

**A  
PROJECT REPORT  
ON  
SMART CONNECTED SIGN BOARDS  
FOR IMPROVED ROAD SAFETY  
SUBMITTED IN FULFILMENT FOR THE COMPLETION OF  
SUMMER INTERNSHIP PROGRAM  
AT  
SMARTBRIDGE  
BY  
MIRZA AKBER NAMAZI  
NEHA DINESH PRABHU  
AMBICA TANDLE  
UNDER THE GUIDANCE OF  
MRS.ANJUSHA,  
PROJECT GUIDE**



**SMART BRIDGE  
JNTU-H, HYDERABAD – 500072**



## CERTIFICATE

This is to certify that the project work entitled “Smart Connected Sign Boards For Improved Road Safety” submitted by **Mirza Akber Namazi, Neha Dinesh Prabhu** and **Ambica Tandle** in fulfilment of the requirements for the completion of Summer Internship Program-2019 in **INTERNET OF THINGS[IOT]** at **SMART BRIDGE**, Hyderabad, is a record of original work done carried out by them under my supervision and guidance, during the period of Internship at **SMART BRIDGE**. The results embodied in this report have not been submitted to any other University for the award of any other Degree or Diploma.

**Project Guide**

**Mrs. Anjusha**

**Smart Bridge, Hyderabad**

## **ABSTRACT**

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. By using the Weather Api we can get the weather reports based on which we can set the speed limit to particular area. 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.

## Table of Contents

PROJECT REPORT .....	1
CERTIFICATE .....	2
1. OBJECTIVE OF THE PROJECT: .....	5
2. EXISTING METHOD:.....	5
3. DESCRIPTION ABOUT THE PROJECT:.....	5
4. HARDWARE REQUIREMENTS: .....	5
5. SOFTWARE REQUIREMENTS:.....	6
6. METHODOLOGY: .....	7
7. USE CASE DIAGRAM: .....	7
8. PROGRAM CODE: .....	8
9. OUTPUT SCREEN:.....	10
10. CONCLUSION AND FUTURE SCOPE:.....	13
11. LEARNING RESOURCES: .....	13

## List of Figures

Figure 1- Device Credentials and Global variables .....	8
Figure 2-Connecting to OpenWeather.org for weather updates .....	8
Figure 3- Establishing WiFi and mqtt Connections.....	9
Figure 4- Publishing data onto platform .....	9
Figure 5-project prototype .....	10
Figure 6 - Output scree .....	10
Figure 7-when Traffic is not detected .....	10
Figure 8-When traffic is detected .....	11
Figure 9- Web App .....	11
Figure 10-Information sign boards .....	11
Figure 11-ISB.....	12
Figure 12-change of speed limit based on weather API .....	12

# **SMART CONNECTED SIGN BOARDS**

## **FOR IMPROVED ROAD SAFETY**

### **1. OBJECTIVE OF THE PROJECT:**

Objective of this project is to digitize the sign boards on road to improve road safety by changing the contents displayed on the board dynamically from an authorized person using Web App based various constraints such as speed limits, accident prone areas and warnings such as road constructions etc.

### **2. EXISTING METHOD:**

In present System the road signs and the speed limits are Static. Signboards like jam ahead take diversion, rocky path ahead, sharp turns sign board etc, are mainly used to indicate traffic jam or construction going on ahead etc. Also the speed Limit cannot be changed based on weather which is not constant over an area.

Hence 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.

### **3. DESCRIPTION ABOUT THE PROJECT:**

This project proposes a system which has digital sign boards on which the signs can be changed dynamically. By using the Weather API we can get the weather reports based on which we can set the speed limit to particular area. 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.

### **4. HARDWARE REQUIREMENTS:**

- **NodeMCU:**

NodeMCU is a Wi-Fi microcontroller board which is an open source IoT platform. You can program the device using the Arduino IDE. This device has 9 digital pins out of 17 GPIO pins.

- **Oled screen:**

light-emitting diode containing thin flexible sheets of an organic electroluminescent material, used for visual displays. The OLED display module is one of the most attractive display available for a microcontroller. It has a good view angle and pixel density which makes it reliable for displaying small level graphics.

- **Ultrasonic sensors:**

An Ultrasonic sensor is a device that can measure the distance to an object by using sound waves. It measures distance by sending out a sound wave at a specific frequency and listening for that sound wave to bounce back. By recording the elapsed time between the sound wave being generated and the sound wave bouncing back, it is possible to calculate the distance between the sonar sensor and the object.

- **Connecting wires:**

Connecting wires are used to allow an electrical current to travel from one point on a circuit to another.

- **Bread board:**

A breadboard is a solderless device for temporary prototype with electronics and test circuit designs. Most electronic components in electronic circuits can be interconnected by inserting their leads or terminals into the holes and then making connections through wires where appropriate.

## **5. SOFTWARE REQUIREMENTS:**

- **Arduino IDE:**

The Arduino integrated development environment is a cross-platform application that is written in the programming language Java. It is used to write and upload programs to Arduino compatible boards, but also, with the help of 3rd party cores, other vendor development boards.

- **OpenWeather API:**

To develop anything based on weather, it is necessary to use a *weather provider* as the source of weather information. These weather providers expose a set of services or, in other words, an API, that supports RESTful web services in the form of XML or json format.

- **IBM Watson IOT platform:**

IBM Watson IOT Platform is a foundational cloud offering that can connect and control IoT sensors, appliances, homes, and industries.

- **IBM Cloud:**

IBM Cloud is a suite of cloud computing services from IBM that offers both platform as a service (PaaS) and infrastructure as a service (IaaS). With IBM Cloud IaaS, organizations can deploy and access virtualized IT resources -- such as compute power, storage and networking -- over the internet. For compute, organizations can choose between bare-metal or virtual servers.

## 6. METHODOLOGY:

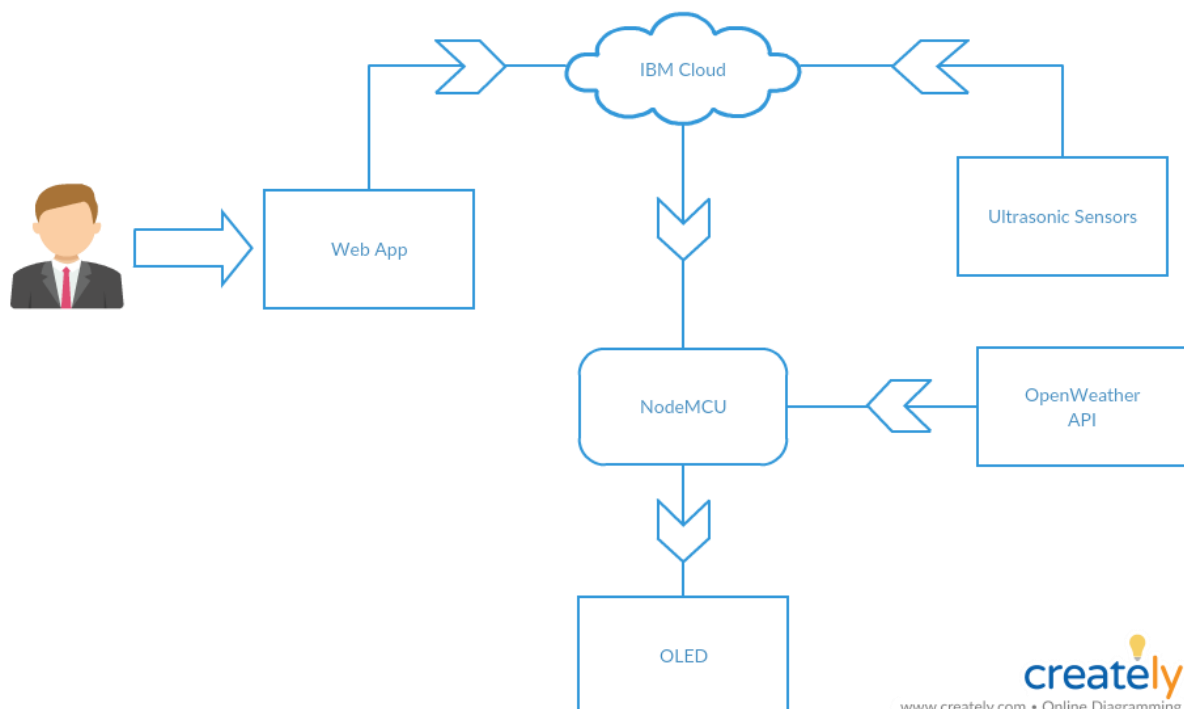
In this project we have two sensors working together, we need to get traffic update through ultra-sonic sensor and weather update through OpenWeather API.

If humidity is high i.e when it is raining oled displays “go slow” and changes the speed limit. If traffic is detected ahead Oled displays “take diversion “, and Admin of the Web App can enter the information of what caused the traffic jam.

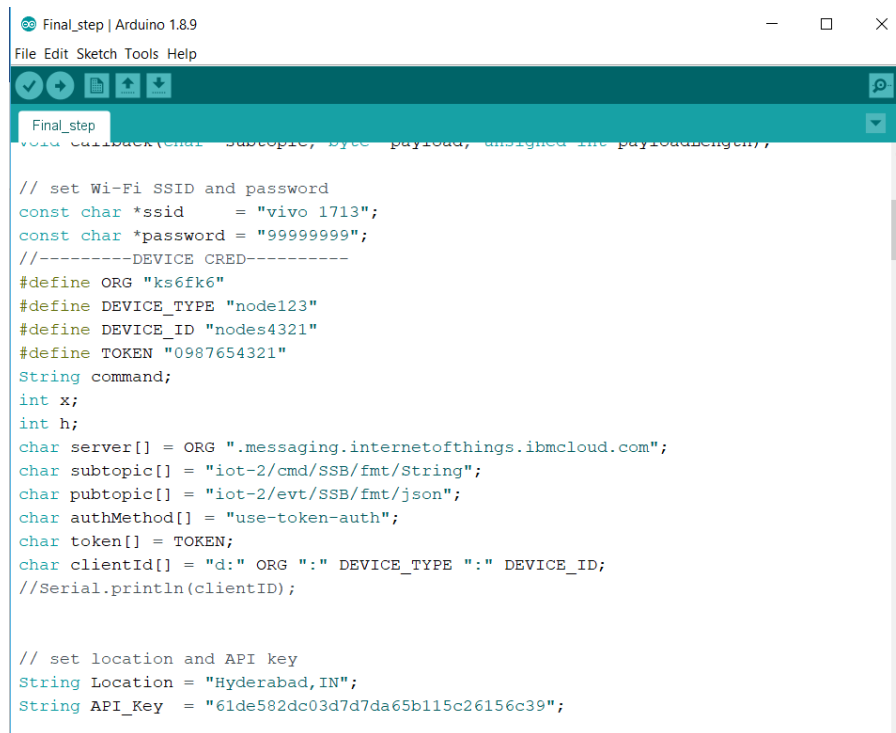
There are 3 modules in the oled screen:

- a) Display of speed limit.
- b) Display of Diversions, Alerts of Accident-prone area.
- c) Information sign boards.

## 7. USE CASE DIAGRAM:



## 8. PROGRAM CODE:



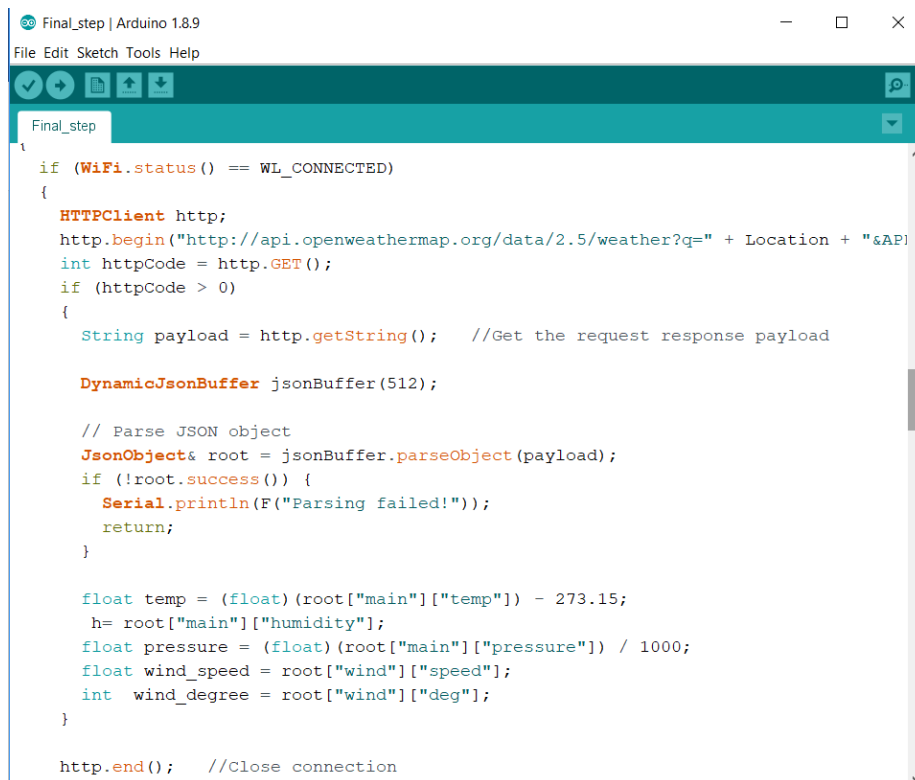
```
Final_step | Arduino 1.8.9
File Edit Sketch Tools Help

Final_step
void callback(char *topic, byte *payload, unsigned int payloadlength);

// set Wi-Fi SSID and password
const char *ssid = "vivo 1713";
const char *password = "99999999";
//-----DEVICE CRED-----
#define ORG "ks6fk6"
#define DEVICE_TYPE "node123"
#define DEVICE_ID "nodes4321"
#define TOKEN "0987654321"
String command;
int x;
int h;
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char subtopic[] = "iot-2/cmd/SSB/fmt/String";
char pubtopic[] = "iot-2/evt/SSB/fmt/json";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
//Serial.println(clientID);

// set location and API key
String Location = "Hyderabad,IN";
String API_Key = "61de582dc03d7d7da65b115c26156c39";
```

Figure 1- Device Credentials and Global variables



```
Final_step | Arduino 1.8.9
File Edit Sketch Tools Help

Final_step
{
  if (WiFi.status() == WL_CONNECTED)
  {
    HTTPClient http;
    http.begin("http://api.openweathermap.org/data/2.5/weather?q=" + Location + "&API");
    int httpCode = http.GET();
    if (httpCode > 0)
    {
      String payload = http.getString(); //Get the request response payload

      DynamicJsonBuffer jsonBuffer(512);

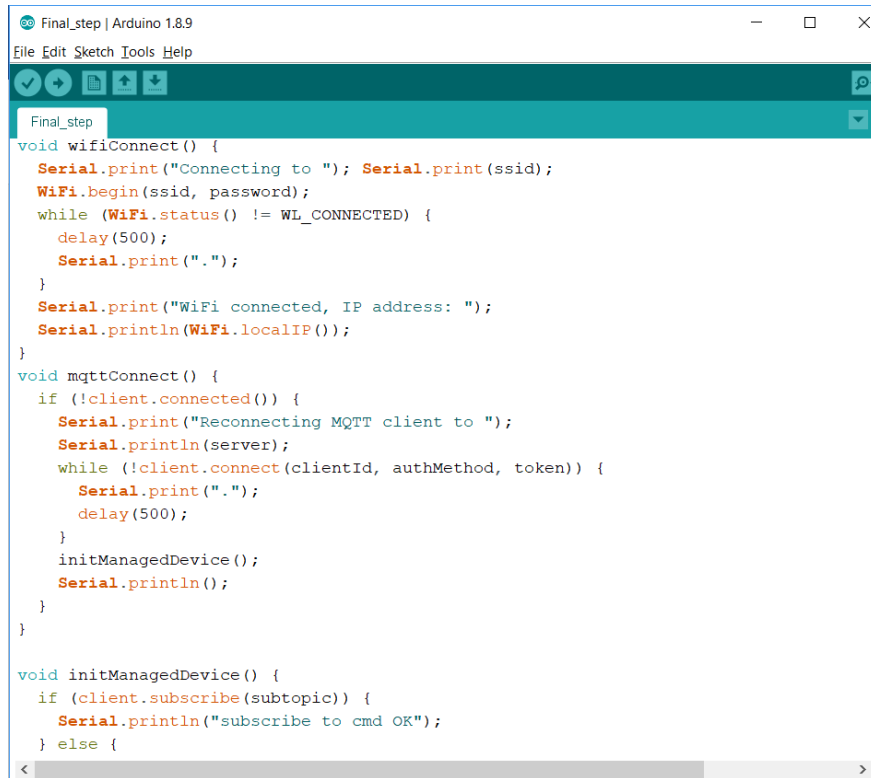
      // Parse JSON object
      JsonObject& root = jsonBuffer.parseObject(payload);
      if (!root.success()) {
        Serial.println(F("Parsing failed!"));
        return;
      }

      float temp = (float)(root["main"]["temp"]) - 273.15;
      h= root["main"]["humidity"];
      float pressure = (float)(root["main"]["pressure"]) / 1000;
      float wind_speed = root["wind"]["speed"];
      int wind_degree = root["wind"]["deg"];
    }

    http.end(); //Close connection
  }
}
```

Figure 2-Connecting to OpenWeather.org for weather updates





```
Final_step | Arduino 1.8.9
File Edit Sketch Tools Help

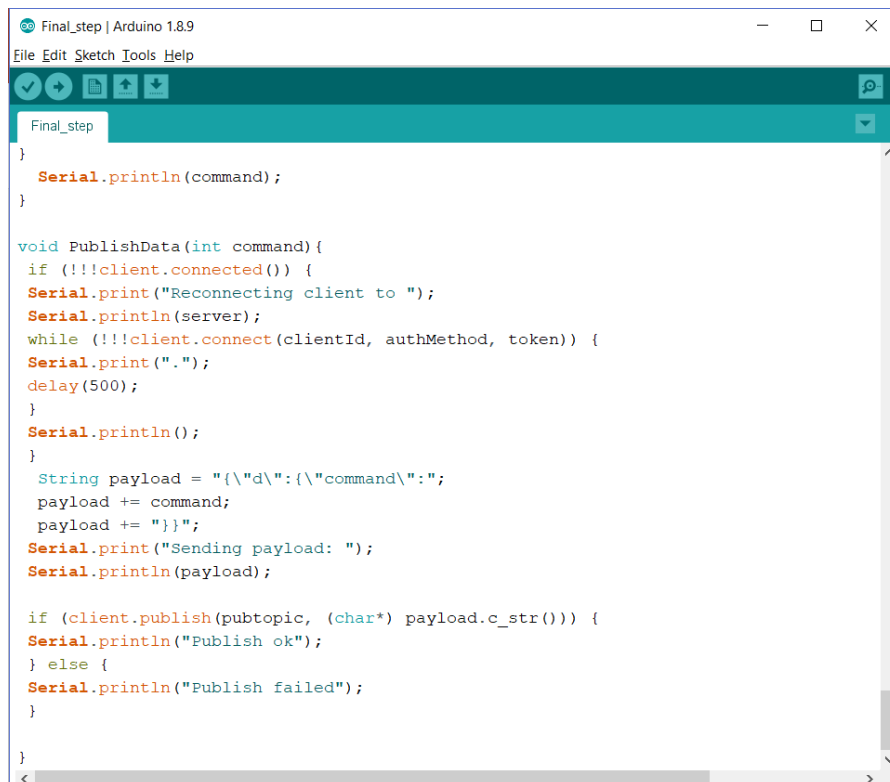
Final_step
void wifiConnect() {
  Serial.print("Connecting to "); Serial.print(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.print("WiFi connected, IP address: ");
  Serial.println(WiFi.localIP());
}

void mqttConnect() {
  if (!client.connected()) {
    Serial.print("Reconnecting MQTT client to ");
    Serial.println(server);
    while (!client.connect(clientId, authMethod, token)) {
      Serial.print(".");
      delay(500);
    }
    initManagedDevice();
    Serial.println();
  }
}

void initManagedDevice() {
  if (client.subscribe(subtopic)) {
    Serial.println("subscribe to cmd OK");
  } else {

```

Figure 3- Establishing WiFi and mqtt Connections



```
Final_step | Arduino 1.8.9
File Edit Sketch Tools Help

Final_step
}
  Serial.println(command);
}

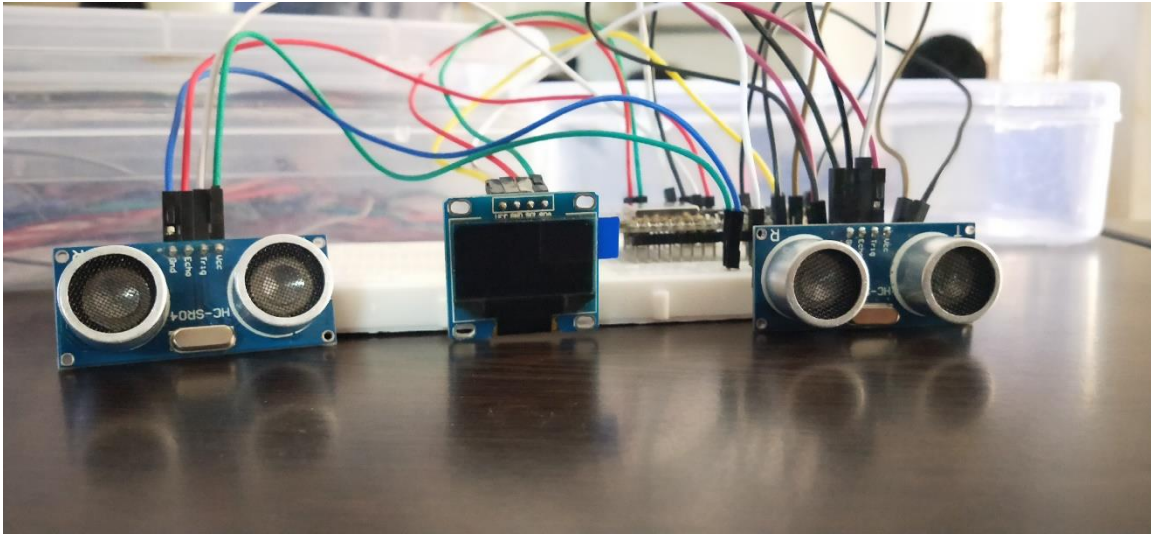
void PublishData(int command){
  if (!client.connected()) {
    Serial.print("Reconnecting client to ");
    Serial.println(server);
    while (!client.connect(clientId, authMethod, token)) {
      Serial.print(".");
      delay(500);
    }
    Serial.println();
  }
  String payload = "{\"d\": \"{command\": ";
  payload += command;
  payload += "}\"}";
  Serial.print("Sending payload: ");
  Serial.println(payload);

  if (client.publish(pubtopic, (char*) payload.c_str())) {
    Serial.println("Publish ok");
  } else {
    Serial.println("Publish failed");
  }
}

```

Figure 4- Publishing data onto platform

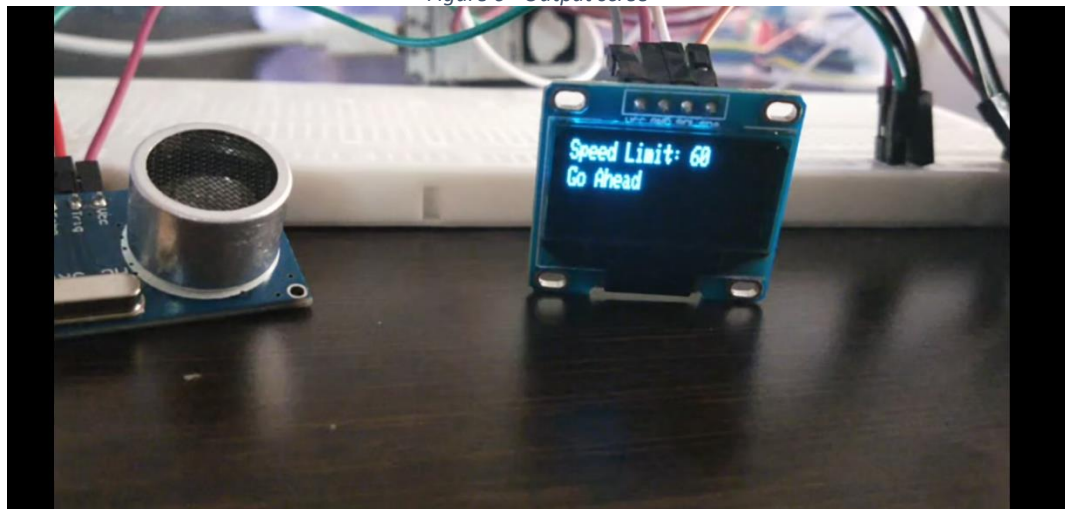
## 9. OUTPUT SCREEN:



*Figure 5-project prototype*



*Figure 6 - Output scree*



*Figure 7-when Traffic is not detected*

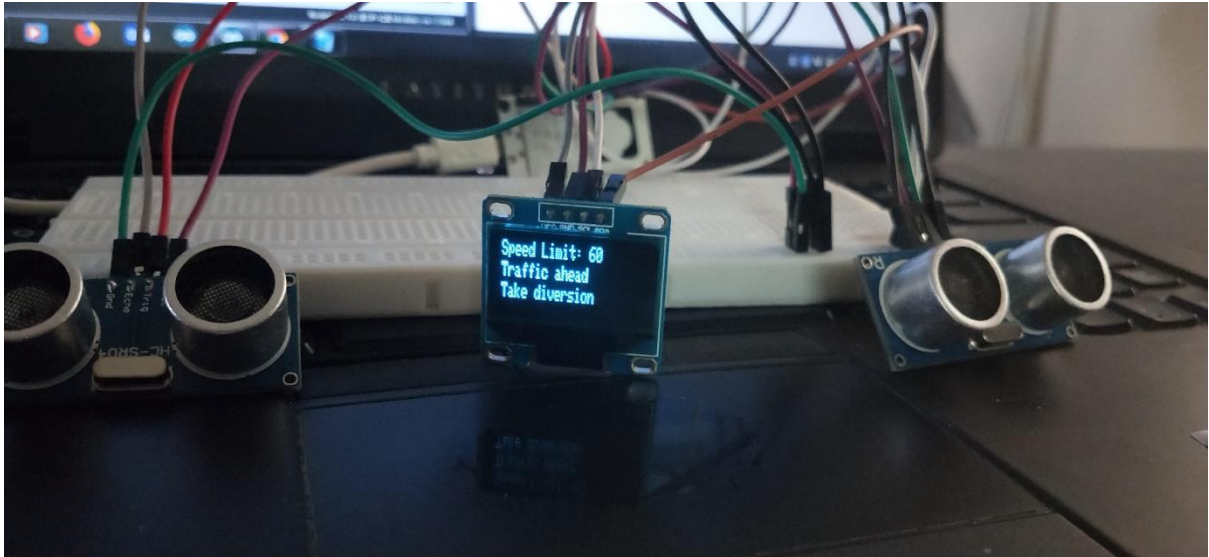


Figure 8-When traffic is detected

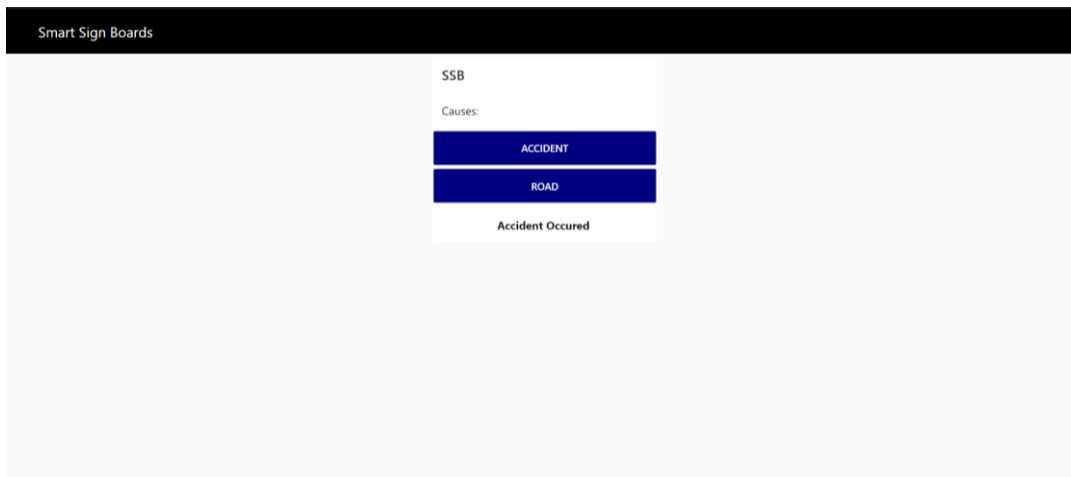


Figure 9- Web App



Figure 10-Information sign boards



*Figure 11-ISB*



*Figure 12-change of speed limit based on weather API*

## **10.CONCLUSION AND FUTURE SCOPE:**

### **Conclusion:**

Therefore, the signboards are digitalised and the information is entered dynamically and remotely. Hence lot of time of the commuter is saved and the changing speed limits will reduce the number of accidents caused due weather fluctuation.

### **Limitations:**

- Lack of secured Authorization

### **Future Scope:**

- Mobile app can be made for easy access.
- To avoid misuse of the application security credentials (login) can be added to the web app.

## **11.LEARNING RESOURCES:**

- <https://www.arduino.cc/>
- [http://www.nodemcu.com/index\\_en.html](http://www.nodemcu.com/index_en.html)
- <https://dzone.com/articles/4-free-weather-providers-api-to-develop-weather-ap-1>