

# IBM NALAIYA THIRAN 2022-23 PROJECT REPORT

Team ID	PNT2022TMID22305
Project Name	Signs with Smart Connectivity for Better Road Safety
Date	19/11/2022

Team Lead           HEMANTH KUMAR R

Team Member 1   DAYANIDHI S

Team Member 2   SARAVANAN S

Team Member 3   KISHORE K S

## SIGNS WITH SMART CONNECTIVITY FOR BETTER ROAD SAFETY

**TEAM ID - PNT2022TMID22305**

### 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 be set using buttons.

## 2. LITERATURE SURVEY

### 2.1 EXISTING PROBLEM

- Static sign board cannot update automatically based on weather condition which leads to accidents.
- School zone are shown for the drivers should watch out for children, reduce speed, and obey any signals from a crossing guard.
- School zone are shown for the drivers should watch out for children, reduce speed, and obey any signals from a crossing guard.
- Damage in static sign board such as paint deterioration, folded sheet and pole bent can lead to many accidents.

### 2.2 REFERENCES

- Andrzej Czyżewski in his paper titled "Development of Intelligent Road Signs with V2X Interface for Adaptive Traffic Controlling", IEEE 2019, developed IOT based intelligent road signs capable of interacting with both the vehicles and other neighbouring sign boards using LORA. These sign boards were capable of communicating with one another and changing the speed limit based on traffic and weather.
- Muhammed O. Sayin, ChungWei Lin, Eunsuk Kang, Shinichi Shiraishi & Tamer Basar in their paper titled "Reliable Smart Road Signs", IEEE 2019, proposed a game theoretical adversarial intervention detection mechanism for reliable smart road signs. A future trend in intelligent transportation systems is "smart road sign" that incorporate smart codes (e.g., visible at infrared) on their surface to provide more detailed information to smart vehicles.
- L.F.P. Oliveira, L.T. Manera, P.D.G. Luz in their paper titled "Smart Traffic Light Controller System", IEEE 2019, developed smart traffic lights capable of traffic accident detection enabling the enhancement of traffic light management systems, blocking and creating alternative routes to not only avoid the traffic jams, but also avoid new accidents.

- Dariusz Grabowski & Andrzej Czyzewski in their paper titled "System for monitoring road slippery based on CCTV cameras and convolutional neural networks" , Springer Publications 2020, made use of Convolutional Neural Networks to identify slippery roads using CCTV cameras

### 2.3 PROBLEM STATEMENT DEFINITION

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

## 3. IDEATION AND PROPOSED SOLUTION

### 3.1 EMPATHY MAP CANVAS



### 3.2 IDEATION & BRAINSTORMING

<https://github.com/IBM-EPBL/IBM-Project-21705-1659788943/blob/main/Project%20Design%20%26%20Planning/Ideation%20Phase/IDEATION.pdf>

#### Step-1: Team Gathering, Collaboration and Select the Problem Statement

The image shows a digital brainstorming template. On the left, a vertical blue bar is labeled 'Template'. The main content area has a light gray background. At the top, there is a circular icon containing a lightbulb with a brain inside, and wavy lines to its right. Below this icon, the title 'Brainstorm & idea prioritization' is displayed in bold. Underneath the title, the subtitle 'Signs With Smart Connectivity for Better Road Safety' is written. Further down, there are three small icons: a speech bubble, a document, and a person. At the bottom of the template area, there is a button labeled 'Share template feedback'. To the right of the template area, a numbered list item '1' is followed by the heading 'Define your problem statement'. Below this heading, a paragraph of text describes the current static road sign system and proposes a digital solution for dynamic sign changes based on traffic and accident data.

**Template**

**Brainstorm & idea prioritization**

Signs With Smart Connectivity for Better Road Safety

Share template feedback

**1**

**Define your problem statement**

In present Systems the road signs and the speed limits are Static. We can consider some cases when there are some road diversions due to heavy traffic or due to accidents the we 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.

## Step-2: Brainstorm, Idea Listing and Grouping

2

### Brainstorm

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

10 minutes

#### Hemant Kumar R

Using speed as an acoustic feedback.  
Smart cars are called to stop after LED display.  
Bringing smart road signs to enhance the driving safety etc.

#### Saravanan S

Superior lane markings can avoid the danger of accidents.  
Detecting the presence of heavy vehicles on roads to avoid accidents.  
Usage of Autonomous emergency braking (AEB).  
Ultra-power light blinking on the road.

#### Dayanidhi S

Class detection marks to indicate which vehicle is in the lane.  
Automatic change of lane markings.  
When people try to make a turn, the road markings change to indicate the direction of traffic.

#### Kishore K S

Lane departure warning (LDW).  
The system can be used for better road engineering and infrastructure.  
Better road engineering and infrastructure.

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 can break it up into smaller sub-groups.

20 minutes

Speed limit suggestion

Notification from front vehicle if emergency brake is applied

Flickering road signs using led can grab drivers attention.

Implementing ADAS feature even on entry level cars

Better road design

## Step-3: 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



### 3.3 PROPOSED SOLUTION

<https://github.com/IBM-EPBL/IBM-Project-21705-1659788943/blob/main/Project%20Design%20%26%20Planning/Project%20Design%20Phase%20-%20I/Proposed%20Solution.pdf>

- Use a ESP32 to drive a display as a replacement for static sign boards.
- Configure IBM cloud server such that upon making a single http request with unique id, usual speed limit & hospital/school zone info, it returns processes the data at cloud and returns only the message to be displayed at the sign board display.
- Another http end point is configured to set the direction to be displayed. Upon accessing this http end point, the direction is set remotely for a display using it's unique id.

The project team shall fill in the following information in the proposed solution template.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	In present the road signs and the speed limits are Static. In some cases the road signs can change automatically. To control the heavy traffic, Accident, Injuries etc. And it used to change the directions, speed limits, emergency limits.
2.	Idea / Solution description	IoT based application for better road safety. Now we are using static signboards it can be replaced by smart sign boards. IoT it connects to open weather map where it collects the data and speed limits from web app by weather API.
3.	Novelty / Uniqueness	Sign according to scenario diversion signs also displayed. Weather monitoring.
4.	Social Impact / Customer Satisfaction	To avoid accidents, road safety is must. The purpose of making signs with smart connectivity road safety is

### 3.4 PROBLEM SOLUTION FIT

- The display replaces the static signs
- Processing requirement of microcontroller is reduced since all the processing is done in the cloud servers.
- Direction can be remotely set by the concerned authorities without needing to personally attend the site.

## 4. REQUIREMENT ANALYSIS

### 4.1 FUNCTIONAL REQUIREMENTS

<https://github.com/IBM-EPBL/IBM-Project-21705-1659788943/blob/main/Project%20Design%20%26%20Planning/Project%20Design%20Phase%20-%20II/Solution%20Requirements.pdf>

#### Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	<b>User Understanding</b>	The signs should be large, clear, and readable in order for the driver to understand them. They can also include illustrations to help the driver understand them. The methods and measures used to prevent road users from being killed or seriously injured are referred to as road traffic safety.
FR-2	<b>User convenience</b>	Passive traffic safety measures sought to avoid influencing driver behaviour while providing maximum comfort to vehicles. The display should be large enough to show all of the signs correctly, even from a distance.
FR-3	<b>User Guidance</b>	Provides complete information about the system's processes to the user. Defines the product's features and functions. Focus on user requirements, which includes explaining the significance of selecting our product. A user manual is included. Tour of the client application
FR-4	<b>User need</b>	Road safety education is just as important as any other learnable skill. Our goal is to provide road safety information to drivers in order to encourage safer driving habits among current and future drivers and to reduce the number of people killed and injured on our roads each year.
FR-5	<b>User visibility</b>	Visibility refers to a driver's ability to see traffic and environmental surroundings on the road while also allowing other drivers to understand the driver in question.

## 4.2 NON-FUNCTIONAL REQUIREMENTS

<https://github.com/IBM-EPBL/IBM-Project-21705-1659788943/blob/main/Project%20Design%20%26%20Planning/Project%20Design%20Phase%20-%20II/Solution%20Requirements.pdf>

### Non-functional Requirements:

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

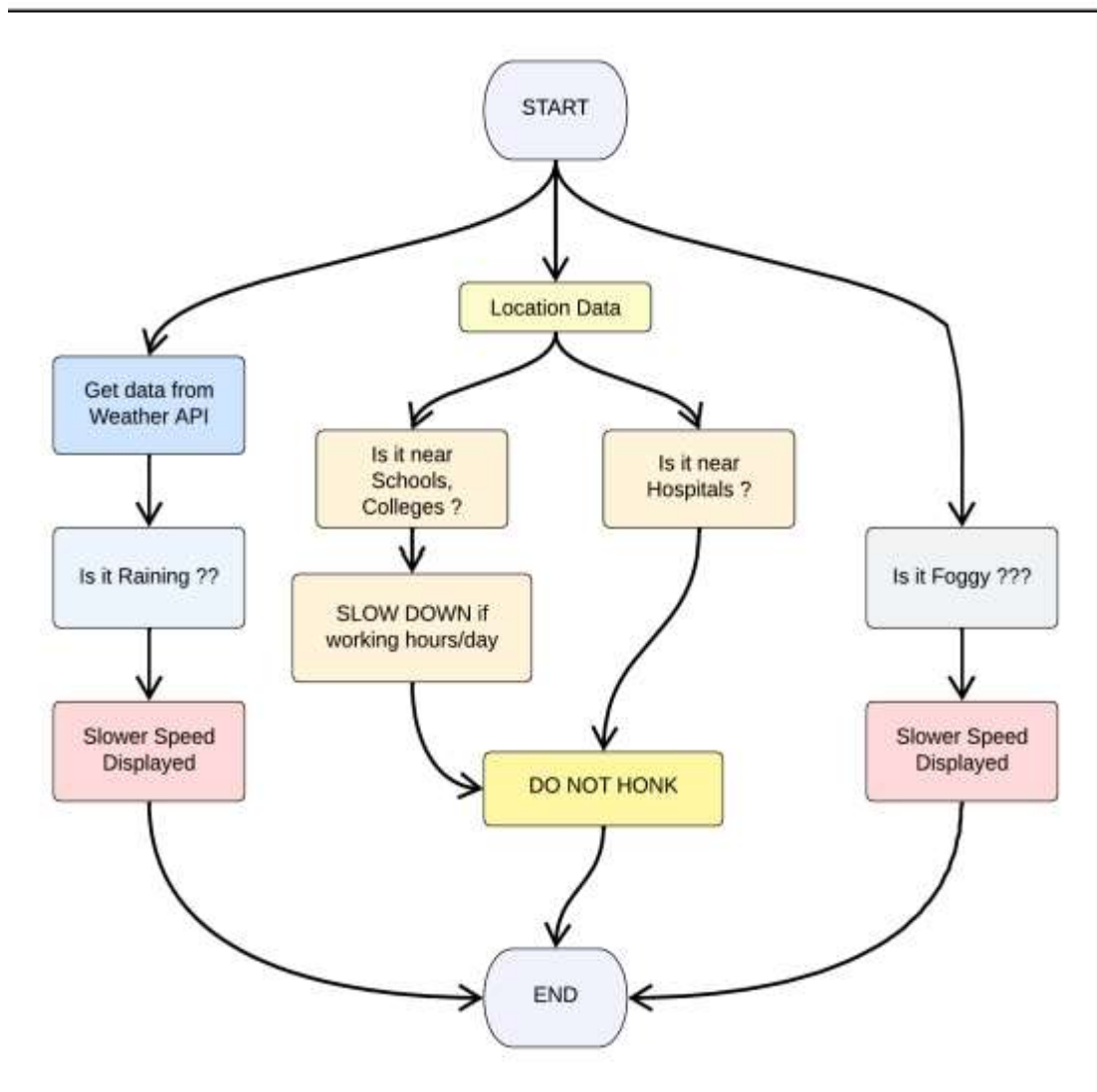
FR No.	Non-Functional Requirement	Description
NFR-1	Usability	The user should read the provided user manual and guide to fully utilise all of the features. The software should be understood, used, and learned by its intended users.
NFR-2	Security	It should have a good security system so that nobody else is able to hack into it and display their own directions.
NFR-3	Reliability	The system that collects information from the surrounding environment and processes it in real time so that drivers can be comfortable that the signs are providing the best results possible.
NFR-4	Performance	A performance review looks at the current state of road safety, helping the government in trying to identify a most important security aspects and suggesting actions to be taken.
NFR-5	Availability	It should be available 24 hours a day, seven days a week to benefit the customer.
NFR-6	Scalability	More features, such as alarms, and many more sensors for the preacquisition of vehicles and death situations, can be added in the future.



## 5. PROJECT DESIGN

### 5.1 DATA FLOW DIAGRAMS

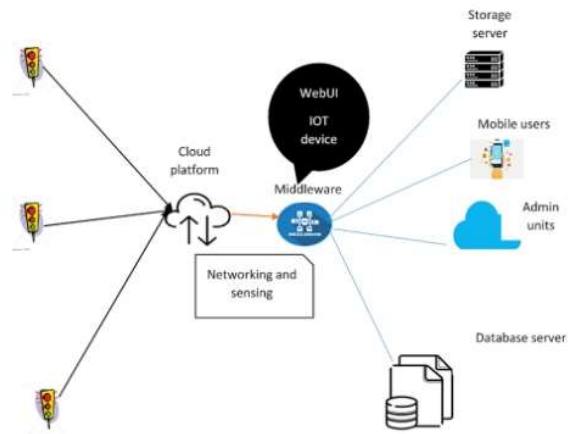
<https://github.com/IBM-EPBL/IBM-Project-21705-1659788943/blob/main/Project%20Design%20%26%20Planning/Project%20Design%20Phase%20%20II/Data%20Flow%20Diagrams%20And%20User%20Stories.pdf>



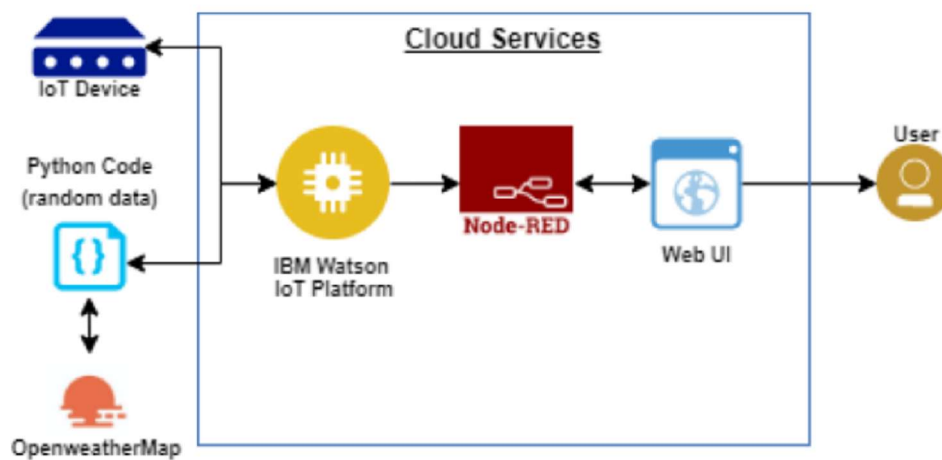
## 5.2 SOLUTION & TECHNICAL ARCHITECTURE

[https://github.com/IBM-EPBL/IBM-Project-21705-1659788943/blob/main/Project Design %26 Planning/Project Design Phase - I/SOLUTION ARCHITECTURE.pdf](https://github.com/IBM-EPBL/IBM-Project-21705-1659788943/blob/main/Project%20Design%20Planning/Project%20Design%20Phase%20-%20I/SOLUTION%20ARCHITECTURE.pdf)

### Solution Architecture Diagram:



### Architecture:



## 5.3 USER STORIES

[https://github.com/IBM-EPBL/IBM-Project-21705-1659788943/blob/main/Project Design %26 Planning/Project Design Phase - II/Data Flow Diagrams And User Stories.pdf](https://github.com/IBM-EPBL/IBM-Project-21705-1659788943/blob/main/Project%20Design%20Planning/Project%20Design%20Phase%20-%20II/Data%20Flow%20Diagrams%20And%20User%20Stories.pdf)

### User Stories

Use the below template to list all the user stories for the product.

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

## 6. PROJECT PLANNING AND SCHEDULING PHASE

### 6.1 SPRINT PLANNING & ESTIMATION

[https://github.com/IBM-EPBL/IBM-Project-21705-1659788943/blob/main/Project Design %26 Planning/Project Planning/Project Planning Phase - Sprint Delivery Plan.pdf](https://github.com/IBM-EPBL/IBM-Project-21705-1659788943/blob/main/Project%20Design%20Planning/Project%20Planning/Project%20Planning%20Phase%20-%20Sprint%20Delivery%20Plan.pdf)

### Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Initializing the Resources	USN-1	Create an account in OpenWeather API	2	High	Dayanidhi S Hemanth Kumar R Kishore K S Saravanan S
Sprint-1	Code in Software is written	USN-2	Write a python script using the inputs given from OpenWeather API	1	High	Dayanidhi S Hemanth Kumar R Kishore K S Saravanan S
Sprint-2	Sending the software to cloud	USN-3	The python code from sprint 1 should be sent to cloud so that it is easily accessible	2	Low	Hemanth Kumar R Dayanidhi S Kishore K S Saravanan S
Sprint-3	Initialising the connection between hardware and cloud	USN-4	The hardware should be inter-grated for the easy access of the cloud functions	2	Medium	Dayanidhi S Saravanan S Hamanth Kumar R Kishore K S
Sprint-4	User input-output	USN-5	Rectify all the shortcomings/errors and initiate	1	High	Dayanidhi S

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
	optimisation and error identification and rectification		the optimisation for better usage			Hamanth Kumar R Kishore K S Saravanan S

### Project Tracker, Velocity & Burn down Chart: (4 Marks)

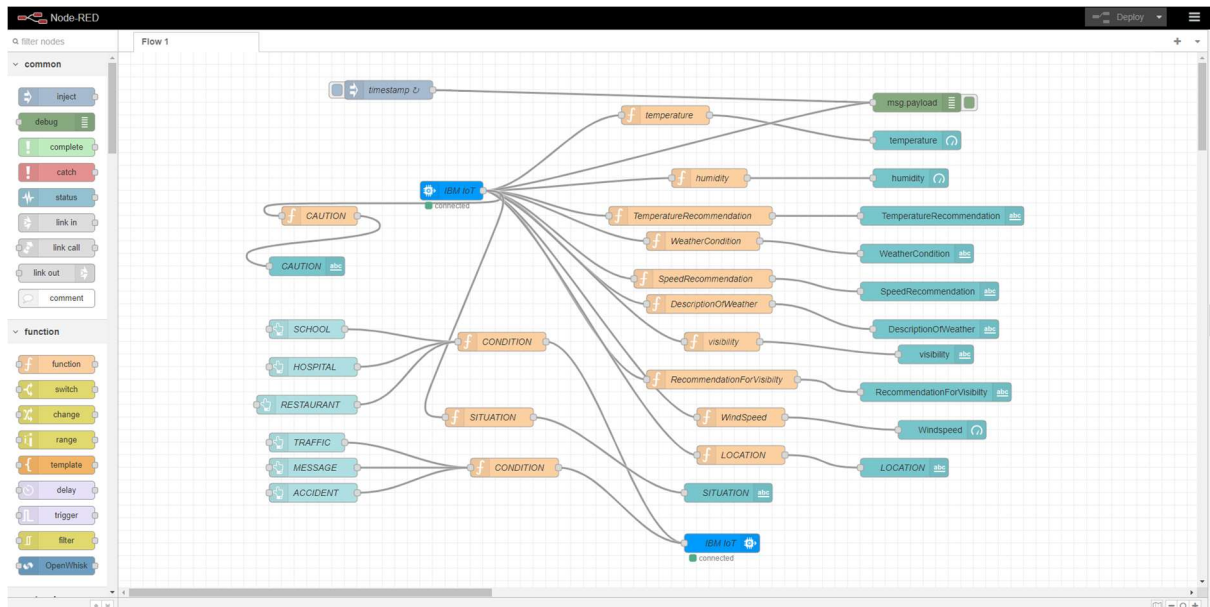
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	02 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	09 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	17 Nov 2022

## 6.2 SPRINT DELIVERY SCHEDULE

[https://github.com/IBM-EPBL/IBM-Project-21705-1659788943/blob/main/Project Design %26 Planning/Project Planning/Project Planning Phase - Sprint Delivery Plan.pdf](https://github.com/IBM-EPBL/IBM-Project-21705-1659788943/blob/main/Project%20Planning/Project%20Planning/Project%20Planning%20Phase%20-%20Sprint%20Delivery%20Plan.pdf)

## 7. CODING & SOLUTIONING

### 7.1 CREATING THE SYSTEM USING NODE-RED



### 7.2 CONNECTING THE PYTHON SCRIPT

```
File Edit Format Run Options Window Help
import wiotp.sdk.device #Importing library files for connecting with CLOUD,sdk=software development kit
import requests #for API request
import json #converting it to json(key:value)
import sys
myconfig = {
    "identity": {
        "orgid": "org1",
        "typeid": "SmartSigns", #configuration wit CLOUD,finding identity
        "deviceid": "SD"
    },
    "auth": {
        "token": "Hrme0yYrgt-08XK" #authenticating with cloud device
    }
}
#INITIALISE AND FATAL SITUATION ALERT MESSAGE DISPLAYING IN WEB UI WHEN THE
client = wiotp.sdk.device.DeviceClient(config=myconfig, logHandler=None) #initialising device client with above myconfig detail
client.connect()
ALERT=""
NOTIFY=""
def myCommandCallback(cmd):
    print("Message received from IBM IoT Platform: %s" % cmd.data["command"])
    m=cmd.data["command"]
    #THIS IF CONDITION BLOCK IS FOR TRAFFIC AND FATAL SITUATION ALERT MESSAGE DISPLAYING IN WEB UI WHEN THE MESSAGE WAS RECEIVED FROM THE ROAD SAFETY OFFICE
    ALERT=""
    NOTIFY=""
    if(m=="TRAFFIC"):
        ALERT="TRAFFIC - PLEASE WAIT OR PREPER ANOTHER ROUTE"
        print("*****//PLEASE WAIT OR PREPER ANOTHER ROUTE//*****")
    elif(m=="ACCIDENT"):
        ALERT="ACCIDENT - TAKE DIVERSION"
        print("*****//TAKE DIVERSION//*****")
    elif(m=="MESSAGE"):
        ALERT="HAVE A NICE DAY!"
        print("HAVE A NICE DAY!")
    #THE BELOW CONDITION BLOCK IS TO DISPLAY HOSPITAL ,SCHOOL, AND RESTAURANT REGIONED AREA AND SPEED RECOMMENDATION
    if(m=="SCHOOL"):
        NOTIFY="SCHOOL REGION MAINTAIN SPEED LIMIT BELOW 40KM/HR"
        print("SCHOOL REGION MAINTAIN SPEED LIMIT BELOW 40KM/HR")
    elif(m=="HOSPITAL"):
        NOTIFY="HOSPITAL REGION DONT USE HORN"
        print("HOSPITAL REGION DONT USE HORN")
    elif(m=="RESTAURANT"):
        NOTIFY="CROWDED AREA PLEASE MAINTAIN SPEED LIMIT"
        print("CROWDED AREA PLEASE MAINTAIN SPEED LIMIT")
    mydata={}
    if(m=="TRAFFIC" or m=="ACCIDENT" or m=="MESSAGE"):
        mydata={"SITUATION":ALERT}
    elif(m=="SCHOOL" or m=="HOSPITAL" or m=="RESTAURANT" ):
        mydata={"CAUTION":NOTIFY}
    client.publishEvent("sd","json",mydata)
def run():
    print("=====")
    AREA = "Chennai, IN"
    weatherData = requests.get("https://api.openweathermap.org/data/2.5/weather?q="+ AREA + "&appid=c033ef9eaf0ba505b059024415b4aeeunits=metric")
    a=weatherData.text
    b=json.loads(a)
    temp = b["main"]["temp"]
    humd = b["main"]["humidity"]
    main = b["weather"][0]["main"] #0th index is taken from the object
    description = b["weather"][0]["description"]
    visibility = b["visibility"]
```

```

finalpy - C:\Users\henan\Desktop\finalfinalpy (3.7.0)
File Edit Format Run Options Window Help
mydata={"temp":25.99,"humidity":75}
client.publishEvent("SS","json",mydata)
while True:
    print("=====")
    AREA = "Chennai, IN"
    weatherData = requests.get("https://api.openweathermap.org/data/2.5/weather?ip=" + AREA + "&appid=c3e4f7eaf0ba55b8595624415b4eakunitmetric")
    a=weatherData.text
    b=json.loads(a)
    temp = b["main"]["temp"]
    humi = b["main"]["humidity"]
    main = b["weather"][0]["main"] #0th index is taken from the object
    description = b["weather"][0]["description"]
    visibility = b["visibility"]
    Windspeed = b["wind"]["speed"]
    TemperatureRecommendation = ""
    SpeedRecommendation = ""
    RecommendationForVisibility = ""
    #print("Temperature(celsius) :",b["main"]["temp"])
    if (temp>33):
        TemperatureRecommendation="Temperature is higher than ideal value"
        #print("Temperature is higher than ideal value")
    elif (temp<19):
        TemperatureRecommendation="Temperature is lower than ideal value"
        #print("Temperature is lower than ideal value")
    else:
        TemperatureRecommendation="Temperature is ideal"
        #print("Temperature is ideal")
    #print("Humidity :",b["main"]["humidity"])
    #print("WeatherCondition",b["weather"][0]["main"])
    if (main == "Rain"):
        #print("Rain")
        SpeedRecommendation = "30KM/HR ,ROAD WILL BE SLIPPERY"
        #print("Main",b["main"],"Rain")
        #print("SPEED RECOMMENDATION : 30KM/HR ,ROAD WILL BE SLIPPERY")
    elif (main == "Cloudy"):
        SpeedRecommendation = "30KM/HR"
        #print("SPEED RECOMMENDATION : 30KM/HR")
    elif (main == "Mist"):
        SpeedRecommendation = "30KM/HR and switch on the headlight"
        #print("SPEED RECOMMENDATION : 30KM/HR and switch on the Headlight")
    elif (main == "Thunderstorm"):
        SpeedRecommendation = "30KM/HR and stay away in the open place"
        #print("SPEED RECOMMENDATION : 30KM/HR and stay away in the open place")
    elif (main == "Clear"):
        SpeedRecommendation = "MAINTAIN NORMAL SPEED LIMIT UP TO 50 KM/HR"
        #print("SPEED RECOMMENDATION : 30KM/HR and stay away in the open place")
    #print("Description of weather :",b["weather"][0]["description"])
    #print("visibility",b["visibility"])
    if (visibility<1000):
        RecommendationForVisibility = "SPEED RECOMMENDATION : 30KM/HR and SWITCH ON THE HEAD LIGHT"
    else:
        RecommendationForVisibility = "visibility range is ideal for vehicles"
    #print("SPEED RECOMMENDATION : 30KM/HR and SWITCH ON THE HEAD LIGHT")
    mydata={"Temperature":temp,"TemperatureRecommendation":TemperatureRecommendation,"humidity":humi,"WeatherCondition":main,"SpeedRecommendation":SpeedRecommendation,"DescriptionOfWeather":description,"Visibility":visibility,"RecommendationForVisibility":RecommendationForVisibility}
    client.publishEvent("SS","json",mydata)
    client.commandCallback = myCommandCallback

```

## 7.3 CONNECTING IBM WATSON ASSISTANT

cdmb9.internetofthings.ibmcloud.com/dashboard/devices/browse

IBM Watson IoT Platform

dayanidhis.cs19@veltechnmultitech.org  
ID: cdmb9

Add Device

Device ID	Status	Type	Location	Device Name	Last Seen	Owner
1234	Disconnected	Node	Device		Oct 25, 2022 6:20 PM	dayanidhis.cs19@veltechnmultitech.org
CL	Disconnected	cloud	Device		Nov 18, 2022 6:17 AM	dayanidhis.cs19@veltechnmultitech.org
FF	Disconnected	Final	Device		Nov 17, 2022 4:31 PM	dayanidhis.cs19@veltechnmultitech.org
SS	Connected	Smart signs	Device		Nov 14, 2022 10:26 AM	dayanidhis.cs19@veltechnmultitech.org

Identity Device Information Recent Events State Logs

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
SS	{"temperature":25.99,"TemperatureRecommend...	json	a few seconds ago
SS	{"temperature":25.99,"TemperatureRecommend...	json	a few seconds ago
SS	{"temperature":25.99,"TemperatureRecommend...	json	a few seconds ago
SS	{"temperature":25.99,"TemperatureRecommend...	json	a few seconds ago
SS	{"temperature":25.99,"TemperatureRecommend...	json	a few seconds ago

Items per page 50 | 1-4 of 4 items

1 of 1 page



## 8. TESTING

### 8.1 TEST CASES

Testcases Report.xlsx - Excel													
Kishore K S Saravanan S													
				Date	19-Nov-22								
				Team ID	PNT2022TMID22305								
				Project Name	s with smart connectivity for bet								
				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	Executed By
LoginPage_TC_001	UI	Home Page	Verify whether user is able to access the URL	URL	<a href="https://code-red-mvp-2022-11-11.eu-gb.mobileapps.net/ui">https://code-red-mvp-2022-11-11.eu-gb.mobileapps.net/ui</a>	URL	Able to access the URL	Failed to access in browser	Fail	Wrong browser selected	NO	101	Dayanidhi S
LoginPage_TC_002	UI	Home Page	Verify whether user is able to access the URL	URL	<a href="https://code-red-mvp-2022-11-11.eu-gb.mobileapps.net/ui">https://code-red-mvp-2022-11-11.eu-gb.mobileapps.net/ui</a>	URL	Now User able to access the URL	Able to access in browser	Pass	Able to access in selected browser	yes	102	Saravanan S
CloudStorage_TC_003	Functional	CLOUD	Verify the temperature and humidity for speed recommendation and visibility	CLOUD	1. Temperature and Humidity levels monitored and stored in cloud for a particular places 2. Displayed speed and range of visibility in dynamic display	Temperature in Celsius & Humidity in Percentage	Speed levels and range of visibility displayed dynamically	Does not change dynamically	Fail	Could not connected	NO	107	Kishore K S
CloudStorage_TC_004	Functional	CLOUD	Verify the temperature and humidity for speed recommendation and visibility	CLOUD	1. Temperature and Humidity levels monitored and stored in cloud for a particular places 2. Displayed speed and range of visibility in dynamic display	Temperature in Celsius & Humidity in Percentage	Now speed levels and range of visibility displayed dynamically	Now it changes dynamically as per the factors	Pass	Could connected	YES	108	Hemanth Kumar R
Response_TC_005	Functional	IOT Device	Verify Traffic clearance if Accident Happens	IOT Device	1. Check whether any accident or path interference happens 2. Displayed as per situation - Prefer another route or Take diversion	Accident - "TAKE DIVERSION" Traffic - "PLEASE WAIT OR PREFER ANOTHER ROUTE"	Displayed Instructions	Does not displayed proper instructions	Fail	Check the inputs	NO	116	Dayanidhi S
Response_TC_006	Functional	IOT Device	Verify Traffic clearance if Accident Happens	IOT Device	1. Check whether any accident or path interference happens 2. Displayed as per situation - Prefer another route or Take diversion	Accident - "TAKE DIVERSION" Traffic - "PLEASE WAIT OR PREFER ANOTHER ROUTE"	Displayed Instructions	Now displayed proper instructions to users	Pass	Now verified	YES	117	Kishore K S
Output_TC_007	Functional	CLOUD	Verify the location and indicates if any school/Hospital/Restaurant are there	CLOUD	1. Take the location data 2. pinned the nearby school/Hospital/Restaurants	School/Hospital/Restaurant's location data	Indication for School/Hospital/Restaurants as per pinned location	Does not included the location data	Fail	School - speed limit should below 40km/hr Hospital - Don't use horn Verified indications School - speed limit should below	NO	129	Saravanan S
Output_TC_008	Functional	CLOUD	Verify the location and indicates if any school/Hospital/Restaurant are there	CLOUD	1. Take the location data 2. pinned the nearby school/Hospital/Restaurants	School/Hospital/Restaurant's location data	Indication for School/Hospital/Restaurants as per pinned location	Location data included	Pass	School - speed limit should below 40km/hr Hospital - Don't use horn Verified indications School - speed limit should below 40km/hr Hospital - Don't use horn	YES	130	Hemanth Kumar R

Testcases Report.xlsx - Excel													
Kishore K S Saravanan S													
				Date	19-Nov-22								
				Team ID	PNT2022TMID22305								
				Project Name	s with smart connectivity for bet								
				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	Executed By
Output_TC_007	Functional	CLOUD	Verify the location and indicates if any school/Hospital/Restaurant are there	CLOUD	1. Take the location data 2. pinned the nearby school/Hospital/Restaurants	School/Hospital/Restaurant's location data	Indication for School/Hospital/Restaurants as per pinned location	Does not included the location data	Fail	School - speed limit should below 40km/hr Hospital - Don't use horn Verified indications School - speed limit should below 40km/hr Hospital - Don't use horn	NO	129	Saravanan S
Output_TC_008	Functional	CLOUD	Verify the location and indicates if any school/Hospital/Restaurant are there	CLOUD	1. Take the location data 2. pinned the nearby school/Hospital/Restaurants	School/Hospital/Restaurant's location data	Indication for School/Hospital/Restaurants as per pinned location	Location data included	Pass	School - speed limit should below 40km/hr Hospital - Don't use horn Verified indications School - speed limit should below 40km/hr Hospital - Don't use horn	YES	130	Hemanth Kumar R
TTS_TC_009	Functional	IOT Device	Indicate the message as per the factors	IOT Device and TTS	Take data's from all require factors and showed response in display	Displayed instructions as per the situation	Showed message for precautionary responses	Message displayed as per situations	Pass	Indication message displayed	YES	135	Dayanidhi S Hemanth Kumar R
Final Output_TC_010	Functional	IOT Device	Verify all the responses are showed and dynamically changed in the single display	IOT Device and TTS	Take all the data's retrieved from cloud and showed responses in display as per the data factors	Displayed all the type of instructions showed in a same display	Showing the indications and messages in a display and changes dynamically according to the situation	Indications and messages showed and	Pass	All factors showed in the display	YES	145	Kishore K S Saravanan S

### 8.2 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.

## 1.Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the web UI which provides "Signs with smart connectivity for better road safety "at the time of the release to User Acceptance Testing (UAT).

## 2.Defect Analysis

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	10	0	0	12
Client Application	22	0	0	22
Security	4	0	0	4

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	8	3	2	2	15
Duplicate	3	0	3	0	6
External	4	2	0	2	8
Fixed	5	1	3	8	17
Not Reproduced	0	0	0	0	0
Skipped	1	0	3	1	5
Won't Fix	0	2	1	1	4
Totals	21	8	12	14	55

## Test Case Analysis

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

Outsource Shipping	5	0	0	5
Exception Reporting	7	0	0	7
Final Report Output	4	0	0	4
Version Control	3	0	0	3



## 9. RESULTS

### 9.1 PERFORMANCE METRICS

Performance Test/alpha - Excel									
File Home Insert Page Layout Formulas Data Review View Tell me what you want to do...									
Calibri 14 A A Wrap Text General Conditional Formatting Styles									
Normal Bad Good Neutral Calculation Check Cell Explanatory T... Input Linked Cell Note									
Insert Delete Format Cells Editing									
E16 Weather, Traffic and fatal situations are displayed.									
A B C D E F G H I J K									
NFT - Risk									
S.No	Project Name	Scope/feature	Functional Changes	Hardware Changes	Software Changes	Impact of Downtime	Load/Volume Changes	Risk Score	Justification
1	Signs with smart connectivity for better road safety	Existing	Moderate	No Changes	Low	A minor delay in runtime may occur	>10 to 30%	GREEN	As we have seen the changes, it adds the setup time
NFT - Detailed									
S.No	Project Overview	NFT Test approach	Assumptions/Dependencies/Risks	Approvals/SignOff					
1	Signs with smart connectivity for better road safety	LOAD	Requirement of Alternates in hardware - Improved accurate	SignOff					
End Of Test									
S.No	Project Overview	NFT Test approach	NFR - Met	Test Outcome	GO/NO-GO decision	Recommendations	Identified Defects (Detected/Closed/Open)	Approvals/SignOff	
1	Providing accurate instructions for road users as per situation using smart signs with LED display	LOAD	NFR	Weather, Traffic and fatal situations are displayed	GO	To have advanced versions in browsers	Closed	Approval	
NFT - RA DTP									

## 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. Appendix**

```
import wiotp.sdk.device #importing library files for connecting with
CLOUD,sdk=software developement kit

import requests        #for API request

import json            #converting it to json(key:values)

import sys

myConfig = {
    "identity": {
        "orgId": "c0mbt9",
        "typeId": "Smartsigns", #configuration wit CLOUD,finding identity
        "deviceId":"SS"
    },
    "auth": {
        "token": "Hrtme!0y*FQT-s@HKf" #authenticating with cloud device
    }
}
```

#TRAFFIC AND FATAL SITUATION ALERT MESSAGE DISPLAYING IN  
WEB UI WHEN THE

```
client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
#initialising device client with above myconfig detail
```

```
client.connect()
```

```
ALERT=""
```

```
NOTIFY=""
```

```
def myCommandCallback(cmd):
```

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

```
    m=cmd.data['command']
```

```
    #THIS IF COMDITION BLOCK IS FOR TRAFFIC AND FATAL
SITUATION ALERT MESSAGE DISPLAYING IN WEB UI WHEN THE
MESSAGE WAS RECEIVED FROM THE ROAD SAFETY OFFICE
```

```
    ALERT=""
```

```
    NOTIFY=""
```

```
    if(m=="TRAFFIC"):
```

```
        ALERT="TRAFFIC - PLEASE WAIT OR PREFER ANOTHER
ROUTE"
```

```
        print("*****//PLEASE WAIT OR PREFER ANOTHER
ROUTE//*****")
```

```
    elif(m=="ACCIDENT"):
```

```
        ALERT="ACCIDENT - TAKE DIVERSION"
```

```
        print("*****//TAKE DIVERSION//*****")
```

```
    elif(m=="MESSAGE"):
```

```
        ALERT="HAVE A NICE DAY!"
```

```
        print("HAVE A NICE DAY!")
```

```
    #THE BELOW CONDITION BLOCK IS TO DISPLAY HOSPITAL
,SCHOOL, AND RESTAURANT REGIONED AREA AND SPEED
RECOMMENDATION
```

```

if(m=="SCHOOL"):
    NOTIFY="SCHOOL REGION MAINTAIN SPEED LIMIT BELOW
40KM/HR"
    print("SCHOOL REGION MAINTAIN SPEED LIMIT BELOW
40KM/HR")
elif(m=="HOSPITAL"):
    NOTIFY="HOSPITAL REGION DONT USE HORN"
    print("HOSPITAL REGION DONT USE HORN")
elif(m=="RESTAURANT"):
    NOTIFY="CROWDED AREA PLEASE MAINTAIN SPEED LIMIT"
    print("CROWDED AREA PLEASE MAINTAIN SPEED LIMIT")
mydata1={}
if(m=="TRAFFIC" or m=="ACCIDENT" or m=="MESSAGE"):
    mydata1={"SITUATION":ALERT}
elif(m=="SCHOOL"or m=="HOSPITAL" or m=="RESTAURANT" ):
    mydata1={"CAUTION":NOTIFY}
client.publishEvent("SS","json",mydata1)
while True:
    print("=====")
    AREA = "Chennai, IN"
    weatherData =
requests.get("https://api.openweathermap.org/data/2.5/weather?q=" + AREA +
"&appid=cd23e4f9eaf0ba585b8598624415b4ae&units=metric")
a=weatherData.text
b=json.loads(a)
temp = b["main"]["temp"]
humi = b["main"]["humidity"]
main = b["weather"][0]["main"]    #0th index is taken from the object
description = b["weather"][0]["description"]

```

```

visibility = b["visibility"]
Windspeed = b["wind"]["speed"]

TemperatureRecommendation = ""

SpeedRecommendation = ""

RecommendationForVisibilty = ""
#print("Temperature(cecius) :",b["main"]["temp"])
if (temp>33):
    TemperatureRecommendation="Temperature is higher than ideal value"
    #print("Temperature is higher than ideal value")
elif (temp<19):
    TemperatureRecommendation="Temperature is lower than ideal value"
    #print("Temperature is lower than ideal value")
else:
    TemperatureRecommendation="Temperature is ideal"
    #print("Temperature is ideal ")
#print("Humidity :",b["main"]["humidity"])
#print("WeatherCondition",(b["weather"][0]["main"]))
if (main == "Rain"):
    rain = b["rain"]["1h"]
    SpeedRecommendation = "30KM/HR ,ROAD WILL BE SLIPPERY"
    #print("Rain:",b["rain"]["1h"])
    #print("SPEED RECOMMENDATION : 30KM/HR ,ROAD WILL BE SLIPPERY")
elif (main == "Drizzle"):
    SpeedRecommendation = "30KM/HR"

```

```

    #print("SPEED RECOMMENDATION : 30KM/HR")
elif (main == "Mist"):
    SpeedRecommendation = "30KM/HR and switch on the headlight"
    #print("SPEED RECOMMENDATION : 30KM/HR and switch on the
Headlight")
elif (main == "Thunderstorm"):
    SpeedRecommendation = "30KM/HR and stay away in the open place"
    #print("SPEED RECOMMENDATION : 30KM/HR and stay away in the
open place")
elif (main == "Clouds"):
    SpeedRecommendation = "MAINTAIN NORMAL SPEED LIMIT UPTO
50 KM/HR"
    #print("SPEED RECOMMENDATION : 30KM/HR and stay away in the
open place")
    #print("Description of weather :", (b["weather"][0]["description"]))
    #print("visibility", (b["visibility"]))
    if (visibility < 1000):
        RecommendationForVisibilty = "SPEED RECOMMENDATION :
30KM/HR and SWITCH ON THE HEAD LIGHT"
    else:
        RecommendationForVisibilty = "visibility range is ideal for vechicles"
    #print("SPEED RECOMMENDATION : 30KM/HR and SWITCH ON THE
HEAD LIGHT")
    mydata = {"temperature": temp,
"TemperatureRecommendation": TemperatureRecommendation, "humidity": humi,
"WeatherCondition": main, "SpeedRecommendation": SpeedRecommendation,
"DescriptionOfWeather": description, "visibility": visibility, "RecommendationFo
rVisibilty": RecommendationForVisibilty, "WindSpeed": Windspeed, "LOCATIO
N": AREA}
    print(mydata)
    client.publishEvent("SS", "json", mydata)

```

```
client.commandCallback = myCommandCallback
```

### **Github & Demo Link**

Github link: <https://github.com/IBM-EPBL/IBM-Project-21705-1659788943>

Video demo link: [https://youtu.be/LMi\\_xMqylTk](https://youtu.be/LMi_xMqylTk)

