Project Report

Date	17 Nov 2022
Team ID	PNT2022TMID39076
Project	Signs with Smart Connectivity for Better Road Safety

1. INTRODUCTION:

1.1 Project Overview:

- The Objective of this is to replace the static signboards. Instead, 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 (for Schools), Warning, and Service (Hospitals, Restaurants) signs are also displayed accordingly.

1.2 Purpose:

The Purpose of this project is to develop a digital sign board system where the normal signs are displayed with their actual names. And also, to create awareness of the road safety to everyone and obey the traffic rules. To create a better view and warn in the night time.

2. Literature Survey:

2.1: Existing Problem:

- Damage criteria in static sign boards:
  - Paint deterioration
  - Folded Sheets
  - Pole Bent
  - Concealment by Vegetation
- Drivers can face confusing road signs at certain circumstances.
- The National Crime Records Bureau (NCRB) 2022 report states that there were 155,622 fatalities, highest since 2014, out of which 69,240 deaths were due to two-wheelers.
- A study by IIT Delhi points out that the national highways constitute only 2% of the length of roads in India, but they account for 30.3% of total road accidents and 36% of deaths.
- Deaths by accidents on roads increased by almost 17 per cent in 2021, indicating an increase in the rate of deaths per 1,000 vehicles.

S.No.	Title and Author	Year and Publication	Inferences
1.	**Wireless digital traffic signs of the future.** Chai K. Toh, Juan-Carlos Cano, Carlos Fernandez-Laguia, Pietro Manzoni, Carlos T. Calafate.	2018, Institution of Engineering and Technology (IET).	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.
2.	**Traffic Sign Board Detection and Recognition using Augmented**	2020, International Research Journal of	Real-time approach for fast and accurate

	Reality. Akshata Anant Prabhu, Deepika V.D. , Muralikrishna .N, P. Vaishnavi Acharya, A.R.Manjula	Engineering and Technology (IRJET).	framework for traffic sign recognition Which superimposes virtual objects onto a real scene under all types of driving situations, including unfavorable weather conditions and gives a voice alert with the help of speakers.
3.	**Automatic Signboard Detection System by the Vehicles** Anushree. A.S, Himanshu Kumar, Idah Iram, Kumar Divyam, Rajeshwari. J	2019, IJESC.	Signboard detection system in the vehicle which will detect the signboard and warn the driver about it. It displays the alert message or information on provided LCD and voice alert through speakers.
4.	**Development and Testing of Road Signs Alert System Using a Smart Mobile Phone** Eric M. Masatu, Ramadhani Sinde, and Anael Sam	2022, Hindawi Journal of Advanced Transportation).	The paper is based on the research about Advanced Driver Assistance system which is one of the salient features of intelligent system in transportation.
5.	**A Wi-Fi based Electronic Road Sign for Enhancing the Awareness of Vehicle.** A Bhawiyuga, R A Sabriansyah, W Yahya, R E Putra.	2016, IOP Publishing Ltd.	Employment of vehicular network concept in which a vehicle can communicate with other vehicles or with the infrastructure installed along the road.
6.	**Automatic Detection of Road Signs to Control Vehicle Speed** Anuja Nanal, Pooja Motwani,	2019, International Journal of Computer Applications.	Electronic Display controller meant for controlling vehicle speed and monitors the zones,

	Pragati Pawar, Rajat Nirhale, Rahul Patil.		and which can also display the speed to the rf reader with the help of unit attached in the car.

2.2. References

1. Torralba, J. P. García-Martín, J. M. González-Romo, M. García-Castellano, J. Peral-López and V. Pérez-Mira, "An Autonomous, Intelligent Sign Control System Using Wireless Communication and LED Signs for Rural and Suburban Roads," in IEEE Intelligent Transportation Systems Magazine, vol. 14, no. 2, pp. 115-128, March-April 2022, doi: 10.1109/MITS.2021.3049375.

2. Toh, C.K., Cano, J.-C., Fernandez-Laguia, C., Manzoni, P. and Calafate, C.T. (2019), Wireless digital traffic signs of the future. IET Netw., 8: 74-78. https://doi.org/10.1049/iet-net.2018.5127

3. A., Aparna & Shiravale, Sankirti. (2016). Real Time Traffic Signboard Detection and Recognition from Street Level Imagery for Smart Vehicle. International Journal of Computer Applications. 135. 18-22. 10.5120/ijca2016908267.

4. A Bhawiyuga R A Sabriansyah, W Yahya and R E Putra *et al* “A Wi-Fi based Electronic Road Sign for Enhancing the Awareness of Vehicle Driver”, in IOP Publishing Ltd 2017 *J. Phys.: Conf. Ser.* 801 012085

5. Karthikeyan D, Enitha C, Bharathi S, Durkadevi K, 2020, Traffic Sign Detection and Recognition using Image Processing, INTERNATIONAL JOURNAL OF

ENGINEERING RESEARCH & TECHNOLOGY (IJERT) NCICCT – 2020 (Volume 8 – Issue 08)

6. Bhawna Saini 1, Rachna Devi 2, Shilpi Dhankhar 3, Mohammad-ziaul-Haque 4, Jagandeep Kaur 5, Smart LED Display Boards, International Journal of Electronic and Electrical Engineering. ISSN 0974-2174 Volume 7, Number 10 (2014), pp. 1057-1067.

7. Ramalingam, Mritha & chandrasegar, & gowrishankar,. (2014). A survey of light emitting diode (LED) Display Board. Indian Journal of Science and Technology. 7. 185-188. 10.17485/ijst/2014/v7i2.3.

8. Eric M. Masatu, Ramadhani Sinde, Anael Sam, Development and Testing of Road Signs Alert System Using a Smart Mobile Phone, Journal of Advanced Transportation, 10.1155/2022/5829607, **2022**, (1-14), (2022).

9. Zoltán Fazekas, Gábor Balázs, Csaba Gyulai, Péter Potyondi, Péter Gáspár, Road-Type Detection Based on Traffic Sign and Lane Data, Journal of Advanced Transportation, 10.1155/2022/6766455, **2022**, (1-19), (2022).

10. Juanhong Xie, Guojian Shi, Weizhi Zhu, Intelligent Recognition Technology for the Segmentation of Traffic Indication Images Concerning Different Pavement Materials, Applied Bionics and Biomechanics, 10.1155/2022/6278240, **2022**, (1-7), (2022).

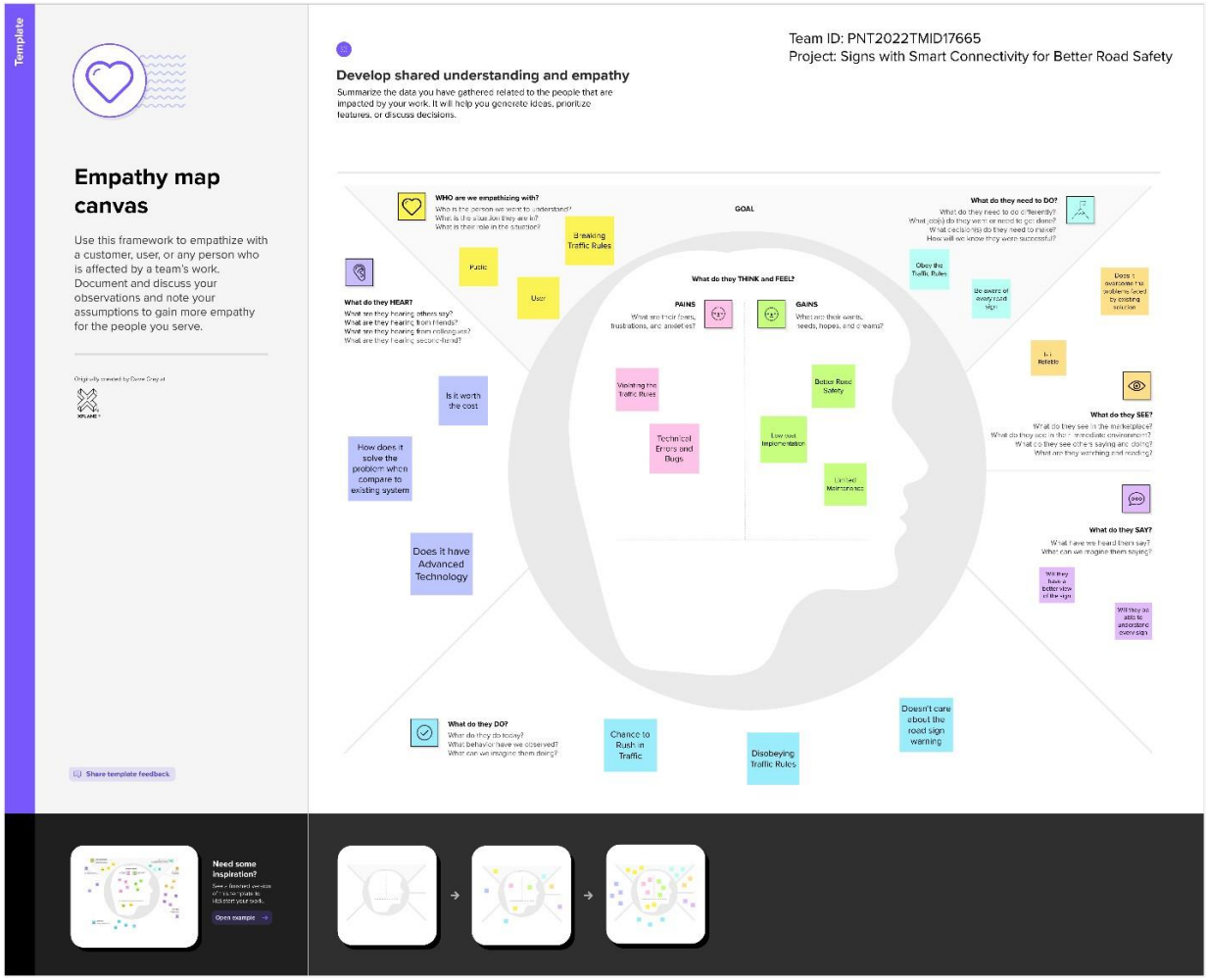
2.3. Problem Statement Definition:

To replace the static signboards, with smart connected digital sign boards. These smart connected sign boards get the speed limitations from weather API

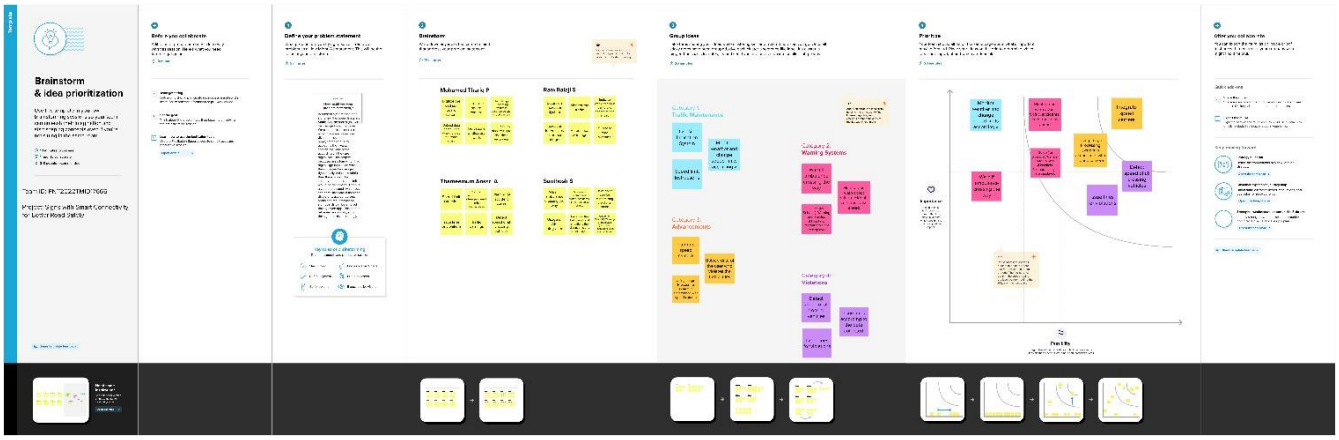
and update automatically. Based on the weather changes the speed may increase or decrease. It will display the normal signs in necessary places with wordings to be aware of the signs. Based on the traffic and fatal situations the diversion signs are displayed. Guide (Schools), Warning and Service (Hospitals, Restaurant) signs are also displayed accordingly. Change of modes will take place automatically.

3. Ideation and Proposed Solution:

3.1. Empathy Map Canvas:



3.2. Ideation & Brainstorming:



3.3. Proposed Solution:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	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.
2.	Idea / Solution description	The Idea is to replace the static signboards. Instead, 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 (for Schools), Warning and Service (Hospitals, Restaurant) signs are also displayed accordingly. Additionally, Speed camera integrated with image processing technique is added to detect any traffic speed violations and charge fines.
3.	Novelty / Uniqueness	Usage of speed camera integrated with Image Processing technique for detection of speed violation.
4.	Social Impact / Customer Satisfaction	Diversion Indication System if traffic or constructions ahead. Speed limit Instructions. Guide (for Schools), Warning and Service (Hospitals, Restaurant) signs are displayed.
5.	Business Model (Revenue Model)	Since Image Processing and APIs are used for monitor, this project employs a decent business strategy and enhance services.
6.	Scalability of the Solution	Low-cost Implementation and Maintenance. Durability is of the product is high.

3.4. Problem Solution Fit:

Project Title: Signs with smart Connectivity for Better Road Safety | **Project Design Phase-I - Solution Fit Template** | **Team ID:** PNT2022TMID17665

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS
Who is your customer?
i.e. working parents of 0-5 y.o. kids

The public who are violating traffic rules like speed violation,..

6. CUSTOMER CONSTRAINTS CC
What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.

If the public violate the traffic rules they will charged with fines.

5. AVAILABLE SOLUTIONS AS
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 these solutions have? i.e. pen and paper is an alternative to digital notetaking

Wireless sign board for the future, Static Sign Board System, Accurate Road Safety Level Assessment for Effective Road Safety, IoT-Driven Road Safety System.

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS J&P
Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides.

In present system, the road signs and the speed limits are static.

There are some cases when road diversions due to heavy traffic or accident zones or construction works ahead situations arise.

Monitoring traffic violations especially speed violations.

9. PROBLEM ROOT CAUSE RC
What is the real reason that this problem exists?
What is the back story behind the need to do
this job?
i.e. customers have to do it because of the change in regulations.

Increased number of Traffic violations day to day. The data shared by police showed that total 1,38,02,973 traffic violations were reported last year as compared to 1,05,80,249 in 2019.

7. BEHAVIOUR BE
What does your customer do to address the problem and get the job done?
i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

Obey the traffic rules.
Awareness of most traffic signs.

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS TR
What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.

Monitoring traffic violations and charging fines

4. EMOTIONS: BEFORE / AFTER EM
How do customers feel when they face a problem or a job and afterwards?
i.e. lost, insecure > confident, in control - use it in your communication strategy & design.

Violating the Traffic Rules.
Technical Errors and Bugs.

Better Road Safety.
Aware of most road signs.

10. YOUR SOLUTION SL
If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.
If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.

The Objective of this is to replace the static signboards. Instead, 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 (for Schools), Warning and Service (Hospitals, Restaurant) signs are also displayed accordingly.

8. CHANNELS of BEHAVIOUR CH
8.1 ONLINE
What kind of actions do customers take online? Extract online channels from #7

8.2 OFFLINE
What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

Data of the violated user will be uploaded to the cloud. The respected authorities monitor it regularly and take necessary actions

Identify strong TR & EM

4. Requirements:

4.1. Functional Requirement:

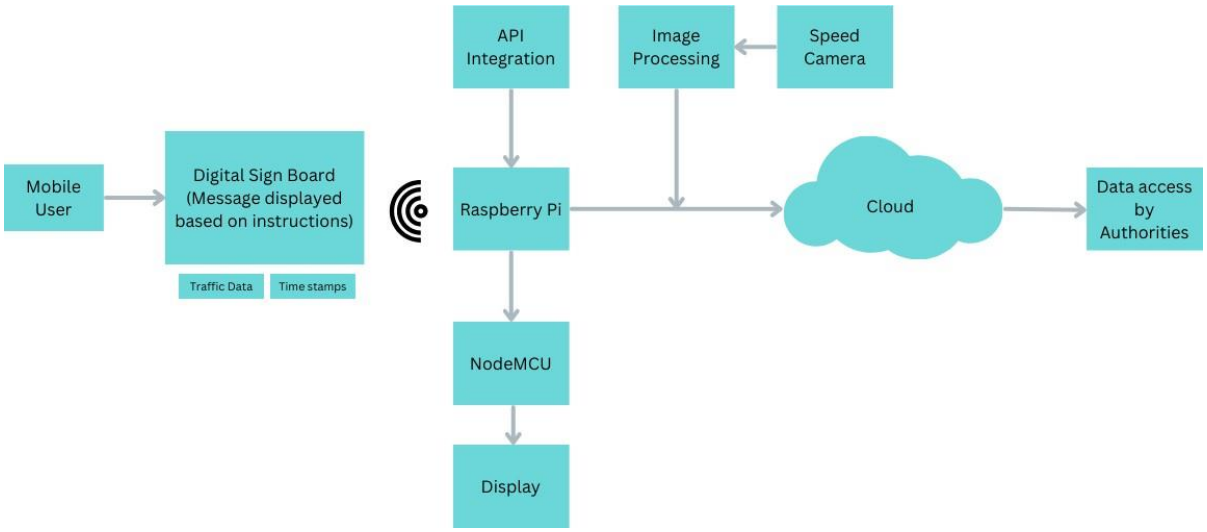
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Visibility	Sign Board will have and clear and interactive UI so that it will be clearly visible to all the users.
FR-2	User Understanding	The signs that are to be displayed in the sign board will be with its respective names, so that the users can clearly understand everything
FR-3	User Convenience	Signs will be displayed flawlessly such that it will be of better convenience.

4.2. Non-Functional Requirement:

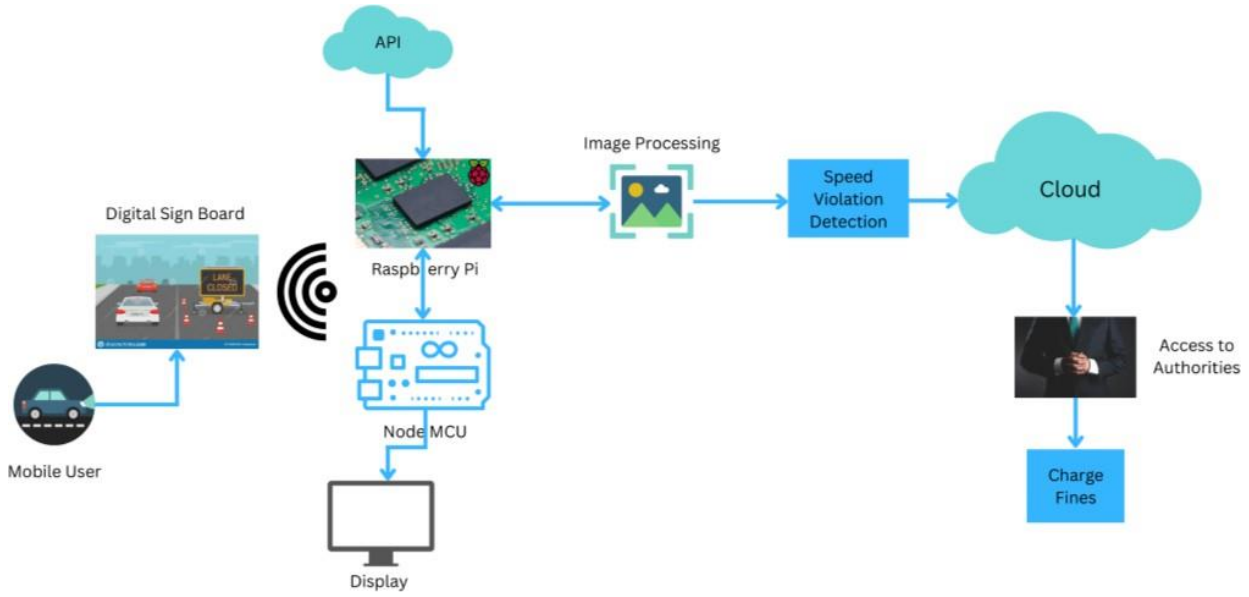
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	For multiple sign display, time stamps will be allocated for each sign. It will automatic and dynamically changeable. No need for manual operations.
NFR-2	**Security**	Only required can will be showed. No chance of security vulnerability.
NFR-3	**Reliability**	More reliable than the existing system
NFR-4	**Performance**	Acceptable performance with dynamic updating of data regarding weather, traffic, etc.
NFR-5	**Availability**	It will available for working every 24/7.
NFR-6	**Scalability**	Implementation and Maintenance cost will be less, so that the product is highly scalable.

5. Project Design:

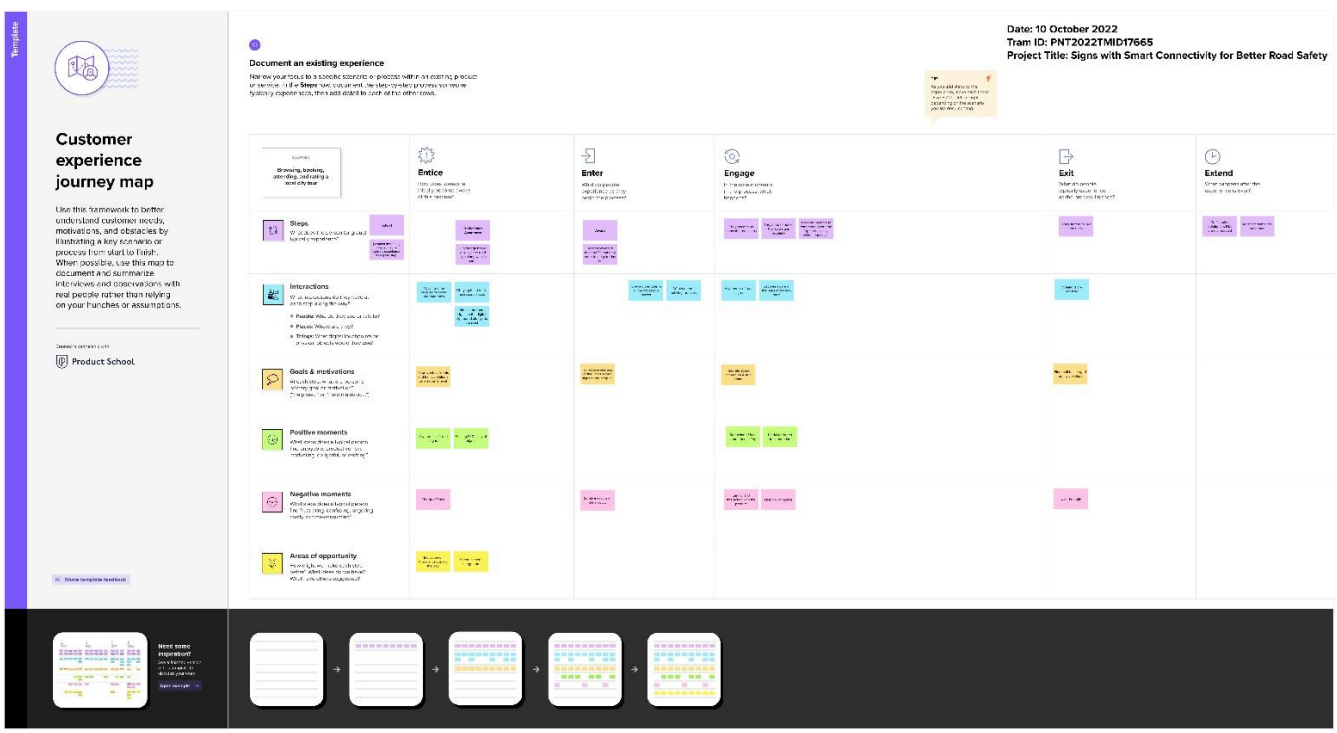
5.1. Data Flow Diagram:



5.2. Solution and Technical Architecture:



5.3. User Stories:



6. Project Planning & Scheduling:

6.1. Sprint Planning & Estimation:

Sprint	Functional Requirement (Epic)	User Story / Task	Story Points	Team Members
Sprint-1	User Registration	As a user, I can register on the website by entering my email, password, and confirming my password.	3	Ram Balaji S
	Admin Registration	As an admin, I can login to the website using my credentials and access the data.	3	Santhosh S
	Login	User and Admin can log into the website by entering email & password.	1	Thameemum Ansari A
	Dashboard	Develop a dashboard for the website for knowledge about road rules	3	Mohamed Thariq P
Sprint	**Functional Requirement (Epic)**	**User Story / Task**	**Story Points**	**Team Members**
Sprint-2	Node-Red UI	Develop a Node-Red UI Flow.	2	Ram Balaji S
	Node-Red Dashboard	Develop a Node-Red UI Dashboard.	2	Thameemum Ansari A
	Node-Red Webpage	Develop a Node-Red Webpage for displaying the data.	3	Mohamed Thariq P
	Node-Red Data Check	Check the data displayed on the Node-Red Dashboard UI.	3	Santhosh S

Sprint	Functional Requirement (Epic)	User Story / Task	Story Points	Team Members
Sprint-3	API Integration	Integrate the necessary API's.	3	Mohamed Thariq P
	Develop Python Code	Develop Python code to integrate the necessary APIs.	2	Ram Balaji S

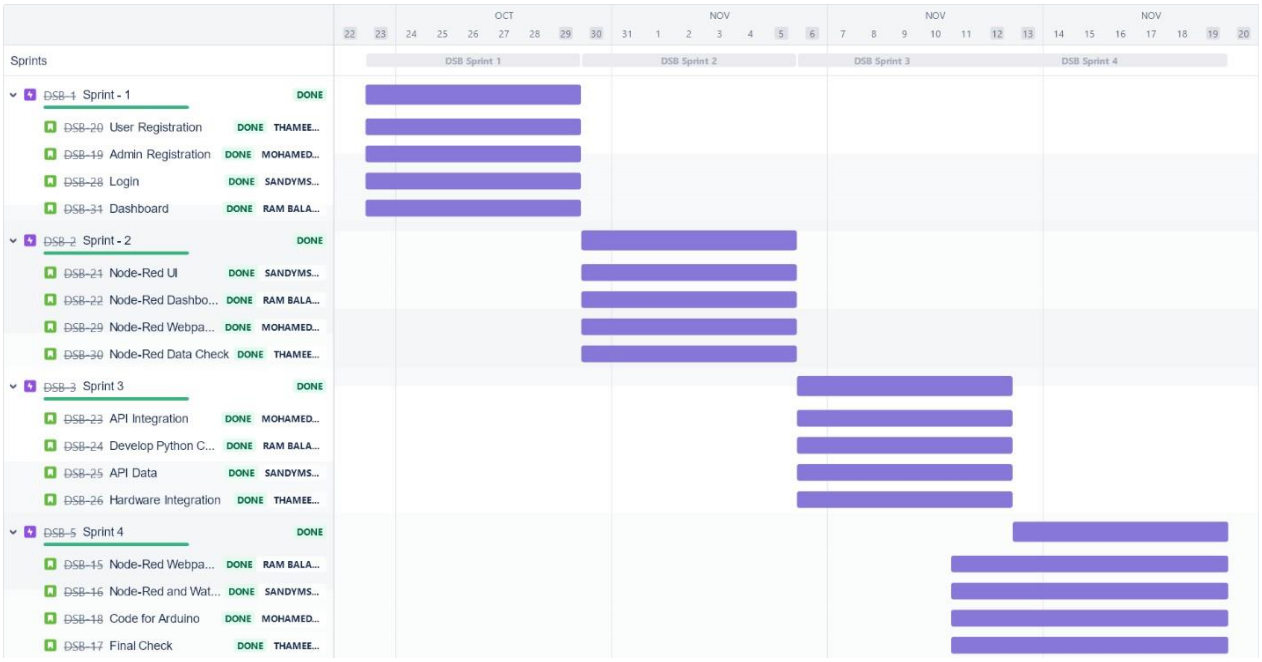
Sprint	Functional Requirement (Epic)	User Story / Task	Story Points	Team Members
	API Data	Check the data from weather API.	2	Santhosh S
	Hardware Integration	Integrate Arduino with TFT Display via simulation.	3	Thameemum Ansari A

Sprint	Functional Requirement (Epic)	User Story / Task	Sprint Points	Team Members
Sprint-4	Node-Red Webpage Data	Develop code to display data on the webpage and check the necessary.	2	Santhosh S
	Node-Red and Watson	Connect Node-Red with IBM Watson platform for data processing (Random Data Generation).	3	Mohamed Thariq P
	Code for Arduino	Develop code to display data in the display screen.	3	Thameemum Ansari A
	Final Check	Checking all the simulation and services working perfectly and display data and final submission of project.	2	Ram Balaji S

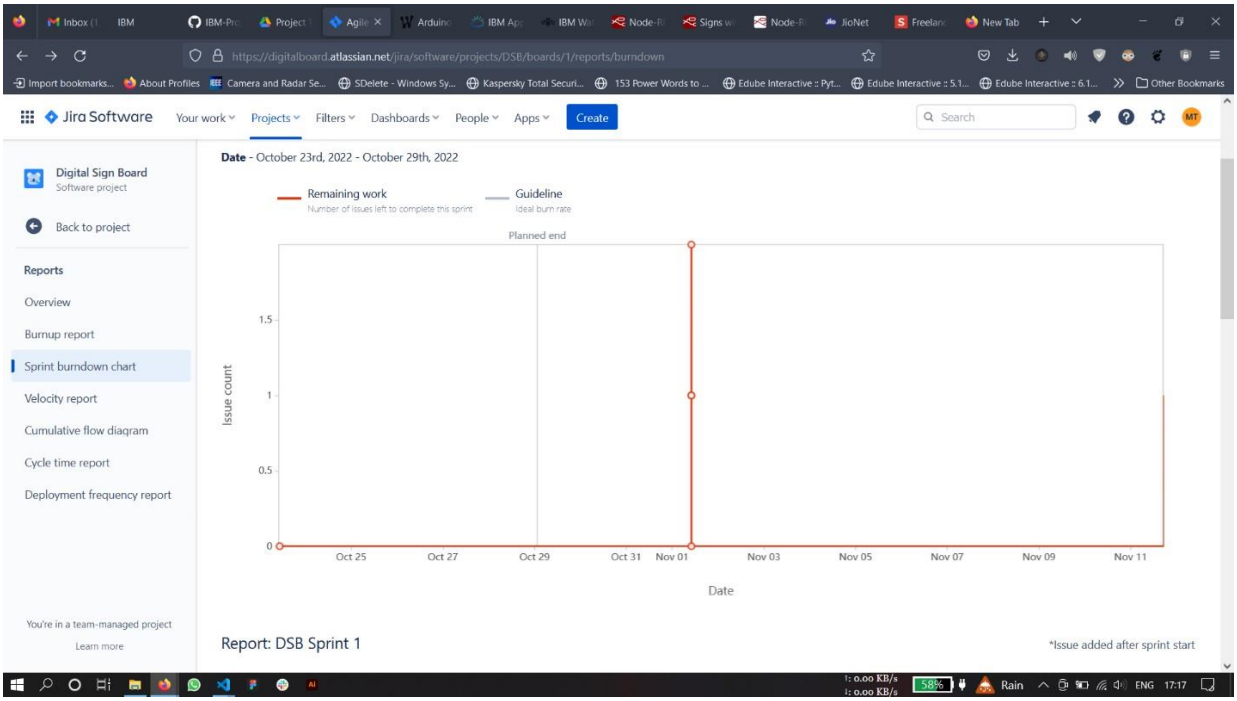
6.2. Sprint Delivery Schedule:

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	10	6 Days	24 Oct 2022	29 Oct 2022	10	29 Oct 2022
Sprint-2	10	6 Days	31 Oct 2022	05 Nov 2022	10	05 Nov 2022
Sprint-3	10	6 Days	07 Nov 2022	12 Nov 2022	10	12 Nov 2022
Sprint-4	10	6 Days	14 Nov 2022	19 Nov 2022	10	19 Nov 2022

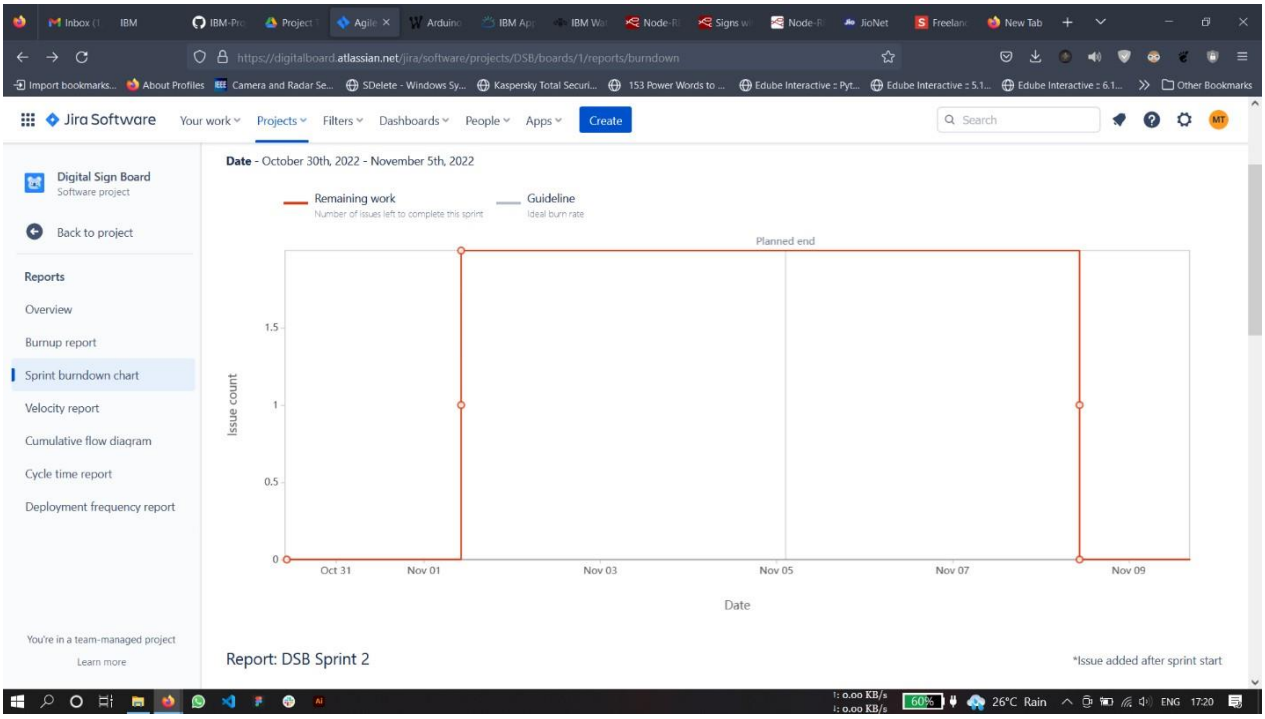
6.3. Report from Jira:



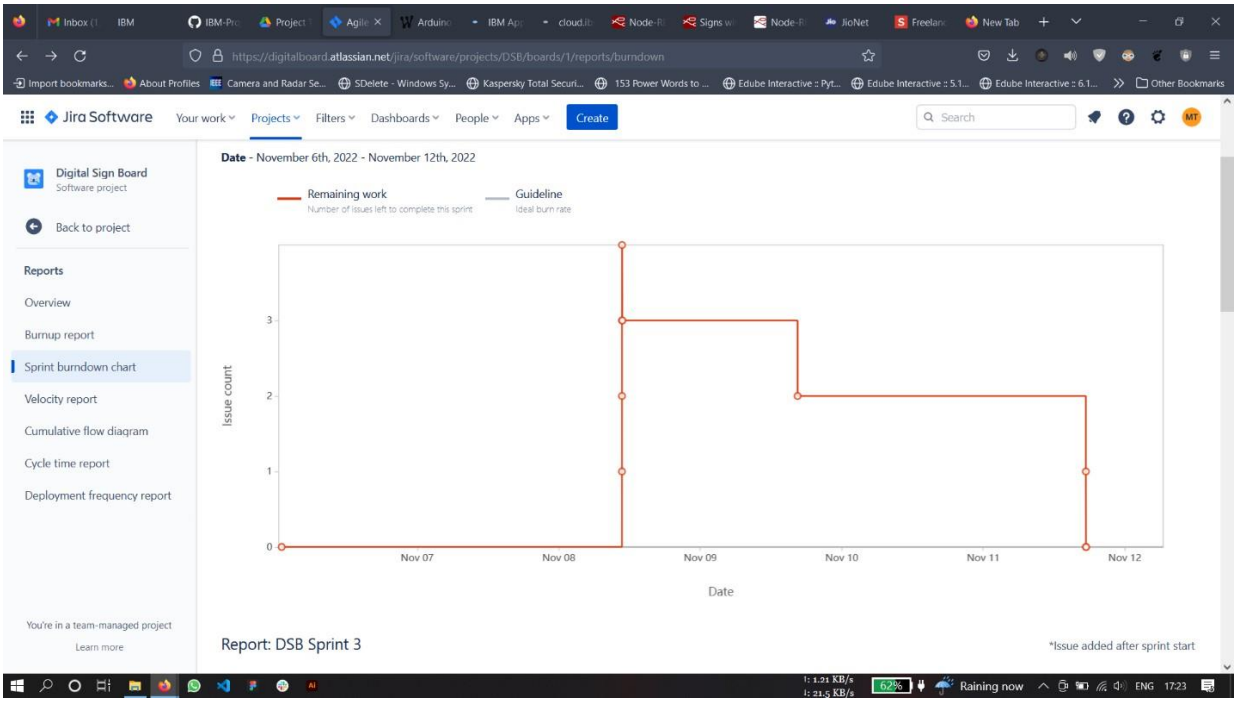
Sprint – 1 Burndown chart:



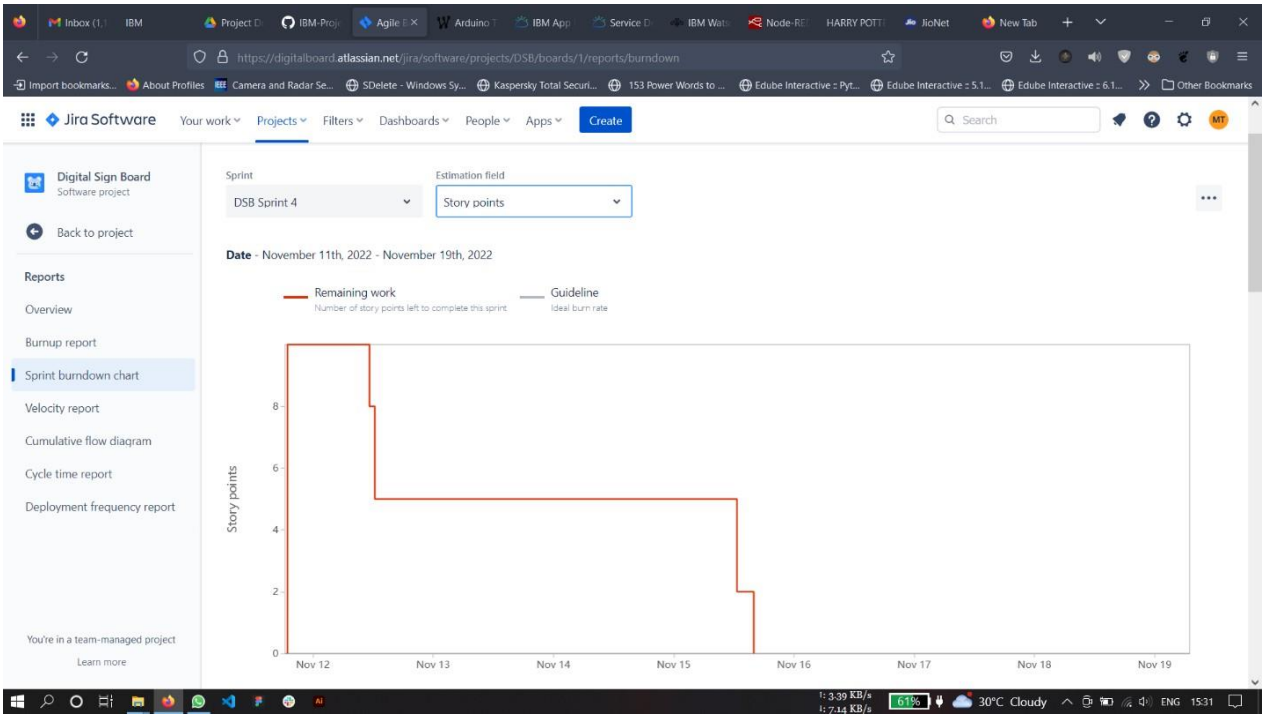
Sprint – 2 Burndown Chart:



Sprint – 3 Burndown Chart:



Sprint – 4 Burndown Chart:



7. Coding & Solutioning:

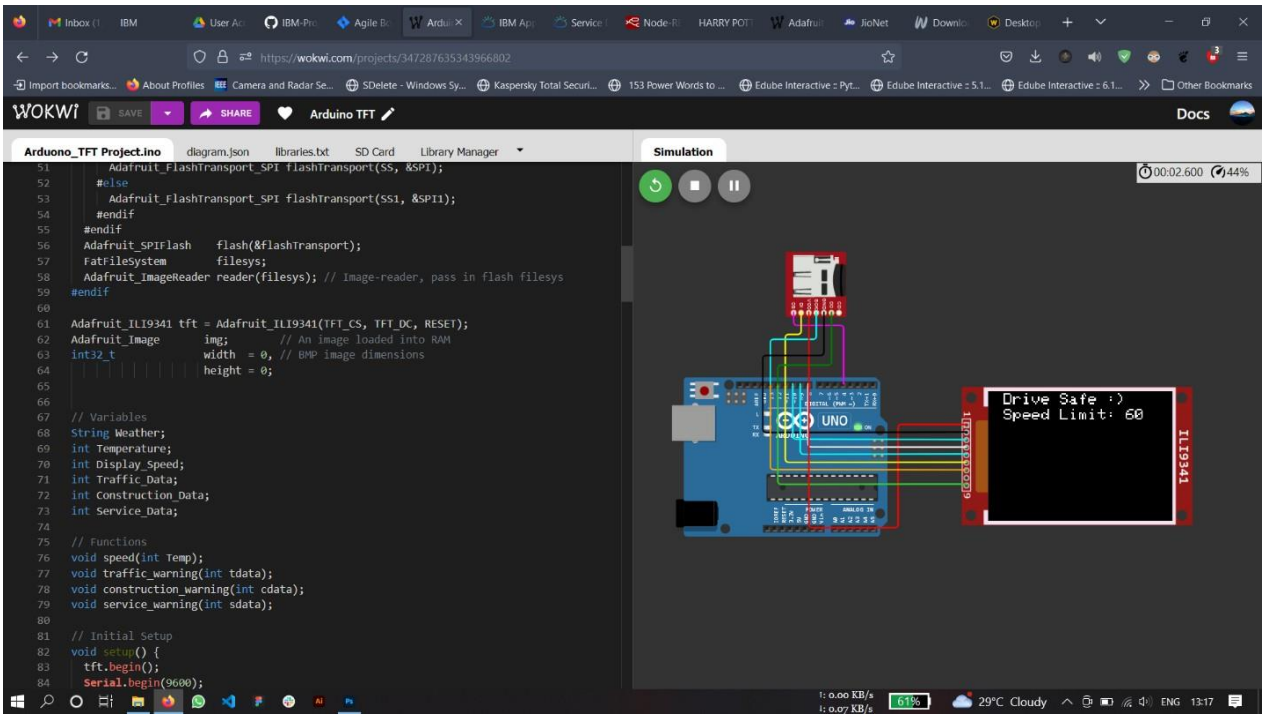
7.1. Feature 1:

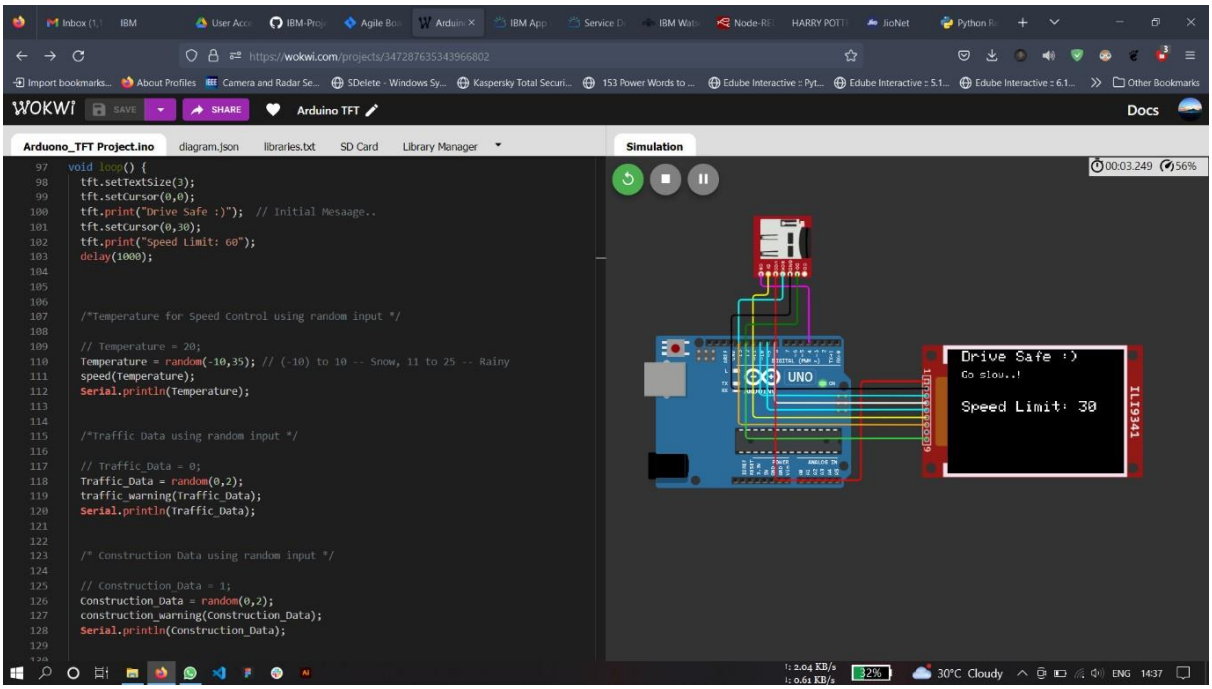
Climate prediction is done from temperature data from the opeanweatherapi. But as for now random values are used. Speed is increased or decreased based on the climate prediction.

```
  /*Temperature for Speed Control using random input */

  // Temperature = 20;
  Temperature = random(-10,35); // (-10) to 10 -- Snow, 11 to 25 -- Rainy
  speed(Temperature);
  Serial.println(Temperature);


/* Speed Control process.. */
void speed(int Temp)
{
  tft.fillScreen(Black);
  if( Temp >= -10 && Temp <= 14) // It's Snow
  {
```

```
    Weather = "Snowy";
    tft.setCursor(0,0);
    tft.print("Drive Safe :)");
    tft.setTextSize(2);
    tft.setCursor(0,40);
    tft.print("Go slow..!");
    tft.setCursor(0,100);
    tft.setTextSize(3);
    tft.print("Speed Limit: 30");
    delay(3000);
    tft.fillScreen(Black);
  }
  else if(Temp >= 15 && Temp <= 25) // It's Rainy
  {
    Weather = "Rainy";
    tft.setCursor(0,0);
    tft.print("Drive Safe :)");
    tft.setTextSize(2);
    tft.setCursor(0,40);
    tft.print("Slippery Road Ahead");
    tft.setCursor(0,70);
    tft.print("Go Slow..!");
    tft.setCursor(0,100);
    tft.setTextSize(3);
    tft.print("Speed Limit: 40");
    delay(3000);
    tft.fillScreen(Black);
  }
  else
  {
    tft.setCursor(0,0);
    tft.print("Drive Safe!!");
    tft.setCursor(0,30);
    tft.print("Speed Limit: 60");
    delay(3000);
    tft.fillScreen(Black);
  }
  tft.fillScreen(Black);
}


/* Traffic Warning System */

void traffic_warning(int tdata)
{
```

```
  tft.fillScreen(Black);
  if (tdata == 0)
  {
    tft.setCursor(0,0);
    tft.print("Drive Safe :)");
    // tft.setTextSize(2);
    tft.setCursor(0,40);
    tft.print("Traffic Ahead..");
    tft.setCursor(0,80);
    // tft.setTextSize(3);
    tft.print("Drive Carefully!");
    delay(3000);
    tft.fillScreen(Black);
  }
  if (tdata == 1)
  {
    tft.setCursor(0,0);
    tft.print("Drive Safe :)");
    tft.setTextSize(2);
    tft.setCursor(0,40);
    tft.print("Traffic Ahead..");
    tft.setCursor(0,80);
    // tft.setTextSize(3);
    tft.print("Take Diversion -->");
    delay(3000);
    tft.fillScreen(Black);
  }
}
```

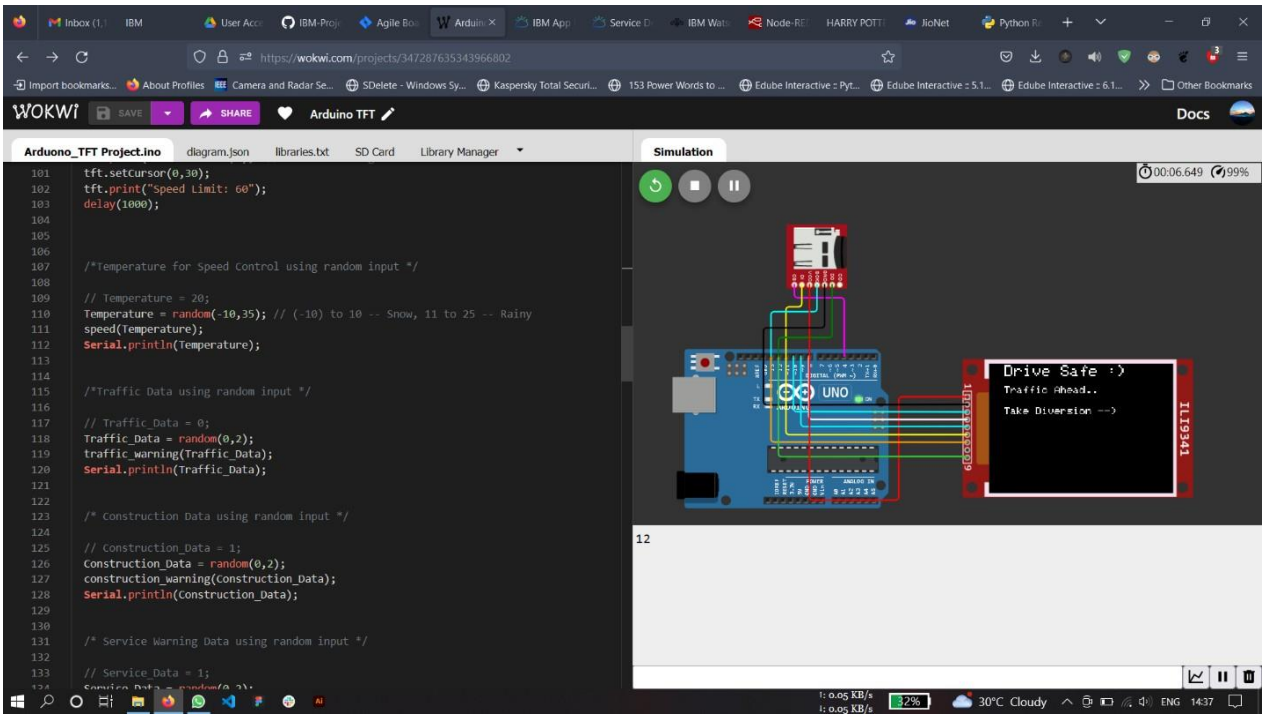


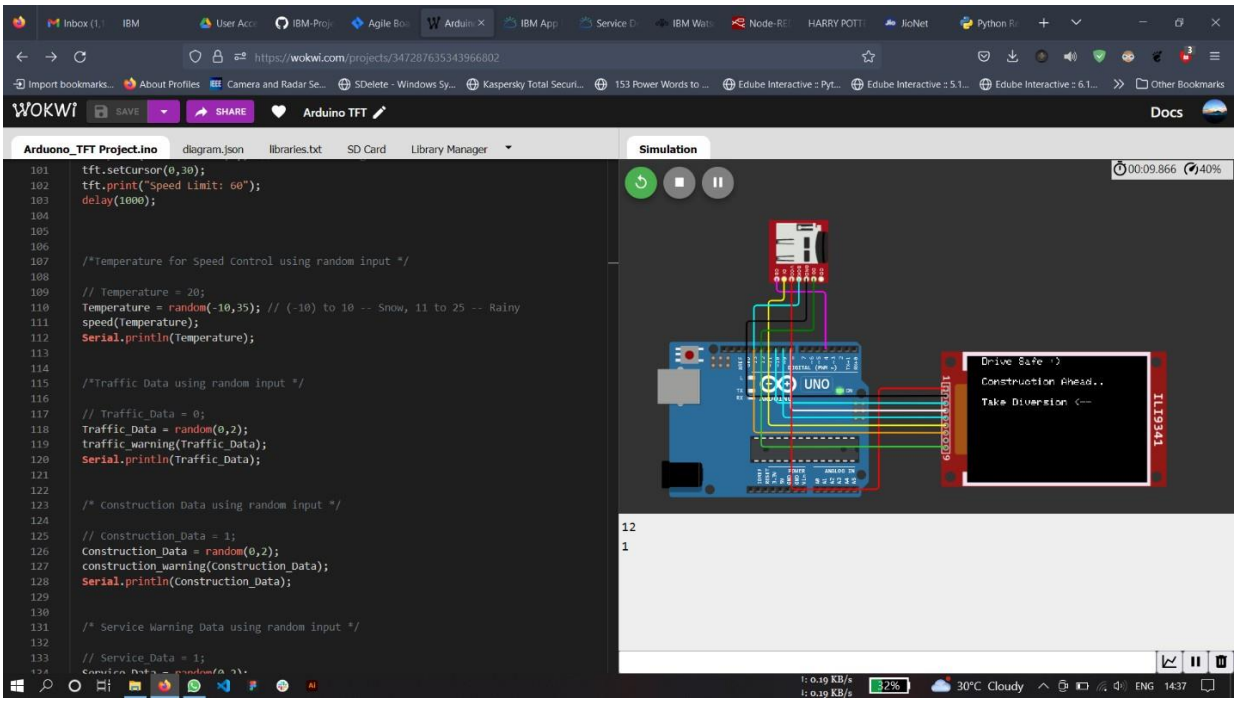


Traffic data and Construction warning data are given with random inputs. Based on traffic and construction data, warning are displayed.

```
  /*Traffic Data using random input */

  // Traffic_Data = 0;
  Traffic_Data = random(0,2);
  traffic_warning(Traffic_Data);
  Serial.println(Traffic_Data);


  /* Construction Data using random input */

  // Construction_Data = 1;
  Construction_Data = random(0,2);
  construction_warning(Construction_Data);
  Serial.println(Construction_Data);

/* Traffic Warning System */

void traffic_warning(int tdata)
{
  tft.fillScreen(Black);
  if (tdata == 0)
  {
    tft.setCursor(0,0);
    tft.print("Drive Safe :)");
    // tft.setTextSize(2);
    tft.setCursor(0,40);
    tft.print("Traffic Ahead..");
    tft.setCursor(0,80);
    // tft.setTextSize(3);
    tft.print("Drive Carefully!");
    delay(3000);
    tft.fillScreen(Black);
  }
  if (tdata == 1)
  {
    tft.setCursor(0,0);
    tft.print("Drive Safe :)");
    tft.setTextSize(2);
    tft.setCursor(0,40);
    tft.print("Traffic Ahead..");
    tft.setCursor(0,80);
    // tft.setTextSize(3);
    tft.print("Take Diversion -->");
    delay(3000);
    tft.fillScreen(Black);
```

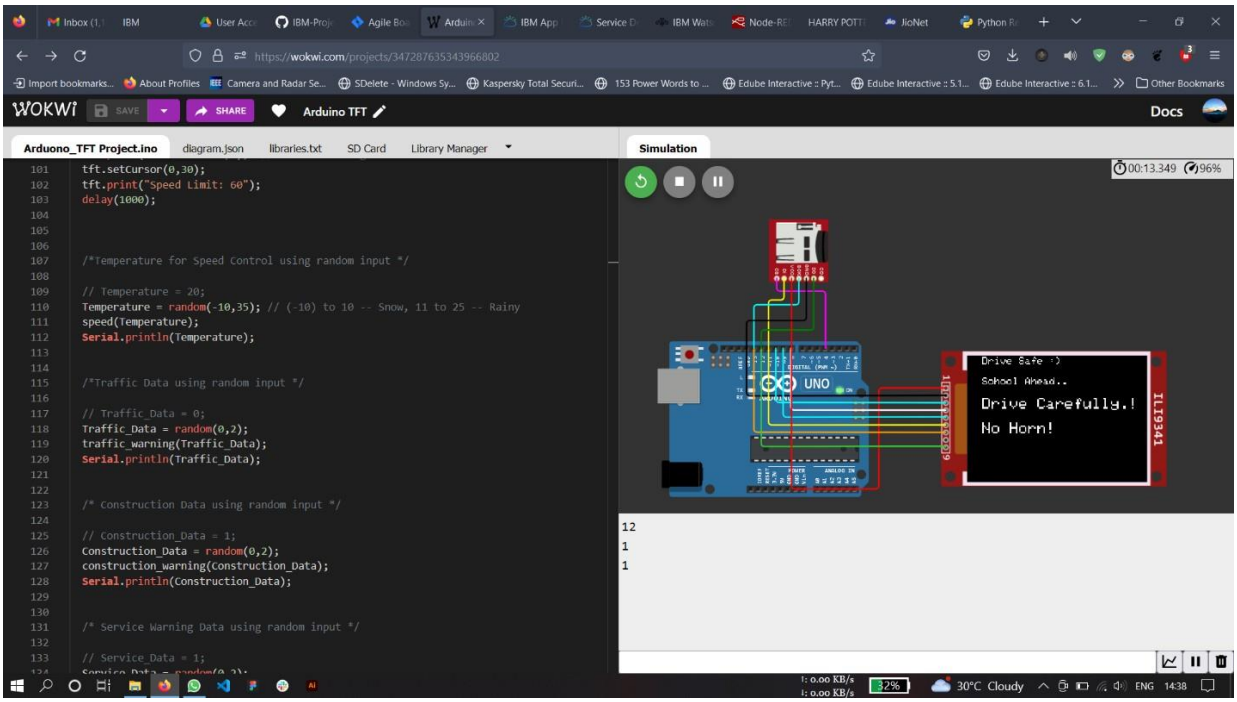
```
  }
}


/* Construction Warning System */

void construction_warning(int cdata)
{
  tft.fillScreen(Black);
  if (cdata == 0)
  {
    tft.setCursor(0,0);
    tft.print("Drive Safe :)");
    tft.setTextSize(2);
    tft.setCursor(0,40);
    tft.print("Construction Ahead..");
    tft.setCursor(0,80);
    tft.setTextSize(3);
    tft.print("Drive Carefully..!");
    delay(2000);
    tft.fillScreen(Black);
  }
  if (cdata == 1)
  {
    tft.setCursor(0,0);
    tft.print("Drive Safe :)");
    tft.setTextSize(2.5);
    tft.setCursor(0,40);
    tft.print("Construction Ahead..");
    tft.setCursor(0,80);
    tft.setTextSize(2.5);
    tft.print("Take Diversion <--");
    delay(2000);
    tft.fillScreen(Black);
  }
}
```


https://wokwi.com/projects/347287635343966802
WOKWI
SAVE
SHARE
Arduino TFT
Docs
Arduono_TFT Project.ino
diagram.json
libraries.txt
SD Card
Library Manager
Simulation
00:06.649
99%
tft.setCursor(0,30);
tft.print("Speed Limit: 60");
delay(1000);
/*Temperature for Speed Control using random input */
// Temperature = 20;
Temperature = random(-10,35); // (-10) to 10 -- Snow, 11 to 25 -- Rainy
speed(Temperature);
Serial.println(Temperature);
/*Traffic Data using random input */
// Traffic_Data = 0;
Traffic_Data = random(0,2);
traffic_warning(Traffic_Data);
Serial.println(Traffic_Data);
/* Construction Data using random input */
// Construction_Data = 1;
Construction_Data = random(0,2);
construction_warning(Construction_Data);
Serial.println(Construction_Data);
/* Service Warning Data using random input */
// Service_Data = 1;
UNO
Drive Safe :)
Traffic Ahead..
Take Diversion --)
ILI9341
12


https://wokwi.com/projects/347287635343966802
WOKWI
SAVE
SHARE
Arduino TFT
Docs
Arduono_TFT Project.ino
diagram.json
libraries.txt
SD Card
Library Manager
Simulation
00:09.866
40%
tft.setCursor(0,30);
tft.print("Speed Limit: 60");
delay(1000);
/*Temperature for Speed Control using random input */
// Temperature = 20;
Temperature = random(-10,35); // (-10) to 10 -- Snow, 11 to 25 -- Rainy
speed(Temperature);
Serial.println(Temperature);
/*Traffic Data using random input */
// Traffic_Data = 0;
Traffic_Data = random(0,2);
traffic_warning(Traffic_Data);
Serial.println(Traffic_Data);
/* Construction Data using random input */
// Construction_Data = 1;
Construction_Data = random(0,2);
construction_warning(Construction_Data);
Serial.println(Construction_Data);
/* Service Warning Data using random input */
// Service_Data = 1;
UNO
Drive Safe :)
Construction Ahead..
Take Diversion <--
ILI9341
12
1

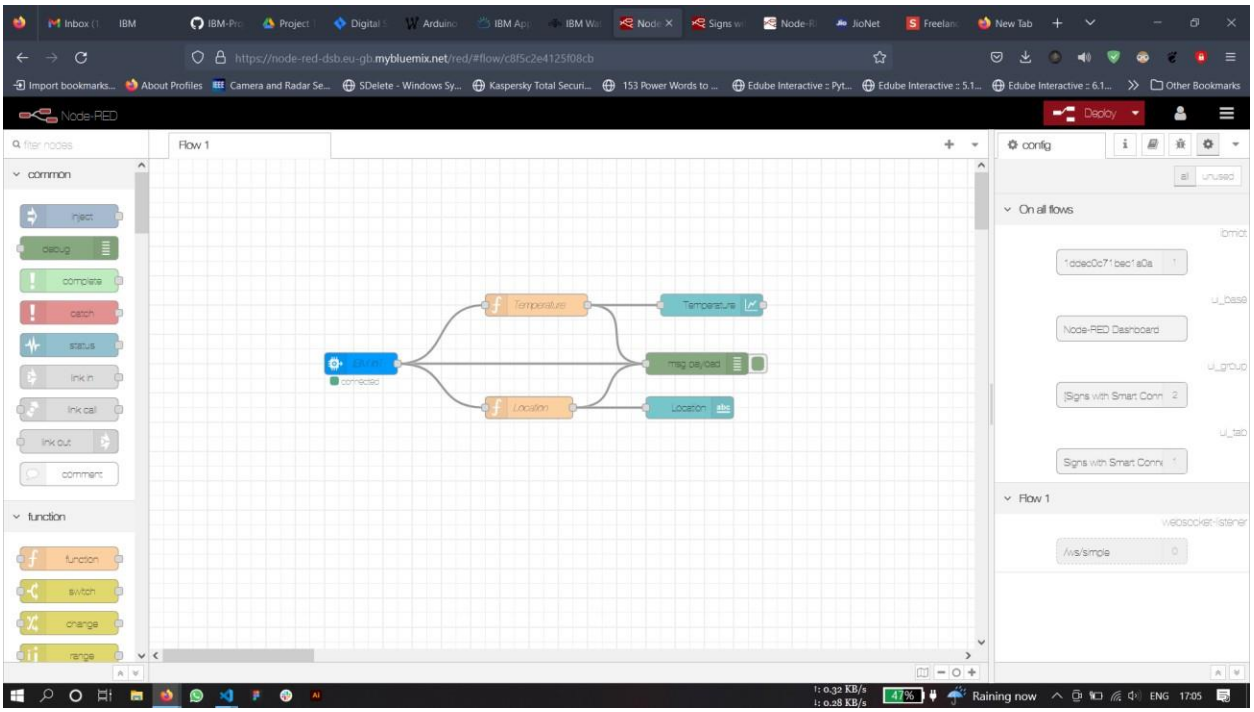
Service warnings like schools, hospitals and holy places are displayed.

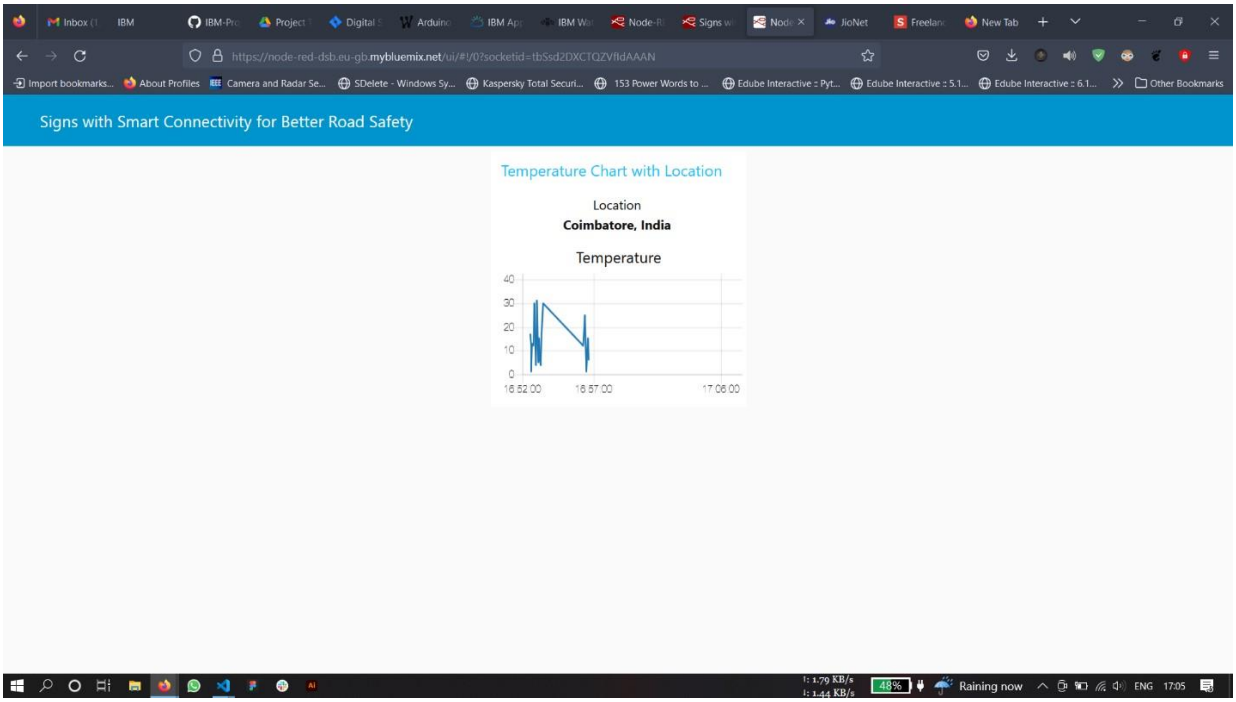
```
  /* Service Warning Data using random input */

  // Service_Data = 1;
  Service_Data = random(0,2);
  service_warning(Service_Data);
  Serial.println(Service_Data);

/* School, Hospital Warning System */

void service_warning(int sdata)
{
  tft.fillScreen(Black);
  if (sdata == 0)
  {
    tft.setCursor(0,0);
    tft.print("Drive Safe :)");
    tft.setTextSize(2);
    tft.setCursor(0,40);
    tft.print("School Ahead..");
    tft.setCursor(0,80);
    tft.setTextSize(3);
    tft.print("Drive Carefully.! \nNo Horn!");
    delay(2000);
    tft.fillScreen(Black);
  }
  if (sdata == 1)
  {
    tft.setCursor(0,0);
    tft.print("Drive Safe :)");
    tft.setTextSize(2.5);
    tft.setCursor(0,40);
    tft.print("Hospital Ahead..");
    tft.setCursor(0,80);
    tft.setTextSize(2.5);
    tft.print("Drive Carefully.! \nNo Horn!");
    delay(2000);
    tft.fillScreen(Black);
  }
}
```

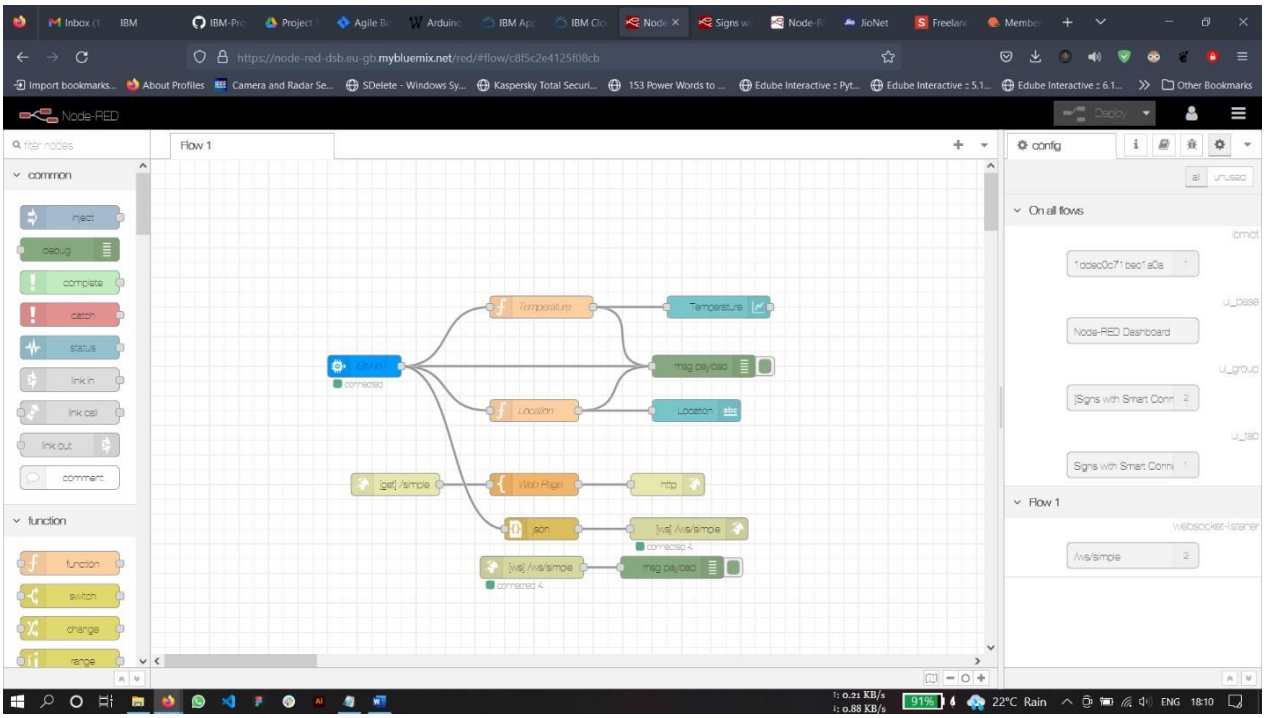

https://wokwi.com/projects/347287635343966802
WOKWI
SAVE
SHARE
Arduino TFT
Docs
Arduono_TFT Project.ino
diagram.json
libraries.txt
SD Card
Library Manager
Simulation
00:13.349
96%
tft.setCursor(0,30);
tft.print("Speed Limit: 60");
delay(1000);
/*Temperature for Speed Control using random input */
// Temperature = 20;
Temperature = random(-10,35); // (-10) to 10 -- Snow, 11 to 25 -- Rainy
speed(Temperature);
Serial.println(Temperature);
/*Traffic Data using random input */
// Traffic_Data = 0;
Traffic_Data = random(0,2);
traffic_warning(Traffic_Data);
Serial.println(Traffic_Data);
/* Construction Data using random input */
// Construction_Data = 1;
Construction_Data = random(0,2);
construction_warning(Construction_Data);
Serial.println(Construction_Data);
/* Service Warning Data using random input */
// Service_Data = 1;
UNO
Drive Safe :)
School Ahead..
Drive Carefully.!
No Horn!
ILI9341
12
1
1
30°C Cloudy
ENG
14:38

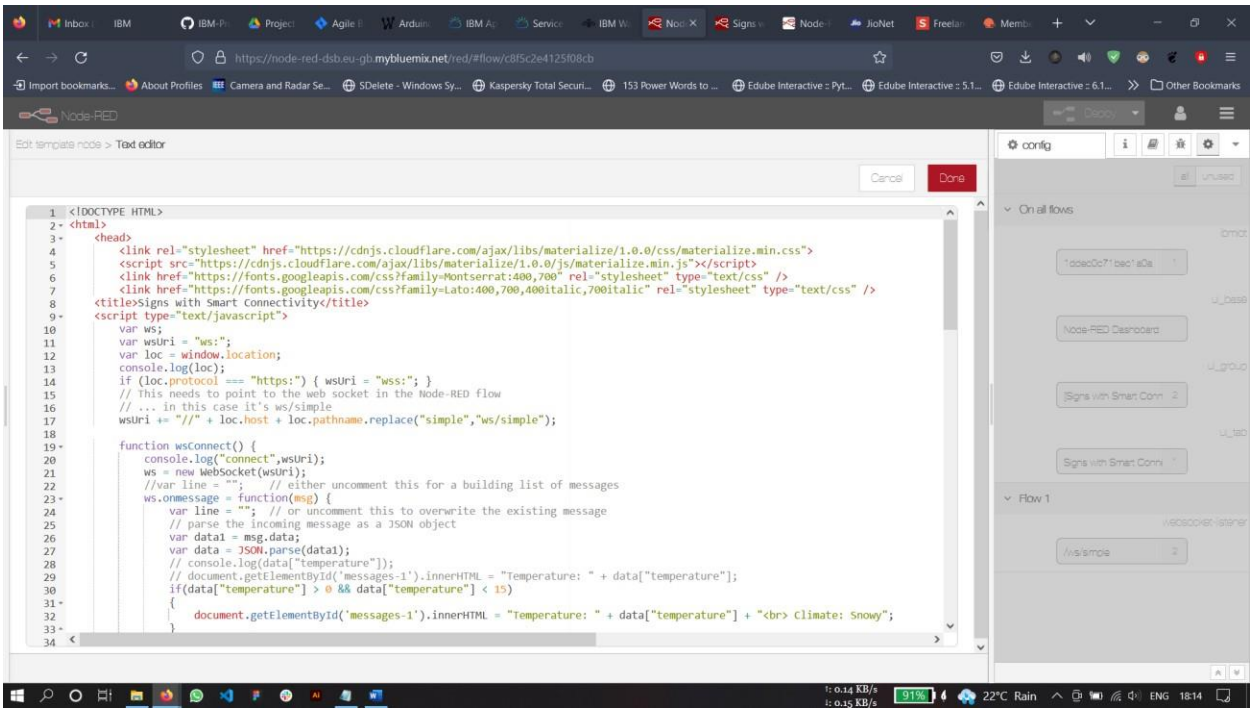
7.2. Feature 2:

The temperature and the location data are exactly displayed in the webpage using Node – Red and the for that is randomized using IBM Watson. A device is created for that purpose and is simulated to send data to node – red.

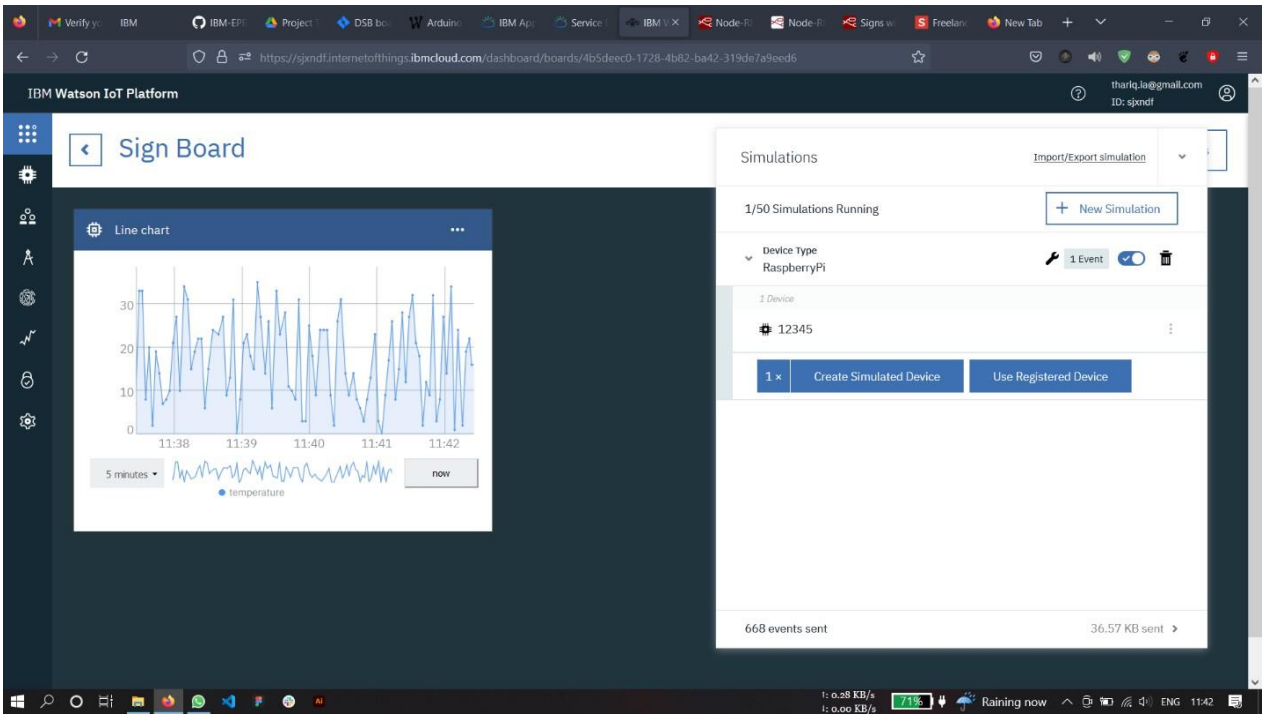


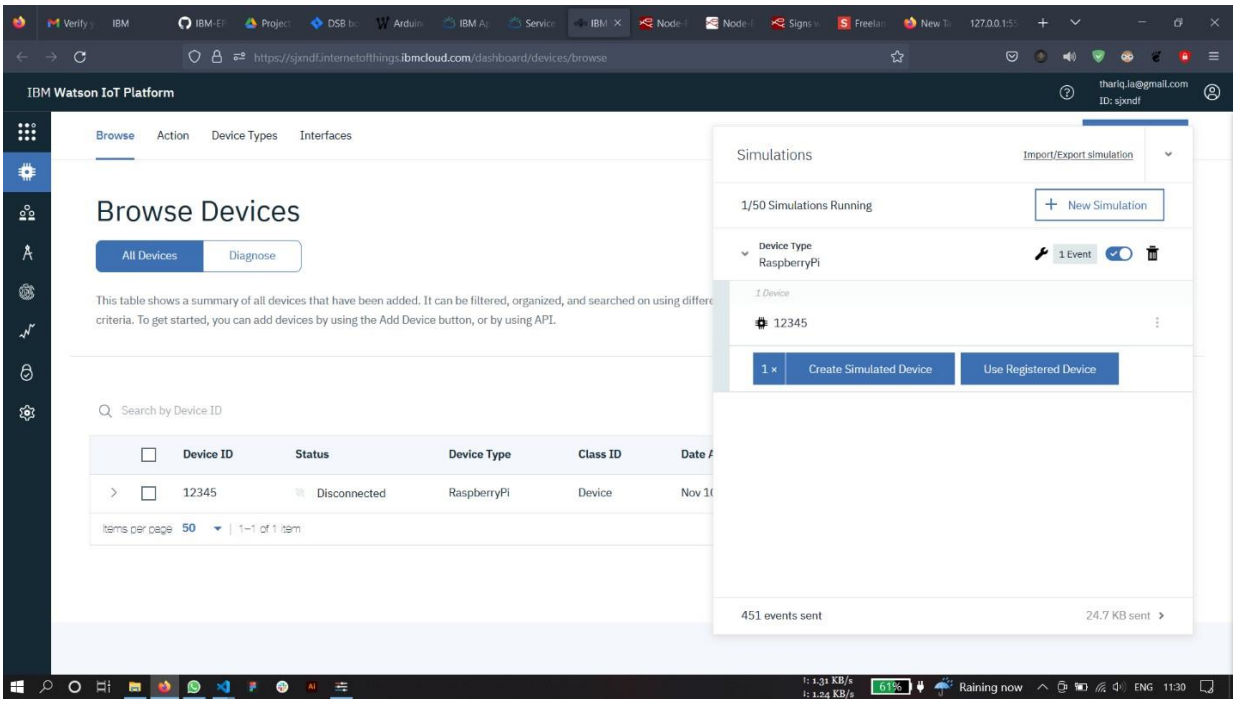


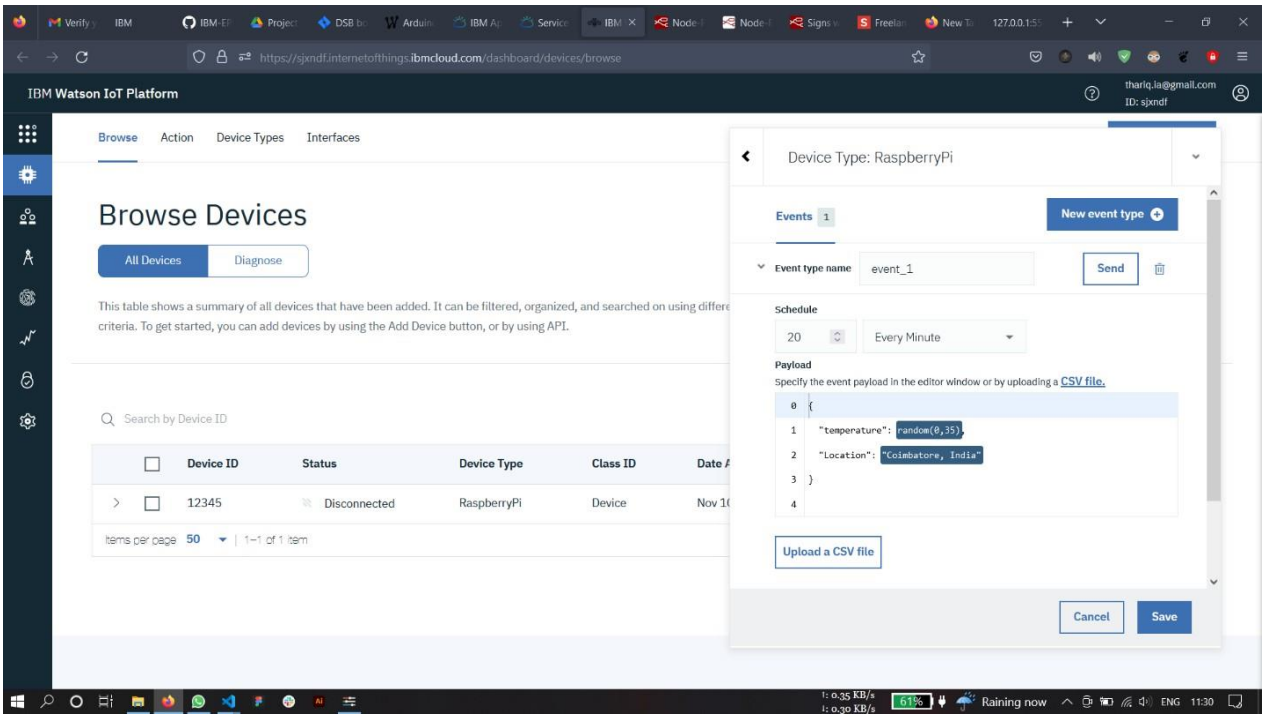





```
<!DOCTYPE HTML>
<html>
    <head>
        <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css/materialize.min.css">
        <script src="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/js/materialize.min.js"></script>
        <link href="https://fonts.googleapis.com/css?family=Montserrat:400,700" rel="stylesheet" type="text/css" />
        <link href="https://fonts.googleapis.com/css?family=Lato:400,700,400italic,700italic" rel="stylesheet" type="text/css" />
    <title>Signs with Smart Connectivity</title>
    <script type="text/javascript">
        var ws;
        var wsUri = "ws:";
        var loc = window.location;
        console.log(loc);
        if (loc.protocol === "https:") { wsUri = "wss:"; }
        // This needs to point to the web socket in the Node-RED flow
        // ... in this case it's ws/simple
        wsUri += "//" + loc.host + loc.pathname.replace("simple","ws/simple");

        function wsConnect() {
            console.log("connect",wsUri);
            ws = new WebSocket(wsUri);
            //var line = "";    // either uncomment this for a building list of messages
            ws.onmessage = function(msg) {
                var line = "";  // or uncomment this to overwrite the existing message
                // parse the incoming message as a JSON object
                var data1 = msg.data;
                var data = JSON.parse(data1);
                // console.log(data["temperature"]);
                // document.getElementById('messages-1').innerHTML = "Temperature: " + data["temperature"];
                if(data["temperature"] > 0 && data["temperature"] < 15)
                {
                    document.getElementById('messages-1').innerHTML = "Temperature: " + data["temperature"] + "<br> Climate: Snowy";
                }
```


IBM Watson IoT Platform
Sign Board
Line chart
30
20
10
0
11:38
11:39
11:40
11:41
11:42
5 minutes
now
temperature
Simulations
Import/Export simulation
1/50 Simulations Running
New Simulation
Device Type
RaspberryPi
1 Event
1 Device
12345
1 ×
Create Simulated Device
Use Registered Device
668 events sent
36.57 KB sent


IBM Watson IoT Platform
Browse
Action
Device Types
Interfaces
Browse Devices
All Devices
Diagnose
This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using differe
criteria. To get started, you can add devices by using the Add Device button, or by using API.
Search by Device ID
Device ID
Status
Device Type
Class ID
12345
Disconnected
RaspberryPi
Device
Items per page 50
1–1 of 1 item
Simulations
Import/Export simulation
1/50 Simulations Running
New Simulation
Device Type
RaspberryPi
1 Event
1 Device
12345
1 ×
Create Simulated Device
Use Registered Device
451 events sent
24.7 KB sent



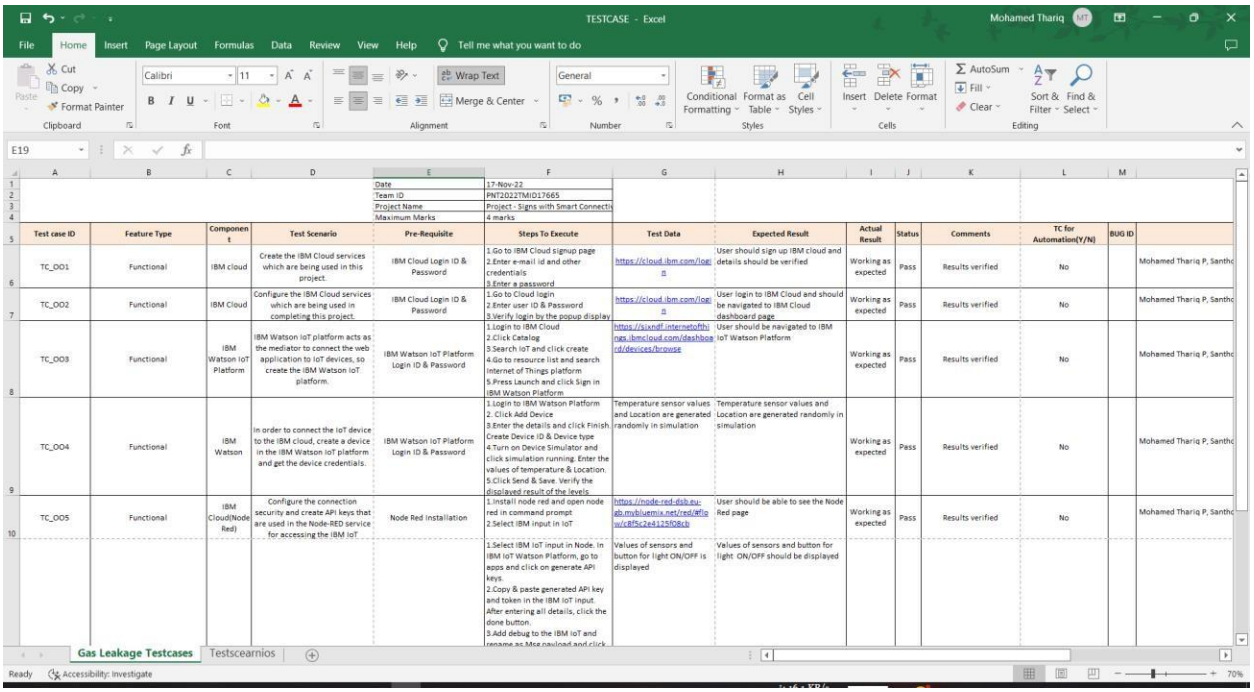
The following features are some ideas we decided to implement but we didn’t had much time...

Additionally, a speed cam will be integrated with the digital sign board which use Image processing & AI, to get the details of the driver who breaks the traffic rules (especially speed) will be updated in the cloud database.

Also, for No parking and One way rule violations can also be detected and appropriate action can be taken. Violations of stop signs in intersection will also be detected using AI.

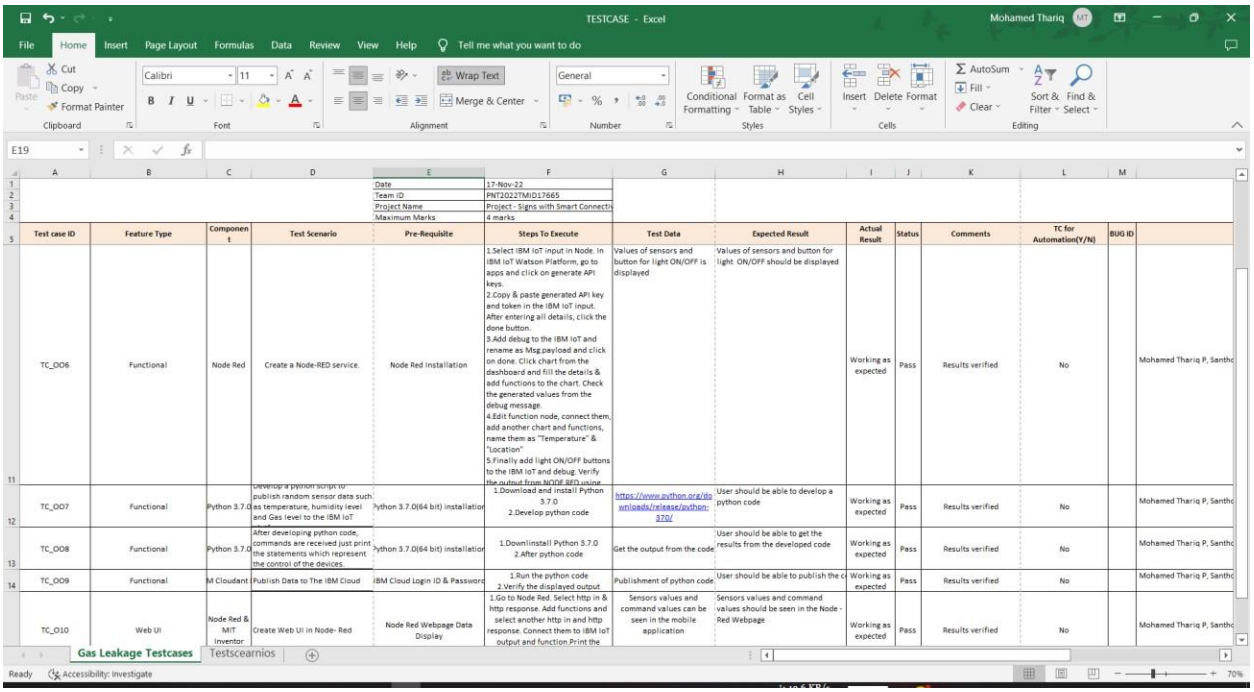
8. Testing:

8.1. Test Cases:



Date	17-Nov-22
Team ID	PNT2022TMID17665
Project Name	Project - Signs with Smart Connectiv
Maximum Marks	4 marks

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	TC for Automation(Y/N)	BUG ID	
TC_OO1	Functional	IBM cloud	Create the IBM Cloud services which are being used in this project.	IBM Cloud Login ID & Password	1.Go to IBM Cloud signup page 2.Enter e-mail id and other credentials 3.Enter a password	https://cloud.ibm.com/login	User should sign up IBM cloud and details should be verified	Working as expected	Pass	Results verified	No		Mohamed Thariq P, Santho
TC_OO2	Functional	IBM Cloud	Configure the IBM Cloud services which are being used in completing this project.	IBM Cloud Login ID & Password	1.Go to Cloud login 2.Enter user ID & Passsword 3.Verify login by the popup display	https://cloud.ibm.com/login	User login to IBM Cloud and should be navigated to IBM Cloud dashboard page	Working as expected	Pass	Results verified	No		Mohamed Thariq P, Santho
TC_OO3	Functional	IBM Watson IoT Platform	IBM Watson IoT platform acts as the mediator to connect the web application to IoT devices, so create the IBM Watson IoT platform.	IBM Watson IoT Platform Login ID & Password	1.Login to IBM Cloud 2.Click Catalog 3.Search IoT and click create 4.Go to resource list and search Internet of Things platform 5.Press Launch and click Sign in IBM Watson Platform	https://sixndf.internetofthings.ibmcloud.com/dashboard/devices/browse	User should be navigated to IBM IoT Watson Platform	Working as expected	Pass	Results verified	No		Mohamed Thariq P, Santho
TC_OO4	Functional	IBM Watson	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.	IBM Watson IoT Platform Login ID & Password	1.Login to IBM Watson Platform 2. Click Add Device 3.Enter the details and click Finish. Create Device ID & Device type 4.Turn on Device Simulator and click simulation running. Enter the values of temperature & Location. 5.Click Send & Save. Verify the displayed result of the levels.	Temperature sensor values and Location are generated randomly in simulation	Temperature sensor values and Location are generated randomly in simulation	Working as expected	Pass	Results verified	No		Mohamed Thariq P, Santho
TC_OO5	Functional	IBM Cloud(Node Red)	Configure the connection security and create API keys that are used in the Node-RED service for accessing the IBM IoT	Node Red Installation	1.Install node red and open node red in command prompt 2.Select IBM input in IoT	https://node-red-dsb.eu-gb.mybluemix.net/red/#flow/c8f5c2e4125f08cb	User should be able to see the Node Red page	Working as expected	Pass	Results verified	No		Mohamed Thariq P, Santho
					1.Select IBM IoT input in Node. In IBM IoT Watson Platform, go to apps and click on generate API keys. 2.Copy & paste generated API key and token in the IBM IoT input. After entering all details, click the done button. 3.Add debug to the IBM IoT and rename as Msg.payload and click	Values of sensors and button for light ON/OFF is displayed	Values of sensors and button for light ON/OFF should be displayed						



Date	17-Nov-22
Team ID	PNT2022TMID17665
Project Name	Project - Signs with Smart Connectiv
Maximum Marks	4 marks

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	TC for Automation(Y/N)	BUG ID	
					1.Select IBM IoT input in Node. In IBM IoT Watson Platform, go to apps and click on generate API keys. 2.Copy & paste generated API key and token in the IBM IoT input. After entering all details, click the done button. 3.Add debug to the IBM IoT and rename as Msg.payload and click on done. Click chart from the dashboard and fill the details & add functions to the chart. Check the generated values from the debug message. 4.Edit function node, connect them, add another chart and functions, name them as "Temperature" & "Location" 5.Finally add light ON/OFF buttons to the IBM IoT and debug. Verify	Values of sensors and button for light ON/OFF is displayed	Values of sensors and button for light ON/OFF should be displayed						
TC_OO6	Functional	Node Red	Create a Node-RED service.	Node Red Installation	the output from NODE RED using			Working as expected	Pass	Results verified	No		Mohamed Thariq P, Santho
TC_OO7	Functional	Python 3.7.0	Develop a python script to publish random sensor data such as temperature, humidity level and Gas level to the IBM IoT	Python 3.7.0(64 bit) installation	1.Download and install Python 3.7.0 2.Develop python code	https://www.python.org/downloads/release/python-370/	User should be able to develop a python code	Working as expected	Pass	Results verified	No		Mohamed Thariq P, Santho
TC_OO8	Functional	Python 3.7.0	After developing python code, commands are received just print the statements which represent the control of the devices.	Python 3.7.0(64 bit) installation	1.Downlinstall Python 3.7.0 2.After python code	Get the output from the code	User should be able to get the results from the developed code	Working as expected	Pass	Results verified	No		Mohamed Thariq P, Santho
TC_OO9	Functional	M Cloudant	Publish Data to The IBM Cloud	IBM Cloud Login ID & Passwor	1.Run the python code 2.Verify the displayed output	Publishment of python code	User should be able to publish the c	Working as expected	Pass	Results verified	No		Mohamed Thariq P, Santho
TC_O10	Web UI	Node Red & MIT Inventor	Create Web UI in Node- Red	Node Red Webpage Data Display	1.Go to Node Red. Select http in & http response. Add functions and select another http in and http response. Connect them to IBM IoT output and function.Print the	Sensors values and command values can be seen in the mobile application	Sensors values and command values should be seen in the Node - Red Webpage	Working as expected	Pass	Results verified	No		Mohamed Thariq P, Santho

8.2. User Acceptance Testing:

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [ProductName] project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	3	20
Duplicate	0	2	2	0	4
External	2	3	0	1	6
Fixed	11	2	4	17	34
Not Reproduced	0	0	1	0	1
Skipped	0	0	0	1	1
Won't Fix	0	1	0	3	4
Totals	23	12	9	25	70

3. Test Case Analysis

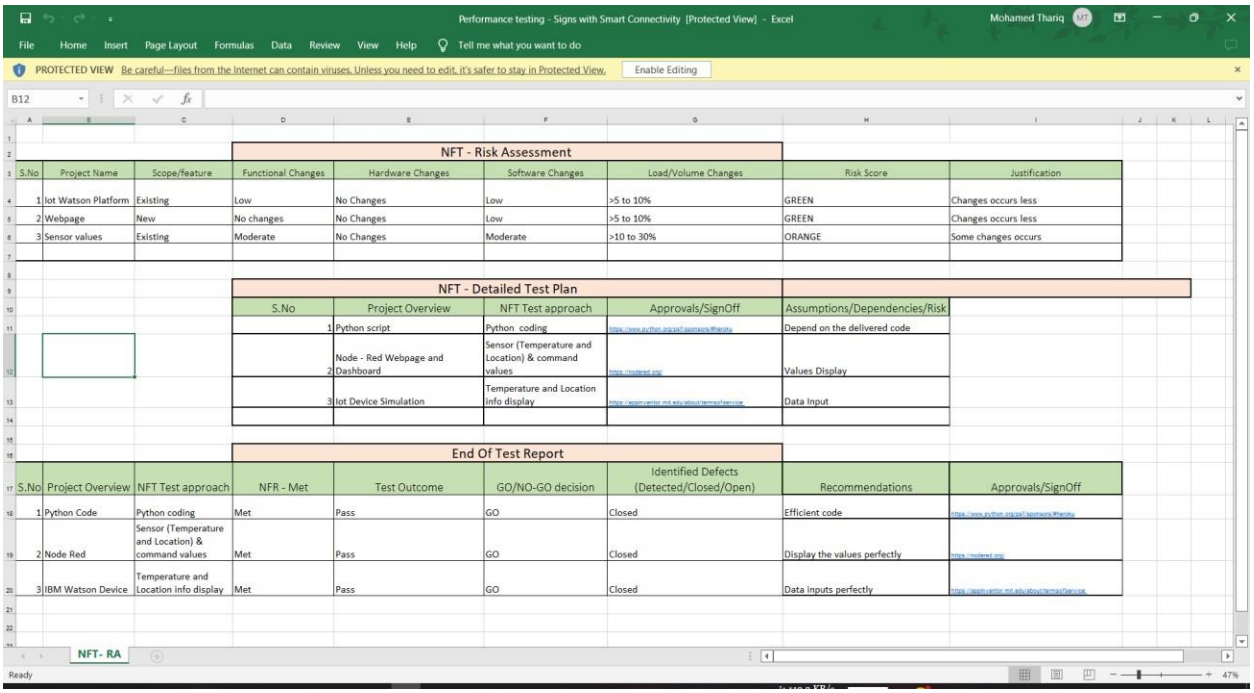
This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	20	0	0	20
Client Application	38	0	0	38
Security	3	0	0	3

Outsource Shipping	3	0	0	3
Exception Reporting	5	0	0	5
Final Report Output	10	0	0	10
Version Control	3	0	0	3

9. Results:

9.1. Performance Metrics:



NFT - Risk Assessment

S.No	Project Name	Scope/feature	Functional Changes	Hardware Changes	Software Changes	Load/Volume Changes	Risk Score	Justification
1	Iot Watson Platform	Existing	Low	No Changes	Low	>5 to 10%	GREEN	Changes occurs less
2	Webpage	New	No changes	No Changes	Low	>5 to 10%	GREEN	Changes occurs less
3	Sensor values	Existing	Moderate	No Changes	Moderate	>10 to 30%	ORANGE	Some changes occurs

NFT - Detailed Test Plan

S.No	Project Overview	NFT Test approach	Approvals/SignOff	Assumptions/Dependencies/Risk
1	Python script	Python coding	https://www.python.org/psf/sponsors/#thanks	Depend on the delivered code
2	Node - Red Webpage and Dashboard	Sensor (Temperature and Location) & command values	https://nodered.org/	Values Display
3	Iot Device Simulation	Temperature and Location info display	https://appinventor.mit.edu/about/termsofservice	Data Input

End Of Test Report

S.No	Project Overview	NFT Test approach	NFR - Met	Test Outcome	GO/NO-GO decision	Identified Defects (Detected/Closed/Open)	Recommendations	Approvals/SignOff
1	Python Code	Python coding	Met	Pass	GO	Closed	Efficient code	https://www.python.org/psf/sponsors/#thanks
2	Node Red	Sensor (Temperature and Location) & command values	Met	Pass	GO	Closed	Display the values perfectly	https://nodered.org/
3	IBM Watson Device	Temperature and Location info display	Met	Pass	GO	Closed	Data inputs perfectly	https://appinventor.mit.edu/about/termsofservice

10. Advantages & Disadvantages:

The main advantage of this project is that it is a dynamic system which can change different modes of operations automatically.

It gathers weather data from open weather API and display speed according based on the climate.

Display service warnings like schools, hospitals and holy places and warn to slow down and be silent.

Traffic data and Construction warning data are given with random inputs. Based on traffic and construction data, warnings are displayed.

Additionally, a speed cam will be integrated with the digital sign board which use Image processing & AI, to get the details of the driver who breaks the traffic rules (especially speed) will be updated in the cloud database.

Also, for No parking and One way rule violations can also be detected and appropriate action can be taken. Violations of stop signs in intersection will also be detected using AI.

11. Conclusion:

The project concluded by replacing the static signboards with smart connected digital sign boards. Digital road signs are an important part of modern infrastructure and are becoming increasingly common. Digital road signs are becoming more common as technology improves and more states adopt them. The use of digital road signs is expected to continue to grow in the future as it would be observed user-friendly, economic, environment friendly, profitable promoting road safety. Digital road signs are designed to improve road safety and efficiency by providing real-time information to drivers. These signs can display a variety of information, including speed limits, traffic conditions, and weather warnings. Digital road signs can help drivers by providing information that is not always available from traditional static signs.

12. Future Scope:

- In the future a speed cam will be integrated with the digital sign board.
- Using Image processing & AI, the details of the driver who breaks the traffic rules will be updated in the cloud database.
- No parking and One way rule violations can also be detected and appropriate action can be taken.
- Violations of stop signs in intersection will also be detected using AI.

13. Appendix:

Code:

```
/*tft.setTextColor(ILI9341_RED);

Future Scope:

  Image Processing:

    Speed Cam

    No Parking

    One-Way

    Stop sign in intersection

*/



// NodeMcU Pins Connection

// #define TFT_MOSI D7

// #define TFT_SCLK D5
```

```
// #define TFT_CS D2
// #define TFT_DC D4
// #define TFT_RST D3


// Library Fuctions
#include "SPI.h"
#include "Adafruit_GFX.h"
#include "Adafruit_ILI9341.h"
#include <SdFat.h>
#include <Adafruit_SPIFlash.h>
#include <Adafruit_ImageReader.h>

// Colours
#define Black 0x000000
#define White 0xffffff

// Arduino Pins Connection
#define USE_SD_CARD
#define SD_CS   4 // SD card select pin
#define TFT_DC 9
```

```
#define TFT_CS 10
#define RESET 8




// Image Initalization..
#if defined(USE_SD_CARD)
  SdFat                SD;         // SD card filesystem
  Adafruit_ImageReader reader(SD); // Image-reader object, pass in SD filesys
#else
  // SPI or QSPI flash filesystem (i.e. CIRCUITPY drive)
  #if defined(__SAMD51__) || defined(NRF52840_XXAA)
    Adafruit_FlashTransport_QSPI flashTransport(PIN_QSPI_SCK,
PIN_QSPI_CS,
      PIN_QSPI_IO0, PIN_QSPI_IO1, PIN_QSPI_IO2, PIN_QSPI_IO3);
  #else
    #if (SPI_INTERFACES_COUNT == 1)
      Adafruit_FlashTransport_SPI flashTransport(SS, &SPI);
    #else
      Adafruit_FlashTransport_SPI flashTransport(SS1, &SPI1);
    #endif
```

```
  #endif
  Adafruit_SPIFlash    flash(&flashTransport);
  FatFileSystem        filesys;
  Adafruit_ImageReader reader(filesys); // Image-reader, pass in flash filesys
#endif

Adafruit_ILI9341 tft = Adafruit_ILI9341(TFT_CS, TFT_DC, RESET);
Adafruit_Image       img;        // An image loaded into RAM
int32_t              width  = 0, // BMP image dimensions
                     height = 0;



// Variables
String Weather;
int Temperature;
int Display_Speed;
int Traffic_Data;
int Construction_Data;
int Service_Data;

// Functions
```

```
void normal_signs();
void speed(int Temp);
void traffic_warning(int tdata);
void construction_warning(int cdata);
void service_warning(int sdata);

// Initial Setup
void setup() {
  ImageReturnCode stat;
  tft.begin();
  Serial.begin(9600);
  tft.setRotation(1);
  tft.setTextColor(ILI9341_WHITE);
}


// Normal signs with wordings
// Weather -- Speed change
// Traffic
// Construction
// School, Hospital Warnings
```

```
void loop() {
  tft.setTextSize(3);
  tft.setCursor(0,0);
  tft.print("Drive Safe :)"); // Initial Mesaage..
  tft.setCursor(0,30);
  tft.print("Speed Limit: 60");
  delay(1000);

  /* Normal Signs Display */
  normal_signs();


  /*Temperature for Speed Control using random input */

  // Temperature = 20;
  Temperature = random(-10,35); // (-10) to 10 -- Snow, 11 to 25 -- Rainy
  speed(Temperature);
  Serial.println(Temperature);
```

```
/*Traffic Data using random input */

// Traffic_Data = 0;
Traffic_Data = random(0,2);
traffic_warning(Traffic_Data);
Serial.println(Traffic_Data);


/* Construction Data using random input */

// Construction_Data = 1;
Construction_Data = random(0,2);
construction_warning(Construction_Data);
Serial.println(Construction_Data);


/* Service Warning Data using random input */

// Service_Data = 1;
Service_Data = random(0,2);
```

```
  service_warning(Service_Data);
  Serial.println(Service_Data);
}

/* Normal Signs Image Display with wordings for awareness */
void normal_signs()
{
  stat = reader.drawBMP("/wokwi.bmp", tft, 0, 0);
  reader.printStatus(stat);
}

/* Speed Control process.. */
void speed(int Temp)
{
  tft.fillScreen(Black);
  if( Temp >= -10 && Temp <= 14) // It's Snow
  {
    Weather = "Snowy";
    tft.setCursor(0,0);
    tft.print("Drive Safe :)");
    tft.setTextSize(2);
```

```
  tft.setCursor(0,40);
  tft.print("Go slow..!");
  tft.setCursor(0,100);
  tft.setTextSize(3);
  tft.print("Speed Limit: 30");
  delay(3000);
  tft.fillScreen(Black);
}
else if(Temp >= 15 && Temp <= 25) // It's Rainy
{
  Weather = "Rainy";
  tft.setCursor(0,0);
  tft.print("Drive Safe :)");
  tft.setTextSize(2);
  tft.setCursor(0,40);
  tft.print("Slippery Road Ahead");
  tft.setCursor(0,70);
  tft.print("Go Slow..!");
  tft.setCursor(0,100);
  tft.setTextSize(3);
  tft.print("Speed Limit: 40");
```

```
    delay(3000);
    tft.fillScreen(Black);
  }
  else
  {
    tft.setCursor(0,0);
    tft.print("Drive Safe!!");
    tft.setCursor(0,30);
    tft.print("Speed Limit: 60");
    delay(3000);
    tft.fillScreen(Black);
  }
  tft.fillScreen(Black);
}



/* Traffic Warning System */


void traffic_warning(int tdata)
{
  tft.fillScreen(Black);
```

```
if (tdata == 0)
{
  tft.setCursor(0,0);
  tft.print("Drive Safe :)");
  // tft.setTextSize(2);
  tft.setCursor(0,40);
  tft.print("Traffic Ahead..");
  tft.setCursor(0,80);
  // tft.setTextSize(3);
  tft.print("Drive Carefully!");
  delay(3000);
  tft.fillScreen(Black);
}
if (tdata == 1)
{
  tft.setCursor(0,0);
  tft.print("Drive Safe :)");
  tft.setTextSize(2);
  tft.setCursor(0,40);
  tft.print("Traffic Ahead..");
  tft.setCursor(0,80);
```

```
    // tft.setTextSize(3);
    tft.print("Take Diversion -->");
    delay(3000);
    tft.fillScreen(Black);
  }
}



/* Construction Warning System */


void construction_warning(int cdata)
{
  tft.fillScreen(Black);
  if (cdata == 0)
  {
    tft.setCursor(0,0);
    tft.print("Drive Safe :)");
    tft.setTextSize(2);
    tft.setCursor(0,40);
    tft.print("Construction Ahead..");
    tft.setCursor(0,80);
```

```
    tft.setTextSize(3);
    tft.print("Drive Carefully..!");
    delay(2000);
    tft.fillScreen(Black);
  }
  if (cdata == 1)
  {
    tft.setCursor(0,0);
    tft.print("Drive Safe :)");
    tft.setTextSize(2.5);
    tft.setCursor(0,40);
    tft.print("Construction Ahead..");
    tft.setCursor(0,80);
    tft.setTextSize(2.5);
    tft.print("Take Diversion <--");
    delay(2000);
    tft.fillScreen(Black);
  }
}
```

```
/* School, Hospital Warning System */


void service_warning(int sdata)
{
  tft.fillScreen(Black);
  if (sdata == 0)
  {
    tft.setCursor(0,0);
    tft.print("Drive Safe :)");
    tft.setTextSize(2);
    tft.setCursor(0,40);
    tft.print("School Ahead..");
    tft.setCursor(0,80);
    tft.setTextSize(3);
    tft.print("Drive Carefully.! \nNo Horn!");
    delay(2000);
    tft.fillScreen(Black);
  }
  if (sdata == 1)
  {
    tft.setCursor(0,0);
```

```
    tft.print("Drive Safe :)");
    tft.setTextSize(2.5);
    tft.setCursor(0,40);
    tft.print("Hospital Ahead..");
    tft.setCursor(0,80);
    tft.setTextSize(2.5);
    tft.print("Drive Carefully.! \nNo Horn!");
    delay(2000);
    tft.fillScreen(Black);
  }
}
```

Node – Red Dashboard (Flow) Link - https://node-red-dsb.eu-gb.mybluemix.net/red/#flow/c8f5c2e4125f08cb

Node – Red Dashboard (UI) Link - https://node-red-dsb.eu-gb.mybluemix.net/ui/#!/0?socketid=fsJfHymZb0JMNE0bAAAD

Node – Red Dashboard (Webpage) Link - https://node-red-dsb.eu-gb.mybluemix.net/simple

GitHub Project Link - https://github.com/IBM-EPBL/IBM-Project-43785-1660719573

Video Drive Link - https://drive.google.com/drive/folders/1KnLe_wOO9nI6Aw2jGRKzIU6zvHhYKSt3?usp=sharing