K.L.N. COLLEGE OF ENGINEERING

(AN AUTONOMOUS INSTITUTION, AFFILIATED TO ANNA UNIVERSITY, CHENNAI)

POTTAPALAYAM – 630 612.

SIVAGANGAI DISTRICT, TAMILNADU, INDIA.

SIGNS WITH SMAT CONNECTIVITY FOR BETTER ROAD SAFETY

IBM PROJECT REPORT

Submitted By

TEAM ID : PNT2022TMID11468

GAUTHAMPRABHAKAR J	**910619104020**
DANIEL AMIRTHARAJ P	**910619104013**
ASHIKROSHAN I	**910619104009**
ABISHEK A G	**910619104003**

In partial fulfillment for the award of the degree

BACHELOR OF ENGINEERING

In

COMPUTER SCIENCE AND ENGINEERING

TABLE OF CONTENT

5. PROJECT DESIGN

5.1. MILESTONE AND ACTIVITY LIST

5.2. SPRINT DELIVERY PLAN

6. PROJECT DEVELOPMENT PHASE

6.1. DELIVERY OF SPRINT – 1

6.2. DELIVERY OF SPRINT – 2

6.3. DELIVERY OF SPRINT – 3

6.4. DELIVERY OF SPRINT – 4

7. CONCLUSION

ABSTRACT

The number of vehicles on the road today is rising. It is difficult for society to control traffic bottlenecks and accidents on the roads around the world as a result. Machine learning (ML) techniques, a type of artificial intelligence (AI), are highly useful for enhancing the functionality of the overall road safety management system. Numerous practical applications of AI are employed to transform any system into a smart system. The Smart Road Traffic Management System (SRTMS) can quickly identify the impact that unpredictable changes have on traffic safety. The SRTMS not only notifies the appropriate authorities of risky driving habits, but also identifies them. The Internet of Things (IOT) is a useful tool for real-time monitoring of human activity. Sensors are a frequent component of Internet of Things (IOT) devices and nodes and are used to recognise and respond to electrical and other signals. The most popular technology today for automating transactions, or the sharing or exchange of information between iot devices or nodes, is Block chain (BC). Information sharing on the network is made possible by BC technology in a decentralised, secure, persistent, anonymous, appropriate, and reliable way. Block chain aims to coordinate communication between nodes without the help of a third party or intermediary organisation thanks to consensus algorithms and smart contracts. AI has the potential to develop robots that are both intelligent and capable of making decisions, much like human minds. In order to address traffic congestion, road accidents, and information dissemination to all stakeholders, this article suggests the SRTMS paradigm.

CHAPTER – 1

INTRODUTION

The primary method of connecting cities and villages is through roads. Due to how simple it is to go by road, cars are now the most common mode of transportation. The vehicle accidents (Vas) are more likely now since there are more automobiles on the road. When travelling, especially in inclement weather, one never knows what may happen on the next road (BWC). Due to the poor sight, driving can be challenging in this circumstance and may result in an accident. Additionally, it was shown that in BWC, informational delays might lead to multiple vehicle accidents (MVCs), whichare dangerous.

According to one research by the Islamabad police, there were 9582 accidents between 2016 and 2017 involving 11,317 cars nationwide in Pakistan, which resulted in 5047 fatalities and 12,696 injuries.

Road safety regulations are changing as a result of digital technology like the Internet of Things (IoT). Around the world, several technological projects are being done to create smarter, safer roadways that can communicate with both vehicles and pedestrians. Several technology-based solutions have been created upon the presumption that accidents may be avoided by providing the driver with information about vehicle technology. The Internet of Things provides the foundation for the newest technologies that academics are developing (IoT). Data is the key to IoT. The world is starting to value data as a resource.

IOT has being embraced by many sectors and companies to enhance communication, manufacturing, energy, and health care performance while reducing mistakes. In its "Save LIVES: Road Safety Technical Package," theWHO lists many actions that can be taken with little financial impact. Realizing economic systems for "monitoring road safety by enhancing data systems" is a cornerstone of these measures.

Mobile phone applications identify the speed limit depending on environmental conditions using built-in sensor data.

The main contributions of this research are

1. A brief overview of the state of the art in pre-accident and post-accidentmodels, frameworks, and techniques;
2. Identifying and disclosing limitations in prior research on accidentdetection;
3. The notion of a smart road with event sensing capabilities, as well as itsimplementation and testing through numerous tests;
4. An innovative and contemporary technique for communicating with adjacent vehicles and EOCs and promptly detecting accidents is demonstrated.

If an incident is not reported to an EOC in a timely manner, there maybe an increased risk of fatalities, injuries, and other harm. By delivering timelyaccident information via an automated process, lives can be saved. Additionally, an alarm system and speedy car collision detection are needed to safeguard approaching automobiles from an MVC. In order to prevent accidents, a number of strategies have been used in advanced vehicles (AVs). A threat of an accident is identified using either Smartphone sensors or sensors fitted in automobiles. Accelerometers, smoke detectors, IR obstacle sensors, proximity sensors, and biosensors have all been employed by earlier researchers to identify accidents.

1.1.PROJECT OVERVIEW:

This project's main objective is to help people automate the roadways by providing them with access to a web app that lets them monitor variables like temperature, speed limit, and visibility on the roads. They also offer services for displaying signage for eateries, hospitals, and school directories.

1.2.PURPOSE :

The fields of accident prevention and accident alarms are the subject of substantial research by numerous academics and professionals. In order to increase safety and prevent accidents, many different techniques are applied. For ease of reference, stand-alone, cooperative, and hybrid strategies are separated in the accident detection and avoidance literature. While stand-alone approaches use sensors for accident avoidance and detection, such as radar and light detection and ranging, cooperative systems rely on V2X technology and hybrid approaches (LIDAR).

CHAPTER – 2

IDEATION PHASE

2.1. LITERATURE SURVEY:

A literature survey or a literature review is that section which shows the various analyses and research made in the field of your interest and the results already published, taking into account the various parameters of the project and the extent of the project**.**

The following papers are studied in the following survey:

[1] "The Role of Blockchain, AI and IoT for Smart Road Traffic Management System" by Ashish Sharma ; Yogesh Awasthi ; Sunil Kumar at (2020 IEEE).

Nowadays vehicles are increasing on the road. Due to this, it is a challenge for society to manage traffic jams and road accidents all over the world. Artificial Intelligence (AI) such as Machine Learning (ML) algorithms are very helpful to improve the performance of the overall road safety management system. AI is used for many real-world applications to make any system be a smart system. The Smart Road Traffic Management System (SRTMS) easily recognizes the influence occurs for random changes on road safety. The SRTMS detects the unsafe driving patterns as well as convey the information to the respective authorities. The Internet of Things (IoT) is a boon technology to observe human activities in real-time. IoT devices or nodes are composed of sensors that are commonly utilized to identify and reply to electrical and other signals. Currently, Blockchain (BC) is the most trending technology to automate transactions, which means sharing or exchange of information between the IoT devices or nodes. BC technology facilitates for sharing of information on the network is decentralized, secure, persistent, anonymity, suitability and trustworthy manner. With consensus algorithms and smart contracts, Blockchain holds to manage communication among nodes without the involvement of a third-party or intermediary body.

Simultaneously, AI has the ability to offer intelligent and decision-making machines similar to human beings' minds. This paper proposes the SRTMS model for solving the road accident, traffic jam and disseminate the information to all stakeholders. This proposed model is a combination of most trending technologies such as AI, BC, and IoT.Keywords— Machine Learning (ML), Blockchain (BC), Consensus Algorithms, Smart Contract, Internet of Things (IoT).

[2]"Communication system for Intelligent Road Signs network" by Janusz Gozdecki ; Krzysztof Łoziak ; Andrzej Dziech ; Wojciech Chmiel ; Joanna Kwiecień ; Jan Derkacz ; Piotr Kadłuczka at (2019 IEEE).

The way of providing the end user with an accurate data regarding the current road conditions is one of the very important components in the area of Intelligent Transportation Systems (ITS). In such cases one of the possibilities is to display the adaptive content on a dedicated road sign. The most important issue in the process of building the trust between the road signalling infrastructure and the end user is the information significance and its value. The ongoing NCBiR project - InZnak - aims to introduce a new type of the road signalling subsystem which relies on intelligent road signs equipped with variety of sensors and adaptive led displays. Sensors feed the autonomous algorithms with data necessary to take decisions on how to react to current road conditions. The proposed system typically consists of a few road signs communicating with each other and exchanging measured data: weather conditions, road surface condition, traffic volume, avg. vehicles speed, detected road events, etc. On the basis of the information exchanged between road signs, each of them runs the autonomous algorithm to process that data and computes the current status of road section driving conditions. The InZnak project focuses on the problem of the traffic control using intelligent autonomous road signs. The process of the speed limit determination is often complicated and in many cases it is defined by numerous legal standards. In general, speed determination should take into account not only the technical aspect, but also social and legal aspects, which makes this process especially

difficult. This paper presents the InZnak communication system architecture with the deployed prototype and its current status of integration. Keywords— intelligent road sign, traffic moderation, intelligent transport system, ITS communications.

[3] "J. Kotus, A. Czyżewski, Counting and tracking vehicles using acoustic vector sensors", 176th Meeting of the Acoustical Society of America and 2018 Acoustics Week in Canada, Victoria, Canada.

We show how to count vehicles and determine their direction of travel using an acoustic vector sensor. Assumptions of how to use the spatial distribution of acoustic intensity determined with the help of the integrated 3D intensity probe are discussed. An intensity probe developed by the authors was used in the experiments. The algorithm's capabilities are presented in relation to the noise characteristics of moving vehicles. Algorithm optimization is based on measurements of the sound intensity emitted by the vehicle under controlled conditions. For this purpose, a test setup was set up with measuring devices installed along roads with varying traffic. Reference data on the number of vehicles and traffic direction were generated using recorded video and a reference traffic analyzer using lidar technology. It has been shown that the developed acoustic method can contribute to improving the effectiveness of current vehicle counting systems with inductive loops or Doppler radars. [Project funded by the Polish National Center for Research and Development (NCBR) from the European Regional Development Fund within the operational program Innovative Economy Number POIR.04.01.04-00-0089/16 "INZNAK – Intelligent Road Signs..." .

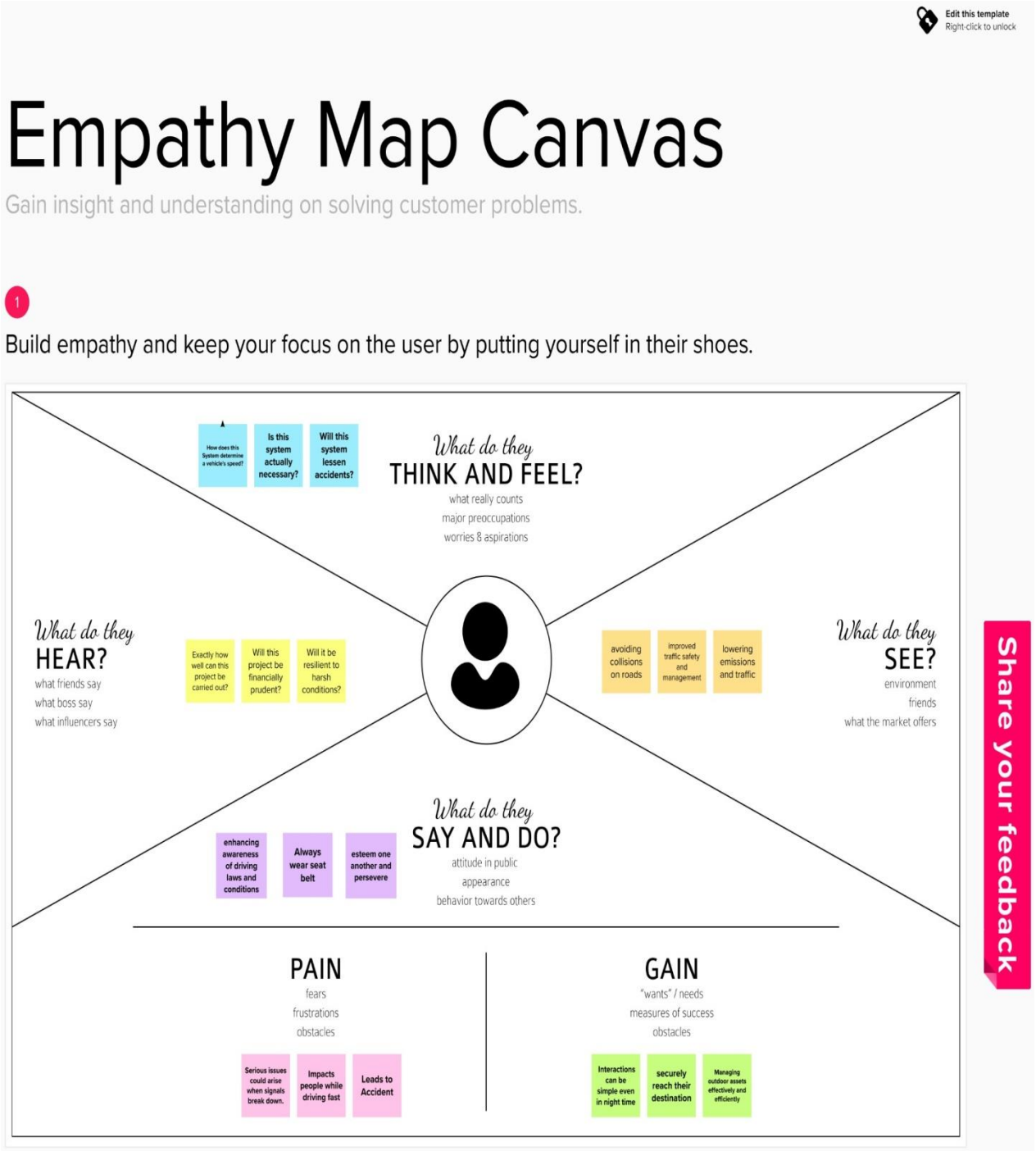
[4] "An IoT Architecture for Assessing Road Safety in Smart Cities" by Abd-Elhamid M. Taha ; Published at 19 November 2018.

The Safe System (SS) approach to road safety emphasizes safety-by-design through ensuring safe vehicles, road networks, and roadusers. With a strong motivation from the World Health Organization (WHO), this approach is increasingly adopted worldwide.Considerations in SS, however, are made for the medium-to-long term.

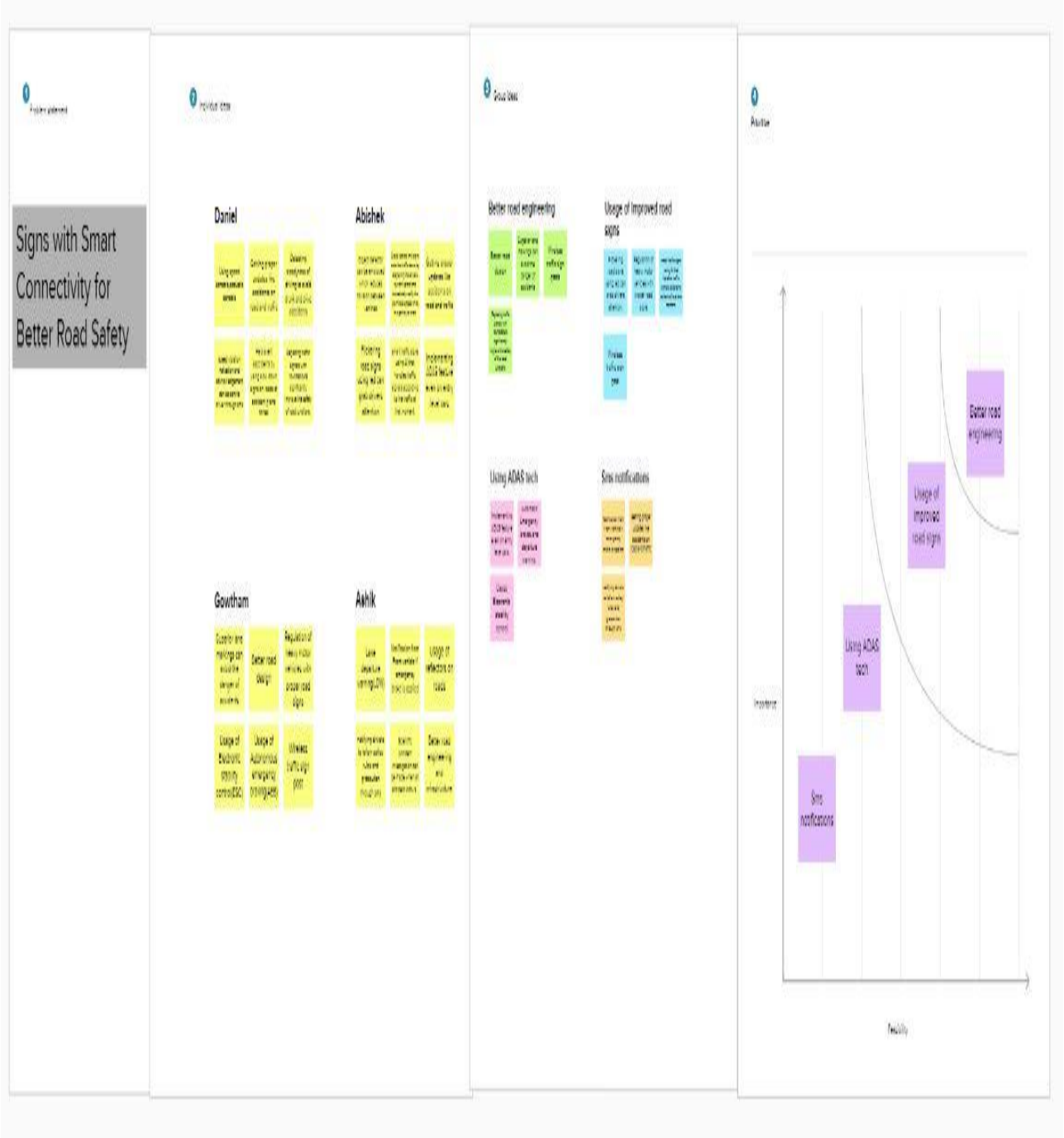
Our interest in this work is to complement the approachwith a short-to-medium term dynamic assessment of road safety. Toward this end, we introduce a novel, cost-efective Internetof Tings (IoT) architecture that facilitates the realization of a robust and dynamic computational core in assessing the safety of aroad network and its elements.

In doing so, we introduce a new, meaningful, and scalable metric for assessing road safety. We alsoshowcase the use of machine learning in the design of the metric computation core through a novel application of Hidden MarkovModels (HMMs). Finally, the impact of the proposed architecture is demonstrated through an application to safety-based route planning.

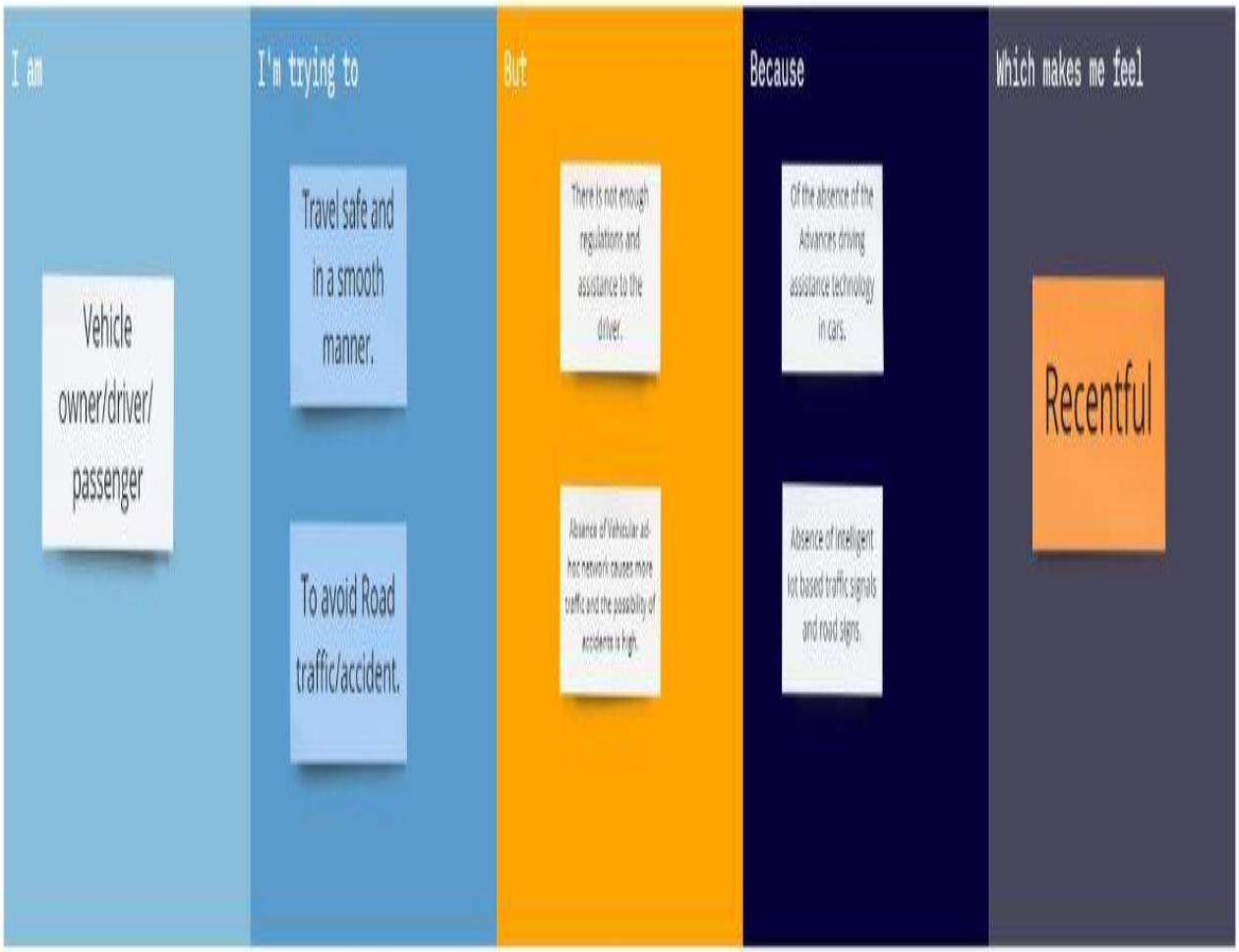
2.2. EMPATHY MAP:



2.3. IDEATION AND BRAINSTORMING:



2.4. PROBLEM STATEMENT:



CHAPTER - 3

PROJECT DESIGN PHASE – I

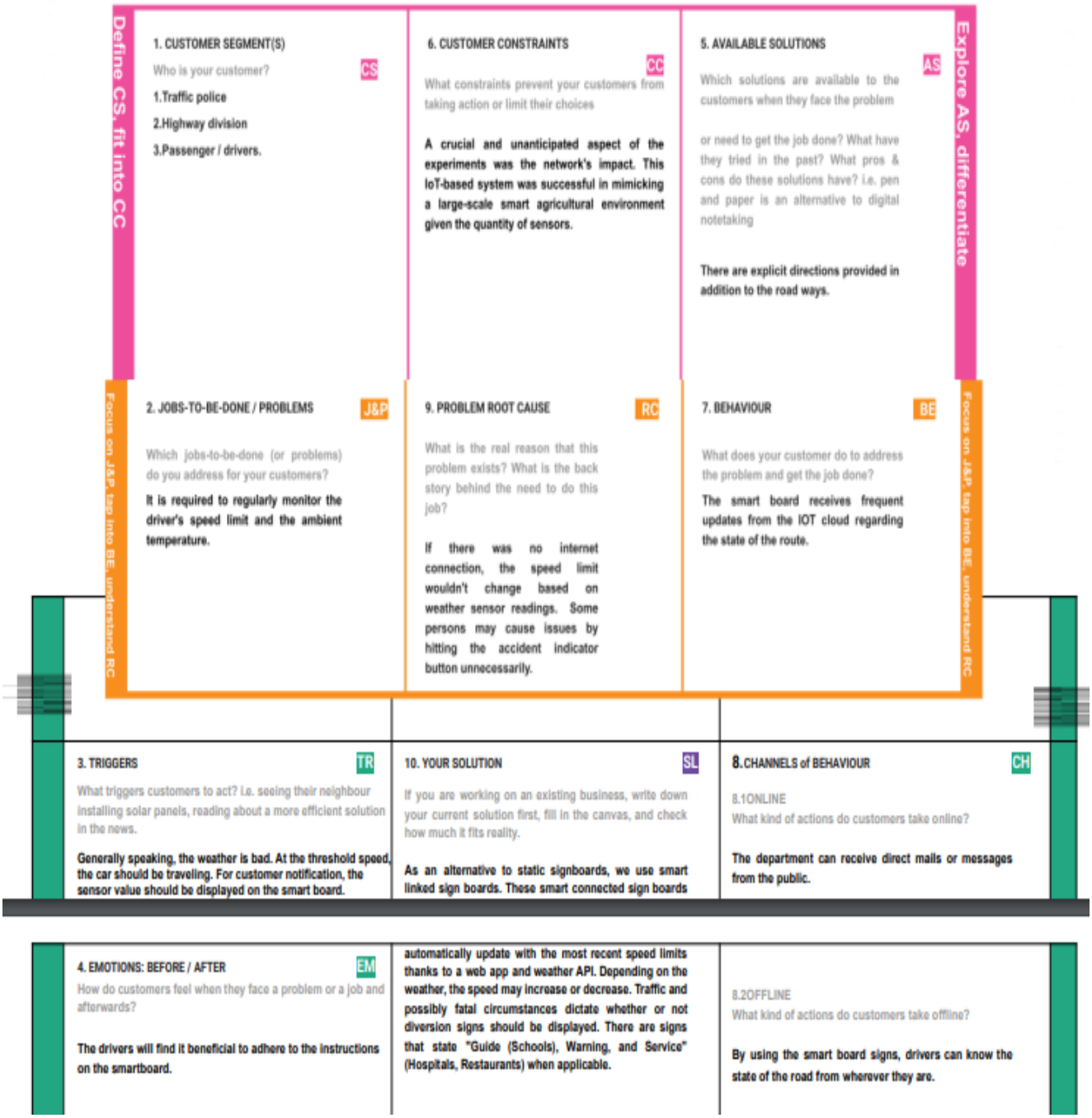
3.1. PROPOSED SOLUTION:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	•Traffic management is an essential part of modern mobility, and traffic signals help optimize the existing network in the best possible way. •It monitors and controls various modes of traffic in order to avoid congestion and to improve traffic flow. •People fail to understand signs and violation of rules. •Dynamic boards show all the information while static boards cannot.
2.	Idea / Solution description	The OpenWeatherMap API is used to receive the weather and temperature information.The speed limit will be automatically changed based on this information and the current weather. Additionally, specifics on any incidents and traffic bottlenecks encountered on the specific road are acquired. Consequently,The traffic is then diverted, followed by a modification in

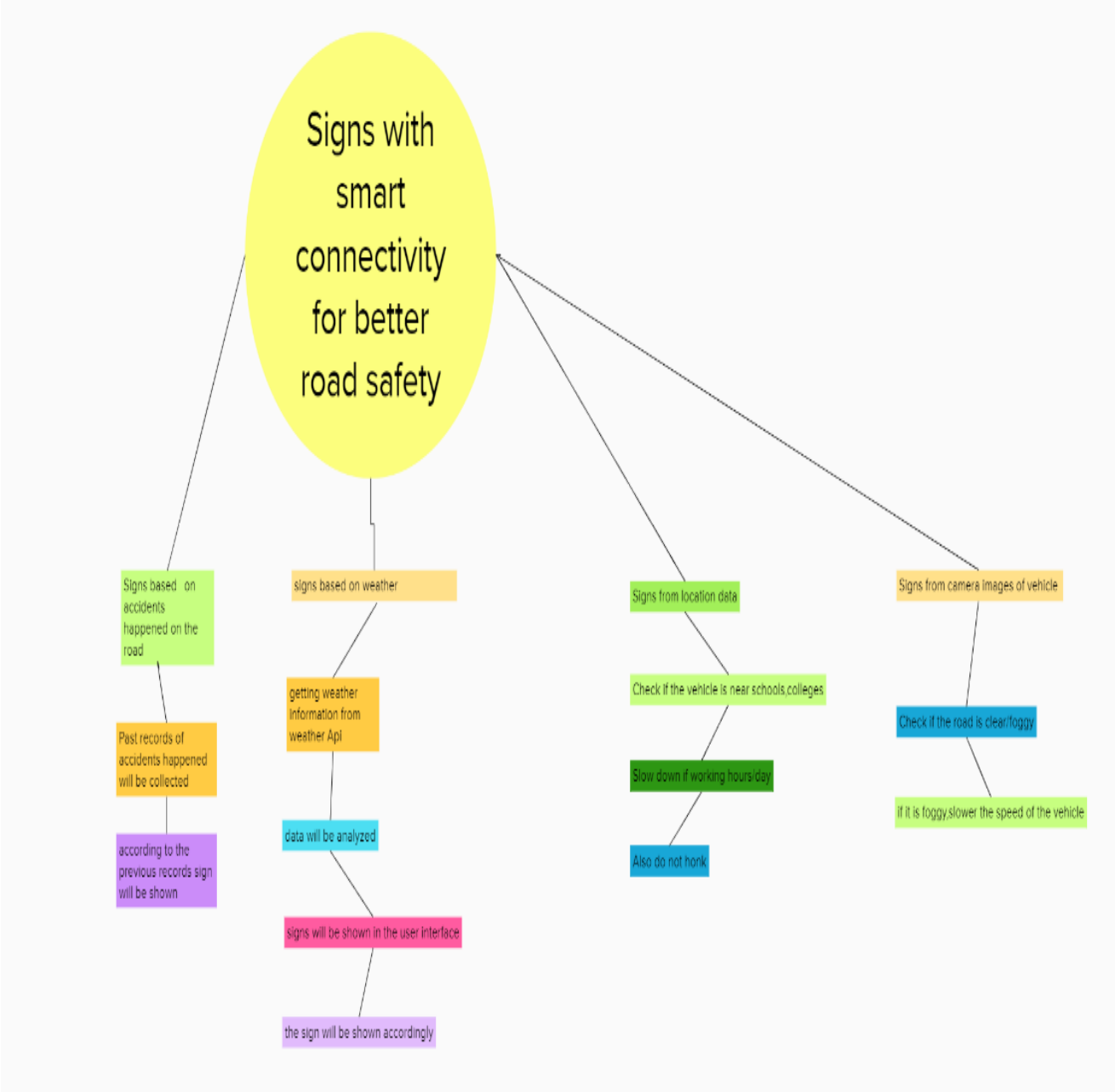
		the map's route.Therefore, some buttons will be installed on the traffic sign board that will be employed to make it more universal; each button will have a features like the ability to modify the warning indicators, which There will be specific, separate signs for the school and hospital zones. By activating this button—either physically or through a web application—the board's sign can be adjusted appropriately, and the speed limit will be established in accordance with the zones. Additionally, if a pedestrian wants to cross the road, they have the option of changing the traffic signs. The traffic will be promptly assessed if the pedestrian presses the button that is located on the post at the end of the road. As a result, the traffic signal's sign will change. In consequence, even without pedestrians, this lessens the frequency with which traffic signs are changed.
3.	Novelty / Uniqueness	• Provides drivers on the road with more thorough information. • Generic Sign board for all applications that uses both buttons and web service for updation
4.	Social Impact / Customer Satisfaction	• Diversion reasons will be displayed • Real time information management • A decline in accident rates • To provide more efficient travel to the

		people .
5.	Business Model (Revenue Model)	• Promoting the project to highway departı • LED signal lamp with a 5-year performance warranty that complies with EN12368 European standards.
6.	Scalability of the Solution	

3.2. PROBLEM SOLTTION FIT:



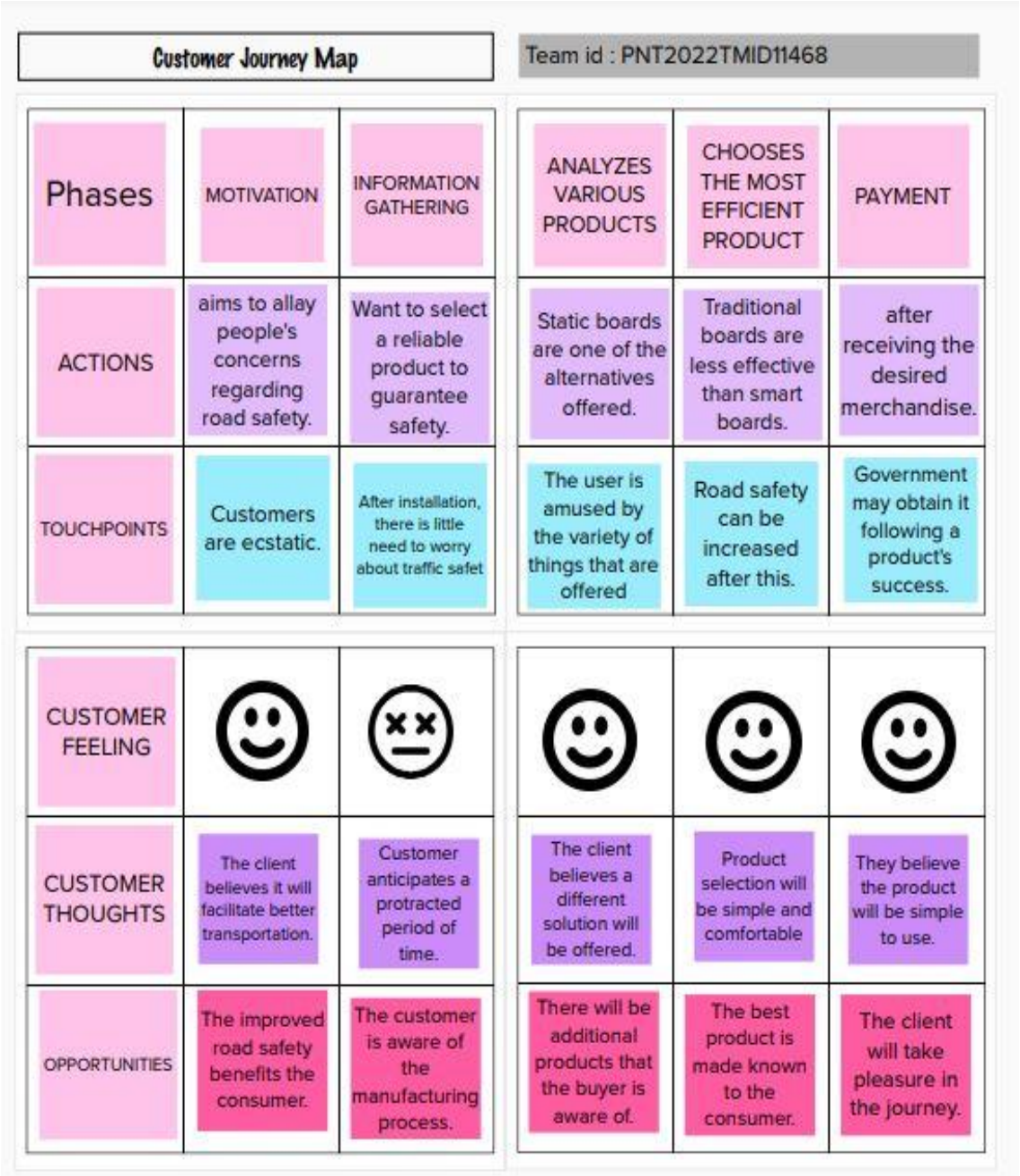
3.3. SOLUTION ARCHITECTURE:



CHAPTER – 4

PROJECT DESIGN PHASE – II

4.1. CUSTOMER JOURNEY:



4.2. REQUIREMENT ANALYSIS:

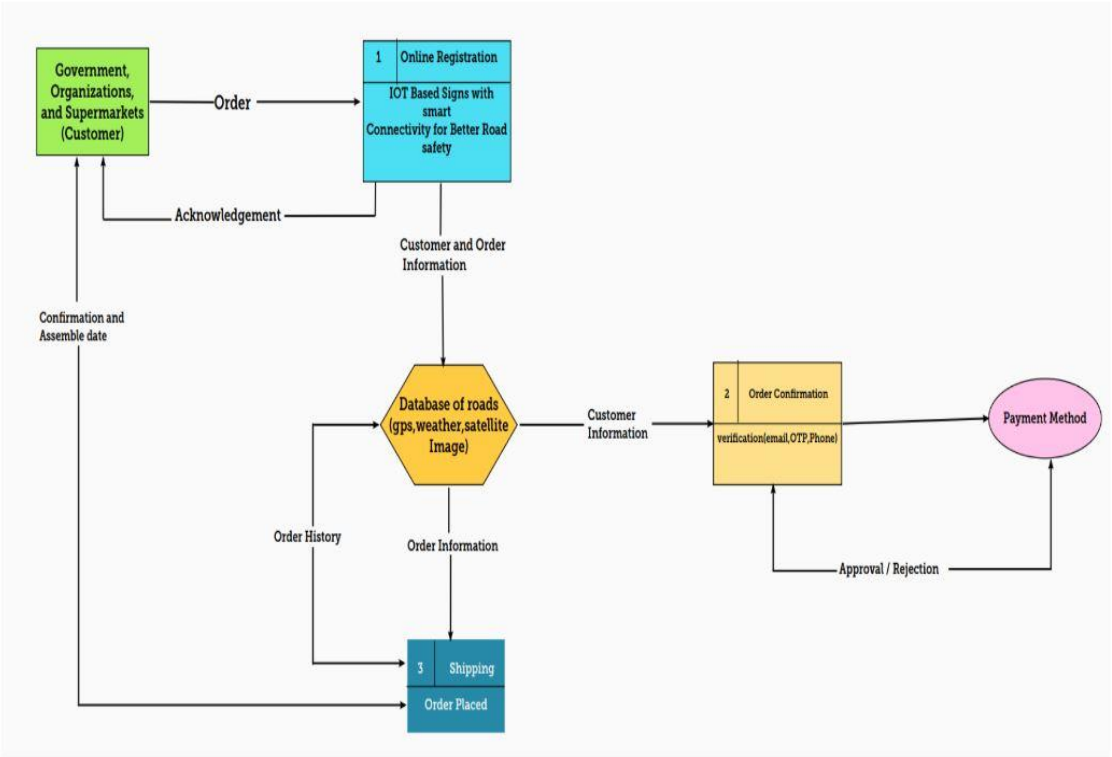
FUNCTIONAL REQUIREMENTS:

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Visibility	Sign boards should be designed with LEDs that are brightly coloured and capable of grabbing the attention of vehicles, but they shouldn't be too distracting or blinding because it could cause accidents.
FR-2	User Need	To reduce user confusion when travelling, smart sign boards should be put more frequently where they are needed and less frequently where they are not.
FR-3	User Understanding	The signs should be big, clear, and legible to help the motorist understand them. They can also incorporate images to help the driver understand the instructions even better.
FR-4	User Convenience	The display should be sufficiently large to be readily visible even from a distance.

NON – FUNCTIONAL REQUIREMENTS:

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	When necessary, it should be able to upgrade and update.
NFR-2	**Security**	So that no one else can hack it and display their own directions, it needs to have a strong security mechanism.
NFR-3	**Reliability**	It ought to be able to present the data accurately and without any mistakes.
NFR-4	**Performance**	When specific weather or traffic issues arise, it need to be able to update itself automatically.
NFR-5	**Availability**	For the benefit of the user, the driver, it should be accessible around-the-clock.
NFR-6	**Scalability**	It need to have the flexibility to quickly adapt to changes in requirements and make upgrades as necessary.

4.3. DATA FLOW DIAGRAM:



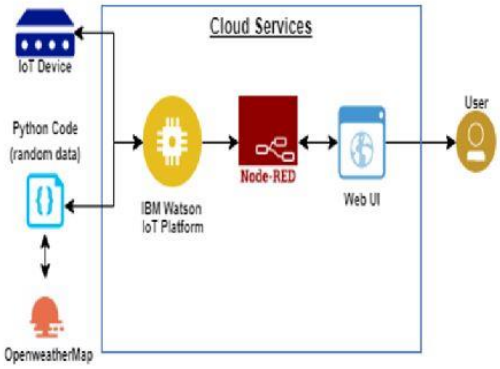
User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	I can get my speed constraint utilizing climate application	I can get speed restrictions	High	Sprint-1
		USN-2	As a client, I can enroll for the application by entering email, secret phrase, and confirming my secret phrase.	I can get to my account/dashboard	Medium	Sprint-2
		USN-3	As a client, I can increment or decrement my speed as indicated by the weather conditions change	I can increment or decrement my speed	High	Sprint-1
		USN-4	As a client, at any point I could get my traffic redirection signs relying upon the traffic and the lethal circumstances	I can get to know my traffic status ahead in movement	Medium	Sprint-1
	Login	USN-5	As a client, I can sign out from the dark climate map by entering email and secret key	I can get to the application through my Gmail login	High	Sprint-2
	Interface	USN-6	As a client the connection point ought to be straightforward and effectively open	I can access the point of interaction without any problem	High	Sprint-1
Customer (Web user)	Data generation	USN-7	As a client I utilize open climate applications to access the information in regards to the weather conditions changes.	I can get to the information concerning climate through the application	High	Sprint-1
Administrator	Problem solving/ Fault clearance	USN-8	As an in authority charge for the legitimate working of the sign sheets need to keep up with it through occasional observing	Authorities can screen the sign sheets for legitimate work.	Medium	Sprint-2

4.4. TECHNOLOGY ARCHITECTURE:

Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2



Guidelines:

- *1. To override the static traffic signs, use the smart sign indicator boards. These connected smart signs use the Weather API to get speed limits from the web app and update automatically Depending on the weather, the speed may increase or decrease and the signs are posted depending on traffic and life-threatening situations. Orientation (schools), warning and service (hospitals, restaurants) signs are displayed accordingly Different operating modes can be selected using buttons)*
- *2. The IBM Watson IoT Platform acts as a intermediate to connect web applications to IoT devices, hence the creation of the IBM Watson IoT Platform.*
- *To connect an IoT device to the IBM cloud, create a device in the IBM Watson IoT platform and obtain device credentials*
- *Configure the connection security and create API keys for Node-RED services to access the IBM IoT platform.*

Table-1 : Components & Technologies:

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

9.	External API-2	Purpose of External API used in the application	Aadhar API, etc.
10.	Machine Learning Model	Purpose of Machine Learning Model	Object Recognition Model, etc.
11.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	Local, Cloud Foundry, Kubernetes, etc.

Table-2: Application Characteristics:

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

CHATER – 5

PROJECT DESIGN

5.1. MILESTONE AND ACTIVITY LIST:

IBM Cloud Services: (Aug 22-Sep02)

Among all the IOT product development stages, cloud services is an important stage for building the best IOT Products. The development team is responsible for building web and mobile-based applications for control of the functionality of products in real-time.

Open Weather Map: (Sep 05-Sep 10)

The Open Weather Map is a service that provides weather data, including current weather data, forecasts, and historical data to the developers of web services and mobile applications. We analyzed the behavior of the metrics for the open weather map model.

Node-Red: (Oct 1-Oct 11)

Node-Red is a programming tool for wiring together hardware devices, API and online services in new and interesting ways. It provides a browser-based editor that makes it easy to wire together flows using the wide range of nodes in the palette that can be deployed to its run time in a single-click.

Python Script: (Sep 20-Sep 27)

The primary objective of running python on an IOT device that pops up in mind is grabbing the Arduino UNO from the table. Python is pre-installed in the operating system, and the only objective left is to write the coding script.

Hardware and Application: (Sep 10-Sep 17)

Hardware part consist of sensors and boards. The Navigational sensor provides a precise geo-spatial orientation of the vehicle as well as trends in driving behavior. The ODAWS Algorithm is used to interpret sensor data and offer real-time notifications to the driver , boosting road safety. Product Hardware Identification is one of the most important parts of IOT product development stages. The development team with great and in-depth knowledge of diverse types of IOT boards, sensors, and connector devices will get a huge success in IOT product development. A traffic signal is used as an instrument device that indicates the road user to act according to the displayed sign. Sensors installed in strategic locations can use IOT technology to collect data on congestion, moving vehicles away from these locations. IOT Big Data solutions can analyze this information, determine alternative routes, and improve traffic signaling to reduce congestion.

Final Deliverables: (Oct 25-Nov 15)

Our project “Signs with smart connectivity for better road safety” in the domain of internet of things (IOT) will soon prove its potential in vehicle maintenance , navigation, monitoring leading to improve transportation on the given sprint delivery plan by using our followed task and assignments like Arduino UNO, IBM cloud services, Open weather map, Node-red, Python IDLE, And Sensor.

5.2. SPRINT DELIVERY PLAN:

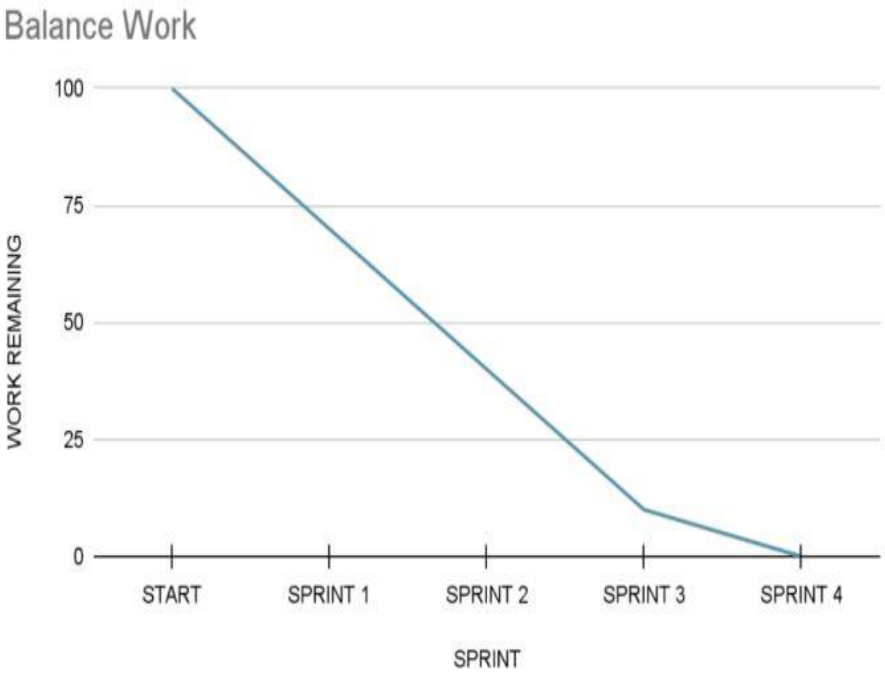
PRODUCT BACKLOG, SPRINT SCHEDULE, AND ESTIMATION:

Sprint	Functional Requirement (Epic)	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Initialization of Resources	Create and initialize accounts in various public APIs like OpenWeather API.	1	Low	Gautham Prabhakar J Daniel Amirtharaj P Ashik Roshan I Abishek A G
Sprint-1	Local Server/Software Run	Write a Python program that outputs results given the inputs like weather and location.	1	Low	Gautham Prabhakar J Daniel Amirtharaj P Ashik Roshan I Abishek A G
Sprint-2	Push the software to cloud	Push the code from Sprint 1 to cloud so it can be accessed from anywhere	2	Medium	Gautham Prabhakar J Daniel Amirtharaj P Ashik Roshan I Abishek A G
Sprint-3	Hardware initialization	Integrate the hardware to be able to access the cloud functions and provide inputs to the same.	2	High	Gautham Prabhakar J Daniel Amirtharaj P Ashik Roshan I Abishek A G
Sprint-4	UI/UX Optimization & Debugging	Optimize all the shortcomings and provide better user experience	2	Medium	Gautham Prabhakar J Daniel Amirtharaj P Ashik Roshan I Abishek A G

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	31 Oct 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	07 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	14 Nov 2022

Burndown Chart:



CHAPTER – 6

PROJECT DEVELOPMENT PHASE

6.1. DELIVERY OF SPRINT – 1:

BRAIN.PY:

```
import weather
from datetime import datetime as dt
def processConditions(myLocation,APIKEY,localityInfo):
    print("Requesting weatherData...")
    weatherData = weather.get(myLocation,APIKEY)
    print ("WeatherData:",weatherData)

    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
    })
```

MAIN.PY:

```
import brain


myLocation = "Madurai,IN"
APIKEY = "7a547a97f79ee73629edd98723ab36fe"


localityInfo = {
    "schools" : {
        "schoolZone" : True,
```

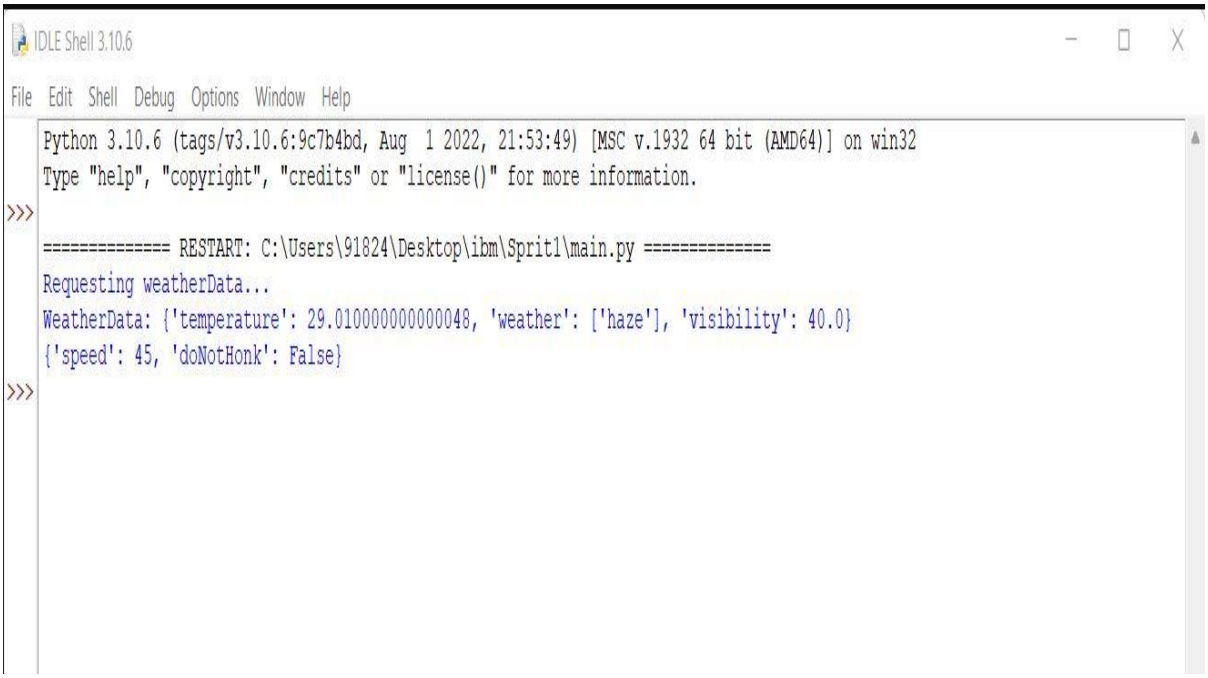
```
"activeTime" : ["7:45","17:30"] # schools active from 7 AM till 5:30 PM
        },
    "hospitalsNearby" : False,
    "usualSpeedLimit" : 45 # in km/hr
}


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

WEATHER.PY:

```
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)
```

OUTPUT.PNG:



6.2. DELIVERY OF SPRINT – 2:

BRAIN.PY:

```
import weather
from datetime import datetime as dt
def processConditions(myLocation,APIKEY,localityInfo):
   print("Requesting weatherData...")
   weatherData = weather.get(myLocation,APIKEY)
   print ("WeatherData:",weatherData)


   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
   })
```

MAIN.PY:

```
import brain

myLocation = "Chennai,IN"
APIKEY = "7a547a97f79ee73629edd98723ab36fe"

localityInfo = {
    "schools" : {
        "schoolZone" : True,
        "activeTime" : ["7:45","17:30"] # schools active from 7 AM till 5:30 PM
        },
    "hospitalsNearby" : False,
    "usualSpeedLimit" : 45 # in km/hr
}

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

WEATHER.PY:

```
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)
```

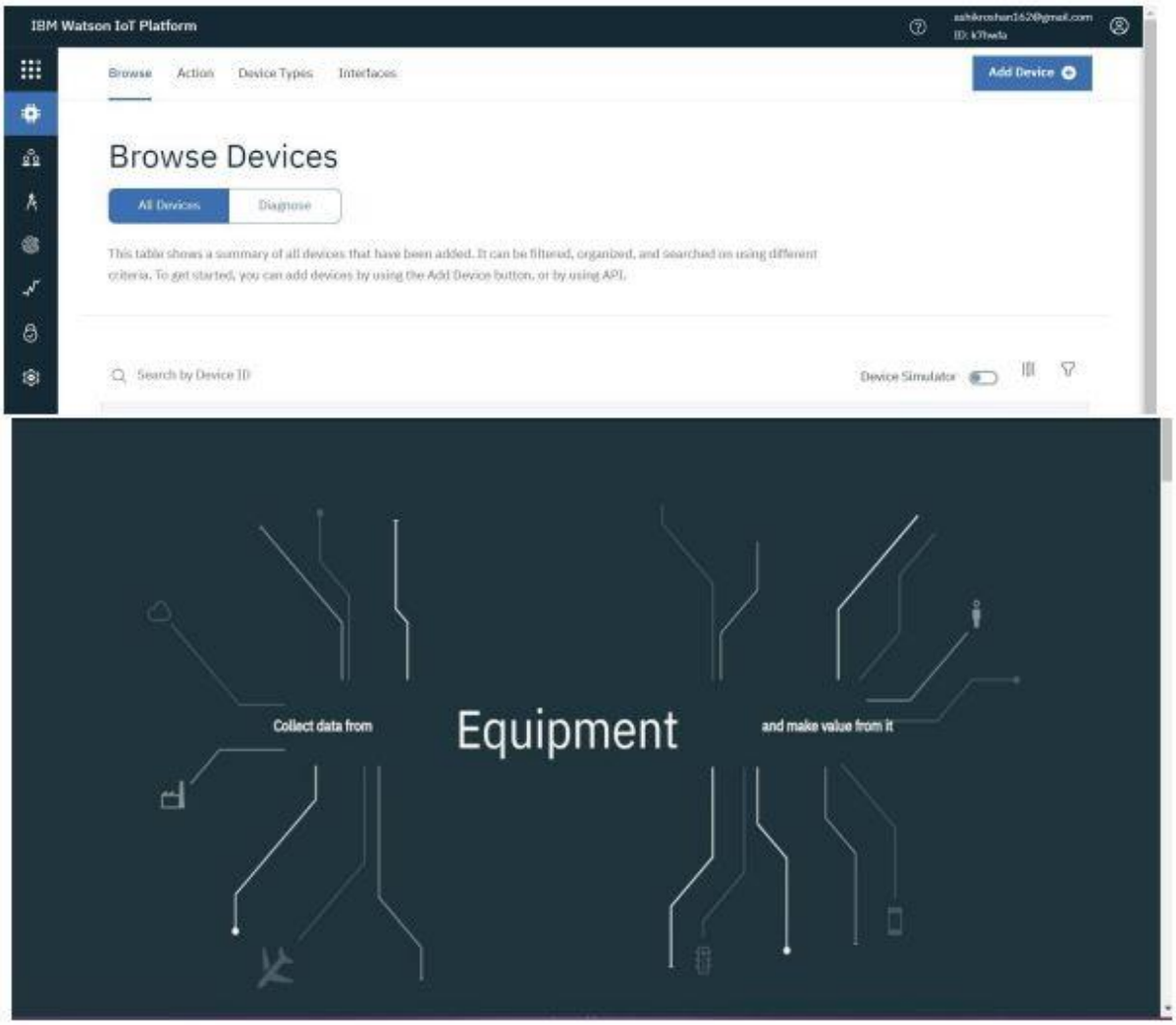
PUBLISH.PY:

```
import wiotp.sdk.device
import time
myConfig = {
    "identity" : {
        "orgId" : "k7hwfa",
        "typeId" : "NodeMcu",
        "deviceId" : "123456"
    },
    "auth" : {
        "token" : "12345678"
    }
}
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)
```

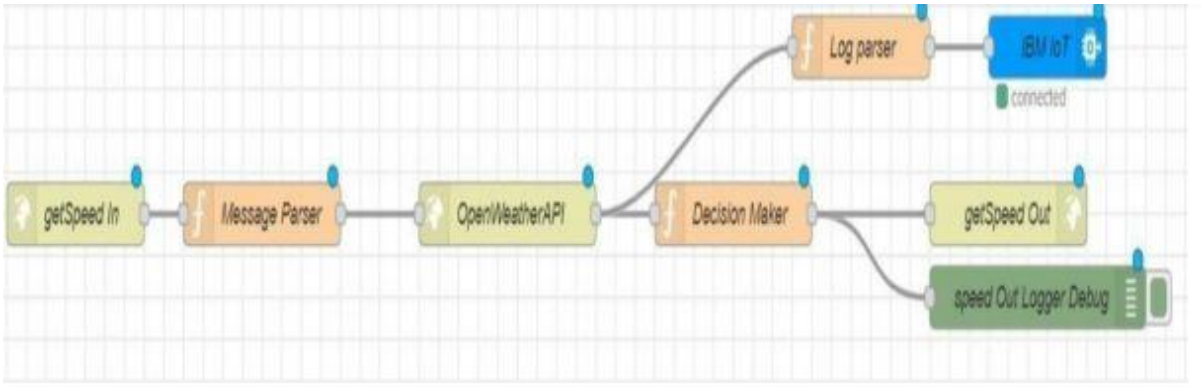
OUTPUT.PNG:

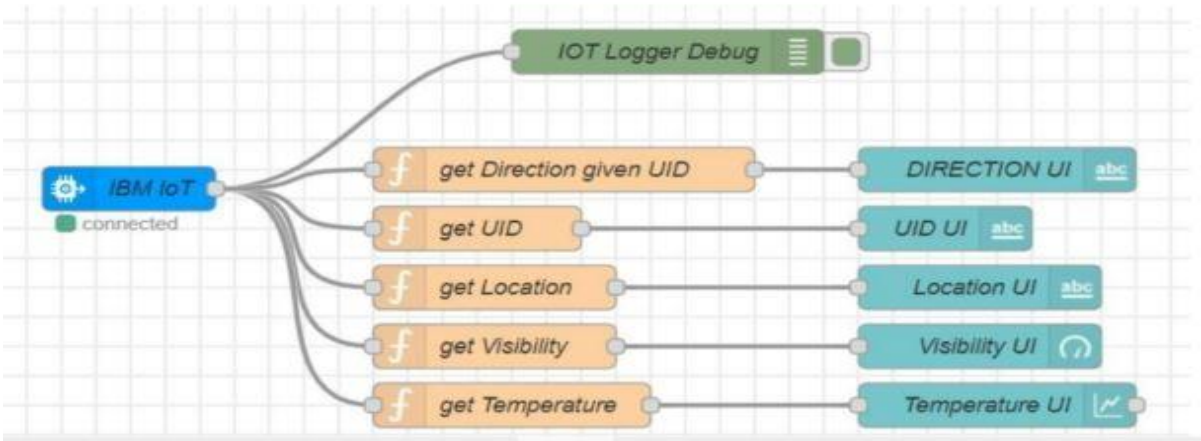
IBM WASTON IOT PLATFORM:

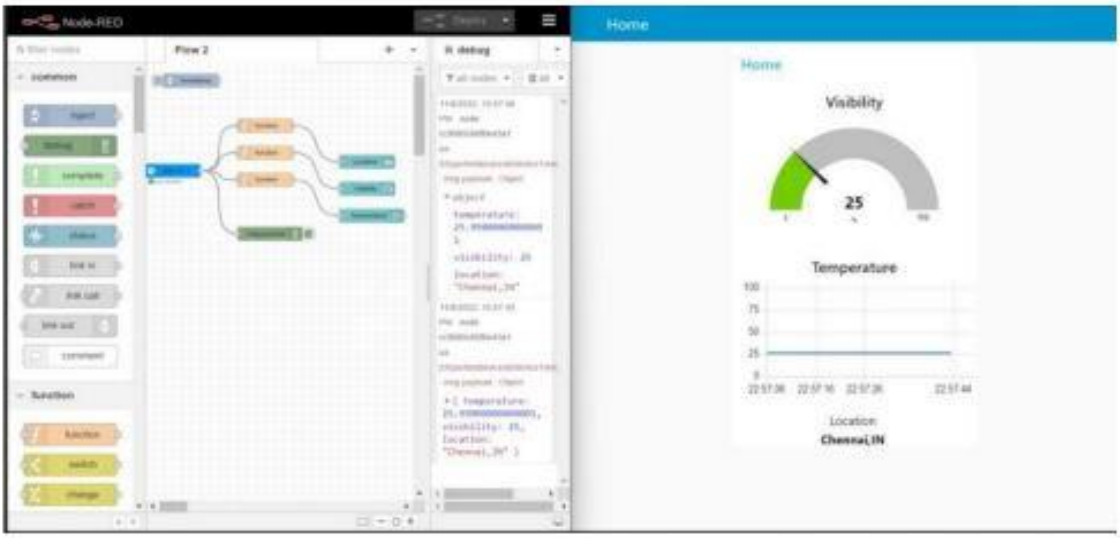


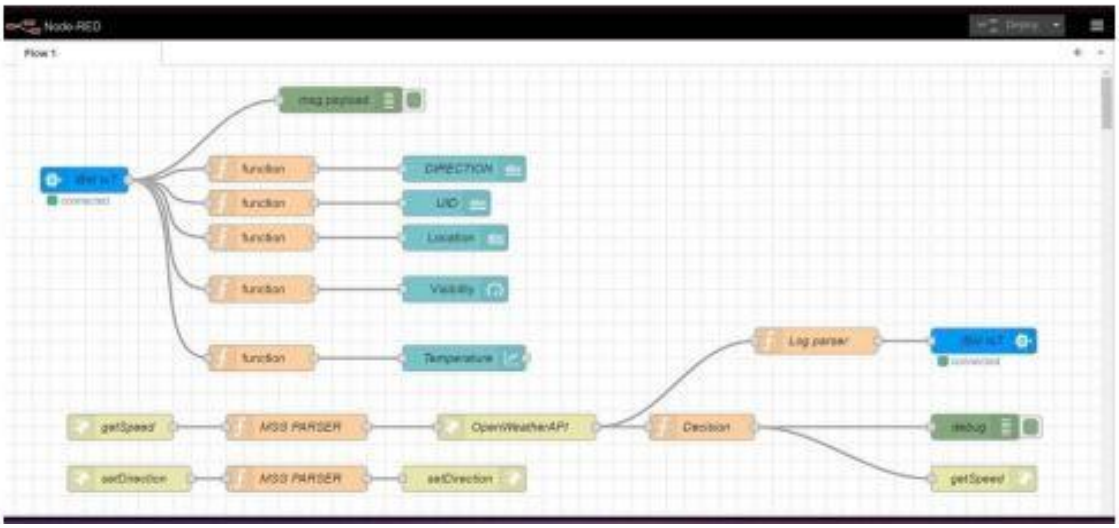
6.3. DELIVERY OF SPRINT – 3:

WORKFLOW FOR IOT SCENARIOs USING LOCAL NODE RED:



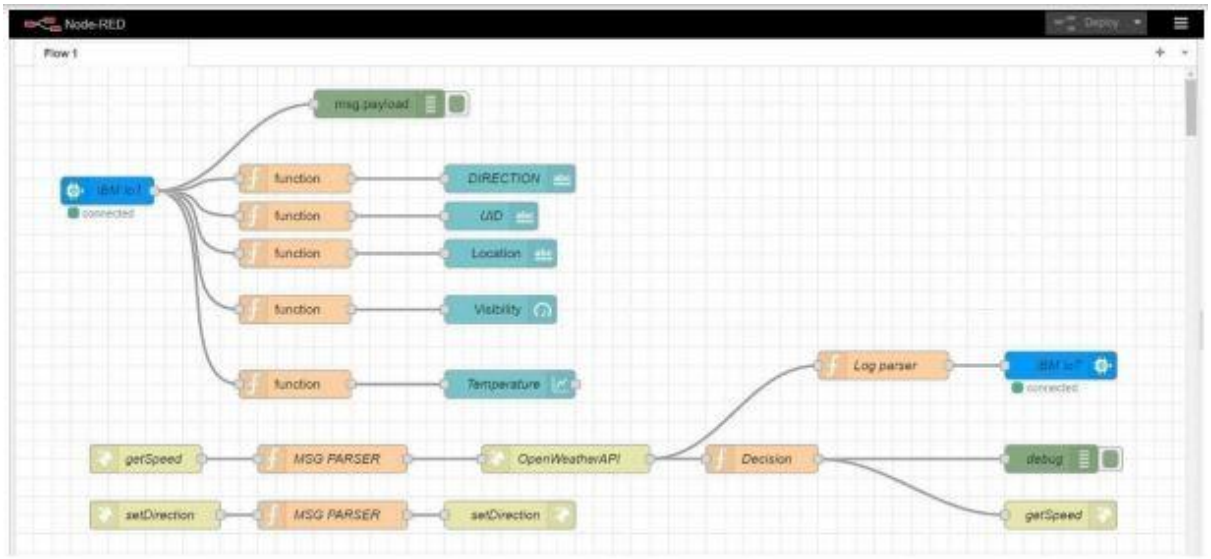


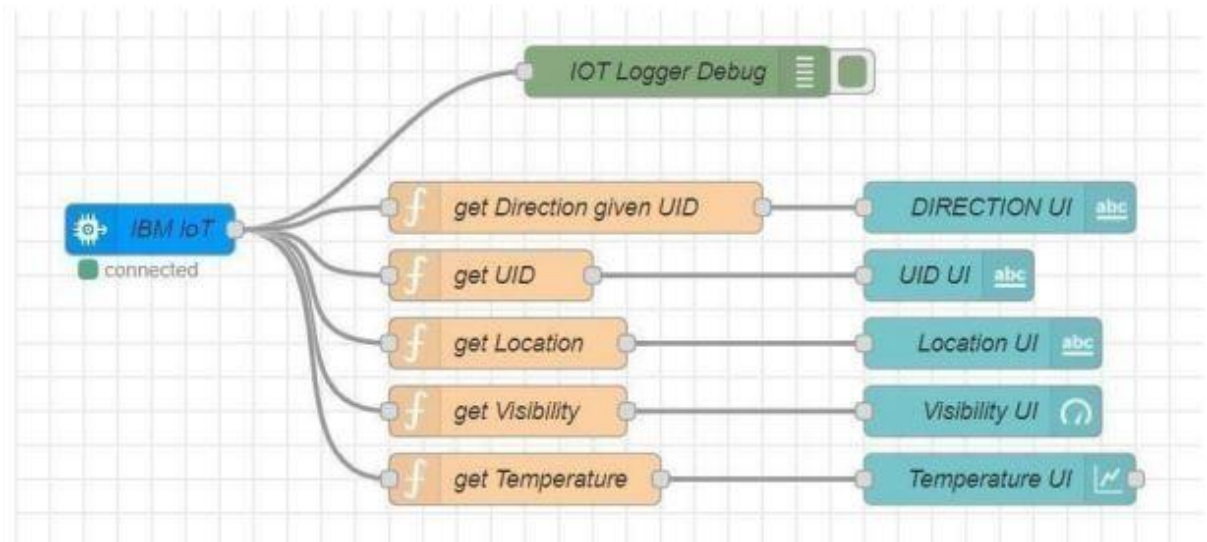

Node-RED
Flow 2
Home
Visibility
25
Temperature
Location:
Chennai,IN


Node-RED
Flow 1
msg.payload
function
DIRECTION
UID
Location
Visibility
Temperature
Log parser
getSpeed
MSG PARSER
OpenWeatherAPI
Decision
debug
setDirection
MSG PARSER
setDirection
getSpeed

6.4. DELIVERY OF SPRINT – 3:

NODE RED FLOW:






```
// get Direction given UID msg.payload = global.get(String(msg.payload.uid));
 return msg;
// get UID msg.payload = msg.payload.uid;
 return msg;
```

// get Location msg.payload = msg.payload.location;

return msg;

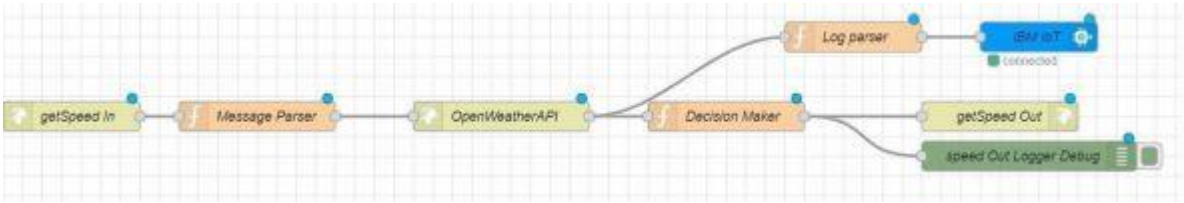
// get Visibility msg.payload = msg.payload.visibility;

return msg;

// get Temperature msg.payload = msg.payload.temperature;

return msg;

GETSPEED API FLOW :



weatherObj = JSON.parse(JSON.stringify(msg.payload));

localityObj=global.get("data");

var suggestedSpeedPercentage = 100

; var preciseObject = { temperature : weatherObj.main.temp - 273.15, location : localityObj.location, visibility : weatherObj.visibility/100, uid : localityObj.uid, direction : global.get("direction") }; msg.payload = preciseObject; return msg;

weatherObj=JSON.parse(JSON.stringify (msg.payload)); localityObj = global.get("data");

var suggestedSpeedPercentage = 100;

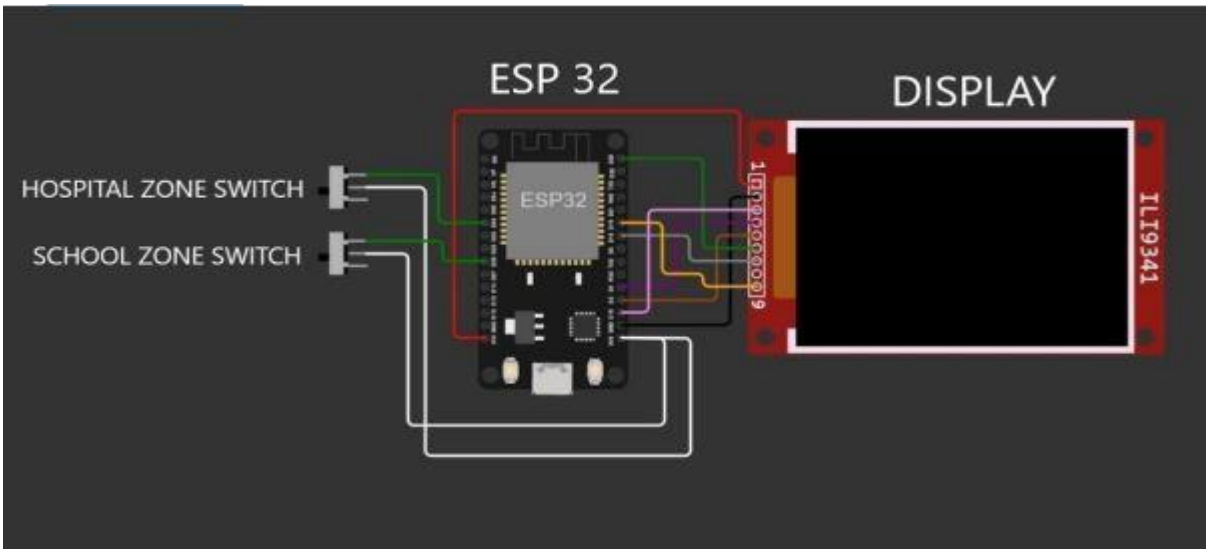
var preciseObject = { temperature : weatherObj.main.temp - 273.15, weather : weatherObj.weather.map(x=>x.id).filter(code => code<=40) suggestedSpeedPercentage -=30 switch(String(preciseObject.weather)[-1]) // https://openweathermap.org/weatherconditions refer weather codes meaning here { case "0" : suggestedSpeedPercentage -=10;break; case "1" : suggestedSpeedPercentage -=20;break; case "2" : suggestedSpeedPercentage -=30;break; } msg.payload = preciseObject; var doNotHonk = 0; if(localityObj.hospitalZone=="1"||localityObj.schoolZone=="1") doNotHonk = 1; var returnObject = { suggestedSpeed : localityObj.usualSpeedLimit*(suggestedSpeedPercentage/100), doNotHonk : doNotHonk } msg.payload = String(returnObject.suggestedSpeed) + " kmph \n\n" + (returnObject.doNotHonk==1?"Do Not Honk":"") + "$" + global.get(String(localityObj.uid));

return msg;

global.set(String(msg.payload.uid),msg.payload.dir);

return msg;

CIRCUIT DIAGRAM :



ESP 32 CODE :

#include const char* ssid = "Wokwi-GUEST";

const char* password = "";

#define TFT_DC 2;

#define TFT_CS 15 Adafruit_ILI9341 tft = Adafruit_ILI9341(TFT_CS, TFT_DC);

String myLocation = "Chennai,IN";

String usualSpeedLimit = "70";

// kmph int schoolZone = 32; int hospitalZone = 26; int uid = 2504;

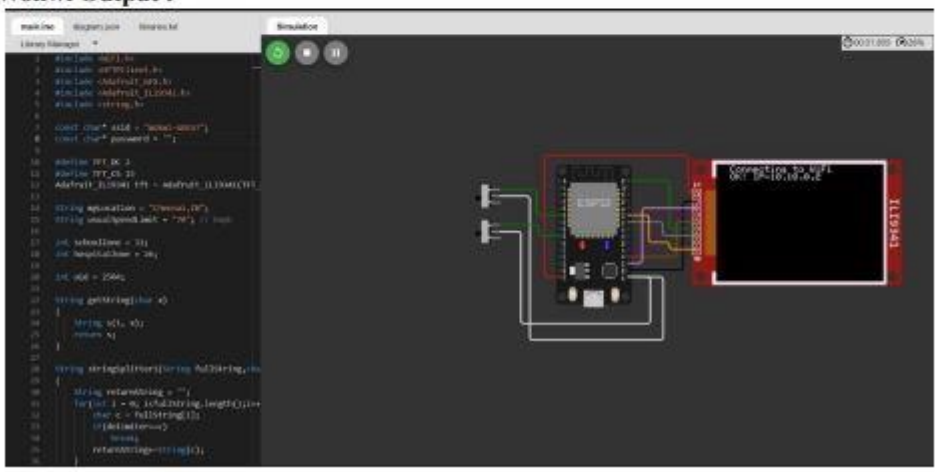
// ID Unique to this Micro Contoller String getString(char x) { String s(1, x); return s; } String stringSplitter1(String fullString,char delimiter='$') { String returnString = "";

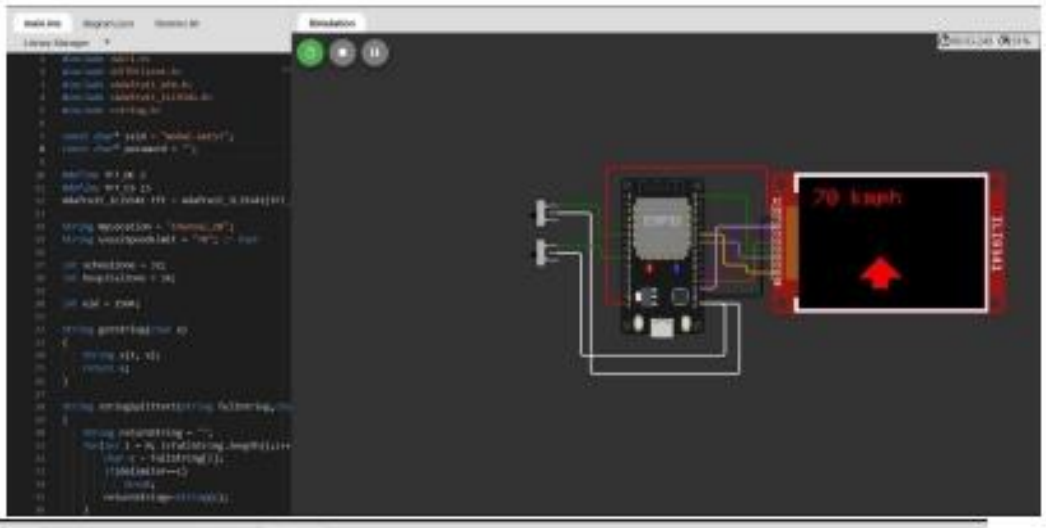
```
for(int i = 0; i0) { String payload = http.getString(); http.end(); return(payload); }

else { Serial.print("Error code: "); Serial.println(httpResponseCode); } http.end(); }

 void myPrint(String contents) { tft.fillScreen(ILI9341_BLACK); tft.setCursor(0, 20);

 tft.setTextSize(4); tft.setTextColor(ILI9341_RED); //tft.println(contents);
tft.println(stringSplitter1(contents));

 String c2 = stringSplitter2(contents); if(c2=="s") // represents Straight { upArrow(); } if(c2=="l") //
represents left { leftArrow(); } if(c2=="r") // represents right { rightArrow(); } } void setup() {
WiFi.begin(ssid, password, 6);

 tft.begin();

 tft.setRotation(1); tft.setTextColor(ILI9341_WHITE); tft.setTextSize(2); tft.print("Connecting to
WiFi"); while (WiFi.status() != WL_CONNECTED) { delay(100); tft.print("."); }

tft.print("\nOK! IP="); tft.println(WiFi.localIP()); } void loop() { myPrint(APICall()); delay(100);

}
```

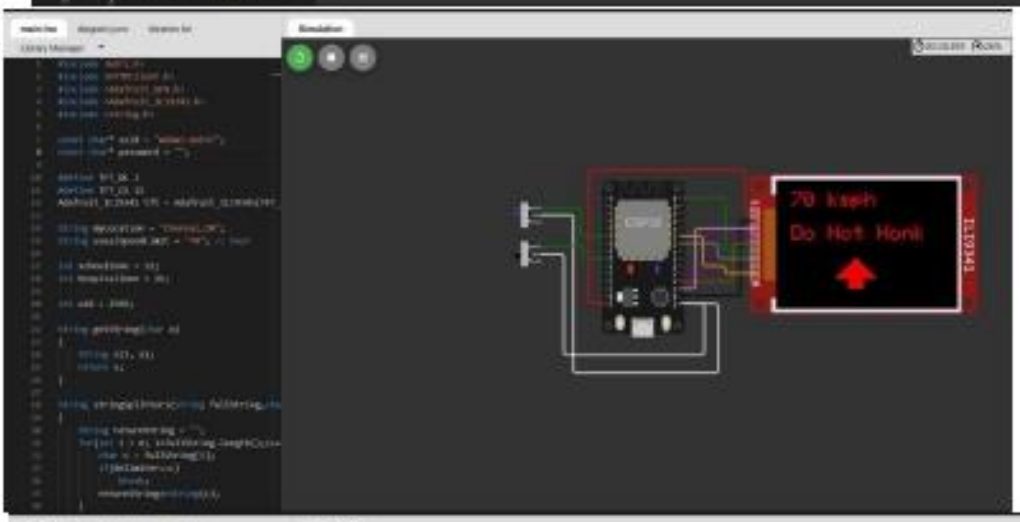
OUTPUT:

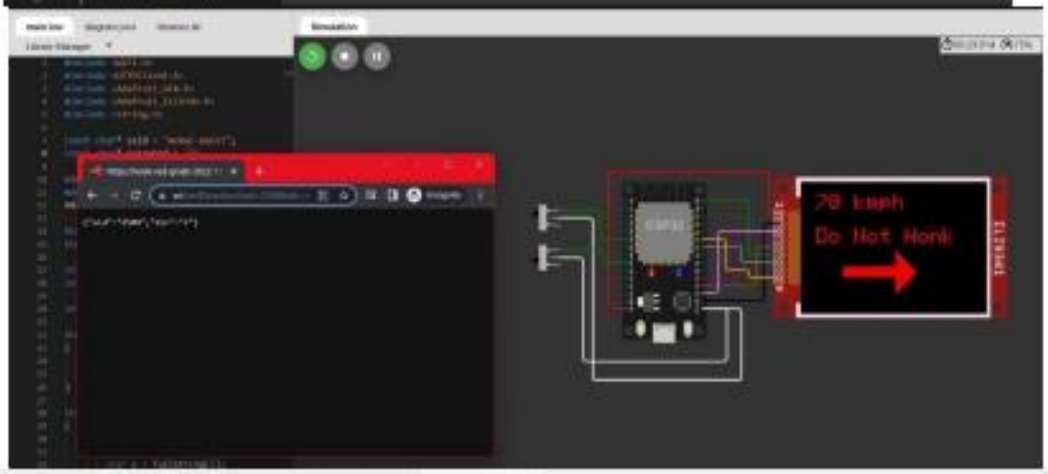
Node RED Dashboard :

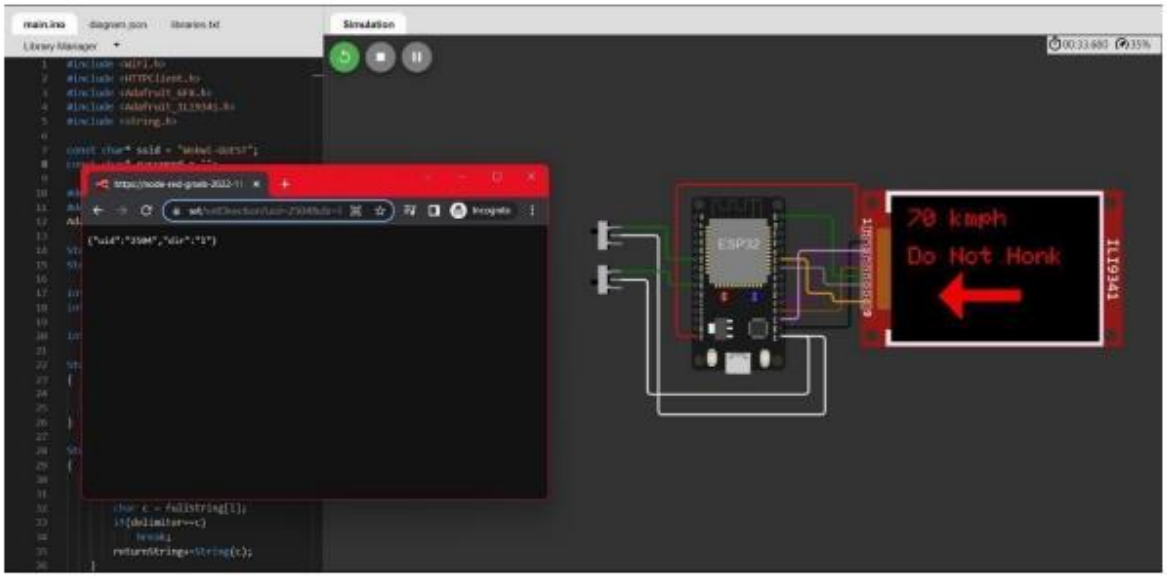
Wokwi Output :




70 kmph


70 kmph
Do Not Honk


70 kmph
Do Not Honk


main.ino
diagram.json
libraries.txt
Library Manager
Simulation
{"uid":"3504","dir":"1"}
ESP32
70 kmph
Do Not Honk

CHAPTER – 7

CONCULSION

The world won't change by itself, but we can make it safer, better and less harmful. The roads aren't safe, so we have technology to make them safer. will do. One of the technologies that will put us on a safer road is the Internet of Things. Let's work together to create a better accident-free world and smart roads for future generations.