Signs with Smart Connectivity for Better Road Safety

Batch: B1-1M3E

TEAM ID: PNT2022 TMID06138

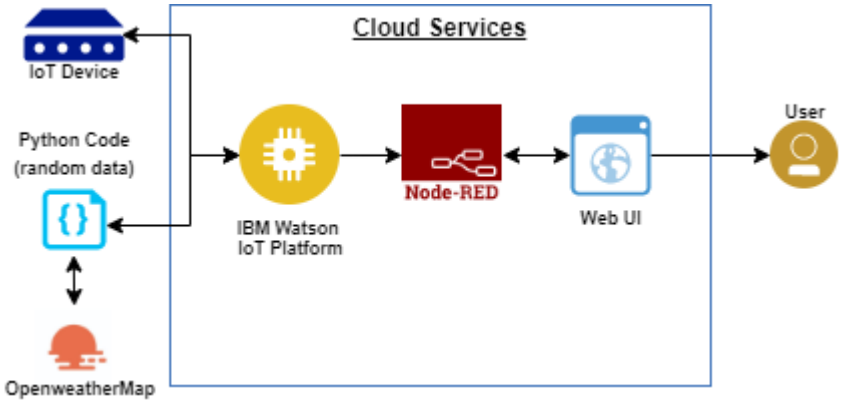
INDEX

INTRODUCTION

Our project is Signs with Smart Connectivity for Better Road Safety. In present Systems the road signs and the speed limits are Static. But the road signs can be changed in some cases. We can consider some cases when there are some road diversions due to heavy traffic or due to accidents then we can change the road signs accordingly if they are digitalized. This project proposes a system which has digital sign boards on which the signs can be changed dynamically. If there is rainfall then the roads will be slippery and the speed limit would be decreased. There is a web app through which you can enter the data of the road diversions, accident prone areas and the information sign boards can be entered through web app. This data is retrieved and displayed on the sign boards accordingly.

Project Overview:

Receiving road sign values to the IBM IOT platform from Node-RED Web UI. Weather conditions can be viewed in the Web Application.



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.
- Based on the traffic and fatal situations the diversion signs are displayed.
- Guide (Schools), Warning and Service (Hospitals, Restaurant) signs are also displayed accordingly.
- Different modes of operations can be selected with the help of buttons.

LITERATURE SURVEY

Existing problem & References

S. NO	ARTICLE	AUTHOR	YEAR	METHODOLOGY	MERIT/DEMERIT
1	An Architecture for Assessing Road Safety in Smart Cities	Abd-Elhamid M.Taha	2018	Use of machine learning in the design of the metric computation core through a novel application of Hidden Markov Models (HMMs). The impact of the proposed architecture is demonstrated through an application to safety-based route planning.	The Safe-System-based Safe Road Transport Systems, with its elements: safe vehicle, safe road, and safe road user
2	Digitalization of highways for vulnerable road safety development with intelligent IoT sensors and machine learning	RajeshSingh, RohitSharma, Shaik Vaseem Akram, Anita Gehlot, Dharam Buddhi, Praveen Kumar Malik, Rajeev Arya	2021	Digitalization of highways using Internet of Things. Smart highway lighting, smart traffic and emergency management for user safety. Real-time implementation of renewable energy sources like wind, solar and piezoelectric on the highways. Smart display board, and AI on highways for smart Vulnerable Road User model.	Embedding the deep learning techniques in the vison node at the traffic junction and the highway lighting controller is able to deliver an intelligent system that provides sustained experience and management of the highways. Smart reflectors, adoption of renewable energy, developing vehicle-to-vehiclecommunication in vehicles, and smart lamp post are the few recommendations forthe implementation of digitalizing highways.
3	Smart transportation system using IoT	P S Saarika, K. Sandhya, T. Sudha	2017	The sign board with embedded RF module and connected sensors working with solar energy as well as in battery will show the place, distance to that place, weather condition, temperature and different routes to those places.	Problems such as traffic congestion, road safety, accident detection, automatic fare collection and limited car parking facilities can be resolved by IoT.

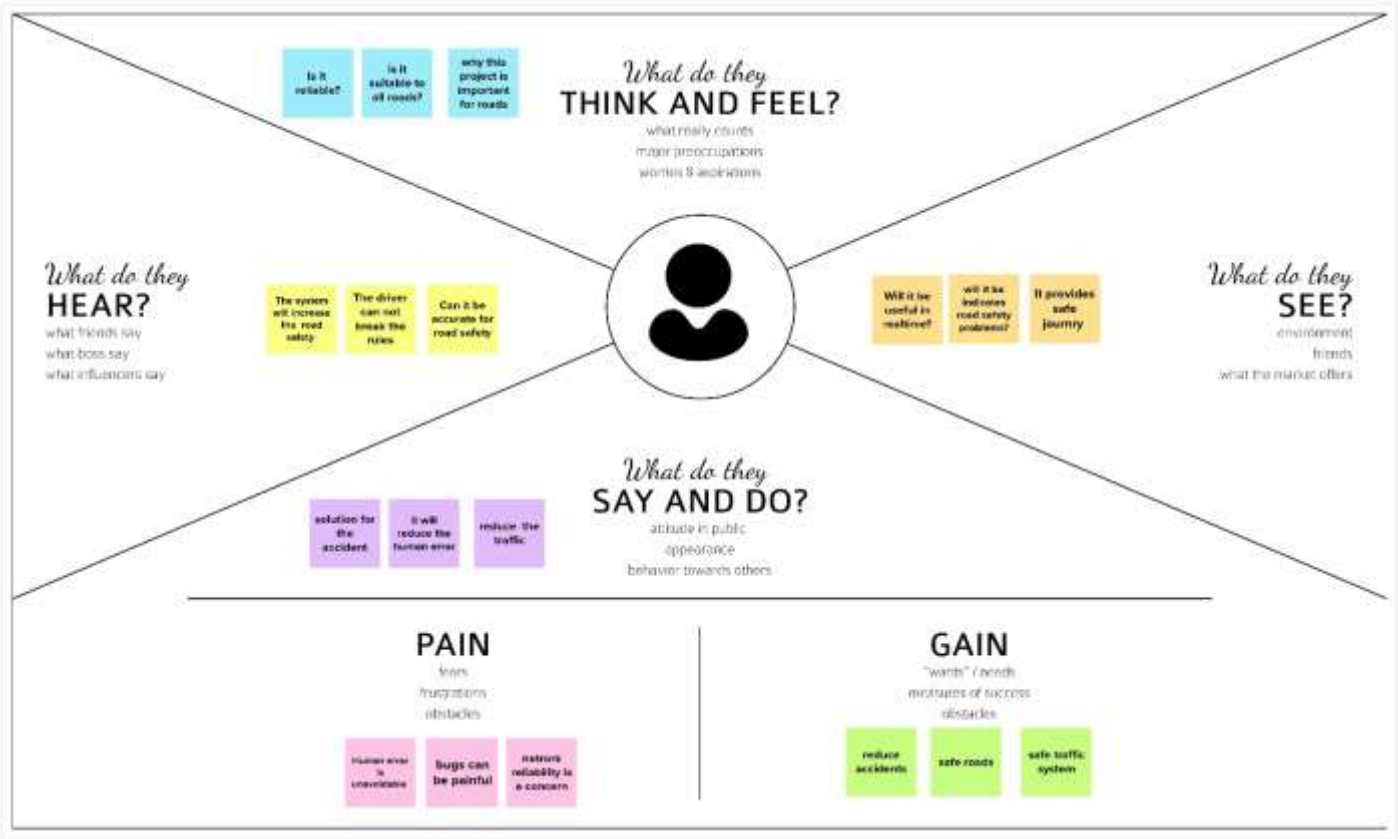
4	IoT Based Intelligent Transportation System (IoT-ITS) for Global Perspective	S.Muthuramalingam, A. Bharathi, S. Rakesh kumar	2018	Multiple regression analysis, Multiple discriminant analysis and logistic regression, Cojoint analysis, Cluster analysis and other big data analytics techniques will merge with IoT and help to build IoT-ITS will be emphasized.	IoT based Intelligent transportation system (IoT-ITS) helps in automating railways, roadways, airways and marine which enhance customer experience about the way goods are transported, tracked and delivered.
5	Internet of Things Based Solutions for Road Safety and Traffic Management in Intelligent Transportation Systems	Arnav Thakur, Reza Malekian, Dijana Capeska Bogatinoska	2017	Road safety techniques studied include distance sensing, improper driving detection and accident prevention, weather related events and negligent driving detection and accident avoidance. Vehicle to vehicle communication and vehicle to infrastructure based channels are studied. Wireless communication technologies suitable for the channels Are studied.	IoT based solutions enabling collection of data from client nodes in a wireless sensor network in the transport environment implementing ITS goals is studied.

Problem Statement Definition

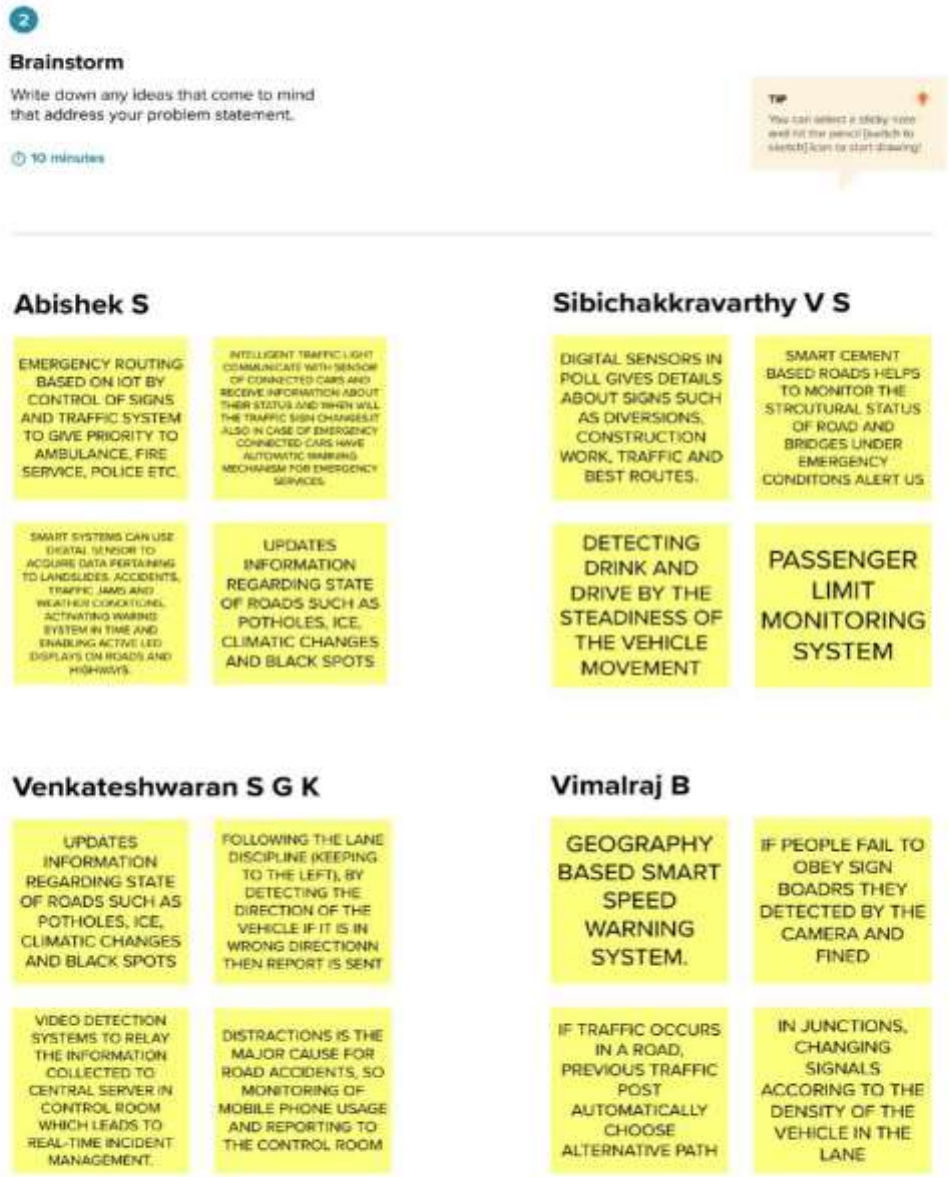
- Man behind the wheel plays an important role in most of the crashes. In most of the cases crashes occurs either due to carelessness.
- In this project reduce the human error and provide the safety roads.
- In this project we have to improve the road safety.
- Road safety refers to the methods and measures taken to prevent a person using the road from being killed or seriously injured.
- It focuses on preventing accidents that result in serious injury and death, keeping in mind that human error can occur.
- The system is monitoring for road so the drivers can not break the rules.

IDEATION & PROPOSED SOLUTION

Empathy Map Canvas:



Ideation & Brainstorming:



3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, 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

IN JUNCTIONS, CHANGING SIGNALS ACCORING TO THE DENSITY OF THE VEHICLE IN THE LANE

SMART PEDESTRAIN CROSSING CONNECTIVITY WITH TRAFFIC SIGNS

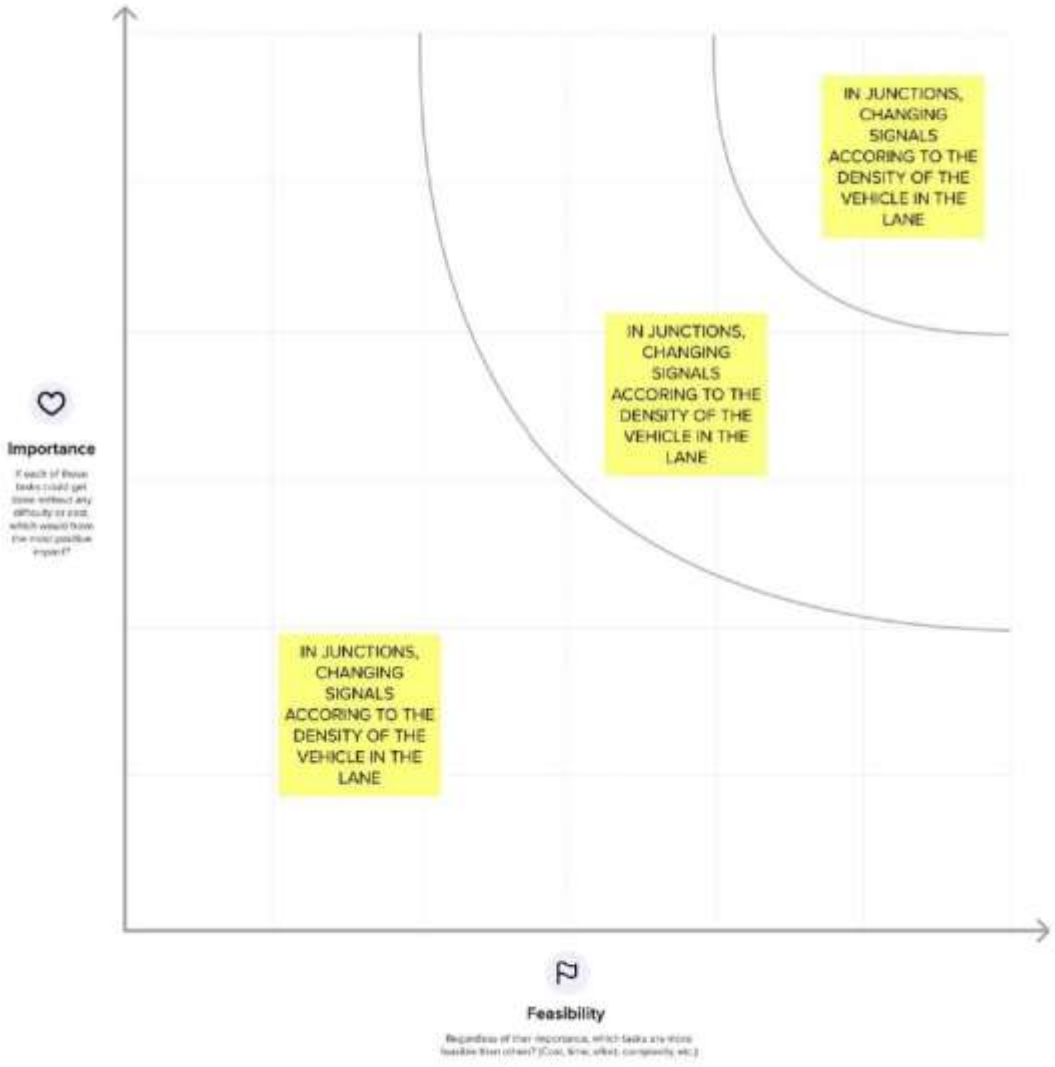
UPDATES INFORMATION REGARDING STATE OF ROADS SUCH AS POTHOLES, ICE, CLIMATIC CHANGES AND BLACK SPOTS

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

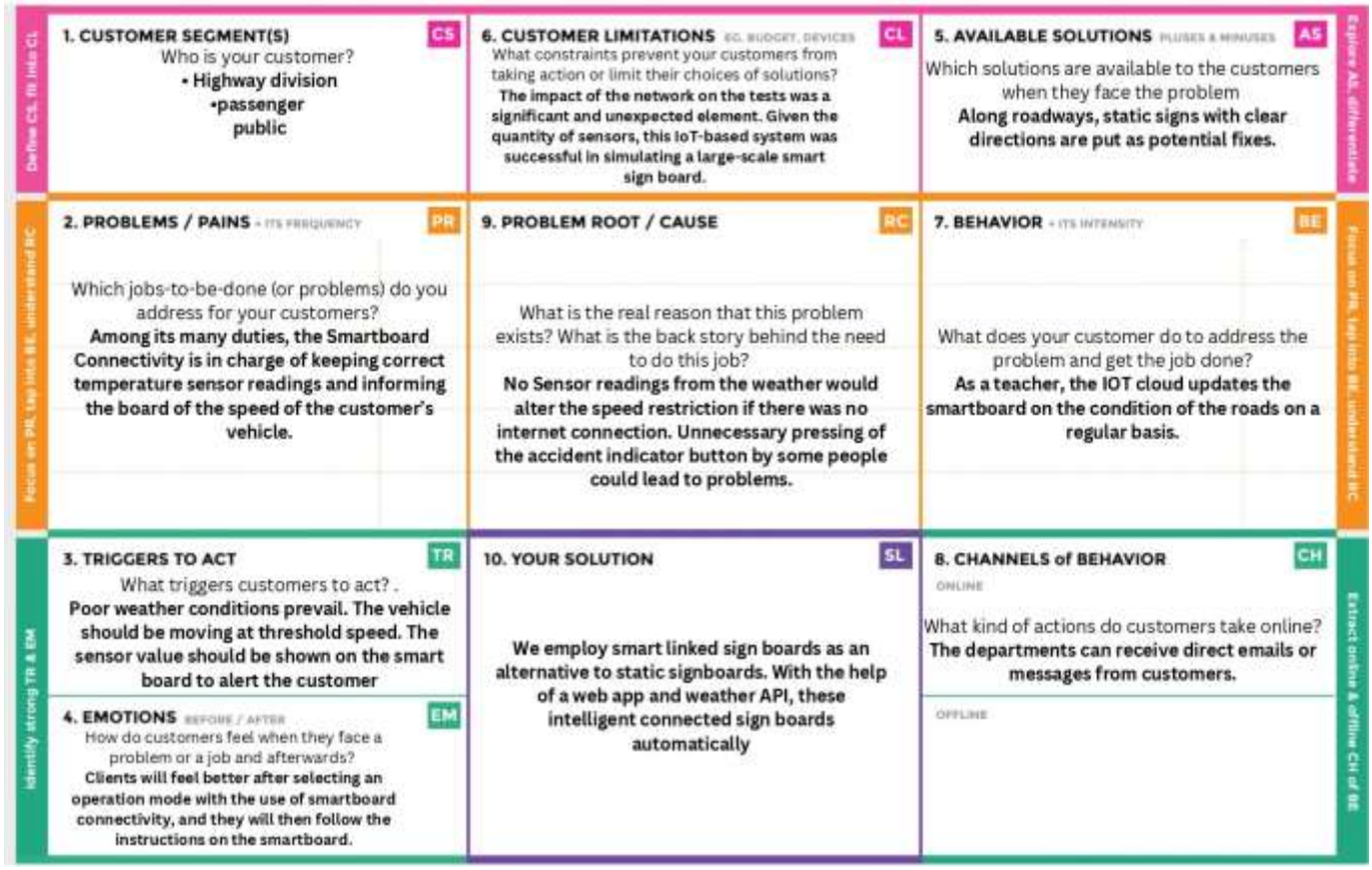


Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to besolved)	Instead of using a static sign board here we are going to use a smart sign simulation system. These smart connected sign boards get the speed limitations from a web app using weather API and update automatically. Based on the weather changes and the accidents happening on the road the speed may increase or decrease Based on the traffic the diversion signs are displayed. Traffic signs will be displayed accordingly. Different modes of operations can be selected with the help of buttons.
2.	Idea / Solution description	The weather and temperature details are obtained from API services.by these details, the speed limit will be updated automatically in accordance with the weather. Also, the details regarding any accidents and traffic congestion faced on the particular road are obtained from previous records.Based on this, the traffic is diverted followed by a change in map path and the traffic is cleared. So in the traffic sign board , some buttons will be placed which will be used to make it generic; where each button will be given a functionality such as changing the warning signs, which are predefined and separate signs will be present for both school and hospital zones.By activating this button, either through the web application or the physical buttons, sign of the board can be changed accordingly, and the speed limit will also be set depending upon the zones. Also, the pedestrians are given an option to change the traffic signs if they want to cross the road. If the pedestrian presses the button that is present on the post at the end of the road, then the traffic will be analyzed immediately.According to the sign signals will Get changed.

3.	Novelty / Uniqueness	Generic Sign board for all applications that uses both buttons and web service for updating Pedestrians are given the access to request the sign change of the signal to cross the road
4.	Social Impact / Customer Satisfaction	Diversion reasons will be displayed If there is no traffic, pedestrians can cross the street without waiting. Customers can reach the Destination before the expected time.
5.	Business Model (Revenue Model)	Since APIs are used to actively monitor the customer's environment, this project employs a business strategy in which revenue will be generated on the basis of the length of time in which the customers actively interact with the product. This product is aimed to be free of cost to the public, but the revenue will be generated by selling this product to the government at a low cost, so there will be less accidents and the public will be aware of the discrepancies or accidents in the particular road. The public will also gain all the information about the road, even if they are checking for an alternate path because of some mishaps that happen on the roads and these functionalities will increase the value of The product in the global market.
6.	Scalability of the Solution	In the future, if any update is required either on the hardware or software side, it can be easily implemented. The hardware components can be directly interfaced with the microcontroller and small modifications can be made in the programming of the existing product. In case Of the software, the website application has to be updated with the additional functionality by creating a new section for the updated hardware. So this will not affect the existing functionality of the product and new functionality can be easily integrated. In addition, a separate circuit will be kept along with the hardware to detect any problem which informs the web application. Also a notification Will be sent to the product service department.

Problem Solution fit:



REQUIREMENT ANALYSIS

Functional Requirement:

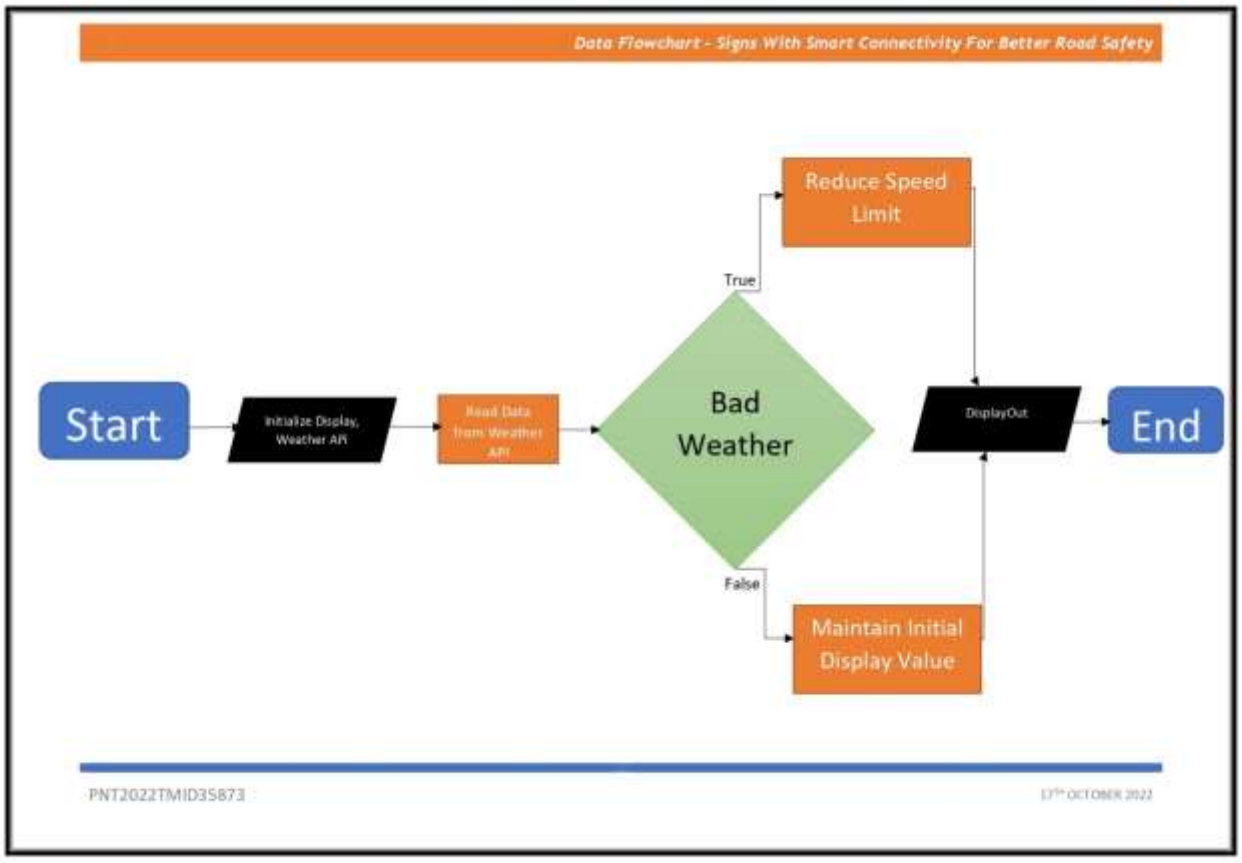
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Requirements	Static signboards will be replaced with smart linked signboards that meet all criteria.
FR-2	User Registration	User Registration can be done through a Website or Gmail
FR-3	User Confirmation	Phone Confirmation Email confirmation OTP authentication
FR-4	Payments options	Bank Transfers
FR-5	Product Delivery and installation	The installation fee will be depend upon the length of the road.
FR-6	Product Feedback	Willbe shared through a website via Gmail

Non-Functional Requirement:

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Will provide the clear product instructions And a self-explanatory product which is simple to use.
NFR-2	**Security**	Cloud data must be contained within the network, collapsing to be the real-time avoidance should Be avoided, and the board will be monitored constantly.
NFR-3	**Reliability**	Hardware will be frequently tested.
NFR-4	**Performance**	The smart board must provide a better user experience and deliver the accuracy output.
NFR-5	**Availability**	All of the functions and the user demands will be provided, depend upon the customer needs.
NFR-6	**Scalability**	The product is based on road safety and should cover the entire highway system.

PROJECT DESIGN

DataFlow Diagram:



Solution & Technical Architecture:

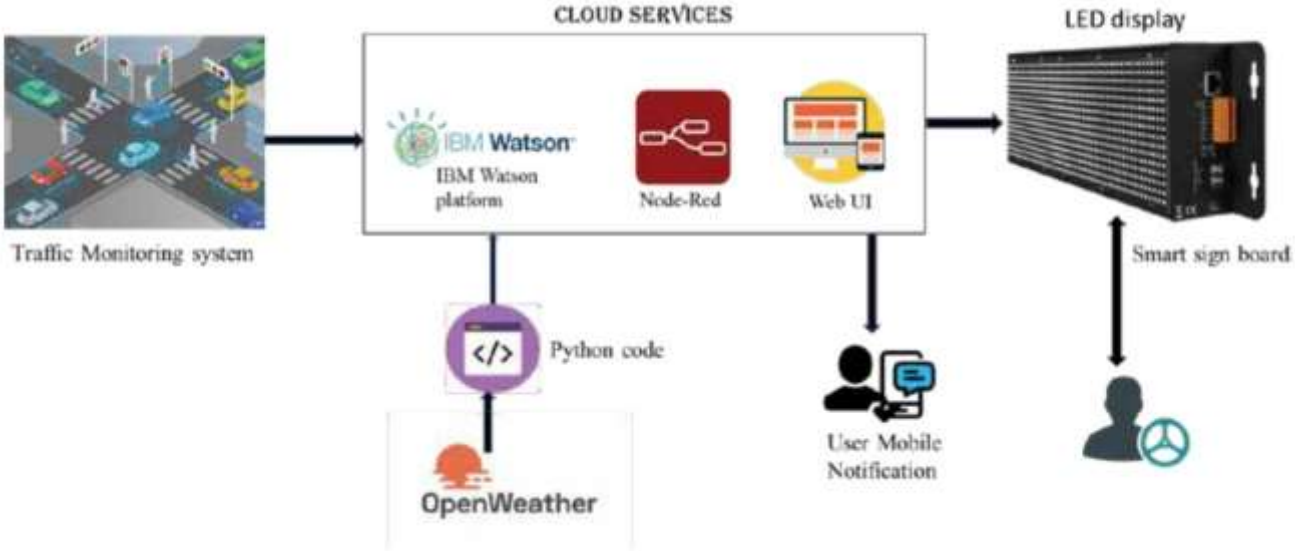


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript / Angular Js / React Js etc.
2.	Application Logic-1	Logic for a process in the application	Java / Python
3.	Application Logic-2	Logic for a process in the application	IBM Watson STT service
4.	Application Logic-3	Logic for a process in the application	IBM Watson Assistant
5.	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc.
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
7.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
8.	External API-1	Purpose of External API used in the application	IBM Weather API, etc.

Table-2: Application Characteristics:

S. No	Characteristics	Description	Technology
1.	Security Implementations	Strong security system that anyone without login credentials and hackers are not allowed to enter the network.	Firewall, Firebase, cyber resiliency strategy
2.	Scalable Architecture	Easy to expand the operating range by increasing the bandwidth of the network.	IoT, internet.
3.	Availability	Available anytime and everywhere 24/7 as long as the user is signed into the network.	IBM Cloud
4.	Performance	Supports a large number of users to access the technology simultaneously.	IBM cloud

USER STORIES:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	Access my account / dashboard	High	Sprint-1
Weather	openweathermap	USN-2	As a user, I want to check the weather of that location	Get the weather of that location	High	Sprint-1
IoT devices	Automation	USN-3	As a user, I want to use IoT devices for automation purposes	Get the work done without manual effort	High	Sprint-2
Python code	Random data	USN-4	As a user, I want to give some input to the devices for performing some action to complete the tasks very easily	Get the data workflow	Medium	Sprint-1
IBM Cloud	Cloud services	USN-5	As a user, I want to deploy these application for public version	Useful for all domain users	High	Sprint-1
Node-Red	Integration	USN-6	As a user, I want to integrate the applications with hardware	To precise for linear workflow	Medium	Sprint-3
Web UI	Interaction	USN-7	As a user, I want to interact with the digital products	To interact with the users	Medium	Sprint-2
Data validation	Checking accuracy	USN-8	As a user, I can check the ability and accuracy of the model in obtaining the required information	Check the capability of the model	High	Sprint-2
Data extraction	Obtaining the data	USN-9	As a user, I can retrieve the result data from the application for data storage for further uses	Download the result in the form of data	High	Sprint-3

PROJECT PLANNING &

SCHEDULING

Sprint Planning & Estimation:

Sprint	User Story Number	User Story / Task	Priority
Sprint-1	US-1	Create the IBM Cloud services which are being used inthis project.	High
Sprint-1	US-2	Configure the IBM Cloud services which are being usedin completing this project.	Medium
Sprint-1	US-3	IBM Watson IoT platform acts as the mediator to connect the web application to IoT devices, so createthe IBM Watson IoT platform.	Medium
Sprint-1	US-4	In order to connect the IoT device to the IBM cloud, create a device in the IBM Watson IoT platform and get the device credentials.	High
Sprint-2	US-1	Configure the connection security and create API keys that are used in the Node-RED service for accessing the IBM IoT Platform.	High
Sprint-2	US-2	Create a Node-RED service.	High
Sprint-3	US-1	Develop a python script to publish random sensor data such as temperature, humidity, rain to the IBM IoT platform	High
Sprint-3	US-2	After developing python code, commands are received just print the statements which represent the control of the devices.	Medium
Sprint-3	US-3	Publish Data to The IBM Cloud	High
Sprint-4	US-1	Create Web UI in Node- Red	High
Sprint-4	US-2	Configure the Node-RED flow to receive data from the IBM IoT platform and also use Cloudant DB nodes to store the received sensor data in the cloudant DB	High

SPRINT DELIVERY SCHEDULE & JIRA REPORT:

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

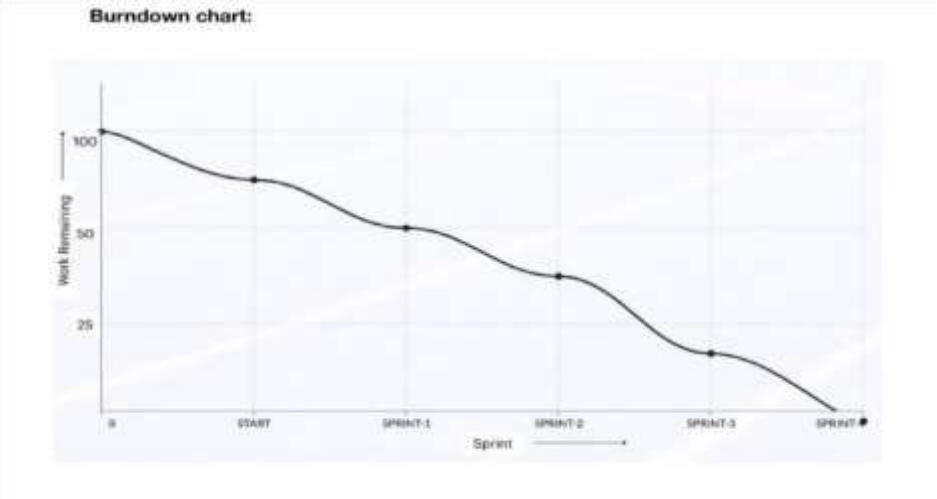
Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) periteration unit (story points per day).

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

Burndown Chart:

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies suchas Scrum. However, burn down charts can be applied to any project containing measurable progress over time.



CODING & SOLUTIONING

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 infoare 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 an http response.
- This data is displayed at the microcontroller. Thus, a lot of battery is saved due to lesser processing time.

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.

TESTING

TEST CASES:

❖ TEST CASE 1
Clear weather - Usual Speed Limit.

❖ TEST CASE 2
Foggy Weather - Reduced Speed Limit.

❖ TEST CASE 3
Rainy Weather - Further Reduced Speed Limit.

❖ TEST CASE 4
School/Hospital zone – Do not horn sign is displayed.

USER ACCEPTANCE TESTING:

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

RESULT

PERFORMANCE METRICS:

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

ADVANTAGES & DISADVANTAGES

ADVANTAGES:

The Internet of Things in the transport industry is the unification of objects into a single network using actuators, built-in sensors, and other devices that collect and transmit data to a single central console.

One of the advantages of IoT technology is that sensors and beacons can be installed in cars and trains and integrated into street lamps, bus stops, and railway platforms. That will ensure consistent visibility and reduce accidents. In addition, transport organizations will be able to analyze the data passing through IoT devices and improve the quality and efficiency of their services. As a result of the traffic management IoT, the industry will benefit from:

Safe travel

The ability to track metrics such as train speed, carriageway temperature, or the number of vehicles at an intersection using IoT technology can help improve the safety of transportation systems around the world.

Efficient processes

Municipalities and organizations adopting IoT technologies are reaping significant productivity benefits. They can better monitor critical infrastructures and design efficient processes to minimize operating costs and increase system throughput.

Improving the environment

IoT-enabled systems can quickly respond to changing traffic patterns and return real-time data to help drivers plan their journeys better with better congestion monitoring. Reducing congestion and energy consumption has a positive impact on the environment.

DISAVANTAGES:

Additional security requirements

As the number of IoT devices connecting to the central network grows, its vulnerability increases the likelihood of hacker attacks to take over confidential data. Therefore, any traffic management solution must have a basic and an additional level of security. Typically that data transmission encryption, user access control, and device authentication.

The need for high-tech network infrastructure

All high-tech solutions require high-speed data transfer methods. Since IoT solutions involve working with large amounts of data and many IP addresses, network facilities must maintain constant communication and collect data from sensors and other IoT devices. Thus, to operate efficiently, you need strong networks that can meet the needs of the Internet of Things as they evolve.

Time spent on adaptation

The IoT-based intelligent sign management system includes thousands of sensors and devices, and it is impossible to manage all of these endpoints manually. A digital traffic solutions provider must provide an intuitive IoT traffic control system with an automatic connection and a centralized control panel.

Additional investment

As mentioned above, the implementation of IoT technology in the long term reduces costs and optimizes the operations of any organization. But the initial integration requires investment.

These are the costs associated with the network infrastructure, the modernization of vehicles directly connected to the IoT solution, and the planning, implementation, management, and security of IoT systems. Therefore, many

Customers immediately submit IoT development on outsourcing to Ukraine, known for its pool of talented programmers and quality solutions at a fair price.

FUTURE SCOPE

The technology enables you to control traffic, catch the lawbreakers, and provide road safety. Light Detection and ranging gun is a weightless and simple tool, which enables law officials to catch and book vehicles that crosses the speed limit.

Using new technology such as smart traffic light and traffic control systems, artificial intelligence, the use of telematics and automotive technology can contribute to prevent and reduce the number of road related accidents and improve road safety.

Connected technologies and the IoT improves in:

- Vehicle safety
- Efficiency
- Convenience
- Overall customer experience
- Operational performance

CONCULUSION

Static signboards are not very efficient and cannot properly help the drivers hence, this leads to accidents, Time wastage and a lot problems .In coming year communication will become very fast due well developing technologies due to that it became reliable. This project will be very helpful and it is a very necessary project which will reduce a whole lot of accidents and save lines this project can be used by the government to improve road safety.

APPENDIX

Code:

Python – code.py

```
import wiotp.sdk.device
import time
import random
import ibmiotf.application
import ibmiotf.device
import requests, json

myConfig = {
    "identity": {
        "orgId": "e4jrbo",
        "typeId": "SignsWithSmartConnectivity",
        "deviceId":"12345"
    },
    "auth": {
        "token": "1234567890"
    }
}

def myCommandCallback(cmd):
    print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
    m=cmd.data['command']

client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()


#OpenWeatherMap Credentials

URL                                                                    =
"http://api.openweathermap.org/data/2.5/weather?q=Kāraikkudi,IN&units=metric&a
ppid=76e08ef85f6173baed5302d8d21a6d24"


while True:
    response = requests.get(URL)
    if response.status_code == 200:
        data = response.json()
```

```
main = data['main']
temperature = main['temp']
humidity = main['humidity']
vis = (data['visibility'])/1000
place=data['name']
wea= data['weather'][0]['main']

vis_ms=""
if vis>=10:
    vis_ms+="Road is visible"
else:
    vis_ms+="Visiblity is Low, Drive safely"

msg=random.randint(0,5)
if msg==1:
    message="GO SLOW, SCHOOL ZONE AHEAD"
elif msg==2:
    message="NEED HELP, POLICE STATION AHEAD"
elif msg==3:
    message="EMERGENCY, HOSPITAL NEARBY"
elif msg==4:
    message="DINE IN, RESTAURENT AVAILABLE"
elif msg==5:
    message="PETROL BUNK NEARBY"
else:
    message=""

speed=random.randint(0,150)
if speed>=100:
    speedMsg="Speed Limit Exceeded"
elif speed>=60 and speed<100:
    speedMsg="Moderate Speed"
else:
    speedMsg="Slow and steady"


if temperature < 24:
    visibility="cold weather, Drive Slow"
elif temperature < 20:
    visibility="Bad Weather, Be Careful"
else:
```

```
        visibility="Clear Weather, Safe Journey"


    sign=random.randint(0,6)
    if sign==1:
        signMsg="Right Diversion"
    elif sign==2:
        signMsg="Speed Breaker"
    elif sign==3:
        signMsg="Left Diversion"
    elif sign==4:
        signmsg="U Turn"
    elif sign==5:
        signMsg="Under Repair"
    else:
        signMsg=""


    myData={'Temperature':temperature,'Visibility':vis,  'temp-msg':visibility,
'Sign_msg':signMsg, 'Vis_msg':vis_ms, 'LM_msg':message, 'Speed_msg':speedMsg,
'Humidity':humidity, 'Place':place, 'Weather':wea}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,
onPublish=None)
    print("Published data Successfully:", myData)
    client.commandCallback = myCommandCallback
    time.sleep(2)
client.disconnect()
```

Simulation – ESP32

```
#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;

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-nwmrt-2022-11-04.eu-gb.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);
}
```

Link:

Github:

https://github.com/IBM-EPBL/IBM-Project-15252-1659595968

Wokwi Simulation:

https://wokwi.com/projects/348220756379828820

Node-Red:

https://signs-with-smart-connectivity.eu-gb.mybluemix.net/ui

MIT APK:

https://github.com/IBM-EPBL/IBM-Project-15252-1659595968/blob/main/Final%20Deliverables/SignsWithSmartConnectivity.apk

Demo Video:

https://drive.google.com/file/d/1oiS2fA9vnHdQRHpKTd7UXa43ZWFHwYjq/view?usp=share_link

Report- google Doc:

https://docs.google.com/document/d/1oKByxT23f2pP3H0c5otr0tBnPXs_23kV/edit?usp=sharing&ouid=114510085926180062030&rtpof=true&sd=true