Signs with Smart Connectivity for Better Road Safety

# Team ID: PNT2022TMID52316

SUBMITTED BY,

|  |
| --- |
| SAHAYA HEBSIBA S 963519106028 |
| BHAVITHRA A 963519106015 |
| SONIYA S 963519106034 |
| SUJI S 963519106036 |

# In partial fulfilment for the award of the degree of

# BACHELOR OF ENGINEERING

**in**

# ELECTRONICS AND COMMUNICATION ENGINEERING

**STELLA MARY’S COLLEGE OF ENGINEERING**

**ARUTHENGANVILAI**

# ANNA UNIVERSITY: CHENNAI – 600 025

# INTRODUCTION

* 1. Project Overview
  2. Purpose

# LITERATURE SURVEY

* 1. Existing problem
  2. References
  3. Problem Statement Definition

# IDEATION & PROPOSED SOLUTION

* 1. Empathy Map Canvas
  2. Ideation & Brainstorming
  3. Proposed Solution
  4. Problem Solution fit

# REQUIREMENT ANALYSIS

* 1. Functional requirement
  2. Non-Functional requirements

# PROJECT DESIGN

* 1. Data Flow Diagrams
  2. Solution & Technical Architecture
  3. User Stories

# PROJECT PLANNING & SCHEDULING

* 1. Sprint Planning & Estimation
  2. Sprint Delivery Schedule
  3. Reports from JIRA

# CODING & SOLUTIONING (Explain the features added in the project along with code)

* 1. Feature 1
  2. Feature 2
  3. Database Schema (if Applicable)

# TESTING

* 1. Test Cases
  2. User Acceptance Testing

# RESULTS

* 1. Performance Metrics

# ADVANTAGES & DISADVANTAGES

1. **CONCLUSION**

# FUTURE SCOPE

1. **APPENDIX**

Source Code

# INTRODUCTION

* 1. **Project Overview**

The goal of this project is to replace the static signboards with smart connected

sign boards to get the speed limitations from a web app using weather API and update it automatically based on the weather conditions, set diversions through API and warn drivers for school zones and hospital zones

* 1. **Purpose**

To replace the static signboards, smart connected sign boards are used.

* These smart connected sign boards get the speed limitations from a web app using weather API and update automatically.
* Based on the weather changes the speed may increase or decrease.
* Traffic diversion signs are displayed.
* Messages indicating schools, hospitals, police station zones

# LITERATURE SURVEY

* 1. **Existing problem**

# Digital Notice Board Based on IOT

This project presents a digital notice board using IoT module. The idea behind this project is to

provide its users with a simple, fast and reliable way to put up important notices in an LED where the user can send a message to be displayed in the LED. The message can be sent through an android application designed in this project, through the IoT module. So, notices can be put up in an LED display from any location in the world. It uses a microcontroller for system control, IoT based technology for communication and sends the message through the android application. The project consists of Arduino UNO board, IoT module, an LED, and an android application for user interface with the hardware. This device can be used anywhere irrespective of the place of deployment provided mobile network connectivity is available. This is a project that displays messages that the user desires, on an LED Display Matrix. The Display consists of 256 LED lights, sequentially arranged in 8 rows and 32 columns (8\*32). Apart from the display, the project consists of a Node MCU controller which helps the system to connect to the Wi-Fi. This system makes use of Google Assistant to accept speech inputs from user, through user’s Android smartphone. User needs to login into their Google account. A USB cable acts as the power cable for the system. The speech input is converted into a text display in an alphanumeric format which is predefined. The displayed message will either scroll or remain static, based on the size of display and length of message. This project can widely use in offices, schools, educational institutions as well as government and corporate offices to display important notices and messages. This can prove to help users save a lot of time as against the use of traditional pin and paper notice display.

# Internet of Things Based Notifications Using Smart Notice Board

Conventional Notice Board employs manual display and monitoring with papers and ledgers. The Target users are unaware of information displayed on the notice board. The objective of the project is to display the message on the notice board from anywhere and anytime, that even provides broadcast alerts to the target users. The system was designed and developed using the Internet of Things. Arduino board integrates the display unit, Mobile App and SMS Agent through Internet. The message to be displayed on the notice board is sent through a mobile app to the board with Arduino. As soon as the message is displayed, SMS alert is sent to the target users. A system of efficient Notice Board display controlled through the Internet is accomplished and presented in this paper.

# Internet of Things Based Notifications Using Smart Notice Board

Conventional Notice Board employs manual display and monitoring with papers and ledgers. The Target users are unaware of information displayed on the notice board. The objective of the project is to display the message on the notice board from anywhere and anytime, that even provides broadcast alerts to the target users. The system was designed and developed using the Internet of Things. Arduino board integrates the display unit, Mobile App and SMS Agent through Internet. The message to be displayed on the notice board is sent through a mobile app to the board with Arduino. As soon as the message is displayed, SMS alert is sent to the target users. A system of efficient Notice Board display controlled through the Internet is accomplished and presented in this paper.

# An IoT based Smart Monitoring System for Vehicles

There is increased adoption of penalty and fine for traffic rule violators in the public sector but there is a tendency for people to evade from those imposed fines and restrictions for their own safety. Our system will completely monitor all the traffic violations namely over speeding, rash driving, drunken driving, driving without a seat belt, and so on right from the starting of the car. There is an increasing demand to develop a system to check passengers without coming out of the vehicle. A new system for the police force to check the vehicle's details with a smart device placed in the vehicle. The device is equipped with speed monitoring, Alcohol detection, Seat belt checking, etc. If any violation is detected the controller sends an emergency data to the cloud, thus the vehicle is in continuous monitoring mode, and RTO will get updates about the vehicles which are violating rules. Alcoholic breath sensor will continuously monitor the driver's breath, speed sensor will be connected with the speedometer and checks for over speeding, Seat belt sensor will warn the driver if he/she is not using the seat belt, vehicle details including license, pollution details, insurance, etc. will be uploaded to the server or cloud. If any of the above things are violated, automatically defaulter will be imposed fines and the details will be sent to the Motor vehicle department.

# Congestion Adaptive Traffic Light Control and Notification Architecture Using Google Maps APIs:

Controlling of traffic signals optimally helps in avoiding traffic jams as vehicle volume density changes on temporally short and spatially small scales. Nowadays, due to embedded system development with the rising standards of computational technology, condense electronics boards as well as software packages, system can be developed for controlling cycle time in real time. At present, the traffic control systems in India lack intelligence and act as an open-loop control system, with no feedback or sensing network, due to the high costs involved. This paper aims to improve the traffic control system by integrating different technologies to provide intelligent feedback to the existing network with congestion status adapting to the changing traffic density patterns. The system presented in this paper aims to sense real-time traffic congestion around the traffic light using Google API crowdsource data and hence avoids infrastructure cost of sensors. Subsequently, it manipulates the signal timing by triggering and conveying information to the timer control system. Generic information processing and communication hardware system designed in this paper has been tested and found to be functional for a pilot run in real time. Both simulation and hardware trials show the transmission of required information with an average time delay of 1.2 seconds that is comparatively very small considering cycle time.

Mishra, Sumit Kumar, Devanjan Bhattacharya and Ankit K. Gupta. “Congestion Adaptive Traffic Light Control and Notification Architecture Using Google Maps APIs.” Data 3 (2018): 67.

# An IoT based Weather Information Prototype Using WeMos:

The Internet of Things (IOT) describes the interconnection of devices and people through the traditional internet and social networks for various day-to-day applications like weather monitoring, healthcare systems, smart cities, irrigation field, and smart lifestyle. IOT is the new revolution of today’s internet world which monitors live streaming of the entire world’s status like temperature, humidity, thunderstorm, earthquake, floods etc. that can stagger an alarm to human life. This paper proposes a low-cost weather monitoring system which retrieves the weather condition of any location from the cloud database management system and shows the output on an OLED display. The proposed system uses an ESP8266-EX microcontroller based WeMos D1 board and it is implemented on Arduino platform which is used to retrieve the data from the cloud. The main objective of this paper is to view weather conditions of any location and allows to access the current data of any station.

R. K. Kodali and A. Sahu, "An IoT based weather information prototype using WeMos," 2016 2nd International Conference on Contemporary Computing and Informatics (IC3I), 2016, pp. 612-616, DOI: 10.1109/IC3I.2016.7918036.

# IOT Based Weather Monitoring and Reporting System Project

The IOT based Weather Monitoring and Reporting System project is used to get Live reporting of weather conditions. It will Monitor temperature, humidity, moisture and rain level. Suppose Scientists/nature analysts want to monitor changes in a particular environment like volcano or a rain-forest. And these people are from different places in the world. In this case, SMS based weather

monitoring system has some limitations. Since it sends SMS to few numbers. And time for sending SMS increases as the number of mobile numbers increases. In order to know the information about weather of a particular place then they have to visit those particular sites. Where everyone can see it.

Anita M. Bhagat ,Ashwini G. Thakare ,Kajal A. Molke , Neha S. Muneshwar ,Prof. V. Choudhary IOT Based Weather Monitoring and Reporting System ,2019 .

# Incorporating Weather Updates for Public Transportation Users of Recommendation Systems:

This work presents a system for augmenting the functionality of Yelp-like recommendation sites by enabling users to search for places bounded by travel-time when using public transportation, and modifying recommendations based on updated weather conditions. Using public transport, although is cheaper and efficient, entails that only fixed places of boarding/exiting may be used which, in turn, implies walking to (from) a particular location from (to) a given station. Given the impact of the weather on the mood and activities, preferences for a certain type of services may need to be dynamically adjusted based on the current weather or the near-future forecast, modulo travel-routes

to preferred locations. In this work, we develop a model to predict a user’s preferred mode of transport (car, or public transit) from their old check-ins and incorporate the weather context into

the recommendation process We use event-based modeling to control the extent of walking depending on user-defined tolerance information and live weather conditions. We implemented a web application (both desktop and mobile platforms), utilizing existing tools such as Google Maps Direction API and OpenWeatherMap API for retrieving real-time information.

# System Natural Data is now becoming more valuable in a day to get real-time data for natural data:

Physical monitoring of the environment allows for the identification of areas suitable for agriculture, industry, and other purposes. In this article, the Arduino-UNO microcontroller- based board is used for the data acquisition strategy and the use of analog and digital sensors. Temperature, humidity, light intensity and gas concentrations can be monitored in real-time [4] [9-12] [13-16].

# The impact of daily weather on daily travel trips :

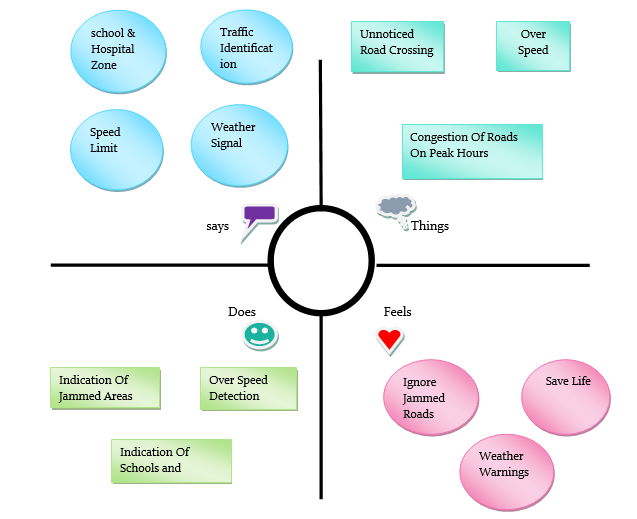
It is of increasing social interest - climate change and increasing scarcity, understanding the climatic implications of travel behavior, especially walking and biking. Recently, various courses are travel, health, and biometeorology.

* 1. **Problem Statement Definition**

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

# IDEATION & PROPOSED SOLUTION

**3.1.EMPATHY MAP CANVAS**



**3.2 Ideation & Brainstorming**

****

**3.3 Proposed Solution**

|  |  |  |
| --- | --- | --- |
| **S No** | **Parameter** | **Description** |
| 1. | Problem statement (Problem to be solved) | We have to come up with the digital sign boards on which the signs can be changed dynamically. |
| 2. | Idea/Solution description | To develop a web application through which the data of the road diversions, accident prone areas and information sign boards can be entered. |
| 3. | Novelty/Uniqueness | Data is retrieved and displayed on the sign boards dynamically to avoid congestion instantly. |
| 4. | Social Impact/Customer Satisfaction | This form of digital signs will ease the burden on drivers, reduce the traffic accidents and enhances the road safety. |
| 5. | Business modal (Revenue Model) | Companies that are dependent on driving and transportation get benefited. |
| 6. | Scalability of the Solution | The software can be used to analysis the road damages, mishaps and to clear the traffic congestion quickly |

**3.4 Problem Solution fit**

|  |  |  |
| --- | --- | --- |
| **1.CUSTOMER SEGMENT CS**     * Awareness towards road infrastructure. * Data is useful in understanding the road user behavior & flow of traffic | **6.CUSTOMER CONTRAINTS CC**   * Customers no need to spend any power (or) Network connection. * If they fail to obey traffic rules, then their money was marked as charged fines as per the court. | **5.AVAILABLE SOLUTIONS AS**     * Record traffic offenses & provide existing data to collect monitor, analyze with the periodic maintenance. * Monitoring the road events in low light (or) in bad weather conditions. |
| **2.JOBS-TO-BE-DONE/PROBLEMS J&P**     * Keep providing of valid data through dynamic sign board system helps to allow people predicting day to day complexities face along the roadway. * Flow of data updating is quick & speedy, convenient and flexible. | **9.PROBLEM ROOT CAUSE RC**   * Especially most of the people busy with their mobile phone actions leads to get distracted & they lose attention of traffic. * Simply, road accidents either due to carelessness(or) due to lack of road safety awareness | **7.BEHAVIOUR BE**     * The IOT cloud behaves as the instructor to the smart board about the road condition in   regular intervals. |
| **3.TRIGGERS TR**   * Creating a note that gives a direction on how to recognize that system effectively. | **10.YOUR SOLUTION SL**   * Pre-function record of specific voice record mode of data along with LED display provide in waiting time at traffic signs. * In this proposed system is interface with Rain Drop Sensor check if it rainy there, to transmit data over IOT helps to display on LED to along with WIFI connection of internet changing data dynamically with current reporting of event sensing flow of data. | **8**.**CHANNELS OF BEHAVIOUR CH**   * The traveler can use the smart board signs to know the road condition whether the traveler from anywhere and the following instructions is one of the big tasks for the traveler. * The customers can directly send a feedback mail or message to the departments (Nearby Patrol Officers) |
| **4.EMOTIONS: BEFORE/AFTER EM**     * Before: Already existing of man-made static boards raising challenges due to un updated real-time issues & current changes of road events. * After: This system is better than existing method, of having automation of road signs & communication strategy in the manner of smart city to alert the drivers to reduce relay & congestion while travelling time. |

# REQUIREMENT ANALYSIS

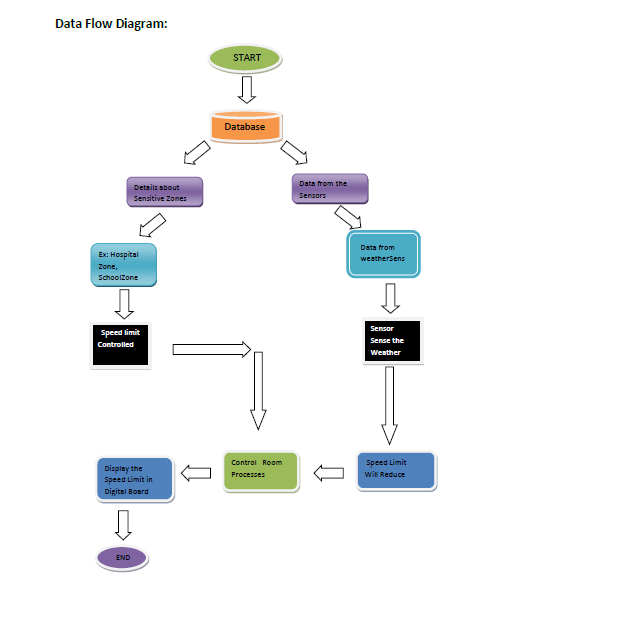
* 1. **Functional requirement**

|  |  |  |
| --- | --- | --- |
| **S. NO** | **FUNCTIONAL**  **REQUIREMENTS (EPIC)** | **SUB REQUIREMETS (STORY/SUB-**  **TASK)** |
| **1.** | User Visibility | Sign Boards should be made with LED’s which are bright colored and are capable of attracting the driver attention but it should also not be too distracting or blinding cause  it may lead to accidents. |
| **2.** | User Need | The smart sign boards should be placed frequently in places it is needed and less in places where it is not needed much to avoid  confusion for the user during travel. |
| **3.** | User Understanding | For better understanding of the driver, the signs should be big, clear and legible and it can also include illustrations which will make  it easily understandable to the driver. |
| **4.** | User Convenience | The display should be big enough that it should even be visible from far distance  clearly. |

**4**.**2 Non-Functional requirements**

|  |  |  |
| --- | --- | --- |
| **S. NO** | **NON-FUNCTIONAL**  **REQUIREMENTS** | **DESCRIPTION** |
| **1.** | Usability | It should be able to Upgrade and Update when  there is a need for it. |
| **2.** | Security | It should have good security system so that no  one is able to hack and display |
| **3.** | Reliability | It should be able to display to information  correctly and error-free. |
| **4.** | Performance | It should be able to automatically update itself  when certain weather or traffic problem occurs. |
| **5.** | Availability | It should be available 24/7 so that it can be  beneficial to the customer i.e. the driver. |
| **6.** | Scalability | It should be able to easily change and upgrade  according to change and need in requirement. |

# PROJECT DESIGN

* 1. **Data Flow Diagrams**
* 

**5.2 Solution & Technical Architecture**

# SOLUTION ARCHITECTURE:

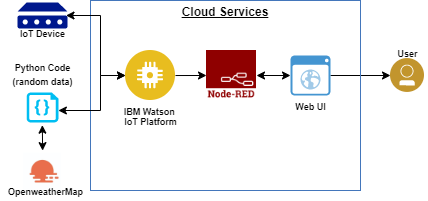
Solution architecture is a complex process with many sub-processes that bridges the gap between business problems and technology solutions. Its goals are to:

* Find the best tech solution to solve existing business problems.
* Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
* Define features, development phases, and solution requirements.

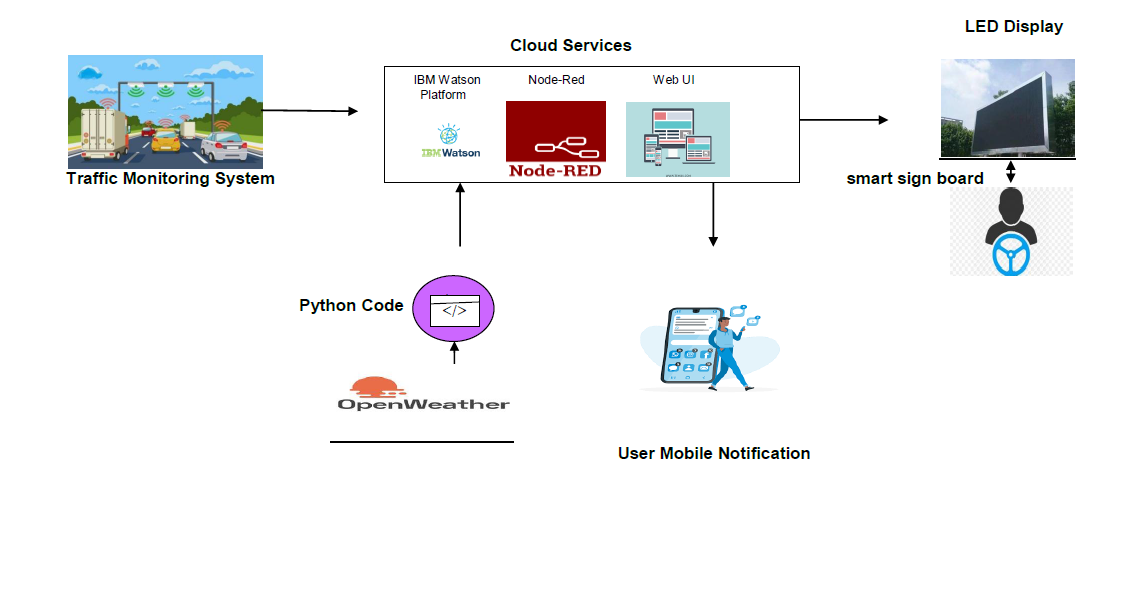
# Provide specifications according to which the solution is defined, managed and delivered

# 

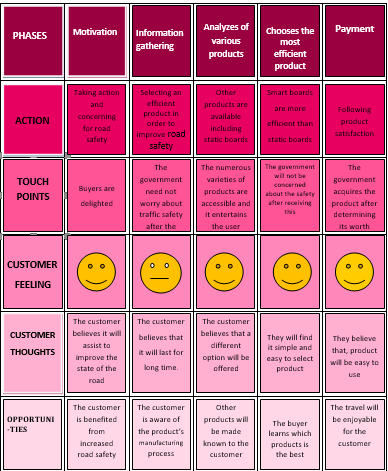
# 



**TECHNICAL ARCHITECTURE:**

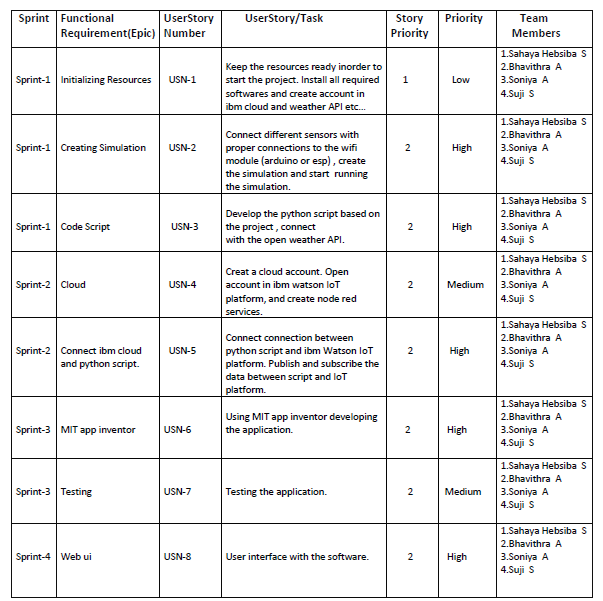
****

* 1. **User Stories**

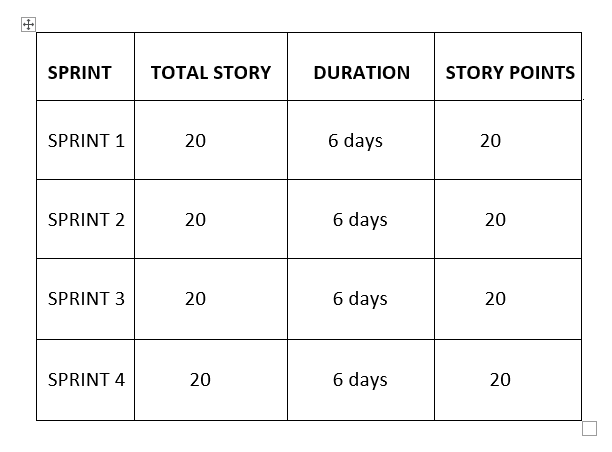


# PROJECT PLANNING & SCHEDULING

* 1. **Sprint Planning & Estimation**



* 1. **Sprint Delivery Schedule**



**Velocity:**

The average velocity (AV) per iteration unit (story points per day) can be defined as sprint duration by velocity (points per sprint)

AV = Sprint duration / Velocity

**Given:**

Sprint duration = 6days

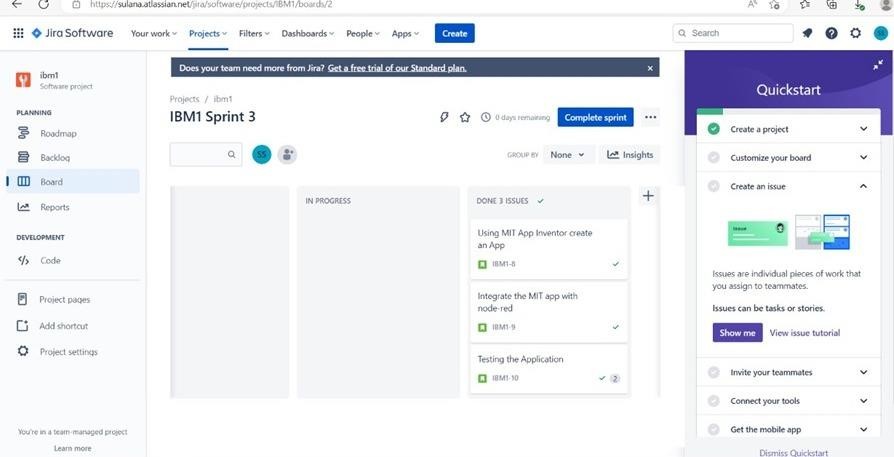
Velocity = 20

AV = 6 / 20

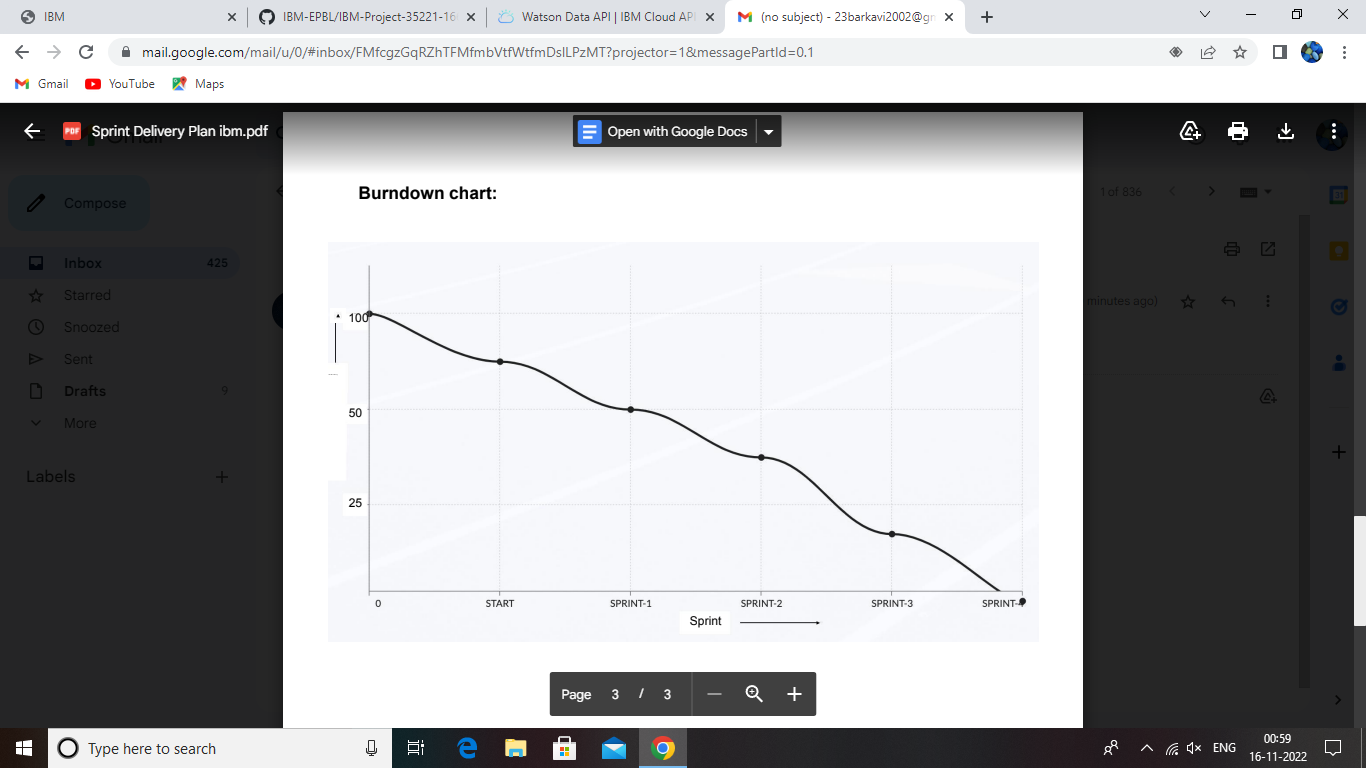
= 0.3

AV = 0.3

**6.3 Report from JIRA**



Burndown Chart:

****

# 7.CODING & SOLUTIONING (Explain the features added in the project along with code)

* 1. **Feature 1 -GET WEATHER DETAILS FOR GIVEN LOCATION**

Program Code:

> weather.py

This file is a utility function that fetches the weather from OpenWeatherAPI. Itreturns only certain required parameters of the API response.

# Python code

import requests as reqs

def get (my Location, APIKEY): apiURL =

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

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)

> publishData.py

This code pushes data to the cloud and logs data. IBM Cloud is configured such that the data is displayed in the following website: CLICK TO OPEN NODE RED DASHBOARD

# Python code

# IMPORT SECTION STARTS

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

# IMPORT SECTION ENDS

# # API CONFIG SECTION STARTS

myConfig = { "identity" : {

"orgId" : "kjbrqi",

"typeId" : "temp", "deviceId" : "89032"

},

"auth" : {

"token" : "WjW4q@Ko(QVhH(GjZN"

}

}

# 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)

# FUNCTIONS SECTION ENDS

> brain.py

This file is a utility function that returns only essential information to be displayed at the hardware side and abstracts all the unnecessary details. This is where the code

flow logic is implemented.

from datetime import datetime as dt

from publishData import logData2Cloud as log2cloud

# IMPORT SECTION ENDS

# # UTILITY LOGIC SECTION STARTS

def processConditions(myLocation,APIKEY,localityInfo): weatherData = weather.get(myLocation,APIKEY)

log2cloud(myLocation,weatherData["temperature"],weatherData["visibility"]) finalSpeed = localityInfo["usualSpeedLimit"] if "rain" not in weatherData else

localityInfo["usualSpeedLimit"]/2

finalSpeed = finalSpeed if weatherData["visibility"]>35 else finalSpeed/2

if(localityInfo["hospitalsNearby"]):# hospital zone

doNotHonk = True else:

if(localityInfo["schools"]["schoolZone"]==False):# neither school nor hospital zone

doNotHonk = False else:

# school zone

now = [dt.now().hour,dt.now().minute] activeTime = [list(map(int,\_.split(":"))) for \_ in

localityInfo["schools"]["activeTime"]]

doNotHonk = activeTime[0][0]<=now[0]<=activeTime[1][0] and activeTime[0][1]<=now[1]<=activeTime[1][1]

return({

"speed" : finalSpeed, "doNotHonk" : doNotHonk

})

# UTILITY LOGIC SECTION ENDS

> main.py

The code that runs in a forever loop in the micro-controller. This calls all the util functions from other python files and based on the return value transduces changesin the output hardware display.

# Python code

# IMPORT SECTION STARTS

import brain

# IMPORT SECTION ENDS

# # USER INPUT SECTION STARTS

myLocation = "Chennai,IN"

APIKEY = "bf4a8d480ee05c00952bf65b78ae826b"

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))

'''

MICRO CONTROLLER CODE WILL BE ADDED IN SPRINT 3 AS PER OURPLANNED SPRINT SCHEDULE

'''

# MICRO-CONTROLLER CODE ENDS

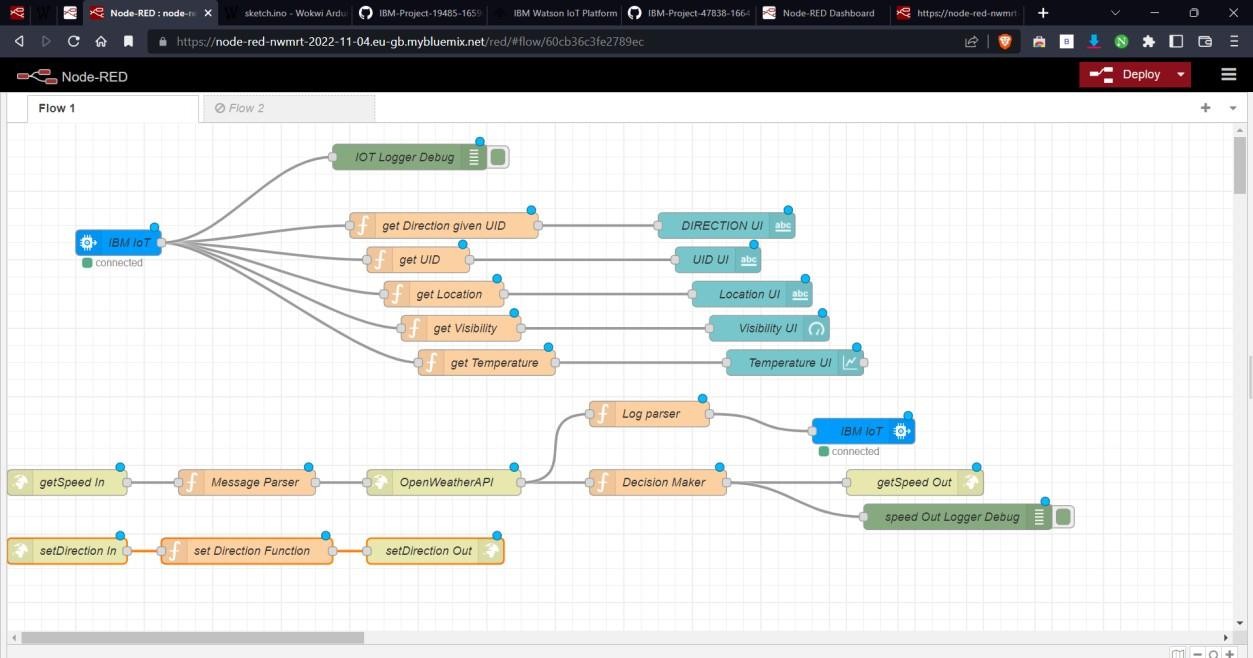
OUTPUT:



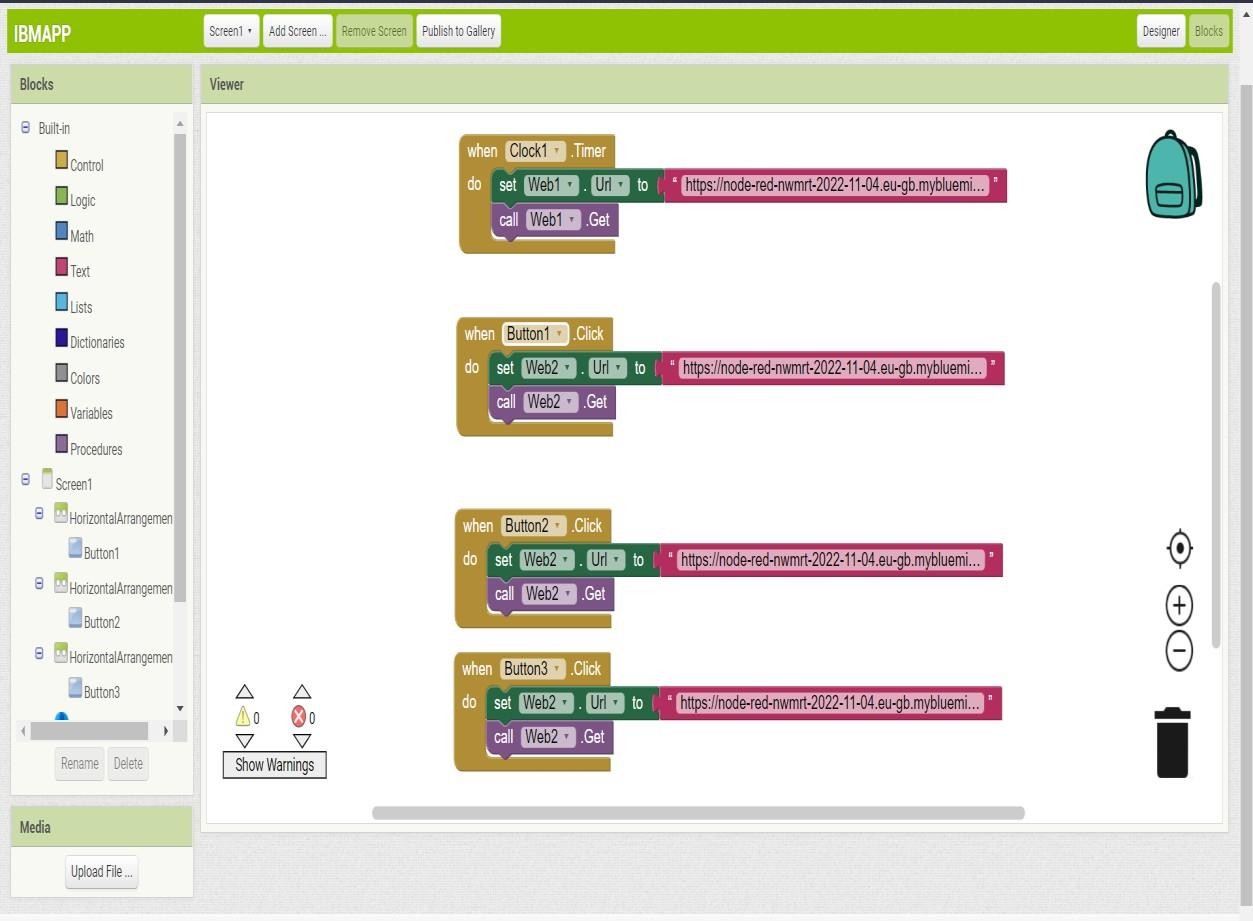
* 1. **Feature 2**

The temperature and the location data are exactly displayed in the webpage using Node – Red and the for that is randomized using IBM Watson. A device is created for that purpose and is simulated to send data to node – red.

OVERALL NODE-RED FLOW



MIT app block code for managing traffic.



# TESTING

* 1. **Test Cases**
* TEST CASE 1

Temperature': 303.03, 'Humidity': 51, 'Pressure': 1010, 'Message': 'SLOW DOWN, SCHOOL IS NEAR', 'Sign': '', 'Speed': '', 'Visibility': 'Clear Weather'

* TEST CASE 2

Temperature': 303.03, 'Humidity': 51, 'Pressure': 1010, 'Message': '', 'Sign': 'Left Diversion

<-', 'Speed': 'SLOW DOWN, Speed Limit Exceeded', 'Visibility': 'Clear Weather'

* TEST CASE 3

Temperature': 303.03, 'Humidity': 51, 'Pressure': 1010, 'Message': 'SLOW DOWN, HOSPITAL NEARBY', 'Sign': 'Left Diversion <-', 'Speed': '', 'Visibility': 'Clear Weather'

* TEST CASE 4

Temperature': 303.03, 'Humidity': 51, 'Pressure': 1010, 'Message': 'NEED HELP, POLICE STATION NEARBY', 'Sign': 'U Turn', 'Speed': 'Moderate Speed', 'Visibility': 'Clear Weather'.

* 1. **User Acceptance Testing**

In engineering and its various sub disciplines, acceptance testing is black box testing performed on a system (e.g. software, lots of manufactured mechanical parts, or batches of chemical products) prior to its delivery. It is also known as functional testing, black-box testing, release acceptance, QA testing, application testing, confidence testing, final testing, validation testing, or factory acceptance testing. In software development, acceptance testing by the system provider is often distinguished from acceptance testing by the customer (the user or client) prior to accepting transfer of ownership. In such environments, acceptance testing performed by the customer is known as user acceptance testing (UAT). This is also known as end-user testing, site (acceptance) testing, or field (acceptance) testing. A smoke test is used as an acceptance test prior to introducing a build to the main testing process. Acceptance test cards are ideally created during sprint planning or iteration planning meeting, before development begins so that the developers have a clear idea of what to develop. Sometimes (due to bad planning!) acceptance tests may span multiple stories (that are not implemented 29 in the same sprint) and there are different ways to test them out during actual sprints. One popular technique is to mock external interfaces or data to mimic other stories which might not be played out during an iteration (as those stories may have been relatively lower business priority). A user story is not considered complete until the acceptance tests have passed. The acceptance test suite is run against the supplied input data or using an acceptance test script to direct the testers. Then the results obtained are compared with the expected results. If there is a correct match for every case, the test suite is said to pass. If not, the system may either be rejected or accepted on conditions previously agreed between the sponsor and the manufacturer. The objective is to provide confidence that the delivered system meets the business requirements of both sponsors and users. The acceptance phase may also act as the final quality gateway, where any quality defects not previously detected may be uncovered. In these testing procedures the project is given to the customer to test whether all requirements have been fulfilled and after the user is fully satisfied. The project is perfectly ready. If the user makes request for any change and if they found any errors those all errors have to be taken into consideration and to be correct it to make a project a perfect project.

# RESULTS

**9.1 Performance Metrics**

The performance of the website varies based on the software chosen for implementation. Built upon NodeJS, a light and high performance engine, NodeRED is capable of handling upto 10,000 requests per second. Moreover, since the system is horizontally scalable, a even higher demand of customers can be served.

# ADVANTAGES & DISADVANTAGES

* **ADVANTAGES**
  + - Signs with smart connectivity are an inexpensive and flexible medium that can help transmit information according to particular situation and entertain passengers.
    - The digital signboards help in reducing the air pollution due the emission of vehicles in heavy traffic area.
    - The drivers can able to know about the weather condition and accordingly follow the speed limit displayed on the sign boards.
    - The increased flexibility of these digital sign boards makes it easy for any private or government department to change the message as per the need of the hour.
    - The driver can easily find the route and navigation instructions to reach the destination.
    - The speed of the vehicle can be identified using location sensor.
    - The digitals sign boards and the app are user-friendly.

# DISADVANTAGES

# The size of the display determines the requirement of the micro controller

* + - Dependent on OpenWeatherAPI and hence the speed reduction is same for a large area in the scale of cities.

# CONCLUSION

The project concluded by replacing the static signboards with smart connected digital sign boards. Digital road signs are an important part of modern infrastructure and are becoming increasingly common. Digital road signs are becoming more common as technology improves and more states adopt them. The use of digital road signs is expected to continue to grow in the future as it would be observed user-friendly, economic, environment friendly, profitable promoting road safety. Digital road signs are designed to improve road safety and efficiency by providing real-time information to drivers. These signs can display a variety of information, including speed limits, traffic conditions, and weather warnings.

Digital road signs can help drivers by providing information that is not always available from traditional static signs.

.

# 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 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.

# APPENDIX Source Code :

import wiotp.sdk.device import time

import random import requests, json

myConfig = { "identity": {

"orgId": "kjbrqi",

"typeId": "temp", "deviceId":"89032"

},

"auth": {

"token": "WjW4q@Kc(QVhH(GjZN"

}

}

def myCommandCallback(cmd):

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

client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None) client.connect()

BASE\_URL = "https://api.openweathermap.org/data/2.5/weather?" CITY = "Coimbatore"

URL = BASE\_URL + "q=" + "Coimbatore" + "&appid=" + "fbcb52a2a6c7bbea1396de2b6b17ea8a"

while True:

response = requests.get(URL) if response.status\_code == 200:

data = response.json() main = data['main']

temperature = main['temp'] humidity = main['humidity'] pressure = main['pressure'] report = data['visibility'] repo=random.randint(0,5)

if repo==1:

prt="SLOW DOWN , SCHOOL IS NEAR"

elif repo==3:

prt="SLOW DOWN , HOSPITAL NEARBY"

elif repo==5:

prt="NEED HELP, POLICE STATION NEARBY"

else:

prt="" speed=random.randint(0,150) if speed>=100:

prt3="SLOW DOWN , Speed Limit Exceeded" elif speed>=60 and speed<100:

prt3="Moderate Speed" else:

prt3="Usual speed limit" sign=random.randint(0,5)

if sign==1:

prt2="Right Diversion ->" elif sign==3:

prt2="Left Diversion <-" elif sign==5:

prt2="U Turn" else:

prt2=""

if temperature<=50:

prt4="Fog Ahead, Drive Slow" else:

prt4="Clear Weather"

else:

print("Error in the HTTP request") myData={'Temperature':temperature,'Humidity':humidity,'Pressure':pressure,

'Message':prt, 'Sign':prt2, 'Speed':prt3, 'Visibility':prt4}

client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)

print("Published data Successfully: %s", myData) client.commandCallback = myCommandCallback time.sleep(5)

client.disconnect()