SIGNS WITH SMART CONNECTIVITY FOR BETTER ROAD SAFETY

IBM PROJECT REPORT

Team ID - PNT2022TMID40209

SRI BALAJI CHOCKALINGAM ENGINEERING COLLEGE,
ARNI

SUBMITTED BY

S GOKUL	512719106005
S SANJAY	512719106015
P KISHORE	512719106302
K SURYA	512719106019

INDEX

Page No.

1. INTRODUCTION

Road safety was always an area that concerned many people around the world and systems that aid the drivers have been appearing ever since cars and computers were combined to make driving safer and more efficient.

While the worldwide figures have, the Global Status Report does indicate higher road fatalities and injuries over countries.

Added to the losses in human lives and wellbeing, considerable monetary losses are incurred in medical expenses, infrastructure repair, and production downtime.

Smart Traffic Management is a system to monitor and control traffic signals using sensors to regulate the flow of traffic and to avoid congestion for a smooth flow of traffic.

Prioritizing traffic like ambulances, police etc. is also one application comes under smart traffic management.

To ensure a smooth and secure flow of traffic, road signs are essential. A major cause of road accidents is negligence in viewing the Traffic signboards and interpreting them incorrectly. The proposed system helps in recognizing the Traffic sign. To improve road-safety by adopting solutions implemented in countries.

2. LITERATURE SURVEY

[1] Nilesh J Uke, Ravindra C Thool, Shailaja Uke, A Vision based Driver Support System for Road Sign Detection, Int. J. on Recent Trends in Engineering and Technology, Vol. 10, No. 1, Jan 2014.

[2] Mrs. P. Shopa, Mrs. N. Sumitha, Dr. P.S. K Patra, Traffic Sign Detection and Recognition Using OpenCV, ICICES2014 - S.A. Engineering College, Chennai, Tamil Nadu, India.

[3] M. Benallal and J. Meunier, Real-time color segmentation of road signs, IEEE CCECE 2003, Canadian Conference on, vol. 3, pp. 1823–1826 vol.3, May 2003.

[4] A. Broggi, P. Cerri, P. Medici, P. Porta, and G. Ghisio, Real time road signs recognition, Intelligent Vehicles Symposium, 2007 IEEE, pp. 981– 986, June 2007

[5] A. D. L. Escalera, J. M. A. Armingol, and M. Mata, Traffic sign recognition and analysis for intelligent vehicles, Image and Vision Computing, vol. 21, pp. 247–258, 2003.

[6] G. Loy, Fast shape-based road sign detection for a driver assistance system, in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2004, pp. 70–75.

[7] C. Paulo and P. Correia, Automatic detection and classification of traffic signs, in Image Analysis for Multimedia Interactive Services, 2007. WIAMIS ’07. Eighth International Workshop on, June 2007.

[8] D. Gavrila, Traffic sign recognition revisited, in DAGM-Symposium, 1999, pp. 86–93.

[9] P. Viola and M. Jones, Robust real-time object detection, in International Journal of Computer Vision, 2001.

[10] autocarindia.com/bike-news/survey-78-2-wheeler-riders-can8217trecognise-half-the-road-signs-404044.

[11] Mrs. P. Shopa, Mrs. N. Sumitha, Dr. P.S.K Patra, Traffic Sign Detection and Recognition Using OpenCV, ICICES2014 - S.A.Engineering College, Chennai, Tamil Nadu, India.

[12] GREENHALGH, Jack; MIRMEHDI, Majid. Detection and Recognition of Painted Road Surface Markings. In: ICPRAM (1). 2015. p. 130-138.

2.1. PROBLEM STATEMENT

This project will replace the static boards to smart signed boards that will change the speed limits according to the weather climate and show diversion messages if there are accidents in the road and alert messages if there is hospital, schools or any roadworks.

- To replace the static signboards, smart connected sign boards are used.
- These smart connected sign boards get the speed limitations from a web app using weather API and update automatically.
- Based on the weather changes the speed may increase or decrease.
- Based on the traffic and fatal situations the diversion signs are displayed.
- Guide (Schools), Warning and Service (Hospitals, Restaurant) signs are also displayed accordingly.
- Different modes of operations can be selected with the help of buttons.

Image 1

Image 2

3. IDEATION & PROPOSED SOLUTION

The weather and temperature details are obtained from the Open Weather Map API. Using these details, the speed limit will be updated automatically in accordance with the weather conditions. Also, the details regarding any accidents and traffic congestion faced on the particular road are obtained. Based on this, the traffic is diverted followed by a change in map path and the traffic is cleared.

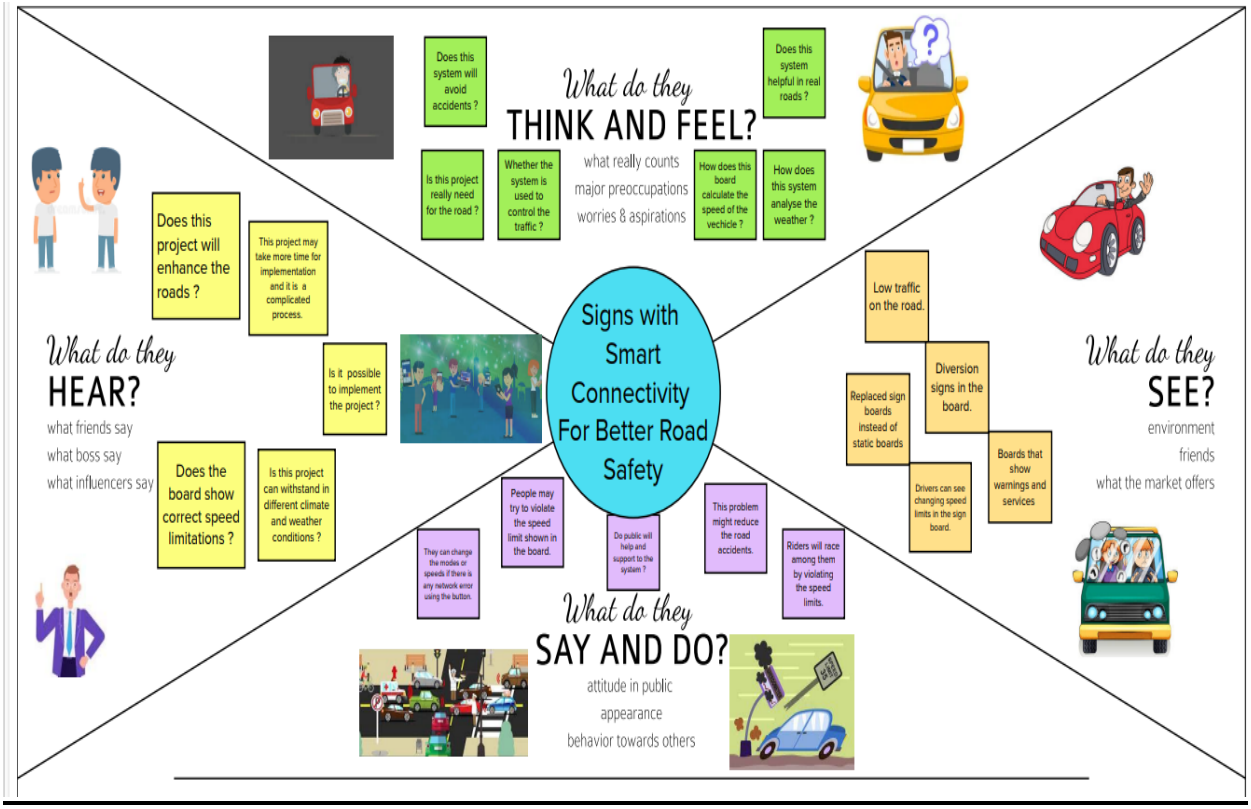
So in the traffic sign board, some buttons will be placed which will be used to make it generic; where each button will be given a functionality such as changing the warning signs, which are predefined and separate signs will be present for both school and hospital zones. By activating this button, either through the web application or the physical buttons, sign of the board can be changed accordingly, and the speed limit will also be set depending upon the zones.

If the pedestrian presses the button that is present on the post at the end of the road, then the traffic will be analysed immediately. Accordingly, the sign of the traffic signal will be changed. This in turn reduces the frequent changing of the traffic signs even if the pedestrians are not present.

3.1. EMPATHY MAPPING:

Build empathy map and keep your focus on the smart sign board

EMPATHY MAP CANVAS



3.2. BRAIN STROMING

Brainstorming is part design thinking. You use it in the ideation phase. It’s extremely popular for design teams because they can expand in all directions.

Brainstorming is a method design teams use to generate ideas to solve clearly defined design problems.

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

4. REQUIREMENTS ANALYSIS

4.1. FUNCTIONAL REQUIREMENTS

FR No.	Functional Requirement(Epic)	Sub Requirement (Story / Sub-Task)
FR-I	User Visibility	Sign Boards should be made of bright colored LEDs capable of attracting driver's attention Not too distracting to cause accidents
FR-2	User Understanding	Should display information through means like images/illustrations with text so that the user can understand the signs correctly
FR-3	User Convenience	Display should be big enough to display all the signs correctly so that it is visible even to far awaydrivers

4.2. NON-FUNCTIONAL REQUIREMENTS

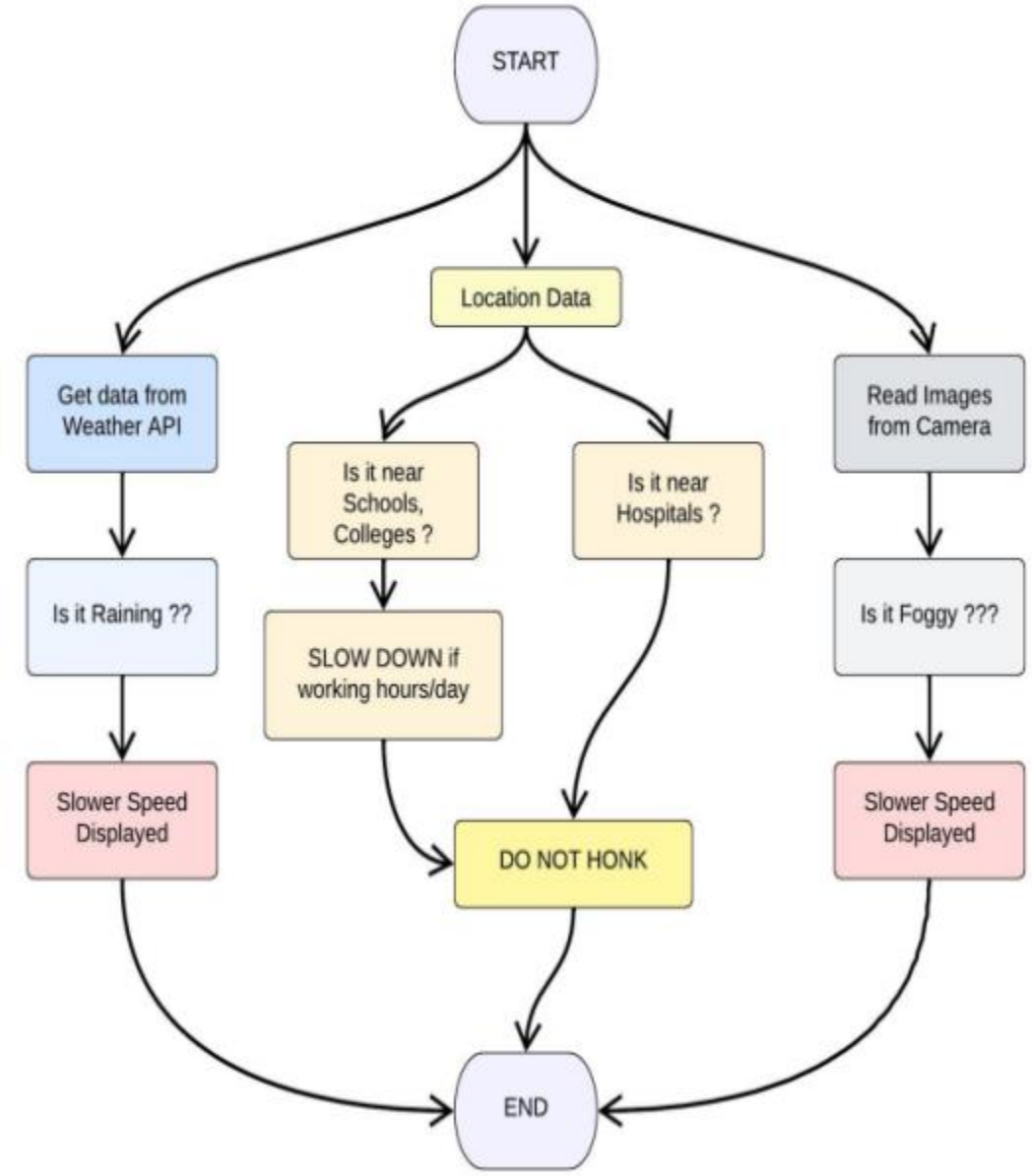
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-I	Usability	Should be able to dynamically update with respectto time.
NFR-2	Security	Should be secure enough that only the intended messages are displayed in the display.
NFR-3	Reliability	Should convey the traffic information correctly.
NFR-4	Performance	Display should update dynamically whenever the weather or traffic values are updated
NFR-5	Availability	Should be on service 24/7
NFR-6	Scalability	Should be modular and hence able to scaleon servers horizontally.

5. PROJECT DESIGN:

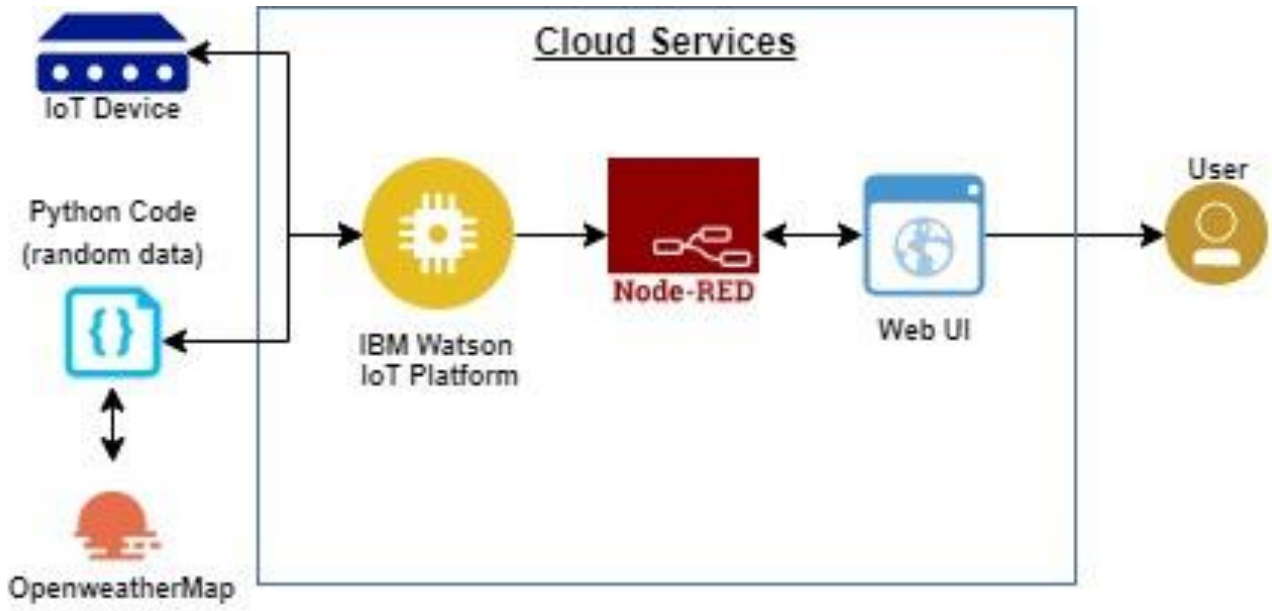
5.1. DATA FLOW DIAGRAMS:

A data-flow diagram is a way of representing a flow of data through a process or a system. The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow — there are no decision rules and no loops.



5.2. SOLUTION &TECHNICAL ARCHITECTURE

Solution architecture, term used in information technology with various definitions such as; “A description of a discrete and focused business operation or activity and how IS/IT supports that operation”



5.3. USER STORIES

A user story is a short, simple description of a feature told from the perspective of the person who desires the new capability, usually a user or customer of the system.

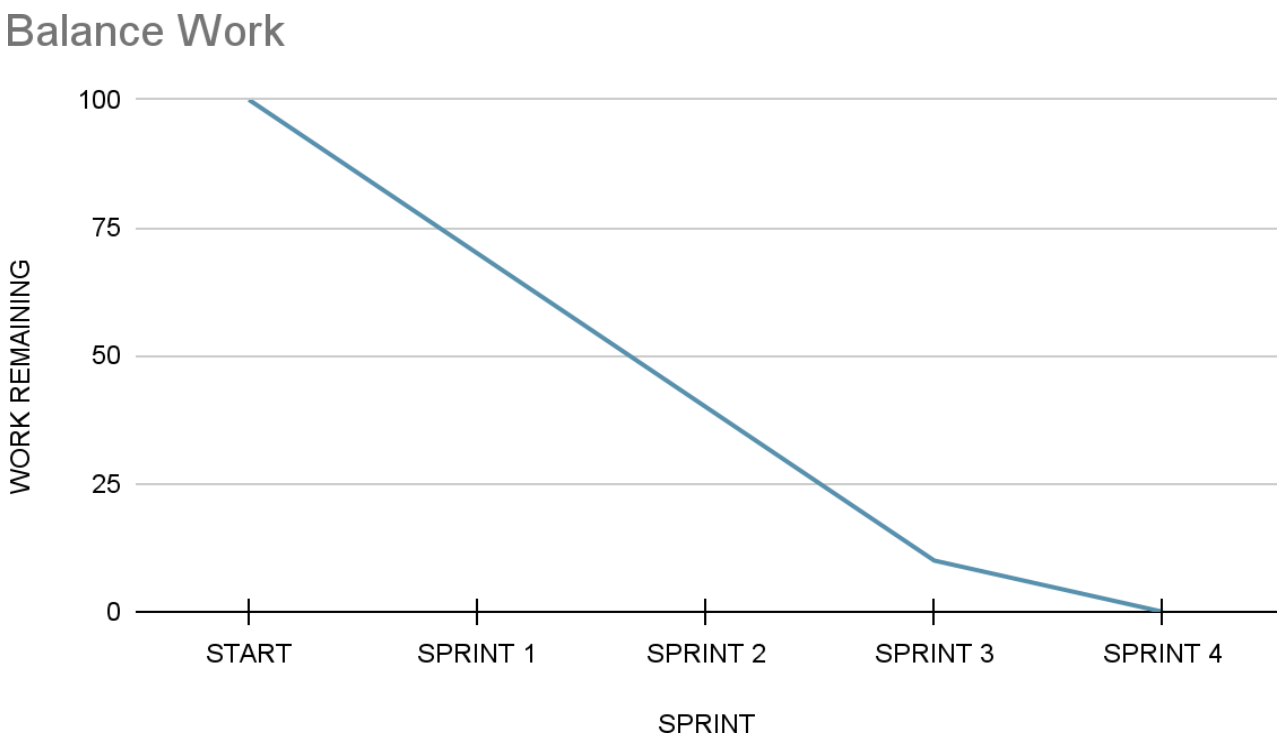
6. PROJECT PLANNING &SCHEDULING

6.1. SPRINT PLANNING &ESTIMATION

Sprint	Functional Requirement (Epic)	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Resources Initialization	Create and initialize accounts in various public APIs like Open Weather Map API.	1	LOW	GOKUL SANJAY KISHORE SURYA
Sprint-1	Local Server/Software Run	Write a Python program that outputs results given the inputs like weather and location.	1	MEDIUM	GOKUL SANJAY KISHORE SURYA
Sprint-2	Push the server/software to cloud	Push the code from Sprint 1 to cloud so it can be accessed from anywhere	2	MEDIUM	GOKUL SANJAY KISHORE SURYA
Sprint-3	Hardware initialization	Integrate the hardware to be able to access the cloud functions and provide inputs to the same.	2	HIGH	GOKUL SANJAY KISHORE SURYA
Sprint-4	UI/UX Optimization & Debugging	Optimize all the shortcomings and provide better user experience.	2	LOW	GOKUL SANJAY KISHORE SURYA

Sprint	Functional Requirement (Epic)	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Resources Initialization	Create and initialize accounts in various public APIs like Open Weather Map API.	1	LOW	GOKUL SANJAY KISHORE SURYA
Sprint-1	Local Server/Software Run	Write a Python program that outputs results given the inputs like weather and location.	1	MEDIUM	GOKUL SANJAY KISHORE SURYA
Sprint-2	Push the server/software to cloud	Push the code from Sprint 1 to cloud so it can be accessed from anywhere	2	MEDIUM	GOKUL SANJAY KISHORE SURYA
Sprint-3	Hardware initialization	Integrate the hardware to be able to access the cloud functions and provide inputs to the same.	2	HIGH	GOKUL SANJAY KISHORE SURYA
Sprint-4	UI/UX Optimization & Debugging	Optimize all the shortcomings and provide better user experience.	2	LOW	GOKUL SANJAY KISHORE SURYA

BURNDOWN CHART



7. CODING & SOLUTION

Python code

Brain.py

```
# IMPORT SECTION STARTS


import weather
from datetime import datetime as dt
from publishData import logData2Cloud as log2cloud


# IMPORT SECTION ENDS
# -----------------------------------------------
# UTILITY LOGIC SECTION STARTS
def processConditions(myLocation,APIKEY,localityInfo):
    weatherData = weather.get(myLocation,APIKEY)


    log2cloud(myLocation,weatherData["temperature"],weatherData["visibility"])


    finalSpeed = localityInfo["usualSpeedLimit"] if "rain" not in weatherData
else localityInfo["usualSpeedLimit"]/2
    finalSpeed = finalSpeed if weatherData["visibility"]>35 else finalSpeed/2


    if(localityInfo["hospitalsNearby"]):
        # hospital zone
        doNotHonk = True
    else:
        if(localityInfo["schools"]["schoolZone"]==False):
            # neither school nor hospital zone
            doNotHonk = False
        else:
            # school zone
            now = [dt.now().hour,dt.now().minute]
            activeTime = [list(map(int,_.split(":"))) for _ in
localityInfo["schools"]["activeTime"]]
            doNotHonk = activeTime[0][0]<=now[0]<=activeTime[1][0] and
activeTime[0][1]<=now[1]<=activeTime[1][1]
```

```
        return({
            "speed" : finalSpeed,
            "doNotHonk" : doNotHonk
        })


    # UTILITY LOGIC SECTION ENDS
```

Main.py

```
import
brain


# IMPORT SECTION ENDS
# ------------------------------------------------
# USER INPUT SECTION STARTS


myLocation = "Chennai,IN"
APIKEY = "f32a1ad47d10959d2275374e9a42866c"


localityInfo = {
    "schools" : {
        "schoolZone" : True,
        "activeTime" : ["8:00","17:30"] # schools active from 7 AM till 5:30 PM
        },
    "hospitalsNearby" : False,
    "usualSpeedLimit" : 40 # in km/hr
}


# USER INPUT SECTION ENDS
# ------------------------------------------------
# MICRO-CONTROLLER CODE STARTS
while True :
    print(brain.processConditions(myLocation,APIKEY,localityInfo))


'''
```

```
MICRO CONTROLLER CODE WILL BE ADDED IN SPRINT 3 AS PER OUR PLANNED SPRINT SCHEDULE
'''


# MICRO-CONTROLLER CODE ENDS
```

publishData.py

```
import
wiotp.sdk.device
# python -m pip
install wiotp
                    import time


                    # IMPORT SECTION ENDS
                    # ---------------------------------------------
                    # API CONFIG SECTION STARTS


                    myConfig = {
                        "identity" : {
                            "orgId" : "f7n1n8",
                            "typeId" : "weatherdector",
                            "deviceId" : "practice"
                        },
                        "auth" : {
                            "token" : "123456789"
                        }
                    }


                    # API CONFIG SECTION ENDS
                    # ---------------------------------------------
                    # FUNCTIONS SECTION STARTS


                    def myCommandCallback(cmd):
                        print("recieved cmd : ",cmd)
```

```
def logData2Cloud(location,temperature,visibility):
    client =
wiotp.sdk.device.DeviceClient(config=myConfig,logHandlers=None)
    client.connect()
    client.publishEvent(eventId="status",msgFormat="json",data={
        "temperature" : temperature,
        "visibility" : visibility,
        "location" : location
    },qos=0,onPublish=None)
    client.commandCallback = myCommandCallback
    time.sleep(1)
    client.disconnect()


# FUNCTIONS SECTION ENDS
```

Weather.py

```
Import
wiotp.sdk.device
# python -m pip
install wiotp
import time


# IMPORT SECTION ENDS
# ---------------------------------------------
# API CONFIG SECTION STARTS


myConfig = {
    "identity" : {
        "orgId" : "f7n1n8",
        "typeId" : "weatherdector",
        "deviceId" : "practice"
    },
    "auth" : {
        "token" : "123456789"
    }
```

```
}


# API CONFIG SECTION ENDS
# ---------------------------------------------
# FUNCTIONS SECTION STARTS


def myCommandCallback(cmd):
    print("recieved cmd : ",cmd)




def logData2Cloud(location,temperature,visibility):
    client =
wiotp.sdk.device.DeviceClient(config=myConfig,logHandlers=None)
    client.connect()
    client.publishEvent(eventId="status",msgFormat="json",data={
        "temperature" : temperature,
        "visibility" : visibility,
        "location" : location
    },qos=0,onPublish=None)
    client.commandCallback = myCommandCallback
    time.sleep(1)
    client.disconnect()


# FUNCTIONS SECTION ENDS
```

Output

iot-2/type/weatherdector/id/practice/evt/cmd/fmt/json : msg.payload : Object

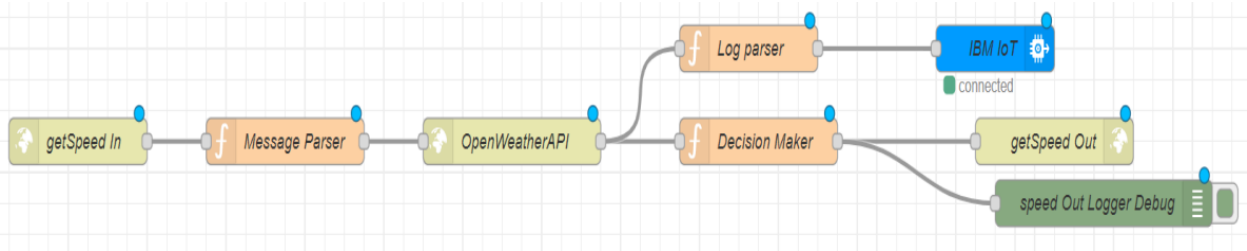
```
{ cm: "Go Straight" }
```

11/19/2022, 8:41:02 PMnode: iotiot-2/type/weatherdector/id/practice/evt/status/fmt/json : msg.payload : Object

```
{ temperature: 26.99000000000001, visibility: 40, location: "Chennai,IN" }
```

7.1. DATABASE SCHEMA

- This part of Node RED flow accepts an http GET end point at "/getSpeed" from which the location, uid, hospital/school zone info are passed.
- Message parser sets the required APIKEY for OpenWeatherAPI for the next block.
- This data is then passed onto Decision Maker which makes all the decisions regarding the message to be output at the display and sends it as a http response.
- This data is displayed at the microcontroller. Thus, a lot of battery is saved due to lesser processing time



8. TESTING

8.1. TEST CASES

- *TEST CASE 1*

  Clear weather - Usual Speed Limit.

- *TEST CASE 2*

  Foggy Weather - Reduced Speed Limit.

- *TEST CASE 3*

  Rainy Weather - Further Reduced Speed Limit.

- *TEST CASE 4*

  School/Hospital Zone - Do not Honk sign is displayed.

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.

9. RESULTS

9.1. PERFORMANCE METRICS

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

10. ADVANTAGES AND DISADVANTAGES

10.1. 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 updating.
- School/Hospital Zone alerts

10.2. DISADVANTAGES

- The size of the display determines the requirement of the micro controller
- Dependent on OpenWeatherMap API and hence the speed reduction is same for a large area in the scale of cities.

11. CONCLUSION

Our project is capable of serving as a replacement for static signs for a comparatively lower cost and can be implemented in the very near future. This will help reduce a lot of accidents and maintain a more peaceful traffic atmosphere in the country.

12. FUTURE SCOPE:

Introduction of intelligent road sign groups in real life scenarios could have great impact on increasing the driving safety by providing the end-user (car driver) with the most accurate information regarding the current road and traffic conditions. Even displaying the information of a suggested driving speed and road surface condition (temperature, icy, wet or dry surface) could result in smoother traffic flows and, what is more important, in increasing a driver's awareness of the road situation.

13. APPENDIX

- GITHUB AND PROJECT DEMO LINK

  https://github.com/IBM-EPBL/IBM-Project-43073-1660712754

- PROJECT DEVELOPMENT PHASE LINK

  https://github.com/IBM-EPBL/IBM-Project-43073-1660712754/tree/main/project%20development

- DEMO VIDEO DOWNLOAD LINK

  https://youtu.be/gC2ovmztJts

Thank You...