PROJECT REPORT

TEAM ID	**PNT2022TMID54322**
PROJECT NAME	**Signs with Smart Connectivity for Better Road Safety**

1. INTRODUCTION

1.1 Project Overview

The goal of this project is to replace the static signboards with smart connected sign boards to get the speed limitations from a web app using weather API and update it automatically based on the weather conditions, set diversions through API and warn drivers for school zones and hospital zones.

1.2 Purpose

- To replace static sign boards, smart connected sign boards are used.
- These smart connected sign boards get the speed limitations from a web app using weather API and update automatically.
- Based on the weather changes the speed may increase or decrease.
- Traffic diversion signs are remotely controlled using APIs.
- “DO NOT HONK” messages displayed at Schools and Hospital zones where we can set buttons.

2. LITERATURE SURVEY

2.1 Existing problem

- Rain makes brakes inefficient and leads to accidents.
- Fog reduces visibility and increases the probability of accidents.
- Traffic diversion requires human intervention

2.2 References

1. R. Chen, L. Hei and Y. Lai, "Image Recognition and Safety Risk Assessment of Traffic Sign Based on Deep Convolutional Neural Network," in *IEEE Access*, vol. 8, pp. 201799-201805, 2020, doi: 10.1109/ACCESS.2020.3032581

2. N. P. Botekar and M. N. Mahalakshmi, "Development of road sign recognition for ADAS using OpenCV," *2017 International Conference on Intelligent Computing and Control (I2C2)*, 2017, pp. 1-4, doi: 10.1109/I2C2.2017.8321941.

3. S. S. M. Sallah, F. A. Hussin and M. Z. Yusoff, "Road sign detection and recognition system for real-time embedded applications," International Conference on Electrical, Control and Computer Engineering 2011 (InECCE), 2011, pp. 213-218, doi: 10.1109/INECCE.2011.5953878.

4. M. Derawi, Y. Dalveren and F. A. Cheikh, "Internet-of-Things-Based Smart Transportation Systems for Safer Roads," 2020 IEEE 6th World Forum on Internet of Things (WF-IoT), 2020, pp. 1-4, doi: 10.1109/WF-IoT48130.2020.9221208.

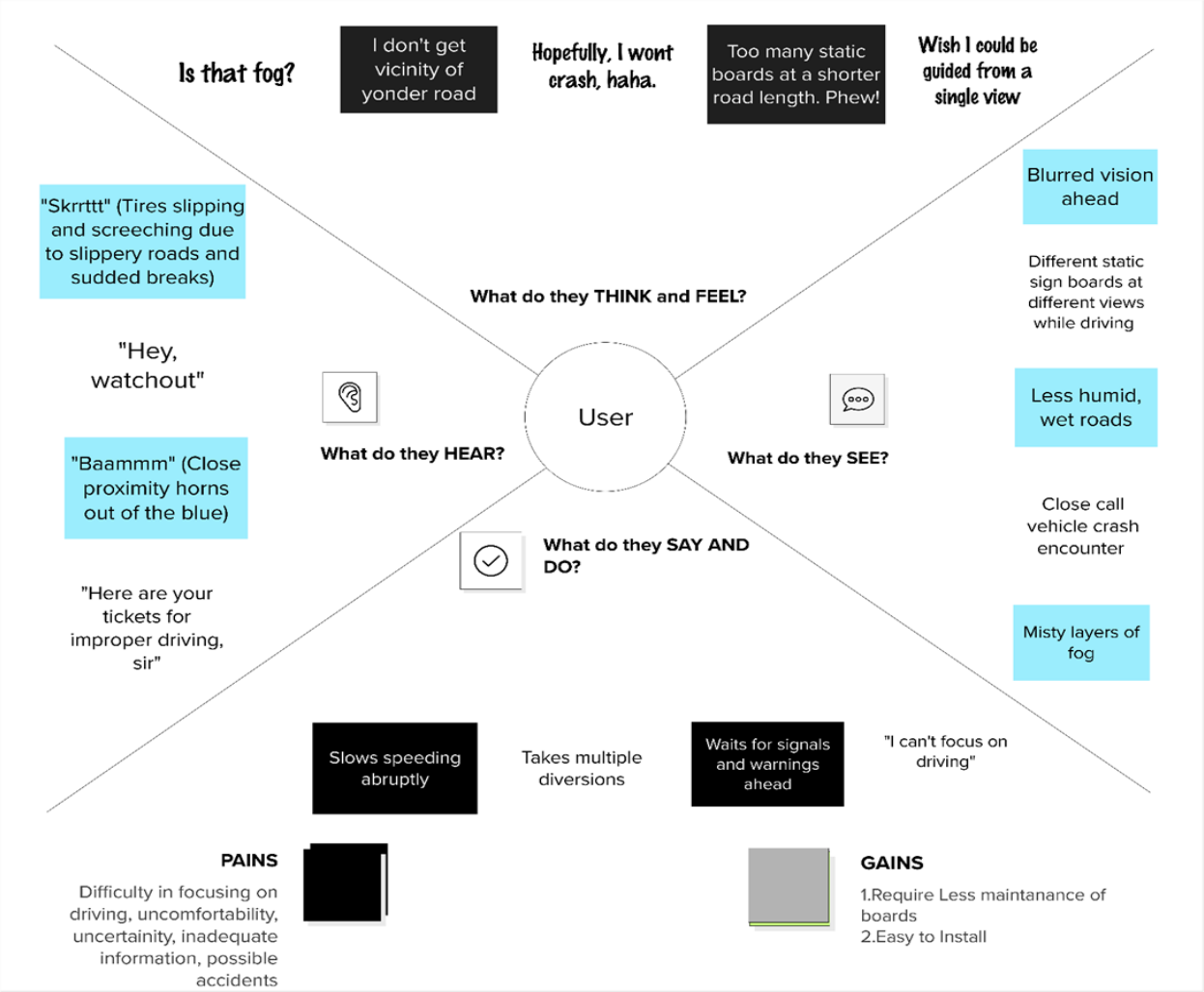
5. S. Kirushanth and B. Kabaso, "Telematics and Road Safety," *2018 2nd International Conference on Telematics and Future Generation Networks (TAFGEN)*, 2018, pp. 103-108, doi: 10.1109/TAFGEN.2018.8580482

2.3 Problem Statement

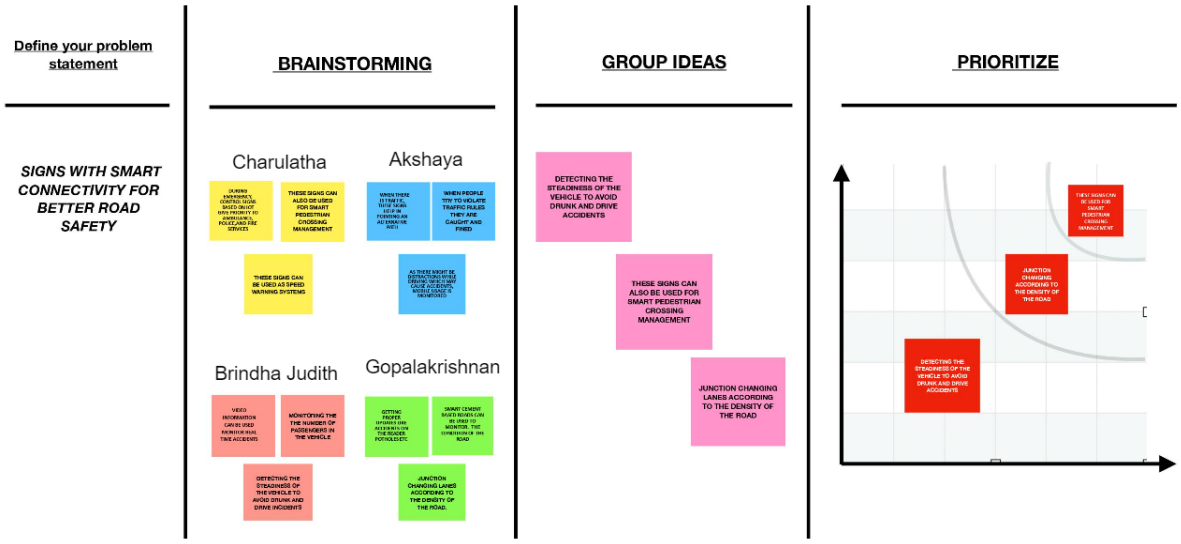
To replace the static signboards with smart connected sign boards to get the speed limitations from a web app using weather API and update it automatically based on the weather conditions, set diversions through API and warn drivers for school zones and hospital zones.

3. IDEATION AND PROPOSED SOLUTIONS

3.1 Empathy Map



3.2 Ideation and Brainstorming



3.3 Proposed Solution

- Use an ESP32 to drive a display as a replacement for static sign boards.
- Configure IBM cloud server such that upon making a single http request with location, unique id, usual speed limit & hospital/school zone info, it processes the data at cloud and returns only the message to be displayed at the sign board display.
- Another http endpoint is configured to set the direction to be displayed. Upon accessing this http endpoint, the direction is set remotely for a display using its unique id.

3.4 Problem Solution Fit

- The display replaces the static signs.
- Processing requirement of the microcontroller is reduced since all the processing is done in the cloud servers.
- Direction can be remotely set by the concerned authorities without needing to personally attend the site.

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

User Visibility	Bright coloured LEDs that can catch drivers' attention but aren't too distracting to cause accidents should be used in sign boards.
User Understanding	So that the user can accurately interpret the signs, information should be displayed using tools like graphics and illustrations with text.
User Convenience	The display needs to be large enough to accurately display each sign so that it can be seen by drivers in the distance.

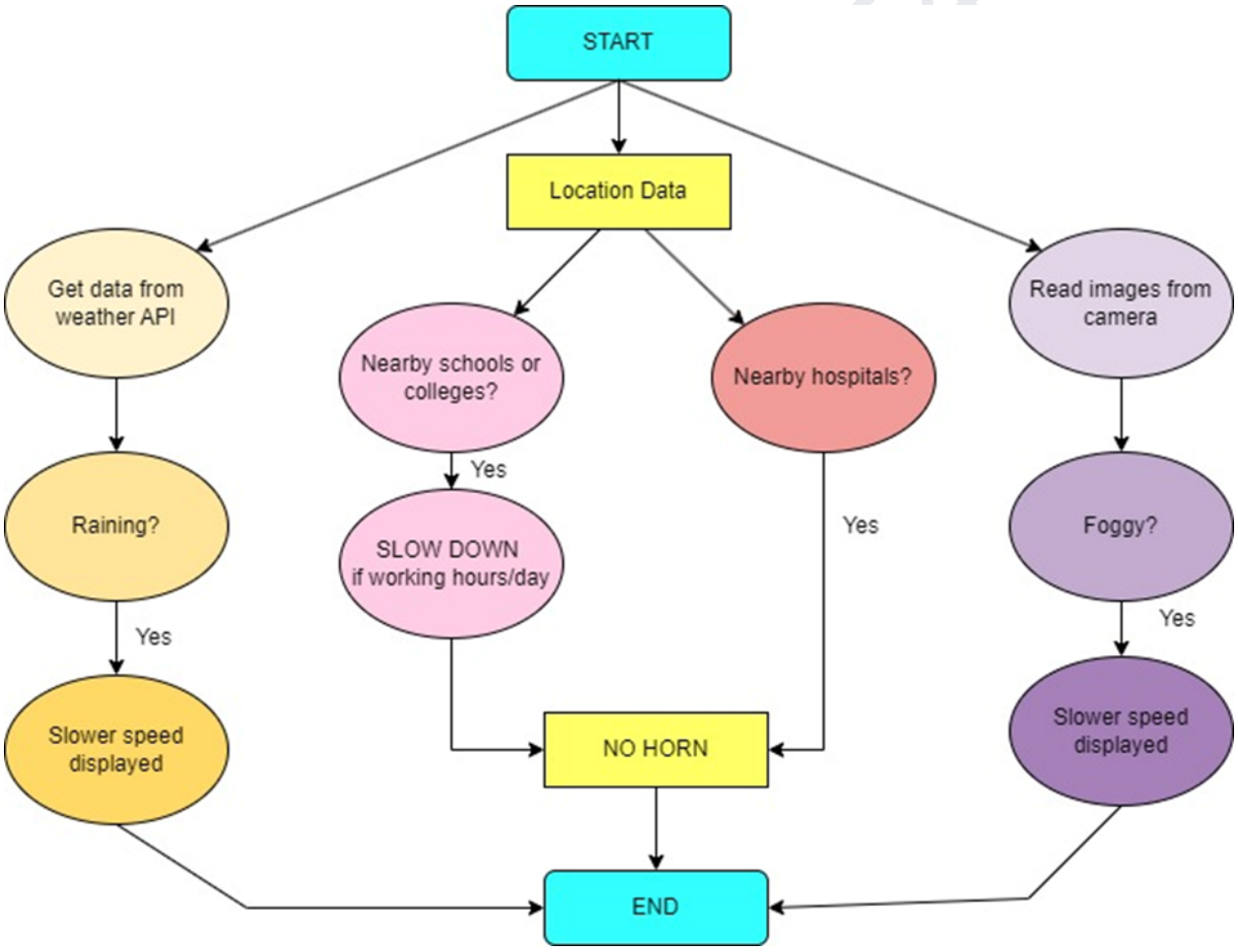
4.2 Non-Functional requirements

Usability	Dynamically updating in a time-based manner should be possible.
Security	To ensure that only the desired messages are shown in the display, it should be secure enough.
Reliability	Should convey the traffic information correctly.
Performance	Every time the traffic or weather values are updated, the display should dynamically refresh.

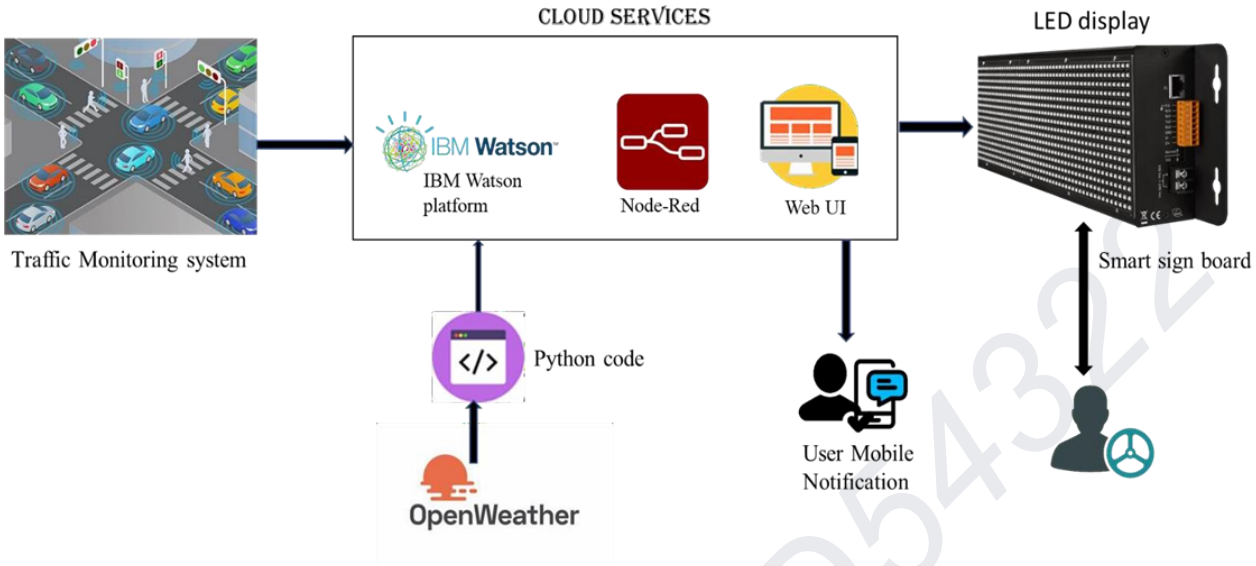
Availability	Should be on service 24/7.
Scalability	In order to extend horizontally between servers, it should be modular and unique.

5. PROJECT DESIGN

5.1 Data Flow Diagrams



5.2 Solution and Technical Architecture



5.3 User Stories

6. PROJECT PLANNING AND SCHEDULING

Title	Description	Date
Literature Survey on The Selected Project and Information Gathering	A Literature Survey is a compilation summary of research done previously in the given topic. Literature survey can be taken from books, research paper online or from any source	3 September 2022
Prepare Empathy Map	Empathy Map is a visualization tool which can be used to get a better insight of the customer.	9 September 2022
Ideation-Brainstorming	Brainstorming is a group problem solving session where ideas are shared, discussed and organized among the team members.	17 September 2022
Define Problem Statement	A Problem Statement is a concise description of the problem or issues a project seeks to address. The problem statement identifies the current state, the desired future state and any gaps between the two.	20 September 2022
Problem Solution Fit	This helps us to understand the thoughts of the customer, their likes, behavior, emotions etc.	26 September 2022
Proposed Solution	Proposed solution shows the current solution and it helps go towards the desired result until it is achieved.	26 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 and features to complete our project.	1 October 2022

Customer Journey	It helps us to analyze from the perspective of a customer, who uses our project.	7 October 2022
Functional Requirement	Here functional and non-functional requirements are briefed. It has specific features like usability, security, reliability, performance, availability and scalability.	10 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.	13 October 2022
Technology Architecture	Technology Architecture is a more well-defined version of solution architecture. It helps us analyze and understand various technologies that need 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.	26 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 November 2022

6.1 Sprint Planning & Estimation

Sprint	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	USN-1	Create the IBM Cloud services which are being used in this project.	6	High	Charulatha S Akshaya M Brindha P J GopalakrishnanT
Sprint-1	USN-2	Configure the IBM Cloud services which are being used in completing this project.	4	Medium	Charulatha S Akshaya M Brindha P J GopalakrishnanT
Sprint-1	USN-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	Charulatha S Akshaya M Brindha P J GopalakrishnanT
Sprint-1	USN-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	Charulatha S Akshaya M Brindha P J GopalakrishnanT

Sprint-2	USN-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	Charulatha S Akshaya M Brindha P J GopalakrishnanT
Sprint-2	USN-2	Create a Node-RED service.	10	High	Charulatha S Akshaya M Brindha P J GopalakrishnanT
Sprint-3	USN-1	Develop a python script to publish random sensor data such as temperature, humidity, rain to the IBM IoT platform	7	High	Charulatha S Akshaya M Brindha P J GopalakrishnanT
Sprint-3	USN-2	After developing python code, commands are received just print the statements which represent the control of the devices.	5	Medium	Charulatha S Akshaya M Brindha P J GopalakrishnanT
Sprint-3	USN-3	Publish Data to The IBM Cloud	8	High	Charulatha S Akshaya M Brindha P J GopalakrishnanT

Sprint-4	USN-1	Create Web UI in Node-Red	10	High	Charulatha S Akshaya M Brindha P J GopalakrishnanT
Sprint-4	USN-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	Charulatha S Akshaya M Brindha P J GopalakrishnanT

6.2 Sprint Delivery Schedule

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

7. CODING & SOLUTIONING

7.1 Feature 1

PYTHON SCRIPT

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

#Provide your IBM Watson Device Credentials
organization = "efr0if"
deviceType = "rasberrypi"
deviceId = "123"
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")
else:
        print("led is off")


#print(cmd)

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)
          data = {'temperature':temp, 'humidity':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()
```

7.2 Feature 2

```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt
#include "DHT.h"// Library for dht11
#define DHTPIN 15     // what pin we're connected to
#define DHTTYPE DHT22   // define type of sensor DHT 11
#define LED 2

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 "efr0if"//IBM ORGANITION ID
#define DEVICE_TYPE "rasberrypi"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "123"//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/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();
  t = dht.readTemperature();
  Serial.print("temp:");
  Serial.println(t);
  Serial.print("Humid:");
  Serial.println(h);

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



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

void PublishData(float temp, float humid) {
  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 += "," "\"Humid\":";
  payload += humid;
  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="";
}
```

8. TESTING

8.1 Test Cases

Section	Total Cases	Not Tested	Fail	Pass
Overall Process	4	0	0	4
Client Side	30	0	0	30
Security and User Information Protection	3	0	0	3
Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Report Output	5	0	0	5
Version Control and	2	0	0	2

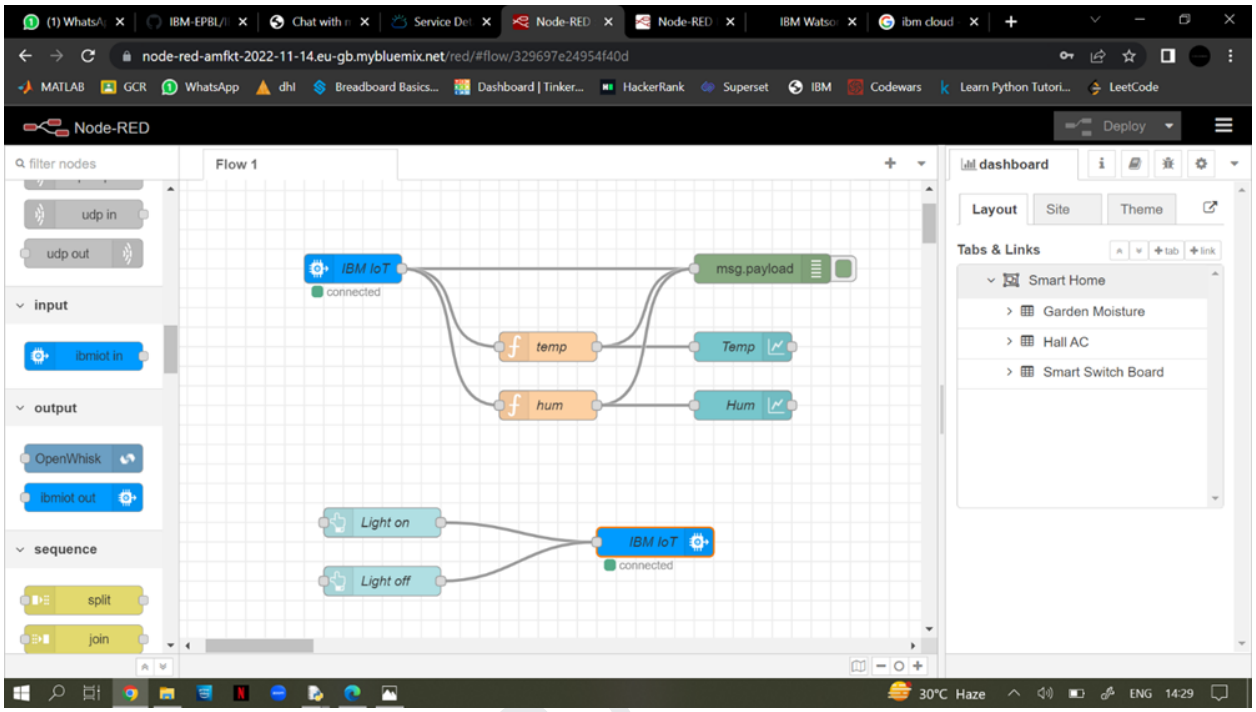
Updation				
Adaptation	2	0	0	2

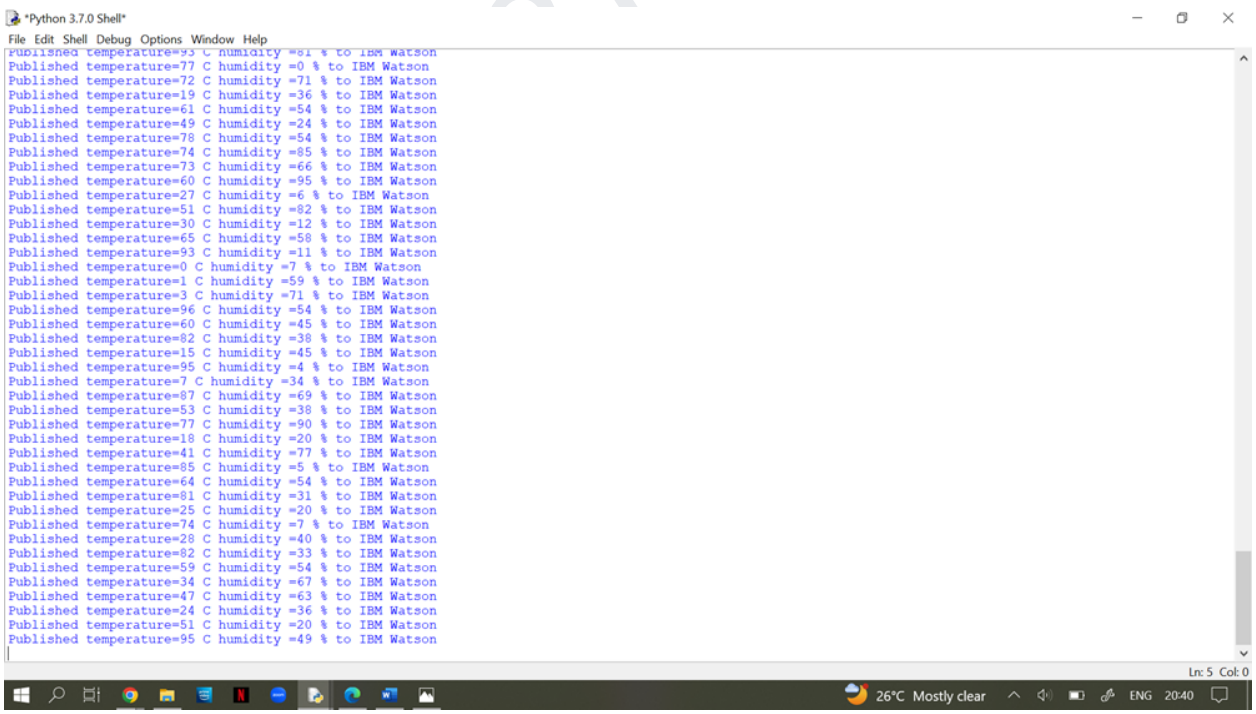
8.2 User Acceptance Testing

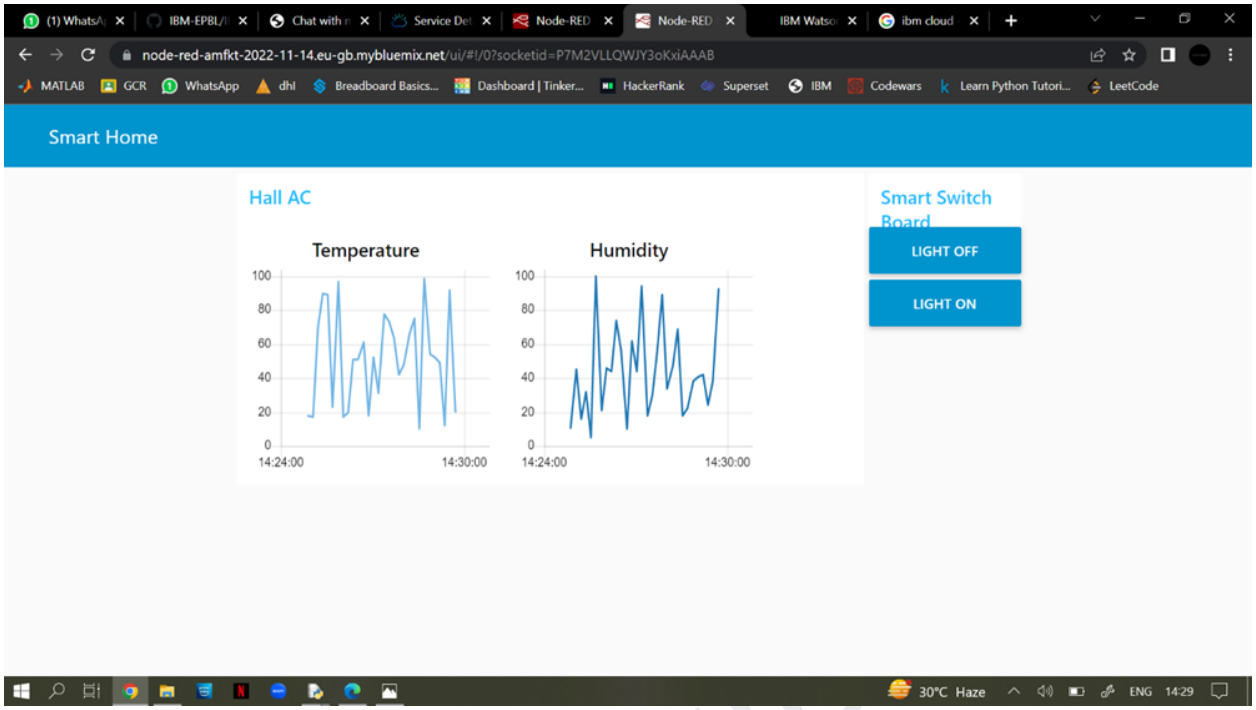
Section	Satisfied	Dissatisfied	No Comments	Score
Use-easiness	7	1	2	10
Reliability	8	1	1	10
Security	8	2	0	10
Availability	3	2	5	10
Performance	9	0	1	10
Scalability	5	2	3	10
Adaption	4	2	4	10

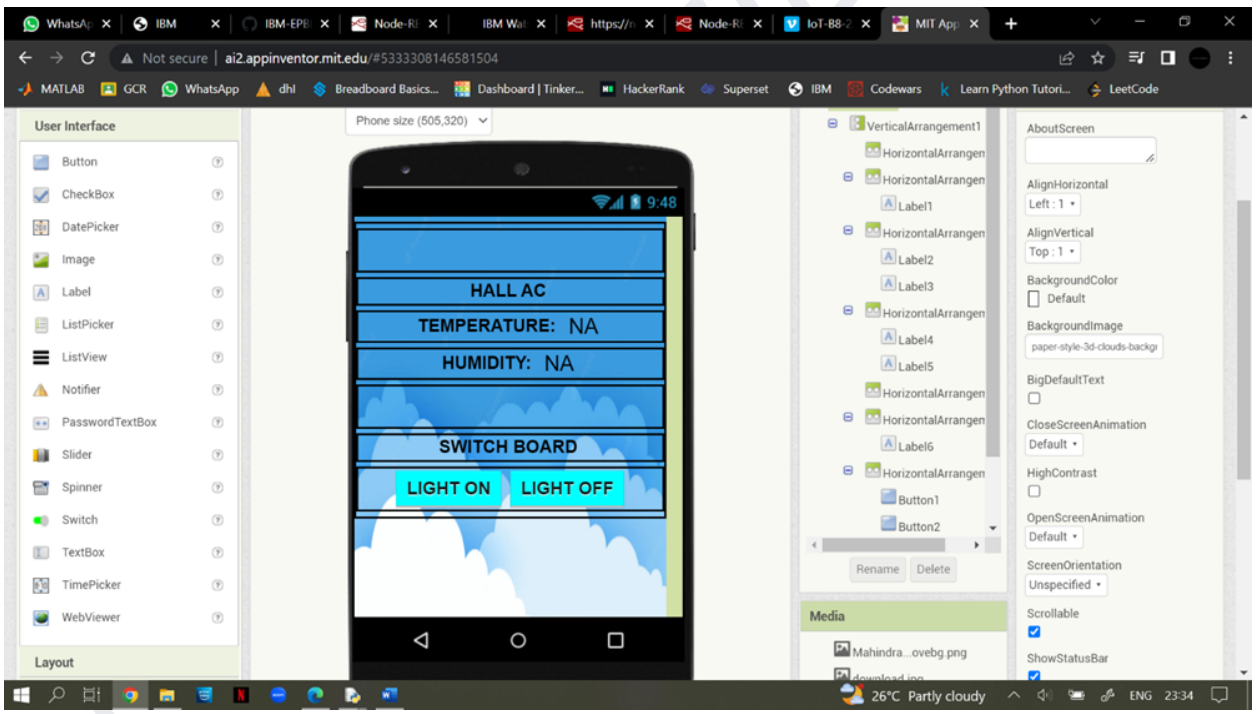
9. RESULTS

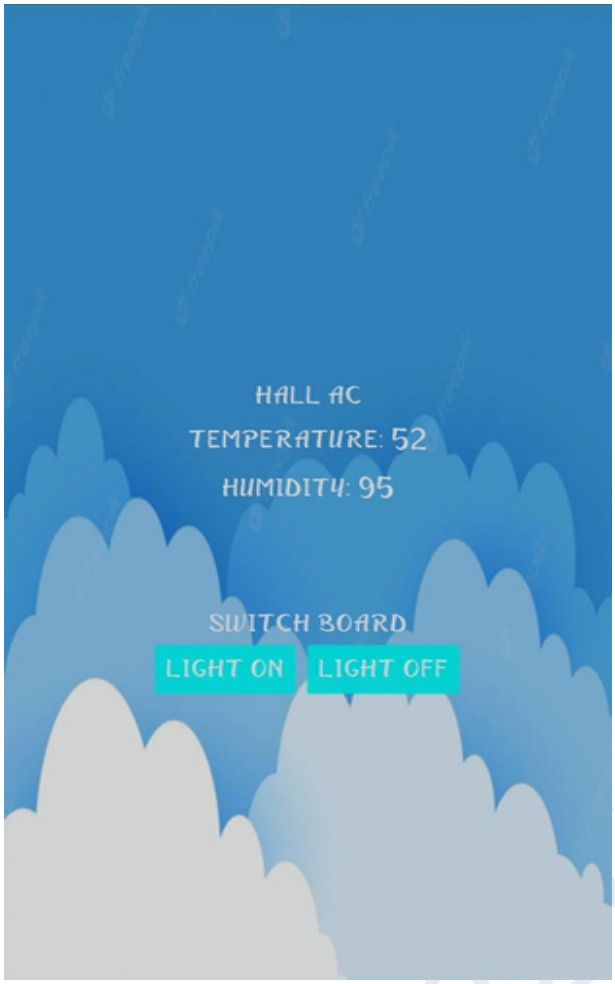
9.1 Performance Metrics

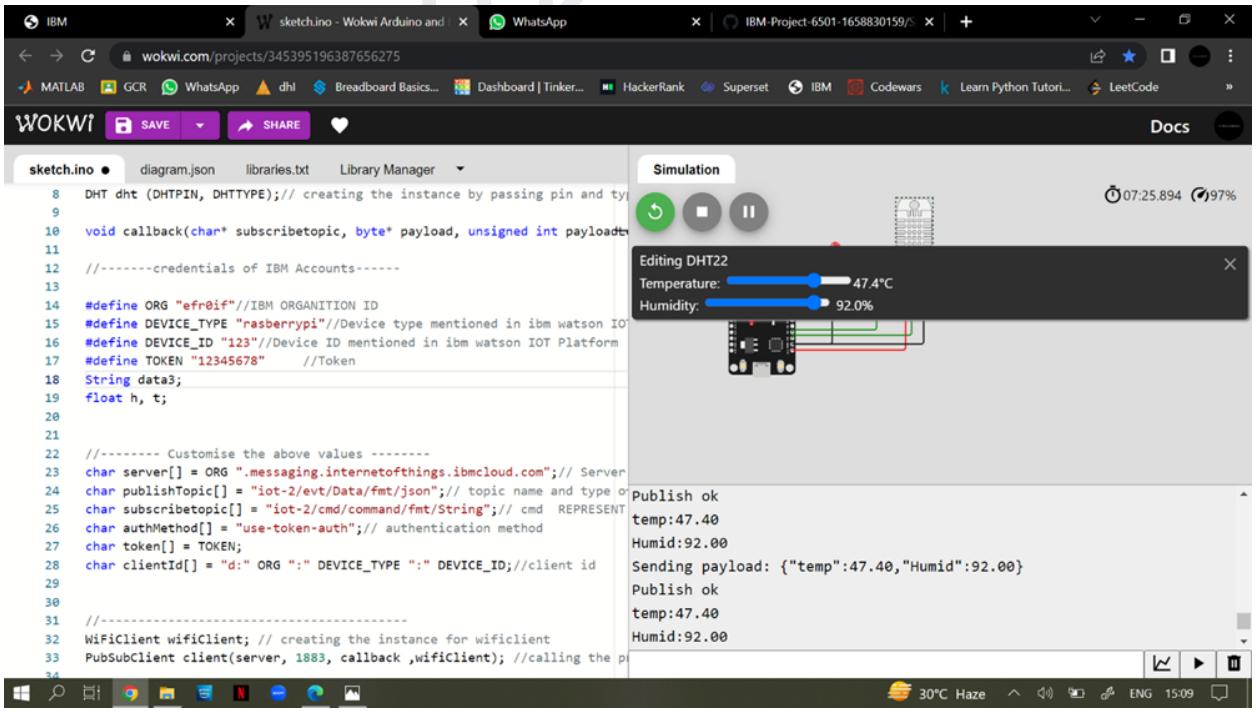





node-red-amfkt-2022-11-14.eu-gb.mybluemix.net/ui/#!/0?socketid=P7M2VLLQWJY3oKxiAAAB
Smart Home
Hall AC
Temperature
Humidity
100
80
60
40
20
0
14:24:00
14:30:00
Smart Switch Board
LIGHT OFF
LIGHT ON


ai2.appinventor.mit.edu/#5333308146581504
User Interface
Button
CheckBox
DatePicker
Image
Label
ListPicker
ListView
Notifier
PasswordTextBox
Slider
Spinner
Switch
TextBox
TimePicker
WebViewer
Layout
Phone size (505,320)
9:48
HALL AC
TEMPERATURE: NA
HUMIDITY: NA
SWITCH BOARD
LIGHT ON
LIGHT OFF
Media
AboutScreen
AlignHorizontal
AlignVertical
BackgroundColor
BackgroundImage
BigDefaultText
CloseScreenAnimation
HighContrast
OpenScreenAnimation
ScreenOrientation
Scrollable
ShowStatusBar





10. ADVANTAGES

- Lower battery consumption since processing is done mostly by Node RED servers in the cloud.
- Cheaper and low requirement micro controllers can be used since processing requirements are reduced.
- Longer lasting systems.
- Dynamic Sign updation.
- School/Hospital Zone alerts.

DISADVANTAGES

- The size of the display determines the requirement of the micro controller.
- Dependent on OpenWeatherAPI and hence the speed reduction is same for a large area in the scale of cities.

11. CONCLUSION

Our project is capable of serving as a replacement for static signs for a comparatively lower cost and can be implemented in the very near future. This will help reduce a lot of accidents and maintain a more peaceful traffic atmosphere in the country.

12. FUTURE SCOPE

Introduction of intelligent road sign groups in real life scenarios could have great impact on increasing the driving safety by providing the end-user (car driver) with the most accurate information regarding the current road and traffic conditions. Even displaying the information of a suggested driving speed and road surface condition (temperature, icy, wet or dry surface) could result in smoother traffic flows and, what is more important, in increasing a driver's awareness of the road situation.

13. APPENDIX

GITHUB link

- https://github.com/IBM-EPBL/IBM-Project-41467-1660642338

Demonstration video link

- https://drive.google.com/file/d/1V7IQEkURkJNJQAGf703_CHVEHQSeurke/view?usp=sharing