



Om Sakthi

ADHIPARASAKTHI COLLEGE OF ENGINEERING

G.B.NAGAR, KALAVAI - 632 506. RANIPET DT.,



SIGNS WITH SMART CONNECTIVITY FOR BETTER ROAD SAFETY

IBM PROJECT

In partial fulfillment for the award of the degree
Of
BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE AND ENGINEERING

ACADEMIC YEAR: 2021-2022 (EVEN SEM)

ANNA UNIVERSITY, CHENNAI - 600 025.

PROJECT REPORT

SIGNS WITH SMART CONNECTIVITY FOR BETTER ROAD SAFETY

Submitted by

Team Id : PNT2022TMID39403

Dinesh V -510119104005

Saran C -510119104019

Manoj J -510119104013

Gunasekaran T -510119104009

TABLE OF CONTENTS:

1 INTRODUCTION	1
1.1 PROJECT OVERVIEW	1
1.2 PURPOSE	1
2 LITERATURE SURVEY	2
2.1 EXISTING PROBLEM	2
2.2 REFERENCES	2
2.3 PROBLEM STATEMENT DEFINITION	4
3 IDEATION AND PROPOSED SOLUTION	6
3.1 EMPATHY MAP CANVAS	5
3.2 IDEATION & BRAINSTORMING	6
3.3 PROPOSED SOLUTION	7
3.4 PROBLEM SOLUTION FIT	8
4 REQUIREMENT ANALYSIS	9
4.1 FUNCTIONAL REQUIREMENTS	9
4.2 NON- FUNCTIONAL REQUIREMENTS	10
5 PROJECT DESIGN	11
5.1 DATA FLOW DIAGRAM	11
5.2 SOLUTION & TECHNICAL ARCHITECTURE	13
5.3 USER STORIES	15
6 PROJECT PLANNING AND SCHEDULING	16
6.1 SPRINT PLANNING AND ESTIMATION	16
6.2 SPRINT DELIVERY SCHEDULE	18
6.3 REPORT FROM JIRA	

7 CODING & SOLUTIONING	19
7.1 FEATURE 1	19
7.2 FEATURE 2	20
8 TESTING	21
8.1 TEST CASES	21
8.2 USER ACCEPTANCE TESTING	23
9 RESULTS	24
9.1 PERFORMANCE METRICS	24
10 ADVANTAGES & DISADVANTAGES	26
ADVANTAGES	26
DISADVANTAGES	26
11 CONCLUSION	27
12 FUTURESCOPE	28
APPENDIX	29
SOURCE CODE	29
GITHUB	40

CHAPTER1

INTRODUCTION

1.1 PROJECT OVERVIEW

- 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.

1.2 PURPOSE:

- In present Systems the road signs and the speed limits are Static. But the road signs can be changed in some cases. We can consider some cases when there are some road diversions due to heavy traffic or due to accidents then we can change the road signs accordingly if they are digitalized. This project proposes a system which has digital sign boards on which the signs can be changed dynamically. If there is rainfall then the roads will be slippery and the speed limit would be decreased. There is a web app through which you can enter the data of the road diversions, accident prone areas and the information sign boards can be entered through web app. This data is retrieved and displayed on the sign boards accordingly.

CHAPTER2

LITERATURE SURVEY

2.1 EXISTING PROBLEM:

- Conventional traffic system does not have proper monitoring system and often requires manual handling at traffic junction. This not only causes mental stress in passengers but also lot of fuel goes wasted due to delay at traffic junctionAccording to a study, one person dies every four minutes in road accidents in India, estimating the cost of such accidents at INR 3.8 lakh crores or 3% of the GDP. As a signatory to the Brasilia Declaration, India is committed to reducing the number of road accidents and fatalities by 50% by 2020.
- Analysis of crash data has suggested a link between roadside advertising signs and safety.Reasearches suggest that crash risk increases by 30% in presence of digital roadside advertising.In addition,drivers showed eye fixation and increased drifting between lanes on the road.

2.2 REFERENCES:

- <https://vriourope.com/en/smart-road-technology-digital-highways-of-the-future>
- <https://www.trafficinfratech.com/technology-for-road-safety>
- [https://www.researchgate.net/publication/221701886 Pedestrian Gap Acceptance for Mid-Block Street Crossin](https://www.researchgate.net/publication/221701886)

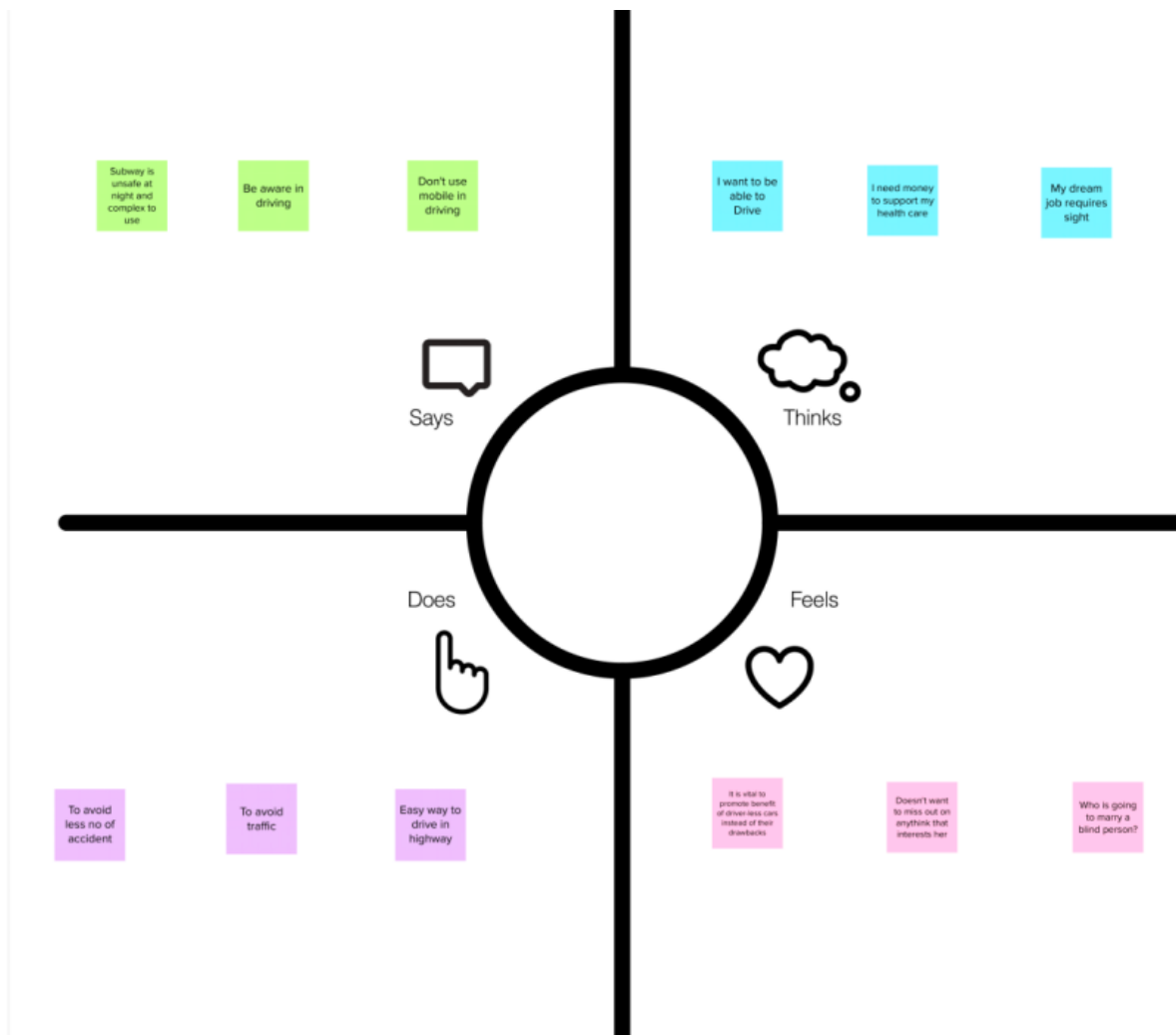
2.1 PROBLEM STATEMENT DEFINITION

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 message if there are accidents in the road and alert messages if there is hospital,school and any roadworks,also using the webapplication the user will get to know about road condition.

CHAPTER 3

IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS:



3.2 IDEATION & BRAINSTORMING

Dinesh

Identify the accident happened and immediately inform to ambulance services

Inform the control room if someone over rule the sign board instruction

Gunasekaran

If someone follows the wrong lane - inform the control room

To the left there is need of bounding long distance for the left turn due to the distribution location inform them the nearest U-turn

Improve control and safety can be achieved through IOT - smart cars in case of over speeding, the notification gets displayed on the car's windshield alerting the driver

Saran

Even if driver doesn't get right information on the road, the information can be provided through the mobile phone. The driver can be notified through the mobile phone about the accident location and the nearest hospital

It should be the company to provide vehicles to the most efficient routes, immediately after the accident made is blocked by an accident

With 5G, vehicles will be able to communicate between them in real time. The generation delay will be less than 1ms, which means that the vehicles can communicate with each other in real time. This will help in avoiding accidents and improving traffic flow

Drivers can receive updates on the road conditions and traffic in real time. This will help them to avoid accidents and improve their driving experience. The system can also be used to provide information about the nearest hospital and the location of the accident

Manoj

Along with the video of the traffic, self drivers can receive updated information on the state of the roads, i.e. potholes, the grade changes, slow spots, etc.

Mobile performance degradation monitoring can be achieved by using the mobile phone's performance monitoring tools. This will help in identifying the causes of the performance degradation and taking the necessary steps to improve it

First responders can monitor from traffic flow velocity in real time, which helps them to save drivers about speeding and other incorrect behaviors

Mentor - Senthilraja

Detect the sign board through the vehicle cam and intimate them through the display or speaker

Intimate the animal interference on the way to the vehicle driver

Ensure safe driving experience with real time accident navigation, and even monitoring during emergency situation

3.3 PROPOSED SOLUTION:

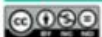
S.No.	Parameter	Description
•	Problem Statement (Problem to be solved)	The major problem statement of our project is animal interference on the way of user ride. Ten in thousands may escape from this incidents and in remaining cases either there will be injury or accident for the user or the animal. User doesn't know about the interference of the animal, it will be just a second of time to notice it so he/she will lose the control of the vehicle.
•	Idea / Solution description	Automobile need to be modified in such a way that each vehicle has a inbuilt wide angle camera of 180 degree that does the image processing through the snapshots of the video capturing. If any unknown object comes in front of the vehicle and then intimate to the user through the alert sounds installed in the vehicle.
•	Novelty / Uniqueness	There is no existing project based on our idea. Our project is a fresh idea that remain unique.
•	Social Impact / Customer Satisfaction	With the successful implementation of our idea users no need to panic because of the prior alert of the interference that saves their life and indirectly the animal life too.
•	Business Model (Revenue Model)	
•	Scalability of the Solution	

3.4 PROBLEM SOLUTIONFIT

Problem-Solution fit canvas 2.0

Purpose / Vision

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> <div>People using roads in cities.</div>	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> <div>Need to follow rules, waiting in traffic for long time increase fuel consumption, Reducing road accidents causes less vehicular damages.</div>	5. AVAILABLE SOLUTIONS <small>Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking</small> <div>Over speed vehicles detection Video based traffic monitoring system to detect accidents and defend the people using pedestrian cross in traffics.</div>	Explore AS, differentiate
	2. JOBS-TO-BE-DONE / PROBLEMS <small>Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides.</small> <div>Traffic congestions improper traffic signs more accident due to overspeeding Afraid to cross the road due to rash driving</div>	9. PROBLEM ROOT CAUSE <small>What is the real reason that this problem exists? What is the back story behind the need to do this job? i.e. customers have to do it because of the change in regulations.</small> <div>Not following the traffic rules properly. Due to usage of more vehicles. Non adherence to lane driving and overtaking in a wrong manner. Lack of conscious on the roads.</div>	7. BEHAVIOUR <small>What does your customer do to address the problem and get the job done? i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)</small> <div>Leaving way to the emergency vehicles. Stop the vehicle to help elderly people to cross the lane.</div>	
	3. TRIGGERS <small>To reach on time (eg: Meetings, exams, etc..)</small> <div></div>	10. YOUR SOLUTION <small>If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality. If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.</small> <div>Our proposed solutions are vehicles density-based traffic controls system with a smart pedestrian crossing connectivity to the smart traffic signs.</div>	8. CHANNELS of BEHAVIOUR <small>8.1 ONLINE</small> <div>Taking videos and photography of a person who got accidents without</div> <small>8.2 OFFLINE</small> <div>Falling to obey the stop signals and signs, Drank driving</div>	
4. EMOTIONS: BEFORE / AFTER <small>Angry, Annoyed, Combative, Exhausted</small> <div></div>				



Problem-Solution fit canvas is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 license
Created by Dana Naprahona / Amaltama.com

AMALTAMA

CHAPTER4

REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Pedestrian Aspects	The field element shall collect pedestrian sensor data and respond to pedestrian crossing via display (pedestrian sign indication) and control traffic signs accordingly.
FR-2	Density based vehicle aspects - Transit Signal priority	<p>The field element collect information about emergency vehicle in which green time is extended or red time is truncated in traffic signs and expedite movement of authorized emergency vehicles.</p> <p>The transit signal priority shall include rules to negotiate competing calls for priority.</p>
FR-3	Monitoring Aspects	The field element shall monitor operation of the traffic signal controllers and report to the centre any instance in which the indicator response does not match that expected from the detectors and sensors.
FR-4	Interface aspects	The field element shall include traffic sensor that receive control information to other field element devices (such as traffic controllers at adjacent intersections and dynamic message signs).
FR-5	Simultaneous working aspects:	<p>Our project focus on both the pedestrian and vehicular users to reach their place safely and on time respectively.</p> <p>This parallel mechanism helps to save time and processing of data</p>

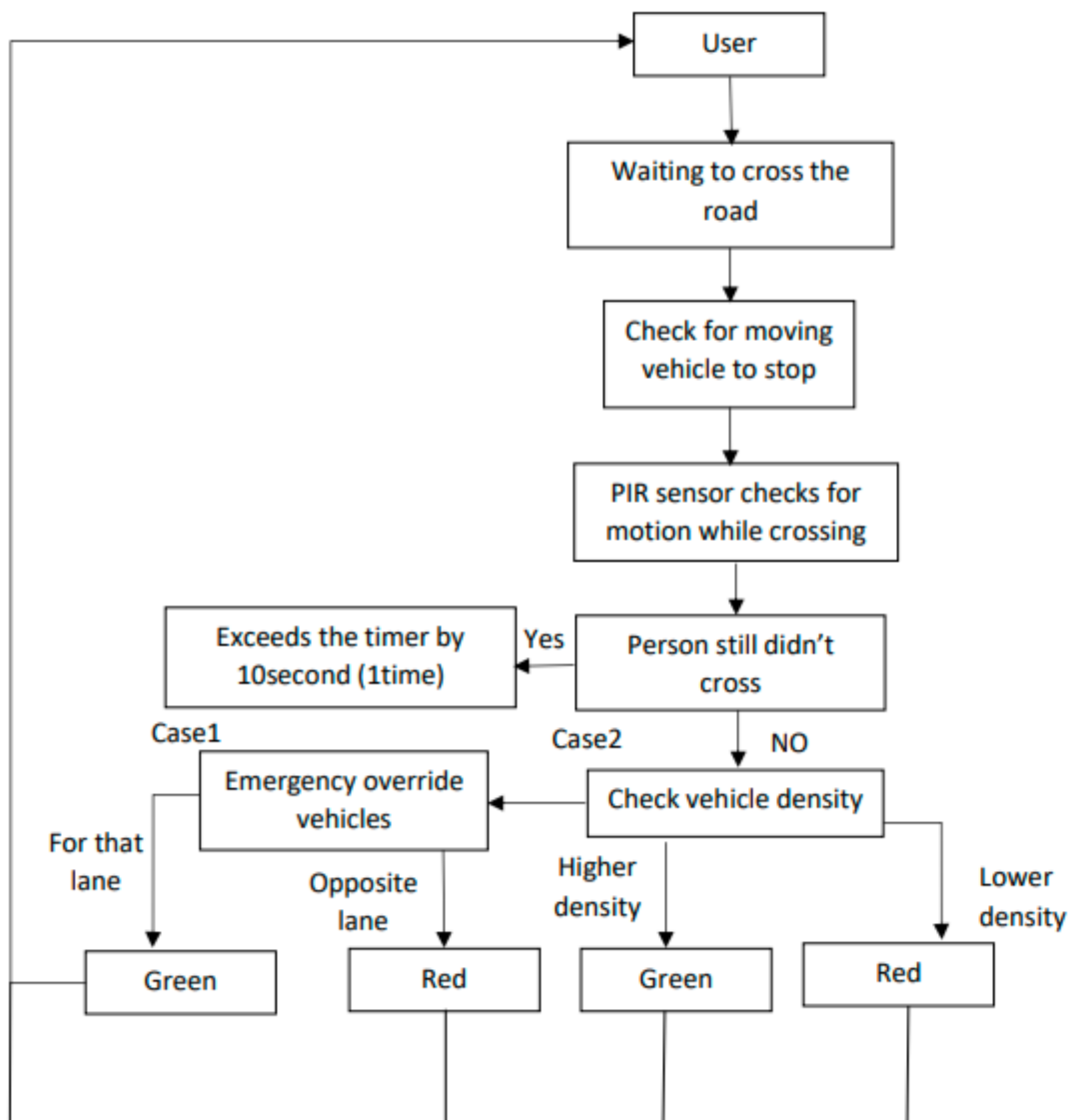
4.2 NON-FUNCTIONAL REQUIREMENTS

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	<p>It's used for the people who cross the road slowly, and for density change its useful for who rushing to the exams or meeting's.</p> <p>It is used in order to increase the safety of pedestrians that can find themselves at the same</p>
		time with vehicles on road, the automobiles and roads are fitted which controls traffic congestion
NFR-2	Security	<p>The storing data are not important for any kind of illegal activity.</p> <p>In this project, security aspects are not much issues. Only the data of vehicles counts and people count detected thus the security won't be an issue. The security on their database includes firewalls to prevent unauthorized access.</p>
NFR-3	Reliability	<p>It significantly improves the efficiency and safety of pedestrian traffic from making the right decision in multimode transport. And it also reduces the traffic congestions and accidents.</p>
NFR-4	Performance	<p>It provides Quality of service by reducing the latency in the traffic sign boards. Thus, the processors are faster</p>
NFR-5	Availability	<p>It's available for 24x7 hours and in any climate conditions.</p>
NFR-6	Scalability	<p>This data are not stored for long time so the disk space can be free it helps to handle large data with a fast execution.</p>

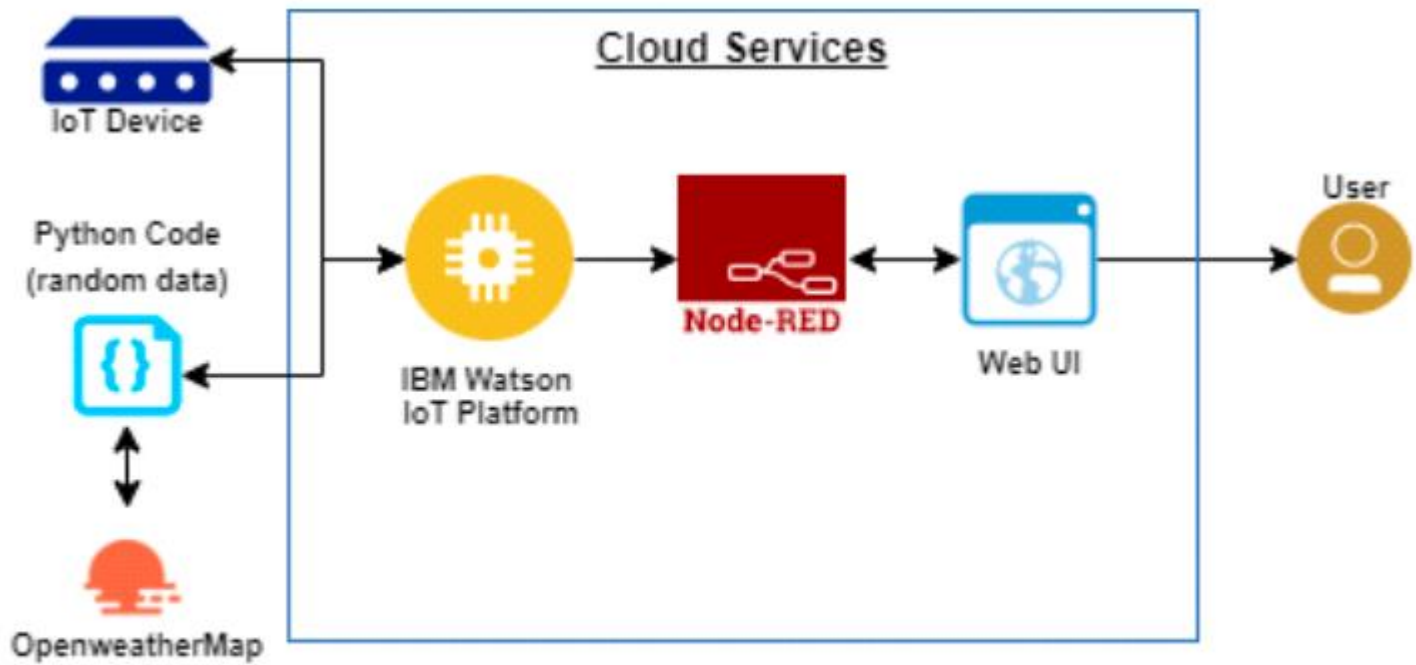
CHAPTER5

PROJECT DESIGN

5.1 DATA FLOW DIAGRAM



5.2 SOLUTION & TECHNICAL ARCHITECTURE



5.3 COMPONENTS & TECHNOLOGIES:

Table-1: Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Mobile App	HTML, CSS, JavaScript
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 StorageService 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 / CloudLocal Server Configuration: Cloud Server Configuration :	Local, Cloud Foundry, Kubernetes, etc.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Technology of Opensource framework
2.	Security Implementations	List all the security / access controls implemented,use of firewalls etc.	e.g. SHA-256, Encryptions, IAM Controls, OWASP etc.
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	Technology used
4.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	Technology used

5.4 USERSTORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Pedestrian)	Pedestrian aspects	USN-1	The user trying to cross the road, and once he stands in pedestrian lane, it detects with pir sensor and if the density of vehicles is low the pedestrian signs go steady green and vehicles to be stopped for sometimes and the pedestrian sign starts flashing green for 10 seconds and by that time all have to cross and vehicles can move forward, by this monitoring and sign connectivity, we can avoid non adherence of traffic rules and no congestion in cities.	The board can receive the detection of people presence in the lane and sent it to the sign board	High	Sprint-1
Emergency vehicles driver	Emergency rider aspects	USN-2	As a emergency responder, i need to know the accurate state of road, routes and schedule and to know the best possible or prioritized route to the incident and to hospitals.	I can able to know the best route to the hospital to fulfill my duty.	High	Sprint-1
Customer	Reporting and safety information aspects	USN-3	Who is operating the vehicle and that i can trust them, and if it is run by a human or by AI	I can sense the level of higher safety	Low	Sprint-2
Driver	Distractions and frustrations	USN-4	Drivers get distracted due to the long wait period in the red signal but the opposite lane is empty. This makes the person to feel frustrated. So, in our project we look after the vehicular density on each lane and based on that data we going to control the timings of the signal on each lane hence traffic control can be monitored efficiently.	I can travel in peaceful manner and reach the destination on time.	Medium	Sprint-1
Student and working persons	Reach on time	USN-5	Due to heavy congestion they cannot reach on time. In our project we maintained efficient traffic signaling system which reduces the heavy traffic.	I can reach the destination earlier	High	Sprint-1

CHAPTER 6

PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1		US-1	Create IBM cloud services.	6	High	Dinesh V Gunasekaran T Saran C Manoj J
Sprint-1		US-3	IBM Watson IoT platform acts as the mediator to connect the web application to IoT devices.	5	Medium	Dinesh V Gunasekaran T
Sprint-1		US-4	In order to connect the IoT device to the IBM cloud, create a device in the IBM Watson IoT platform and get the device credentials.	5		Saran C Manoj J
Sprint-2		US-1	Configure open weather map services.	7	High	Saran C Manoj J
Sprint-2		US-2	Configure the connection security and create API keys that are used in the Node-RED service for accessing the IBM IoT Platform.	7	Medium	Gunasekaran T Saran C
Sprint-2		US-3	Create Node-red services.	5	High	Dinesh V Manoj J

Sprint-3		US-1	Develop a python script to publish random sensor data such as temperature, humidity, rain to the IBM IoT platform	7	Medium	Dinesh V Gunasekaran T
Sprint-3		US-2	After developing python code, commands are received just print the statements which represent the control of the devices.	5	High	Saran C Manoj J
Sprint 3		US-3	Publish Data to The IBM Cloud	8	High	Dinesh V Saran C
Sprint-4		US-1	Create Web UI in Node- Red	10	High	Dinesh V Gunasekaran T Saran C Manoj J
Sprint-4		US-2	Configure the Node-RED flow to receive data from the IBM IoT platform and also use Cloudant DB nodes to store the received sensor data in the cloudant DB	10	High	Dinesh V Gunasekaran T Saran C Manoj J

6.2 SPRINT DELIVERY SCHEDULE

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	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

6.3 REPORT FROM JIRA

ISSUES CREATED FOR SPRINT :

The screenshot displays the Jira Software interface for a project named 'Signs with Smart Connectivity for Better Road Safety'. The 'Backlog' view is active, showing two sprints:

- SWSCFBRS Sprint 2** (31 Oct – 5 Nov, 3 issues):
 - SWSCFBRS-34: Configure open weather map services. (7 points, DONE)
 - SWSCFBRS-36: Create Node-red services. (5 points, DONE)
 - SWSCFBRS-35: Configure the connection security and create API keys. (7 points, DONE)
- SWSCFBRS Sprint 3** (7 Nov – 12 Nov, 3 issues):
 - SWSCFBRS-37: Develop a python script to publish random sensor data such as temperature, humidity, rain to the IBM IoT platform. (7 points, DONE)
 - SWSCFBRS-38: After developing python code, commands are received just print the statements which represent the control of the devices. (5 points, DONE)
 - SWSCFBRS-39: Publish Data to The IBM Cloud. (8 points, DONE)

The interface includes a sidebar with navigation options like Roadmap, Backlog, Board, Reports, Code, Project pages, Add shortcut, and Project settings. A 'Quickstart' button is visible at the bottom right.

MIT App Inventor x MIT App Inventor x [JIRA] (SWSCFBRS-36) Create No... x Signs with Smart Connectivity for... x Signs with Smart Connectivity for... x

pn12022tmk39434.atlassian.net/jira/software/projects/SWSCFBRS/boards/2/backlog

Jira Software Your work Projects Filters Dashboards People Apps Create Search

Signs with Smart Conn... Software project

PLANNING Roadmap Backlog Board Reports

DEVELOPMENT Code Project pages Add shortcut Project settings

You're in a team-managed project Learn more

Projects / Signs with Smart Connectivity for Better Road Safety

Backlog

Search [] [HU] [KK] [RB] [S] Epic

SWSCFBRS Sprint 3 7 Nov - 12 Nov (3 issues) 0 0 (20) Complete sprint

- SWSCFBRS-37 Develop a python script to publish random sensor data such as temperature, humidity,rain to the IBM IoT platform 7 DONE [KK]
- SWSCFBRS-38 After developing python code, commands are received just print the statements which represent the control of the devices 5 DONE [RB]
- SWSCFBRS-39 Publish Data to The IBM Cloud 8 DONE [S]

+ Create issue

SWSCFBRS Sprint 4 12 Nov - 19 Nov (2 issues) 0 0 (20) Complete sprint

- SWSCFBRS-41 Use Cloudant DB nodes to store the received sensor data 10 DONE [KK]
- SWSCFBRS-40 Create Web UI in Node- Red 10 DONE [HU]

+ Create issue

Quickstart

MIT App Inventor x MIT App Inventor x [JIRA] (SWSCFBRS-36) Create No... x SWSCFBRS board - Agile board - x Signs with Smart Connectivity for... x

pn12022tmk39434.atlassian.net/jira/software/projects/SWSCFBRS/boards/2

Jira Software Your work Projects Filters Dashboards People Apps Create Search

Signs with Smart Conn... Software project

PLANNING Roadmap Backlog Board Reports

DEVELOPMENT Code Project pages Add shortcut Project settings

You're in a team-managed project Learn more

Projects / Signs with Smart Connectivity for Better Road Safety

All sprints

Search [] [HU] [KK] [RB] [S] Epic Sprint

GROUP BY None Insights

TO DO

IN PROGRESS

DONE 11 ISSUES

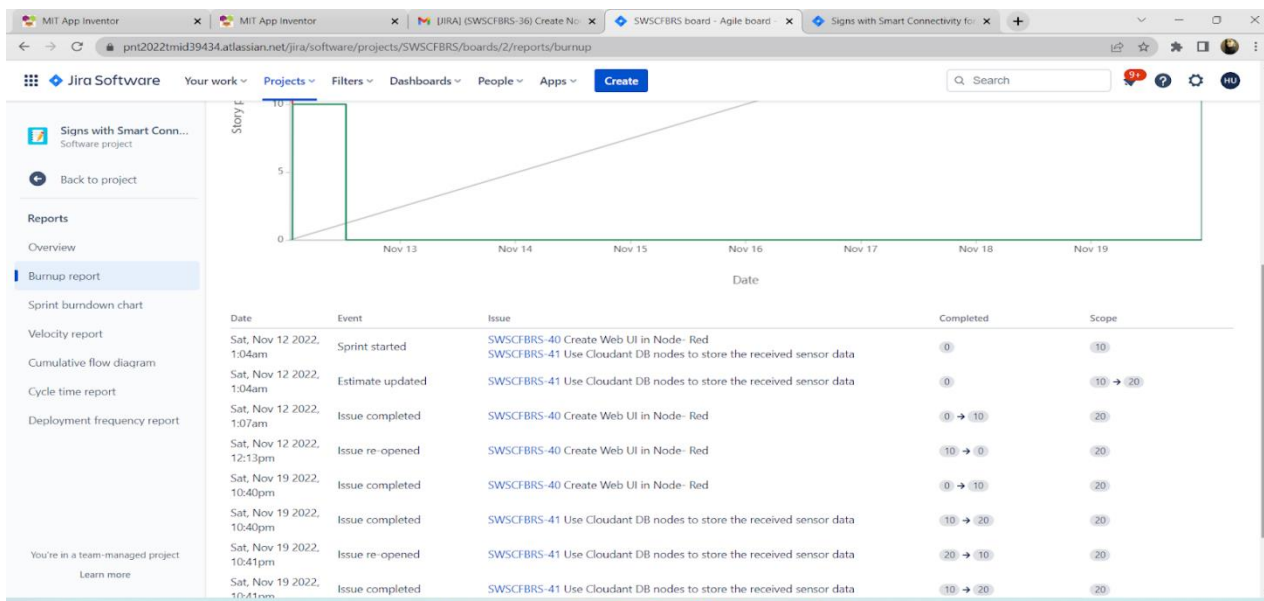
- Create IBM cloud services SWSCFBRS-31 6 [KK]
- Create IBM Watson IoT platform SWSCFBRS-32 5 [HU]
- Create the device in IBM IoT platform SWSCFBRS-33 5 [HU]
- Configure open weather map services. SWSCFBRS-34 7 [S]
- Develop a python script to publish random sensor data such as temperature,

Quickstart

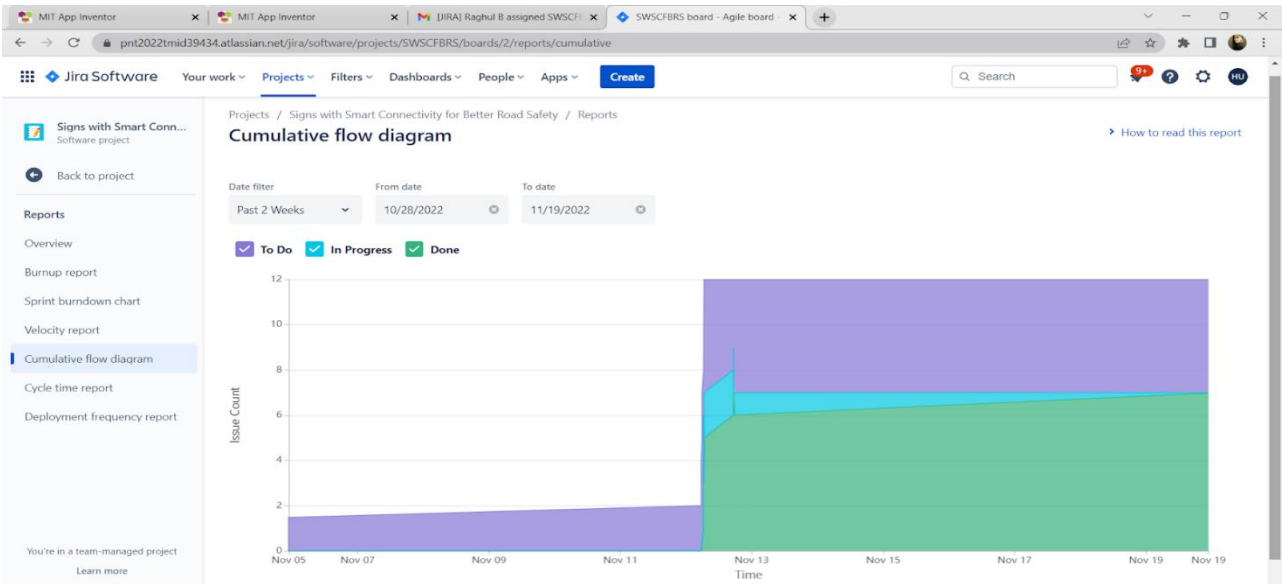
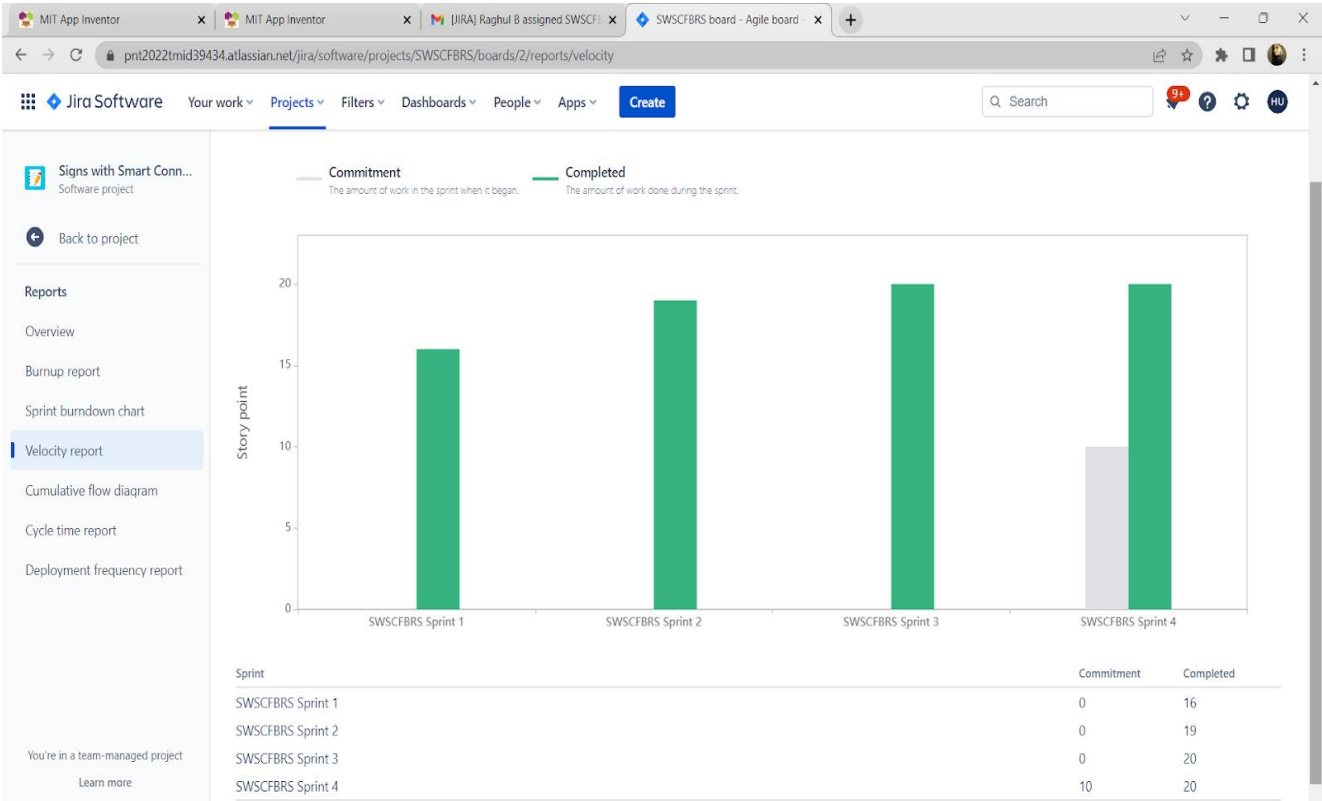
23°C Cloudy

ENG IN 22:58 19-11-2022

BURNDOWN CHART:



FINAL VELOCITY OF PROJECTS REPORT:



CHAPTER 7

CODING & SOLUTION

7.1 FEATURE 1

PYTHON CODE TO PUBLISH THE DATA TO THE IBM IOT DEVICE :

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random

#Provide your IBM Watson Device Credentials
organization = "8ahq3z"
deviceType = "SENSORS"
deviceId = "2211"
authMethod = "token"
authToken = "22112001"

# Initialize GPIO
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="lighton":
        print ("led is on")
    else :
        print ("led is off")
```



```
#print(cmd)

try:

    deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method":
authMethod, "auth-token": authToken}

    deviceCli = ibmiotf.device.Client(deviceOptions)
```

```
#.....
```

```
except Exception as e:

    print("Caught exception connecting device: %s" % str(e))

    sys.exit()
```

```
# Connect and send a datapoint "hello" with value "world" into the cloud as an event of type
"greeting" 10 times
```

```
deviceCli.connect()
```

```
while True:
```

```
    #Get Sensor Data from DHT11
```

Python 3.7.0 Shell

File Edit Shell Debug Options Window Help

Published Temperature = 70 C Humidity = 92 % to IBM Watson
Published Temperature = 68 C Humidity = 50 % to IBM Watson
Published Temperature = 47 C Humidity = 29 % to IBM Watson
Published Temperature = 72 C Humidity = 52 % to IBM Watson
Published Temperature = 50 C Humidity = 88 % to IBM Watson
Published Temperature = 63 C Humidity = 83 % to IBM Watson
Published Temperature = 55 C Humidity = 98 % to IBM Watson
Published Temperature = 74 C Humidity = 27 % to IBM Watson
Published Temperature = 16 C Humidity = 45 % to IBM Watson
Published Temperature = 48 C Humidity = 50 % to IBM Watson
Published Temperature = 25 C Humidity = 30 % to IBM Watson
Published Temperature = 0 C Humidity = 71 % to IBM Watson
Published Temperature = 64 C Humidity = 75 % to IBM Watson
Published Temperature = 88 C Humidity = 93 % to IBM Watson
Published Temperature = 47 C Humidity = 70 % to IBM Watson
Published Temperature = 31 C Humidity = 74 % to IBM Watson
Published Temperature = 32 C Humidity = 41 % to IBM Watson
Published Temperature = 7 C Humidity = 100 % to IBM Watson
Published Temperature = 54 C Humidity = 47 % to IBM Watson
Published Temperature = 12 C Humidity = 65 % to IBM Watson
Published Temperature = 70 C Humidity = 77 % to IBM Watson
Published Temperature = 31 C Humidity = 87 % to IBM Watson
Published Temperature = 3 C Humidity = 22 % to IBM Watson
Published Temperature = 20 C Humidity = 58 % to IBM Watson
Published Temperature = 37 C Humidity = 14 % to IBM Watson
Published Temperature = 6 C Humidity = 76 % to IBM Watson
Published Temperature = 9 C Humidity = 32 % to IBM Watson
Published Temperature = 18 C Humidity = 40 % to IBM Watson
Published Temperature = 35 C Humidity = 70 % to IBM Watson
Published Temperature = 21 C Humidity = 75 % to IBM Watson
Published Temperature = 76 C Humidity = 74 % to IBM Watson
Published Temperature = 55 C Humidity = 55 % to IBM Watson
Published Temperature = 4 C Humidity = 39 % to IBM Watson
Published Temperature = 69 C Humidity = 45 % to IBM Watson
Published Temperature = 74 C Humidity = 61 % to IBM Watson
Published Temperature = 82 C Humidity = 100 % to IBM Watson
Published Temperature = 19 C Humidity = 44 % to IBM Watson
Published Temperature = 11 C Humidity = 24 % to IBM Watson
Published Temperature = 52 C Humidity = 24 % to IBM Watson
Published Temperature = 59 C Humidity = 67 % to IBM Watson

Ln 10, Col 0

IBM Watson IoT Platform

Browse Action Device Types Interfaces Add Device

The recent events listed show the live stream of data that is co

Event	Value
IoT Sensor	["temp":89,"Humid":62]
IoT Sensor	["temp":52,"Humid":24]
IoT Sensor	["temp":11,"Humid":24]
IoT Sensor	["temp":49,"Humid":44]
IoT Sensor	["temp":82,"Humid":100]

```
#Get Sensor Data from DHT11

temp=random.randint(-30,100)
Humid=random.randint(10,90)

data = { 'temp': temp, 'Humid': Humid }
#print data
def myOnPublishCallback():
    print ("Temperature = %s C" % temp, "Humidity = %s %" % Humid, "to IBM Watson")

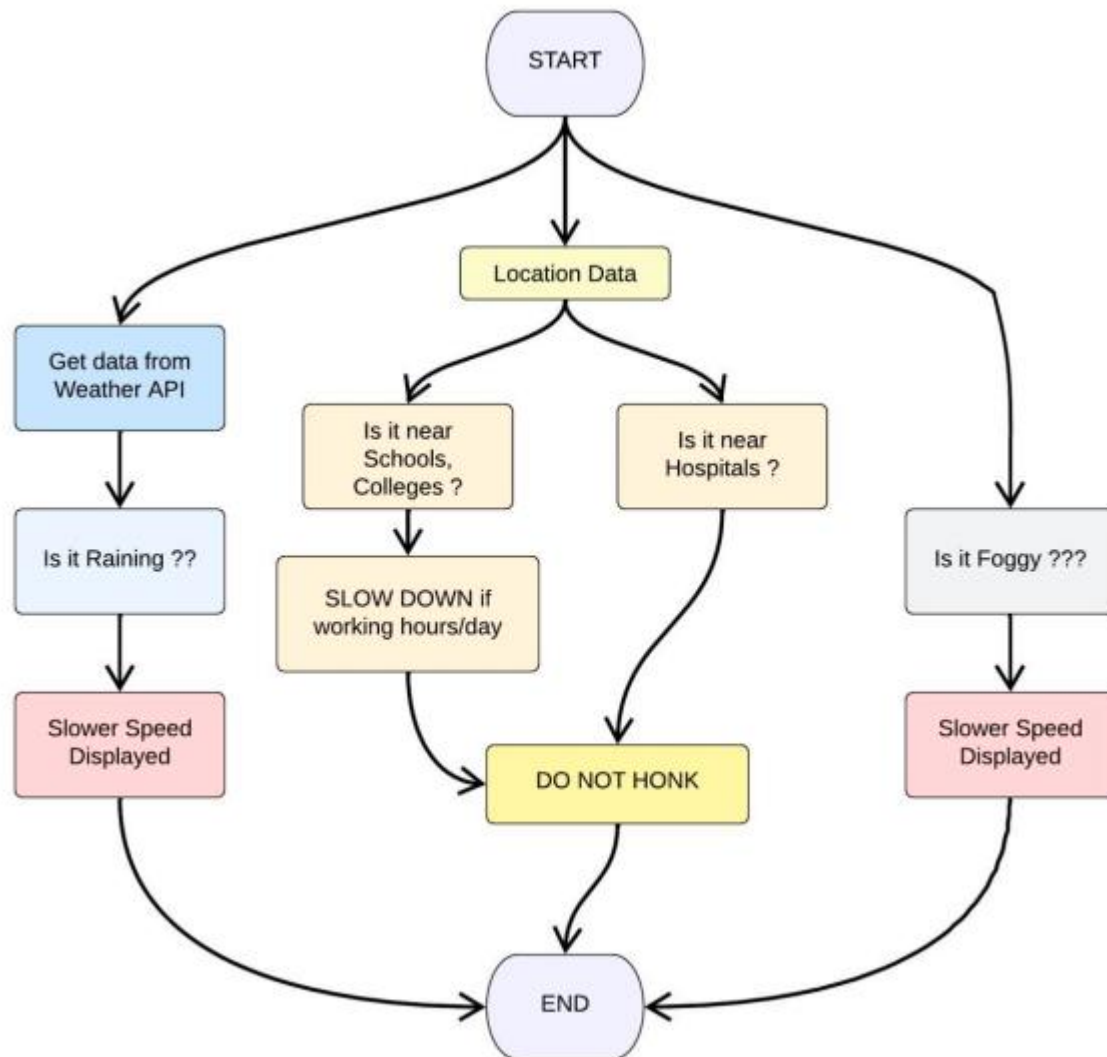
    success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,
on_publish=myOnPublishCallback)
    if not success:
        print("Not connected to IoT")
        time.sleep(10)

    deviceCli.commandCallback = myCommandCallback

# Disconnect the device and application from the cloud

deviceCli.disconnect()
```

CODE FLOW:



brain.py

```
# 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"]):
    # hzone of the hospital
        doNotHonk = True
    else:
        if(localityInfo["schools"]["schoolZone"]==False):
    # neither hospital zone nor school
        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
    })
```

brain.py - D:\suganya\S,RAHUL KUMAR\python\brain.py (3.11.0)

File Edit Format Run Options Window Help

```
#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"]):
        # hzone of the hospital
        doNotHonk = True
    else:
        if(localityInfo["schools"]["schoolZone"]==False):
            # neither hospital zone nor school
            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
    })
```

weather.py

```
import requests as reqs
def get(myLocation,APIKEY):
    apiURL =
    "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,
    }
    if("rain" in responseJSON):
        returnObject["rain"] = [responseJSON["rain"][key] for key in
responseJSON["rain"]] return(returnObject)
```

weather.py - D:/suganya/S.RAHUL KUMAR/python/weather.py (3.11.0)

File Edit Format Run Options Window Help

```
import requests as reqs

def get(myLocation,APIKEY):
    apiURL = "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,
    }
    if("rain" in responseJSON):
        returnObject["rain"] = [responseJSON["rain"][key] for key in responseJSON["rain"]]
    return(returnObject)
```

main.py

```
import brain
# IMPORT SECTION ENDS
# USER INPUT SECTION STARTS
myLocation = "Chennai,IN"
APIKEY =
"c7388b7d0d823ee0ee0be65c6fd40411"
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
while True :
    print(brain.processConditions(myLocation,APIKEY,localityInfo))
```


main.py - D:\suganya\S.RAHUL KUMAR\python\main.py (3.11.0)

File Edit Format Run Options Window Help

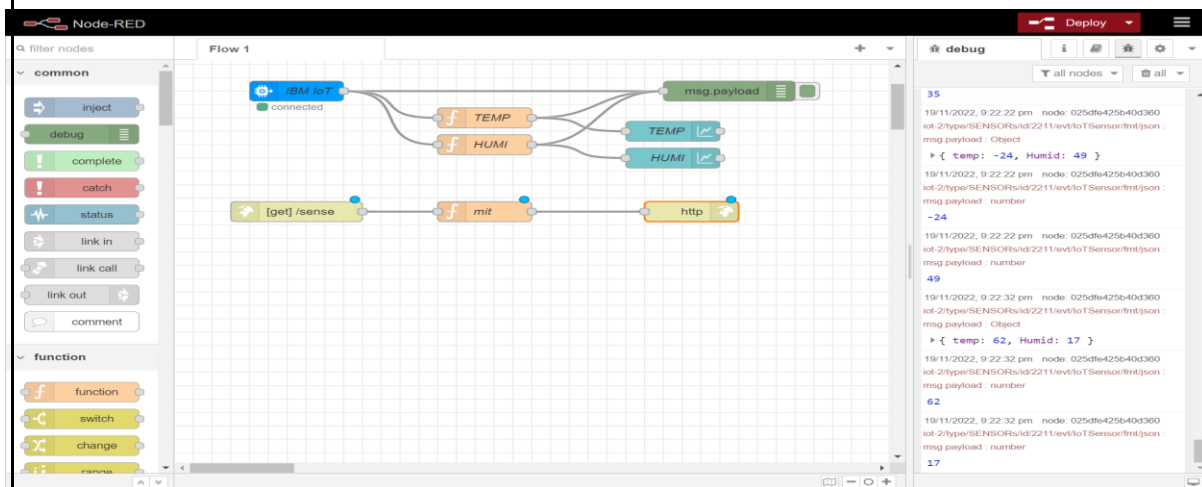
```
import brain
# IMPORT SECTION ENDS
# USER INPUT SECTION STARTS
myLocation = "Chennai,IN"
APIKEY = "c7388b7d0d823ee0ee0be65c6fd40411"
localityInfo = {
    "schools" : {
        "schoolZone" : True,
        "activeTime" : ["7:00","17:30"] # schools active from 7 AM till 5:30 PM
    },
    "hospitalsNearby" : False,
    "usualSpeedLimit" : 40 # in km/hr
}
# USER INPUT SECTION ENDS
# MICRO-CONTROLLER CODE STARTS
while True :
    print(brain.processConditions(myLocation,APIKEY,localityInfo))
```

Output :

Code Output

{'speed': 40, 'doNotHonk': False}

NODE RED WEB UI:



Browse

Action

Device Types

Interfaces

Add Device

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
IoTSensor	{"temp":-7,"Humid":51}	json	a few seconds ago
IoTSensor	{"temp":82,"Humid":42}	json	a minute ago
IoTSensor	{"temp":40,"Humid":47}	json	a minute ago
IoTSensor	{"temp":3,"Humid":47}	json	a minute ago
IoTSensor	{"temp":71,"Humid":43}	json	a minute ago

>

SENSORS_1

Disconnected

SENSORS

Items per page 50

1-4 of 4 items

0 Simulations running

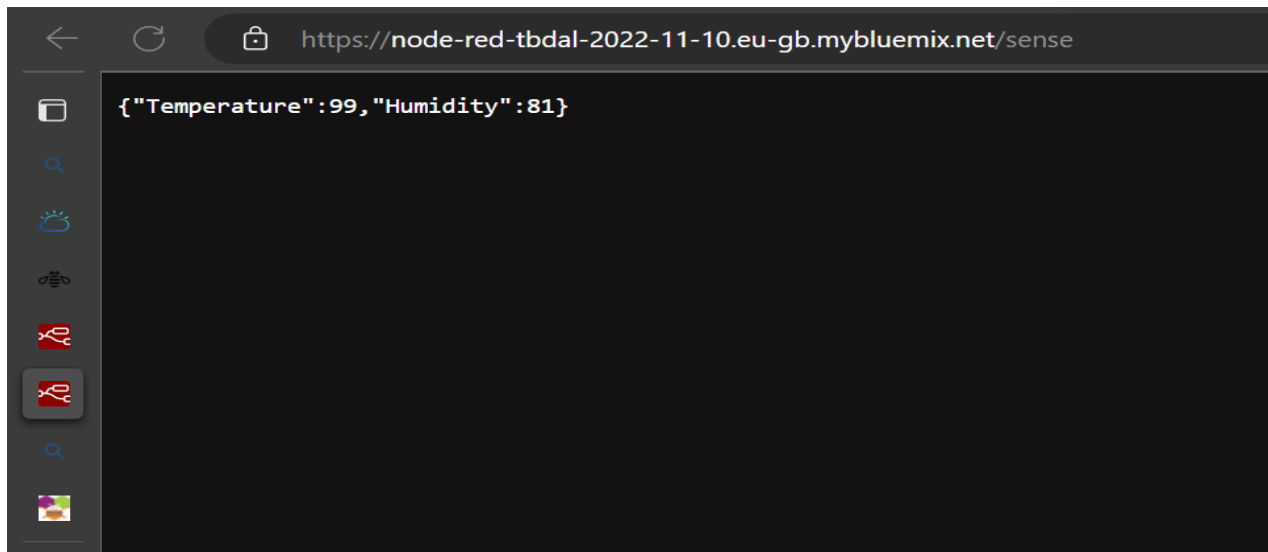
Python 3.7.0 Shell

File Edit Shell Debug Options Window Help

```

Temperature = 6 C Humidity = 44 % to IBM Watson
Temperature = 18 C Humidity = 23 % to IBM Watson
Temperature = 82 C Humidity = 26 % to IBM Watson
Temperature = 67 C Humidity = 69 % to IBM Watson
Temperature = 58 C Humidity = 33 % to IBM Watson
Temperature = 61 C Humidity = 43 % to IBM Watson
Temperature = 20 C Humidity = 79 % to IBM Watson
Temperature = 20 C Humidity = 38 % to IBM Watson
Temperature = 64 C Humidity = 36 % to IBM Watson
Temperature = 56 C Humidity = 57 % to IBM Watson
Temperature = -5 C Humidity = 31 % to IBM Watson
Temperature = 22 C Humidity = 49 % to IBM Watson
Temperature = 15 C Humidity = 83 % to IBM Watson
Temperature = 34 C Humidity = 80 % to IBM Watson
Temperature = 72 C Humidity = 87 % to IBM Watson
Temperature = -1 C Humidity = 20 % to IBM Watson
Temperature = 38 C Humidity = 71 % to IBM Watson
Temperature = 51 C Humidity = 37 % to IBM Watson
Temperature = 89 C Humidity = 13 % to IBM Watson
Temperature = -24 C Humidity = 27 % to IBM Watson
Temperature = 56 C Humidity = 41 % to IBM Watson
Temperature = 79 C Humidity = 27 % to IBM Watson
Temperature = 87 C Humidity = 60 % to IBM Watson
Temperature = -24 C Humidity = 12 % to IBM Watson
Temperature = -16 C Humidity = 59 % to IBM Watson
Temperature = 8 C Humidity = 39 % to IBM Watson
Temperature = -17 C Humidity = 61 % to IBM Watson
Temperature = 7 C Humidity = 54 % to IBM Watson
Temperature = 45 C Humidity = 28 % to IBM Watson

```



FEATURE 2 :

The screenshot displays two side-by-side windows. The left window is a Python 3.7.0 Shell showing a list of published data points. The right window is the IBM Watson IoT Platform dashboard, showing a table of recent events.

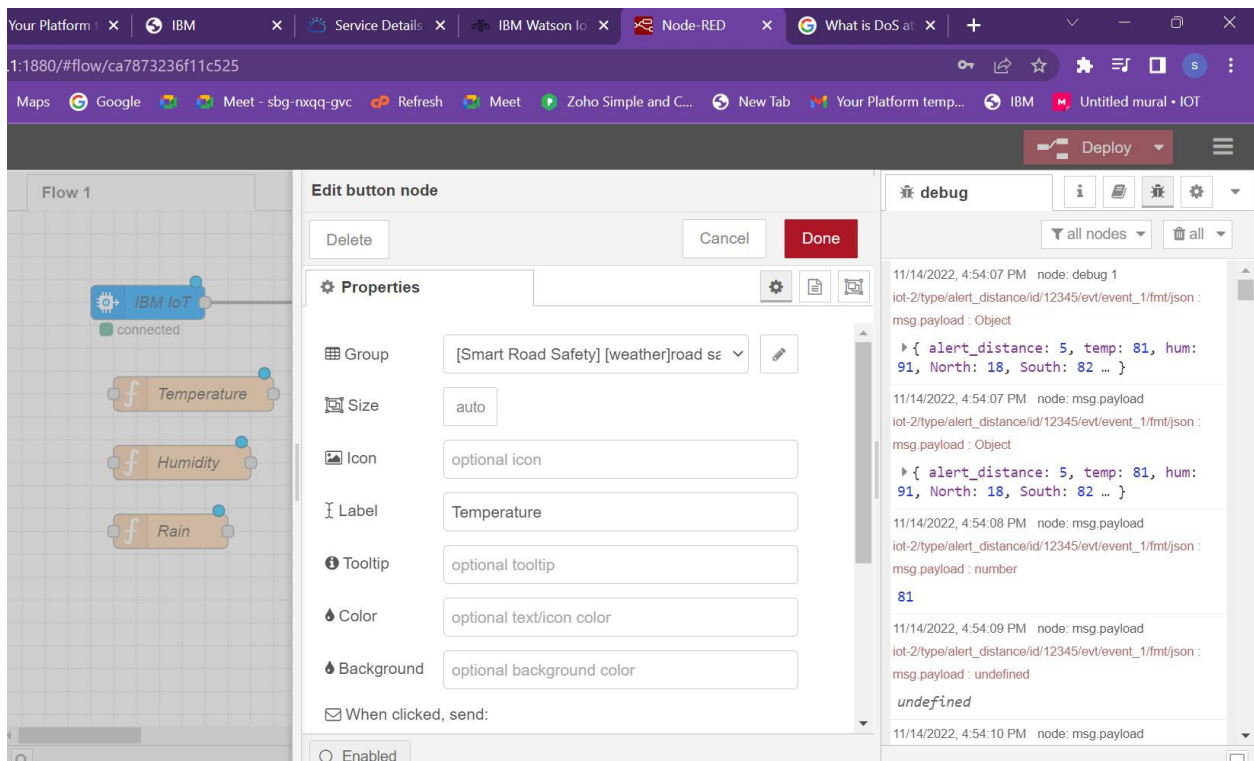
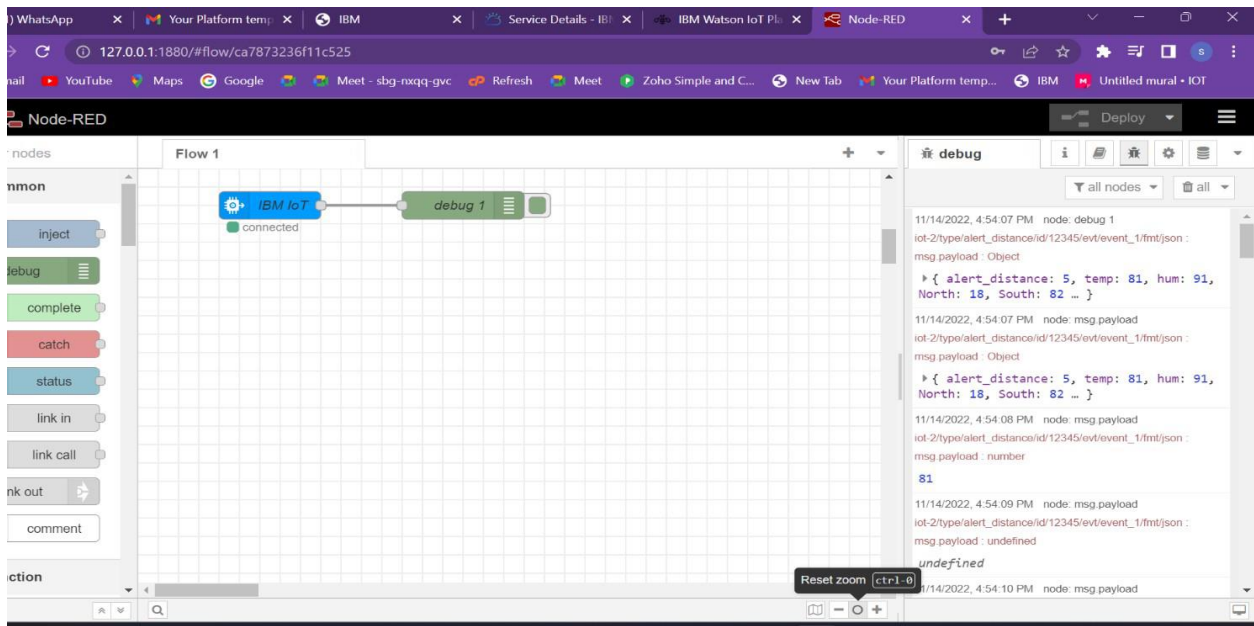
Python 3.7.0 Shell Output:

Published Temperature	Published Humidity	to IBM Watson
70 C	92 %	to IBM Watson
68 C	50 %	to IBM Watson
47 C	29 %	to IBM Watson
72 C	52 %	to IBM Watson
59 C	89 %	to IBM Watson
63 C	87 %	to IBM Watson
55 C	98 %	to IBM Watson
74 C	27 %	to IBM Watson
46 C	43 %	to IBM Watson
48 C	50 %	to IBM Watson
25 C	30 %	to IBM Watson
0 C	71 %	to IBM Watson
64 C	75 %	to IBM Watson
88 C	98 %	to IBM Watson
47 C	75 %	to IBM Watson
51 C	74 %	to IBM Watson
32 C	41 %	to IBM Watson
7 C	100 %	to IBM Watson
56 C	47 %	to IBM Watson
12 C	65 %	to IBM Watson
79 C	77 %	to IBM Watson
31 C	81 %	to IBM Watson
9 C	22 %	to IBM Watson
20 C	58 %	to IBM Watson
37 C	16 %	to IBM Watson
8 C	76 %	to IBM Watson
9 C	32 %	to IBM Watson
16 C	90 %	to IBM Watson
35 C	70 %	to IBM Watson
51 C	75 %	to IBM Watson
76 C	75 %	to IBM Watson
55 C	55 %	to IBM Watson
4 C	39 %	to IBM Watson
69 C	45 %	to IBM Watson
76 C	61 %	to IBM Watson
82 C	100 %	to IBM Watson
19 C	44 %	to IBM Watson
11 C	24 %	to IBM Watson
52 C	24 %	to IBM Watson
59 C	67 %	to IBM Watson

IBM Watson IoT Platform Dashboard:

The recent events listed show the live stream of data that is up

Event	Value
IoT Sensor	[{"temp":89,"Humid":52}]
IoT Sensor	[{"temp":52,"Humid":24}]
IoT Sensor	[{"temp":11,"Humid":24}]
IoT Sensor	[{"temp":49,"Humid":54}]
IoT Sensor	[{"temp":82,"Humid":100}]





CHAPTER8

TESTING

8.1 TEST CASES

- TEST CASE 1
CLEAR WEATHER - USUAL SPEED LIMIT
- TEST CASE 2
COLD WEATHER-REDUCE SPEED LIMIT AND ALERT THE DRIVER THROUGH MOBILE APP TO DRIVE SLOWLY.
- TEST CASE 3
RAINY WEATHER-FURTHER REDUCE SPEED LIMIT.

8.2.USER ACCEPTANCE TESTING:

- To avoid traffic and reduce the accidents ,dynamic speed & diversion variations based on weather conditions is developed to have a safe journey.The user would be happy and feel less stressed while riding and welcome this concept to implement everywhere.

CHAPTER9

RESULTS

9.1.Performance metrics:

- The system is horizontally scalable,an even higher demands of customers can be served with the NODE RED ,since it is capable of handling upto 10,000 requests per second,It is very light and high performance.Based on the IBM pack we chose, the performance of the website varies.

CHAPTER10

ADVANTAGES&DISADVANTAGES

ADVANTAGES:

- **Helps in reducing the frequency and severity of different types of crashes or accidents**
- **Increases traffic handling capacity at a given intersection**
- **Dynamic sign updating**
- **Cheaper and low requirement of the microcontroller**
- **Low battery consumption since processing is done mostly by node red service in cloud.**

DISADVANTAGES:

- **Excessive delay due to time allocated by the traffic signals.**
- **Some drivers disobey these signals**
- **Dependent on opeanweather api and speed reduction is same for large area in scale of cities.**

CHAPTER11

CONCLUSION

- **Our project is used for serving as a replacement of static sign boards for a comparatively lower cost and can be implemented in a very near future.It mainly helps to reduce the number of accidents and maintain a more peaceful traffic atmosphere to have a smooth journey.**

CHAPTER12

FUTURESCOPE

- **Intelligence Networks of integrated sensors detect weather conditions that impact road safety. Road Weather Information Systems (RWIS) in use today are limited because they only collect data from a small set of weather stations. A larger future network could use automated weather stations to collect atmospheric and weather data and instantly upload it to the cloud. Dynamic temperature-sensitive paint could be used to highlight invisible roadway conditions like black ice,which results in smoother traffic flows and increase accuracy about road conditions.**

GITHUB LINK:

<https://github.com/IBM-EPBL/IBM-Project-4001-1658678914>

Team Id : PNT2022TMID39403

Project Name : SIGNS WITH SMART CONNECTIVITY FOR BETTER ROAD SAFETY

Team Id : PNT2022TMID39403

Dinesh V -510119104005

Saran C -510119104019

Manoj J -510119104013

Gunasekaran T -510119104009