



SIGNS WITH SMART CONNECTIVITY FOR BETTER ROAD SAFETY

IBM PROJECT REPORT

Team ID - PNT2022TMID07534

SUBMITTED BY

G.V.JANANI (721719106019)

G.GOWRIPRIYA(721719106016)

N.MAKESWARI(721719106030)

S.A.MYTHILY(721719106034)

Final Deliverables Report

Date	18.11.2022
Team ID	PNT2022TMID07534
Project Name	Signs with Smart Connectivity for Better Road Safety

Team members and their Contributions:

Name	Roll no	Contribution
G.V.JANANI	721719106019	CREATED SOURCE CODE FOR THE WOKWISIMULATOR AND MIT APP CODE.
G.GOWRIPRIYA	721719106016	CREATED NODE RED AND IOT WATSON PLATFORM.
N.MAKESWARI	721719106030	PROJECT REPORT MAKING PROCESS AND GATHERING IDEAS FOR CREATING PROJECT.
S.A.MYTHILY	721719106034	WORKINGS IN NODE RED FLOW AND IBM CLOUD DEPLOYMENT.

Introduction:

1. Sprint 1 – Create and initialize accounts in various public APIs like OpenWeatherMap API, and write a Python program that outputs results given the inputs like weather and location.
2. Sprint 2 – Push data from local code to cloud
3. Sprint 3 – Hardware & Cloud integration
4. Sprint 4 – UI/UX Optimization & Debugging

CONTENTS

1. INTRODUCTION	
1.1 Project Overview	5
1.2 Purpose	5
2. LITERATURE SURVEY	
2.1 Existing problem	6
2.2 References	6
2.3 Problem Statement Definition	7
3. IDEATION & PROPOSED SOLUTION	
3.1 Empathy Map Canvas	8
3.2 Ideation & Brainstorming	9
3.3 Proposed Solution	10
3.4 Problem Solution fit	13
4. REQUIREMENT ANALYSIS	
4.1 Functional requirement	14
4.2 Non-Functional requirements	15
5. PROJECT DESIGN	
5.1 Data Flow Diagrams	16
5.2 Solution & Technical Architecture	17
6. PROJECT PLANNING & SCHEDULING	
6.1 Sprint Planning & Estimation	20
6.2 Sprint Delivery Schedule	22

7. CODING & SOLUTIONING

7.1 Feature 1	24
7.2 Feature 2	25

8. TESTING

8.1 Test Cases	26
8.2 User Acceptance Testing	26

9. RESULTS

9.1 Performance Metrics	27
-------------------------	----

10. ADVANTAGES & DISADVANTAGES

11. CONCLUSION

12. FUTURE SCOPE

13. APPENDIX

Source Code	29
-------------	----

1. INTRODUCTION

1.1 Project Overview

- To replace the static signboards, smart connected signboards 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, Restaurants) signs are also displayed accordingly.
- Different modes of operations can be selected with the help of buttons.

1.2 Purpose

- 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.

2. LITERATURE SURVEY

2.1 Existing problem

- Analysis of crash data has suggested a link between roadside advertising signs and safety.
- Research suggests that crash risk increases by approximately 25–29% in the presence of digital roadside advertising signs compared to control areas.
- On the other hand, static roadside advertising signs have not been linked with differences in the crash count.
- However, this finding is contrary to previous research that suggests differences in crash counts exist in the presence of static roadside advertising.
- The quantity and quality of available evidence limit our conclusion.
- Fixed object, side swipe and rear end crashes are the most common types of crashes in the presence of roadside advertising signs.
- In addition, drivers showed increased eye fixations and increased drifting between lanes on the road.

2.2 References

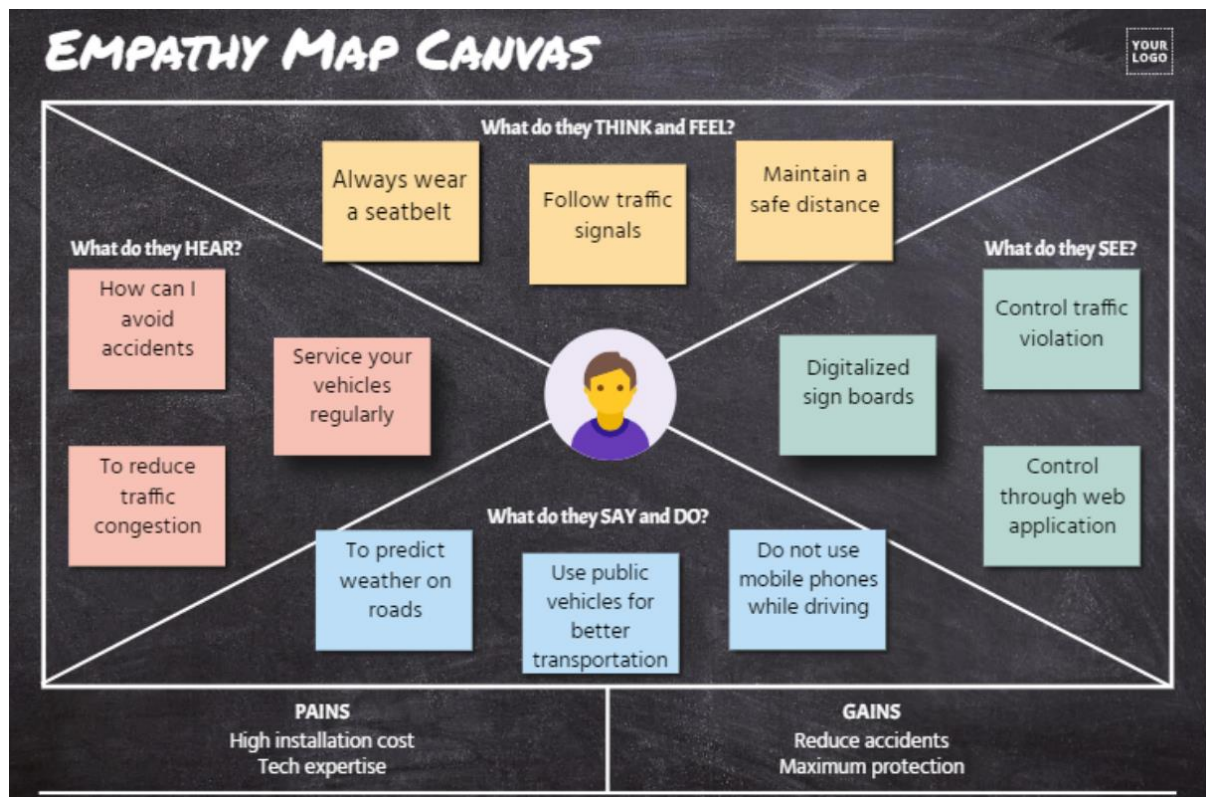
- [Cairney and Gunatillake, 2000; Sisiopiku et al., 2015](#)
- [Islam, 2015; Sisiopiku et al., 2015](#)
- [Yannis et al., 2013, Staffeld \(1953\) and Ady \(1967\)](#)

2.3 Problem Statement Definition

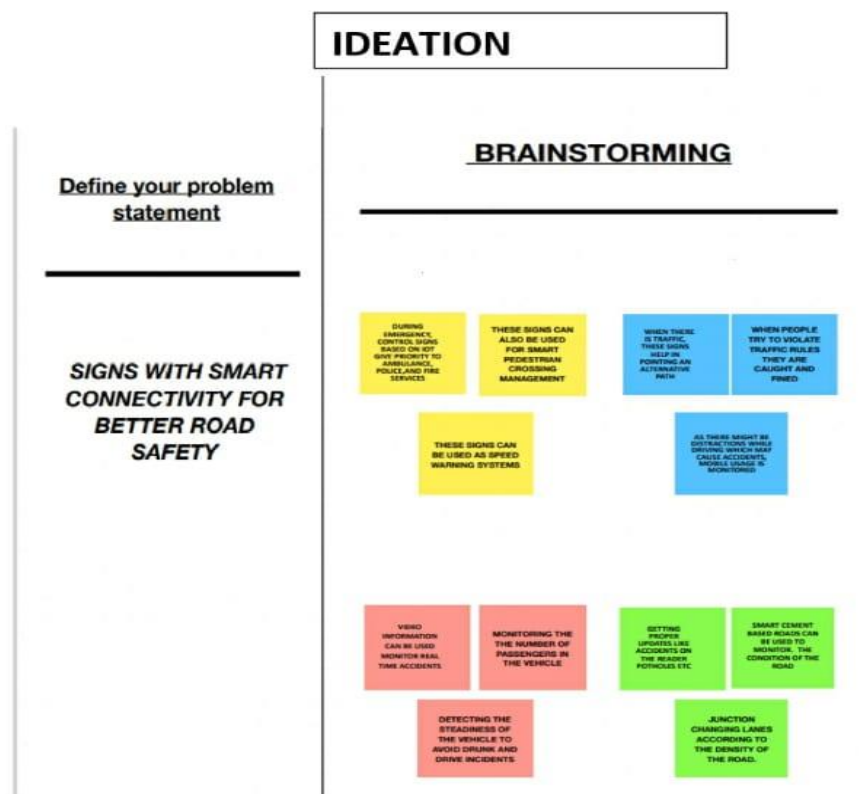
This project will replace the static boards to smart signed boards that will change the speed limits according to the weather climate and show diversion messages if there are accidents in the road and alert messages if there is hospital, schools or any roadworks.

3. IDEATION AND PROPOSED SOLUTION

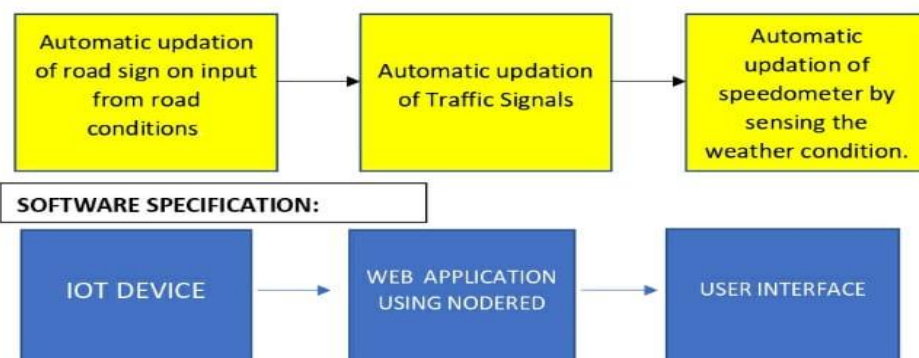
3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming Map



IDEAS



3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement	<p>To replace the static signboards, smart connected sign boards are used.</p> <p>These smart connected sign boards get the speed limitations from a web app using weather API and update automatically.</p> <p>Based on the weather changes the speed may increase or decrease</p> <p>Based on the traffic and fatal situations the diversion signs are displayed.</p> <p>Guide (Schools), Warning and Service (Hospitals, Restaurant) signs are also displayed accordingly.</p> <p>Different modes of operations can be selected with the help of buttons.</p>
2.	Idea description	<p>The weather and temperature details are obtained from the OpenWeatherMap 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</p>

S.No.	Parameter	Description
		hospital zones. By activating this button, either through the web application or the physical buttons, sign of the board can be changed accordingly, and the speed limit will also be set depending upon the zones. Also, the pedestrians are given an option to change the traffic signs if they want to cross the road. If the pedestrian presses the button that is present on the post at the end of the road, then the traffic will be analyzed immediately. 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.	Novelty	Generic Sign board for all applications that uses both buttons and web service for updation. Pedestrians are given the access to request the sign change of the signal to cross the road
4.	Customer Satisfaction	Diversion reasons will be displayed If there is no traffic, pedestrians can cross the street without waiting. Customer can reach the destination before the expected time

5.	Business Model	<p>Since APIs are used to actively monitor the customer's environment, this project employs a business strategy in which revenue will be generated on the basis of the length of time in which the customers actively interact with the product.</p> <p>This product is aimed to be free of cost to the public, but the revenue will be generated by selling this product to the government at a low cost, so there will be less accidents and the public will be aware of the discrepancies or accidents in the particular road. The public will also gain all the information about the road, even if they are checking for an alternate path because of some mishaps that happen on the roads and these functionalities will increase the value of the product in the global market.</p>
6.	Scalability of the Solution	<p>In the future, if any update is required either on the hardware or software side, it can be easily implemented. The hardware components can be directly interfaced with the microcontroller and small modifications can be made in the programming of the existing product. In case of the software, the website application has to be updated with the additional functionality by creating a new section for the updated hardware. So, this will not affect the existing functionality of the product and new functionality can be easily integrated. In addition, a separate circuit will be kept along with the hardware to detect any problem which informs the web application. Also, a notification will be sent to the product service department.</p>

3.4 Problem Solution fit

PREPARE SOLUTION FIT			
Define CS, fit into CC	1. CUSTOMER SEGMENT(S) <small>Who is your customer? i.e. working parents of 0-5 y.o. kids</small> Drivers, Passengers, Motorists	6. CUSTOMER CONSTRAINTS <small>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.</small> 1. Budget problem. 2. Pressure of how it will be received by the public.	5. AVAILABLE SOLUTIONS <small>What 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 unattractive to digital nativemaking</small> Send a petition to the collector to improve the lighting and rain water draining system for the roads
	2. JOBS-TO-BE-DONE / PROBLEMS <small>Which jobs-to-be-done (or problems) do you address for your customer? There could be more than one, and/or different ones.</small> 1. When there is high traffic, road diversions can be signalled through these signboards. 2. When the weather condition is extreme i.e heavy rains, typhoon, storms etc. The roads will be slippery so the speed limit can be shown on the signboards	9. PROBLEM ROOT CAUSE <small>What is the real issue 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 changing regulations.</small> 1. (i) Too many vehicles (ii) Roads being narrow. 2. (i) No proper draining system on the road. (ii) Less no street lights on the road.	7. BEHAVIOUR <small>What does your customer do to address the problem and get the job done? i.e. directly related: Find the right value based solution, calculate usage and benefits, indirectly associated: customers spend less time on volunteering work i.e. Groupwork</small> Directly related: Tries to find a solution to prevent these problem. Indirectly related: Tries to go slowly during extreme weather conditions and tries to avoid the road by knowing the accurate time the road will have high traffic.
Focus on J&P, fit into BE, understand RC	3. TRIGGERS <small>What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.</small> 1. Time is wasted while waiting in traffic. 2. A lot of accidents occur cause of the weather condition.	10. YOUR SOLUTION <small>If you are working on an existing business, write down your current solution first, list its benefits, and describe 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, eliminates and practices customer behaviour.</small> By making signs with smart connectivity for better road safety using IOT Watson, Node MCU	8. CHANNELS of BEHAVIOUR 8.1 ONLINE <small>What kind of actions do customers take online? Extract online channels from #2</small> People post on social media about the problems they faced while on the road which may lead to it trending and the Government taking action faster 8.2 OFFLINE <small>What kind of actions do customers take offline? Extract offline channels from #2 and use them for customer development.</small> Awareness about this issue can be made by giving out fliers, sending petitions, doing rallies etc
Identify strong TR & EM			Extract online & offline CH of BE

4 REQUIREMENT ANALYSIS

4.1 Functional Requirements

Following are the functional requirements of the proposed solution.

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 away drivers

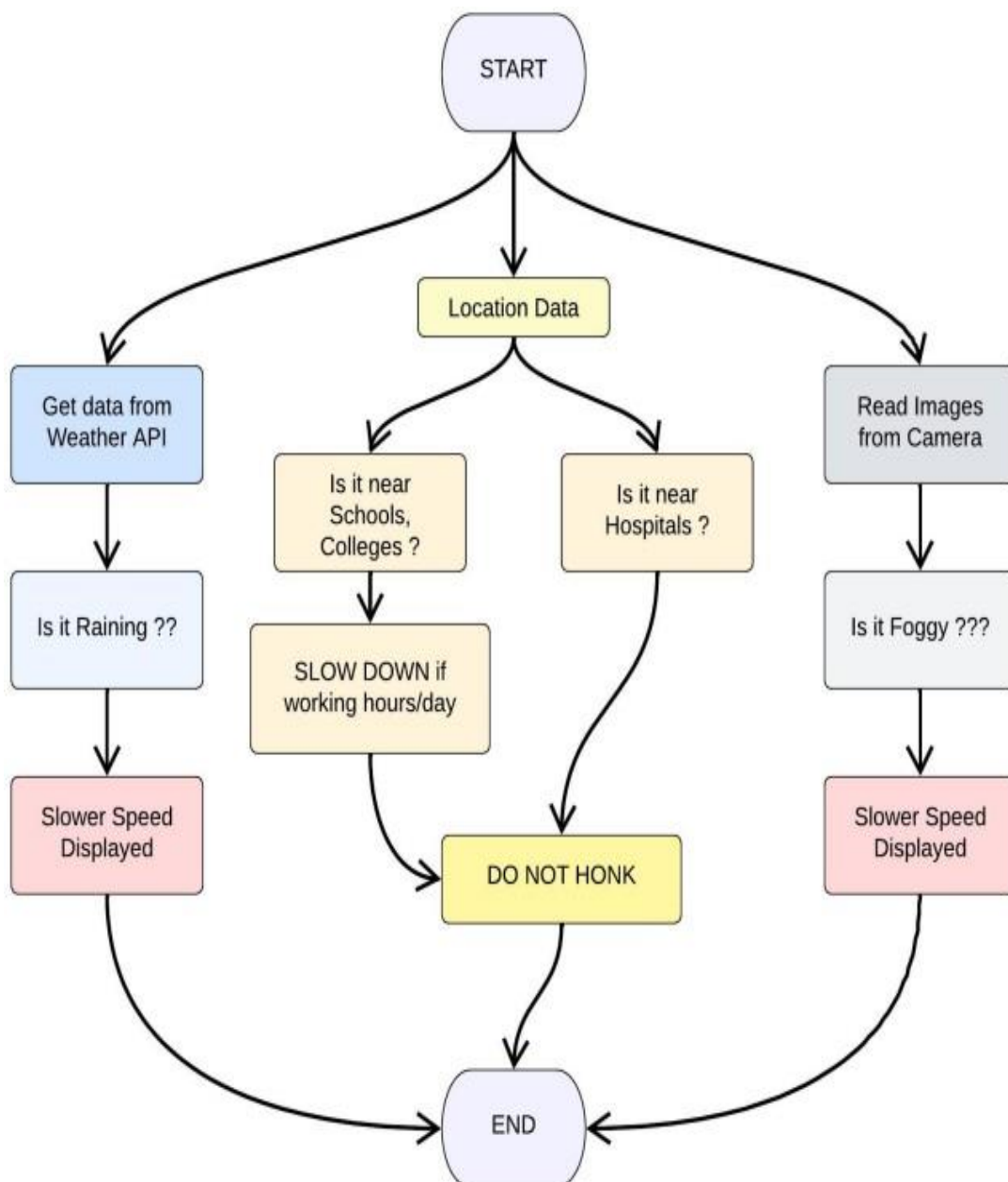
4.2 Non-Functional Requirements

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

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	Should be able to dynamically update with respect to 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 scale on servers horizontally.

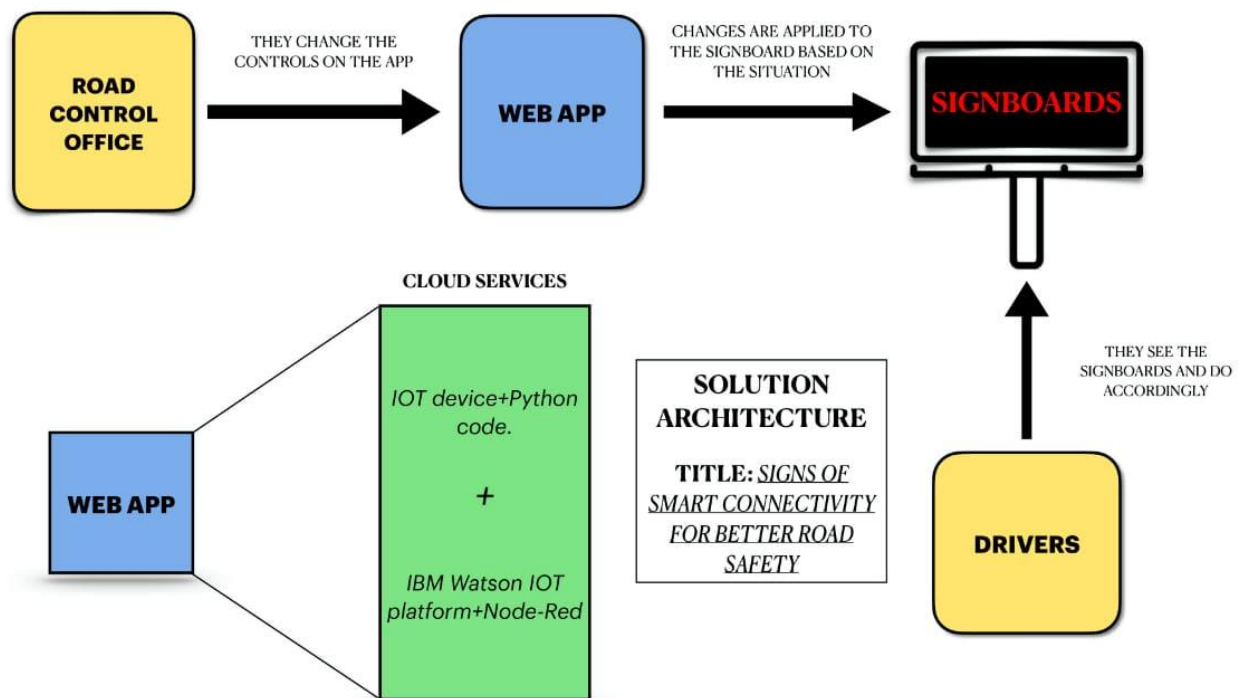
5 PROJECT DESIGN

5.1 Data Flow Diagram



5.2 Solution & Technical Architecture

Following is the Technical Architecture with slight change and is without the implementation of OpenCV API.



Following is the Solution Built

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1	User Interface	User can interact with the app using MIT App	HTML, CSS, JavaScript / Angular Js /React Js
2	Application Logic-1	Logic for a process in the application	Java / Python
3	Application Logic-2	Logic for a process in the application	IBM Watson STT service
4	Application Logic-3	Logic for a process in the application	IBM Watson Assistant
5	Database	Data Type, Configurations etc.	IBM Cloud
6	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
7	File Storage	File storage requirements	IBM Block Storage or Other StorageService or Local Filesystem
8	External API-1	Purpose of External API used in the application	Open Weather Map API
9	External API-2	Purpose of External API used in the application	IBM Watson Platform, Node - Red
10.	Infrastructure (Server / Cloud)	Application Deployment on Local System / CloudLocal Server Configuration: Cloud Server Configuration:	Local, Cloud Foundry, Kubernetes

Table-2: Application Characteristics:

S. No	Characteristics	Description	Technology
1.	Open-Source Frameworks	<i>OpenWeatherMap, NODE-RED, IBM WATSON, MIT App Inventor</i>	IoT, internet
2.	Security Implementations	<i>Powerful security system for everyone's peace of mind No access data Hackers cannot access network</i>	Firewall, Firebase, cyber resiliency, strategy
3.	Scalable Architecture	<i>EASY TO EXTEND THE NETWORK WITH THE AID OF THE BANDWIDTH OF THE NETWORK</i>	IBM Cloud
4.	Availability	<i>Available every time and everywhere 24/7 so long as the consumer is signed into the network.</i>	IBM Cloud
5.	Performance	<i>AIDS MASSIVE RANGE OF USERS TO USE TECHNOLOGY</i>	IBM Cloud

6 PROJECT PLANNING AND SCHEDULING

6.1 Sprint Planning & Estimation

Sprint	Functiona l Requireme nt (Epic)	User Stor y Nu mbe r	UserStory/Task	Story Points	Priori ty	Te a m Me m be rs
Sprint-1	IDE	USN-1	Installingall thesoftwareswhich isrequiredlikepythonIDE	2	High	Jana ni Mak eswa ri Gow ripri ya Myt hily
Sprint-1	Check ing the simula tion with condit ions	USN-1	Simulatingthecircuitsandexperime nting	2	High	Jana ni Mak eswa ri Gow ripri ya Myt hily
Sprint-2	Software	USN-2	- IBMWatsonIot - NodeRedintegration	2	High	Jana ni Mak eswa ri Gow ripri ya Myt hily
Sprint-2	Software	USN-2	Testthedeviceandworkflow.	2	High	Jana ni Mak eswa ri Gow ripri ya Myt hily

Sprint-3	Application Development	USN-3	Using MIT App Inventor create an App	2	High	Jana ni Mak eswa ri Gow ripri ya Myth ily
Sprint-3	Testing	USN-3	Testing the Application.	2	High	Jana ni Mak eswa ri Gow ripri ya Myth ily
Sprint-4	WEB UI	USN-4	User interface with the Software	2	High	Jana ni Mak eswa ri Gow ripri ya Myth ily

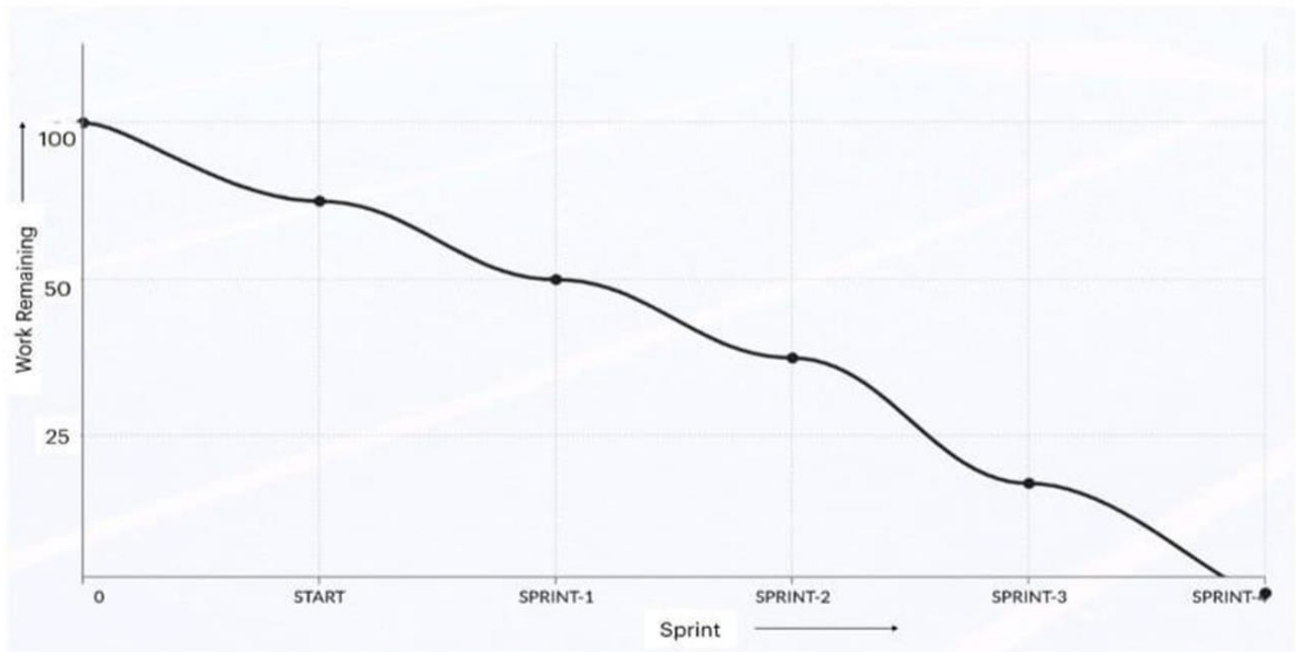
6.2 Sprint Delivery Schedule

Project Tracker, Velocity & Burndown Chart:

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

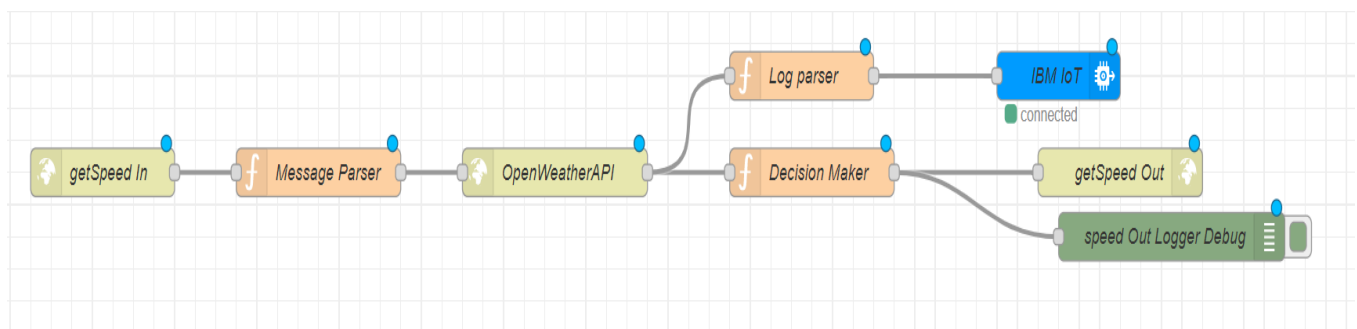
Burndown Chart:

Burndown chart:



7 CODING AND SOLUTIONING

7.1 Feature 1 - GET SPEED FOR GIVEN LOCATION & CLIMATE



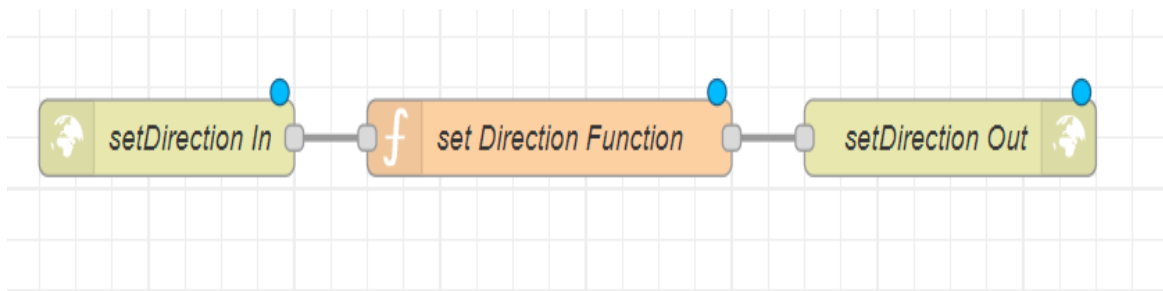
This part of Node RED flow accepts an http GET end point at **"/getSpeed"** from which the location, uid, hospital/school zone info are passed.

Message parser sets the required APIKEY for **OpenWeatherAPI** for the next block.

This data is then passed onto Decision Maker which makes all the decisions regarding the message to be output at the display and sends it as a http response.

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

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



This part of Node RED flow accepts an **http GET** end point at **"/setDirection"** from which the uid and direction information are passed by the respective authorities. **Set Direction** Function block adds the direction information to the database and returns the same as an http response. This data is sent to the microcontroller along with the **"/getSpeed"** path and the microcontroller displays it.

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 & 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 updating.
- School/Hospital Zone alerts

- **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

• SOURCE CODE

Python code

import requests as reqs

def get(myLocation,APIKEY):

 apiURL =

 f"https://api.openweathermap.org/data/2.5/weather?q={myLocation}&appid={APIKEY}"

 responseJSON = (reqs.get(apiURL)).json()

 returnObject = {

 "temperature" : responseJSON['main']['temp'] - 273.15,

 "weather" : [responseJSON['weather'][_]['main'].lower() for _ in
range(len(responseJSON['weather']))],

 "visibility" : responseJSON['visibility']/100, # visibility in percentage where 10km is
100% and 0km is 0%

 }

 if("rain" in responseJSON):

 returnObject["rain"] = [responseJSON["rain"][key] for key in responseJSON["rain"]]

 return(returnObject)

Python code

IMPORT SECTION STARTS

import weather

from datetime import datetime as dt

IMPORT SECTION ENDS

UTILITY LOGIC SECTION STARTS

def processConditions(myLocation,APIKEY,localityInfo):

 weatherData = weather.get(myLocation,APIKEY)

 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 zo...

```

Python code

IMPORT SECTION STARTS

import brain

IMPORT SECTION ENDS

USER INPUT SECTION STARTS

myLocation = "Chennai,IN"

APIKEY = "92eedd4b0b4cd6c543c365f562a59ab3"

```

localityInfo = {
    "schools" : {
        "schoolZone" : True,
        "activeTime" : ["7:00","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

print(brain.processConditions(myLocation,APIKEY,localityInfo))

MICRO-CONTROLLER CODE ENDS

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

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

myConfig = {
    "identity" : {
        "orgId" : "epmoec",
        "typeId" : "testDevice",
        "deviceId" : "device0"
    },
    "auth" : {
        "token" : "?-KDXUPMvDo_TK2&b1"
    }
}

# 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
    client.disconnect()
    time.sleep(1)
```

