IBM NALAIYA THIRAN 2022-23 PROJECT REPORT

SIGNS WITH SMART CONNECTIVITY FOR BETTER ROAD SAFETY

TEAM ID - PNT2022TMID10531

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

2.2 REFERENCES

S.No	Author	Paper Title	Journal & Year	Remarks / Critics
1.	Ashish Dhar	Traffic and road condition monitoring system	Indian Institute of Technology, Mumbai. - 2008.	• Reports severity, intensity and dimension of a damaged road segment. • Proposed a different solution using AMR Magnetic Sensor.
2.	Pooja Pawar, Suvarna Langade, Mohini Bandgar	IOT Based digital Notice Board using Arduino ATMega 328.	International Research Journal of Engineering and Technology(IRJET). - 2019.	• Circulates notice regularly & reduce physical efforts. • Send message at any distant location within a second.
3.	Sandeep Chaware, Trushitha Chaware.	Proposed Algorihm for Smart Traffic Control using Ultrasonic Sensor.	International Journal of Engineering and Advanced Technology(IJEAT).	The outcome of the project is to learn insights of the traffic controlling and management at the signal with the dynamically changing in timing of timer as per need.

			- 2019.	
4.	Kamna Singh, Deepa Bura	IOT: distinct algorithms for the Sensor Connectivity with Comparative Study between node MCU and Arduino MCU.	NVEO Journal – 2021	• Presents different algorithms for the connection between different types of sensors. • Brief description of node MCU & Arduino MCU. • Stepby step solution to provide connectivity with IOT technology.
5.	Jack Greenhaigh	Recognizing Text Based Traffic Signs.	IEEE – 2015	• Detect all possible Road sign candidates. • Reduce total regions based on contextual constraints.

				A Novel System for the automatic detection and recognition of text in traffic sign based on MSER & MSV.

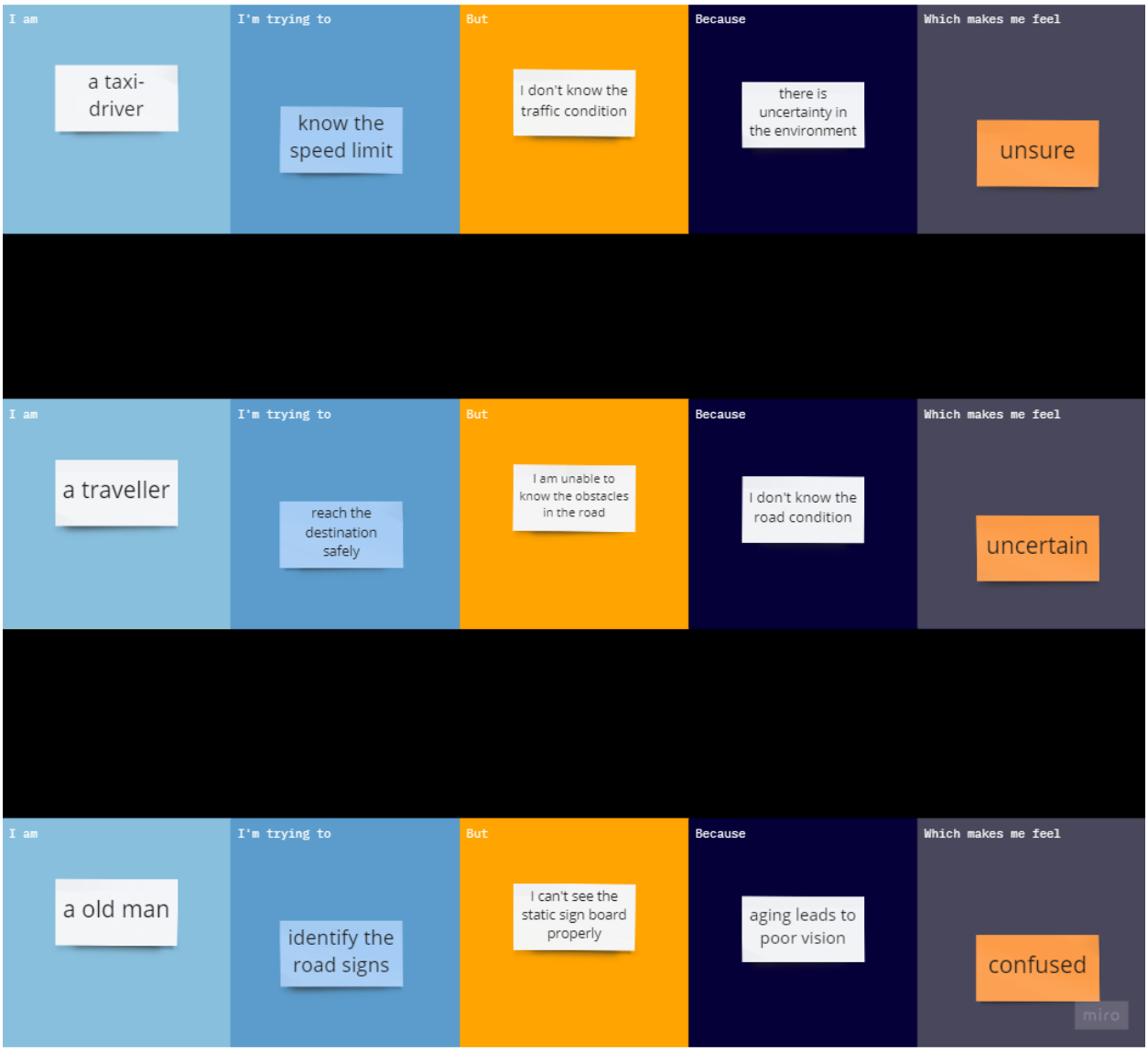
6.	Bhumika.R, Harshita. S.A, Meena. D, Asha. N	Accident Prevention and Road Safety in Hilly Region using IOT Module	International Research Journal of Engineering and Technology(IRJET). – 2021	Stay away from mishap & forestall clog in sloping region & hairclip twist. As a significant part of street mathematical plan bended street portion
7.	Sowparnika. B	IOT Road Safety		This project paves a system to alert the driver about the speed limit in specific areas and to reduce the speed of vehicles in sensitive public zones without any interference of drivers where controls are taken automatically by use of wireless local area network.
8.	S.S. Sugania, D. S. Vishalis Hwaran, J. Vignesh Kumar.	Automated System for Road Safety Enhancement using big data reports.		The speed is controlled accordingly to situations to give suggestions. The suggested system can control the vehicle but at same time can collect data and manipulate it using the big data technologies.
9.		IOT Based Smart Road Safety & Vehicle Accident prevent		This system is divided into 2 half (Accident Detection & Prevention) and alerting the members of family by causation message and placement of accidental place.

		System for Mountain roads.		
10.	Shweta Vyas, Pooja Awhale, Shreya Kukdeja, Prashant Jawalkar.	A Modern Approach to identify Traffic Sign Symbols in Color Images.		In this technique proposed more reliable and robust method of Traffic Sign Detection Recognition (TSDR).
11.	Deepika K. N, Sangeetha Thirumoorthy.	Internet Of Things Based Notifications using Smart Notice Board.	Sri Krishna College of Technology. - 2018	By using this system in the field of wireless communication we can make communication more effective, fast and very easy handling method. With the help of this, displaying of notices can be updated by every second from anywhere and anytime through a mobile phone.
12.	Chai K. Toh, Juan-Carlos Cano, Carlos Fernandez-Laguia, Pietro Manzoni,	Wireless digital traffic signs of the future.	The Instituition of Engineering and Technology(IET)	In this architecture notify the sign can be narrated via voice to driver, in addition to displaying on the dashboard. Changing a sign is easy as reprogramming it with advanced electronics and radio hardware embedded into poles, will be present to transmit programmed traffic signs wirelessly on the road.

	Carlos T. Calafate.			

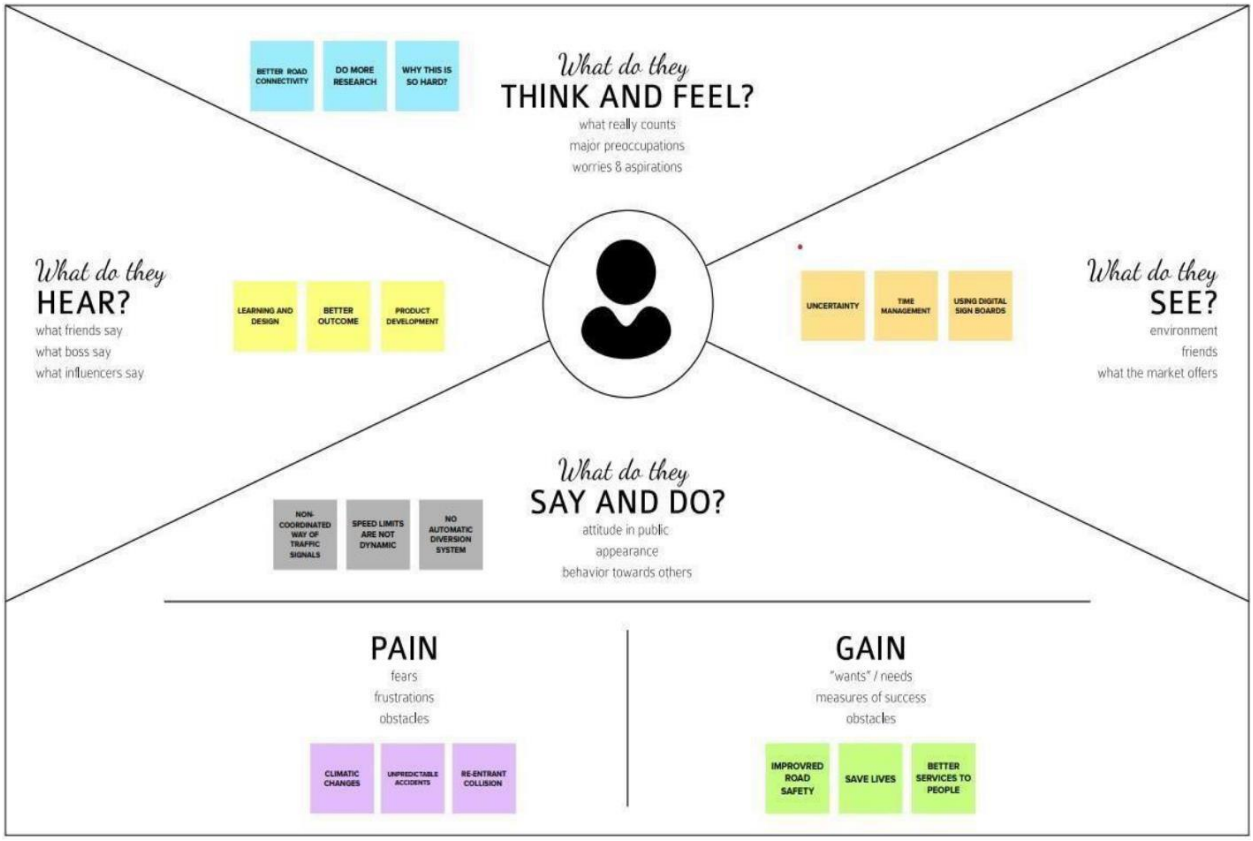
2.3 PROBLEM STATEMENT DEFINITION

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.

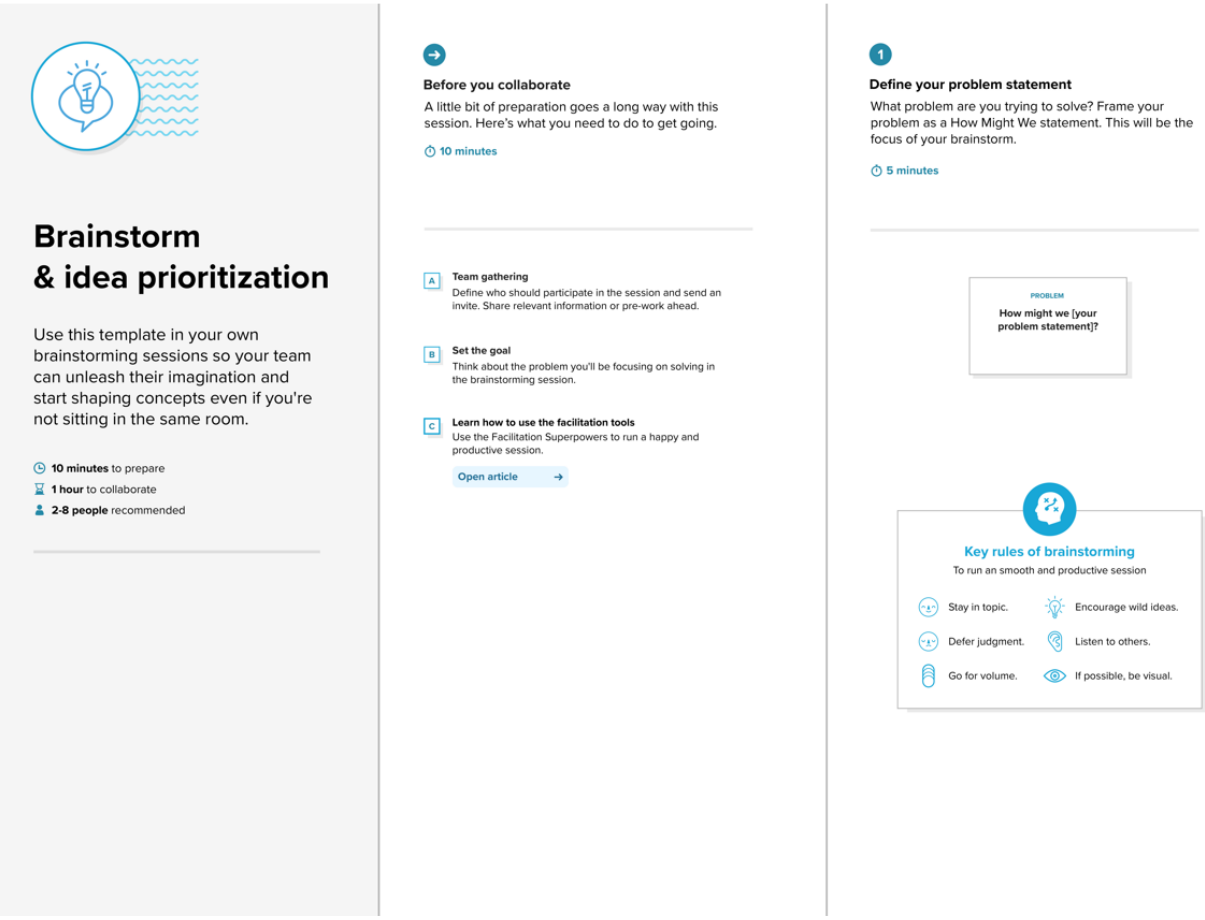


3. IDEATION AND PROPOSED SOLUTION

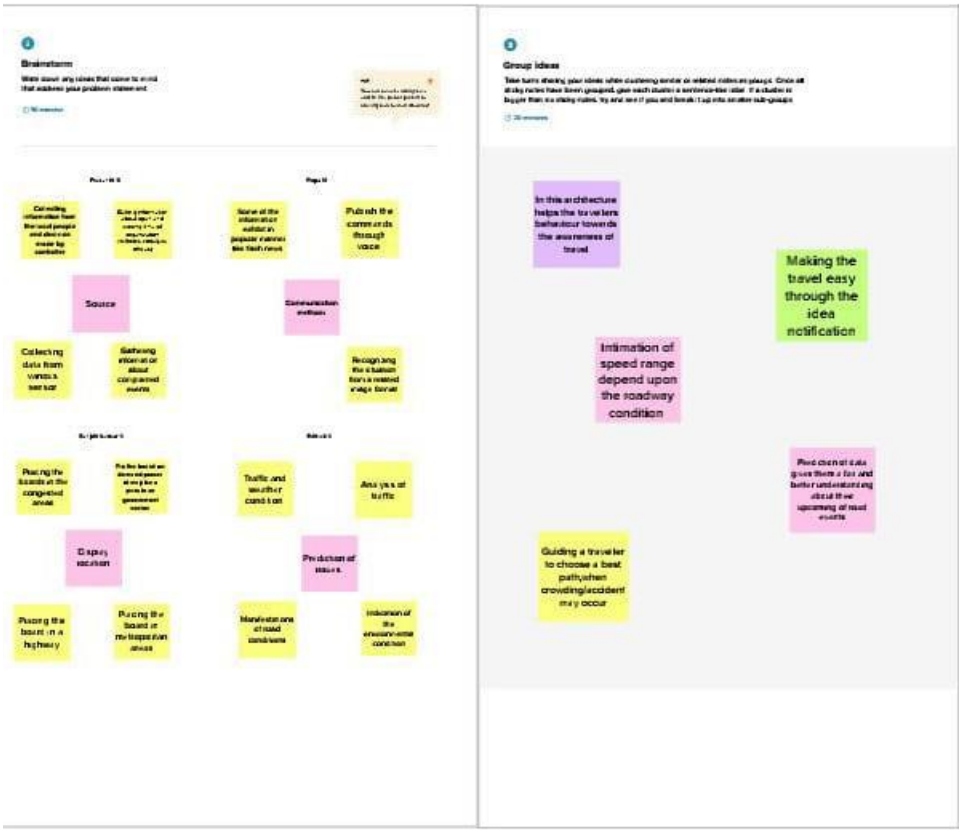
3.1 EMPATHY MAP CANVAS



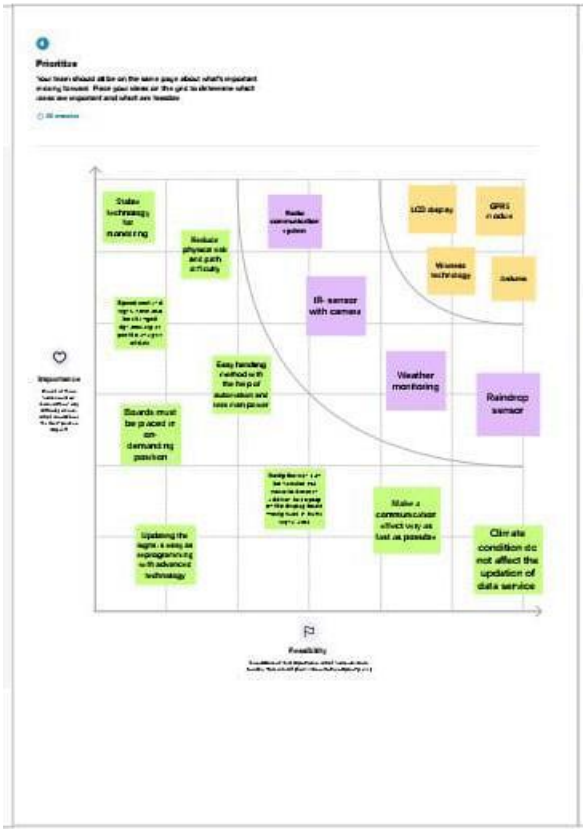
3.2 IDEATION & BRAINSTORMING



Brainstorm, Idea Listing and Grouping



Idea Prioritization



3.3 PROPOSED SOLUTION

S. No	Parameter	Description
1.	Problem Statement (Problem to be solved)	• The actual problem is that drivers are unable to know whether the road conditions is safe to travel or not. • Hence there will be a need of guidance data for providing safety and to avoid travelling inconvenience to reach destination.

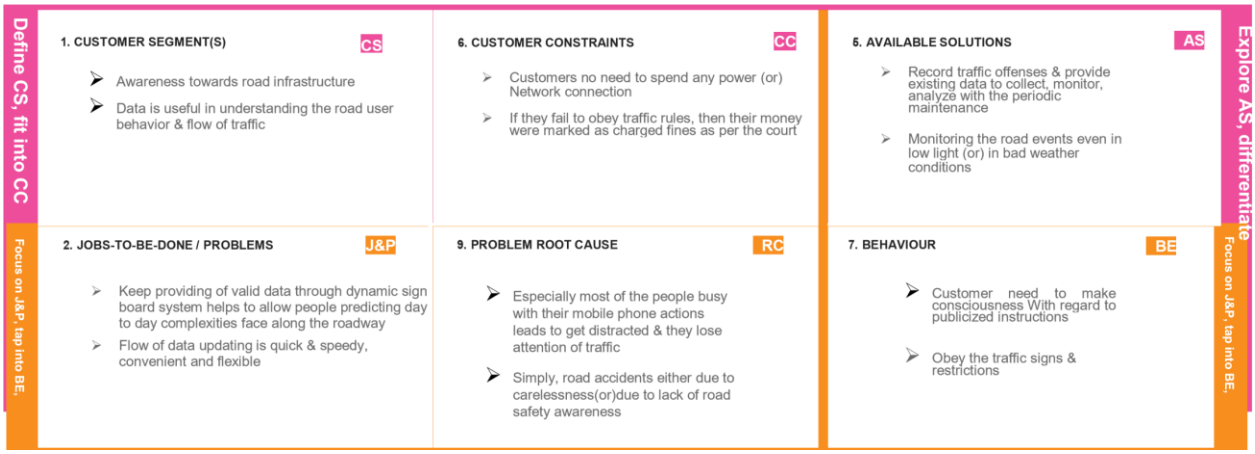
SIGNS WITH SMART CONNECTIVITY FOR BETTER ROAD SAFETY

2.	Idea / Solution description	This problem can be overcome by introducing the GPRS Module, IR Sensor with Camera to sense the traffic intensity even in dark areas. Rain drop sensor to indicate the accumulation of rain has occurred. And also collecting information from the local peoples and decision made by controller, who controls display manually(Manpower).
3.	Novelty / Uniqueness	Voice indicators are placed in near, the display board location adjusted to that traffic signal area. It will indicate the road dangers to the public as it senses the nearby vehicles. Speed limit changes according to the weather condition using rain drop sensor.
4.	Social Impact / Customer Satisfaction	Large number of accidents may be minimized by replacing smart signs instead of static signs. Obvious information only displayed. Reports severity.
		Sign changes dynamically depending upon the upcoming events.
5.	Business Model (Revenue Model)	Systematic reduces manpower. The systems can be used in public and private sectors which gives good revenue. This type of system is helpful for education and medical institutions.

6.	Scalability of the Solution	User friendly interface. Accessibility of data is easy from source. Precise information in sign boards can be easily captured.

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.



	3. TRIGGERS Creating a note that gives a direction on how to recognize that system effectively	TR	10. YOUR SOLUTION Pre-functional record of specific voice record mode of data along with LED display provide in waiting time at traffic signs In this proposed system is interface with Rain drop sensor check if it rainy there, to transmit data over IOT helps to display on LED to along with WIFI connection of internet changing

			data dynamically with current reporting of event sensing flow of data
	4. EMOTIONS: BEFORE / AFTER ○ Before: Already existing of man-mac issu static boards raising challenges due to un upda realcurrent changes of road events ○ After: This system is better than exist method, of having automation of road signs & communication strategy in the manner of sma city to alert t reduce relay & congestion while travelling time	EM rivers to	

4. REQUIREMENT ANALYSIS 4.1 FUNCTIONAL REQUIREMENTS

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User visibility	• Informations can be written in short form in the sign boards so that it can be very easily captured by drivers. • Place sign boards on popular places. • Symbols can be used so that drivers can save some amount of time in reading. • Static signs can be replaced by smart signs to reduce accidents.
FR-2	User convenience	• Display should be larger which can be visible from far distance.
FR-3	User need	• Awareness programmes should be conducted to bring awareness among the users about road safety. • Road safety education is essential for users.

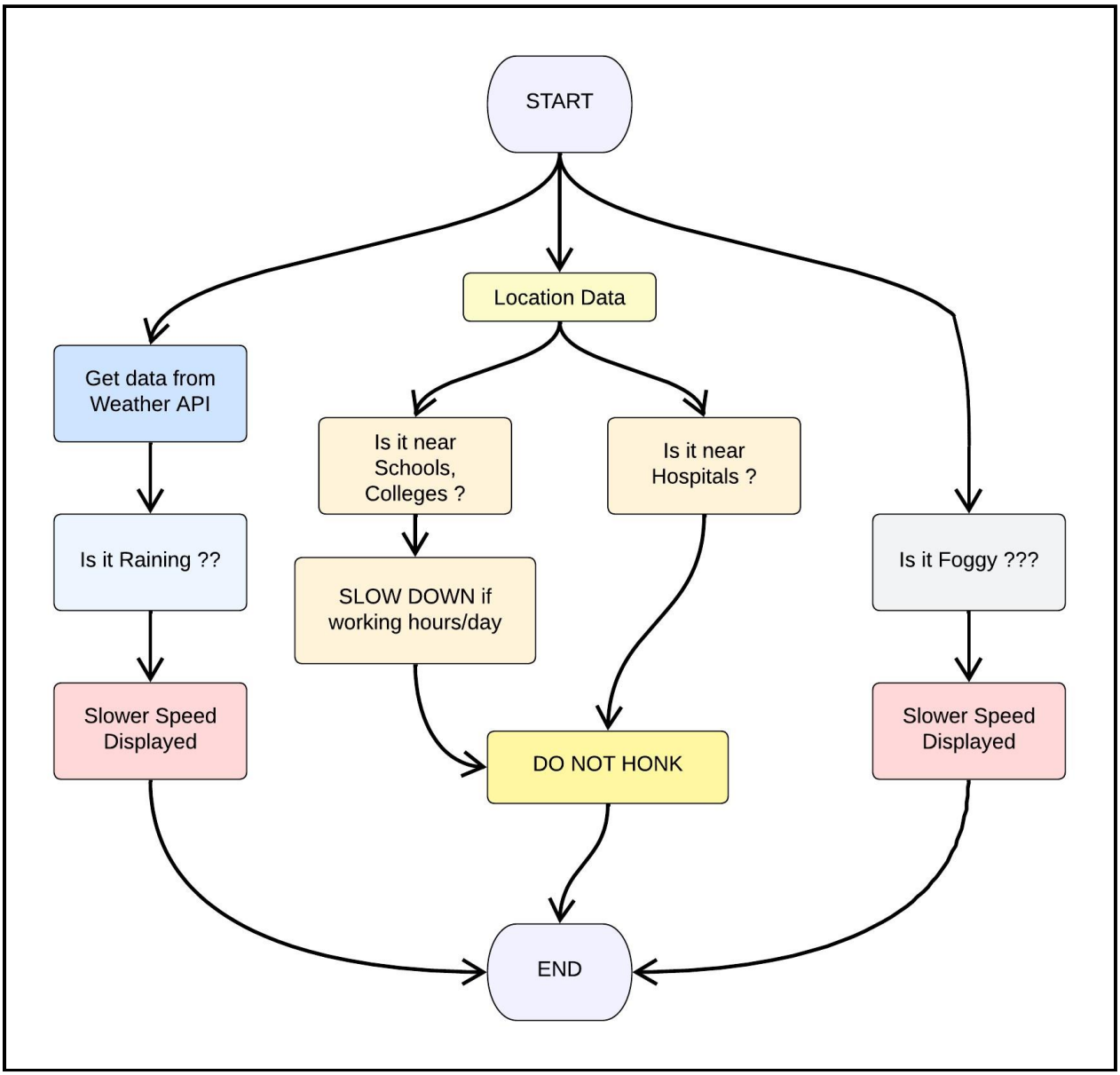
4.2 NON-FUNCTIONAL REQUIREMENTS

FR No.	Non-Functional Requirement	Description

NFR-1	**Usability**	• When crowd on accident occur it guides the travellers to choose best path. • Intimates the speed range depending upon roadway condition. • Ensure the vehicles are redirected to right path without causing much trouble for other drivers. • Easy to follow instructions based on given data on the digital board.
NFR-2	**Security**	• Prediction of data gives them a fair and better road understanding about their upcoming of toad events.
NFR-3	**Reliability**	• Helps to travellers behaviour towards awareness of travel.
NFR-4	**Performance**	• Pre-functional record of voice record along with LED display provide in waiting time at traffic signs. • There is a rain drop sensor which checks whether there is a rain, to transmit data over IOT helps to display on LED along with wifi connection of internet changing data dynamically with current reporting of event sensing flow of data
NFR-5	**Availability**	• Monitors the road events even in low light on poor weather conditions. • Record traffic offenses
NFR-6	**Scalability**	• It is user friendly interface. • Data accessibility is easy from source.

5. PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS



5.2 SOLUTION & TECHNICAL ARCHITECTURE

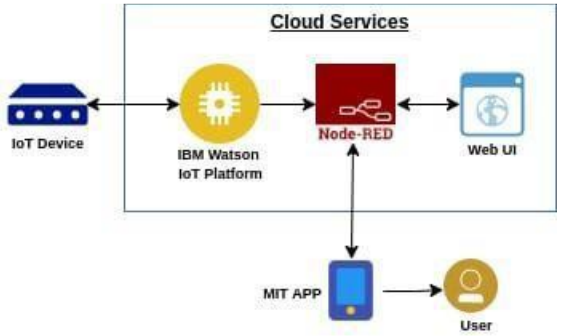
Solution architecture and Technical architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

• Find the best tech solution to solve existing business problems.

• Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.

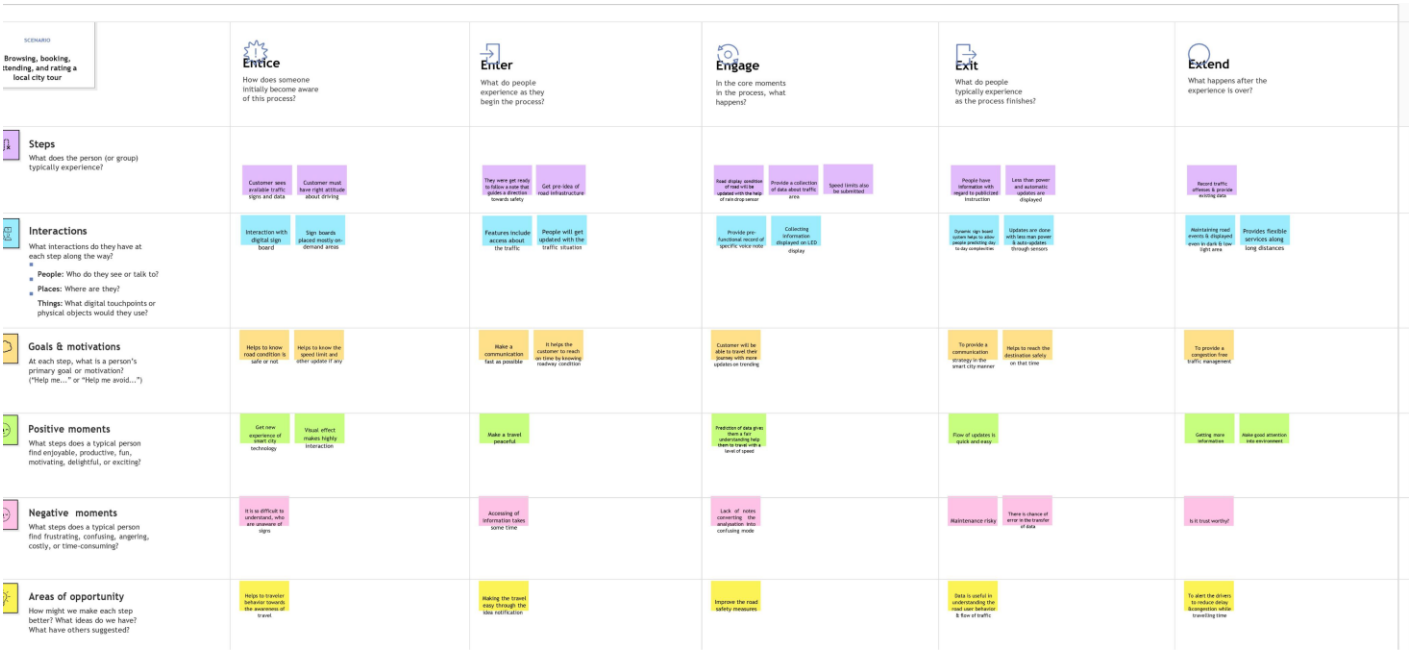
• Define features, development phases, and solution requirements.

• Provide specifications according to which the solution is defined, managed, and delivered.



SIGNS WITH SMART CONNECTIVITY FOR BETTER ROAD SAFETY

5.3 USER STORIES



6. PROJECT PLANNING AND SCHEDULING PHASE

6.1 SPRINT PLANNING & ESTIMATION

Sprint	Functional Requirement (Epic)	User Story/Task	Story Points	Priority
Sprint-1	Intializing the Resources	Create an account in Open Weather API	1	LOW
Sprint-1	Code in Software is written	Write a python script using the inputs given from OpenWeather API	2	MEDIUM
Sprint-2	Sending the software to cloud	The python code from sprint 1 should be sent to cloud so that it is easily accessible	1	MEDIUM

Sprint-3	Initialising the connection between hardware and cloud	The hardware should be intergrated for the easy access of the cloud functions	2	HIGH
Sprint-4	User input-output optimisation and error identification and rectification	Rectify all the shortcomings/errors and initiate the optimisation for better	3	HIGH

6.2 SPRINT DELIVERY SCHEDULE

TITLE	DESCRIPTION	STATUS
Literature Survey & Information gathering	**A literature review is a comprehensive summary of previous researches on the topic. The literature review surveys scholarly articles, books, and other sources relevant to a particular area of research.**	**Completed**
Prepare Empathy Map	**An empathy map is a collaborative tool teams can use to gain a deeper insight into their customers. It helps us to understand the customer's difficulties from their point of view.**	**Completed**

Ideation- Brainstorming	**Brainstorming is a group problem-solving method that helped us to gather and organize various ideas and thoughts from team members.**	**Completed**
Define Problem statement	**The problem statement helps us to focus on what matters to create experiences the people. This allowed us to find the ideal solution for the challenges.**	**Completed**
Problem Solution Fit	**It helped us understand and analyze all the thoughts of our customers, their choice of options, problems, behavior and emotions.**	**Completed**

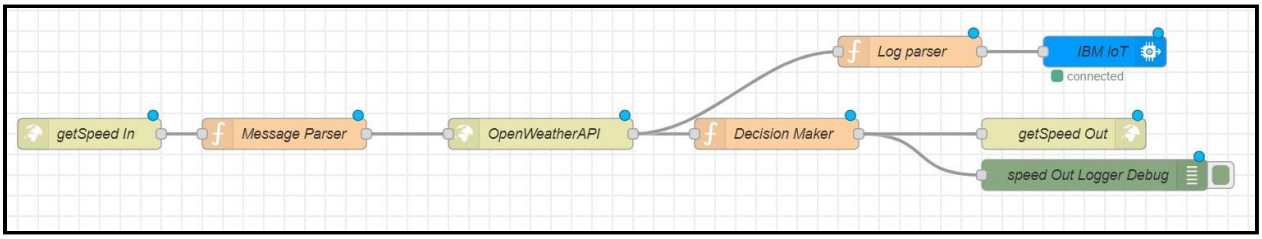
Proposed solution	**It helped us analyze and examine our solution more in the grounds of uniqueness, social impact, business model, scalability etc.**	**Completed**
Solution Architecture	**Solution architecture is a complex process with many sub-processes that bridges the gap between business problems and technology solutions. It helped us understand the features and components used to complete the project.**	**Completed**
Customer journey map	**It helped to analyze the various steps, interactions, goals and motivation, positives , negatives and opportunities.**	**Completed**

Solution requirements	**It briefs about functional and non-functional requirements. It involves the various steps in the entire process. It also specifies features usability, security, reliability, performance, availability and scalability.**	**Completed**
Technology stack	**A tech stack is the combination of technologies a company uses to build and run an application or project. It helps us analyze and understand various technologies that needs to be implemented in the project.**	**Completed**
Data Flow	**A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enter and leave the system, what changes their formation, and where data is stored.**	**Completed**
Sprint Delivery plan	**Sprint Planning is an event in scrum that defines what can be delivered in the upcoming sprint and how that work will be achieved. It helps us to organize and complete the work effectively and efficiently.**	**Completed**
Prepare milestone and activity list	**Helps us understand and evaluate our progress and accuracy so far.**	**Completed**

Project Development - Delivery of Sprint-1,2,3,4	**Develop and submit the developed code**	**Completed**

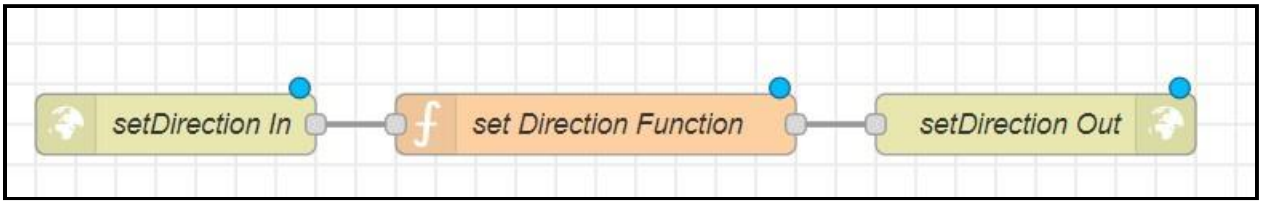
7. CODING & SOLUTIONING

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.

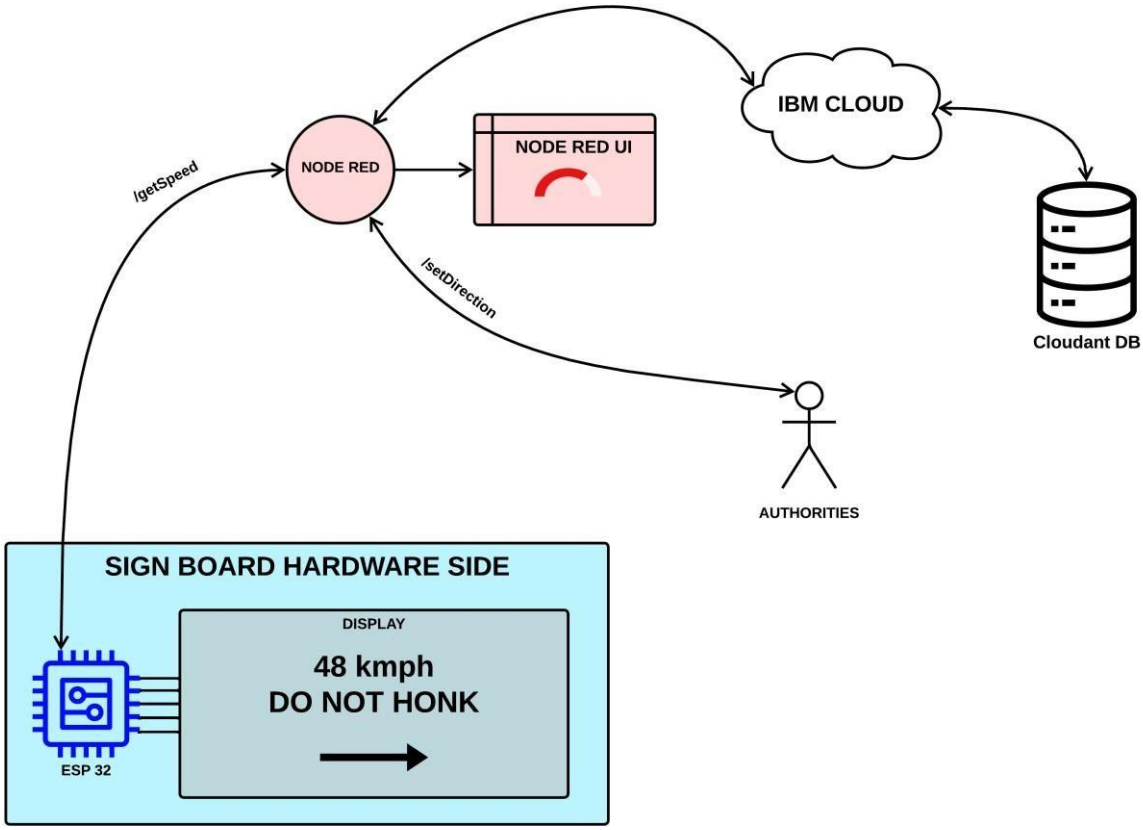
7.2 FEATURE 2 - SET DIRECTION REMOTELY FOR A GIVEN SIGN BOARD



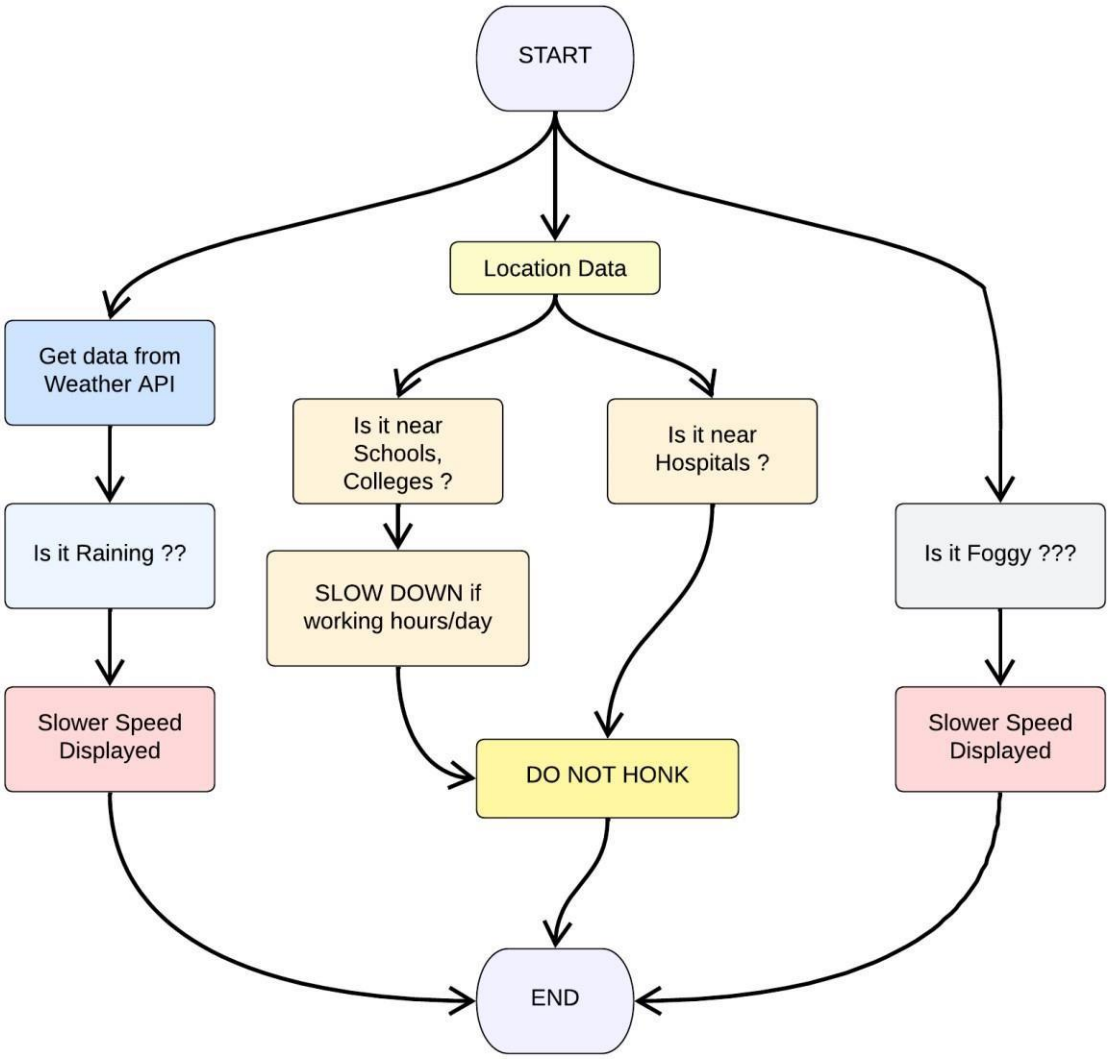
This part of Node RED flow accepts an http GET end point at **"/setDirection"** from which the uid and direction information are passed by the respective authorities. Set Direction Function block adds the direction information to the database and returns the same as an http response. This data is sent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

Flow :

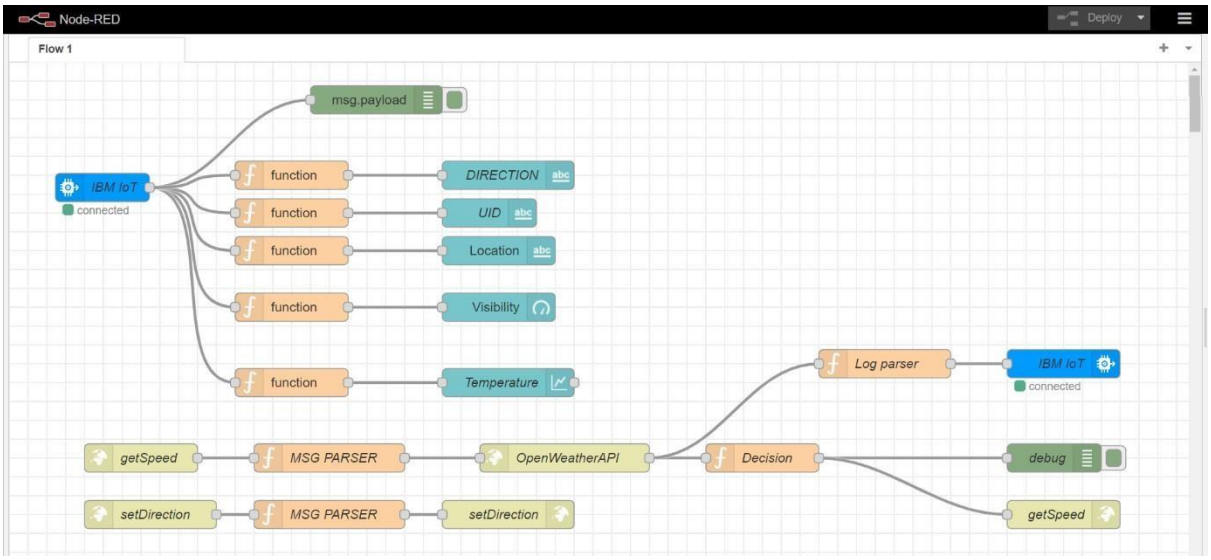

IBM CLOUD
NODE RED
NODE RED UI
/getSpeed
/setDirection
Cloudant DB
AUTHORITIES
SIGN BOARD HARDWARE SIDE
DISPLAY
48 kmph
DO NOT HONK
ESP 32

Code Flow :


START
Location Data
Get data from Weather API
Is it near Schools, Colleges ?
Is it near Hospitals ?
Is it Foggy ???
Is it Raining ??
SLOW DOWN if working hours/day
Slower Speed Displayed
DO NOT HONK
Slower Speed Displayed
END

Node RED :

Node RED flow :

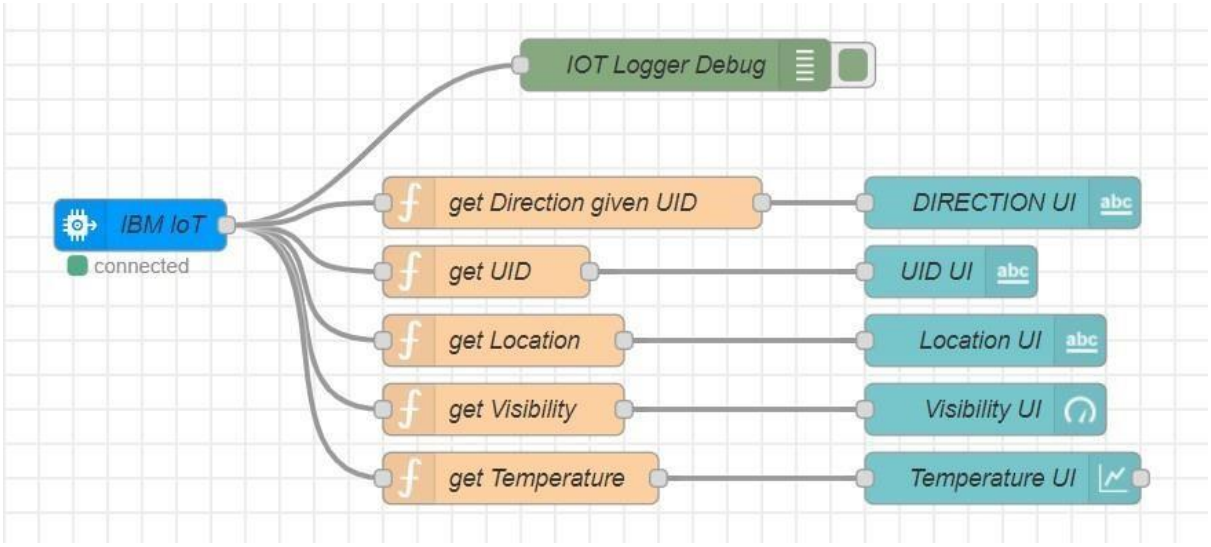


There are 3 flows in the above Node RED flow. They are

1.Node RED UI flow

2./getSpeed API flow 3. /setDirection API flow **1.**

Node RED UI flow :



1. **"IBM IOT"** node connects the backend to Node RED UI

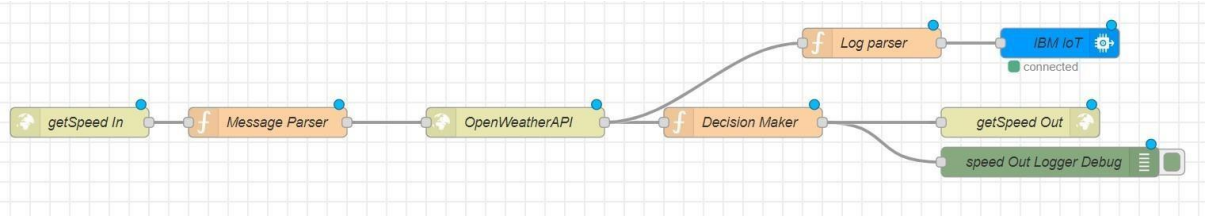
2. The function nodes such as **"get Direction given UID"**, **"get UID"**, **"get Location"**, **"get Visibility"** & **"get Temperature"** extract the respective data out and provides them to the UI nodes **"Direction UI"**, **"UID UI"**, **"Location UI"**, **"Visibility UI"** & **"Temperature UI"**.

```
// get Direction given UID

msg.payload = global.get(String(msg.payload.uid)); return

msg;

// get UID

msg.payload = msg.payload.uid; return

msg;

// get Location

msg.payload = msg.payload.location; return

msg;

// get Visibility

msg.payload = msg.payload.visibility; return

msg;

// get Temperature

msg.payload = msg.payload.temperature; return msg;
```

3. **"IOT Logger Debug"** node logs the data at debugger.

2.

/getSpeed API flow :



1. **"getSpeed In"** node is an http end point. It accepts parameters like microcontroller UID, location, school & hospital zones info.

2. **"Message Parser"** node parses the data and passes on only required information to the next node

```
global.set("data",msg.payload);

 msg.payload.q = msg.payload.location;

msg.payload.appid = "bf4a8d480ee05c00952bf65b78ae826b"; return

msg;
```

3. **"OpenWeatherAPI"** node is a http request node which calls the OpenWeather API and send the data to the next node.

4. **"Log Parser"** node extracts specific parameters from the weather data and and sends it to the next node.

```
weatherObj = JSON.parse(JSON.stringify(msg.payload)); localityObj

= global.get("data");

 var suggestedSpeedPercentage = 100;

 var preciseObject = {

temperature :

weatherObj.main.temp

- 273.15, location :

localityObj.location,

visibility
```

```
: weatherObj.visibility/100,     uid : localityObj.uid,

   direction : global.get("direction")

};

msg.payload = preciseObject;   return

msg;
```

5. **"IBM IoT"** node here (IBM IoT OUT)connects the **"IBM IoT"** node (IBM IoT IN) metioned in the **Node RED UI flow** which enables UI updation and logging.

6. **"Decision Maker"** node processes the weather data and other information from the micro controller to form the string that is to be displayed at the Sign Board

```
weatherObj = JSON.parse(JSON.stringify(msg.payload)); localityObj

= global.get("data");

 var suggestedSpeedPercentage = 100;

 var preciseObject = {

   temperature : weatherObj.main.temp - 273.15,

   weather : weatherObj.weather.map(x=>x.id).filter(code => code<700),

visibility : weatherObj.visibility/100

};              if(preciseObject.visibility<=40)
suggestedSpeedPercentage -=30

    switch(String(preciseObject.weather)[-1]) //

https://openweathermap.org/weatherconditions refer weather codes meaning here

{

   case "0" : suggestedSpeedPercentage -=10;break;
```

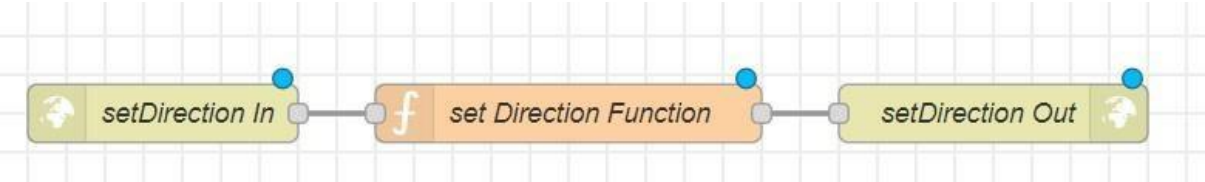
```
case "1" : suggestedSpeedPercentage -=20;break;     case
"2" : suggestedSpeedPercentage -=30;break;
}   msg.payload = preciseObject;
 var doNotHonk = 0;
if(localityObj.hospitalZone=="1"||localityObj.schoolZone=="1")
doNotHonk = 1;
 var returnObject = {
  suggestedSpeed : localityObj.usualSpeedLimit*(suggestedSpeedPercentage/100),
doNotHonk : doNotHonk
}    msg.payload = String(returnObject.suggestedSpeed) + " kmph \n\n" +
(returnObject.doNotHonk==1?"Do Not Honk":"") + "$" + global.get(String(localityObj.uid));
 return msg;
```

7. **"getSpeed Out"** node returns a http response for the request at node **"getSpeed In"**.

8. **"speed Out Logger Debug"** logs the data for debugging.

3.

/setDirection API flow :



1. **"setDirection In"** node is an http end point. It accepts parameters like microcontroller UID & direction.

2. **"set Direction Function"** node sets the direction for the given UID.

```
global.set(String(msg.payload.uid),msg.payload.dir); return

msg;
```

3. **"setDirection Out"** node returns a http response for the request at node **"setDirection In"**.

Click on this link to change direction to Straight

Click on this link to change direction to Left

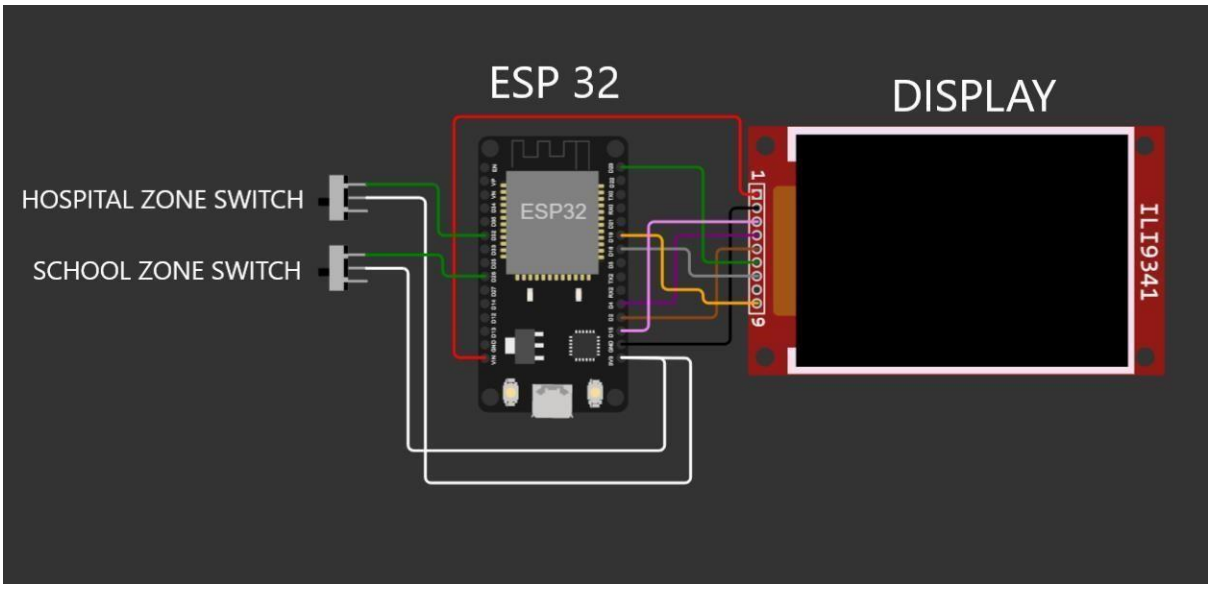
Click on this link to change direction to Right

Wokwi Circuit :

Wokwi Code

Wokwi Link

Circuit Diagram :



ESP 32 CODE :

```
#include <WiFi.h>

#include <HTTPClient.h>

#include <Adafruit_GFX.h>

#include <Adafruit_ILI9341.h>

#include   <string.h>        const
char* ssid = "Wokwi-

GUEST"; const char* password =

"";


#define TFT_DC 2

#define TFT_CS 15

Adafruit_ILI9341 tft = Adafruit_ILI9341(TFT_CS, TFT_DC);

String   myLocation   =   "Chennai,IN";     String
usualSpeedLimit   =   "70";   //   kmph        int
schoolZone = 32;

int hospitalZone = 26;

 int uid = 2504; // ID Unique to this Micro Contoller

String getString(char x)

{

   String s(1, x);     return s;

}

String stringSplitter1(String fullString,char delimiter='$')

{
```

```
   String returnString = "";              for(int i = 0;
i<fullString.length();i++) {
char c = fullString[i];        if(delimiter==c)          break;
       returnString+=String(c);
   }
   return(returnString);
}
String stringSplitter2(String fullString,char delimiter='$')
{
   String returnString = "";    bool flag = false;
   for(int i = 0; i<fullString.length();i++) {
char c = fullString[i];        if(flag)
          returnString+=String(c);
if(delimiter==c)          flag = true;     }
   return(returnString);
}
void rightArrow()
{
  int refX = 50;
  int refY = tft.getCursorY() + 40;
   tft.fillRect(refX,refY,100,20,ILI9341_RED);
tft.fillTriangle(refX+100,refY-
30,refX+100,refY+50,refX+40+100,refY+10,ILI9341_RED);
```

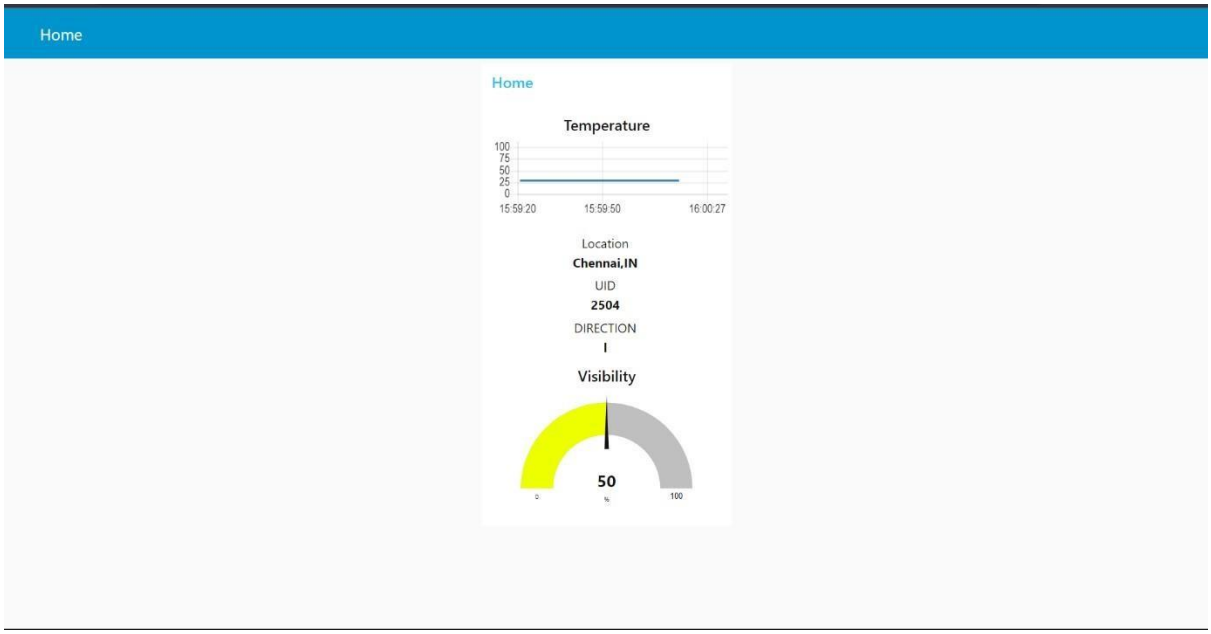
```
}
void leftArrow()
{
  int refX = 50;
  int refY = tft.getCursorY() + 40;
    tft.fillRect(refX+40,refY,100,20,ILI9341_RED);
  tft.fillTriangle(refX+40,refY-30,refX+40,refY+50,refX,refY+10,ILI9341_RED);
}                 void
upArrow()
{
  int refX = 125;
  int refY = tft.getCursorY() + 30;
    tft.fillTriangle(refX-40,refY+40,refX+40,refY+40,refX,refY,ILI9341_RED);
tft.fillRect(refX-15,refY+40,30,20,ILI9341_RED);
}
String APICall() {
  HTTPClient http;
  String url = "https://node-red-grseb-2022-11-05test.eugb.mybluemix.net/getSpeed?";   url +=
"location="+myLocation+"&";                                    url          +=
"schoolZone="+(String)digitalRead(schoolZone)+(String)"&";              url     +=
"hospitalZone="+(String)digitalRead(hospitalZone)+(String)"&";           url     +=
"usualSpeedLimit="+(String)usualSpeedLimit+(String)"&";                 url     +=
```

```
"uid="+(String)uid;  http.begin(url.c_str());  int httpResponseCode = http.GET();
if (httpResponseCode>0) {
  String payload = http.getString();
http.end();  return(payload);
 } else {
  Serial.print("Error code: ");
  Serial.println(httpResponseCode);
 } http.end();
} void myPrint(String contents)
{
tft.fillScreen(ILI9341_BLACK);  tft.setCursor(0,
20);  tft.setTextSize(4);
 tft.setTextColor(ILI9341_RED);
 //tft.println(contents);
  tft.println(stringSplitter1(contents));  String c2
=  stringSplitter2(contents);  if(c2=="s")  //
represents Straight
 {
  upArrow();
 }
 if(c2=="l") // represents left
 {
  leftArrow();
```

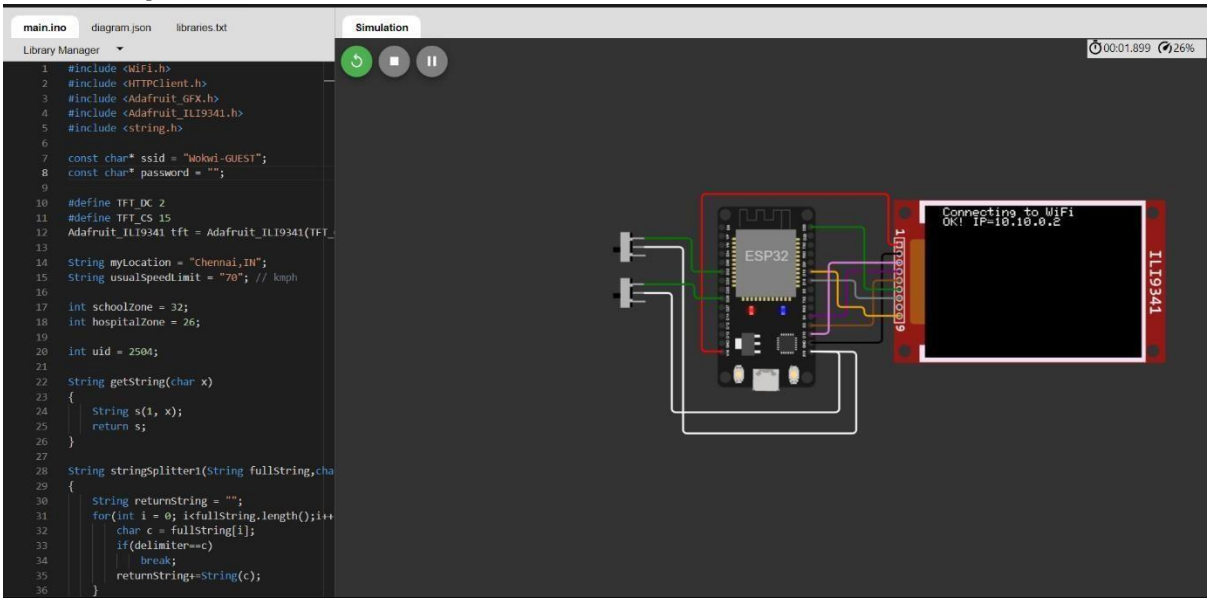
```
 }
 if(c2=="r") // represents right
 {
   rightArrow();
 }  }
void setup() {
 WiFi.begin(ssid, password, 6);    tft.begin();
tft.setRotation(1);
  tft.setTextColor(ILI9341_WHITE);
tft.setTextSize(2);
 tft.print("Connecting to WiFi");
   while (WiFi.status() != WL_CONNECTED) {
delay(100);    tft.print(".");
 }
 tft.print("\nOK! IP=");
tft.println(WiFi.localIP());
}       void    loop()    {
myPrint(APICall());
delay(100);  }
```

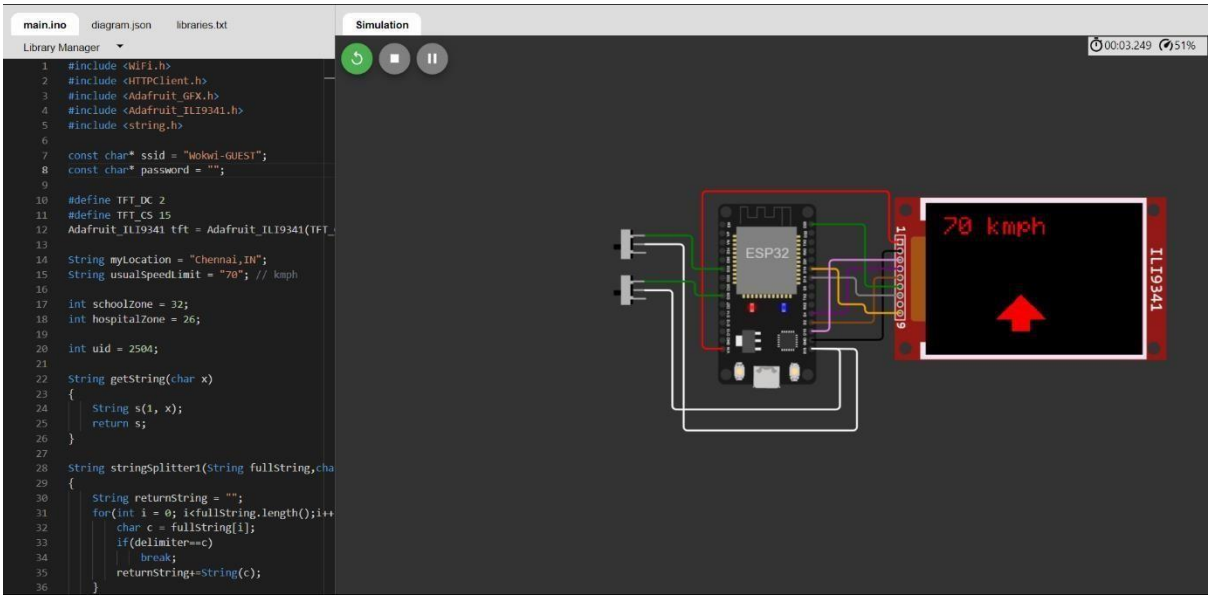
Output :

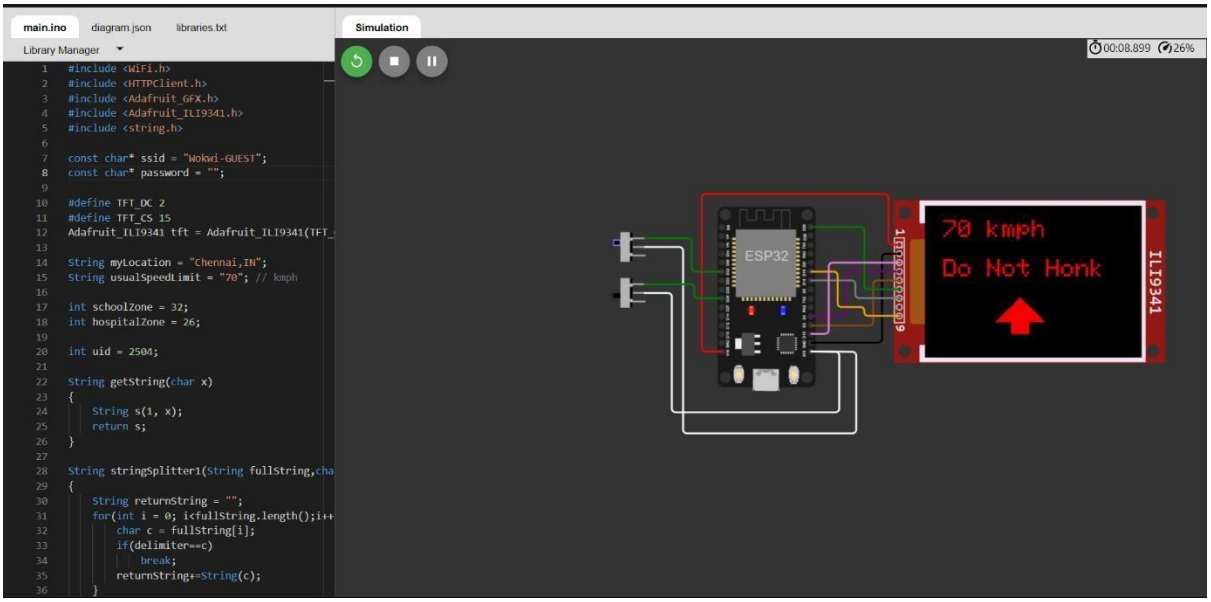
Node RED Dashboard :

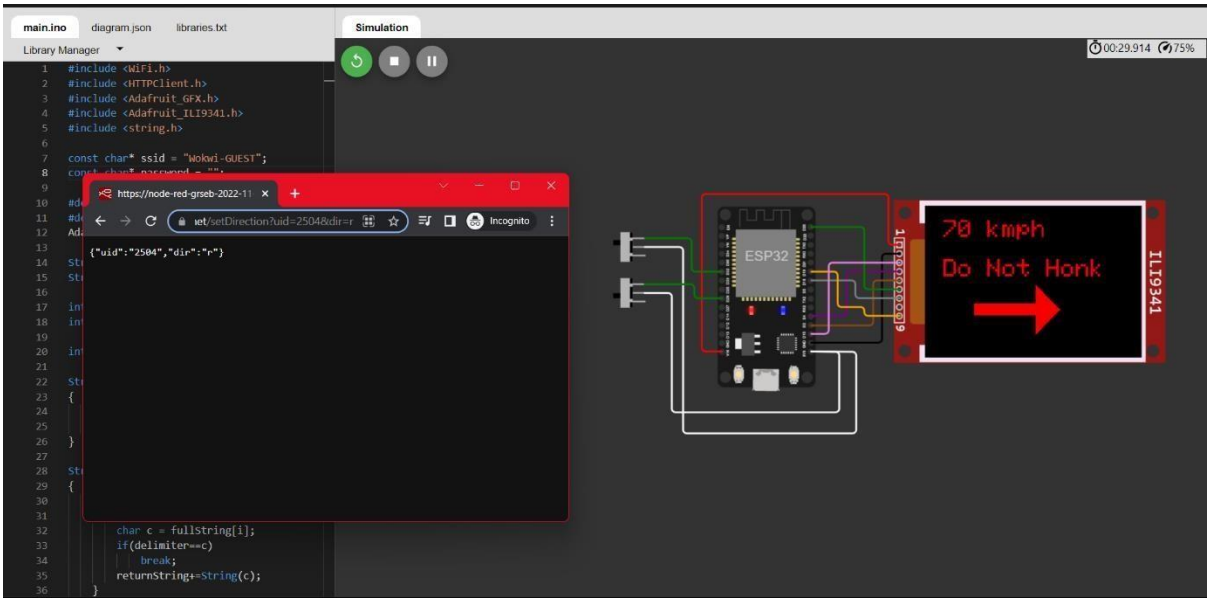


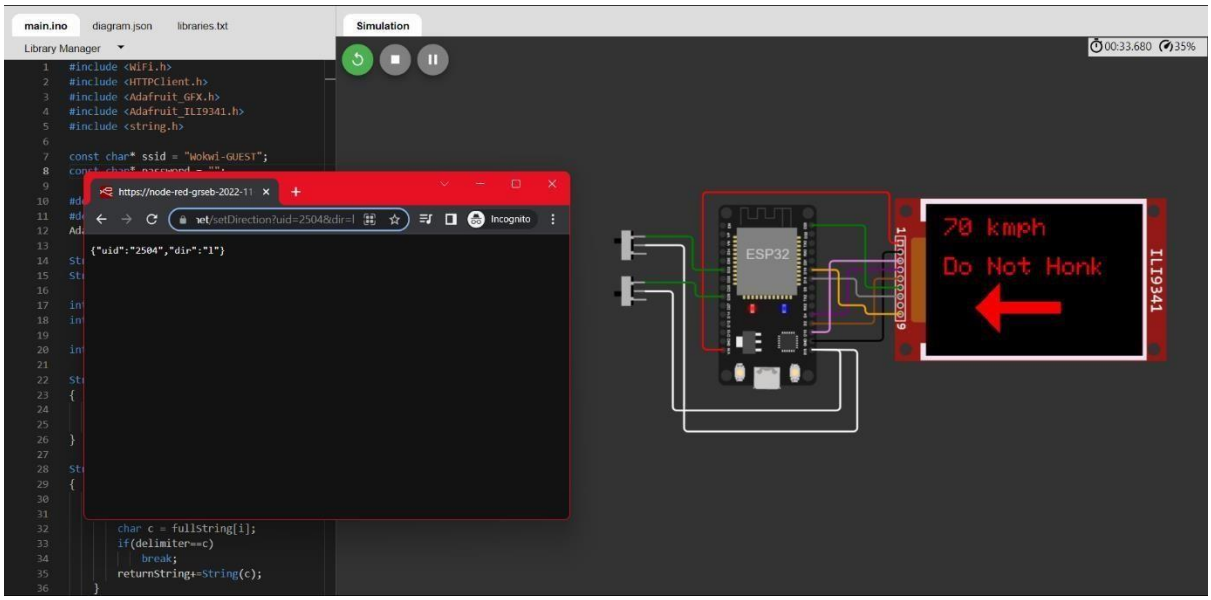
Wokwi Output :

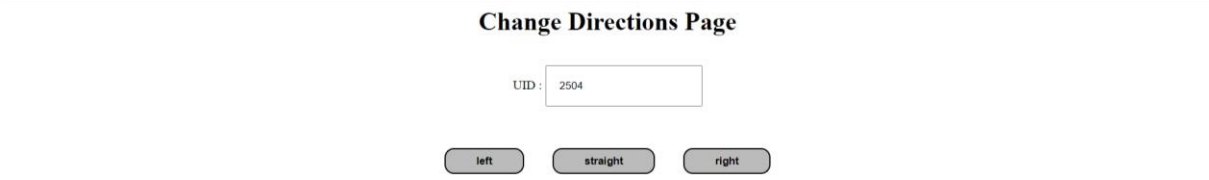



main.ino
diagram.json
libraries.txt
Library Manager
Simulation
00:03.249
51%
#include <WiFi.h>
#include <HTTPClient.h>
#include <Adafruit_GFX.h>
#include <Adafruit_ILI9341.h>
#include <string.h>
const char* ssid = "Wokwi-GUEST";
const char* password = "";
#define TFT_DC 2
#define TFT_CS 15
Adafruit_ILI9341 tft = Adafruit_ILI9341(TFT_
String myLocation = "Chennai,IN";
String usualSpeedLimit = "70"; // kmph
int schoolZone = 32;
int hospitalZone = 26;
int uid = 2504;
String getString(char x)
{
String s(1, x);
return s;
}
String stringSplitter1(String fullString,cha
{
String returnString = "";
for(int i = 0; i<fullString.length();i++
char c = fullString[i];
if(delimiter==c)
break;
returnString+=String(c);
}
ESP32
70 kmph
ILI9341


main.ino
diagram.json
libraries.txt
Library Manager
Simulation
00:08.899
26%
#include <WiFi.h>
#include <HTTPClient.h>
#include <Adafruit_GFX.h>
#include <Adafruit_ILI9341.h>
#include <string.h>
const char* ssid = "Wokwi-GUEST";
const char* password = "";
#define TFT_DC 2
#define TFT_CS 15
Adafruit_ILI9341 tft = Adafruit_ILI9341(TFT_
String myLocation = "Chennai,IN";
String usualSpeedLimit = "70"; // kmph
int schoolZone = 32;
int hospitalZone = 26;
int uid = 2504;
String getString(char x)
{
String s(1, x);
return s;
}
String stringSplitter1(String fullString,cha
{
String returnString = "";
for(int i = 0; i<fullString.length();i++
char c = fullString[i];
if(delimiter==c)
break;
returnString+=String(c);
}
ESP32
70 kmph
Do Not Honk
ILI9341


main.ino
diagram.json
libraries.txt
Library Manager
Simulation
00:29.914
75%
https://node-red-grseb-2022-11
Incognito
{"uid":"2504","dir":"r"}
ESP32
70 kmph
Do Not Honk
ILI9341


main.ino
diagram.json
libraries.txt
Library Manager
Simulation
00:33.680
35%
https://node-red-grseb-2022-11
Incognito
{"uid":"2504","dir":"l"}
ESP32
70 kmph
Do Not Honk
ILI9341


Change Directions Page
UID :
2504
left
straight
right

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-28140-1660107294

- **DEMO VIDEO DOWNLOAD LINK**

https://drive.google.com/drive/folders/1qKDtdxzpOgDeDKG2L_mCRqmSEIGjDXmY

- **SOURCE CODE - ESP 32**

```
#include <WiFi.h>
#include <HTTPClient.h> 3 #include <Adafruit_GFX.h> 4 #include <Adafruit_ILI9341.h> 5 #include <string.h>

7 const char* ssid = "Wokwi-GUEST";
8 const char* password = "";

10#define TFT_DC 2
11#define TFT_CS 15
12Adafruit_ILI9341 tft = Adafruit_ILI9341(TFT_CS, TFT_DC);

14String myLocation = "Chennai,IN";
```

```
String                    "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 = "";
```

```
bool        false
for int     0
```

```
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);
69                                                  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       +=
    "usualSpeedLimit="+(String)usualSpeedLimit+(String)"&";
89                                                  url += "uid="+(String)uid;
90                                                  http.begin(url.c_str());
91                                                  int httpResponseCode = http.GET();
92
```

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   }
```

- **GITHUB AND PROJECT DEMO LINK**

https://github.com/IBM-EPBL/IBM-Project-28140-1660107294

- **DEMO VIDEO DOWNLOAD LINK**

https://drive.google.com/drive/folders/1qKDtdxzpOgDeDKG2L_mCRqmSEIGjDXmY