

# **Project Report**

**Project Name: SIGNS WITH SMART CONNECTIVITY FOR BETTER ROAD SAFE**

**Team ID: PNT2022TMID51479**

**TEAM LEADER : MUTHULEKSHMI P**

**TEAM MEMBER : VINI SHALINI M**

**SARAL N**

**ANNS A**

## **I INTRODUCTION**

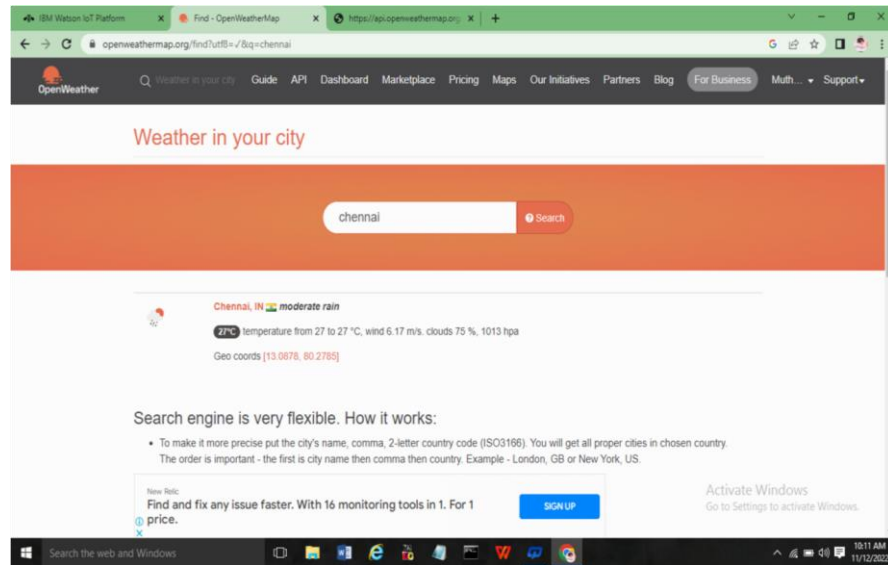
### **1.1 Project Overview**

In present Systems the road signs and the speed limits are Static. But the road signs can be changed in some cases. We can consider some cases when there are some road diversions due to heavy traffic or due to accidents then we can change the road signs accordingly if they are digitalized. This project proposes a system which has digital sign boards on which the signs can be changed dynamically. If there is rainfall then the roads will be slippery and the speed limit would be decreased. There is a web app through which you can enter the data of the road diversions, accident prone areas and the information sign boards can be entered through web app. This data is retrieved and displayed on the sign boards accordingly.

### **1.2 Purpose**

The purpose of this project is to report and get relieved from the issues related to road safety.

## **1.OPEN WEATHER MAP**



## 2. LITERATURE SURVEY

### 2.1 Existing problem

- Climate changes
- the non-use of motorcycle helmets, seat-belts, and child restraint,
- not so clear vision of sign boards
- distraction, including the use of mobile phones, leading to impaired driving,
- unsafe vehicles and unsafe road infrastructure can negatively impact safety on the roads,
- inadequate post-crash care,

### 2.2 References

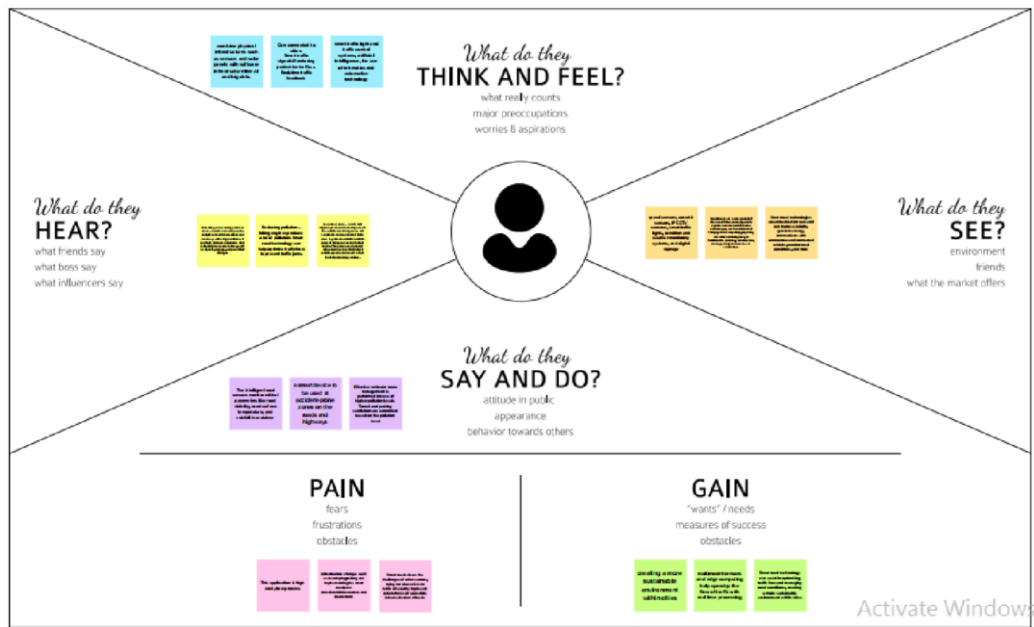
1. Karnadi, F. K., Zhi, H. M., & Kun-chan, L., "Rapid Generation of Realistic Mobility Models for VANET", Wireless Communications and Networking Conference, WCNC2007.
2. General safety features by Transport Canada, 2007.
3. Shulman, M., & Deering, R., Vehicle Safety Communications in the United States, 2007.
4. Yang, X., Liu, J., Zhao, F., & Vaidya, N (2004), Vehicle-to-Vehicle Communication Protocol for Cooperative Collision Warning.
5. Kurt Dresner & Peter Stone (2008), Replacing the Stop Sign: Unmanaged Intersection Control, pp.94-101, Estoril, Portugal.
6. ARIB STD-T75, 2001 Zing Zhu, Sumit Roy. Dedicated Short Range Communication System.

## 2.3 Problem Statement Definition

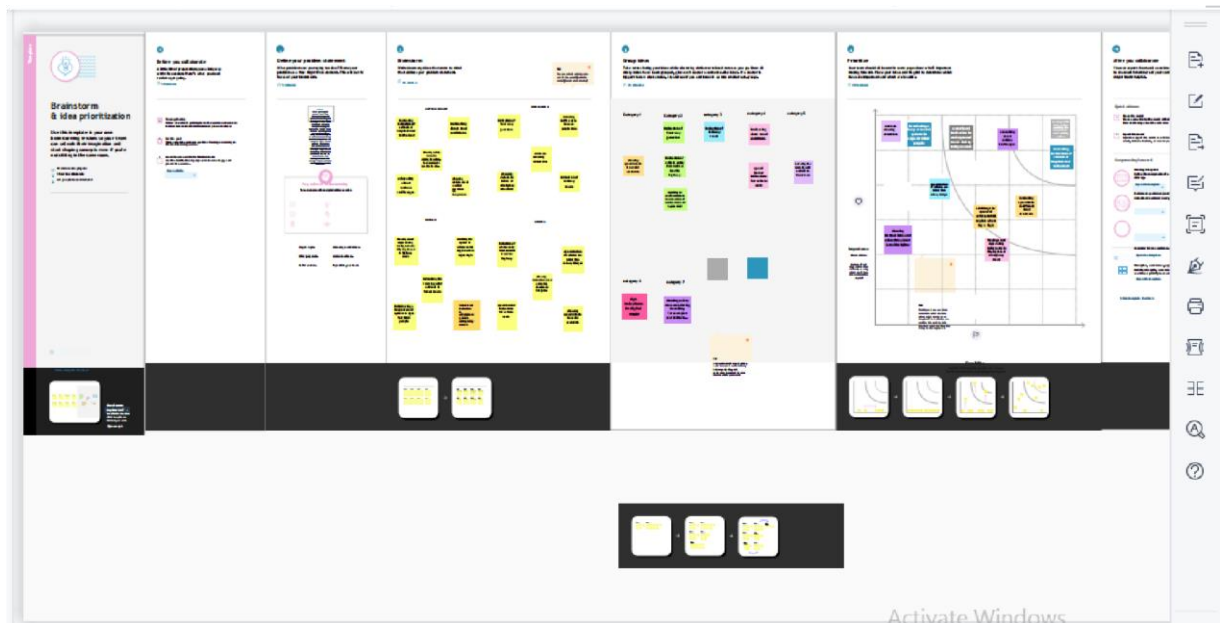
Smart Solutions for road safety are designed to reduce the risk for the user.

### 3. IDEATION & PROPOSED SOLUTION

### 3.1 Empathy Map Canvas



### 3.2 Ideation & Brainstorming



### 3.3 Proposed Solution

**Project Design Phase-I  
Proposed Solution**

Date	19 September 2022
Team ID	PNT2022TMD51479
Project Name	SIGNS WITH SMART CONNECTIVITY FOR BETTER ROAD SAFETY
Maximum Marks	2 Marks

**Proposed Solution Template:**

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	<ul style="list-style-type: none"> <li>The actual problem is that drivers are unable to know whether the road condition is safe to travel or not.</li> <li>Hence there will be a need of guidance to provide the safety and to avoid the inconvenience to reach the destination.</li> </ul>
2.	Idea / Solution description	<ul style="list-style-type: none"> <li>This problem can be overcome by using rain drop sensor to indicate there is rain is occurs or not.</li> <li>Using GPRS and IR sensor with camera to sense the traffic in dark areas.</li> <li>Collecting information from the local</li> </ul>

		peoples and the decisions are made by the controller who control the display manually.
3.	Novelty / Uniqueness	<ul style="list-style-type: none"> <li>Digital sign boards are convey the information to the driver using Embedded and IOT technology.</li> <li>Speed limit changes according to the weather condition using rain drop sensor.</li> </ul>
4.	Social Impact / Customer Satisfaction	<ul style="list-style-type: none"> <li>The proposed system provides many facilities which helps the drivers to maintain the safety.</li> <li>Signs change based on upcoming events.</li> </ul>
5.	Business Model (Revenue Model)	<ul style="list-style-type: none"> <li>This prototype can be developed as product with minimum cost with high performance.</li> <li>Reduces manpower.</li> </ul>
6.	Scalability of the Solution	<ul style="list-style-type: none"> <li>User friendly.</li> <li>Easy to access the data from the source.</li> <li>Information in sign boards can be easily captured.</li> </ul>

### 3.4 Problem Solution fit

Problem-Solution Fit canvas			Purpose / Vision	Version
1. CUSTOMER SEGMENT(S) <b>CS</b>	2. PROBLEMS / PAINS - ITS FREQUENCY <b>PR</b>	3. TRIGGERS TO ACT <b>TR</b>	4. CUSTOMER LIMITATIONS <b>CL</b>	5. AVAILABLE SOLUTIONS <b>AS</b>
> Automobiles	> Geo fencing (for younger generation). > Vehicle to vehicle communication. > Remote parking.	> Excessive accidents. > Reading about a more efficient solution in the news.	Lack of knowledge, budget, no cash, network connection, lack of available devices.	> Limited speed > Follow rule and regulations > Vehicles maintenance
6. CUSTOMER LIMITATIONS <b>CL</b>	7. PROBLEM ROOT / CAUSE <b>RC</b>	8. BEHAVIOR - ITS INTENSITY <b>BE</b>	9. YOUR SOLUTION <b>SL</b>	10. CHANNELS of BEHAVIOR <b>CH</b>
> High speed > Lack of time management	> Efficient > Low cost > High performance	> Our solution is based on IOT we where plan to use GPS tracking with digital display and voice display. > It use us to solve our required problem.	> Geo fencing (for younger generation). > Vehicle to vehicle communication. > Remote parking.	> Excessive accidents. > Reading about a more efficient solution in the news.
11. TRIGGERS TO ACT <b>TR</b>	12. EMOTIONS BEFORE / AFTER <b>EM</b>	13. YOUR SOLUTION <b>SL</b>	14. CHANNELS of BEHAVIOR <b>CH</b>	15. EMOTIONS BEFORE / AFTER <b>EM</b>
> Excessive accidents. > Reading about a more efficient solution in the news.	> Time delay because of traffic and suffered a lot. > Time management and update and secure life.	> Our solution is based on IOT we where plan to use GPS tracking with digital display and voice display. > It use us to solve our required problem.	> The program should be error free > The apparatus we require should be affordable	> Excessive accidents. > Reading about a more efficient solution in the news.

## 4. REQUIREMENT ANALYSIS

## 4.1 Functional requirement & non functional requirement

### Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Requirements	Static signboards will be replaced with smart linked sign boards that meet all criteria.
FR-2	User Registration	User Registration can be done through a Website or Gmail
FR-3	User Confirmation	Phone Confirmation Email confirmation OTP authentication
FR-4	Payments options	Bank Transfers
FR-5	Product Delivery and installation	The installation fee will be depend upon the length of the road.
FR-6	Product Feedback	Will be shared through a website via Gmail

### Non-functional Requirements:

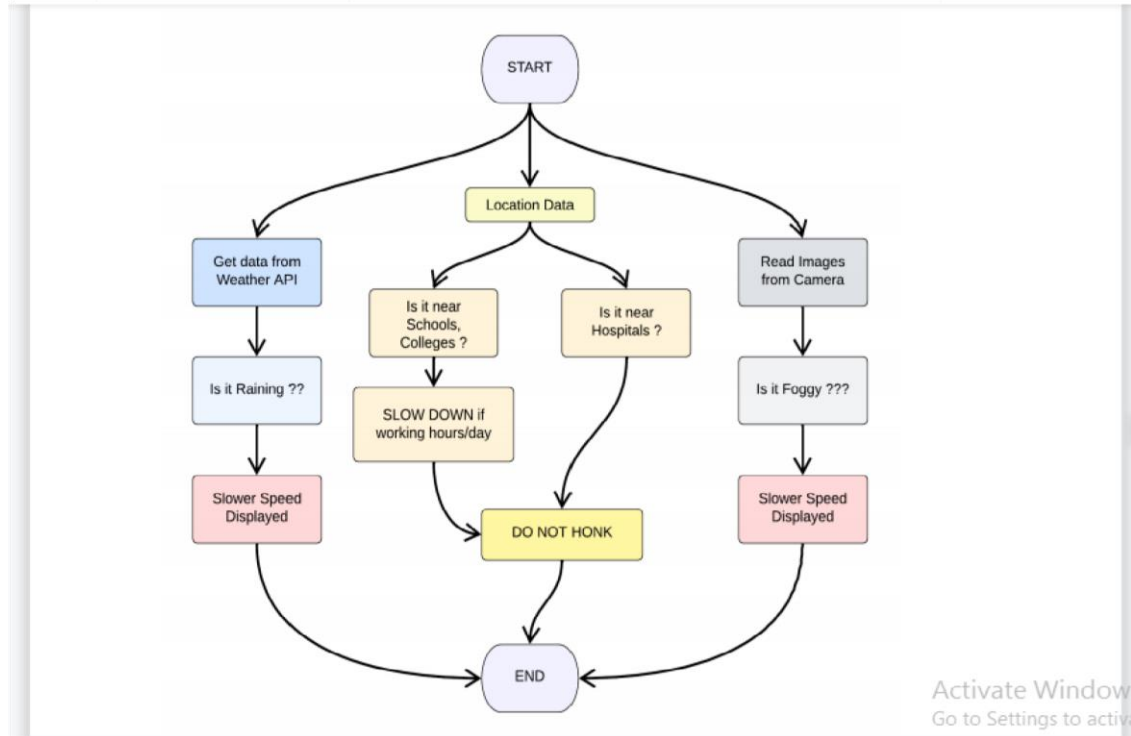
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	Will provide the clear product instructions and a self-explanatory product which is simple to use.
NFR-2	Security	Cloud data must be contained within the network, collapsing to be the real-time avoidance should be avoided, and the board will be monitored constantly.
NFR-3	Reliability	Hardware will be frequently tested.
NFR-4	Performance	The smart board must provide a better user experience and deliver the accuracy output.
NFR-5	Availability	All of the functions and the user demands will be provided, depend upon the customer needs.
NFR-6	Scalability	The product is based on road safety and should cover the entire highway system.

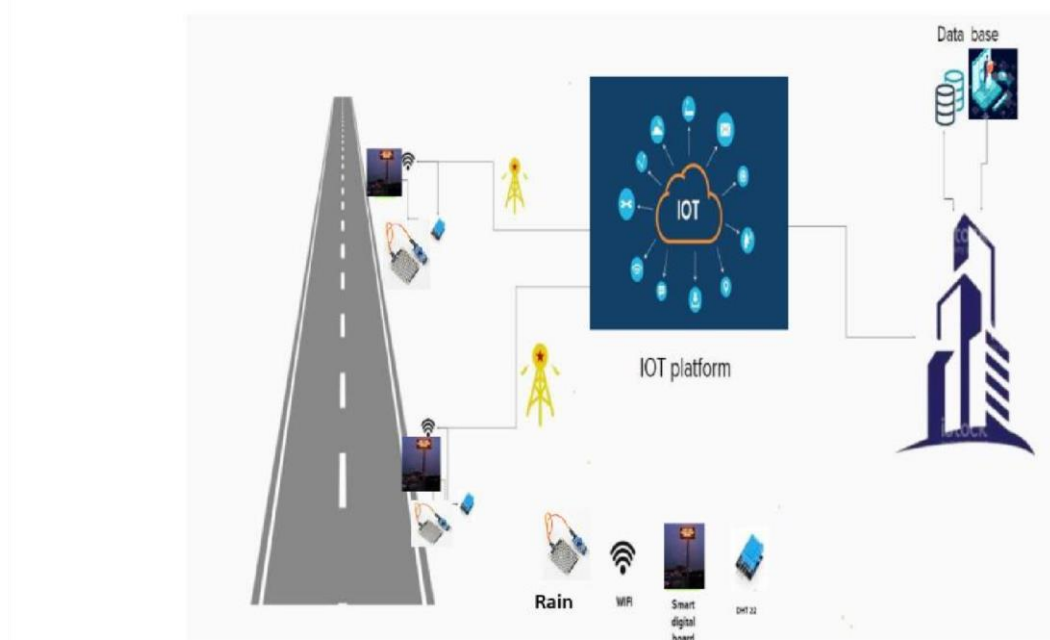
Activ  
Go trails

## 5. PROJECT DESIGN

### 5.1 Data Flow Diagrams



## 5.2 Solution Architecture



## 5.3 User Stories

### Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1		US-1	Create the IBM Cloud services which are being used in this project.	6	High	Muthulekshmi, Vinishalini
Sprint-1		US-2	Configure the IBM Cloud services which are being used in completing this project.	4	Medium	Saral ,Anns
Sprint-2		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	Muthulekshmi, Vinishalini
Sprint-2		US-4	In order to connect the IoT device to the IBM cloud, create a device in the IBM Watson IoT platform and get the device credentials.	5	High	Muthulekshmi, Vinishalini
Sprint-3		US-1	Configure the connection security and create API keys that are used in the Node-RED service for accessing the IBM IoT Platform.	10	High	Saral ,Anns

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team members
Sprint-3		US-2	Create a Node-RED service.	10	High	Saral ,Anns
Sprint-3		US-1	Develop a python script to publish random sensor data such as temperature, moisture, soil and humidity to the IBM IoT platform	7	High	Muthulekshmi, Vinishalini
Sprint-3		US-2	After developing python code, commands are received just print the statements which represent the control of the devices.	5	Medium	Muthulekshmi, Vinishalini
Sprint-4		US-3	Publish Data to The IBM Cloud	8	High	Saral ,Anns
Sprint-4		US-1	Create Web UI in Node- Red	10	High	Saral ,Anns
Sprint-4		US-2	Configure the Node-RED flow to receive data from the IBM IoT platform and also use Cloudant DB nodes to store the received sensor data in the cloudant DB	10	High	Muthulekshmi, Vinishalini

## 6. PROJECT PLANNING & SCHEDULING

### 6.1 Sprint Planning & Estimation

<b>STEP 1</b>	<b>Identify the problem</b>
---------------	-----------------------------



STEP 2	Prepare an abstract, problem statement
STEP 3	List required objects needed
STEP 4	Create a code and run it
STEP 5	Make a prototype
STEP 6	Test with the created code and check the designed prototype is working
STEP 7	Solution for the problem is found

## 6.2 Reports

### SPRINT 1

WOKWI

sketch.ino

```

1 #include <WiFi.h> //library for wifi
2 #include <PubSubClient.h> //library for MQTT
3 #include "DHT.h" // Library for dht11
4 #define DHTPIN 5 // what pin we're connected to
5 #define DHTTYPE DHT22 // define type of sensor DHT 11
6
7 DHT dht (DHTPIN, DHTTYPE); // creating the instance by passing pin and type of
8
9 void callback(char* subscribtopic, byte* payload, unsigned int payloadLength)
10
11 //-----credentials of IBM Accounts-----
12
13 #define ORG "psh4py" //IBM ORGANITION ID
14 #define DEVICE_TYPE "alert-device" //Device type mentioned in ibm watson IOT Platform
15 #define DEVICE_ID "4571" //Device ID mentioned in ibm watson IOT Platform
16 #define TOKEN "12345678" //Token
17 String data3;
18 float h, t;
19
20
21 //----- Customise the above values -----
22 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // Server Name
23 char publishTopic[] = "iot-2/evt/Data/fmt/json"; // topic name and type of event
24 char subscribtopic[] = "iot-2/cmd/command/fmt/json"; // cmd REPRESENT command
25 char authMethod[] = "use-token-auth"; // authentication method
26 char token[] = TOKEN;
27 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; //client id
28
29
30 //-----

```

Simulation

00:09.316 76%

```

{"temp":1.30,"humidity":86.00,"North":0,"South":0,"East":0,"West":0}
Publish ok
temp:1.30
humidity:86.00
Sending payload:
{"temp":1.30,"humidity":86.00,"North":0,"South":0,"East":0,"West":0}
Publish ok

```

Go to Settings to activate Windows.

## SPRINT 2

IBM Watson IoT Platform

12345 Disconnected NodeMCU Device Nov 7, 2022 7:39 PM

Identity Device Information Recent Events State Logs

The recent events listed show the live stream of data that is coming and going from this device.

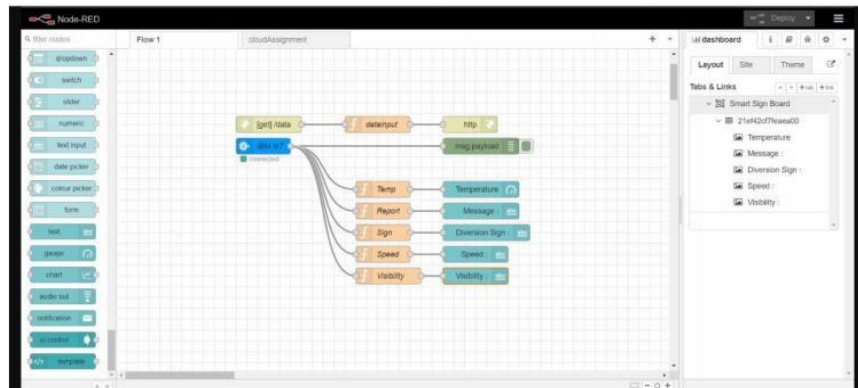
Event	Value	Format	Last Received
eventflow	{"randomNumber":100,"temp":95,"HUM":84}	json	a few seconds ago
eventflow	{"randomNumber":37,"temp":94,"HUM":81}	json	a minute ago
eventflow	{"randomNumber":7,"temp":98,"HUM":65}	json	2 minutes ago
eventflow	{"randomNumber":35,"temp":92,"HUM":91}	json	3 minutes ago
eventflow	{"randomNumber":70,"temp":98,"HUM":73}	json	4 minutes ago

1 Simulation running

Activate Windows  
Go to Settings to activate Windows.

Step – 3:

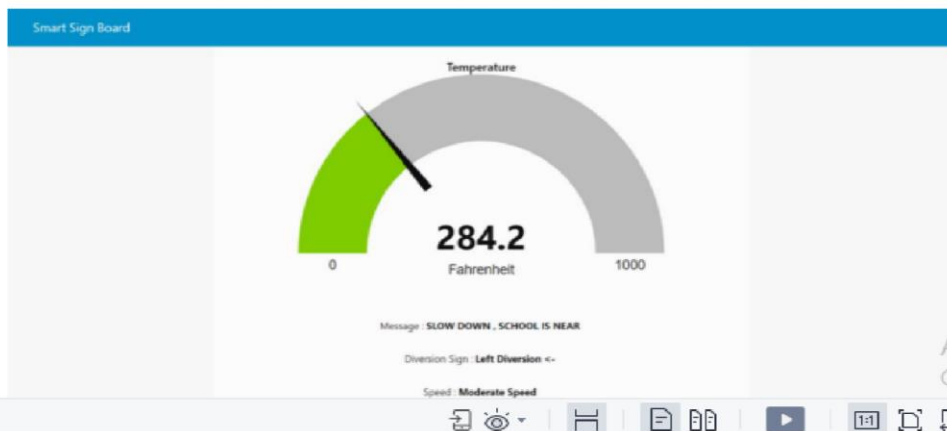
Creating Node-Red:



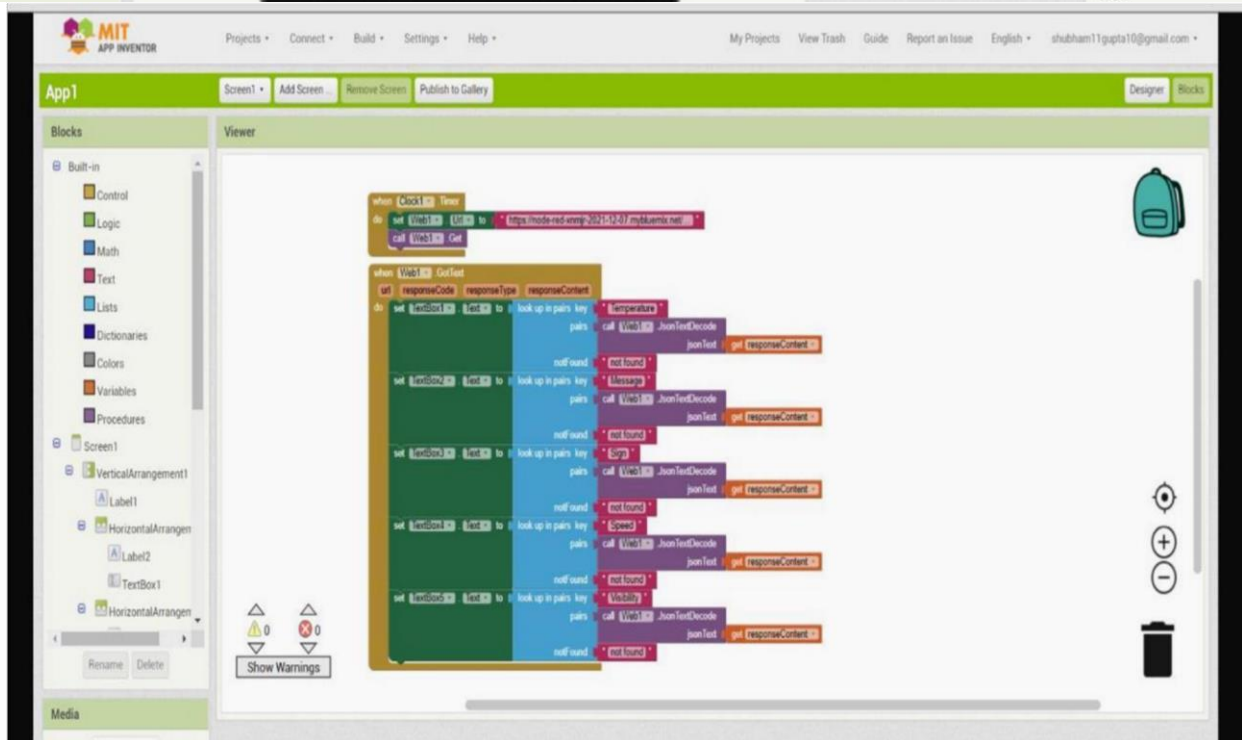
Find and Replace

Activate Windows  
Go to Settings to activate Windows

Altering Link using ui to get the Graphical output :



Activate Windows  
Go to Settings to activate Windows



## SPRINT 4

### Project Development Phase Sprint - 4

Date	13-11-2022
Team ID	PNT 2022T MID5 1479
Project Name	Signs with Smart Connectivity for Better Road Safety

**Image – 1: Web-UI**



**Image – 2: Node-red**

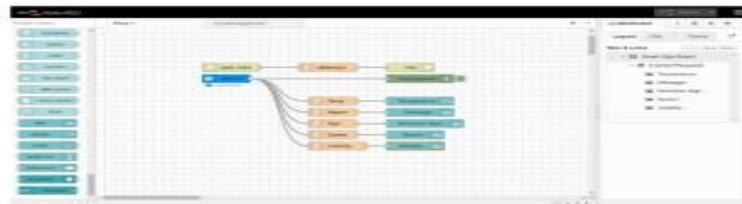
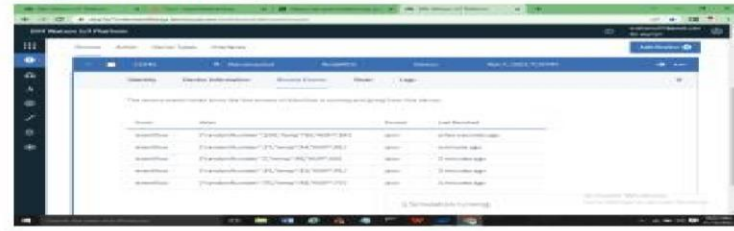


Image – 3: IOT Device output



Name	Model	Version	Last Received
SmartSignBoard	SmartSignBoard-v1.0.0	1.0.0	2020-07-14 10:00:00
SmartSignBoard	SmartSignBoard-v1.0.0	1.0.0	2020-07-14 10:00:00
SmartSignBoard	SmartSignBoard-v1.0.0	1.0.0	2020-07-14 10:00:00
SmartSignBoard	SmartSignBoard-v1.0.0	1.0.0	2020-07-14 10:00:00
SmartSignBoard	SmartSignBoard-v1.0.0	1.0.0	2020-07-14 10:00:00

Image – 4: MIT App inventor Front end



## 7. CODING & SOLUTIONING

### 7.1 Feature 1

- IoT device
- IBM Watson Platform
- Node red
- Web UI
- MIT App Inventor
- Python code

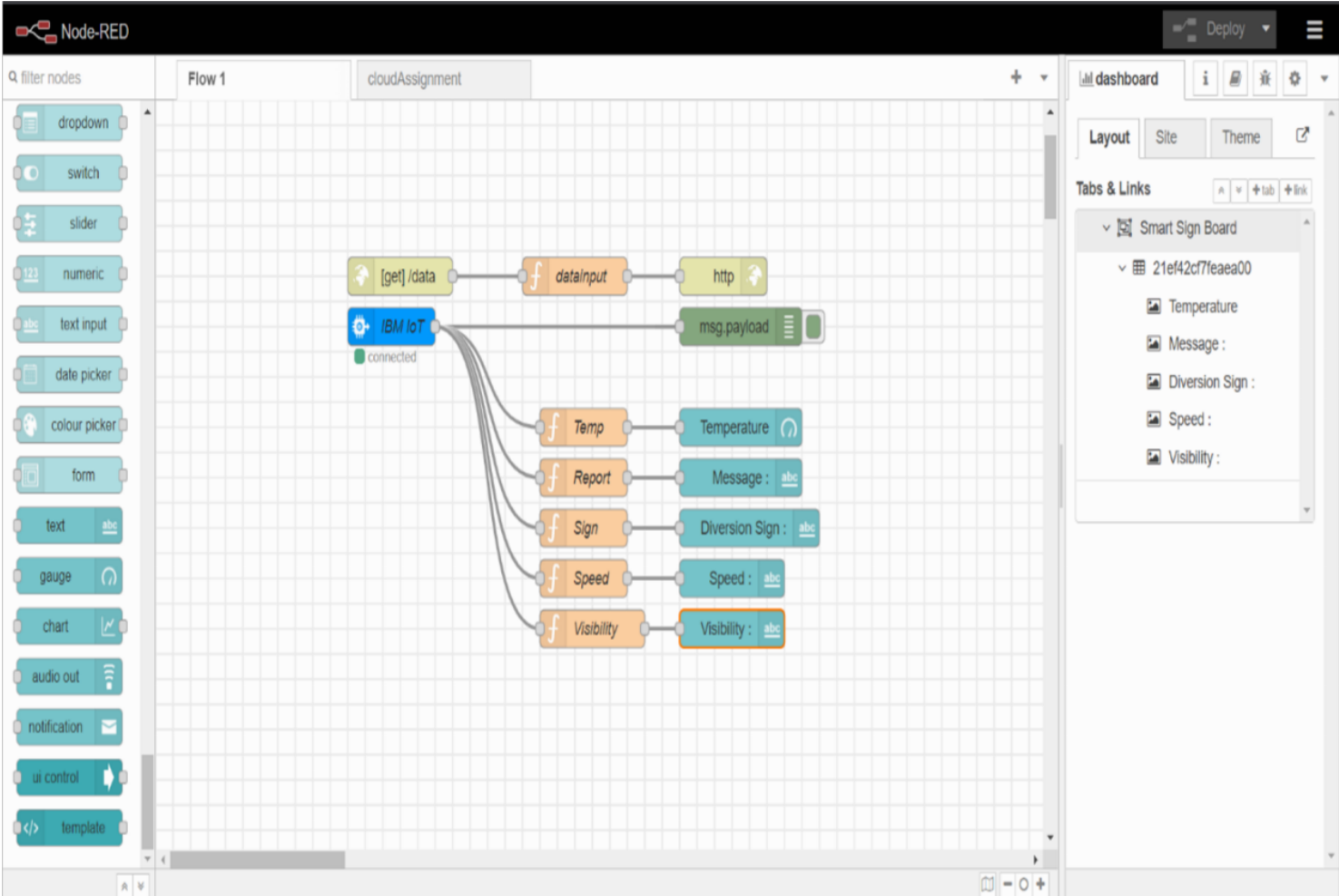
### 7.2 Feature 2

- Login
- Verification
- Showing the features
- Adding rating

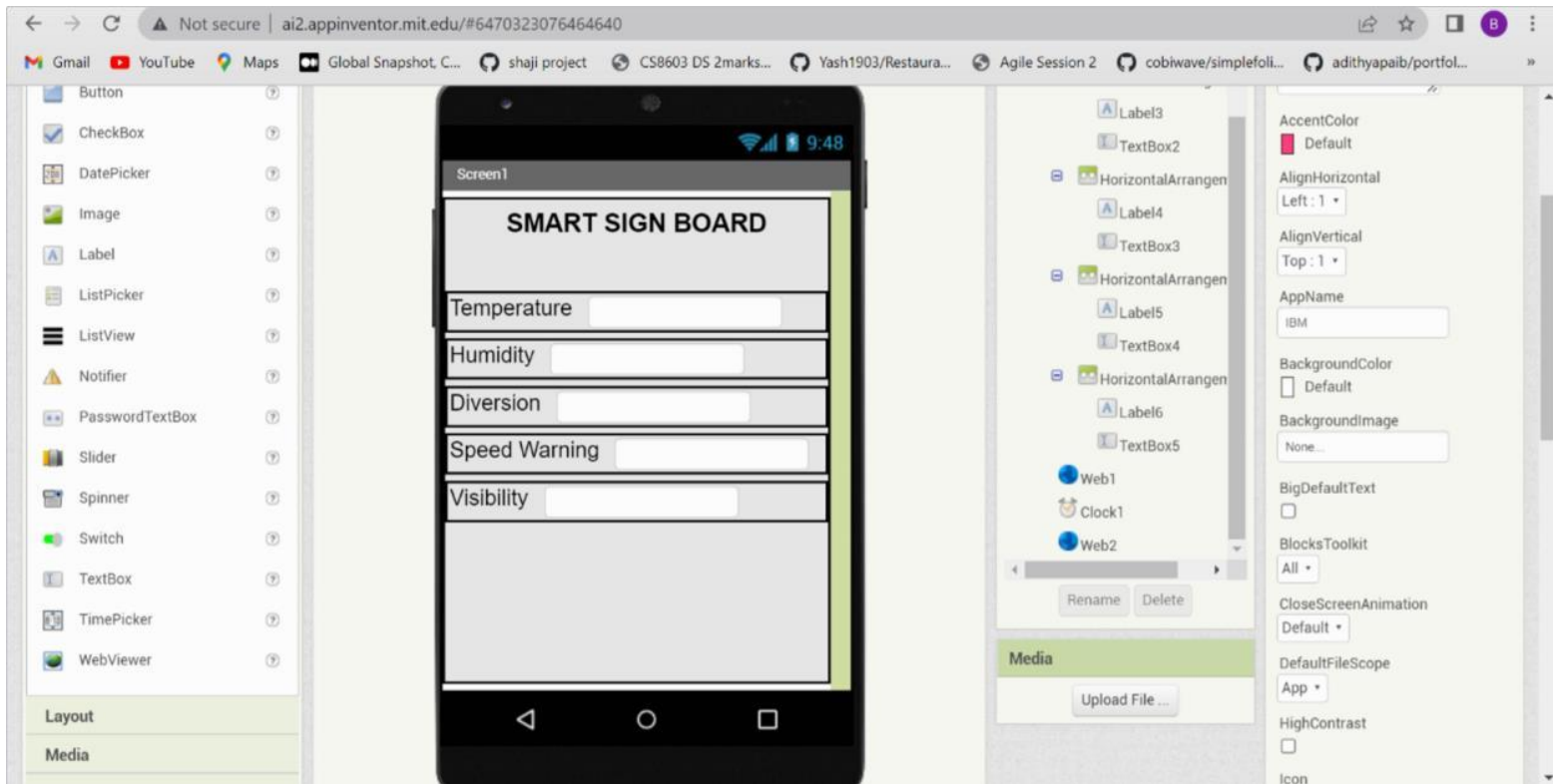
## 8. TESTING AND RESULTS

### 8.1 Test Cases

# Test case 1



## Test case 2





## Test case 3

The screenshot displays the MIT App Inventor web interface. The top navigation bar includes the MIT App Inventor logo, a menu (Projects, Connect, Build, Settings, Help), and user information (My Projects, View Trash, Guide, Report an Issue, English, shubham11gupta10@gmail.com). Below this is a project bar for 'App1' with buttons for 'Screen1', 'Add Screen', 'Remove Screen', and 'Publish to Gallery'. On the right of the project bar are 'Designer' and 'Blocks' tabs.

The left sidebar, titled 'Blocks', contains a 'Built-in' section with categories: Control, Logic, Math, Text, Lists, Dictionaries, Colors, Variables, and Procedures. Under 'Screen1', there are components: VerticalArrangement1, Label1, HorizontalArrangement, Label2, TextBox1, and HorizontalArrangement. At the bottom of the sidebar are 'Rename' and 'Delete' buttons, and a 'Media' section.

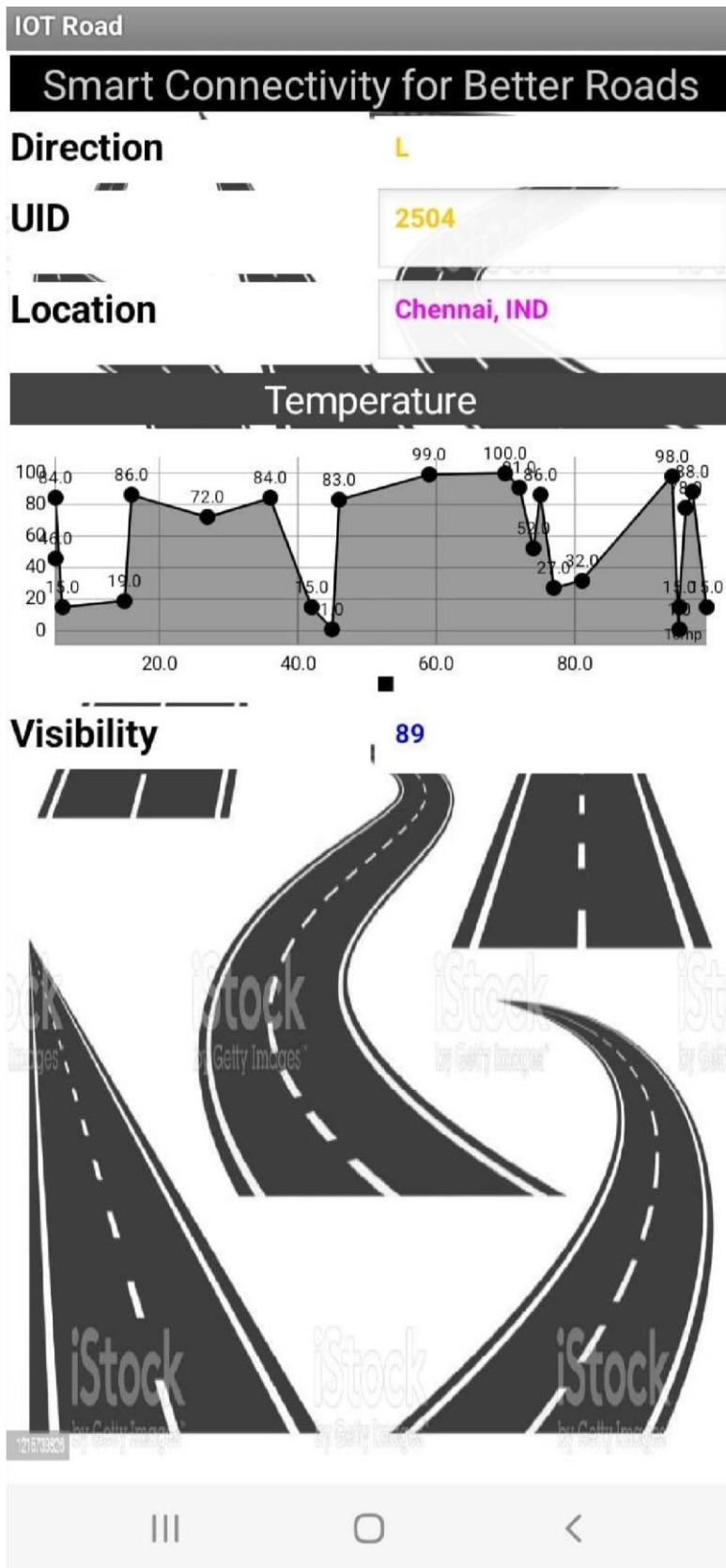
The main workspace is titled 'Viewer' and shows a visual programming logic block. The logic starts with a 'when Clock1 - Timer' block. Inside its 'do' loop, there is a 'set Web1 - Url' block set to 'https://node-red-rmp-2021-12-07-mybluemix.net/' and a 'call Web1 - Get' block. This is followed by a 'when Web1 - GetText' block. Its 'do' loop contains a table with three columns: 'responseCode', 'responseType', and 'responseContent'. The table has five rows, each representing a different weather parameter: 'Temperature', 'Message', 'Sign', 'Speed', and 'Visibility'. For each row, the logic performs a 'look up in pairs' operation, followed by a 'call Web1 - JsonTextDecode' block, and then a 'jsonText - get responseContent' block. If a key is 'not found', the logic also includes a 'not found' block. At the bottom left of the workspace, there are warning indicators (0 yellow triangles and 0 red crosses) and a 'Show Warnings' button. On the right side of the workspace, there are icons for a backpack, a target, zoom in (+), zoom out (-), and a trash can.

## Test case 4

The screenshot displays the MIT App Inventor web interface. On the left is a component palette with categories: User Interface, Layout, Media, Drawing and Animation, Maps, Charts, Sensors, Social, Storage, Connectivity, LEGO MINDSTORMS, Experimental, and Extension. The 'Connectivity' category is expanded, showing 'Web' as the selected component. The central workspace shows a mobile app design for 'IOT Road' with the title 'Smart Connectivity for Better Roads'. The app layout includes input fields for 'Direction', 'UID', and 'Location', a 'Temperature' section with a line chart titled 'Temp', and a 'Visibility' section. The line chart has a y-axis from 0 to 3 and an x-axis with labels 'C', 2, 3, and 4. The chart shows a line starting at (C, 0) and rising to (4, 3). On the right, the 'Properties' pane shows settings for the selected 'Web' component, including 'Enabled', 'FontBold', 'FontSize' (14.0), 'FontTypeface' (default), 'Height' (Automatic), 'Width' (Fill parent), 'Hint', 'MultiLine', 'NumbersOnly', 'ReadOnly', 'Text', and 'TextAlignment' (left: 0). Below the workspace, the 'Non-visible components' section shows 'Web1', 'Clock1', 'Clock2', and 'Web2'.

Non-visible components

- Web1
- Clock1
- Clock2
- Web2



## 8.ADVANTAGES

1. Traffic signals help for movement of traffic securely without any collision.
2. They can reduce the number of accidents on roads like pedestrian accident and right-angle collision of two cars
3. Signals can increase the capacity of traffic handling at the intersection.
4. The traffic signals help for the safe movement of slow-moving traffic by interrupting heavy traffic at regular intervals.

## 9. DISADVANTAGES

- Network issues may arise.

## 10. CONCLUSION

Almost all the countries across the globe strive to meet the demand for safe, fast, and reliable road services. Lack of operational efficiency and reliability, safety, and security issues, besides aging road systems and practices are haunting various countries to bring about a change in their existing road system infrastructure. Often, they suffer from the lack in smart technologies and latest technological updates to provide the most efficient services. This is expected to induce to build traffic systems that are smarter and more efficient. Most significant improvements have been evidenced by more informative and user-friendly websites, mobile applications for real-time information about vehicles in motion, and weather condition, construction works, schools hospitals nearby. With the rise of Industry, companies can now ensure that they are prepared to avoid the surprise of equipment downtime. this also help to avoid accidents like death, Like above mentioned, the developed application of our project can lead the passenger who travel can travel safely without any fear.

## 11. FUTURE SCOPE

This application is ensured for safety for the peoples while they are travelling alone as well as they travel with their family or friends.

In future, this application may also be used improve the safety the peoples who travel through it . By further enhancement of the application the peoples can explore more features regarding their safety.

## 12. APPENDIX

```
#include <WiFi.h> //library for wifi
#include <PubSubClient.h> //library for MQTT
#include "DHT.h" // Library for dht11
#define DHTPIN 5 // what pin we're connected to
#define DHTTYPE DHT22 // define type of sensor DHT 11

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 "psh4py"//IBM ORGANITION ID
#define DEVICE_TYPE "alert-device"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "4571"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "12345678" //Token
String data3;
float h, t;

//----- Customise the above values -----
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Name
char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of event perform
and format in which data to be send
char subscribetopic[] = "iot-2/cmd/command/fmt/json";// 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(33, INPUT); //North
  pinMode(25, INPUT); // South
  pinMode(26, INPUT); // East
  pinMode(27, INPUT); // West
  delay(10);
  Serial.println();
  wificonnect();
  mqttconnect();
}

int n, s, e, w;

void loop()// Recursive Function

```

```

{

    h = dht.readHumidity();
    t = dht.readTemperature();
    Serial.print("temp:");
    Serial.println(t);
    Serial.print("humidity:");
    Serial.println(h);

    n = digitalRead(33);
    s = digitalRead(25);
    e = digitalRead(26);
    w = digitalRead(27);

    PublishData(t, h, n, s, e, w);
    delay(1000);
    if (!client.loop()) {
        mqttconnect();
    }
}

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

void PublishData(float temp, float humid, int n, int s, int e, int w) {
    mqttconnect();//function call for connecting to ibm
    /*
        creating the String in in form JSON to update the data to ibm cloud
    */
    String payload = "{\"temp\":";
    payload += temp;
    payload += "," " \"humidity\":";
    payload += humid;
    payload += "," " \"North\":";
    payload += n;
    payload += "," " \"South\":";
    payload += s;
    payload += "," " \"East\":";
    payload += e;
    payload += "," " \"West\":";
    payload += w;
    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="";
}

```

OUTPUT:

Node RED Dashboard:

LINK TO NODE RED DASHBOARD

## 13.2 GitHub GitHub

**link:**

<https://github.com/IBM-EPBL/IBM-Project-42700-1660707213>