TEAM ID –PNT2022TMID45631 Project Report

1. INTRODUCTION

1.1 Project Overview

The project implemented using IOT that predict traffic , rainfall ,ambulance dectection ,human present in road , any alert sound . Using arduino UNO interface with sensors, buzzer, LED etc. Sensor such as Ultrasonic sensor , Temperature sensor , PIR sensor. The user will see the display on SMART SIGNS on the road . The device we designed UI (WEB based Application) given to the traffic controller. They display on SIGNS. The main advantages of the project is prevent the traffic ,give path to ambulance , in rainy days give instruction to the driver about weather using open weather app they go fast or slow.

1.2 Purpose

The purpose of the project is make easy road travel with IOT devices. It also save lives. In this UI we have information of temperature , sound ,human dectection . It makes travel easer.

2. LITERATURE SURVEY

2.1 Existing Problem

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.

The early effects to prevent road accidents and to ensure road safety includes the use of speed detection devices,CCTVs,speed limiters and emergency accident units as the first phase.Despite achieving the state-of-the-art performance, the existing systems suffer from two main problems.

- Over Speed : These systems cannot control speed at some specific zones.

- Exact location of accident occured: These systems cannot give the precise location of accident .

2.2 References

Authors	Type	Availability	Research	Findings
Houser ,Pierowicz ,& Fuglewicz (2005)	FMCSA report	Public	Areporttoprovideabetterunderstandingofthefunctionofon-boardsafetysystemsandprovideinsightintothesafetyandefficiencybenefitsofusingsuchsystems.	Describestheconceptofoperationsandthevoluntaryrequirementsfortheuseof VSS forlargetrucksgreaterthan10,000 pounds GVWR.
Berg ,Niewohner, Burkle,& Morschheuser (2001)	Journal article	Public	Aninvestigationof109reallifetruckcrashesandacras htestinvolvingaMercedes-BenzActros.	Safetybeltsinheavytruckshaveapotentialtosavedriversandpassengers.Ejected truck occupant shave the greatest probability of beingk ilEdina crash.
Trevorrow &Eady(2010)	Australian roads report	Public	Areporttoimproveknowledgeandunderstanding of heavyvehiclebrakesafetyonlongsteepandveryste eproads. literature review ,review of crash data ,and a vehicle test	Advancedbrakingsystemsofferincreasedsafetyinanemergencyonsteeproadsduetotheautomaticapplicationoftheservicebrakespreventingroll-overorrun-off-roadcrashes. Whilebrakefailurecrashesaccountedforlessthanonequarteroffatal truck crashes ,break failure crashes were found to be mores curious. Fatal break failure crashes were more likely on horizontal curves ,how ever brake failure crashes on acombination of horizontal curveandverticalgradeweremoreseriousthanthoseoccurringonverticalgradealone. Themainsafetyissuehighlightedwasthedrivers'interactionwiththeauxiliarybraking system.Inadequateownersmanualinformationandalackofreal-timedriverfeedbackregardingtheperformance(orlackthereofbrakeswereidenti
Lambert & Rechnitzer (2002)	MUARC report	Public	A review and report of the Issue of rear and side under run crashes.	Twomajoreffectsofunderrunontheoutcomesofcrasheswereidentified:underrun canexposelightvehicleoccupantstotherigidstructuresofthetruckbeforethesafetyfeaturesofthelightvehiclecomeintoeffect;anddamagetoheavyvehiclecompo

				nents(e.g.,steering,braking,etc.) can reduce the controllability of the truck during or after the crash. Thereislittleevidencesuggestingthatimprovementsintruckunderrunprotectioncan notbeachieved. Thereissomeevidencethatenforcementofunderrunrequirementsandstandardsislacking. Performanceoffrontbarriersmusthave asignificantlyhigherstandard,atleast twice that of rear under run barriers. The requirements of barriers should extend to vehicles of 3.0tonnesGVM.
Hart(2010)	Conference paper	Public	Describesthedevelopmentof theAustralianbrakebalancecodeofpracticetoguidetheintermixingofbraketechnologiesonheavyvehiclecombinationvehicles.	Awiderangeofbrakingtechnologiescannowbeintermixedoncombinationvehicles,e.g.,advancedelectroniccontrolsarebeingconnectedtobasicvehicles.Therecommendedperformancelevelsetoutbythecodeisthatacombinationvehiclebeabletoachieve aninstantaneousdecelerationlevelonasealed60km/h

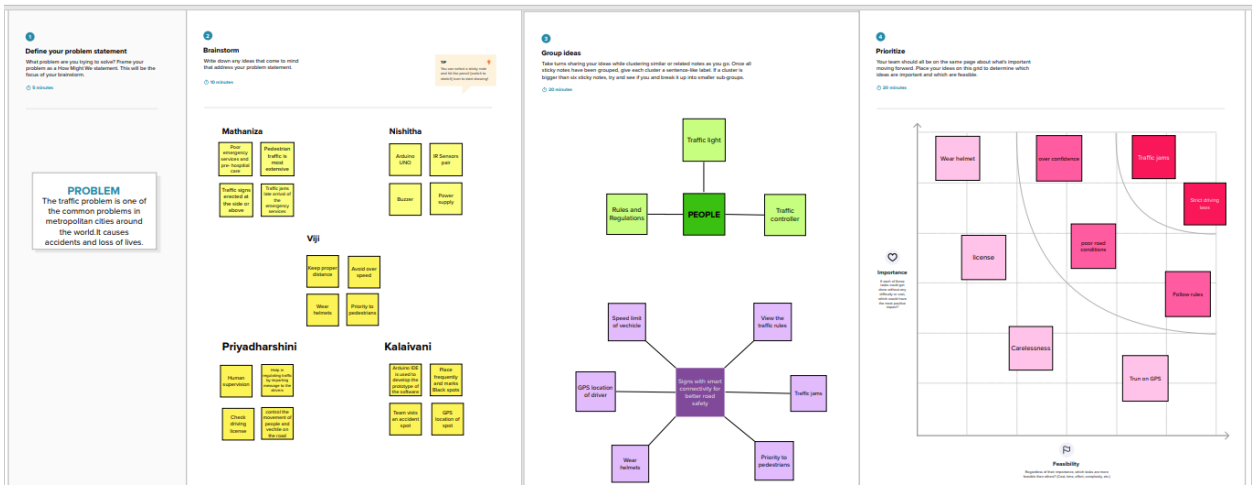
2.3 Problem Statement Definition

The traffic problem is one or the common problems in metropolitan cities around theworld. It causes accidents and loss of lives .We cannot control the occurrences of accidents but taking precautions to avoid life threatening injuries due to road accident is in our hands-by wearing helmet.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

3.2 Ideation and Brain storming

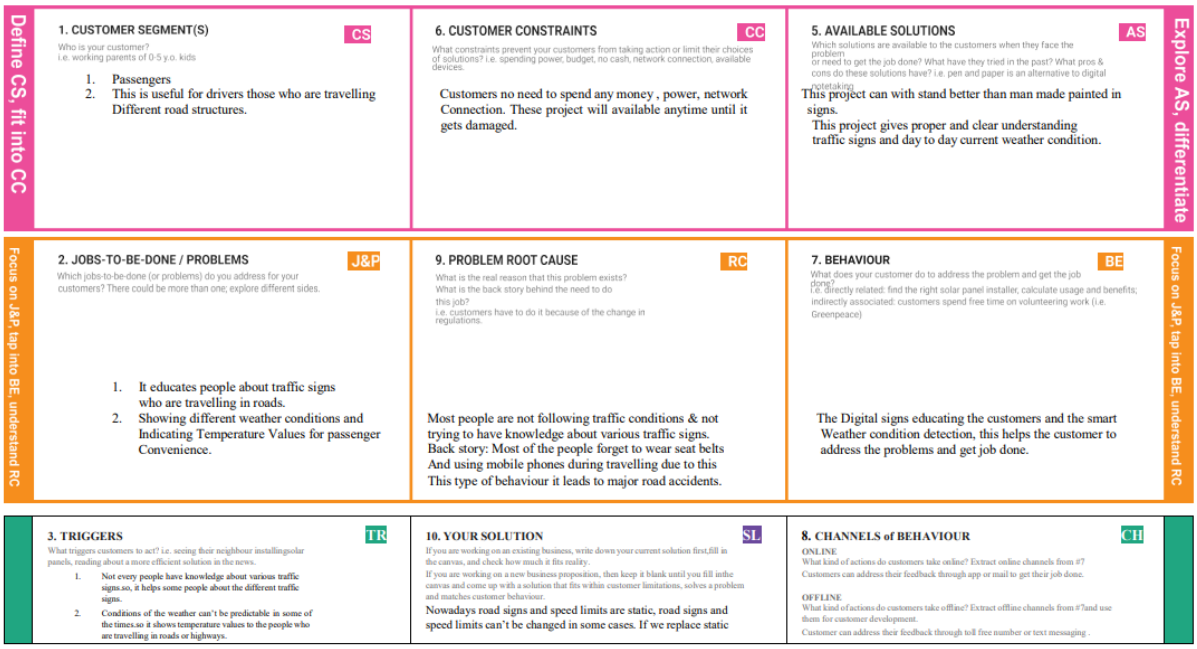


3.3 Proposed Solution

PROPOSED SOLUTION

S.NO.	PARAMETER	DESCRIPTION
1.	**Problem Statement**	The traffic problem is one or the common problems in metropolitan cities around the world. It causes accidents and loss of lives.
2.	**Solution**	Traffic management Rules and regulations must be impressed upon everyone from drivers and passengers to pedestrians that safety comes firsts. Including road safety issue in textbooks. Finding alternative to road transportation and emphasizing on waterways and rail communication.
3.	**Uniqueness**	An algorithm is given to predict the traffic solidity for future to minimize the traffic congestion. Development of IOT based traffic management system. Identify and penalize traffic violators and help officials identify unauthorized drivers. Reroute the ambulance to the low congestion roads to help get medical care at the earliest.
4.	**Social Impact**	It helps the driver free from traffic jam also save lot of time. Minimize road accident by regulating the traffic also get free flow of traffic without unnecessary interruption and congestion. Promotes driver confidence.
5.	**Business Model**	Drivers are under pressure to reach the destination in correct time due to traffic jams. To overcome this pressure, they can make use of predictive models which help them to ease the smart signs. Drivers free from congestion.
6.	**Scalability**	Further to reduce the immense pressure faced by the drivers to travel on road, the model can also helped driver travel safely.

3.4 Problem Solution Fit



4. REQUIREMENT ANALYSIS

4.1 Functional Requirement

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR No.1	Drivers and number of passenger on the vehicle	Number of passenger on the vehicle are noted
FR No.2	Predicting vehicle speed using sensors	IR sensor, Proximity sensor etc
FR.No.3	Pre-processing the speed of vehicle	Determination of Moving Vehicle speed using Image Processing
FR.No.4	Classification of sensor	A few examples of analog sensors are: accelerometers, pressure sensors, light, and sound sensors. Digital Sensors (also known as electronic or electrochemical sensors) convert

		the data transmission, digitally. Examples include digital accelerometers, pressure, and temperature sensors
FR.No.5	Building and training the system	The proposed system uses a set of ultrasonic sensors and has two modules: one for vehicle monitoring and other for priority management.
FR.No.6	Testing the model	In this phase, we tested the accuracy of the models with the test dataset that was formed in previous phase and the most accurate model is figured out.

4.2 Functional Requirement

Non-functional Requirements:

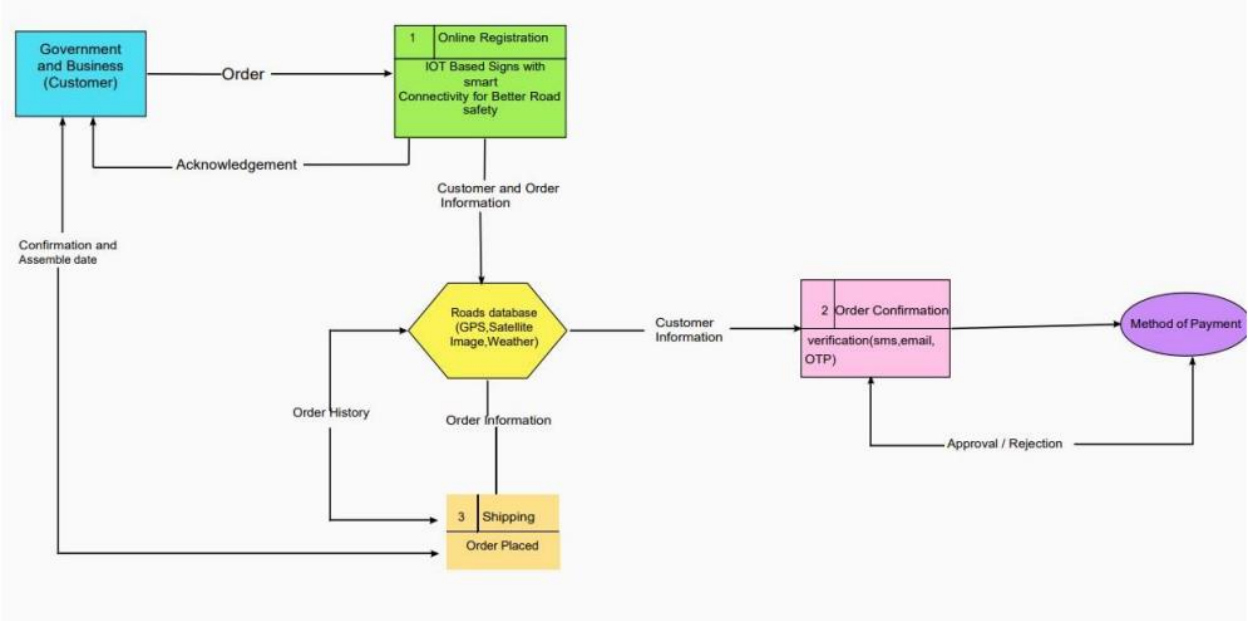
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	It is simpler and possible to predict speed of the vehicle, accident at an earlier stage. Because that it benefits all kinds of people, it is a life saving option.
NFR-2	**Security**	Predict speed of the vehicle, accident helps to saves the life.
NFR-3	**Reliability**	This approach offers excellent performance and scalability, making it more dependable.
NFR-4	**Performance**	It provides accuracy of over 90%. Thus, it has a high performance rate.
NFR-5	**Availability**	By having few basic data set of people we can predict the accident ,speed of the vehicle
NFR-6	**Scalability**	It has more efficiency in detecting speed of the vehicle ,accident than any other models.

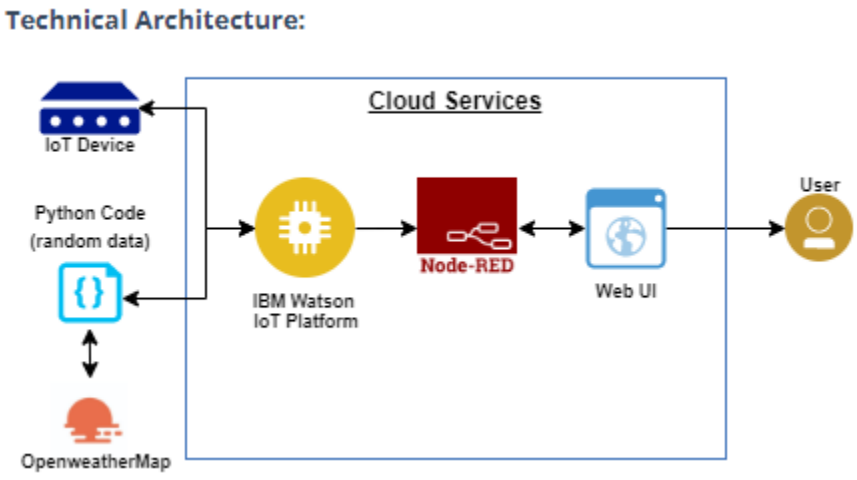
5. PROJECT DESIGN

5.1 Data Flow Diagram

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



5.2 Solution and Technical Architecture



5.3 User Stories

User Stories

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and conforming my password.	Access my account / dashboard	High	Sprint-1
Weather	Open weather map	USN-2	As a user, I want to check the weather of that location	Get the weather of that location	High	Sprint-1
Iot devices	Automation	USN-3	As a user, I want to use Iot devices for automation purposes	Get the work done without manual effort	High	Sprint-2
Python code	Random data	USN-4	As a user, I want to give some input to the devices for performing some action to complete the tasks very easily	Get the data Work flow	Medium	Sprint-1
IBM Cloud	Cloud services	USN-5	As a user, I want to deploy these application for public version	Useful for all domain users	High	Sprint-1
Node-Red	Integration	USN-6	As a user, I want to integrate the applications withhardware	iot precise for linear work flow	Medium	Sprint-3
Web UI	Interaction	USN-7	As a user, I want to interact with the digital products	IOT interact with the users	Medium	Sprint-2

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning and Estimation

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Resource	Create a account in Tinkercad ,Open weather map app, MIT etc	1	Low	R.Mathaniza M.Viji K.Nishitha
Sprint-2	Software	Python IDLE	1	Medium	B.Priyadharshini R.Kalavani
Sprint-3	Interface	Interface wokwi with IBM Iot Watson. Interface MIT with IBM Iot Watson	2	High	R.Mathaniza M.Viji
Sprint-3	Interface	Interface Node-red with cloud	2	Medium	K.Nishitha M.Viji
Sprint-4	Hardware	Integrate the hardware components to the IBM cloud(IOT Watson)	2	High	M.Viji R.Mathaniza
Sprint-4	Road Safety	User has better road safety	1	Low	R.Mathaniza

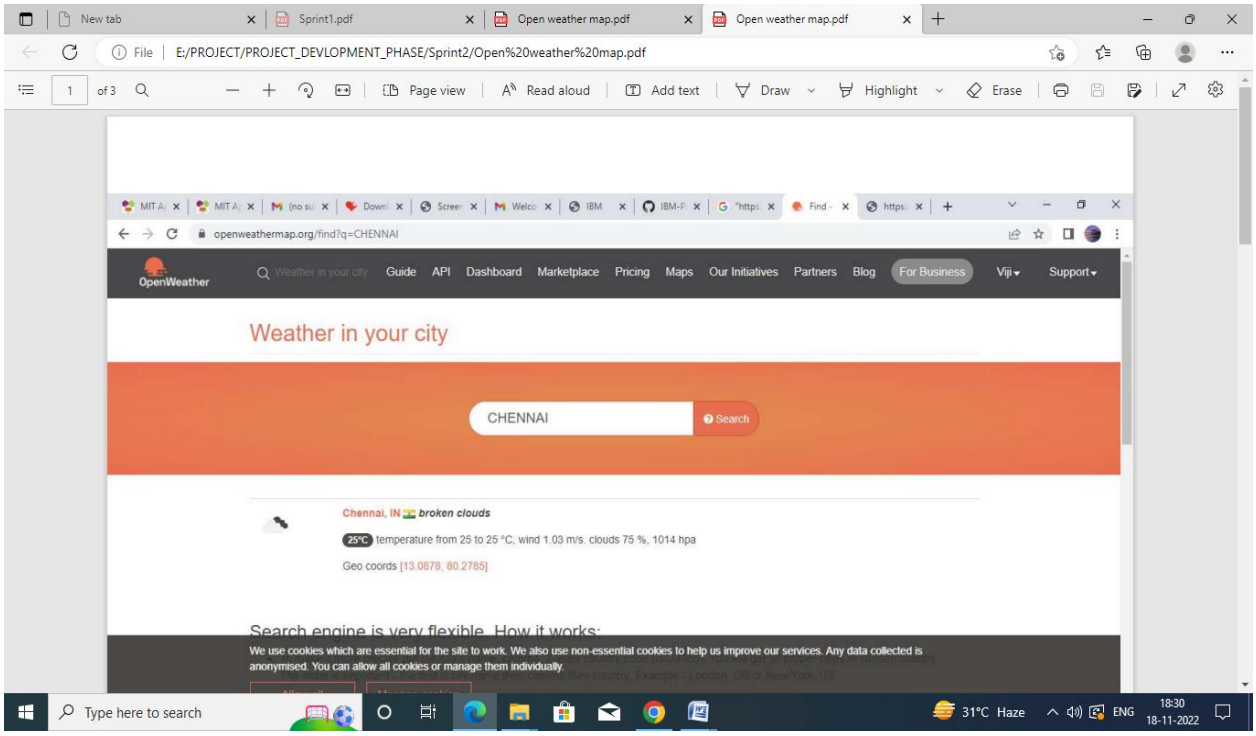
6.2 Sprint Delivery Schedule

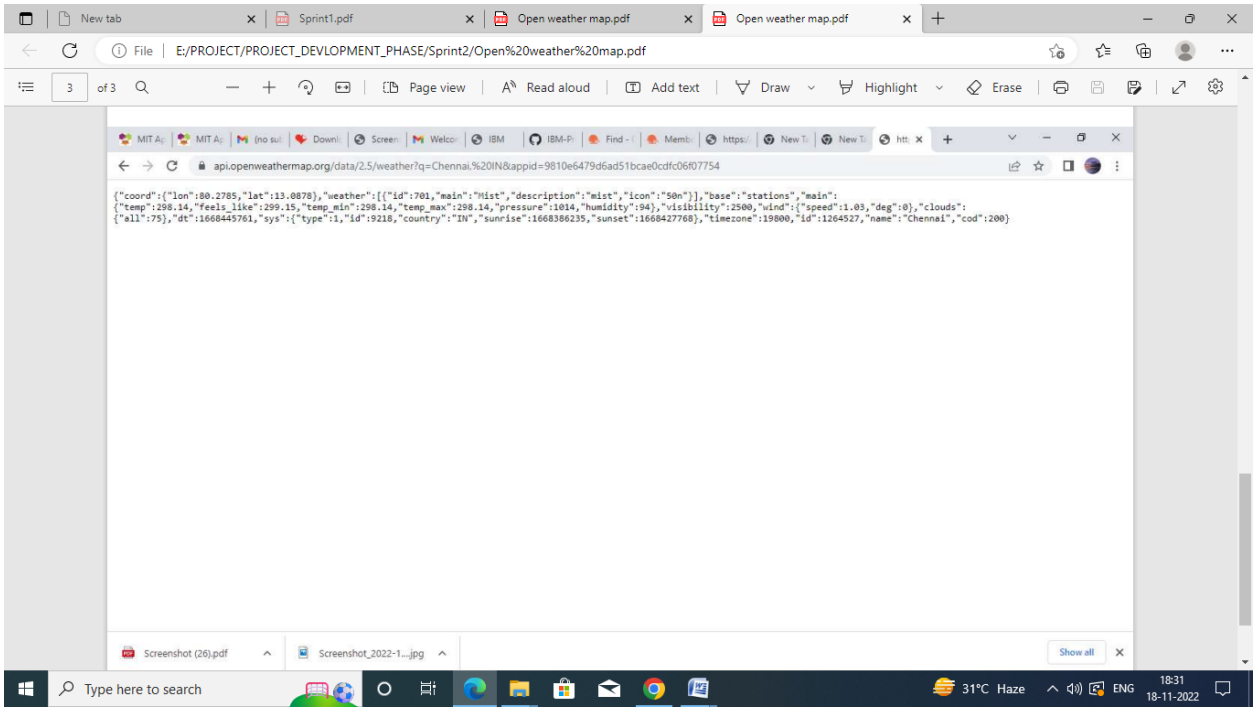
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	5 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

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

7.1 Feature1

OPEN WEATHER MAP



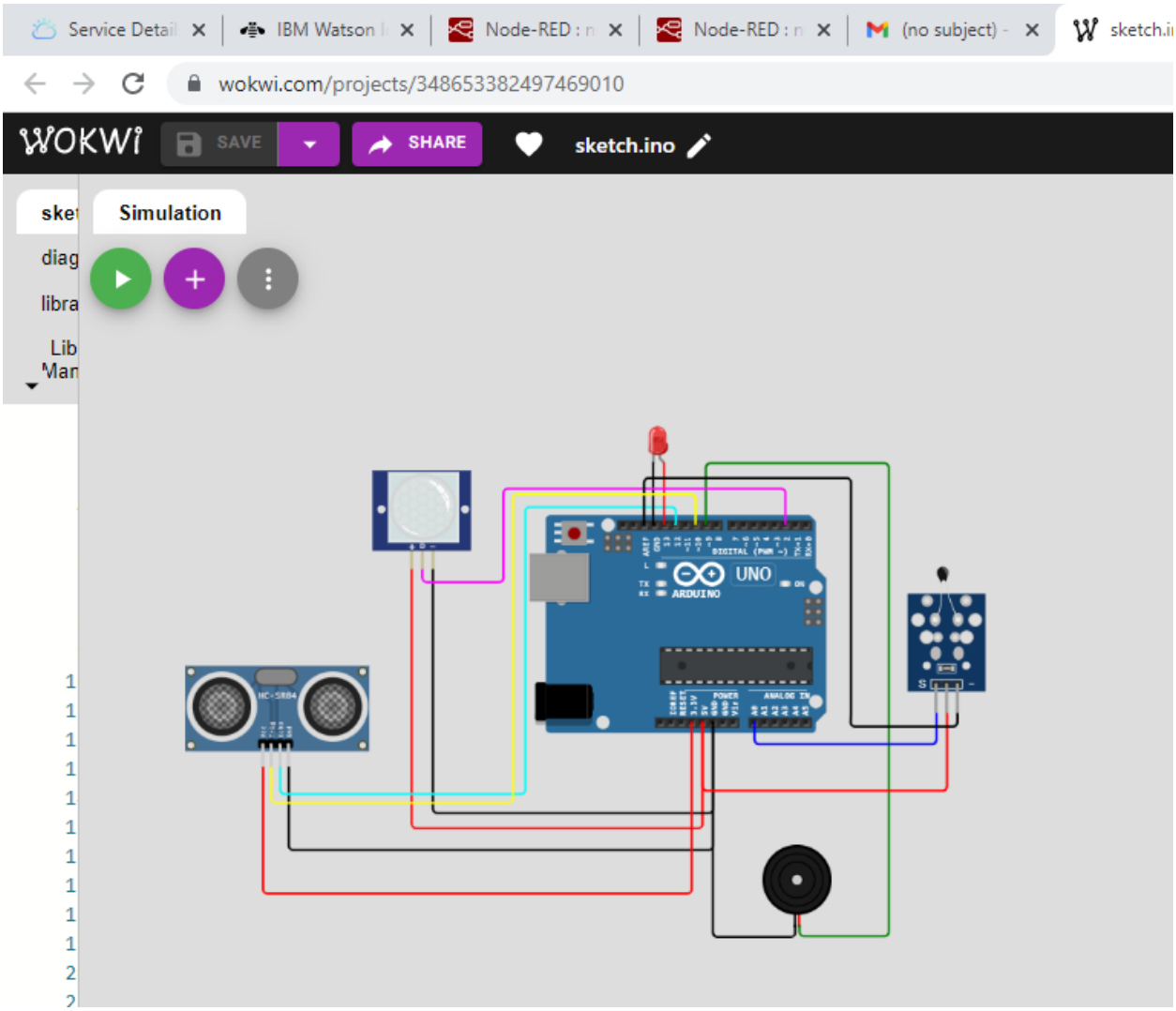


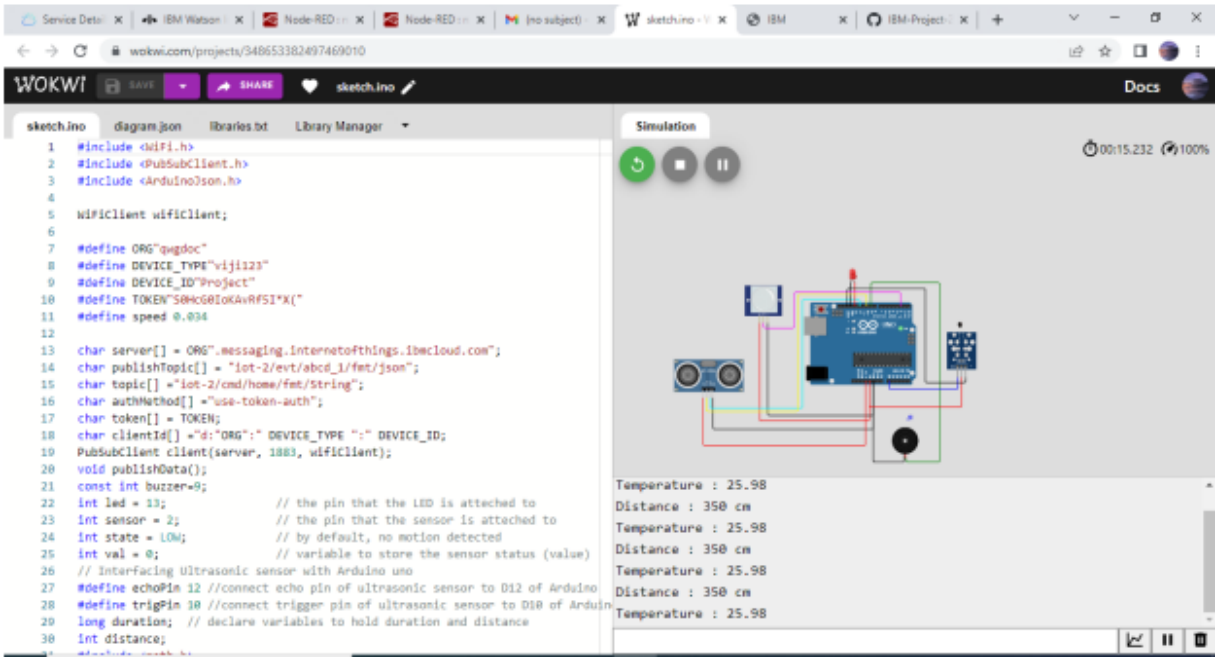
Coding

```
import requests #importing a library
#replace the url and it should be in ""
a="https://api.openweathermap.org/data/2.5/weather?q=Chennai,%20IN&appid=bc453a0b339cb9ee1ad10
d2dd64d0bc0"
r=requests.get(url=a)

print(r)
```

7.2 Feature2

Wokwi


Service Detail
IBM Watson
Node-RED : n
Node-RED : n
(no subject) -
sketch.i
wokwi.com/projects/348653382497469010
WOKWI
SAVE
SHARE
sketch.ino
Simulation
UNO
ARDUINO



Coding

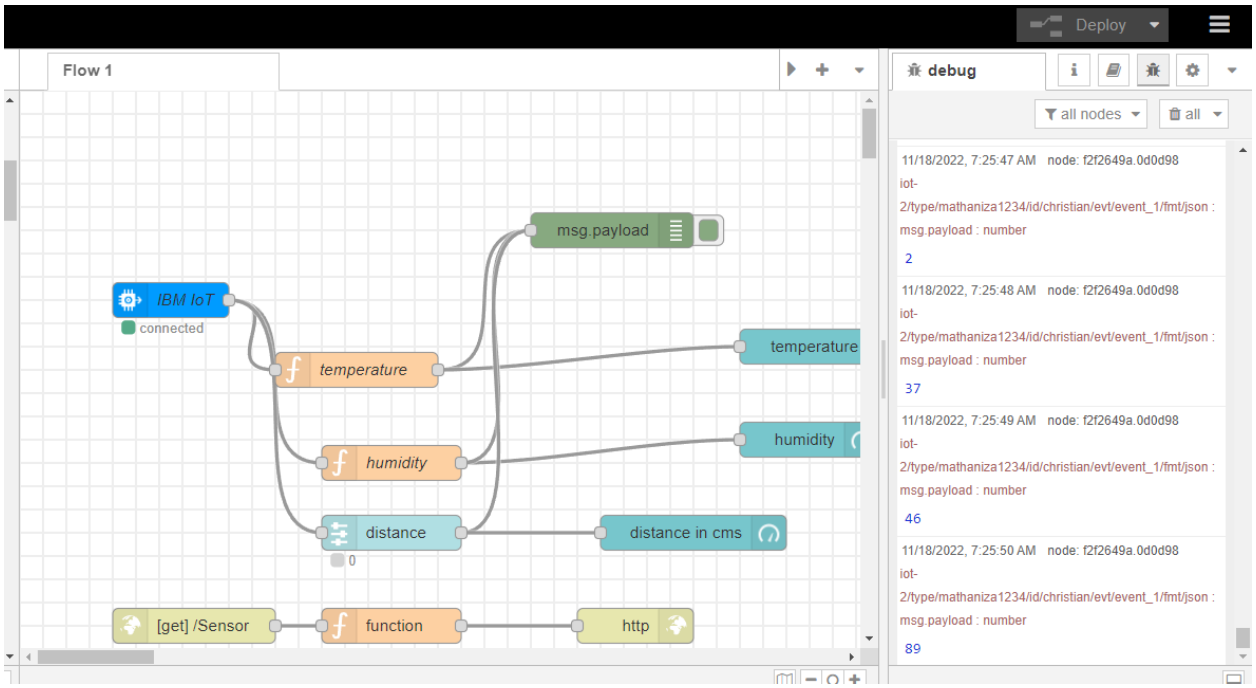
```
#include <WiFi.h>
#include <PubSubClient.h>
#include <ArduinoJson.h>

WiFiClient wifiClient;

#define ORG"qwgdoc"
#define DEVICE_TYPE"viji123"
#define DEVICE_ID"Project"
#define TOKEN"S0HcG0IoKAvRf5I*X("
#define speed 0.034

char server[] = ORG".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/abcd_1/fmt/json";
char topic[] ="iot-2/cmd/home/fmt/String";
char authMethod[] ="use-token-auth";
char token[] = TOKEN;
char clientId[] ="d:"ORG":" DEVICE_TYPE ":" DEVICE_ID;
PubSubClient client(server, 1883, wifiClient);
void publishData();
const int buzzer=9;
int led = 13;                // the pin that the LED is atteched to
int sensor = 2;              // the pin that the sensor is atteched to
int state = LOW;             // by default, no motion detected
int val = 0;                 // variable to store the sensor status (value)
// Interfacing Ultrasonic sensor with Arduino uno
#define echoPin 12 //connect echo pin of ultrasonic sensor to D12 of Arduino
#define trigPin 10 //connect trigger pin of ultrasonic sensor to D10 of Arduino
```

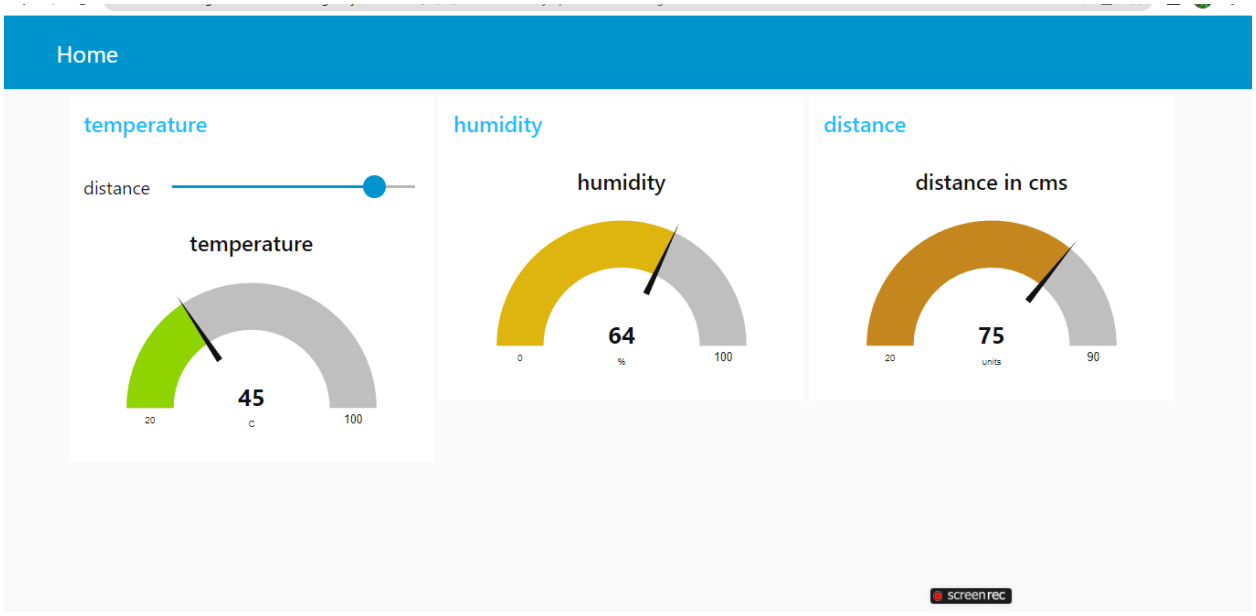
```
long duration;  // declare variables to hold duration and distance
int distance;
#include <math.h>

int sensorPin = A0; // select the input pin for the potentiometer

double Thermistor(int RawADC) {
  double Temp;
  Temp = log(10000.0*((1024.0/RawADC-1)));
  Temp = 1 / (0.001129148 + (0.000234125 + (0.0000000876741 * Temp * Temp ))* Temp );
  Temp = Temp - 273.15;            // Convert Kelvin to Celcius
   //Temp = (Temp * 9.0)/ 5.0 + 32.0; // Convert Celcius to Fahrenheit
   return Temp;
}
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(buzzer, OUTPUT);
  // PIR Sensor
  pinMode(led, OUTPUT);      // initalize LED as an output
  pinMode(sensor, INPUT);    // initialize sensor as an input
  Serial.begin(9600);        // initialize serial
  pinMode(trigPin,OUTPUT); //set trigPin as output pin of Arduino
  pinMode(echoPin,INPUT);  //set echoPin as output pin of Arduino

}

// the loop function runs over and over again forever
void loop() {
    val = digitalRead(sensor);   // read sensor value
  if (val == HIGH) {           // check if the sensor is HIGH
    digitalWrite(led, HIGH);   // turn LED ON
    delay(500);                // delay 100 milliseconds

    if (state == LOW) {
      Serial.println("Motion detected!");
      state = HIGH;       // update variable state to HIGH
    }
  }
  else {
      digitalWrite(led, LOW); // turn LED OFF
      delay(500);             // delay 200 milliseconds

      if (state == HIGH){
        Serial.println("Motion stopped!");
        state = LOW;       // update variable state to LOW
    }
  }
  digitalWrite(trigPin,LOW); //generate square wave at trigger pin
 delayMicroseconds(2);
 digitalWrite(trigPin,HIGH);
 delayMicroseconds(10);
 digitalWrite(trigPin,LOW);
 duration=pulseIn(echoPin,HIGH);//calculation of distance of obstacle
```

```
 distance=(duration*0.034/2);
 Serial.print("Distance : ");
 Serial.print(distance);
 Serial.println(" cm ");
 delay(1000);
 int readVal=analogRead(sensorPin);
 double temp =  Thermistor(readVal);
 Serial.print("Temperature : ");
 Serial.println(temp);  // display tempature
 //Serial.println(readVal);  // display tempature

 delay(500);
 digitalWrite(buzzer, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                       // wait for a second
  digitalWrite(buzzer, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                       // wait for a second
}
```

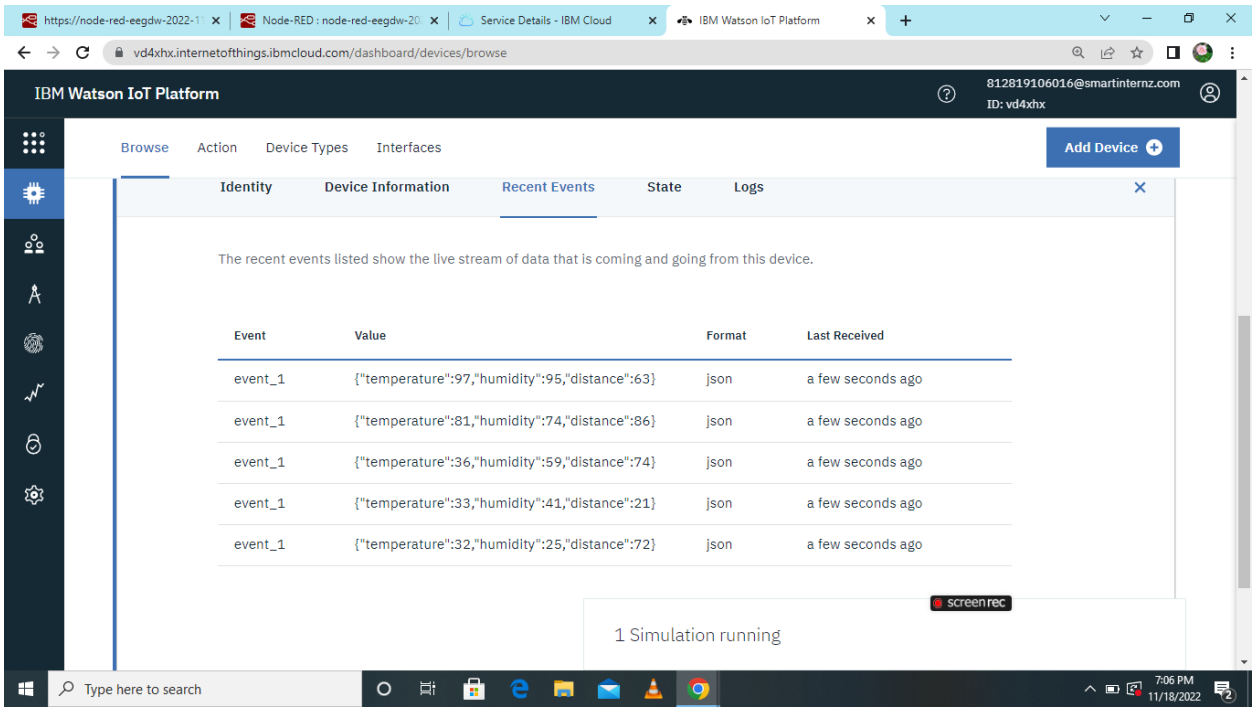
8. TESTING

8.1 Test Cases



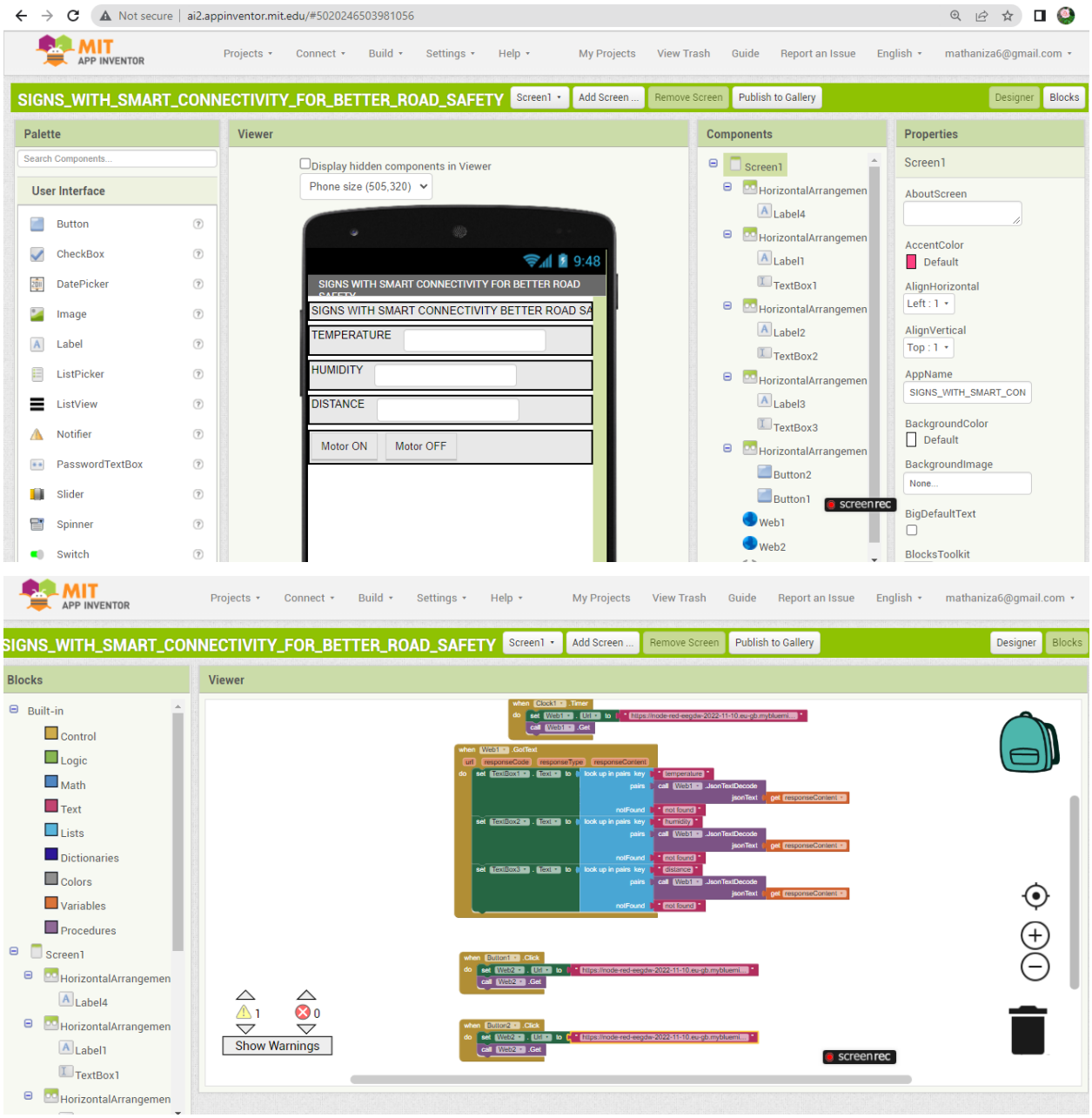


IBM Cloud output



8.2 User Acceptance Testing

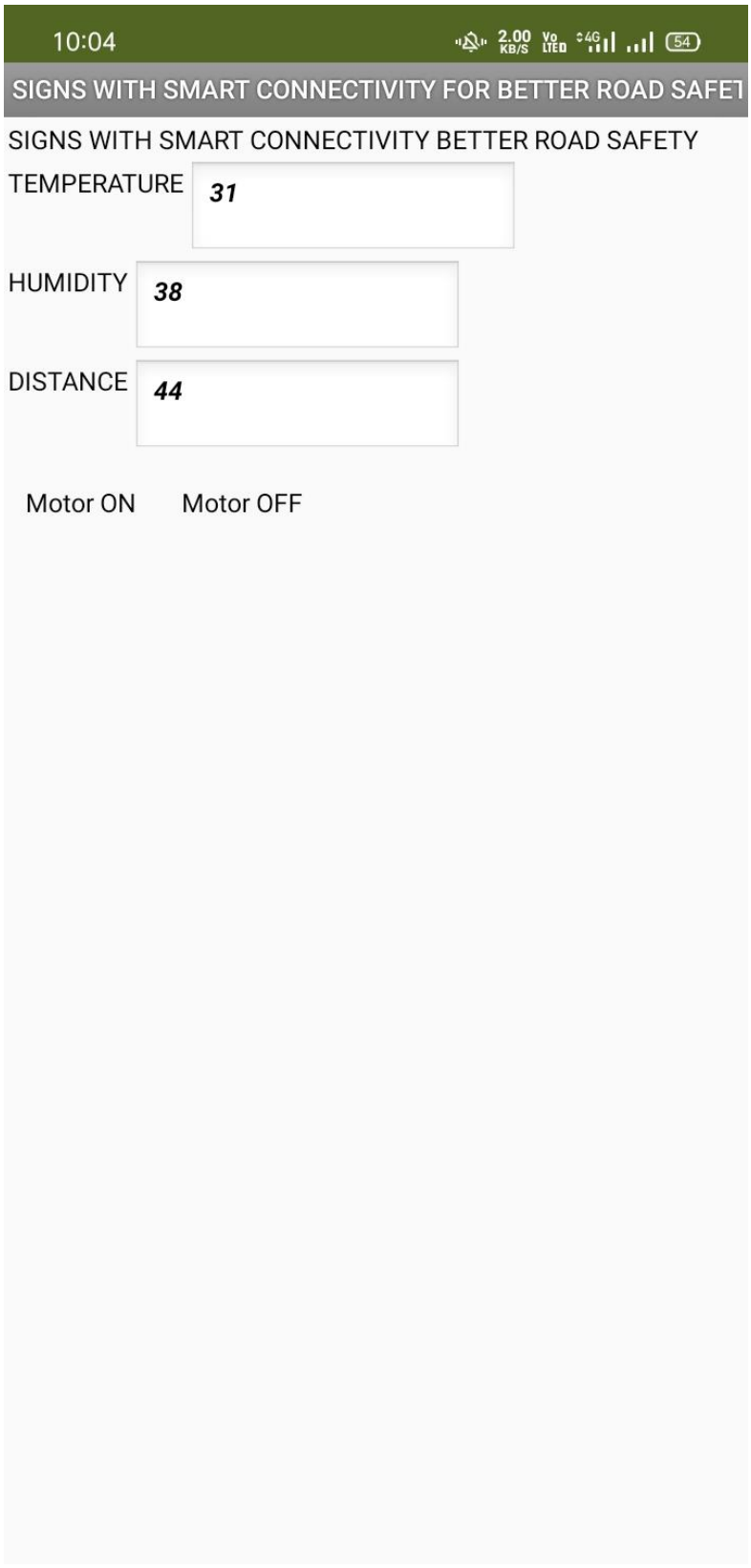
MIT APP Inventor



9. RESULTS

9.1 Performance Metrics

Mobile view of user


10:04
SIGNS WITH SMART CONNECTIVITY FOR BETTER ROAD SAFET
SIGNS WITH SMART CONNECTIVITY BETTER ROAD SAFETY
TEMPERATURE
31
HUMIDITY
38
DISTANCE
44
Motor ON
Motor OFF

10.ADVANTAGES AND DISADVANTAGES

Advantages	Disadvantages
Minimizes the human work and effort	Increased privacy concerns
Saves time and effort	Increased unemployment rates
Good for personal safety and security	Highly dependent on the internet
Useful in traffic and other tracking or monitoring systems	Lack of mental and physical activity by humans leading to health issues.
Beneficial for the healthcare industry	Complex system for maintenance
Improved security in homes and offices	Lack of security
Reduced use of many electronic devices as one device does the job of a lot of other devices	Absence of international standards for better communication

674 × 469

11.CONCLUSION

We have presented a system, to alert the driver about the speed limits in specific areas and reduce the speed of the vehicles in sensitive public zones without any interference of the drivers where controls are taken automatically by the use of a wireless local area network. In the initial phase, we designed the basic block and circuit diagram for the system. In the implementation phase, we executed the Stimulation with the help of IoT connecting tecnologies such as MIT APP. Extensive experiments conducted on IoT and other connecting technologies.

12.FUTURE SCOPE

We can be enhanced this system by implementing camera using Raspberri pi, GSM module in case of network unavailability and low RAM module/zigbee module for long range communication.

13.APPENDIX

Source code

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

#provide your ibm watson Device credentials
organization = "vd4xhx"
devicetype="mathaniza1234"
deviceid="christian"
authmethod="token"
authtoken="JLJdQ8p?5Rizji2Xa"
```

```
#Initialize GPIO

def myCommandCallBack(cmd):
    print("Command received: %s"% cmd.data('command')
    print(cmd)



/*
   PIR sensor tester
*/

(9600);
}

void loop() {int ledPin = 13;                // choose the pin for the LED
int inputPin = 2;               // choose the input pin (for PIR sensor)
int pirState = LOW;             // we start, assuming no motion detected
int val = 0;                    // variable for reading the pin status

void setup() {
  pinMode(ledPin, OUTPUT);      // declare LED as output
  pinMode(inputPin, INPUT);     // declare sensor as input

  Serial.begin
  val = digitalRead(inputPin);  // read input value
  if (val == HIGH) {            // check if the input is HIGH
    digitalWrite(ledPin, HIGH);  // turn LED ON
    if (pirState == LOW) {
      // we have just turned on
      Serial.println("Motion detected!");
      // We only want to print on the output change, not state
      pirState = HIGH;
    }
  } else {
    digitalWrite(ledPin, LOW); // turn LED OFF
    if (pirState == HIGH) {
      // we have just turned of
      Serial.println("Motion ended!");
      // We only want to print on the output change, not state
      pirState = LOW;
    }
  }
}


import json
#replace the url and it should be in""
a="https://api.openweathermap.org/data/2.5/weather?q=chennai,%20IN&appid=bc453a0b339cb9ee1ad10d2dd64d0bc0"
r=requests.get(url=a)
print(r)
data =r.json ()
tem=data ('main')('h')
print(tem)

/*
  HC-SR04 Ultrasonic Sensor Example.
```

```
  Turn the LED on when an object is within 100cm range.

  Copyright (C) 2021, Uri Shaked
*/

#define ECHO_PIN 2
#define TRIG_PIN 3

void setup() {
  Serial.begin(115200);
  pinMode(LED_BUILTIN, OUTPUT);
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
}

float readDistanceCM() {
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  int duration = pulseIn(ECHO_PIN, HIGH);
  return duration * 0.034 / 2;
}

void loop() {
  float distance = readDistanceCM();

  bool isNearby = distance < 100;
  digitalWrite(LED_BUILTIN, isNearby);

  Serial.print("Measured distance: ");
  Serial.println(readDistanceCM());

  delay(100);
}


DEMO using the server test.mosquitto.org

You can use any MQTT client with the following settings
Server : test.mosquitto.org
no login / no password
port: 1883 or 8081 for websocket
Topic: /AnnexTest
Subscribe: /AnnexTx

Or you can use the free MQTT online client

https://www.cicciocb.com/MQTT/

this is already configured so just
- click on Connect
- write your message in "Publish Message"
- Press "Publish" to send your message that will be shown in the scrolling display
- Click on Subscribe to receive the temperature sensor data
```

```
# Disconnect the device and the application from the cloud
devicecli.disconnect()
```

GITHUB LINK: https://github.com/IBM-EPBL/IBM-Project-20790-1659763214

DEMO LINK: https://drive.google.com/file/d/1DTZk3AGML1iaFbhNa-Tm3wNQ6xPG6bbV/view?usp=share_link