IBM NALAIYA THIRAN 2022-23 PROJECT REPORT

SIGNS WITH SMART CONNECTIVITY FOR BETTER ROAD SAFETY

TEAM ID - PNT2022TMID32727

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 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.
- "DO NOT HONK" message 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

- Fog reduces visibility and increases the probability of accidents
- Traffic diversion requires human intervention

2.2 REFERENCES

- Jeong Ah Jang *, Hyun Suk Kim and Han Byeog Cho in their paper titled

" Smart Roadside System for Driver Assistance and Safety " which was published in 25 July 2011 , introduced a smart roadside system that utilizes Various sensors for driver assistance and traffic safety warnings. This paper shows two road Application models for a smart roadside system and sensors: a red-light violation warning System for signalized intersections, and a speed advisory system for highways. Evaluation Results for the two services are then shown using a micro-simulation method.

- Chai k. Toh and Juan Carlos Cano in their paper titled "Wireless Digital Traffic Signs of the Future " published in 7th September 2018 , the wireless digital traffic sign post architecture replaces the existing physical traffic signs with posts embedded with electronics hardware and software. The hardware functions as a communication device and server. The traffic sign is then wirelessly transmitted to a client receiver residing in the vehicle. The client device usually a smart phone or a car HUD. It can also be displayed on a car ADAS terminal.
- Roopa. M.S., Ayesha Siddiq. S., Rajkumar Buyya., Venugopal. K.R., Iyengar. S.S., & Patnaik. L. M. in their paper titled " Dynamic Management of Traffic signals through social IOT" published in the year 2020, they proposed a Dynamic congestion control with Throughput Maximization scheme based on Social Aspect (D-TMSA) utilizing the social, behavioral and preference-based

relationships. Their proposed scheme along with the various social relationship types allocates green signal to maximize the traffic flow passing through an intersection. Simulation results show that the DTMSA outperforms the existing work by achieving high throughput, lowering the total traveling time and reducing the average waiting time to better the flow of traffic based on their social attributes with each other.

- Bhumika R, Harshitha S A, Meena D and Asha M in their paper " Accident Prevention and Road Safety in Hilly Region using IOT Module" published in 20th July 2021,examined a module which means to screen and improve the security in sloping regions by utilizing Wireless Sensor Network and Internet of Things.

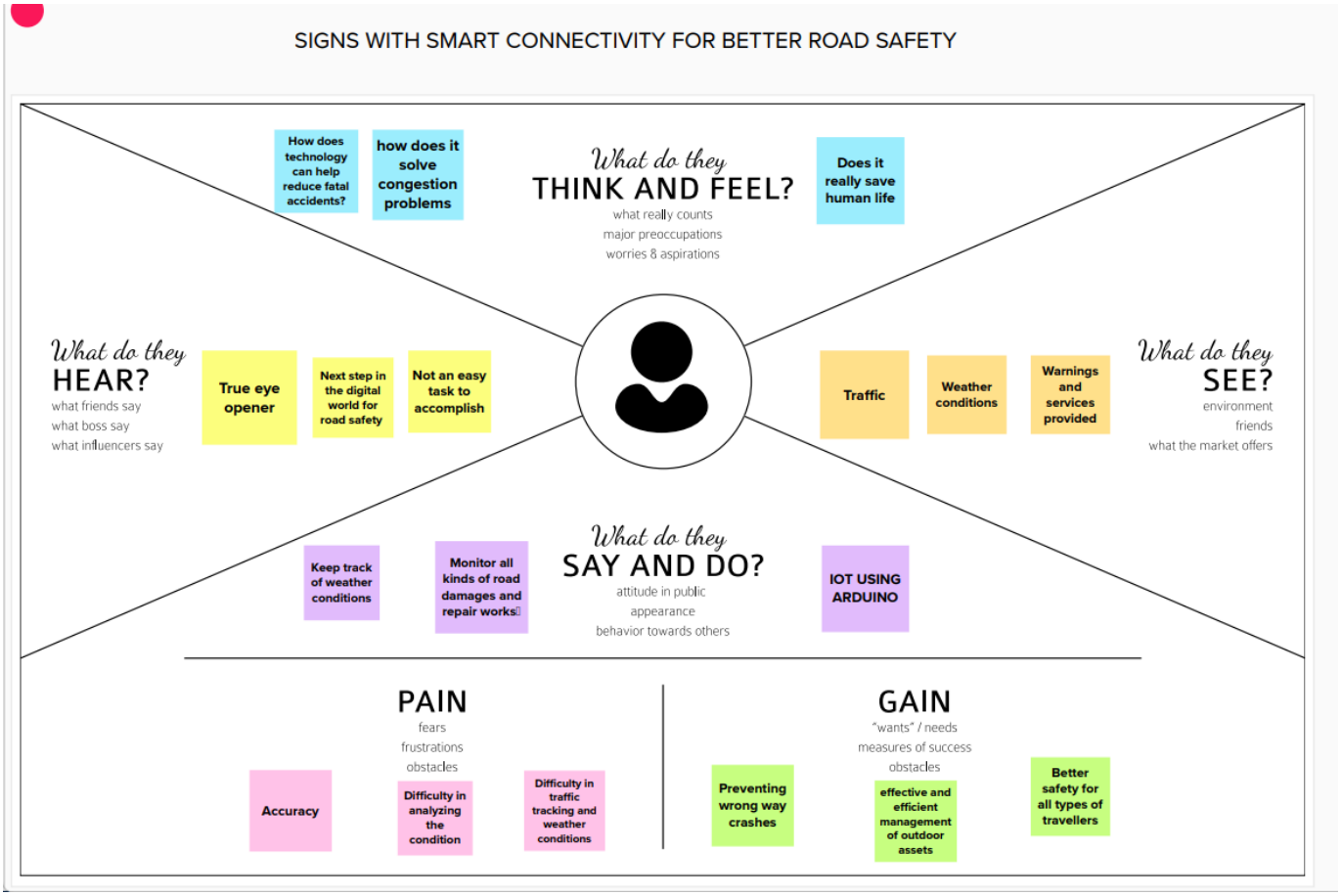
2.3 PROBLEM STATEMENT DEFINITION

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 SOLUTION

3.1 EMPATHY MAP CANVAS

https://github.com/IBM-EPBL/IBM-Project-15566-1659600519/tree/main/Ideation%20phase



3.2 IDEATION & BRAINSTORMING

https://github.com/IBM-EPBL/IBM-Project-15566-1659600519/tree/main/Ideation%20phase

3.3 PROPOSED SOLUTION

- Use a 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 returns processes the data at cloud and returns only the message to be displayed at the sign board display.

SIGNS WITH SMART CONNECTIVITY FOR BETTER ROAD SAFETY

- Another http end point is configured to set the direction to be displayed. Upon accessing this http end point, the direction is set remotely for a display using it's unique id.

3.4 PROBLEM SOLUTION FIT

- The display replaces the static signs
- Processing requirement of 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 REQUIREMENTS

https://github.com/IBMEPBL/IBMProject-155661659600519/blob/main/DESIGN%20PHASE%202/SOLUTION%20REQUIREMENT.pdf

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User visibility	Visibility refers to a motorist's ability to clearly see traffic and environmental surroundings on the road.
FR-2	User need	A proper knowledge about traffic rules is very important for any driver, our aim is to give the user solution for better road safety and to reduce number of accidents that occur every year.
FR-3	User understanding	Road traffic safety refers to the methods and measures used to prevent traveller from being killed by an accident or seriously injured.
FR-4	User convenience	Passive traffic safety measures sought to avoid influencing the behavior of drivers while giving vehicles maximum convenience.

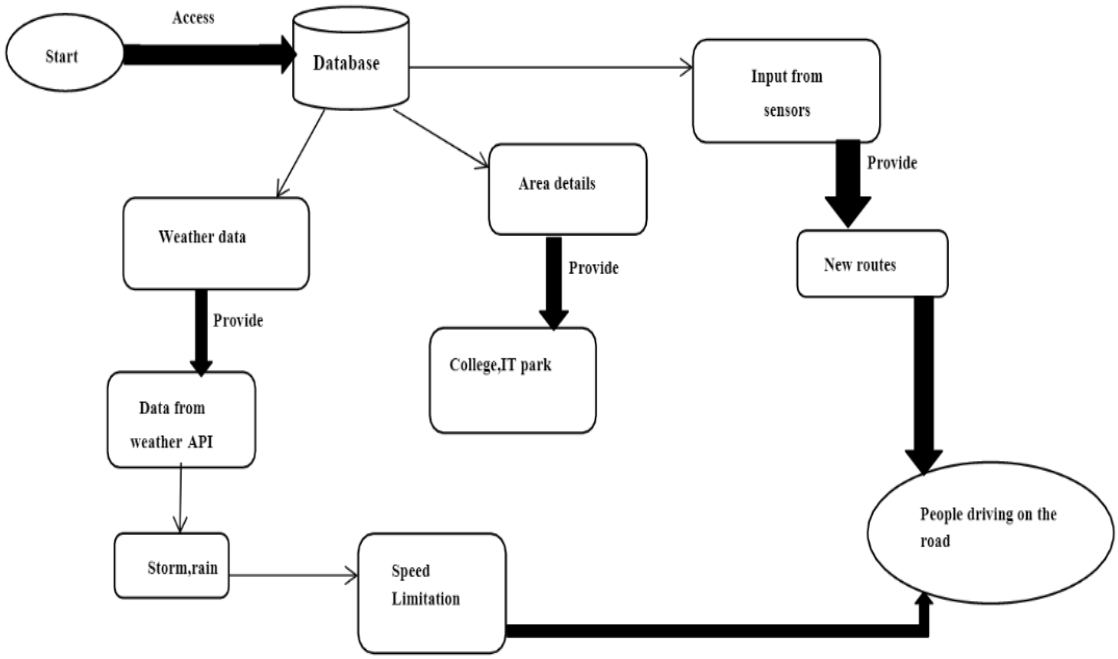
4.2 NON-FUNCTIONAL REQUIREMENTS https://github.com/IBM-EPBL/IBM-Project-15566-1659600519/blob/main/DESIGN%20PHASE%202/SOLUTION%20REQUIREMENT.pdf

Non-functional Requirements:

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

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	It should easy to see.
NFR-2	**Security**	Two wheelers must always wear helmets and for four wheelers both driver and passengers put on their seatbelts.
NFR-3	**Reliability**	Traffic and weather updates must be great accuracy And errorless.
NFR-4	**Performance**	It must work in high speed.
NFR-5	**Availability**	It must be available for the user 24/7.
NFR-6	**Scalability**	The device must be able to adapt to the changes in the environment and meet the changing needs in the future.

5. PROJECT DESIGN



5.2 SOLUTION & TECHNICAL ARCHITECTURE https://github.com/IBM-EPBL/IBM-Project-15566-1659600519/blob/main/DESIGN%20PHASE%20-1/Solution%20Architecture.pdf

5.3 USER STORIES

https://github.com/IBM-EPBL/IBM-Project-15566-1659600519/blob/main/DESIGN%20PHASE%202/Customer%20journey.pdf

6. PROJECT PLANNING AND SCHEDULING PHASE 6.1

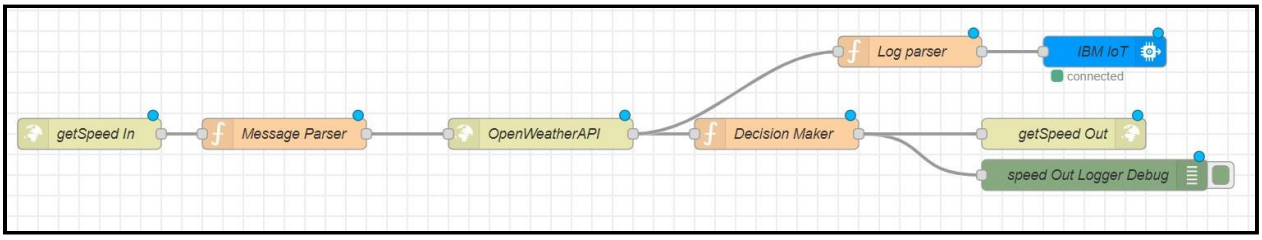
SPRINT PLANNING & ESTIMATION

https://github.com/IBM-EPBL/IBM-Project-15566-1659600519/blob/main/Project%20planning/Sprint%20delivery%20plan.pdf

6.2 SPRINT DELIVERY SCHEDULE **https://github.com/IBM-EPBL/IBM-Project-15566-1659600519/blob/main/Project%20planning/Sprint%20delivery%20plan.pdf**

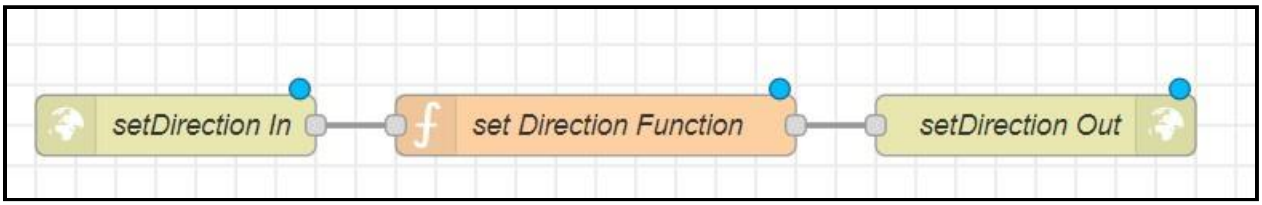
7. CODING & SOLUTIONING

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 microcontroller. Thus a lot of battery is saved due to lesser processing time.

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-15566-1659600519/tree/main/Project%20Development%20Phase/Sprint%204

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/Hosipital Zone - Do not Honk sign is displayed.

8.2 USER ACCEPTANCE TESTING

Dynamic speed & divertion 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.

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.

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-15566-1659600519

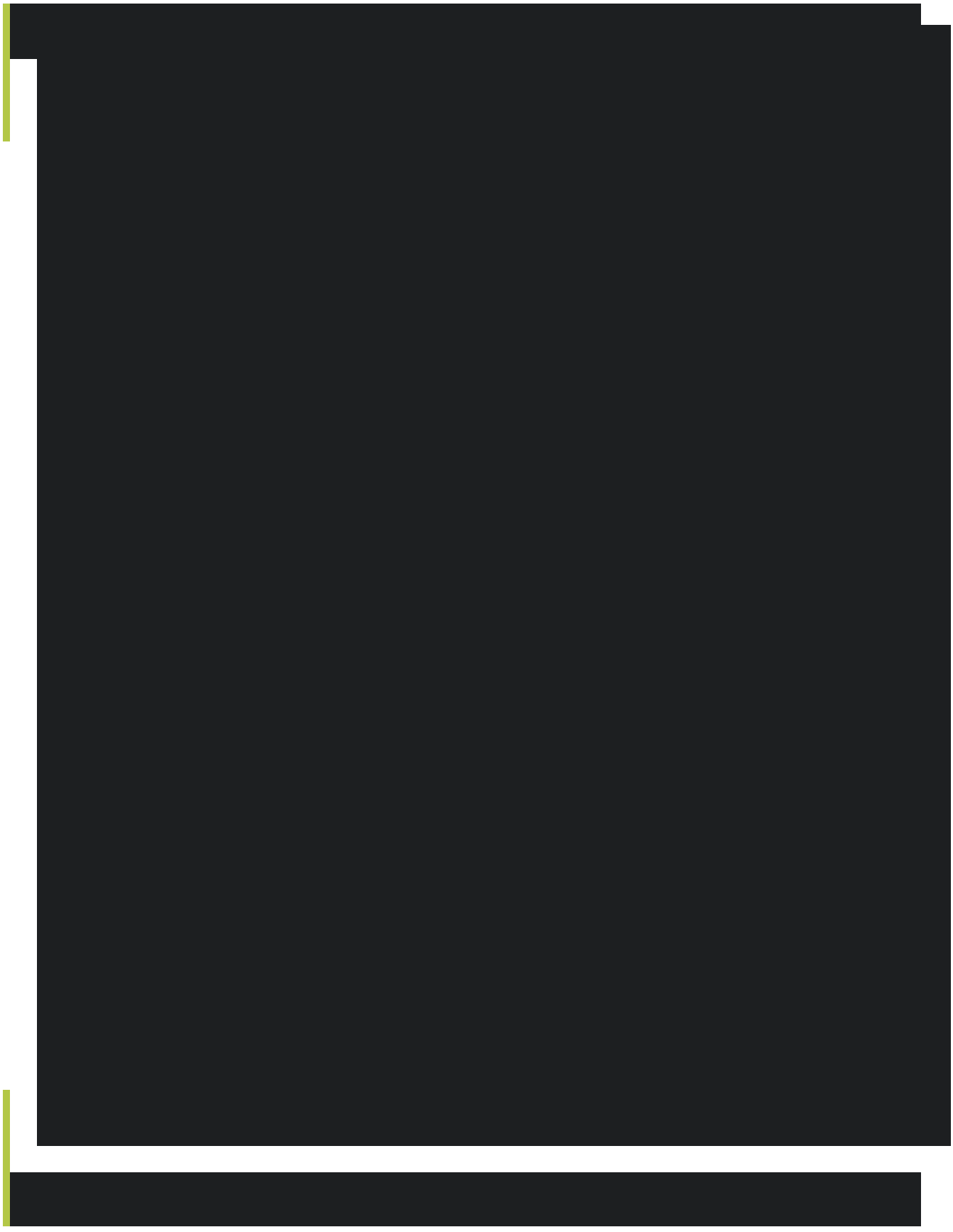
- **DEMO VIDEO DOWNLOAD LINK**

  https://github.com/IBM-EPBL/IBM-Project-15566-1659600519/blob/main/Final%20Deliverables/Demovideo%20(2).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
12Adafruit_ILI9341 tft = Adafruit_ILI9341(TFT_CS, TFT_DC);
13
14String myLocation = "Chennai , IN";
15String usualSpeedlimit = "70";//kmph
```

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



SIGNS WITH SMART CONNECTIVITY FOR BETTER ROAD SAFETY

```
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
54void rightArrow()
55{
56    int refX = 50;
57    int refY = tft.getCursorY() + 40;
58
59                                              tft.fillRect(refX,refY,100,20,ILI9341_RED);
60                                              tft.fillTriangle(refX+100,refY-
  30,refX+100,refY+50,refX+40+100,refY+10,ILI9341_RED);
61}
62
63void leftArrow() 64{
65    int refX = 50;
66    int refY = tft.getCursorY() + 40;
      67
```

```
68 tft.fillRect(refX+40,refY,100,20,ILI9341_RED);
        tft.fillTriangle(refX+40,refY-
        30,refX+40,refY+50,refX,refY+10,ILI9341_RED);
```

```
72void upArrow()
73{
74      int refX = 125;
75      int refY = tft.getCursorY() + 30;
76
77                                                      tft.fillTriangle(refX-
    40,refY+40,refX+40,refY+40,refX,refY,ILI9341_RED);
78                                                      tft.fillRect(refX-
                                                        15,refY+40,30,20,ILI9341_RED);
79}
80
81String APICall() { 82  HTTPClient
http;
83
84                                                  String url = "https://node-red-grseb-
                                                    2022-11-05-
                                                    test.eugb.mybluemix.net/getSpeed?";
85                                                  url += "location="+myLocation+"&";
86                                                  url       +=
                    "schoolZone="+(String)digitalRead(schoolZone)+(String)"&";
87                                                  url       +=
    "hospitalZone="+(String)digitalRead(hospitalZone)+(String)"& ";
```

```
88                                                          url      +=
89"usualSpeedLimit="+(String)usualSpeedLimit+(String)"&";
90 url += "uid="+(String)uid;
    http.begin(url.c_str());
91 int httpResponseCode = http.GET();
92
93 if (httpsResponseCode>0){
94 String payload = https.getString();
```

if 0 String

```
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         }
            if      "r"  // represents right
```

```
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   }
```