FINAL PROJECT ATTRIBUTES

Date	19 NOVEMBER 2022
Team ID	PNT2022TMID11013
Project Name	Signs and Smart Connectivity for better Road Safety

TEAM MEMBERS :

- **KARTHICK.V**

- **MANOJ KUMAR.S**

- **MARIAN ROSHAN.P**

- **KARTHIKEYAN**

INTRODUCTION

PROJECT OVERVIEW :

To replace the static signboards, smart connected sign boards are used. These smart connected sign boards get the weather data from a web app using weather API and update automatically. Based on the weather changes the speed limit displayed may increase or decrease. Based on the traffic and fatal situations the diversion signs are displayed. Schools, hospitals, warning signs are also displayed accordingly. We make use of random values in python for sensor data as physical hardware is not used

LITERATURE SURVEY :

A literature review is a comprehensive summary of previous researches on the topic. The literature review surveys scholarly articles, books, and other sources relevant to a particular area of research.

- Usha Devi Gandhi Published on "International Conference on Reliability Optimization and Information Technology (ICROIT) 2014". The use of connected vehicles aims to address some of the main problems with transportation in the areas of environment, mobility, and safety. One of the key goals of this project is the application of the Intelligent Transport System (ITS) for safety. The goal of safety application research and industrial projects is to develop the vehicle industry globally. In this project, we concentrate on vehicle-to-vehicle (V2V) communication, which allows linked vehicles to share information with other vehicles on the road and helps to decrease highway accidents. Vehicle to vehicle (V2V) and vehicle to infrastructure (V2I) connectivity based on Wi-Fi, GPS, and dedicated short range communication are ultimately used to connect automobiles (DSRC). VANETS are regarded as one of the most significant safety simulations.

- · Ching-Hao Lai Published on "International Conference on Technologies and Applications of Artificial Intelligence 2014". Smart phones and intelligent automobiles have grown in popularity over the past few years. One type of driving assistance system (DAS) that automatically notifies the driver of traffic sign information via a head-up display (HUD), monitor, or speaker device is the traffic sign recognition system. As a result, this technology helps to decrease driver distraction and raise driving safety. An effective real-time traffic sign recognition technique for intelligent vehicles is proposed in this work. The suggested approach can create an in-vehicle traffic sign recognition system by fusing smart phones with in-vehicle computer gear. The four main stages of this approach are character/icon extraction and identification, traffic sign detection, image preprocessing, and video frame capturing and transmission.

- Ashish Sharma Published on "IEEE India Council International Subsections Conference (INDISCON) 2020".There are more cars on the road today. As a result, managing traffic bottlenecks and accidents on the roads around the world is difficult for society. The performance of the total road safety management system
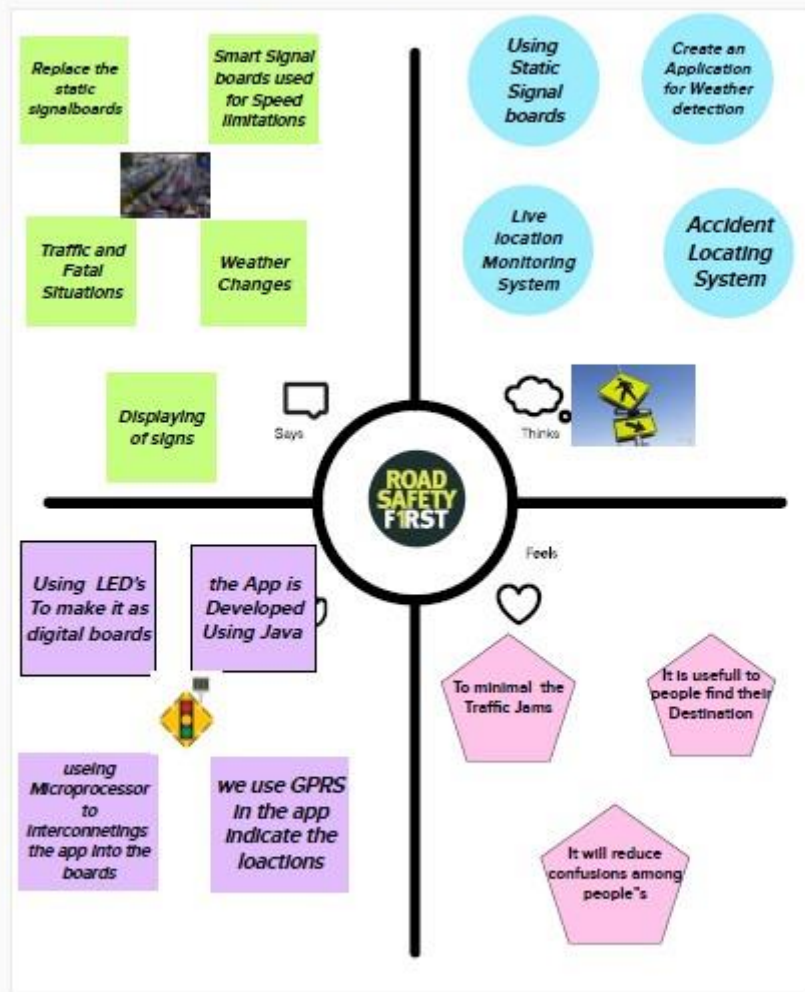
can be greatly enhanced by artificial intelligence (AI) technologies like machine learning (ML) algorithms. Numerous practical applications of AI are employed to make any system intelligent. The Smart Road Traffic Management System (SRTMS) recognises the impact that unplanned modifications have on traffic safety with ease. The SRTMS not only notifies the appropriate authorities but also recognises risky driving behaviours. Real-time monitoring of human activity is made possible via the Internet of Things (IoT).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 Blockchain (BC). Ishan Mishra Published in web source on "April 2020". In terms of travel, the railroad is the most affordable and convenient form of transportation in our nation. Many individuals in India utilize trains to travel to different locations, and some use Indian Railways on a daily basis. As a result, a lot of people entrust the railroads with their lives on a regular basis, putting their lives in danger if they are unsafe or prone to accidents. The level crossing where the railroad tracks and the road converge is the site of many railroad accidents, most of which are caused by human error.

W. H. D. Fernando Published on " International Research Conference on Smart Computing and Systems Engineering (SCSE) 2021". 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 Blockchain (BC). Information sharing on the network is made possible by BC technology in a decentralised, secure, persistent, anonymous, appropriate, and reliable way. Blockchain 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. This recommended model in intelligent transportation systems, but challenging procedure. To address the significant problems they encounter, numerous initiatives have been made. Using a method that first detects a traffic sign's bounding box, the goal of this work is to address the detection and recognition of road traffic signs. When a traffic sign is noticed, it will then be

EMPATHY MAP :

An empathy map is a collaborative tool teams can use to gain a deeper insight into their customers. It helps us to understand the customer's pain, gain and difficulties from their point of view.
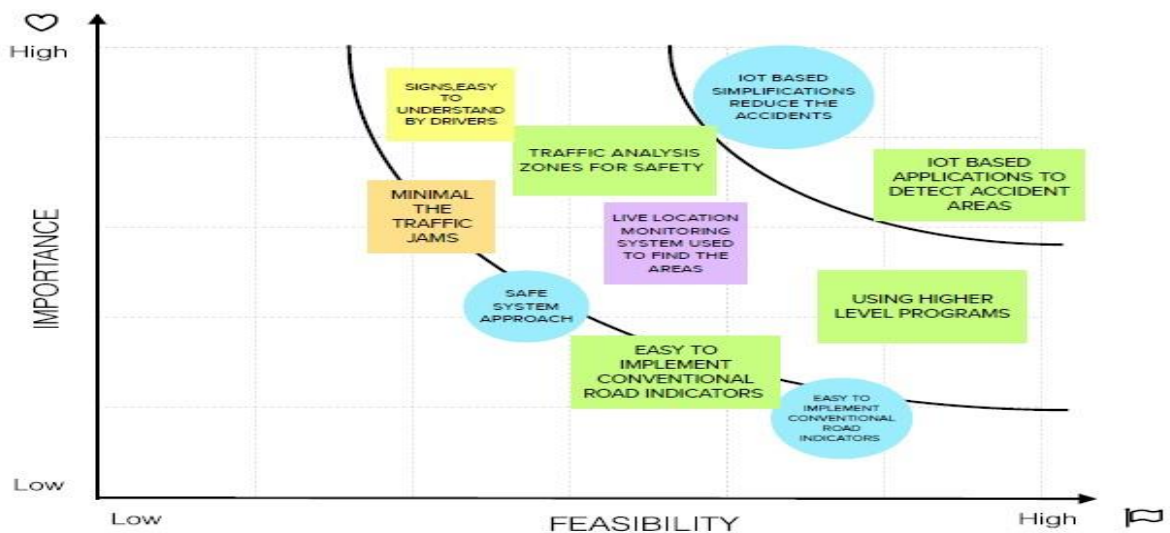
Empathy Map

Says:
- Replace the static signalboards
- Smart Signal boards used for Speed limitations
- Traffic and Fatal Situations
- Weather Changes
- Displaying of signs

Thinks:
- Using Static Signal boards
- Create an Application for Weather detection
- Live location Monitoring System
- Accident Locating System

ROAD SAFETY FIRST

Does:
- Using LED's To make it as digital boards
- the App is Developed Using Java
- useing Microprocessor to interconnetings the app into the boards
- we use GPRS in the app Indicate the loactions

Feels:
- To minimal the Traffic Jams
- It is usefull to people find their Destination
- It will reduce confusions among people"s

IDEATION :

Brainstorming is a group problem-solving method that helped us to gather and organize various ideas and thoughts from teammembers.

BIG IDEAS

LIVE LOCATION MONITORING SYSTEM USED TO FIND THE AREAS

USING HIGHER LEVEL PROGRAMS

SENSOR USED TO AVOID ACCIDENTS

IOT BASED SIMPLIFICATIONS REDUCE THE ACCIDENTS

SIGNS, EASY TO UNDERSTAND BY DRIVERS

SAFE SYSTEM APPROACH

WIRELESS DIGITAL TRAFFIC SIGNS POST

IOT BASED APPLICATIONS TO DETECT ACCIDENT AREAS

TRAFFIC ANALYSIS ZONES FOR SAFETY

MINIMAL THE TRAFFIC JAMS

USE SOME APPLICTIONS TO DETECT WEATHER AND SPEED

EASY TO IMPLEMENT CONVENTIONAL ROAD INDICATORS

Idea Prioritization

High

IMPORTANCE

Low

SIGNS, EASY TO UNDERSTAND BY DRIVERS

IOT BASED SIMPLIFICATIONS REDUCE THE ACCIDENTS

TRAFFIC ANALYSIS ZONES FOR SAFETY

IOT BASED APPLICATIONS TO DETECT ACCIDENT AREAS

MINIMAL THE TRAFFIC JAMS

LIVE LOCATION MONITORING SYSTEM USED TO FIND THE AREAS

SAFE SYSTEM APPROACH

USING HIGHER LEVEL PROGRAMS

EASY TO IMPLEMENT CONVENTIONAL ROAD INDICATORS

EASY TO IMPLEMENT CONVENTIONAL ROAD INDICATORS

Low   FEASIBILITY   High

PROPOSED SOLUTION :

It helped us to analyze and examine our solution more in the grounds of uniqueness, social impact, business model, scalability etc.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Smarts signs connectivity for better road safety to reduce the traffic and accidents by showing us the speed limit for certain areas and to minimize the travel time using the direction signs.
2.	Idea / Solution description	This application is very useful it shows us that how much speed we have to drive in the certain areas,and it will adoptively intimate the weather changes it show us direction using signs very helpful to prevent accidents.
3.	Novelty / Uniqueness	By knowing the certain speed that must be maintained in the certain areas we can prevent the accidents .
4.	Social Impact / Customer Satisfaction	To avoid the traffic and accidents and it shows the right route and directions,they experiecne a pieceful journey.
5.	Business Model (Revenue Model)	Reduction of Traffic and accidents and it will save them a lot of time by knowing the areas and by choosing the correct route.
.	Scalability of the Solution	It can be used for the long travels and in the certain areas we don't know anything about the areas.

REQUIREMENT ANALYSIS :

It briefs about functional and non-functional requirements. It involves the various steps in the entire process. It also specifies features usability, security, reliability, performance, availability and scalability.

FUNCTIONAL REQUIREMENTS :

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)

FR-1	User Requirements	Static signalboards will be replaced with smart linked sign boards that meet all criteria
FR-2	User Registration	Registration through webpage Registration through Gmail
FR-3	User Confirmation	Confirmation via Phone Confirmation via Email Confirmation via OTP
FR-4	Payment options	Cash on Delivery Net Banking/UPI Credit/Debit/ATM Card
FR-5	Product Delivery and Installations	Through Gmail ThroughMessages

NON-FUNCTIONAL REQUIREMENTS:

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Have a clear and self-explanatory manual. Easier to use It is user friendly and it can be used by anyone
NFR-2	**Security**	The network must contain cloud data, condensing it to be Avoid real-time avoidance, and keep an eye on the board at all times.
NFR-3	**Reliability**	Hardware requires a regular checking and service Software may be updated periodicallyImmediate alert is provided in case of any system failure
NFR-4	**Performance**	The application must have a good user interface It gives us the precise output ,it gives us the correct speed limitation in the correct weather time
NFR-5	**Availability**	All the features will be available when the user requires.It depends on the need of the customer and the customization the user has done.

PROJECT DESIGN :

DATA FLOW DIAGRAM :

A Data Flow Diagram (DFD) is a traditional visual representation of 11 October 2022 the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.
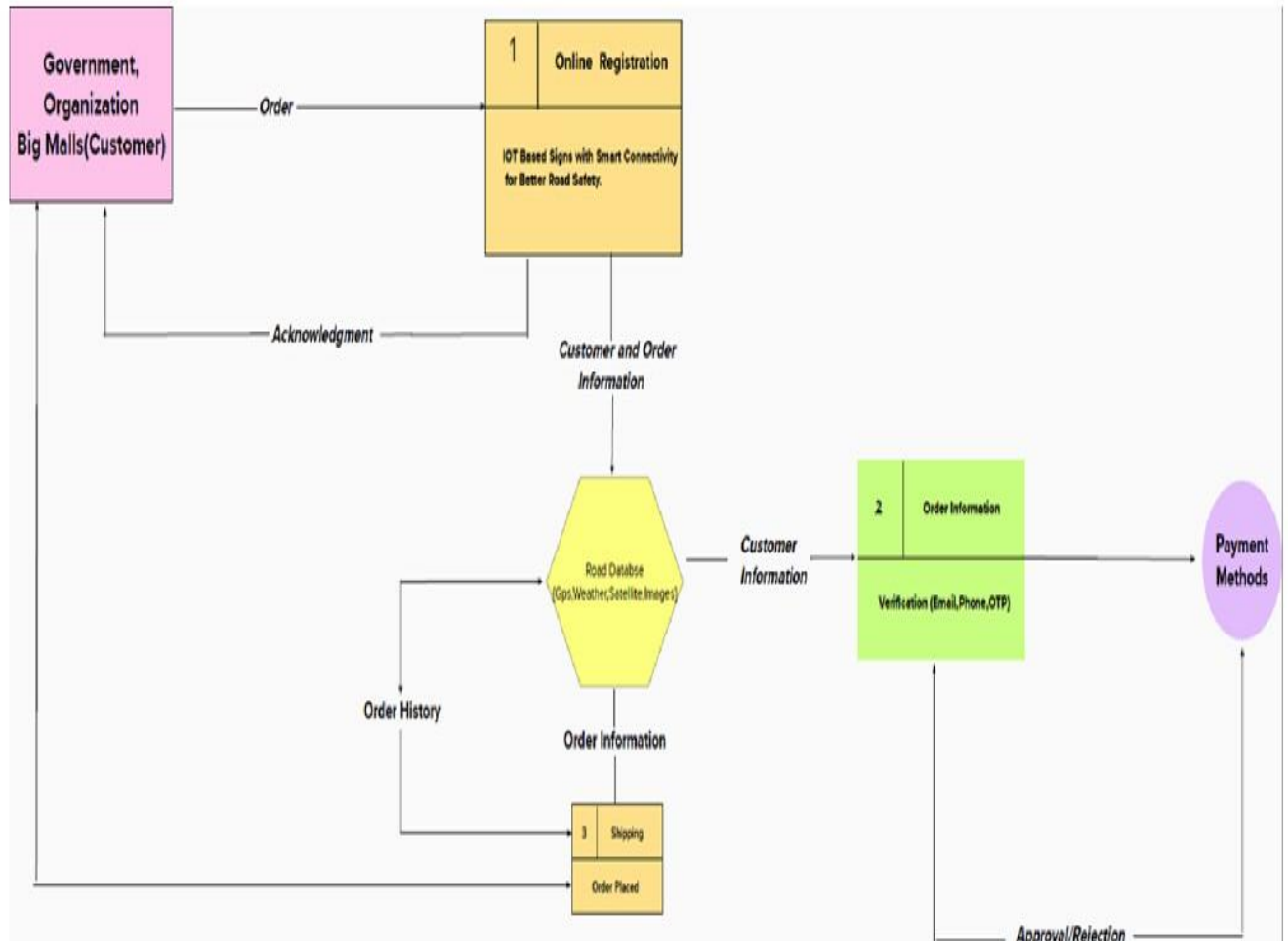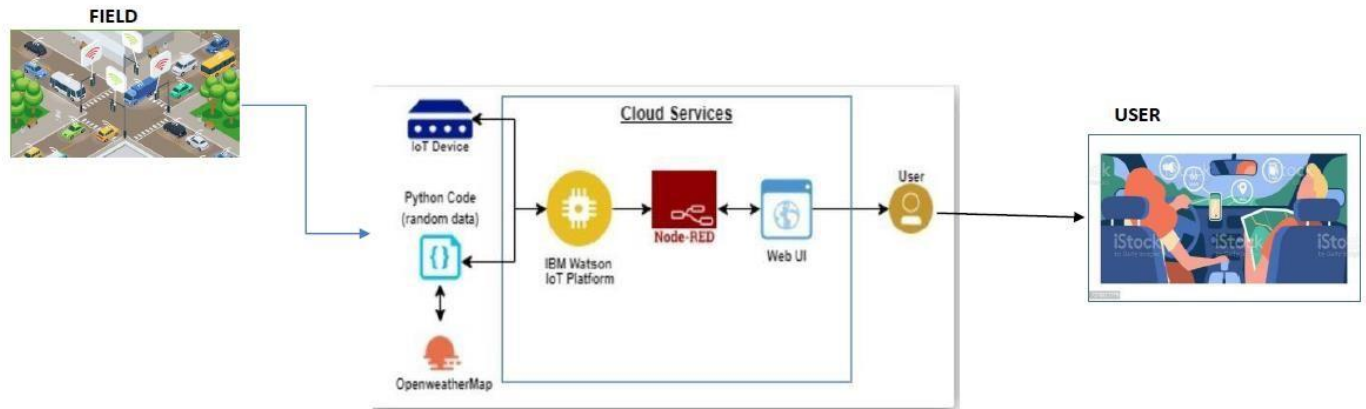

SOLUTION ARCHITECTURE :

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. It helped us understand the features and components used to complete the project.
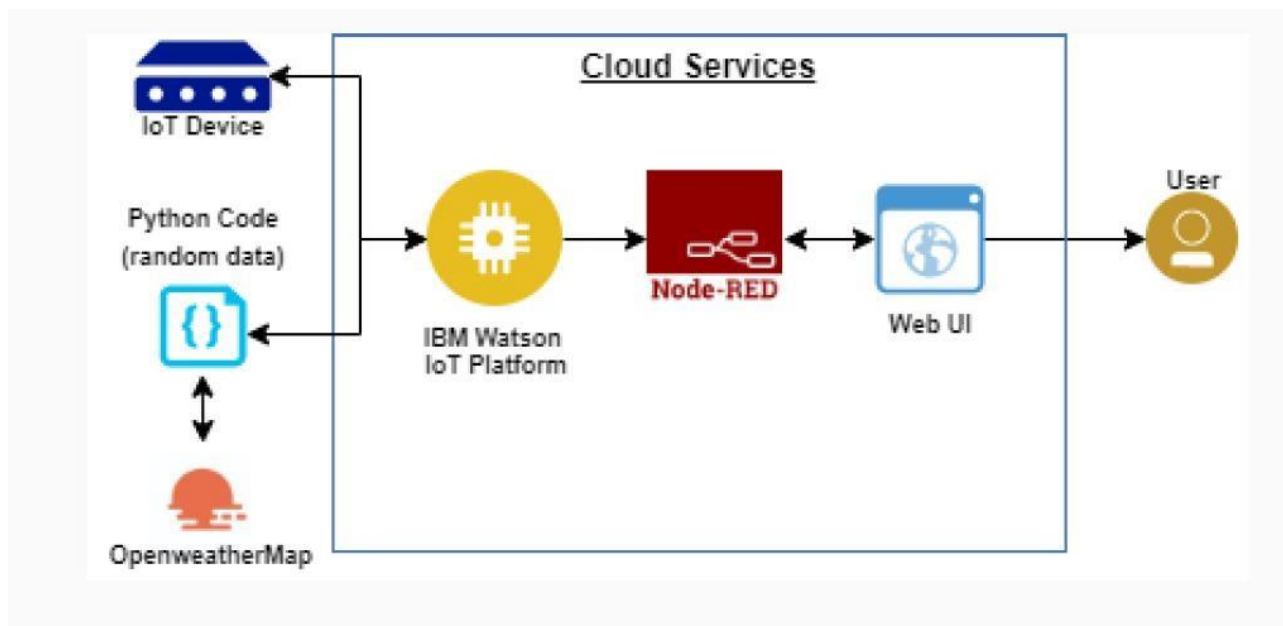
Technology Architecture:

Solution Architecture Diagram:

Architecture and data fow of the IOT Based Signs With Smart Connectivity For Better Road Safety.

Customer Journey Map:

Phases	Phase 1 *Motivation*	Phase 2 *Online Search and In person*	Phase 3 *Clicks the Specifications*	Phase 4 *Reads the review of the products*	Phase 5 *Contacting the dealer*	Phase 6 *Payment*	Phase 7 *Installation*	Phase 8 *Maintenance*
Activities Performed	Looking to purchase a gadget for a better road safety.	Searches for a cost-effective road safety gadget online and in person through dealers	Checks the specification for the gadget which would suit their requirement	Acquire the positive and negative feedback from people who already use it	Contacts the dealer through online or in person and makes arrangement for the purchase	Makes the payment for the suitable gadget.	The technician arrives at the customer's place for installing the road safety gadget in the vehicle which we have.	Reports the issue on the gadget to the company sends a technician to resolve this issue.
Emotions	Happy and little bit concerned.	Happy as a customer finds numerous options .	Disappointed as the first product didn't match his requirements	Very happy as he finds positive reviews on a suitable product.	Happy as he easily finds the contactdetails of the dealer online.	Happy and relieved as he was finally able buy the required product.	Excited to see how the detector works	Worried as the issue needs to be fixed .
Overall Experiences	Good	Good	Bad	Good	Good	Good	Good	Average as the customer is uncertain about the working condition.
Customer Expectations	Easy availability of quality product which is cost effective and useful.	Easy availability of quality product which is cost effective and durable.	Should be able to find his product which exactly satisfies his requirements .	A product with higher ratings, positive reviews and good Track record.	A dealer who is located in close proximity to the customer is preferred	We can pay even in cash or in online as safe and secure and quick and efficient delivery of the product.	We can also install this product but for some extra precautions and to install without any damages swell experienced technician will be a right choice.	A highly skilled service engineer who quickly and effectively identifies and resolves

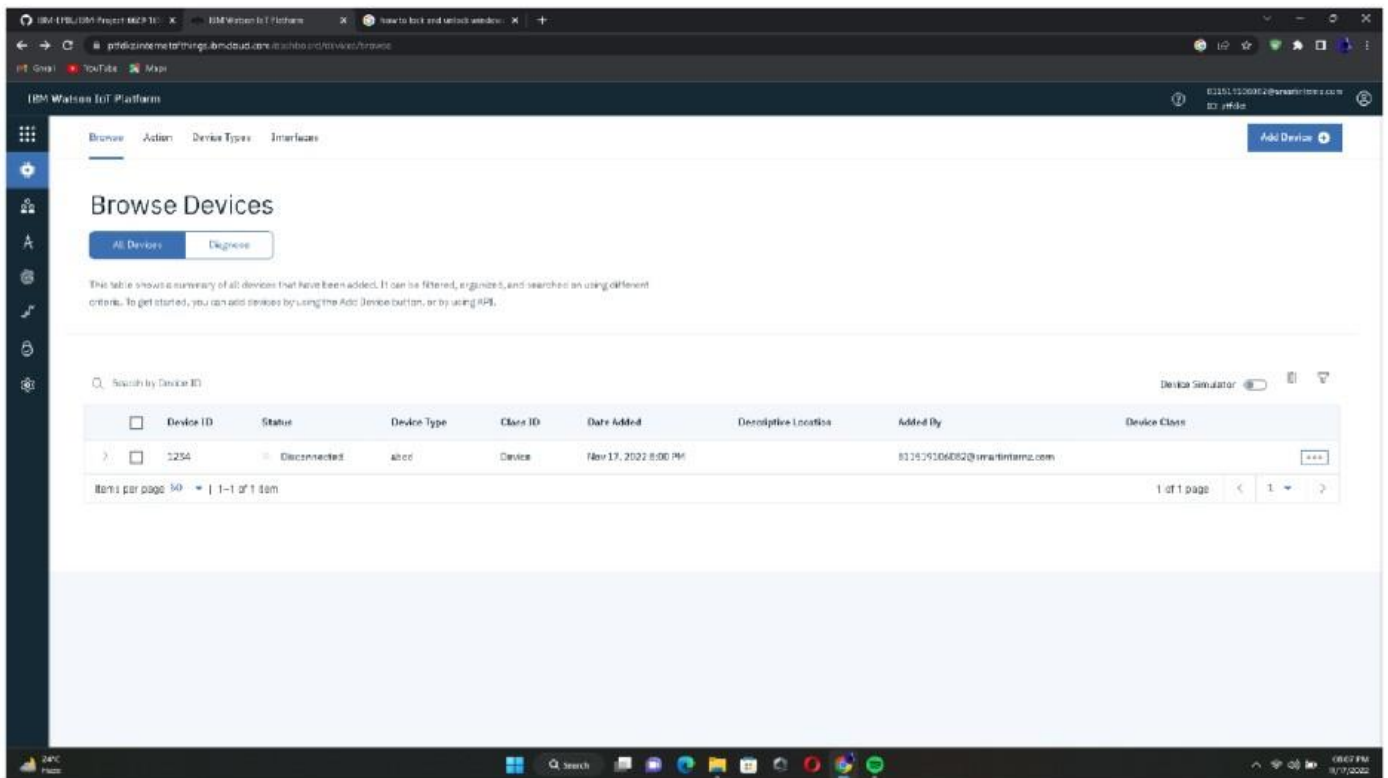IBM CLOUD SERVICES:
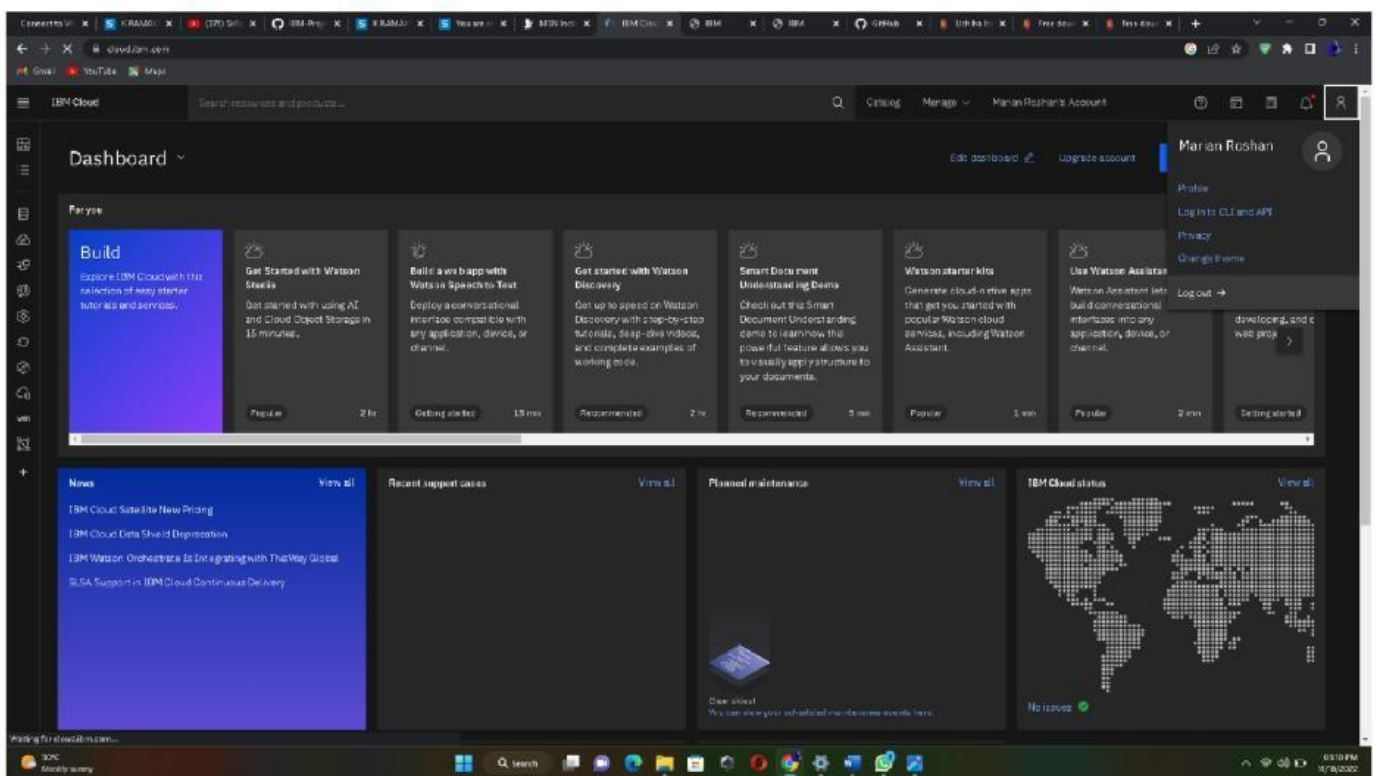
IBM WEATHER MAP SITE:
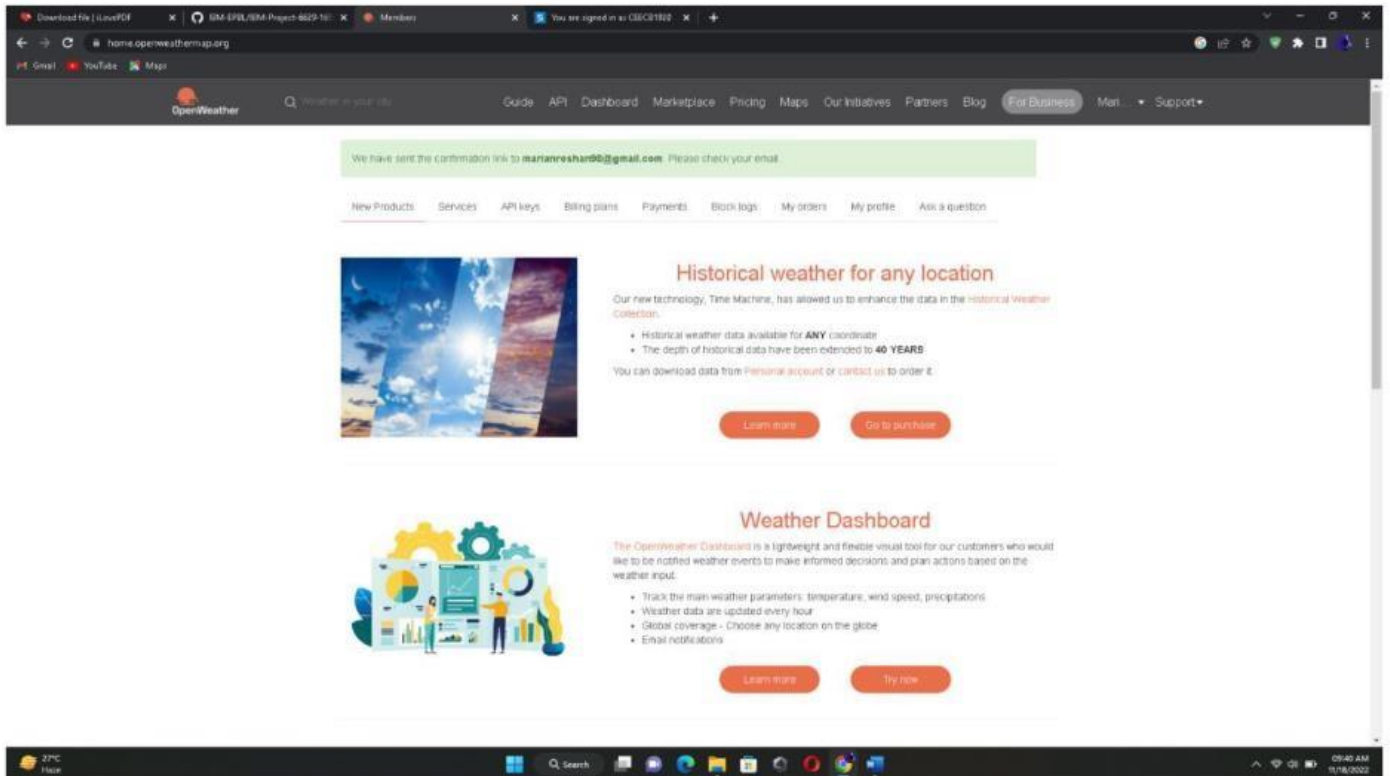
SOFTWARE:

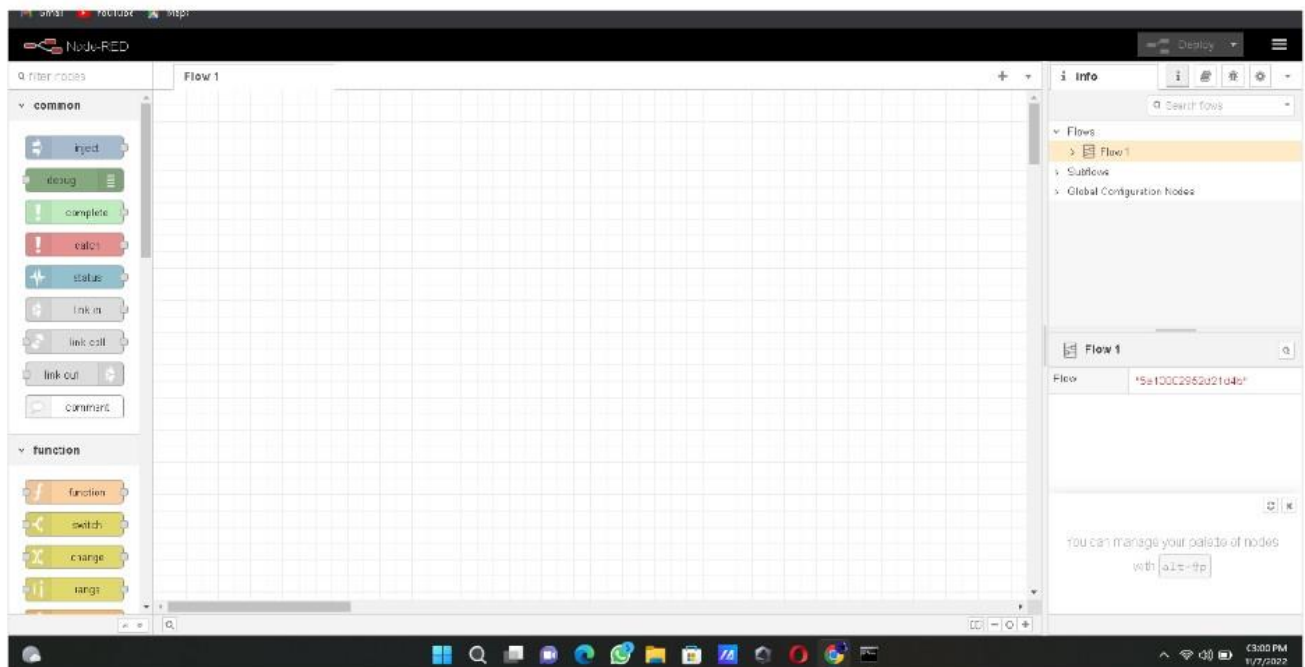| Date	18 October 2022
Team ID	PNT2022TMID11013
Project Name	Project – Signs with smart connectivity for Better road safety

Signs with smart connectivity for Better road safety

IBM WATSON IOT PLATFORM AND DEVICE :

NODE RED SERVICE:


PYTHON SCRIPTS :

DEVELOP THE PYTHON SCRIPT

PYTHON CODE :

```python
1   import wiotp.sdk.device
2   import time
3   from random import randint as ri
4
5   myConfig = {
6     "identity" : {
7        "orgId" : "epmoec",
8        "typeId" : "testDevice",
9        "deviceId" : "device0"10
        },
11    "auth" : {
12       "token" : "?-KDXUPMvDo_TK2&b1"13
        }
14 }
15
16 def myCommandCallback(cmd):
17     print("recieved cmd : ",cmd)
18
19
20 client = wiotp.sdk.device.DeviceClient(config=myConfig,logHandlers=None)
21 client.connect()
22
23 while True:
24     client.publishEvent(eventId="status",msgFormat="json",data={
25        "temperature" : ri(28,35),
26        "visibility" : ri(56,78),
27        "location" : "Chennai,IN"
28     },qos=0,onPublish=None)
29     client.commandCallback = myCommandCallback
30     time.sleep(1)
31
32 client.disconnect()
```

OUTPUT :
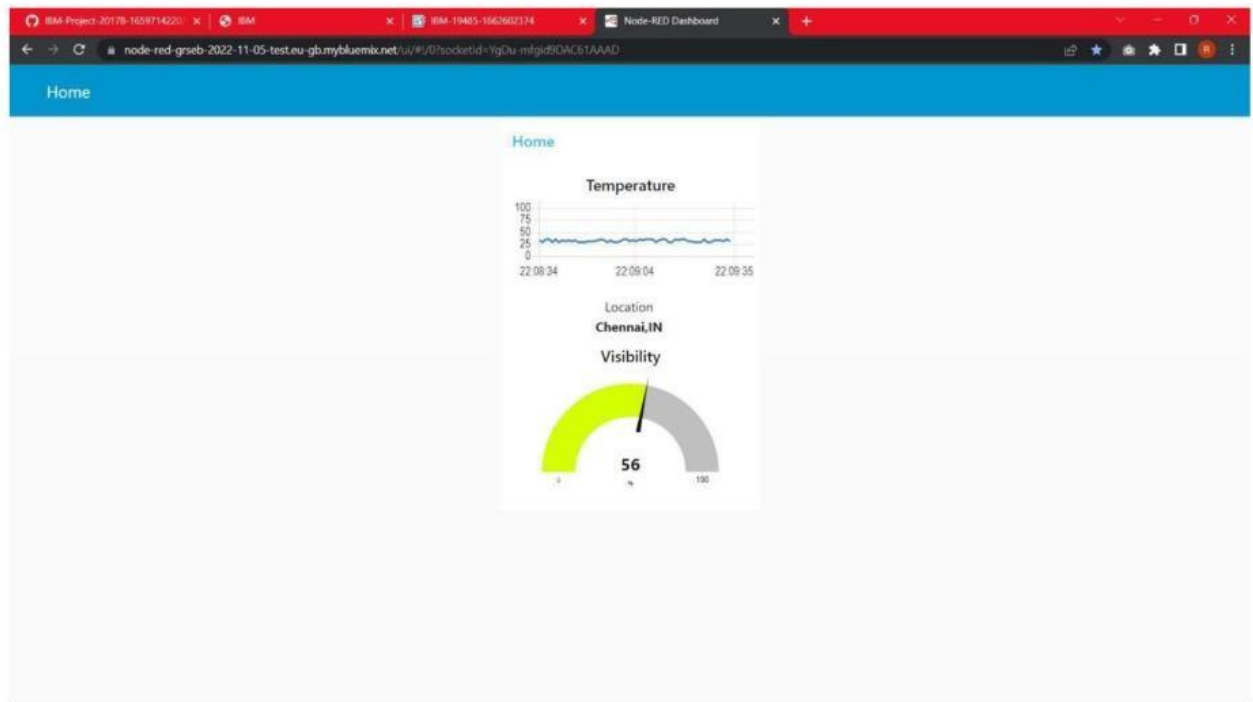
DEVELOP THE WEB APPLICATION USING NODE RED:
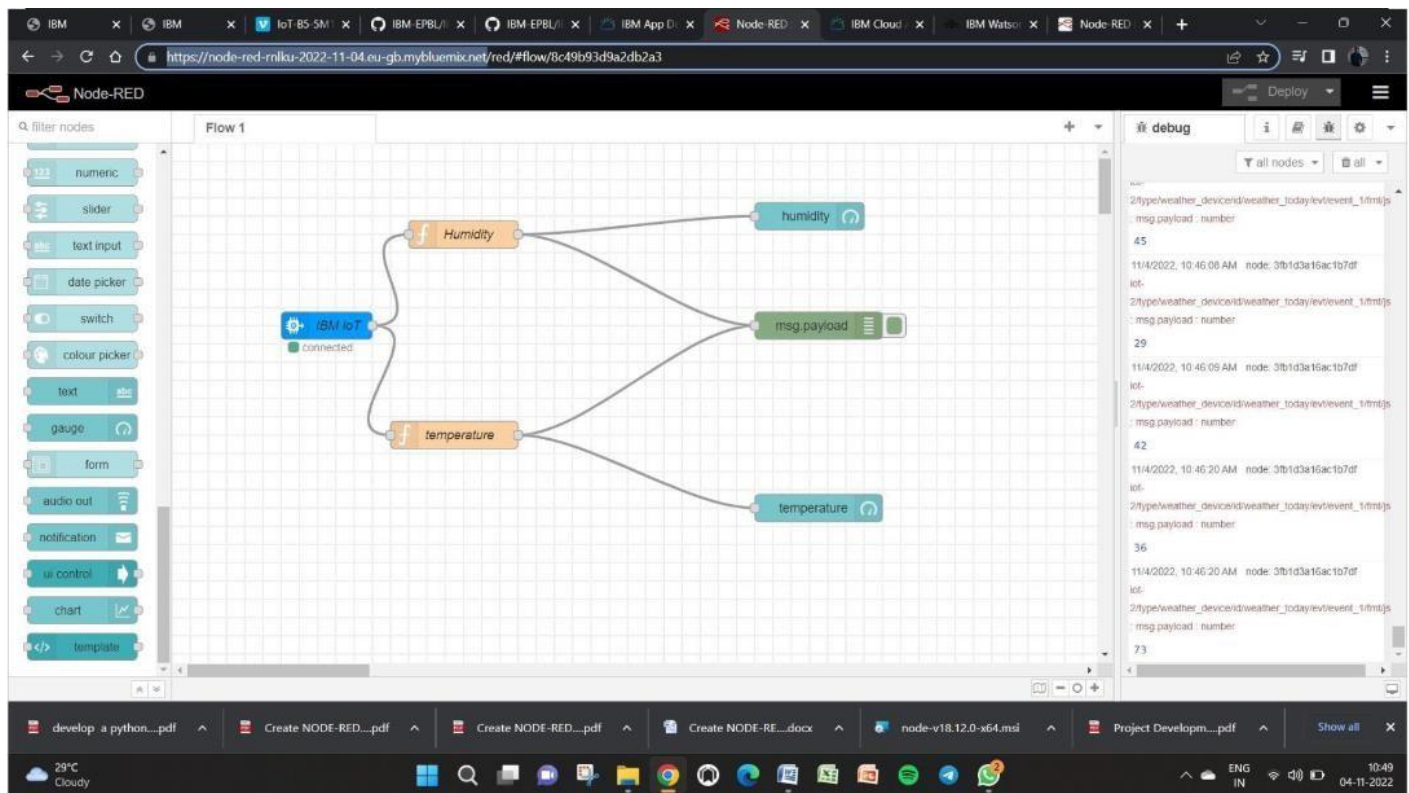
USE DASHBOARD NODES TO CREATE WEB UI:
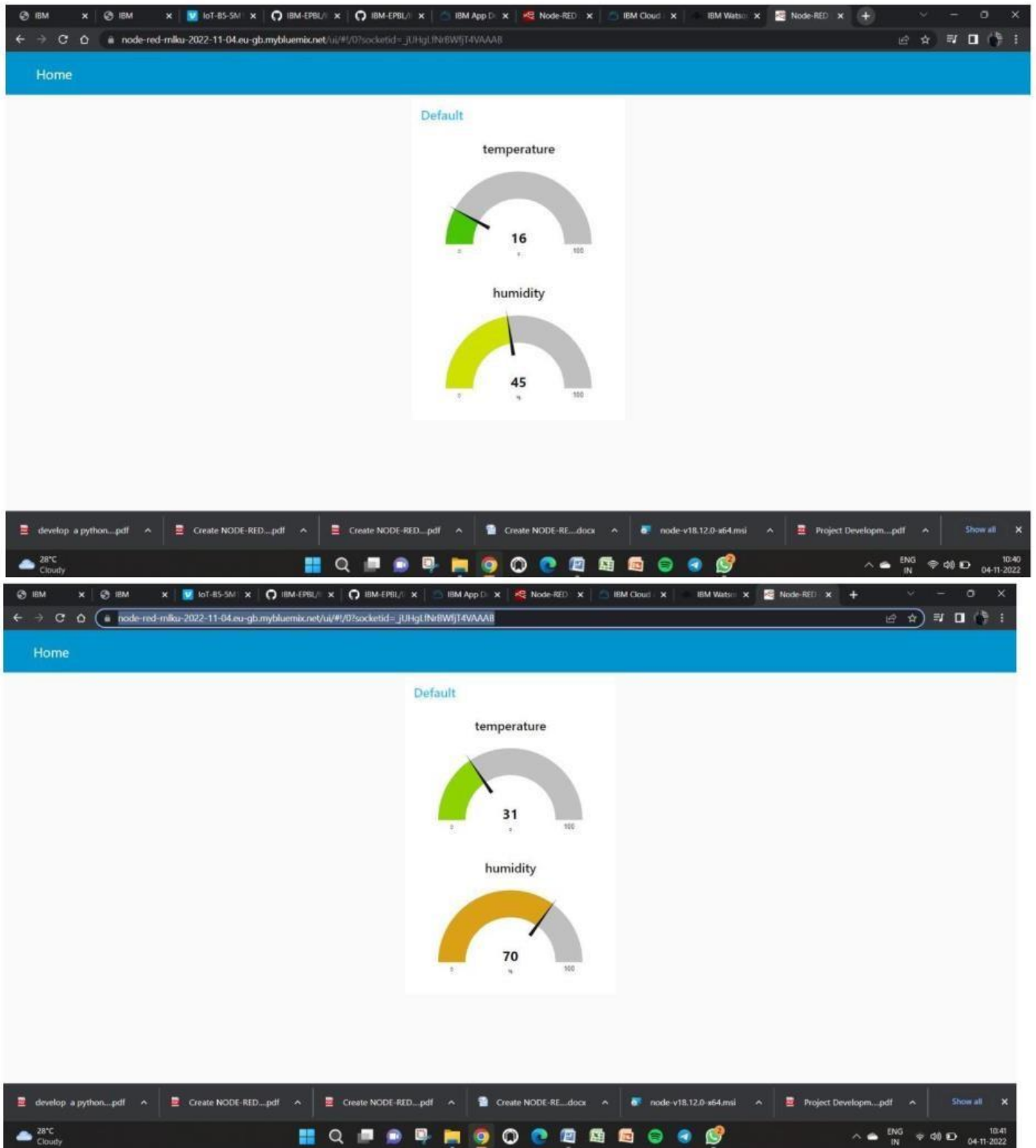
PROJECT DEVELOPMENT PHASE:

SPRINT 1:

PYTHON PROGRAM

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

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

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

[DEBUG ON]
[DEBUG OFF]
```

Sprint 1 ii)

```python
#python file
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)
```

Sprint 1 iii)
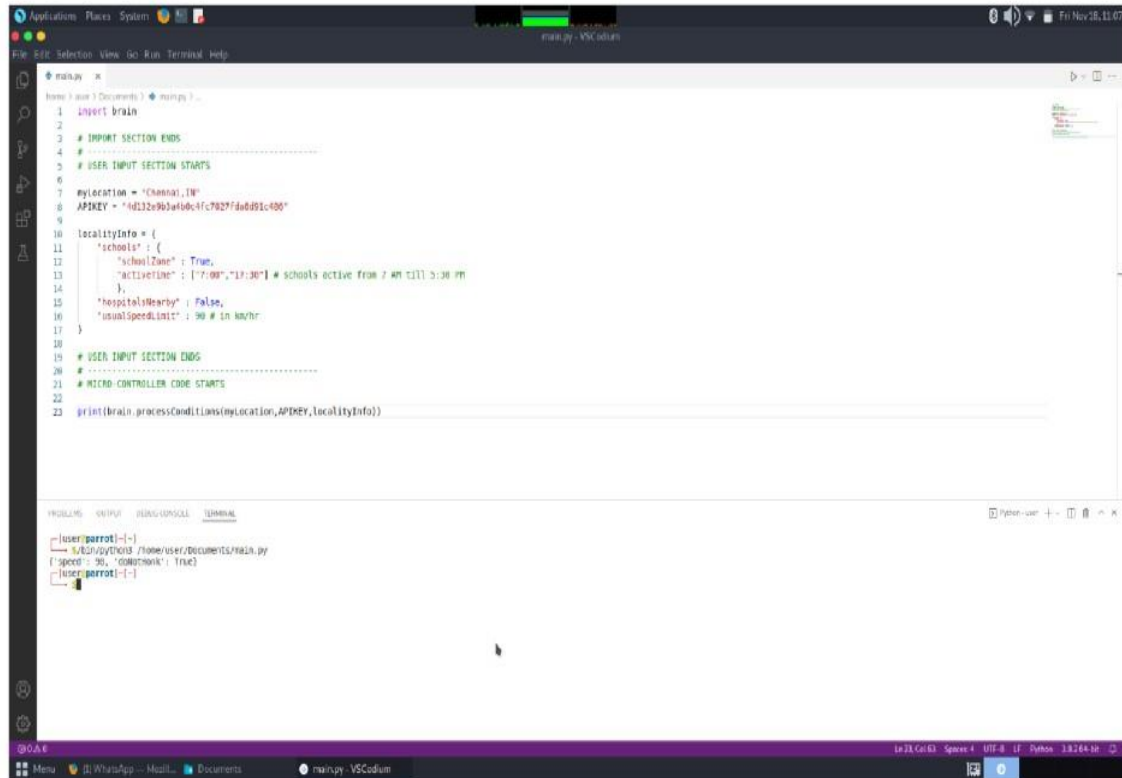
```python
import brain

myLocation = "Chennai,IN"
APIKEY = "c76d51c15c0e7c6c5f2002ad65efcec1"

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
}

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

OUTPUT SPRINT !:

SPRINT 2: PYTHON CODE:

```cpp
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include
<BlynkSimpleEsp8266.h>
 char
auth[]="q6FAQIggdIxznS2kMIbxAPn8E6nnv116";
char ssid[] = "hellow";
char pass[] = "12345678";
String str; void setup() {
Serial.begin(9600);
Blynk.begin(auth, ssid,
pass);
}
void loop() { Blynk.run(); if
(Serial.available()>0)   str   =
Serial.readStringUntil('/');   //
Serial.print(str);
// Blynk.notify("location:
");;Blynk.notify(str);
}
}
```

SPRINT 3 PYTHON CODE:

```
#include <TinyGPS++.h>
#include <SoftwareSerial.h>
TinyGPSPlus gps;
SoftwareSerial ss (3,4); char
n;
int a;

void setup() {
Serial.begin(9600);
ss.begin(9600); pinMode (2,
INPUT): pinMode (6,
OUTPUT); pinMode(11,
OUTPUT); pinMode(10,
OUTPUT);  pinMode (9,
OUTPUT); pinMode (12,
OUTPUT); //apr
digitalWrite(11,HIGH);
digitalWrite(6,HIGH);
attachInterrupt (digitalPinToInterrupt (2), piezo,CHANGE);
}
void loop() { n-
Serial.read(); //
Serial.println(" ");
delay (200);
if (n=='3') {
digitalWrite(6,HIGH);
digitalWrite(11,HIGH);
digitalWrite(12,HIGH);
delay(200);
digitalWrite(12,LOW); }
else if (n=='2')
digitalWrite(6,LOW);
digitalWrite(11,LOW);
```

```
delay(200);
digitalWrite(12,LOW); }
else if (n=='1')
analogWrite(11,100);
analogWrite(6,100);
digitalWrite(12,HIGH);
delay(200);
digitalWrite(12,LOW);
}
}
// while (ss.available() > 0)
// if (gps.encode(ss.read()))
// displayInfo(); void
displayInfo()
{
 // Serial.print (F("Location: ")); if
(gps.location.isValid())
Serial.print(gps.location.lat(), 6);
Serial.print (F(","));
Serial.print(gps.location. lng(), 6); } else
// Serial.print (F ("INVALID"));
Serial.print("10.305125");
Serial.print(',');
Serial.print("76.389582");
}
/* Serial.print(F(" Date/Time: "));
if (gps.date.isValid())
{
Serial.print(gps.date.month());
Serial.print (F("/"));
Serial.print(gps.date.day());
Serial.print (F("/"));
```

```
Serial.print(gps.date.year());
}
else
{
Serial.print(F("INVALID"));
}
Serial.print (F("  "));
if (gps.time.isValid())
{
if (gps.time.hour() < 10) Serial.print (F("0"));
Serial.print(gps.time.hour()); Serial.print
(F(":"));
if (gps.time.minute() < 10) Serial.print(F("0"));
Serial.print (gps.time.minute()); Serial.print
(F(":"));
if (gps.time.second() < 10) Serial.print(F("0"));
Serial.print(gps.time.second()); Serial.print
(F("."));
 if (gps.time.centisecond() < 10) Serial.print(F("0"));
Serial.print(gps.time.centisecond());
}


else
{
// Serial.print (F("INVALID"));
}*/
Serial.println();
}
```

SPRINT 4 PYHTON CODE:

```
#include <TinyGPS++.h>
#include <SoftwareSerial.h>
TinyGPSPlus gps;
SoftwareSerial ss (3,4); char
n;
int a;

void setup() {
Serial.begin(9600);
ss.begin(9600); pinMode (2,
INPUT): pinMode (6,
OUTPUT); pinMode(11,
OUTPUT); pinMode(10,
OUTPUT);  pinMode (9,
OUTPUT); pinMode (12,
OUTPUT); //apr
digitalWrite(11,HIGH);
digitalWrite(6,HIGH);
attachInterrupt (digitalPinToInterrupt (2), piezo,CHANGE);
}
void loop() { n-
Serial.read(); //
Serial.println(" ");
delay (200);
if (n=='3') {
digitalWrite(6,HIGH);
digitalWrite(11,HIGH);
digitalWrite(12,HIGH);
delay(200);
digitalWrite(12,LOW); }
else if (n=='2')
digitalWrite(6,LOW);
digitalWrite(11 LOW):
```

```
delay(200);
digitalWrite(12,LOW); }
else if (n=='1')
analogWrite(11,100);
analogWrite(6,100);
digitalWrite(12,HIGH);
delay(200);
digitalWrite(12,LOW);
}
}
// while (ss.available() > 0)
// if (gps.encode(ss.read()))
// displayInfo(); void
displayInfo()
{
 // Serial.print (F("Location: ")); if
(gps.location.isValid())
Serial.print(gps.location.lat(), 6);
Serial.print (F(","));
Serial.print(gps.location. Ing(), 6); } else
// Serial.print (F ("INVALID"));
Serial.print("10.305125");
Serial.print(',');
Serial.print("76.389582");
}
/* Serial.print(F(" Date/Time: "));
if (gps.date.isValid())
{
Serial.print(gps.date.month());
Serial.print (F("/"));
Serial.print(gps.date.day());
Serial.print (F("/"));
Serial.print(gps.date.year());
```

```
else
{
Serial.print(F("INVALID"));
}
Serial.print (F(" "));
if (gps.time.isValid())
{
if (gps.time.hour() < 10) Serial.print (F("0"));
Serial.print(gps.time.hour()); Serial.print
(F(":"));
if (gps.time.minute() < 10) Serial.print(F("0"));
Serial.print (gps.time.minute()); Serial.print
(F(":"));
if (gps.time.second() < 10) Serial.print(F("0"));
Serial.print(gps.time.second()); Serial.print
(F("."));
 if (gps.time.centisecond() < 10) Serial.print(F("0"));
Serial.print(gps.time.centisecond());
}

else
{
// Serial.print (F("INVALID"));
}*/
Serial.println();
}
void piezo()
{
while (ss.available() > 0) if
(gps.encode(ss.read()))
displayInfo();
}
int a=0,b=0,c=0,d=0;
void setup() { pinMode
(D1, INPUT); pinMode
(D2, INPUT); pinMode
(D3, INPUT); pinMode
(D4, INPUT);
digitalWrite(D1,LOW);
digitalWrite(D2, LOW);
digitalWrite(D3, LOW);
digitalWrite(D4, LOW);
Serial.begin(9600);
}
void loop()
{
a=digitalRead(D1);

a=digitalRead(D1);
if (a==1) {
Serial.print("1"); }
b=digitalRead (D2);
if (b==1) {
Serial.print("2"); }
d=digitalRead(D4);
if (d==1)
{
Serial.print("3");
}
}
}
```

OUTPUT:

```python
import wiotp.sdk.device
import time
import random
import ibmiotf.application
import ibmiotf.device
import requests, json
myConfig = {
#Configuration
"identity": {
"orgId": "gi5mpx",
"typeId": "NodeMCU",
"deviceId":"12345670"
},
#API Key
"auth": {
"token": "1234567890"
}
}
#Receiving callbacks from IBM IOT platform
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()
#OpenWeatherMap Credentials
    BASE_URL = "https://api.openweathermap.org/data/2.5/weather?"
    CITY = "TRICHY, IN"
    URL = BASE_URL + "q=" + CITY + "&units=metric"+"&appid=" + "4d132e9b3a"
    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']
#message part
    msg=random.randint(0,5)
    if msg==1:
        message="GO SLOW, SCHOOL ZONE AHEAD"
    elif msg==2:
        message="NEED HELP, POLICE STATION AHEAD"
    elif msg==3:
```

```
Python 3.10.4 (tags/v3.10.4:9d38120, Mar 23 2022, 23:13:41) [MSC v.1929 64
bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
=============================== RESTART: D:\main.py =============================
=====
22,EMERGENCY, HOSPITAL NEARBY,Speed Breaker,Moderate,Fog Ahead, Drive Slow
```

Ln: 16 Col: 0   Ln: 6 Col: 0