IBM NAALAIYA THIRAN PROJECT REPORT TEAM ID - PNT2022TMID42985

IOT BASED SIGNS WITH SMART CONNECTIVITY FOR BETTER ROAD SAFETY

A PROJECT REPORT

Submitted by

ARULKUMAR.R	**(712819106709)**
SOUNDERRAJAN.R	**(712819106707)**
AKSHAY.K	**(712819106708)**
PRAVEEN.M	**(712819106302)**
VASANTHARAJ.V	**(712819106303)**

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

RVS COLLEGE OF ENGINEERING AND TECHNOLOGY, COIMBATORE

ANNA UNIVERSITY:CHENNAI 600 025

NOVEMBER 2022

-

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

Wireless Digital Traffic Signs of the Future

Authors :

CHAI K.TOH PIETRO
MANZONI
CARLOS TAVARES CALAFATE

Traffic signs have come a long way since the first automobile was invented. They have long served the purpose of warning and guiding drivers and also enforcing the traffic laws governing speed, parking, turns, and stopping. In this study, the authors discuss the issues and challenges facing current traffic signs, and how it will evolve into a next-generation traffic sign architecture using advanced wireless communications technologies. With technological advances in the areas of wireless communications and embedded electronics and software, we foresee that, in the future, digital traffic sign posts will be capable of transmitting the traffic sign information wirelessly to road users, and this will transform our roads into intelligent roads, where signs will appear promptly and automatically on in-vehicle displays to alert the driver. There is no longer the need to watch out for traffic signs since the detection will be automatic and performed wirelessly. This transformation will lessen burden on the drivers, so that they can then focus more on the traffic ahead while driving.Also, this evolution into wireless digital sign posts will fit well with the vision of future smart cities, where smart transportation technologies will be present to transform how we drive and commute, yielding greater safety, ease, and assistance to drivers.

Application of machine learning methods for traffic signs recognition

Authors :

D V FILATOV
A V DEVIATKIN SERYKH ELENA

This paper focuses on solving a relevant and pressing safety issue on intercity roads. Two approaches were considered for solving the problem of traffic signs recognition; the approaches involved neural networks to analyze images obtained from a camera in the real-time mode. The first approach is based on a sequential image processing. At the initial stage, with the help of color filters and morphological operations (dilatation and erosion), the area containing the traffic sign is located on the image, then the selected and scaled fragment of the image is analyzed using a feed forward neural network to determine the meaning of the found traffic sign. Learning of the neural network in this approach is carried out using a back propagation method. The second approach involves convolution neural networks at both stages, i.e. when searching and selecting the area of the image containing the traffic sign, and when determining its meaning. Learning of the neural network in the second approach is carried out using the intersection over union function and a loss function. For neural networks to learn and the proposed algorithms to be tested, a series of videos from a dash cam were used that were shot under various weather and illumination conditions. As a result, the proposed approaches for traffic signs recognition were analyzed and compared by key indicators such as recognition rate percentage and the complexity of neural networks' learning process.

Integration of Image-Based Fog Detection with Autonomous Decision System for Intelligent Road Sign

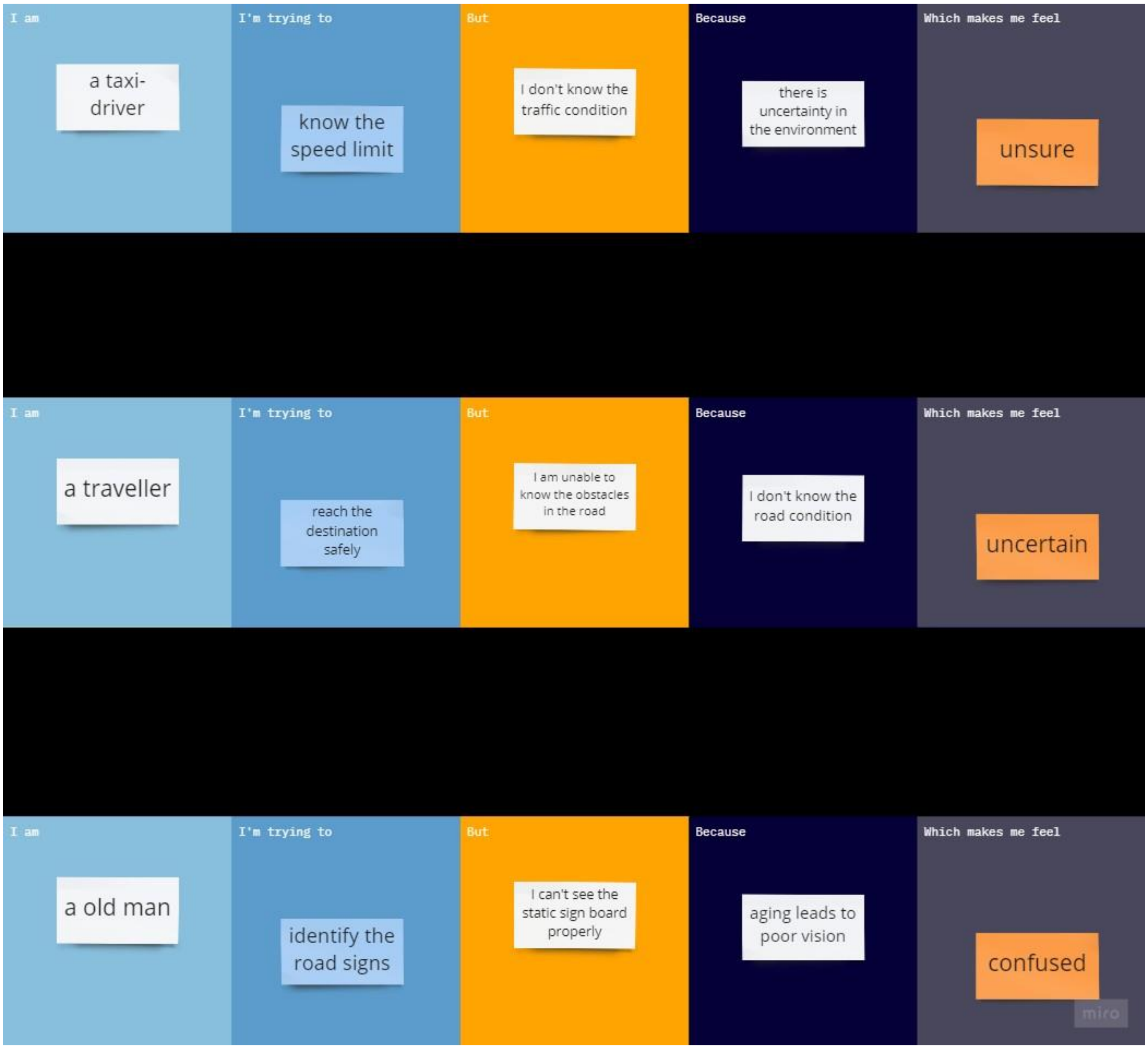
Authors :

WOJCIECH CHMIEL JAN DERKACZ
JANUSZ GOZDECKI ANDRZEJ DZIECH

The paper presents the description of the decision system implemented for Intelligent Road Signs. It focuses on the implementation of the novel air transparency analysis system and its integration with the rule system and the speed control infrastructure. Moreover, there are presented issues of making decisions about the content displayed in the case of autonomous and cooperating signs. To reflect more closely on real-life situations, it is assumed that the content presented by the IRS changes dynamically, depending on the road traffic and weather parameters. The IRS system operation was presented using fog detection as an example.

2.3 PROBLEM STATEMENT DEFINITION

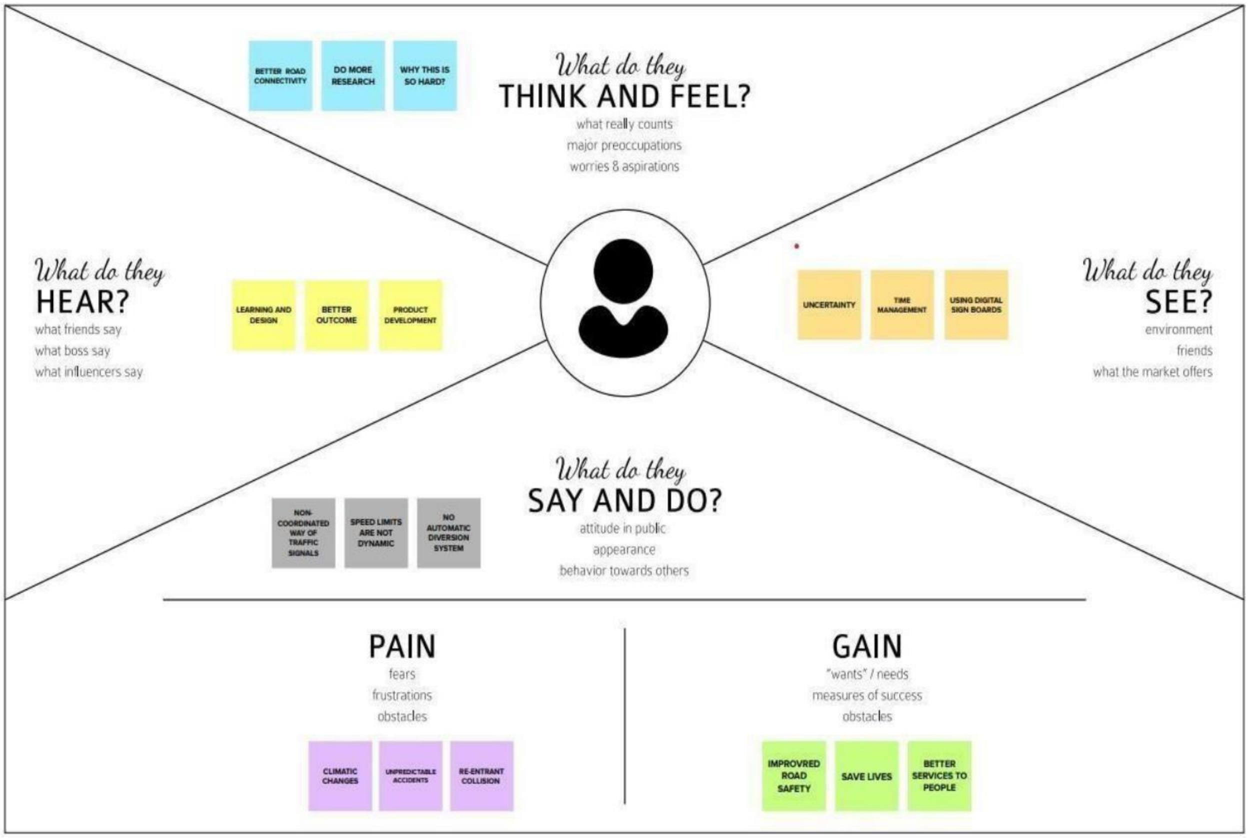
To replace the static signboards with smart connected sign boards to get the speed limitations from a web app using weather API and update it automatically based on the weather conditions, set diversions through API and warn drivers for school zones and hospital zones.



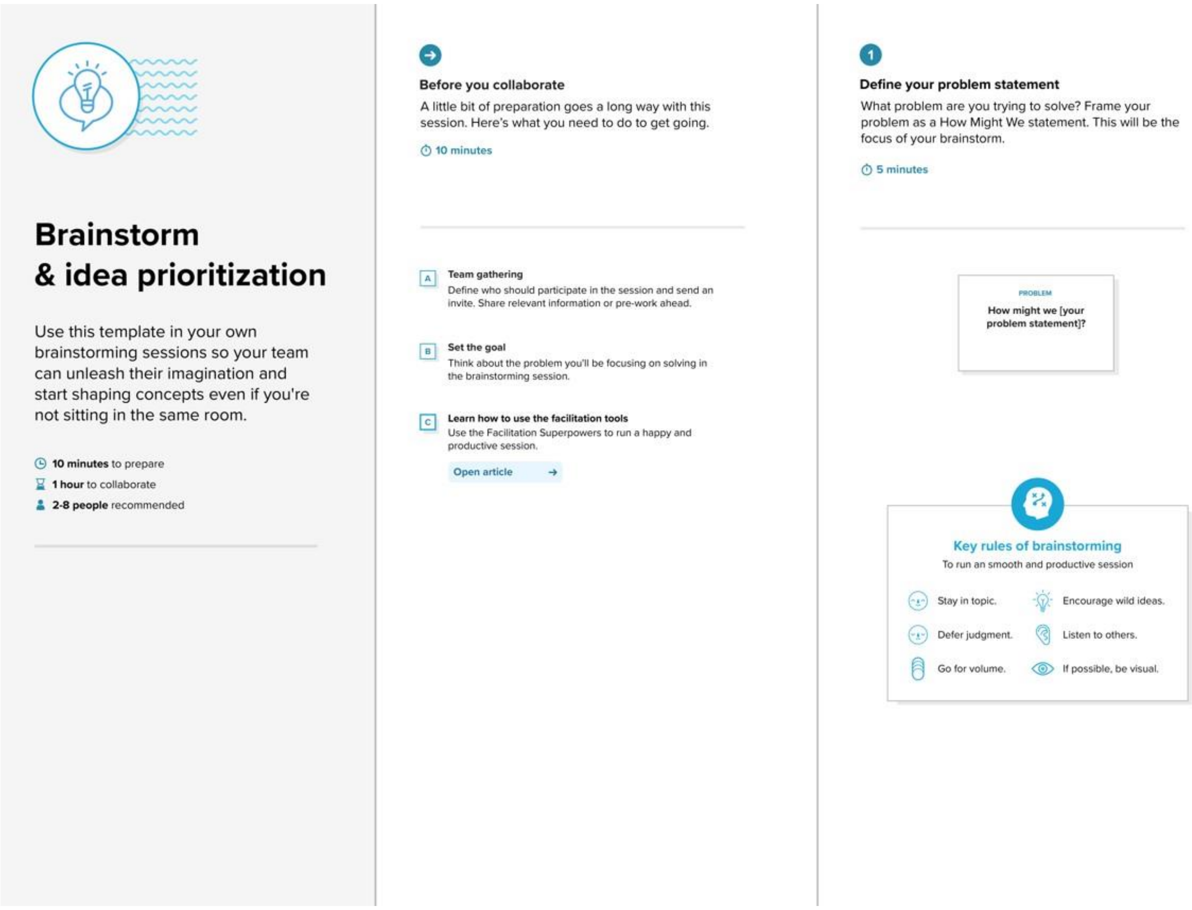
I am	I'm trying to	But	Because	Which makes me feel
a taxi-driver	know the speed limit	I don't know the traffic condition	there is uncertainty in the environment	unsure
a traveller	reach the destination safely	I am unable to know the obstacles in the road	I don't know the road condition	uncertain
a old man	identify the road signs	I can't see the static sign board properly	aging leads to poor vision	confused

3. IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



3.2 IDEATION & BRAINSTORMING



Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

- **10 minutes** to prepare
- **1 hour** to collaborate
- **2-8 people** recommended

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

A **Team gathering**
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B **Set the goal**
Think about the problem you'll be focusing on solving in the brainstorming session.

C **Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.

Open article →

1 **Define your problem statement**

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

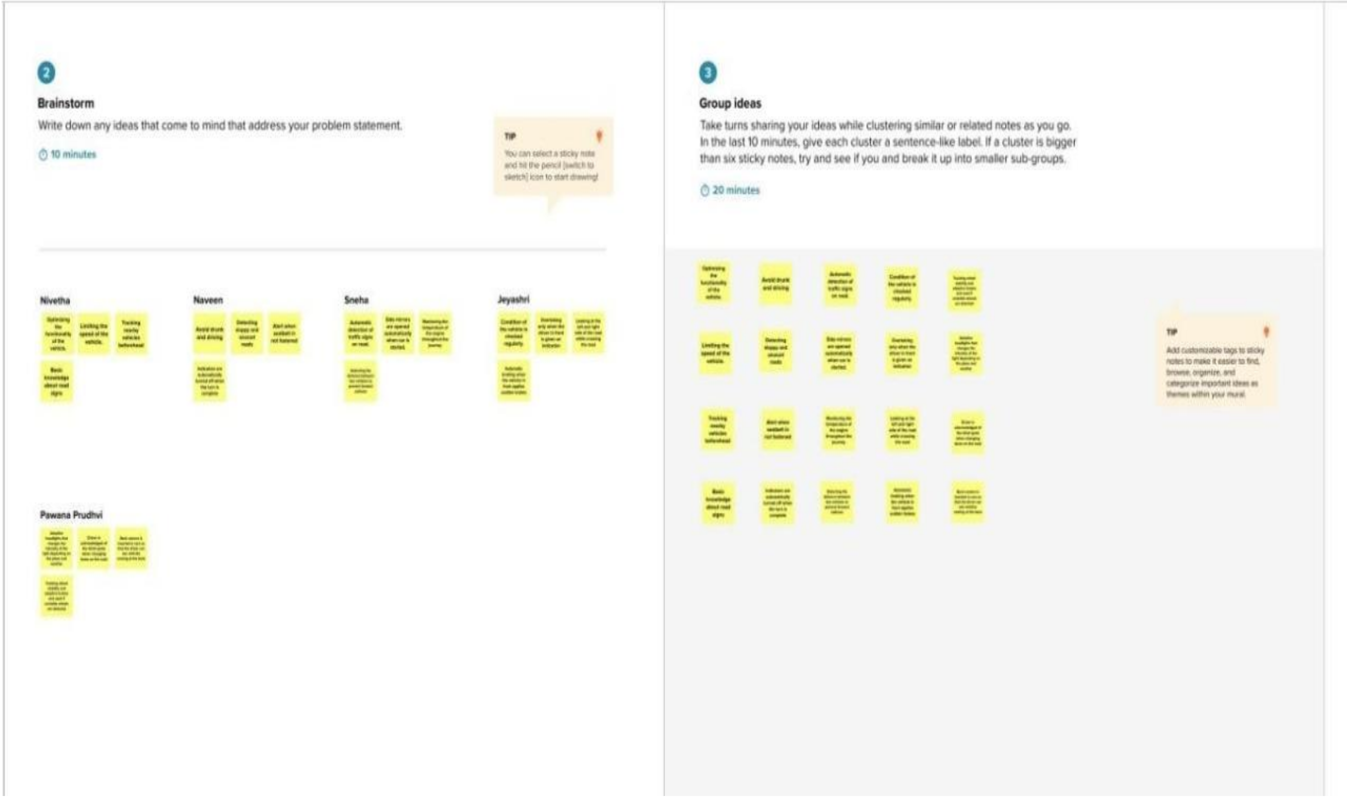
5 minutes

PROBLEM
How might we [your problem statement]?

Key rules of brainstorming
To run an smooth and productive session

- Stay in topic.
- Encourage wild ideas.
- Defer judgment.
- Listen to others.
- Go for volume.
- If possible, be visual.

Brainstorm, Idea Listing and Grouping



2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

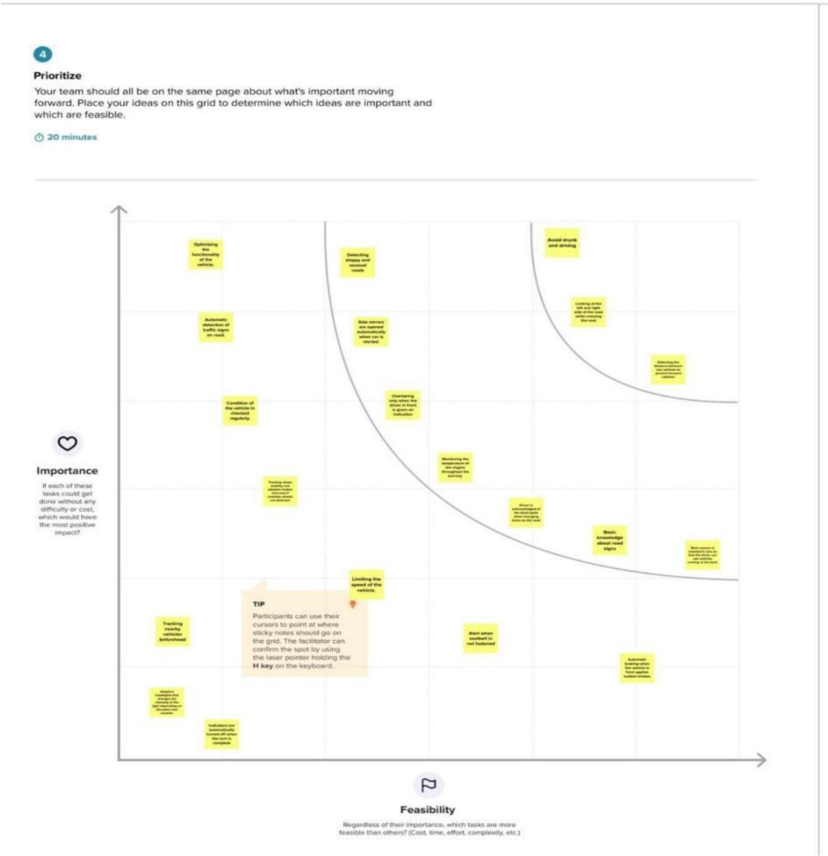
TIP

Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

Nivetha
Optimizing the functionality of the vehicle.
Limiting the speed of the vehicle.
Tracking nearby vehicles beforehand
Basic knowledge about road signs
Naveen
Avoid drunk and driving
Detecting slippy and unsound roads
Alert when seatbelt is not fastened
Sneha
Automatic detection of traffic signs on road.
Side mirrors are opened automatically when car is started.
Jeyashri
Condition of the vehicle is checked regularly
Pawana Prudhvi

SIGNS WITH SMART CONNECTIVITY FOR BETTER ROAD SAFETY

Idea Prioritization



4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

TIP

Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the **H key** on the keyboard.

Importance

If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

Feasibility

Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

Optimizing the functionality of the vehicle.
Detecting slippy and unusual roads
Avoid drunk and driving
Automatic detection of traffic signs on road.
Side mirrors are opened automatically when car is started.
Looking at the left and right side of the road while crossing the road
Condition of the vehicle is checked regularly.
Overtaking only when the driver in front is given an indication
Monitoring the temperature of the engine throughout the journey
Basic knowledge about road signs
Limiting the speed of the vehicle.
Tracking nearby vehicles beforehead
Alert when seatbelt in not fastened
Automatic braking when the vehicle in front applies sudden brakes
Indicators are automatically turned off when the turn is complete

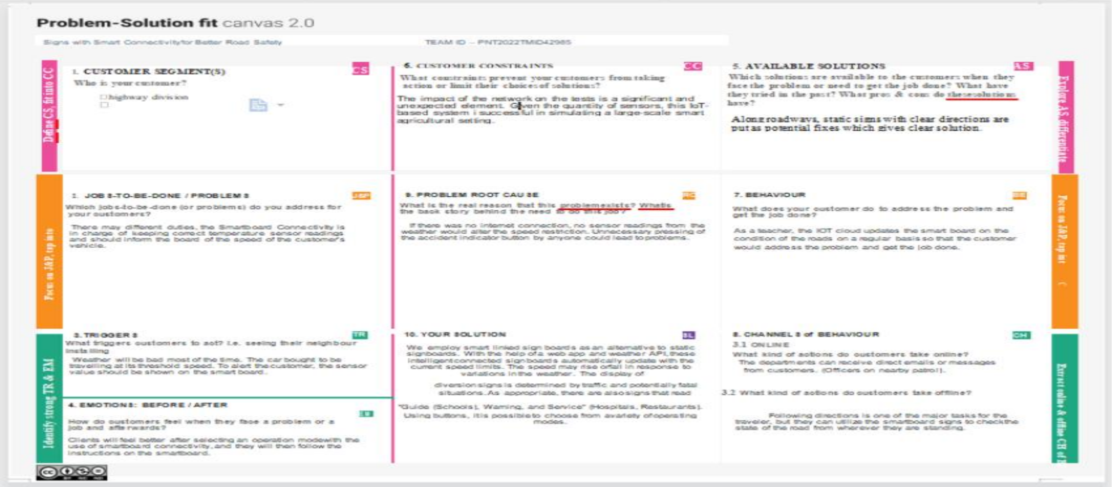
3.3 PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	• To prevent the road accidentsfrom happening using IOT.
2.	Idea / Solution description	• By Preparing smart signs using IOT instead of regular signs hung on the road. • Smart signs are built with IOT and LEDare used.

3.	Novelty / Uniqueness	• Since LED'S are used which is visiblefrom after. • The smart signs consists oftemperature, humidity, wind speed. • These information are received from weather monitoring app. • It also gives information about nearby places such as hospitals,schools,etc….,so that the users can decide their speeding according to that information.
4.	Social Impact / Customer Satisfaction	• These create a noticable impact onthe road safety department. • By deciding a speed limit for theuser,there is significant chance in reducing the accidents.
5.	Business Model (Revenue Model)	• By executing these for commoners by the government, it is great initiative increating a awareness among the people. • A separate budget can be alloted for this by the government, which paves a way for a safer environment.
6.	Scalability of the Solution	• It has greater chance in reducing the risk for the people as it is more visible thanthe normal signs, which saves a lot of lives at stake.

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.



Problem-Solution fit canvas 2.0

Signs with Smart Connectivityfor Better Road Safety

TEAM ID – PNT2022TMID42985

1. CUSTOMER SEGMENT(S)

Who is your customer?

□highway division

□

6. CUSTOMER CONSTRAINTS

What constraints prevent your customers from taking action or limit their choices of solutions?

The impact of the network on the tests is a significant and unexpected element. Given the quantity of sensors, this IoT-based system i successful in simulating a large-scale smart agricultural setting.

5. AVAILABLE SOLUTIONS

Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do thesesolutions have?

Alongroadways, static signs with clear directions are putas potential fixes which gives clear solution.

1. JOBS-TO-BE-DONE / PROBLEMS

Which jobs-to-be-done (or problems) do you address for your customers?

There may different duties, the Smartboard Connectivity is in charge of keeping correct temperature sensor readings and should inform the board of the speed of the customer's vehicle.

9. PROBLEM ROOT CAUSE

What is the real reason that this problemexists? Whatis the back story behind the need to do this job?

If there was no internet connection, no sensor readings from the weather would alter the speed restriction. Unnecessary pressing of the accident indicator button by anyone could lead to problems.

7. BEHAVIOUR

What does your customer do to address the problem and get the job done?

As a teacher, the IOT cloud updates the smart board on the condition of the roads on a regular basis so that the customer would address the problem and get the job done.

3. TRIGGERS

What triggers customers to act? i.e. seeing their neighbour installing

Weather will be bad most of the time. The car bought to be travelling at its threshold speed. To alert the customer, the sensor value should be shown on the smart board.

4. EMOTIONS: BEFORE / AFTER

How do customers feel when they face a problem or a job and afterwards?

Clients will feel better after selecting an operation modewith the use of smartboard connectivity, and they will then follow the instructions on the smartboard.

10. YOUR SOLUTION

We employ smart linked sign boards as an alternative to static signboards. With the help of a web app and weather API,these intelligentconnected signboards automatically update with the current speed limits. The speed may rise orfall in response to variations inthe weather. The display of

diversionsigns is determined by traffic and potentially fatal situations. As appropriate, there are alsosigns that read

"Guide (Schools), Warning, and Service" (Hospitals, Restaurants). Using buttons, itis possibleto choose from avariety ofoperating modes.

8. CHANNELS of BEHAVIOUR

3.1 ONLINE

What kind of actions do customers take online?

The departments can receive direct emails or messages from customers. (Officers on nearby patrol).

3.2 What kind of actions do customers take offline?

Following directions is one of the major tasks for the traveler, but they can utilize the smartboard signs to checkthe state of the road from wherever they are standing.

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 toreduce 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.

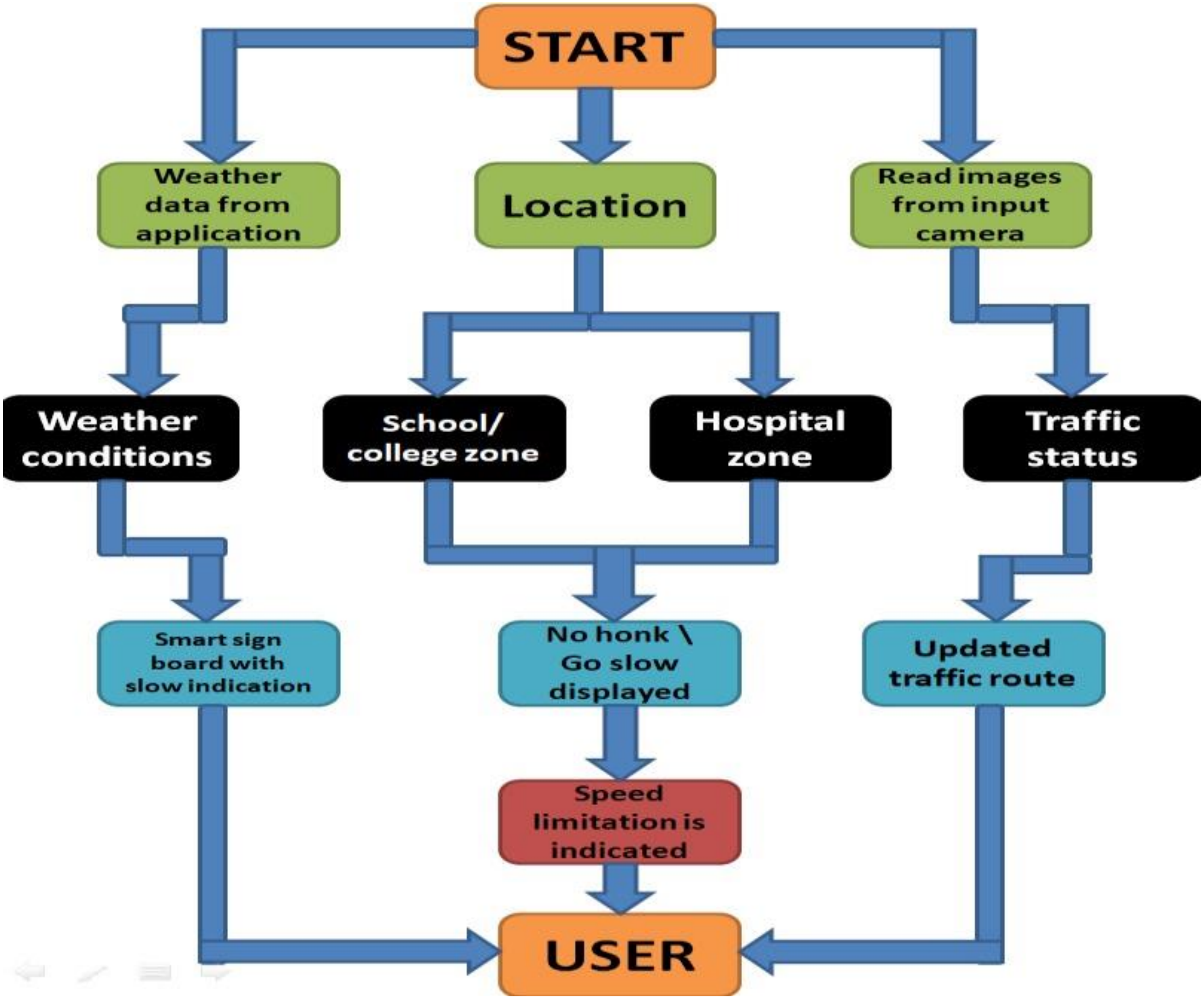
4.2 NON-FUNCTIONAL REQUIREMENTS

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

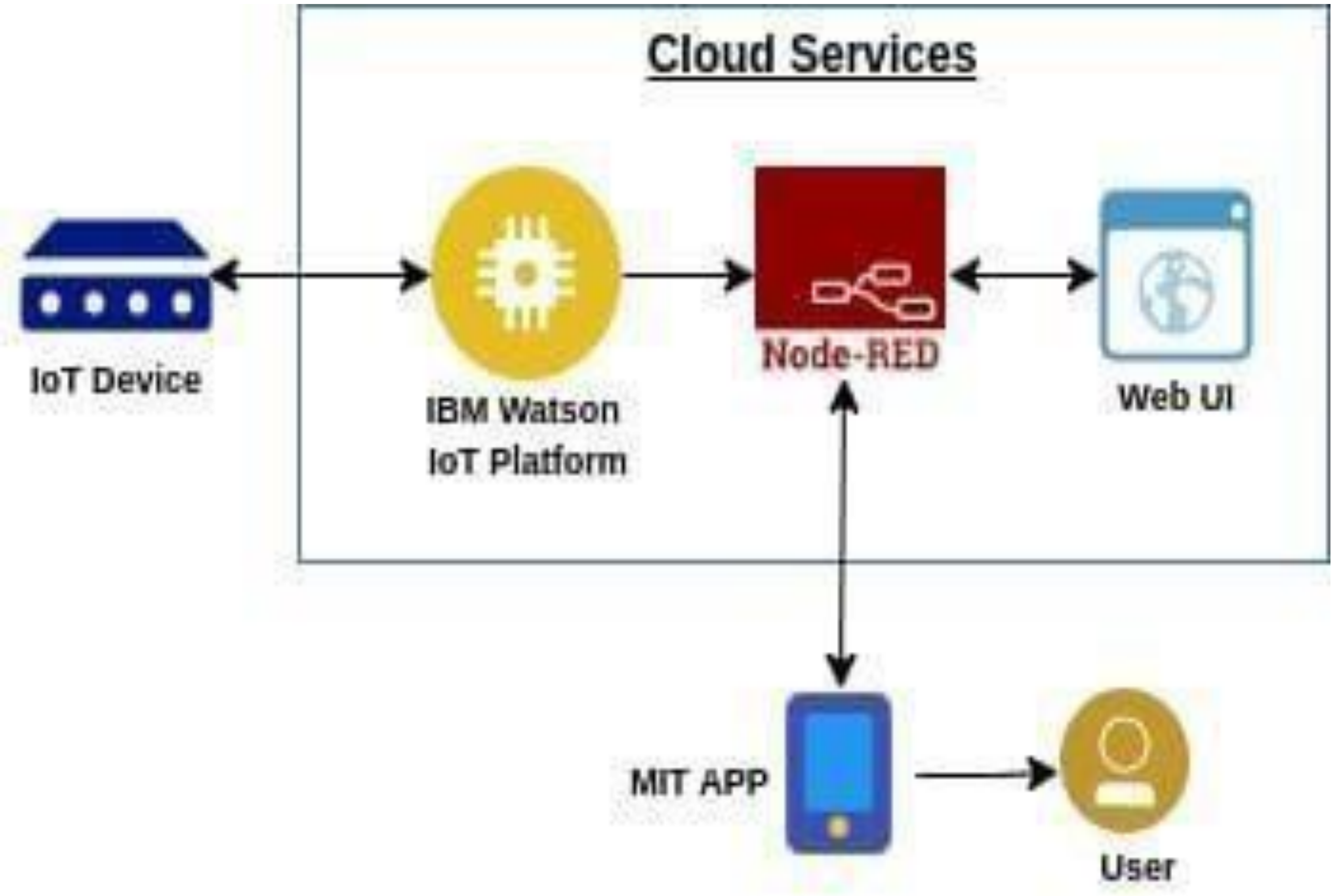
5.1 DATA FLOW DIAGRAMS


START
Weather data from application
Location
Read images from input camera
Weather conditions
School/ college zone
Hospital zone
Traffic status
Smart sign board with slow indication
No honk \ Go slow displayed
Updated traffic route
Speed limitation is indicated
USER

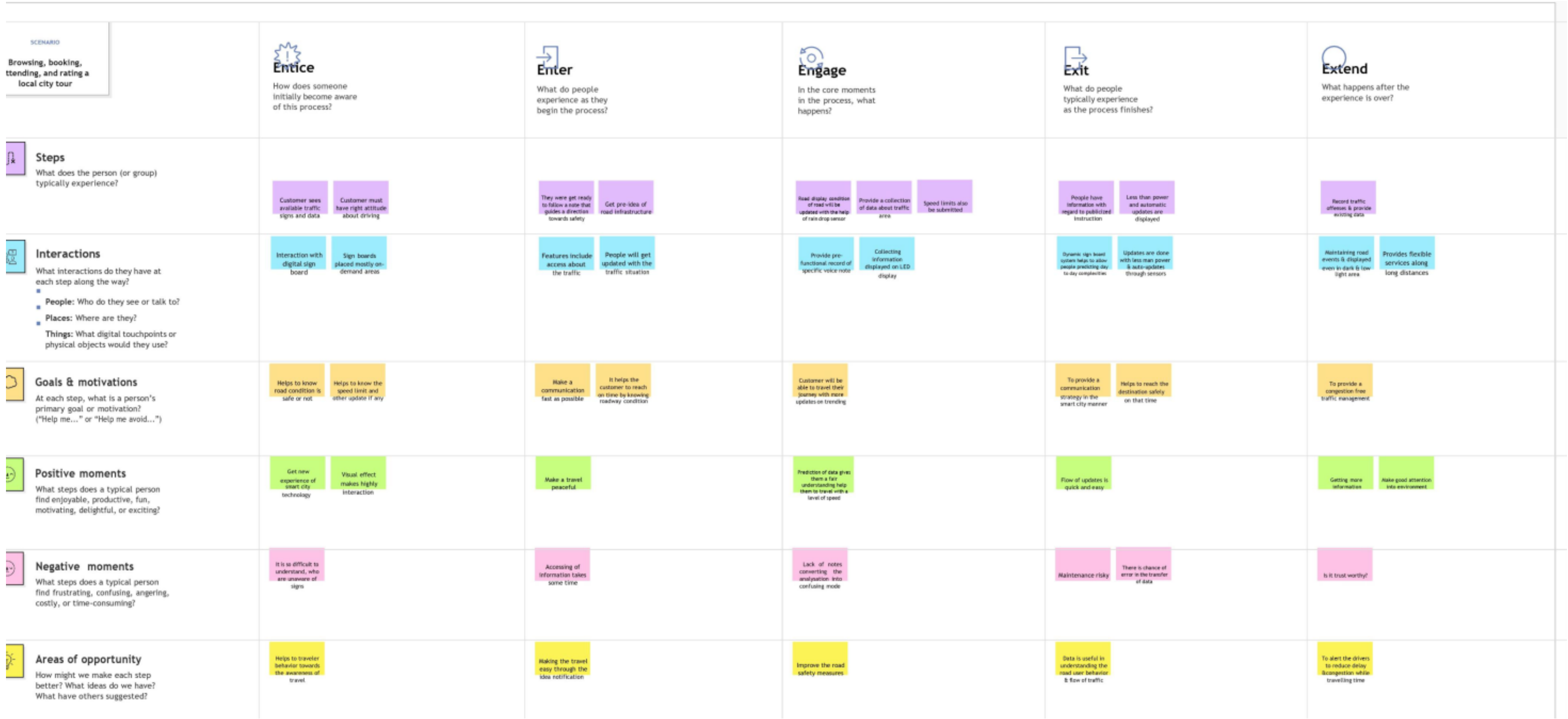
5.2 SOLUTION & TECHNICAL ARCHITECTURE

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.



5.3 USER STORIES



SCENARIO Browsing, booking, ttending, and rating a local city tour	Entice How does someone initially become aware of this process?	Enter What do people experience as they begin the process?	Engage In the core moments in the process, what happens?	Exit What do people typically experience as the process finishes?	Extend What happens after the experience is over?
Steps What does the person (or group) typically experience?	Customer sees available traffic signs and data Customer must have right attitude about driving	They were get ready to follow a note that guides a direction towards safety Get pre-idea of road infrastructure	Road display condition of road will be updated with the help of rain drop sensor Provide a collection of data about traffic area Speed limits also be submitted	People have information with regard to publicized instruction Less than power and automatic updates are displayed	Record traffic offenses & provide existing data
Interactions What interactions do they have at each step along the way? **People:** Who do they see or talk to? **Places:** Where are they? **Things:** What digital touchpoints or physical objects would they use?	Interaction with digital sign board Sign boards placed mostly on-demand areas	Features include access about the traffic People will get updated with the traffic situation	Provide pre-functional record of specific voice note Collecting information displayed on LED display	Dynamic sign board system helps to allow people predicting day to day complexities Updates are done with less man power & auto-updates through sensors	Maintaining road events & displayed even in dark & low light area Provides flexible services along long distances
Goals & motivations At each step, what is a person's primary goal or motivation? ("Help me..." or "Help me avoid...")	Helps to know road condition is safe or not Helps to know the speed limit and other update if any	Make a communication fast as possible It helps the customer to reach on time by knowing roadway condition	Customer will be able to travel their journey with more updates on trending	To provide a communication strategy in the smart city manner Helps to reach the destination safely on that time	To provide a congestion free traffic management
Positive moments What steps does a typical person find enjoyable, productive, fun, motivating, delightful, or exciting?	Get new experience of smart city technology Visual effect makes highly interaction	Make a travel peaceful	Prediction of data gives them a fair understanding help them to travel with a level of speed	Flow of updates is quick and easy	Getting more information Make good attention into environment
Negative moments What steps does a typical person find frustrating, confusing, angering, costly, or time-consuming?	It is so difficult to understand, who are unaware of signs	Accessing of information takes some time	Lack of notes converting the analysation into confusing mode	Maintenance risky There is chance of error in the transfer of data	Is it trust worthy?
Areas of opportunity How might we make each step better? What ideas do we have? What have others suggested?	Helps to traveler behavior towards the awareness of travel	Making the travel easy through the idea notification	Improve the road safety measures	Data is useful in understanding the road user behavior & flow of traffic	To alert the drivers to reduce delay &congestion while travelling time

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	Write a python script	2	MEDIUM

	written	using the inputs given from OpenWeather API		
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**

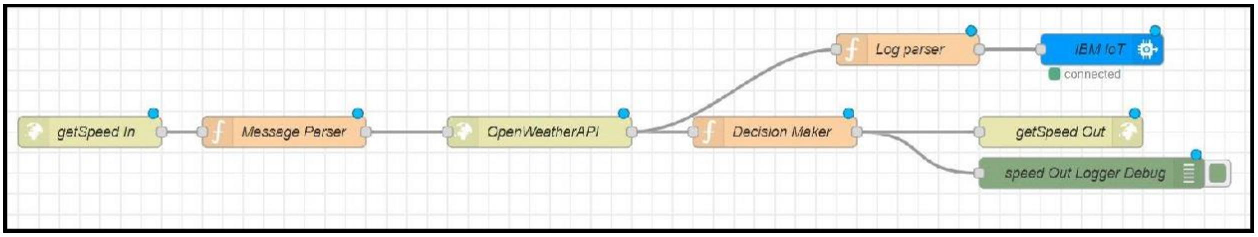
Problem Solution Fit	It helped us understand and analyze all the thoughts of our customers, their choice of options, problems, behavior and emotions.	Completed

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**

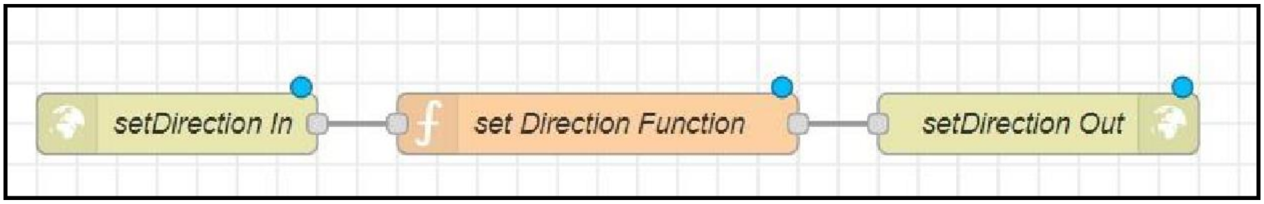
7. CODING & SOLUTIONING

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.

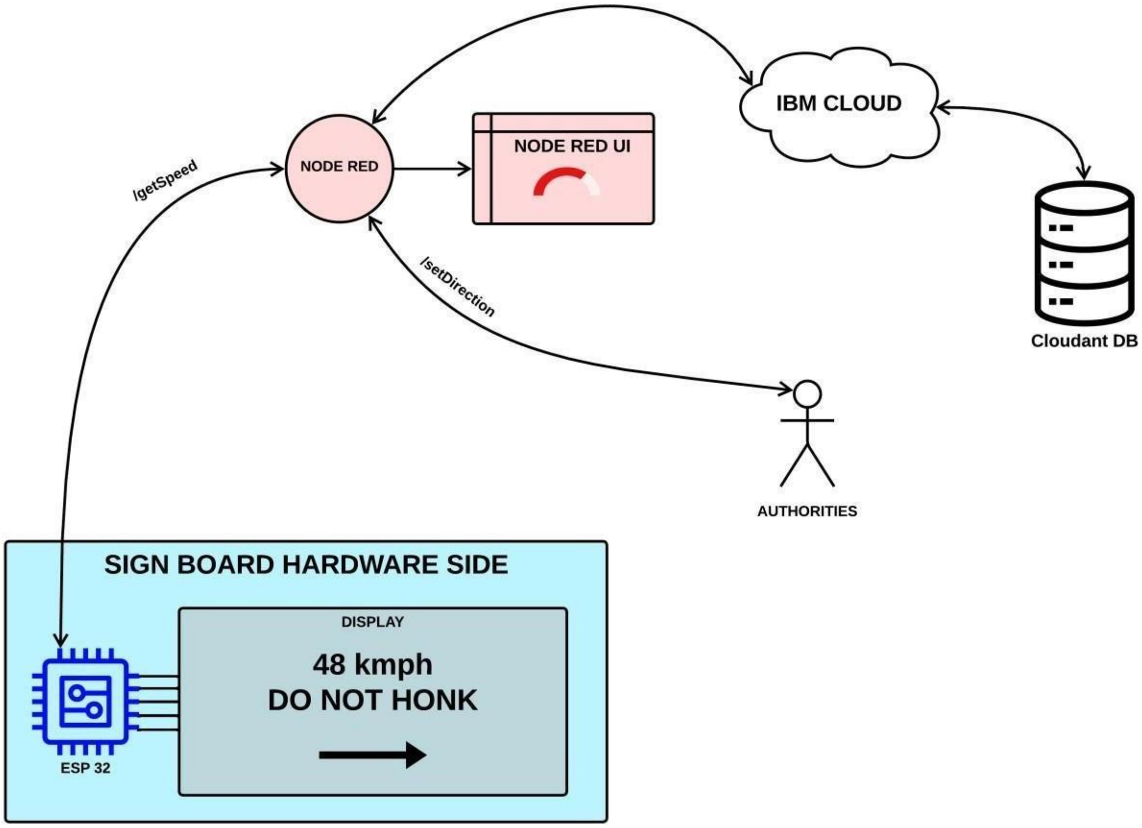
7.2 FEATURE 2 - SET DIRECTION REMOTELY FOR A GIVEN SIGN BOARD



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.

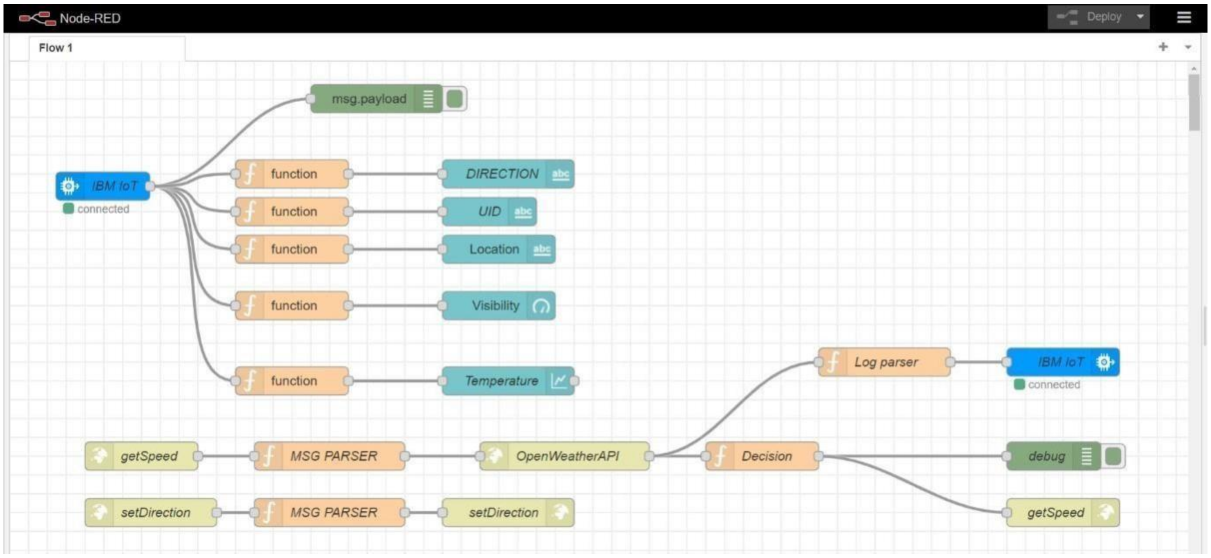
A detailed documentation of all the workflows is available at the following :

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

Node RED :

Node RED flow :

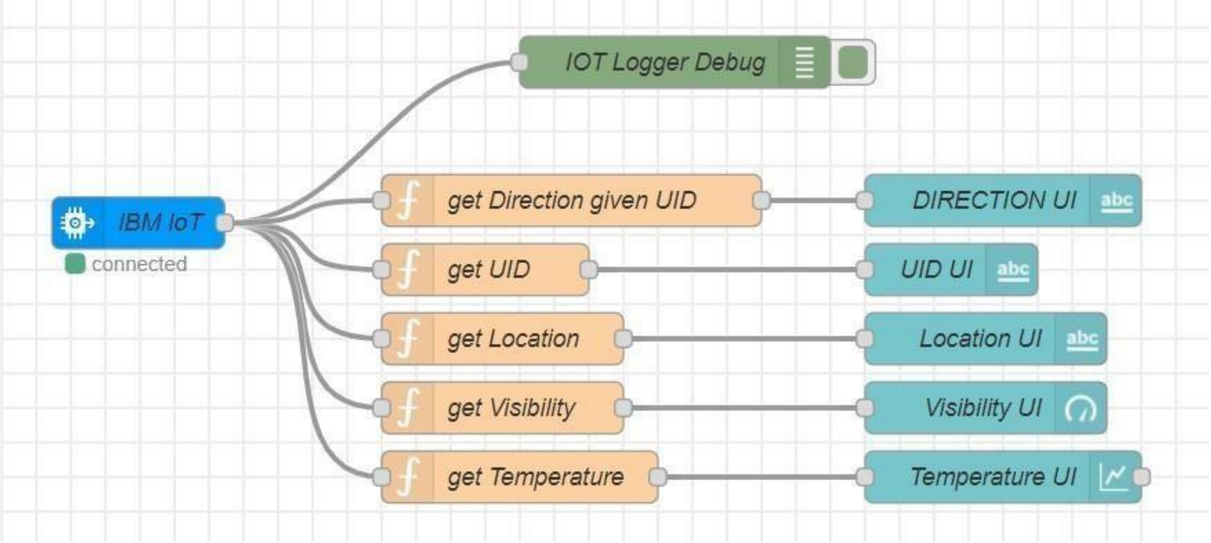


There are 3 flows in the above Node RED flow. They are

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

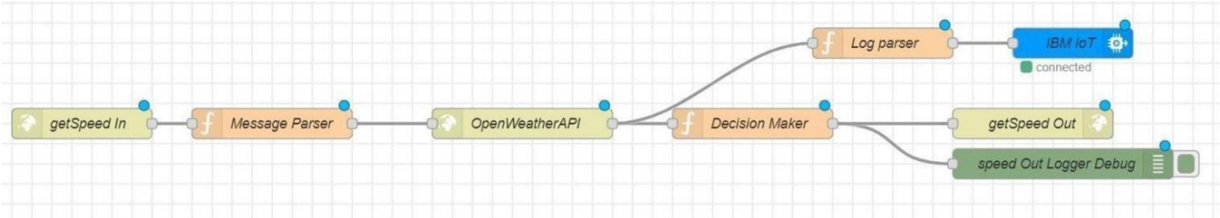
2. The function nodes such as **"get Direction given UID"**, **"get UID"**, **"get Location"**, **"get Visibility"** & **"get Temperature"** extract the respective data out and provides them to the UI nodes **"Direction UI"**, **"UID UI"**, **"Location UI"**, **"Visibility UI"** & **"Temperature UI"**.

```
// 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

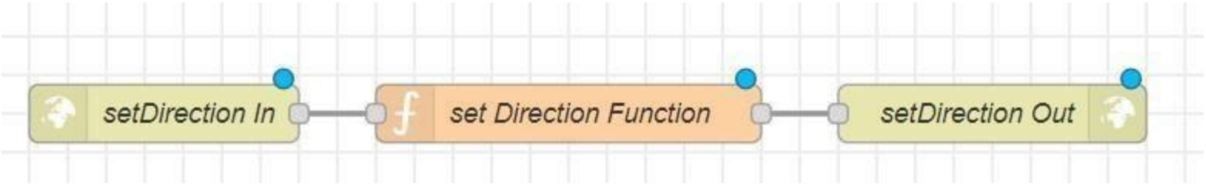
```
weatherObj = JSON.parse(JSON.stringify(msg.payload)); localityObj

= 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

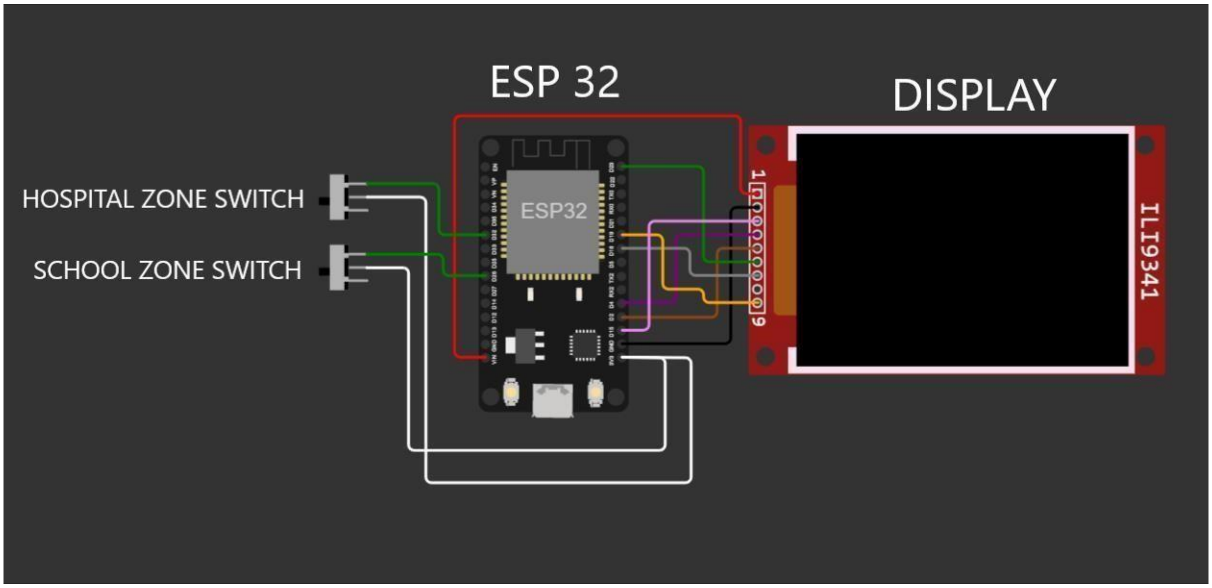
Click on this link to change direction to Right

Wokwi Circuit :

Wokwi Code

Wokwi Link

Circuit Diagram :



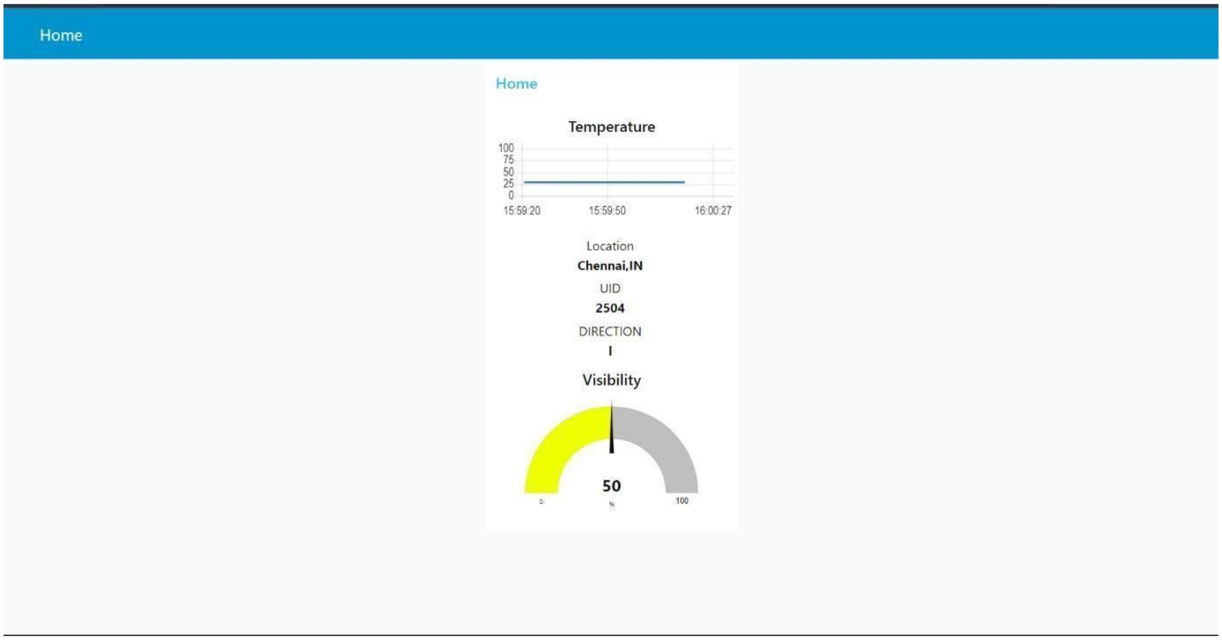
ESP 32 CODE :

```
#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);  }
```

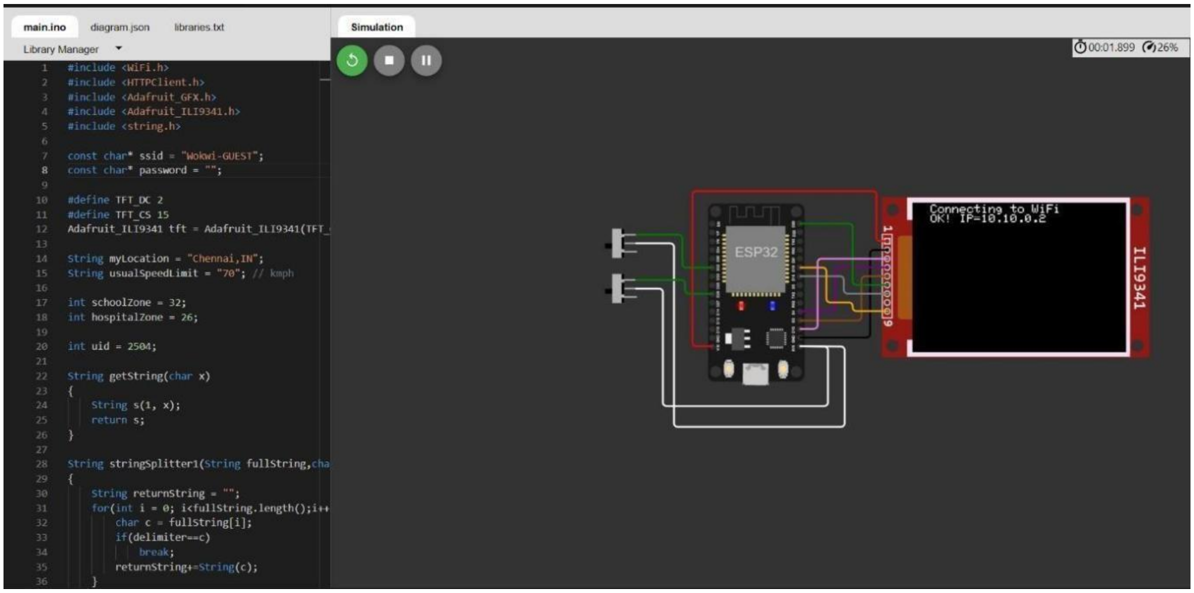
Output :

Node RED Dashboard :



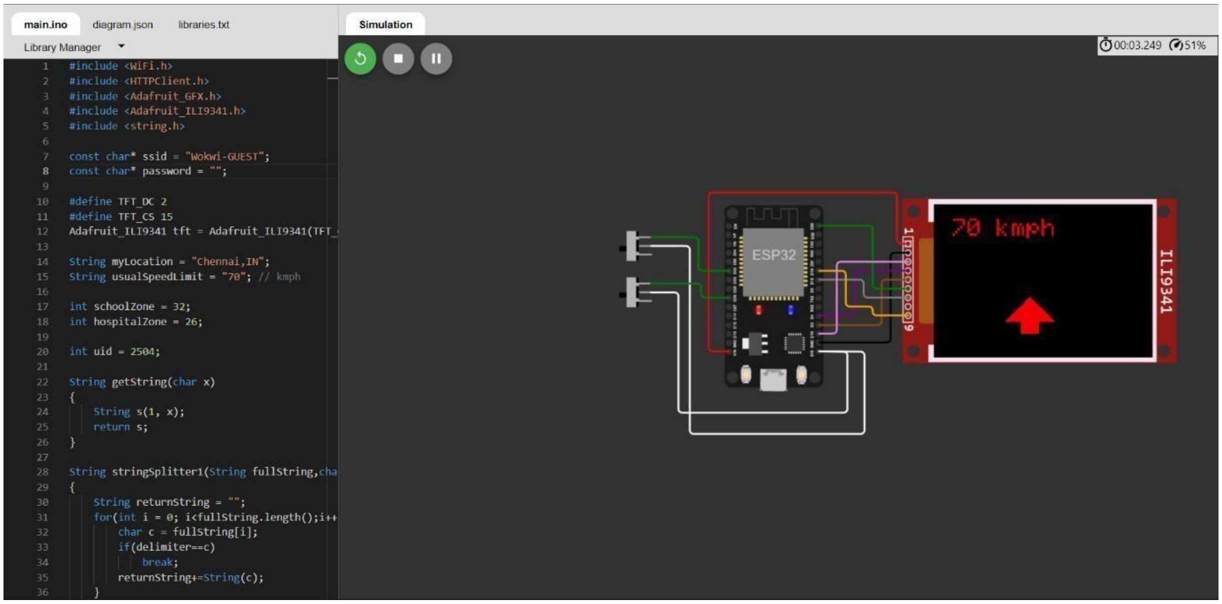
Location
Chennai,IN
UID
2504
DIRECTION
I

Wokwi Output :



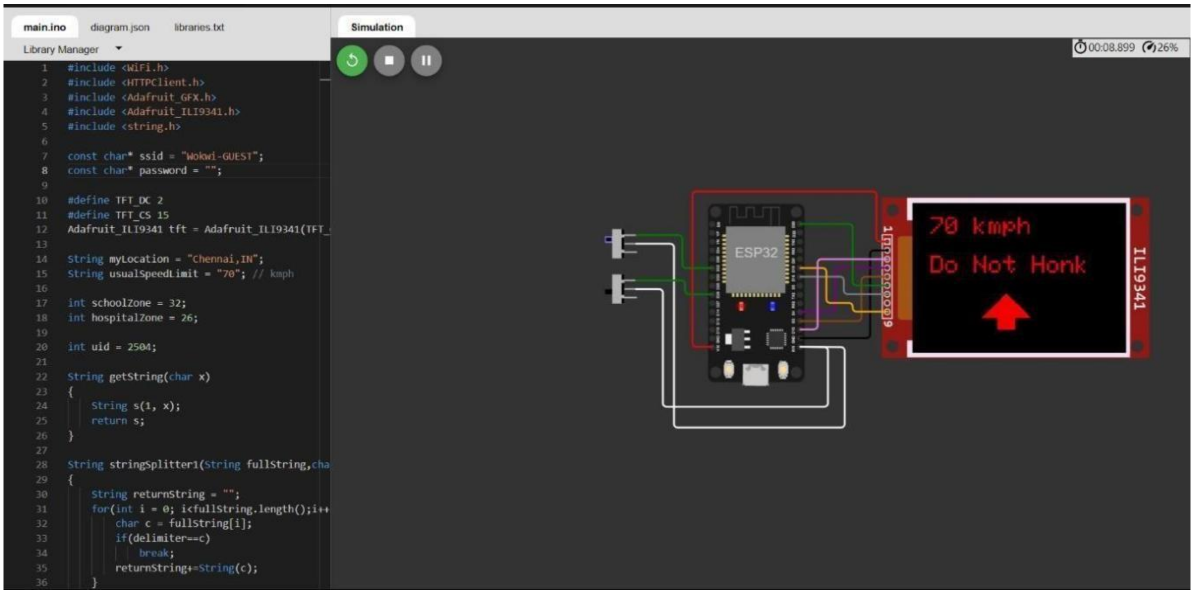
```
#include <WiFi.h>
#include <HTTPClient.h>
#include <Adafruit_GFX.h>
#include <Adafruit_ILI9341.h>
#include <string.h>

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);
    }
```

```
Connecting to WiFi
OK! IP=10.10.0.2
```



```
#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);
    }
```

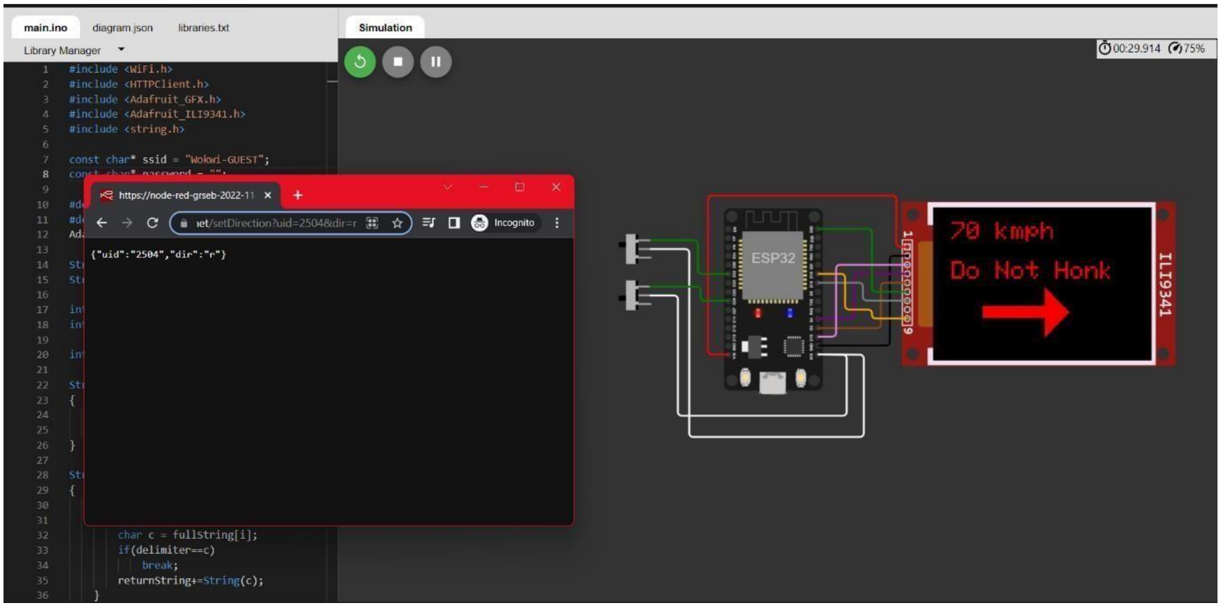
```
70 kmph
```



```
#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);
    }
```

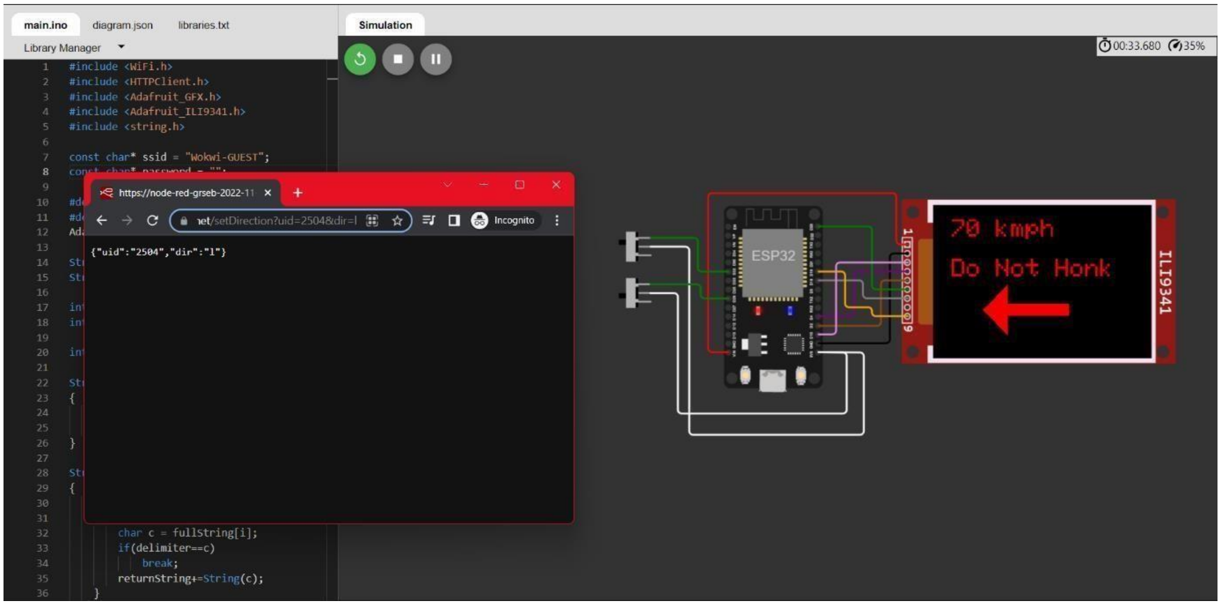
```
70 kmph
Do Not Honk
```

SIGNS WITH SMART CONNECTIVITY FOR BETTER ROAD SAFETY



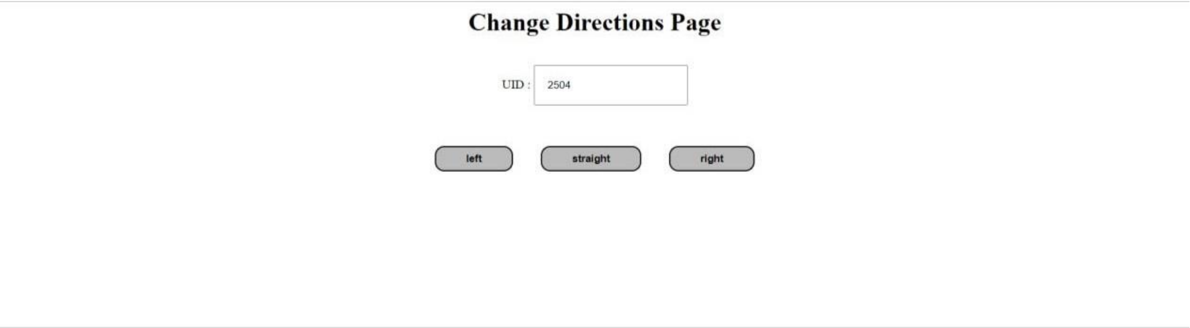
```
#include <WiFi.h>
#include <HTTPClient.h>
#include <Adafruit_GFX.h>
#include <Adafruit_ILI9341.h>
#include <string.h>

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

#d
#d
Ad

St
St

in
in

in

St
{

}

St
{


      char c = fullString[i];
      if(delimiter==c)
        break;
      returnString+=String(c);
    }
```

```
{"uid":"2504","dir":"r"}
```



```
#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 = "";

#d
#d
Ad

St
St

in
in

in

St
{

}

St
{


      char c = fullString[i];
      if(delimiter==c)
        break;
      returnString+=String(c);
    }
```

```
{"uid":"2504","dir":"l"}
```



Change Directions Page

UID : 2504

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

- **SOURCE CODE - ESP 32**

```
#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                    "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        false
for int     0
```

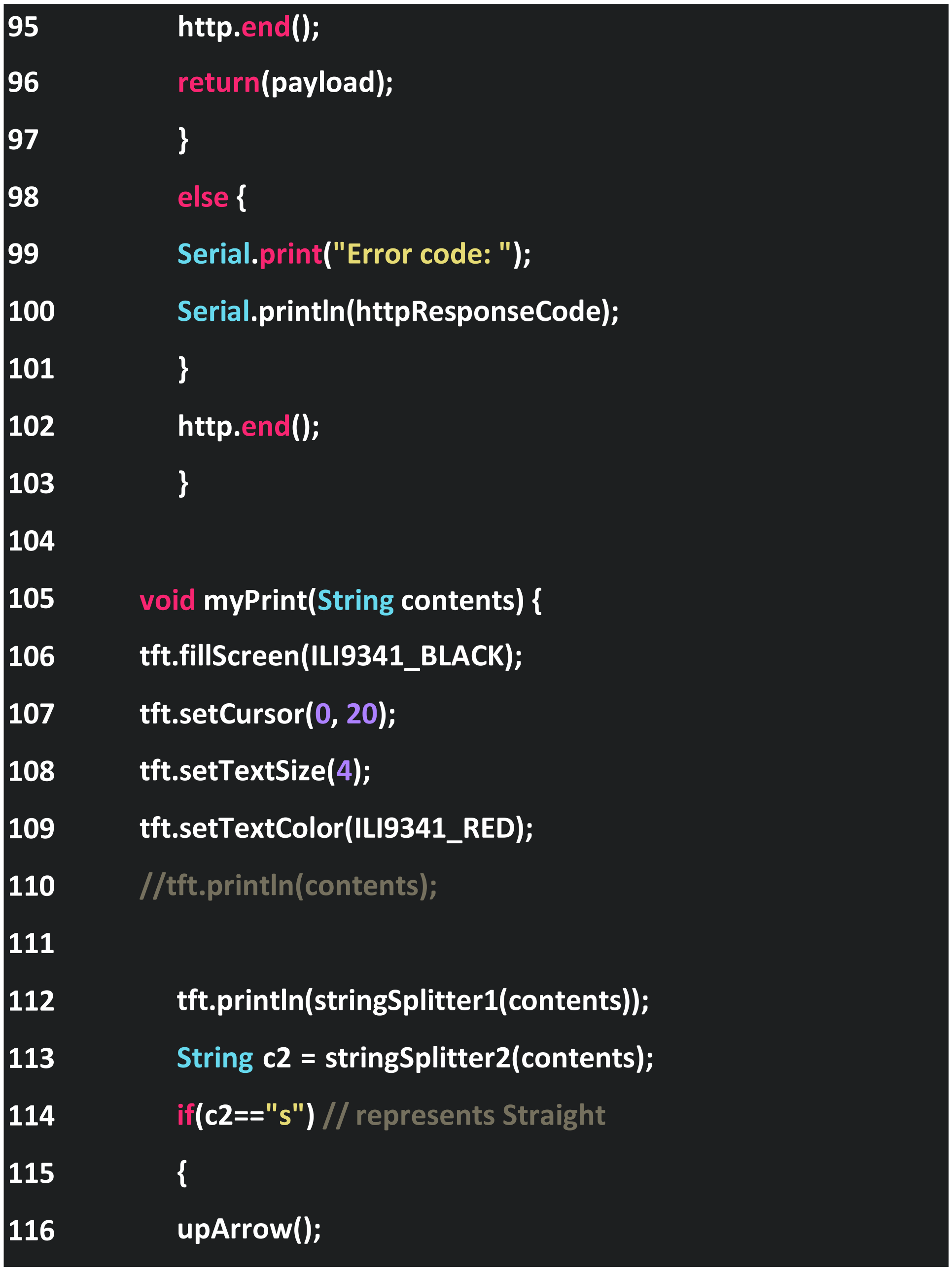
```
     char c = fullString[i];
     if(flag)
     returnString+=String(c);
     if(delimiter==c)
             flag = true;
     }
        return(returnString);
}

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-2022-11-05-test.eugb.mybluemix.net/getSpeed?";
    url += "location="+myLocation+"&";
    url += "schoolZone="+(String)digitalRead(schoolZone)+(String)"&";
    url += "hospitalZone="+(String)digitalRead(hospitalZone)+(String)"& ";
```

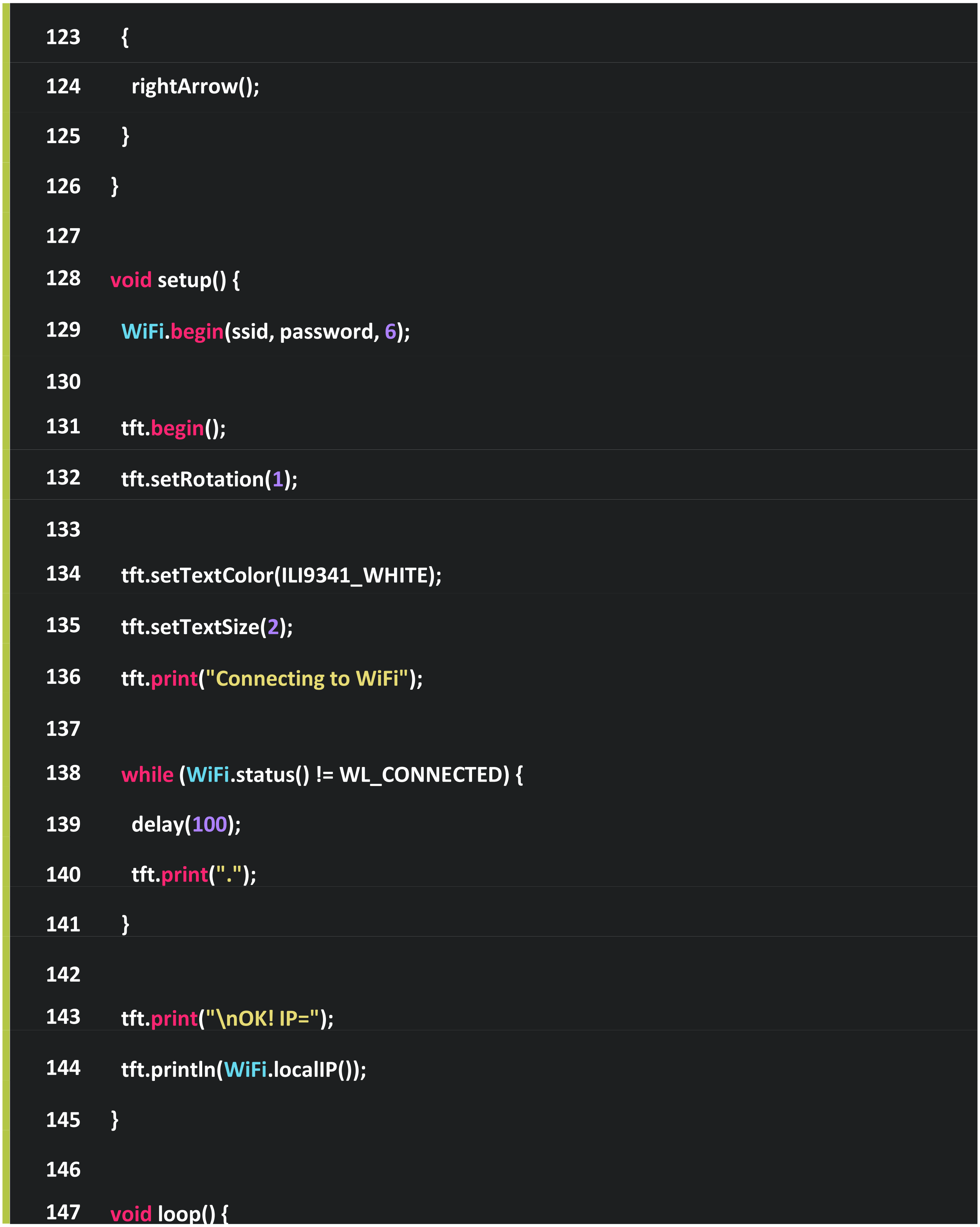
```
        url += "usualSpeedLimit="+(String)usualSpeedLimit+(String)"&";
        url += "uid="+(String)uid;
        http.begin(url.c_str());
        int httpResponseCode = http.GET();

```

if 0 String



```
    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      "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);
}
```

- **GITHUB AND PROJECT LINK**

https://github.com/IBM-EPBL/IBM-Project-47665-1660801062