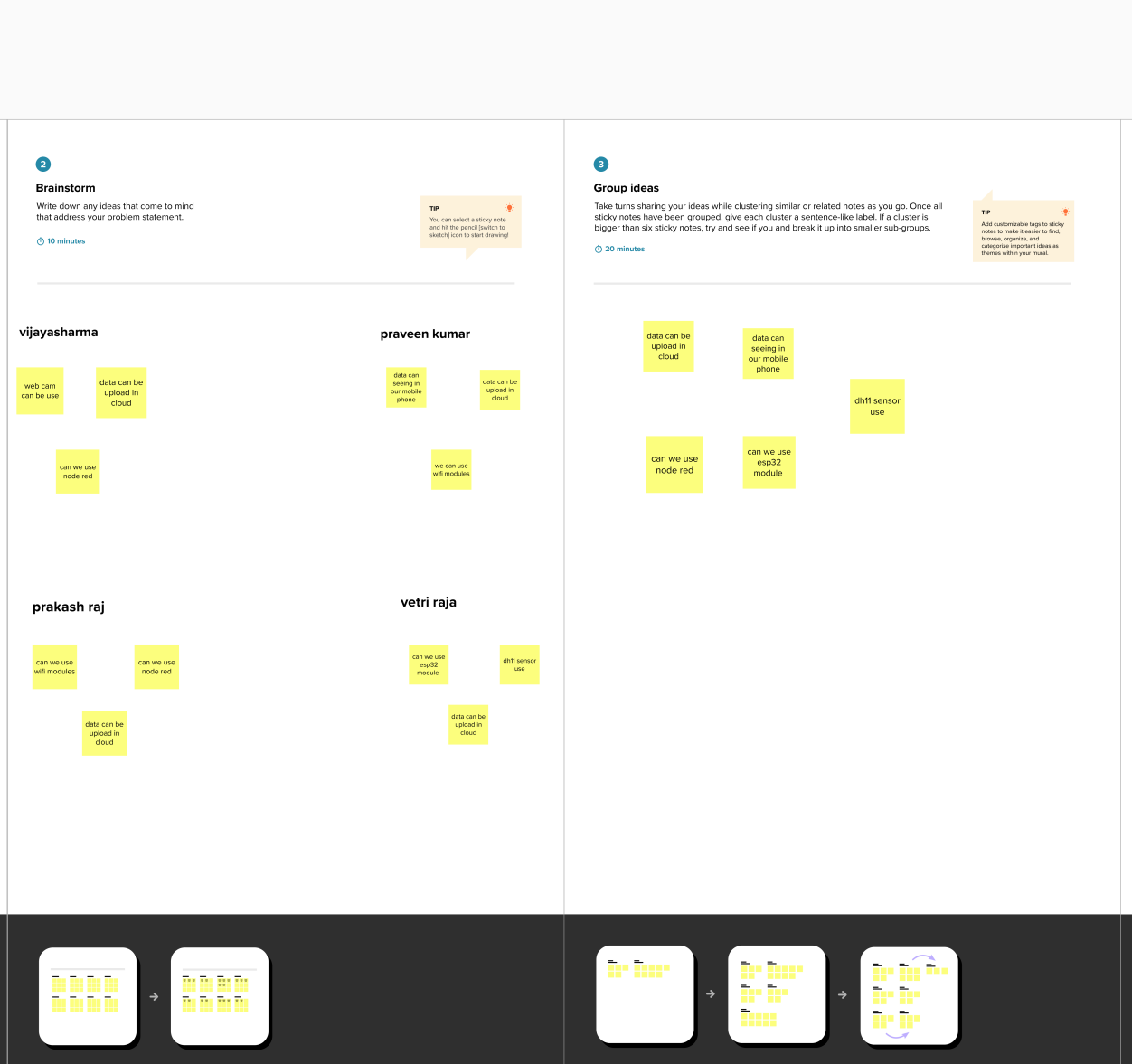
2.2 Empathy Map Canvas:

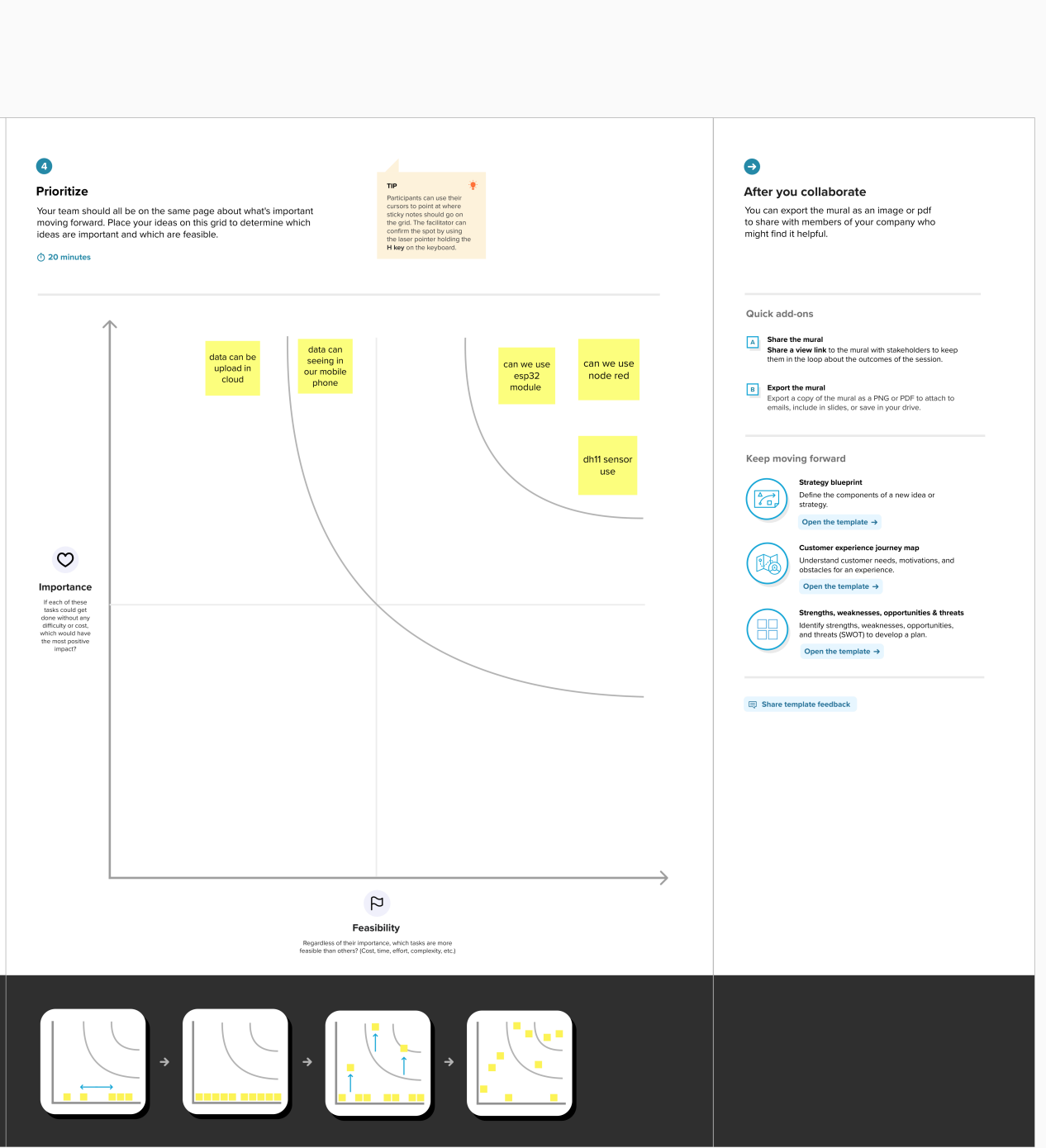


Referance: https://app.mural.co/t/iot7678/m/iot7678/1684331861410/35ef3e04525d1bb88c111d8b9740c502d12c0b8c?sender=u3123f66c8abe1ac662872156

2.3 Ideation & Brainstorming







Referance:

https://app.mural.co/t/iot7678/m/iot7678/1684335143201/3a603a676d5118b6b68411d21bf12fd3e2760e0a?sender=u3123f66c8abe1ac662872156

2.4 Proposed Solution:

Project team shall fill the following information in proposed solution template.

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Parameter** | **Description** |
|  | Problem Statement (Problem to be solved) | Workers health issues |
|  | Idea / Solution description | To check the temperature and humidity to safe the workers |
|  | Novelty / Uniqueness | Using iot devices |
|  | Social Impact / Customer Satisfaction | Gives more health consciousness,Easy implements iot device on low cast |
|  | Business Model (Revenue Model) | It has moderate business profit but very use workers |
|  | Scalability of the Solution | In iot easy to implement |

1. **REQUIREMENT ANALYSIS:**

3.1 Functional requirement

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 Gmail |
| FR-2 | User Confirmation | Confirmation via Email  Confirmation via OTP |

3.2 Non-Functional requirements:

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

|  |  |  |
| --- | --- | --- |
| **FR No.** | **Non-Functional Requirement** | **Description** |
| NFR-1 | **Usability** | Mobile support. Users must be able to interact in the same roles and on the same tasks on computers and mobile devices where practical, given mobile capabilities. |
| NFR-2 | **Security** | Data security. All persisted data requires secure access.Device security. Devices must register and communicate securely, such as by using transport layer security (TLS). Unauthorized devices are prohibited. |
| NFR-3 | **Reliability** | Its easy to use |
| NFR-4 | **Performance** | Function offload to the edge. The system must have a centralized configuration with offload to the edge for rules-based decision-making and streaming analytics.Hybrid cloud. It must be possible to deploy the platform layer on premises, in the cloud, or part on premises and part in the cloud. |
| NFR-5 | **Availability** | High availability (HA). Some IoT solutions and domains demand highly available systems for 24x7 operations. That said, this type of system isn't a critical production application, which means that operations or production don't go down if the IoT solution is down |
| NFR-6 | **Scalability** | Horizontal scalability. The system must handle expanding load and data retention needs that are based on the upscaling of the solution scope, such as extra manufacturing facilities and extra buildings. |

1. **PROJECT DESIGN:**

4.1 Data Flow Diagrams:

**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.

**Architecture**

Gain knowledge of Watson IoT Platform

Explore Wokwi Platform

Explore the devices and its simulation of the wokwi platform.

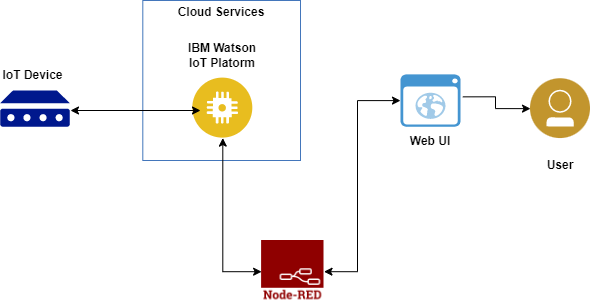
Explore the libraries present in Wokwi.

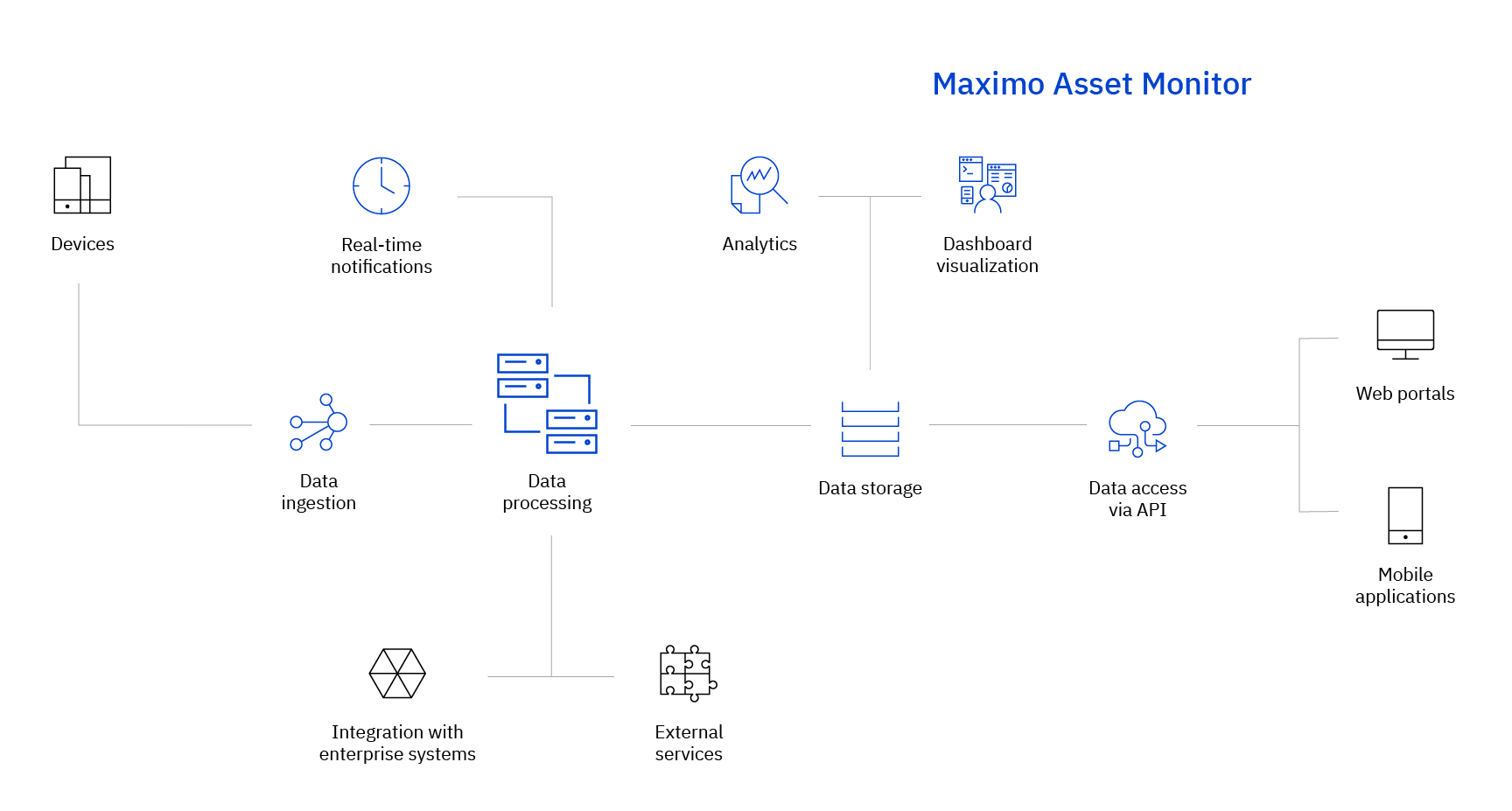
Will be able to code to connect the devices across the cloud platform.

Connecting the devices on wokwi to the IoT platform device to exchange the sensor data.

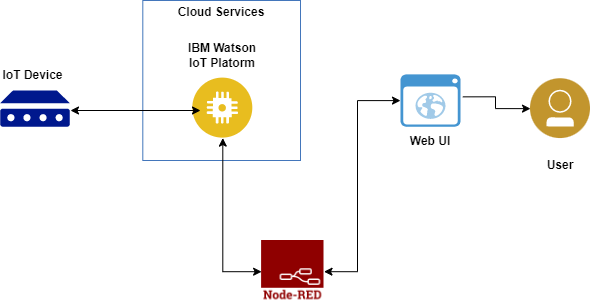
Gain knowledge of creating devices and platforms using IBM Watson IoT Platform.

Gain knowledge of web application development through node-red.

****



4.2 Solution & Technical Architecture:

****

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Component** | **Description** | **Technology** |
|  | User Interface | How user interacts with application e.g.  Web UI, Mobile App, Chatbot etc. | HTML, CSS, JavaScript / Angular Js / React Js etc. |
|  | Application Logic-1 | Logic for a process in the application | Java / Python |
|  | Application Logic-2 | Logic for a process in the application | IBM Watson STT service |
|  | Application Logic-3 | Logic for a process in the application | IBM Watson Assistant |
|  | Database | Data Type, Configurations etc. | MySQL, NoSQL, etc. |
|  | Cloud Database | Database Service on Cloud | IBM DB2, IBM Cloudant etc. |
|  | File Storage | File storage requirements | IBM Block Storage or Other Storage Service or Local Filesystem |
|  | External API-1 | Purpose of External API used in the application | IBM Weather API, etc. |
|  | External API-2 | Purpose of External API used in the application | Aadhar API, etc. |
|  | Machine Learning Model | Purpose of Machine Learning Model | Object Recognition Model, etc. |
|  | 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** |
| --- | --- | --- | --- |
|  | Open-Source Frameworks | List the open-source frameworks used | Technology of Opensource framework |
|  | Security Implementations | List all the security / access controls implemented, use of firewalls etc. | e.g. SHA-256, Encryptions, IAM Controls, OWASP etc. |
|  | Scalable Architecture | Justify the scalability of architecture (3 – tier, Micro-services) | Technology used |
|  | Availability | Justify the availability of application (e.g. use of load balancers, distributed servers etc.) | Technology used |
|  | Performance | Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN’s) etc. | Technology used |

4.3 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** | **Team Member** |
| --- | --- | --- | --- | --- | --- | --- |
| Customer | Registration | USN-1 | As a user, I can register for the application by entering my email, password, and confirming my password. | I can access my account / dashboard | High | Prakash raj |
|  |  | USN-2 | As a user, I will receive confirmation email once I have registered for the application | I can receive confirmation email & click confirm | High | Praveen kumar |
|  |  | USN-3 | As a user, I can register for the application through Gmail |  | Medium | vijayasharma |
|  | Login | USN-4 | As a user, I can log into the application by entering email & password |  | High | Vetri raja |
|  | Dashboard | USN-5 | User can the see data of the devices from cloud |  | high | Prakash raj |

**5. CODING & SOLUTIONING:**

5.1 Feature 1:

A temperature sensor is an electronic device that measures the temperature of its environment and converts the input data into electronic data to record, monitor, or signal temperature changes. There are many different types of temperature sensors. Some temperature sensors require [direct contact](https://www.electronics-tutorials.ws/io/io_3.html) with the physical object that is being monitored (contact temperature sensors), while others indirectly measure the temperature of an object (non-contact temperature sensors).

Code:

industrial workers health and safety system based on IoT

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

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

#include "DHT.h"// Library for dht11

#define DHTPIN 4     // what pin we're connected to

#define DHTTYPE DHT11   // define type of sensor DHT 11

//#define LED 5

DHT dht (DHTPIN, DHTTYPE);//// creating the instance by passing pin and typr of dht connected

void setup()// configureing the ESP32

{

**Serial**.begin(115200);

  dht.begin();

  //pinMode(LED,OUTPUT);

  delay(10);

**Serial**.println();

  wificonnect();

  mqttconnect();

}

void loop()// Recursive Function

{

  //h = dht.readHumidity();

  p= random(60,150);

  t = dht.readTemperature();

**Serial**.print("temperature:");

**Serial**.println(t);

**Serial**.print("Pulse:");

**Serial**.println(p);

PublishData(t, p);

  delay(1000);

  if (!client.loop()) {

    mqttconnect();

  }

}

dig

5.2 Feature 2 :

[Cloud computing](https://www.oracle.com/cloud/what-is-cloud-computing/) benefits organizations in many ways. In fact, the benefits are so numerous that it makes it almost impossible not to consider moving business operations to a cloud-based platform. And yet many organizations rely on outdated and inefficient processes because they don’t understand the benefits.

We using cloud to access a data anywhere in the world, having the cloud account they gives a token id, device name, token authentication id etc, if you wants to see a data authorized persons only can see data.

code:

//-------credentials of IBM Accounts------

#define ORG "96ei56"//IBM ORGANITION ID

#define DEVICE\_TYPE "worker"//Device type mentioned in ibm watson IOT Platform

#define DEVICE\_ID "0104"//Device ID mentioned in ibm watson IOT Platform

#define TOKEN "04012023"     //Token

String data3;

float t;

int p;

/-------- Customise the above values --------

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Name

char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of event perform and format in which data to be send

char subscribetopic[] = "iot-2/cmd/test/fmt/String";// cmd  REPRESENT command type AND COMMAND IS TEST OF FORMAT STRING

char authMethod[] = "use-token-auth";// authentication method

char token[] = TOKEN;

char clientId[] = "d:" ORG ":" DEVICE\_TYPE ":" DEVICE\_ID;//client id

//-----------------------------------------

WiFiClient wifiClient; // creating the instance for wificlient

PubSubClient client(server, 1883, callback ,wifiClient); //calling the predefined client id by passing parameter like server id,portand wificredential

/\*.....................................retrieving to Cloud...............................\*/

void PublishData(float temp, int pulse) {

  mqttconnect();//function call for connecting to ibm

  /\*

     creating the String in in form JSon to update the data to ibm cloud

  \*/

  String payload = "{\"temperature\":";

  payload += temp;

  payload += "," "\"Pulse\":";

  payload += pulse;

  payload += "}";

**Serial**.print("Sending payload: ");

**Serial**.println(payload);

  if (client.publish(publishTopic, (char\*) payload.c\_str())) {

**Serial**.println("Publish ok");// if it sucessfully upload data on the cloud then it will print publish ok in Serial monitor or else it will print publish failed

  } else {

**Serial**.println("Publish failed");

  }

}

void mqttconnect() {

  if (!client.connected()) {

**Serial**.print("Reconnecting client to ");

**Serial**.println(server);

    while (!!!client.connect(clientId, authMethod, token)) {

**Serial**.print(".");

      delay(500);

    }

     initManagedDevice();

**Serial**.println();

  }

}

void wificonnect() //function defination for wificonnect

{

**Serial**.println();

**Serial**.print("Connecting to ");

  WiFi.begin("Wokwi-GUEST", "", 6);//passing the wifi credentials to establish the connection

  while (WiFi.status() != WL\_CONNECTED) {

    delay(500);

**Serial**.print(".");

  }

**Serial**.println("");

**Serial**.println("WiFi connected");

**Serial**.println("IP address: ");

**Serial**.println(WiFi.localIP());

}

void initManagedDevice() {

  if (client.subscribe(subscribetopic)) {

**Serial**.println((subscribetopic));

**Serial**.println("subscribe to cmd OK");

  } else {

**Serial**.println("subscribe to cmd FAILED");

  }

}

void callback(char\* subscribetopic, byte\* payload, unsigned int payloadLength)

{

**Serial**.print("callback invoked for topic: ");

**Serial**.println(subscribetopic);

  for (int i = 0; i < payloadLength; i++) {

    //Serial.print((char)payload[i]);

    data3 += (char)payload[i];

  }

**Serial**.println("data: "+ data3);

  /\*

  if(data3=="lighton")

  {

Serial.println(data3);

digitalWrite(LED,HIGH);

  }

  else

  {

Serial.println(data3);

digitalWrite(LED,LOW);

  }

  \*/

data3="";

}

1. **RESULTS:**

6.1 Performance Metrics:

Gain knowledge of Watson IoT Platform

Explore Wokwi Platform

Explore the devices and its simulation of the wokwi platform.

Explore the libraries present in Wokwi.

Will be able to code to connect the devices across the cloud platform.

Connecting the devices on wokwi to the IoT platform device to exchange the sensor data.

Gain knowledge of creating devices and platforms using IBM Watson IoT Platform.

Gain knowledge of web application development through node-red.

**7. ADVANTAGES & DISADVANTAGES:**

Advantage:

The Internet of Things (IoT) is a network of physical objects that are embedded with sensors and other technologies to share data with other devices over the Internet. Evolution took place due to the convergence of multiple technologies, machine learning, embedded systems, commodity sensors, and real-time analytic s. Traditional wireless sensor networks, control systems, and others enable the Internet of Things.

[IoT in a working environment](https://www.iotforall.com/working-during-a-pandemic-what-organizations-can-do-with-iot" \t "https://www.iotforall.com/_blank) is one of the most advanced ways to increase safety that a company could implement. It has managed to help companies worldwide by improving working conditions, data collection, streamlining operations, and increasing productivity.

The introduction of training programs for employees would be an added advantage. It would enable identifying any potential threats to their safety. Therefore, it allows the implementation of IoT, which would monitor and maintain records and a safe working environment.

Disadvantage:

Privacy of Data. Privacy is the biggest challenge with IoT, as all the connected devices transfer data in real-time. Personal data can be hacked if this end to end connection is not secure. Accuracy. Accuracy issues may come due to handling such massive data in real-time,Cost,Data siphoning. The transmission of information by endpoint devices can be intercepted via an eavesdropping attack. Device hijacking. Distributed denial-of-service.

**8. CONCLUSION**

Smart offices offer several growth opportunities to businesses. From speeding up work with office automation using IoT to efficient energy consumption, organizations can gain value. It is up to you to decide if the benefits it offers outweigh the concerns. Regardless, rapid technological advancement is certain. Organizations that adopt smart offices now might just stand a chance of future-proofing their business and gaining a competitive edge.If you need IoT smart office solutions for your company, [contact Relevant](https://relevant.software/blog/iot-in-workplace/" \l "contact-us). We can help you with the professional IoT development services you need for your project. IoT has changed the way people look at healthcare and hospitality systems. Adoption of these technologies will improve product development and will bring a long-term effect by bringing those minor change.The Internet of Things (IoT) has the potential to make workplaces safer by providing real-time monitoring, predictive maintenance, safety alerts, remote access control, location tracking, inventory management, wearable devices, vehicle tracking, process automation, safety data analysis, and data collection.

1. **FUTURE SCOPE:**

**Safety report management:** It is the duty of the workers to observe all the recommended rules. Any faulty equipment, improper maintenance and any suspicion of breaking the safety law must be reported and documented. A safety manager should be held responsible to maintain and keep all the safety report up-to-date.

**Financial Responsibility:** It is a legal requirement for employers to have insurance and also to plan for uninsured costs with their consequences. This will reduce the financial liability towards loss of property, equipment and productivity.

**Important considerations for Industrial Safety:**

* Health & Safety Management Systems
* Managing Risk
* Risk Assessment
* Monitoring & measuring Health & Safety
* Manage workplace health & safety

1. **APPENDIX:**

**Source code:**

industrial workers health and safety system based on IoT

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

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

#include "DHT.h"// Library for dht11

#define DHTPIN 4     // what pin we're connected to

#define DHTTYPE DHT11   // define type of sensor DHT 11

//#define LED 5

DHT dht (DHTPIN, DHTTYPE);//// creating the instance by passing pin and typr of dht connected

void callback(char\* subscribetopic, byte\* payload, unsigned int payloadLength);

//-------credentials of IBM Accounts------

#define ORG "96ei56"//IBM ORGANITION ID

#define DEVICE\_TYPE "worker"//Device type mentioned in ibm watson IOT Platform

#define DEVICE\_ID "0104"//Device ID mentioned in ibm watson IOT Platform

#define TOKEN "04012023"     //Token

String data3;

float t;

int p;

//-------- Customise the above values --------

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Name

char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of event perform and format in which data to be send

char subscribetopic[] = "iot-2/cmd/test/fmt/String";// cmd  REPRESENT command type AND COMMAND IS TEST OF FORMAT STRING

char authMethod[] = "use-token-auth";// authentication method

char token[] = TOKEN;

char clientId[] = "d:" ORG ":" DEVICE\_TYPE ":" DEVICE\_ID;//client id

//-----------------------------------------

WiFiClient wifiClient; // creating the instance for wificlient

PubSubClient client(server, 1883, callback ,wifiClient); //calling the predefined client id by passing parameter like server id,portand wificredential

void setup()// configureing the ESP32

{

**Serial**.begin(115200);

  dht.begin();

  //pinMode(LED,OUTPUT);

  delay(10);

**Serial**.println();

  wificonnect();

  mqttconnect();

}

void loop()// Recursive Function

{

  //h = dht.readHumidity();

  p= random(60,150);

  t = dht.readTemperature();

**Serial**.print("temperature:");

**Serial**.println(t);

**Serial**.print("Pulse:");

**Serial**.println(p);

  PublishData(t, p);

  delay(1000);

  if (!client.loop()) {

    mqttconnect();

  }

}

/\*.....................................retrieving to Cloud...............................\*/

void PublishData(float temp, int pulse) {

  mqttconnect();//function call for connecting to ibm

  /\*

     creating the String in in form JSon to update the data to ibm cloud

  \*/

  String payload = "{\"temperature\":";

  payload += temp;

  payload += "," "\"Pulse\":";

  payload += pulse;

  payload += "}";

**Serial**.print("Sending payload: ");

**Serial**.println(payload);

  if (client.publish(publishTopic, (char\*) payload.c\_str())) {

**Serial**.println("Publish ok");// if it sucessfully upload data on the cloud then it will print publish ok in Serial monitor or else it will print publish failed

  } else {

**Serial**.println("Publish failed");

  }

}

void mqttconnect() {

  if (!client.connected()) {

**Serial**.print("Reconnecting client to ");

**Serial**.println(server);

    while (!!!client.connect(clientId, authMethod, token)) {

**Serial**.print(".");

      delay(500);

    }

     initManagedDevice();

**Serial**.println();

  }

}

void wificonnect() //function defination for wificonnect

{

**Serial**.println();

**Serial**.print("Connecting to ");

  WiFi.begin("Wokwi-GUEST", "", 6);//passing the wifi credentials to establish the connection

  while (WiFi.status() != WL\_CONNECTED) {

    delay(500);

**Serial**.print(".");

  }

**Serial**.println("");

**Serial**.println("WiFi connected")

**Serial**.println("IP address: ");

**Serial**.println(WiFi.localIP());

}

void initManagedDevice() {

  if (client.subscribe(subscribetopic)) {

**Serial**.println((subscribetopic));

**Serial**.println("subscribe to cmd OK");

  } else {

**Serial**.println("subscribe to cmd FAILED");

  }

}

void callback(char\* subscribetopic, byte\* payload, unsigned int payloadLength)

{

**Serial**.print("callback invoked for topic: ");

**Serial**.println(subscribetopic);

  for (int i = 0; i < payloadLength; i++) {

    //Serial.print((char)payload[i]);

    data3 += (char)payload[i];

  }

**Serial**.println("data: "+ data3);

  /\*

  if(data3=="lighton")

  {

Serial.println(data3);

digitalWrite(LED,HIGH);

  }

  else

  {

Serial.println(data3);

digitalWrite(LED,LOW);

  }

  \*/

data3="";

}