

**PROJECT REPORT**

**IBM NALAYATHIRAN 2022-2023**

**PROJECT NAME :**

**SIGNS WITH SMART CONNECTIVITY FOR  
BETTER ROAD SAFETY**

**TEAM ID :**

**PNT2022TMID54258**

**TEAM MEMBERS :**

**NARESH .S (TEAM LEADER)**

**NIRMAL .S**

**GOPINATH .R**

**SAI VIGNESH .S**

# PROJECT REPORT

## Project Report Format

1. **INTRODUCTION**
  - 1.1 Project Overview
  - 1.2 Purpose
2. **LITERATURE SURVEY**
  - 2.1 Existing problem
  - 2.2 References
  - 2.3 Problem Statement Definition
3. **IDEATION & PROPOSED SOLUTION**
  - 3.1 Empathy Map Canvas
  - 3.2 Ideation & Brainstorming
  - 3.3 Proposed Solution
  - 3.4 Problem Solution fit
4. **REQUIREMENT ANALYSIS**
  - 4.1 Functional requirement
  - 4.2 Non-Functional requirements
5. **PROJECT DESIGN**
  - 5.1 Data Flow Diagrams
  - 5.2 Solution & Technical Architecture
  - 5.3 User Stories
6. **PROJECT PLANNING & SCHEDULING**
  - 6.1 Sprint Planning & Estimation
  - 6.2 Sprint Delivery Schedule
  - 6.3 Reports from JIRA
7. **CODING & SOLUTIONING (Explain the features added in the project along with code)**
  - 7.1 Feature 1
  - 7.2 Feature 2
  - 7.3 Database Schema (if Applicable)
8. **TESTING**
  - 8.1 Test Cases
  - 8.2 User Acceptance Testing
9. **RESULTS**
  - 9.1 Performance Metrics
10. **ADVANTAGES & DISADVANTAGES**
11. **CONCLUSION**
12. **FUTURE SCOPE**
13. **APPENDIX**
  - Source Code
  - GitHub & Project Demo Link

## **SIGNS WITH SMART CONNECTIVITY FOR BETTER ROAD**

**SAFETY TEAM ID - PNT2022TMID54258**

### **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 driver unable to see sign board and update it automatically based on API and warn drivers for school zones and hospital zones.

#### **1.2 PURPOSE**

- To replace the static signboards, 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.
  - It has displayed at School and Hospital Zones which can we set using buttons.

### **2. LITERATURE SURVEY**

#### **2.1 EXISTING PROBLEM**

- Rain makes brakes inefficient and leads to accidents
- The road area is awareness in resoures in reduces accidents
- Traffic diversion requires human intervention

## PROJECT REPORT

### 2.2 REFERENCES

- Andrzej Czyżewski in his paper titled "**Development of Intelligent Road Signs with V2X Interface for Adaptive Traffic Controlling**", IEEE 2019, developed IOT based intelligent road signs capable of interacting with both the vehicles and other neighbouring sign boards using LORA. These sign boards were capable of communicating with one another and changing the speed limit based on traffic and weather.
- Muhammed O. Sayin, Chung-Wei Lin, Eunsuk Kang, Shinichi Shiraishi & Tamer Basar in their paper titled "**Reliable Smart Road Signs**", IEEE 2019, proposed a game theoretical adversarial intervention detection mechanism for reliable smart road signs. A future trend in intelligent transportation systems is "smart road signs" that incorporate smart codes (e.g., visible at infrared) on their surface to provide more detailed information to smart vehicles.
- L.F.P. Oliveira, L.T. Manera, P.D.G. Luz in their paper titled "**Smart Traffic Light Controller System**", IEEE 2019, developed smart traffic lights capable of traffic accident detection enabling the enhancement of traffic light management systems, blocking and creating alternative routes to not only avoid the traffic jams, but also avoid new accidents.
- Dariusz Grabowski & Andrzej Czyzewski in their paper titled "**System for monitoring road slippery based on CCTV cameras and convolutional neural networks**", Springer Publications 2020, made use of Convolutional Neural Networks to identify slippery roads using CCTV cameras.

### 2.3 PROBLEM STATEMENT DEFINITION

In present system the road sign and speed limits are static

In some cases due to heavy traffic there we can change the road signs if they are digitalized.

Rainfall the roads will be slippery and speed limit would be decreased.

## PROJECT REPORT

### 3. IDEATION AND PROPOSED SOLUTION

#### 3.1 EMPATHY MAP CANVAS

<https://github.com/IBM-EPBL/IBM-Project-42203-1660656005/blob/8eedde90221897f268fc394c65d7bfe242b6eb6d/Project%20Phase%20Planning/ideation%20phase/Empathy%20map.Team.pdf>



## PROJECT REPORT

### 3.2 IDEATION & BRAINSTORMING

<https://github.com/IBM-EPBL/IBM-Project-42203-1660656005/blob/8eedde90221897f268fc394c65d7bfe242b6eb6d/Project%20Phase%20Planning/ideation%20phase/Brainstorm%20&Idea%20Template.pdf>

### 3.3 PROPOSED SOLUTION

<https://github.com/IBM-EPBL/IBM-Project-42203-1660656005/blob/8eedde90221897f268fc394c65d7bfe242b6eb6d/Project%20Phase%20Planning/Project%20phase%20-I/prosposed%20solution%20.team.pdf>

### 3.4 PROBLEM SOLUTION FIT

<https://github.com/IBM-EPBL/IBM-Project-42203-1660656005/blob/8eedde90221897f268fc394c65d7bfe242b6eb6d/Project%20Phase%20Planning/Project%20phase%20-I/Solution%20Fit..pdf>

# PROJECT REPORT

## 4. REQUIREMENT ANALYSIS

### 4.1 FUNCTIONAL REQUIREMENTS

<https://github.com/IBM-EPBL/IBM-Project-42203-1660656005/blob/8eedde90221897f268fc394c65d7bfe242b6eb6d/Project%20Phase%20Planning/Project%20Phase%20-II/solution%20requirements.pdf>

### 4.2 NON-FUNCTIONAL REQUIREMENTS

<https://github.com/IBM-EPBL/IBM-Project-42203-1660656005/blob/8eedde90221897f268fc394c65d7bfe242b6eb6d/Project%20Phase%20Planning/Project%20Phase%20-II/solution%20requirements.pdf>

#### Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement(Epic)	Sub Requirement(Story/Sub-Task)
FR-1	User Visibility	Sign Boards should be made with LED's which are bright colored and are capable of attracting the drivers attention but it should also not be too distracting or blinding cause it may lead to accidents.
FR-2	User Understanding	For better understanding of the driver, the signs should be big, clear and legible and it can also include illustrations which will make it easily understandable to the driver.
FR-3	User Convenience	The display should be big enough that it should even be visible from far distance clearly.

#### Non-Functional Requirements:

Following are the Non-Functional Requirements of the proposed solution

FR No.	Non-Functional Requirements	Description
NFR-1	Usability	It should be able to Upgrade and Update when there is a need for it.
NFR-2	Security	It should have good security system so that no other person is able to hack and display their own directions.
NFR-3	Reliability	It should be able to display to information correctly and error-free.
NFR-4	Performance	It should be able to automatically update itself when certain weather or traffic problem occurs.
NFR-5	Availability	It should be available 24/7 so that it can be beneficial to the customer i.e the driver.
NFR-6	Scalability	It should be able to easily change and upgrade according to change and need in requirement.

# PROJECT REPORT

## 5. PROJECT DESIGN

### 5.1 DATA FLOW DIAGRAMS



### 5.2 SOLUTION & TECHNICAL ARCHITECTURE

<https://github.com/IBM-EPBL/IBM-Project422031660656005/blob/8eedde90221897f268fc394c65d7bfe242b6eb6d/Project%20Phase%20Planning/Project%20Phase%20-II/Technology%20Architecture.pdf>



## PROJECT REPORT

### 5.3 USER STORIES

<https://github.com/IBM-EPBL/IBM-Project-42203-1660656005/blob/8eedde90221897f268fc394c65d7bfe242b6eb6d/Project%20Phase%20Planning/Project%20Phase%20-II/Custom%20Journey%20Map.pdf>

## 6. PROJECT PLANNING AND SCHEDULING PHASE

### 6.1 SPRINT PLANNING & ESTIMATION

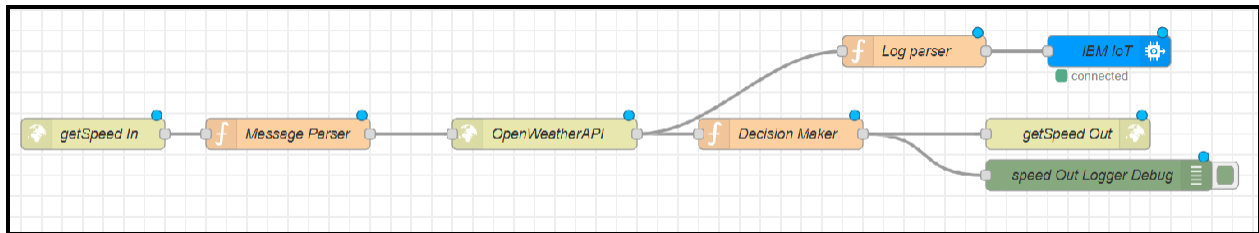
<https://github.com/IBM-EPBL/IBM-Project-42203-1660656005/blob/8eedde90221897f268fc394c65d7bfe242b6eb6d/Project%20Phase%20Planning/project%20planning/Milestone%20and%20Activity%20List..pdf>

### 6.2 SPRINT DELIVERY SCHEDULE

[https://github.com/IBM-EPBL/IBM-Project-42203-1660656005/blob/8eedde90221897f268fc394c65d7bfe242b6eb6d/Project%20Phase%20Planning/project%20planning/Project%20&%20Sprint%20Delivery%20Plan%20\(1\).pdf](https://github.com/IBM-EPBL/IBM-Project-42203-1660656005/blob/8eedde90221897f268fc394c65d7bfe242b6eb6d/Project%20Phase%20Planning/project%20planning/Project%20&%20Sprint%20Delivery%20Plan%20(1).pdf)

## 7. CODING & SOLUTIONING

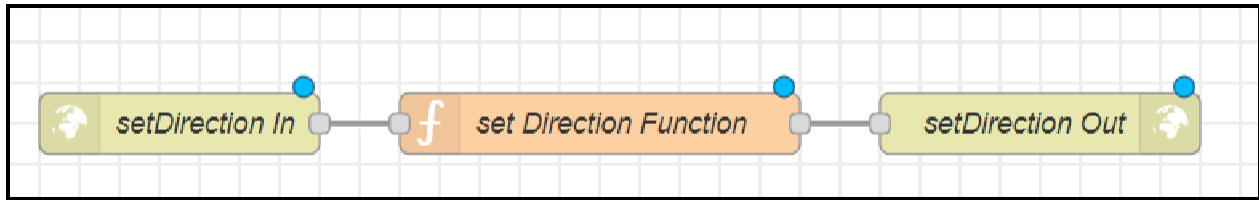
### 7.1 FEATURE 1 - GET SPEED FOR GIVEN LOCATION & CLIMATE



This part of Node RED flow accepts an http GET end point at **"/getSpeed"** from which the location, uid, hospital/school zone info are passed. Message parser sets the required APIKEY for OpenWeatherAPI for the next block. This data is then passed onto Decision Maker which makes all the decisions regarding the message to be output at the display and sends it as a http response. This data is displayed at the micro- controller. Thus a lot of battery is saved due to lesser processing time.

## PROJECT REPORT

### 7.2 FEATURE 2 - SET DIRECTION REMOTELY FOR A GIVEN SIGN BOARD



This part of Node RED flow accepts an http GET end point at **"/setDirection"** from which the uid and direction information are passed by the respective authorities. Set Direction Function block adds the direction information to the database and returns the same as an http response. This data is sent to the microcontroller along with the **"/getSpeed"** path and the microcontroller displays it.

A detailed documentation of all the workflows is available at the following link :  
<https://github.com/IBM-EPBL/IBM-Project-42203-1660656005/blob/8eedde90221897f268fc394c65d7bfe242b6eb6d/project%20development%20phase/sprint%201/sprint%204/readme.py.pdf>

## 8. TESTING

### 8.1 TEST CASES

- **TEST CASE 1**

Clear weather - Usual Speed Limit.

- **TEST CASE 2**

Foggy Weather - Reduced Speed Limit.

- **TEST CASE 3**

Rainy Weather - Further Reduced Speed Limit.

- **TEST CASE 4**

School/Hospital Zone - Do not Honk sign is displayed.

### 8.2 USER ACCEPTANCE TESTING

Dynamic speed & diversion variations based on the weather and traffic helps user to avoid traffic and have a safe journey home. The users would welcome this idea to be implemented everywhere.

## SIGNS WITH SMART CONNECTIVITY FOR BETTER ROAD

# PROJECT REPORT

## 9. RESULTS

### 9.1 PERFORMANCE METRICS

Based on the IBM pack we chose, the performance of the website varies. Built upon NodeJS, a light and high performance engine, NodeRED is capable of handling upto 10,000 requests per second. Moreover, since the system is horizontally scalable, a even higher demand of customers can be served.

## 10. ADVANTAGES & DISADVANTAGES

- **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.

## PROJECT REPORT

### 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 AND PROJECT DEMO LINK

<https://github.com/IBM-EPBL/IBM-Project-42203-1660656005>

- DEMO VIDEO DOWNLOAD LINK

[https://github.com/IBM-EPBL/IBM-Project-42203-1660656005/blob/7c868e63bd99352a10fcad389c1dd6bfc6957c22/final%20deliverable/demo%20link/QV\\_1668826079028.mp4](https://github.com/IBM-EPBL/IBM-Project-42203-1660656005/blob/7c868e63bd99352a10fcad389c1dd6bfc6957c22/final%20deliverable/demo%20link/QV_1668826079028.mp4)

- SOURCE CODE - ESP 32

```
1 #include <WiFi.h>
2 #include <HTTPClient.h>
3 #include <Adafruit_GFX.h>
4 #include <Adafruit_ILI9341.h>
5 #include <string.h>
6
7 const char* ssid = "Wokwi-GUEST";
8 const char* password = "";
9
10 #define TFT_DC 2
11 #define TFT_CS 15
12 Adafruit_ILI9341 tft = Adafruit_ILI9341(TFT_CS, TFT_DC);
13
14 String myLocation = "Chennai,IN";
```

## PROJECT REPORT

```
15 String usualSpeedLimit = "70"; // kmph
16
17 int schoolZone = 32;
18 int hospitalZone = 26;
19
20 int uid = 2504; // ID Unique to this MicroContoller
21
22 String getString(char x)
23 {
24     String s(1, x);
25     return s;
26 }
27
28 String stringSplitter1(String fullString, char delimiter='$')
29 {
30     String returnString = "";
31     for(int i = 0; i<fullString.length();i++) {
32         char c = fullString[i];
33         if(delimiter==c)
34             break;
35         returnString+=String(c); 36
36     }
37     return(returnString);
38 }
39
40 String stringSplitter2(String fullString, char delimiter='$')
41 {
42     String returnString = "";
43     bool flag = false;
```

## PROJECT REPORT

```
44     for(int i = 0; i<fullString.length();i++) {
45         char c = fullString[i];
46         if(flag)
47             returnString+=String(c);
48         if(delimiter==c)
49             flag = true; 50     }
51     return(returnString);
52 }
53
54 void rightArrow()
55 {
56     int refX = 50;
57     int refY = tft.setCursorY() + 40;
58
59     tft.fillRect(refX,refY,100,20,ILI9341_RED);
60     tft.fillTriangle(refX+100,refY-
61         30,refX+100,refY+50,refX+40+100,refY+10,ILI9341_RED);
62 }
63 void leftArrow()
64 {
65     int refX = 50;
66     int refY = tft.setCursorY() + 40;
67
68     tft.fillRect(refX+40,refY,100,20,ILI9341_RED);
69     tft.fillTriangle(refX+40,refY-
70         30,refX+40,refY+50,refX,refY+10,ILI9341_RED);
71 }
```

## PROJECT REPORT

```
71
72 void upArrow()
73 {
74     int refX = 125;
75     int refY = tft.getCursorY() + 30;
76
77         tft.fillTriangle(refX-
78             40,refY+40,refX+40,refY+40,refX,refY,ILI9341_RED);
79     tft.fillRect(refX-15,refY+40,30,20,ILI9341_RED);
80 }
81 String APICall() {
82     HTTPClient http;
83
84     String url = "https://node-red-grseb-2022-11-05-test.eu-
85         gb.mybluemix.net/getSpeed?";
86     url += "location="+myLocation+"&";
87     url += "schoolZone="+ (String) digitalRead(schoolZone) + (String) "&";
88     url += "hospitalZone="+ (String) digitalRead(hospitalZone) + (String) "&";
89     url += "usualSpeedLimit="+ (String) usualSpeedLimit + (String) "&";
90     http.begin(url.c_str());
91     int httpResponseCode = http.GET();
92
93     if (httpResponseCode>0) {
```

## PROJECT REPORT

```
94     String payload = http.getString();
95     http.end();
96     return(payload);
97 }
98 else {
99     Serial.print("Error code: ");
100     Serial.println(httpResponseCode);
101 }
102 http.end();
103 }
104
105 void myPrint(String contents) {
106     tft.fillScreen(ILI9341_BLACK);
107     tft.setCursor(0, 20);
108     tft.setTextSize(4);
109     tft.setTextColor(ILI9341_RED);
110     //tft.println(contents);
111
112     tft.println(stringSplitter1(contents));
113     String c2 = stringSplitter2(contents);
114     if(c2=="s") // represents Straight
115     {
116         upArrow();
117     }
118     if(c2=="l") // represents left
119     {
120         leftArrow();
121     }
122     if(c2=="r") // represents right
```



## PROJECT REPORT

```
123     {
124         rightArrow();
125     }
126 }
127
128 void setup() {
129     WiFi.begin(ssid, password, 6);
130
131     tft.begin();
132     tft.setRotation(1);
133
134     tft.setTextColor(ILI9341_WHITE);
135     tft.setTextSize(2);
136     tft.print("Connecting to WiFi");
137
138     while (WiFi.status() != WL_CONNECTED) {
139         delay(100);
140         tft.print(".");
141     }
142
143     tft.print("\nOK! IP=");
144     tft.println(WiFi.localIP());
145 }
146
147 void loop() {
148     myPrint(APICall());
149     delay(100);
150 }
```