**CHAPTER 1**

**INTRODUCTION**

* 1. **Project overview:**

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.

* 1. **Problem Definition:**

SMART CONNECTED SIGN BOARDS are used in place of static signboards. These smart connected sign boards receive automatic updates and speed limitations via a web application that uses the weather API. The speed could both increase and decrease based on weather changes. the diversion signs are displayed based on traffic conditions and fatal situations. Guide, warning, and service signs are also displayed appropriately (hospitals, restaurants). With the help of buttons, different modes of operation can be selected.

* 1. **Objective:**
* The main objective of our project is to reduce the speed the speed limit of the vehicles
* By using the weather API we will change the speed limit in the sign board according to the current weather.
* We will generate the fine who are crossing the speed limit in digital sign board
* By using this sign board we will avoid the unwanted noise from the vehicles

**CHAPTER 2**

**LITERATURE SURVEY**

1.Ashish Dhar: **Traffic and road condition monitoring system** Indian Institute of Technology, Mumbai. - 2008.

•Reports severity, intensity and dimension of a damaged road segment.

• Proposed a different solution using AMR Magnetic Sensor.

2. Pooja Pawar, Suvarna Langade, Mohini Bandgar**: IOT Based digital Notice Board using Arduino ATMega 328.**

International Research Journal of Engineering and Technology(IRJET).- 2019.

•Circulates notice regularly & reduce physical efforts.

• Send message at any distant location within a second.

3. Sandeep Chaware, Trushitha Chaware: **Proposed Algorithm for Smart Traffic Control using Ultrasonic Sensor.**

International Journal of Engineering and Advanced Technology (IJEAT).- 2019.

•The outcome of the project is to learn insights of the traffic controlling and management at

the signal with the dynamically changing in timing of timer as per need.

4. KamnaSingh, Deepa Bura**: IOT distinct algorithms for the Sensor Connectivity with Comparative Study between node MCU and Arduino MCU.**

NVEO Journal– 2021

• Presents different algorithms for the connection between different types of sensors.

•Brief description of node MCU & Arduino MCU.

• Step by step solution to provide connectivity with IOT technology.

5. Jack Greenhaigh: **Recognizing Text Based Traffic Signs.**

IEEE – 2015

• Detect all possible Road sign candidates.

•Reduce total regions based on contextual constraints.

• A Novel System for the automatic detection and recognition of text in traffic sign based on MSER & MSV.

6. Bhumika.R, Harshita. S.A, Meena. D, Asha. N: **Accident Prevention and Road Safety in Hilly Region using IOT Module**

International Research Journal of Engineering and Technology (IRJET) – 2021

• Stay away from mishap & forestall clog in sloping region & hairclip twist.

• As a significant part of street mathematical plan bended street portion

7. Sowparnika: **IOT Road Safety**

•This project paves a system to alert the driver about the speed limit in specific areas and to

reduce the speed of vehicles in sensitive public zones without any interference of drivers where

controls are taken automatically by use of wireless local area network.

8. S.S. Sugania, D. S. Vishalis Hwaran, J. Vignesh Kumar: **Automated System for Road Safety Enhancement using big data reports.**

• The speed is controlled accordingly to situations to give suggestions.

• The suggested system can control the vehicle but at same time can collect data and

manipulate it using the big data technologies.

9**. IOT Based Smart Road Safety & Vehicle Accident prevent System for Mountain roads**.

•This system is divided into 2 half (Accident Detection & Prevention) and alerting the members of family by causation message and placement of accidental place.

10. Shweta Vyas, Pooja Awhale, Shreya Kukdeja, Prashant Jawalkar: **A Modern Approach to identify Traffic Sign Symbols in Color Images.**

•In this technique proposed more reliable and robust method of Traffic Sign Detection Recognition (TSDR).

**CHAPTER 3**

**SYSTEM ANALYSIS**

**3.1 EXISTING SYSTEM:**

The Existing road system and connectivity, emphasis on the traffic and route reckoning features which cordially provisions the user acceptability to have better connectivity management. But, this often results in nonparallel road conditions and high noise ratios through the calibrations. It reiterates various subjections in its compilation and leading to segmentation error throughout. It penetrates the various unit cases in order to subsequently manifest the output. This alternatively symbolizes the ineffectively programmed web user interface. The IOT based model of our project complies of the verdict to specify the soft zone in the path. It manually ask the user to turn off the horn, which in variably decreases the decibel level of the power output. Illustratively, it confides the work schematics of the precedent evaluation under the system and allows the user to access the terminals of the app nodes variably. IBM Cloud indefinitely helps in reviving the data sets required in web application. MIT app inventor segments the creation of the user interface.

**3.1.1 DISADVANTAGE:**

* + However, this finding is contrary to previous research that suggests differences in crash counts exist in the presence of static roadside advertising.
  + The quantity and quality of available evidence limit our conclusion.
  + Fixed object, side swipe and rear end crashes are the most common types of crashes in the presence of roadside advertising signs.

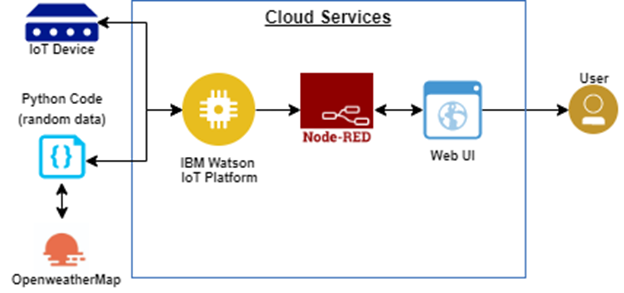
**3.2 PROPOSED SYSTEM:**

This result defies earlier study, which claimed that the presence of static roadside advertising led to disparities in collision statistics. The amount and calibre of the evidence that is available constrain our decision. The most frequent sorts of collisions where there are roadside advertising signs are those involving fixed objects, side swipes, and rear ends.

**3.2.1 ADVANTAGES:**

* Low-cost, low-requirement micro controllers can be employed as processing is mostly handled by Node RED servers, resulting in lower battery usage.
* Systems with longer lifespans.
* Dynamic sign-up updates.
* Alerts for the School/Hospital Zone.

**3.3ARCHITECTURE DESIGN:**



**3.5 DESCRIPTION OF MODULE:**

* Wokwi NodeMCU
* Dashboard
* Vehicles no Detection

**3.5.1 WOKWI NODEMCU;**

Wokwi Nodemcu is an improvement board that is intended to assist producers and specialists with getting everything rolling with the NodeMCU stage. It includes an ESP8266 microcontroller, which is a minimal expense Wi-Fi empowered chip that can be customized utilizing the Arduino IDE.

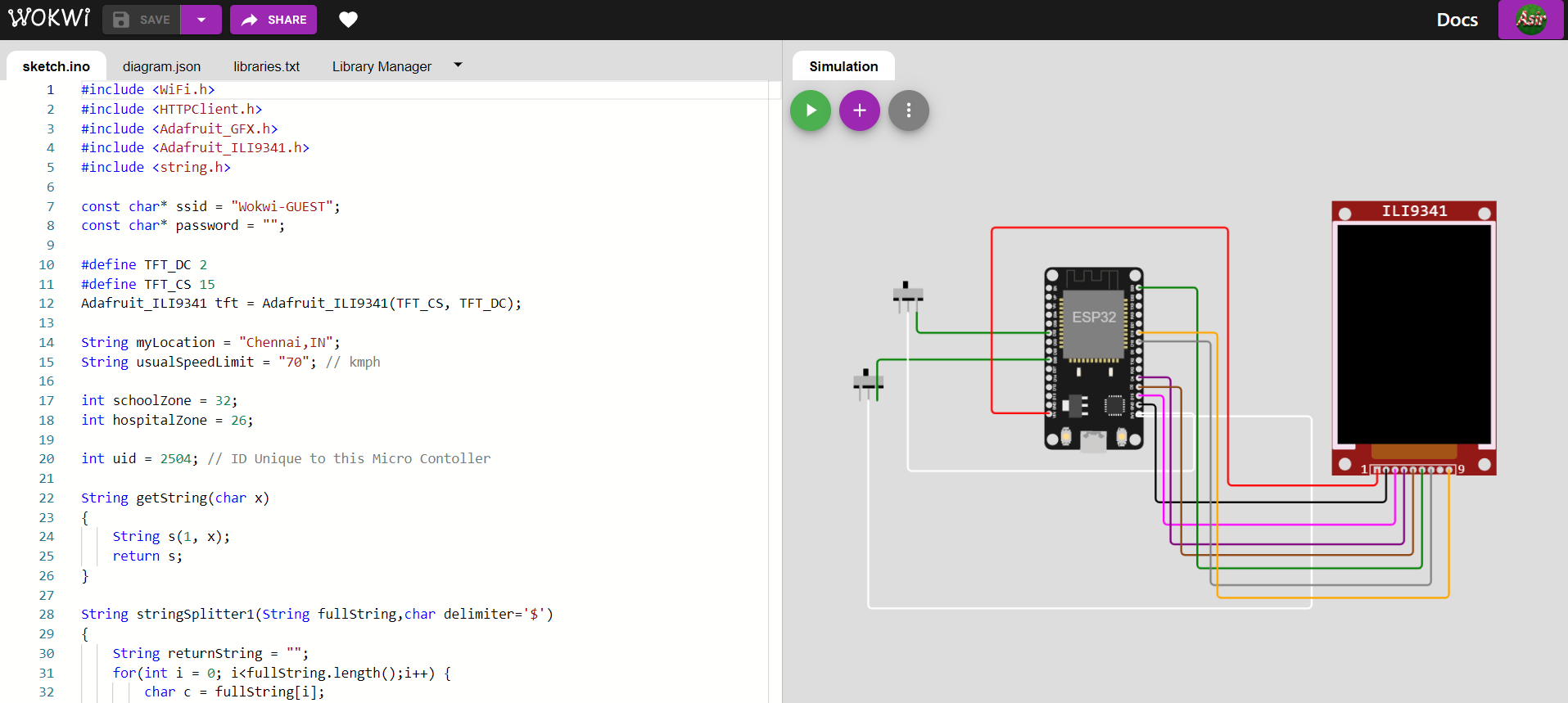
****

Fig 3.5.1 Wokwi Interface

**3.5.2 DASHBOARD:**

A dashboard is a user interface that displays key performance indicators and other significant data graphically. It provides a summary of the current state of traffic rule violations and fines assessed. By analysing this data, we can identify the overseeded cars

and put a stop to them.

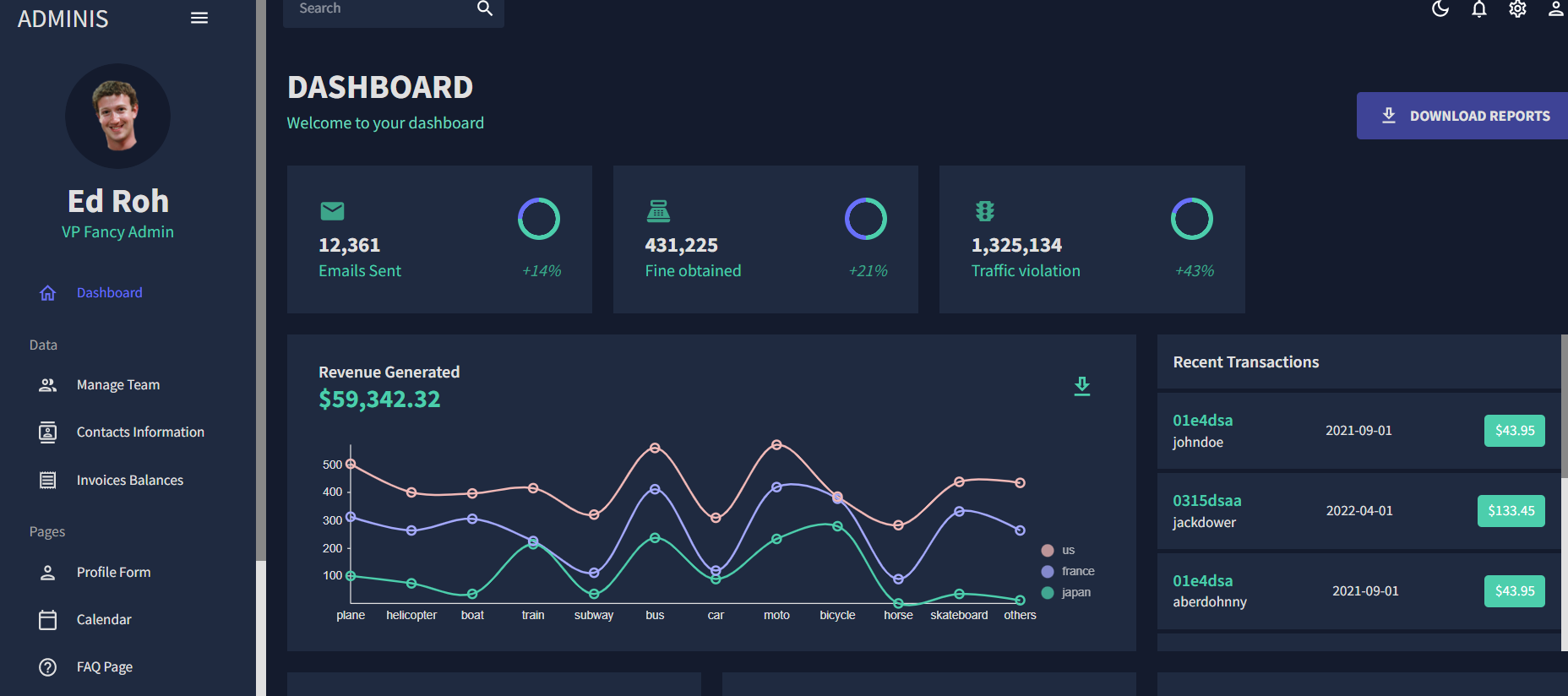


Fig 3.5.2 Dashboard

**3.5.3 VECHILE NO DETECTION:**

By using this we can detect the vehicle no by using cameras, so that we can easily find the traffic rules violation vehicles and we can generate the fine or we can take the details of the vehicle. In this we will detect only the no plate of the vehicles.



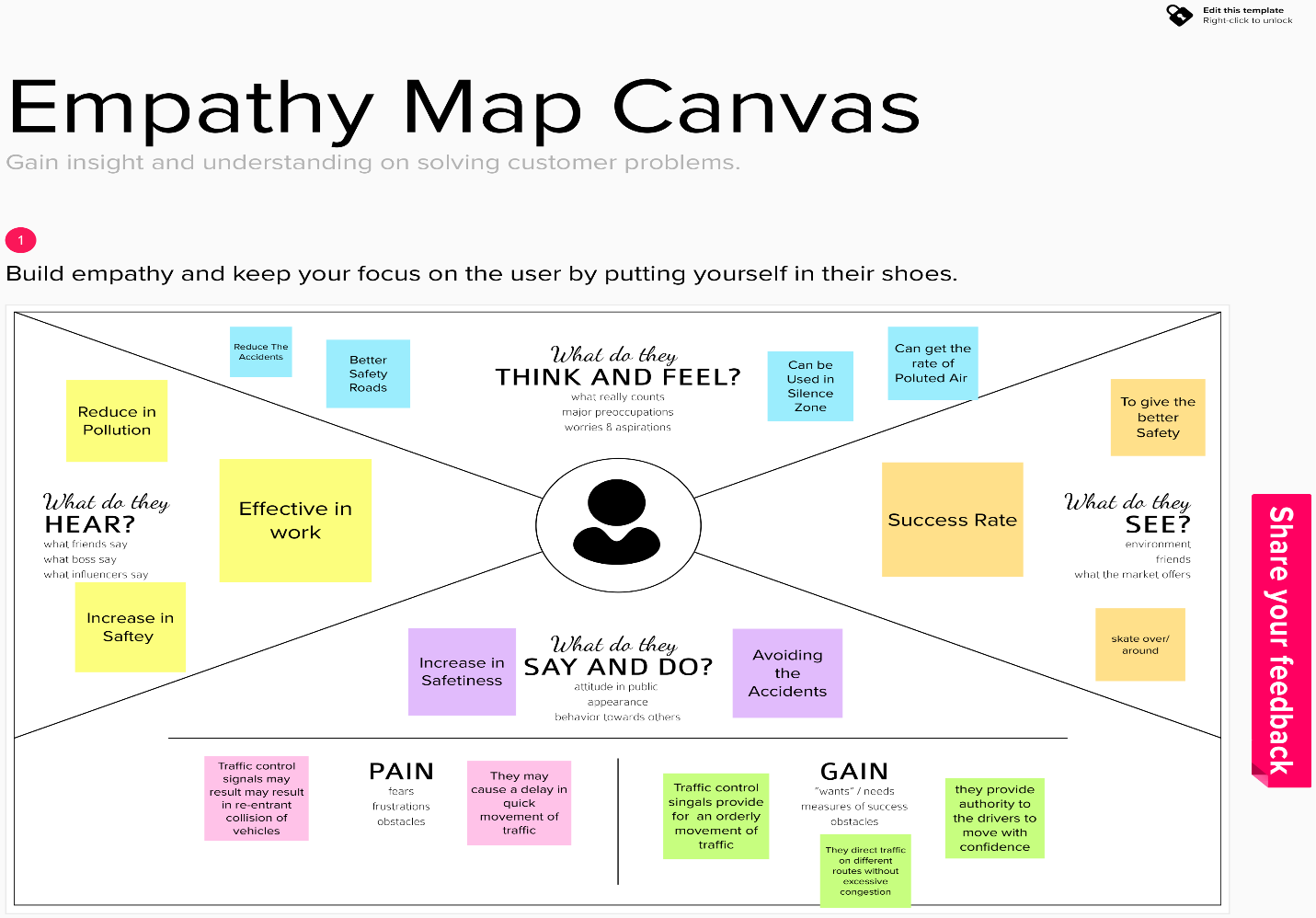
Fig 3.5.3 Vehicles No Detection

**CHAPTER 4**

**IDEATION PHASE**

**4.1 Empathy Map Canvas:**

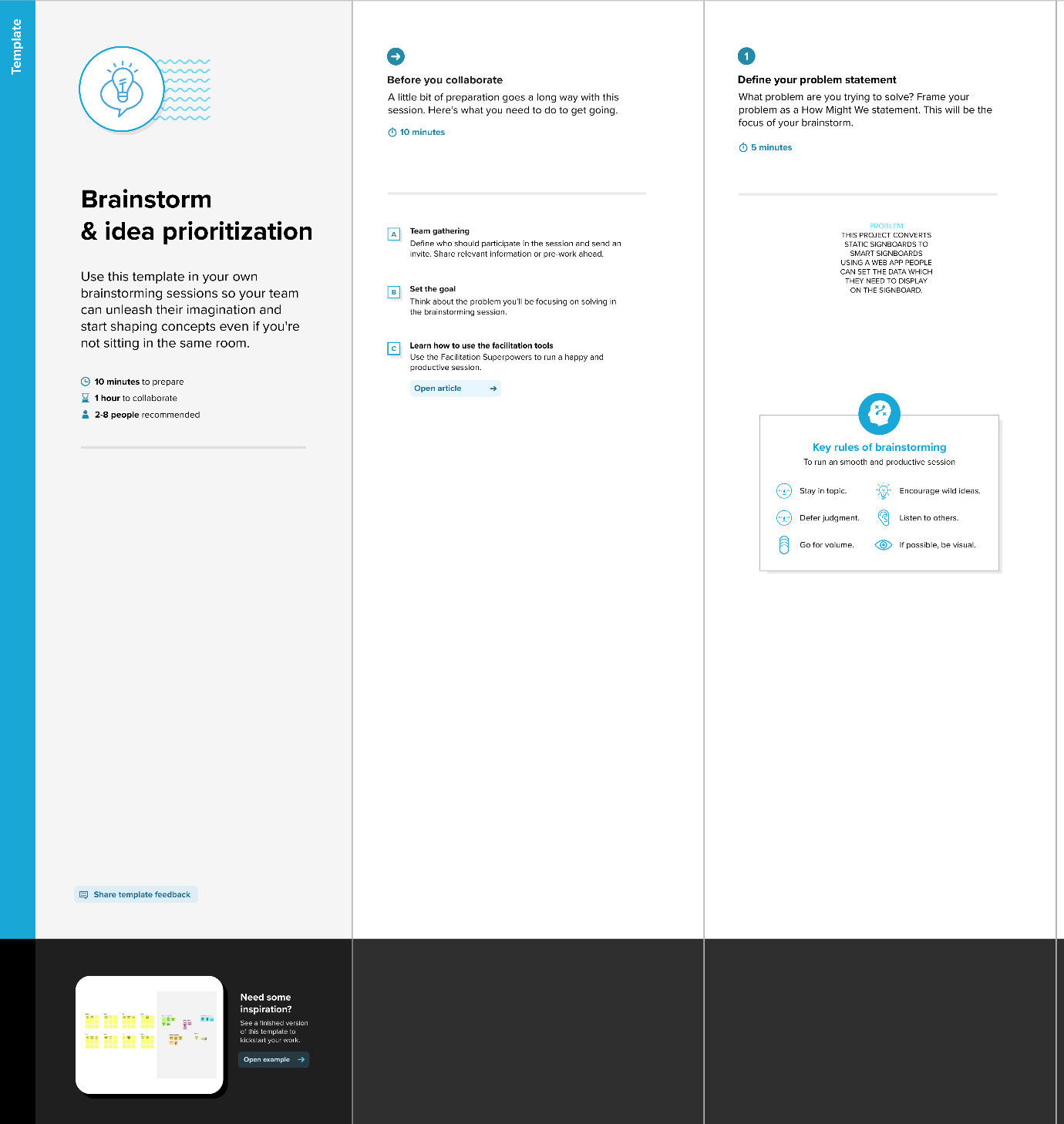
A compassion map is a basic, simple to-process visual that catches information about a client's ways of behaving and perspectives. It is a valuable device to assist teams with better comprehension of their clients.Making a compelling arrangement requires grasping the genuine issue and the individual who is encountering it. The activity of making the guide helps members consider things according to the client's point of view alongside their objectives and challenges



**4.2 Ideation & Brainstorming:**

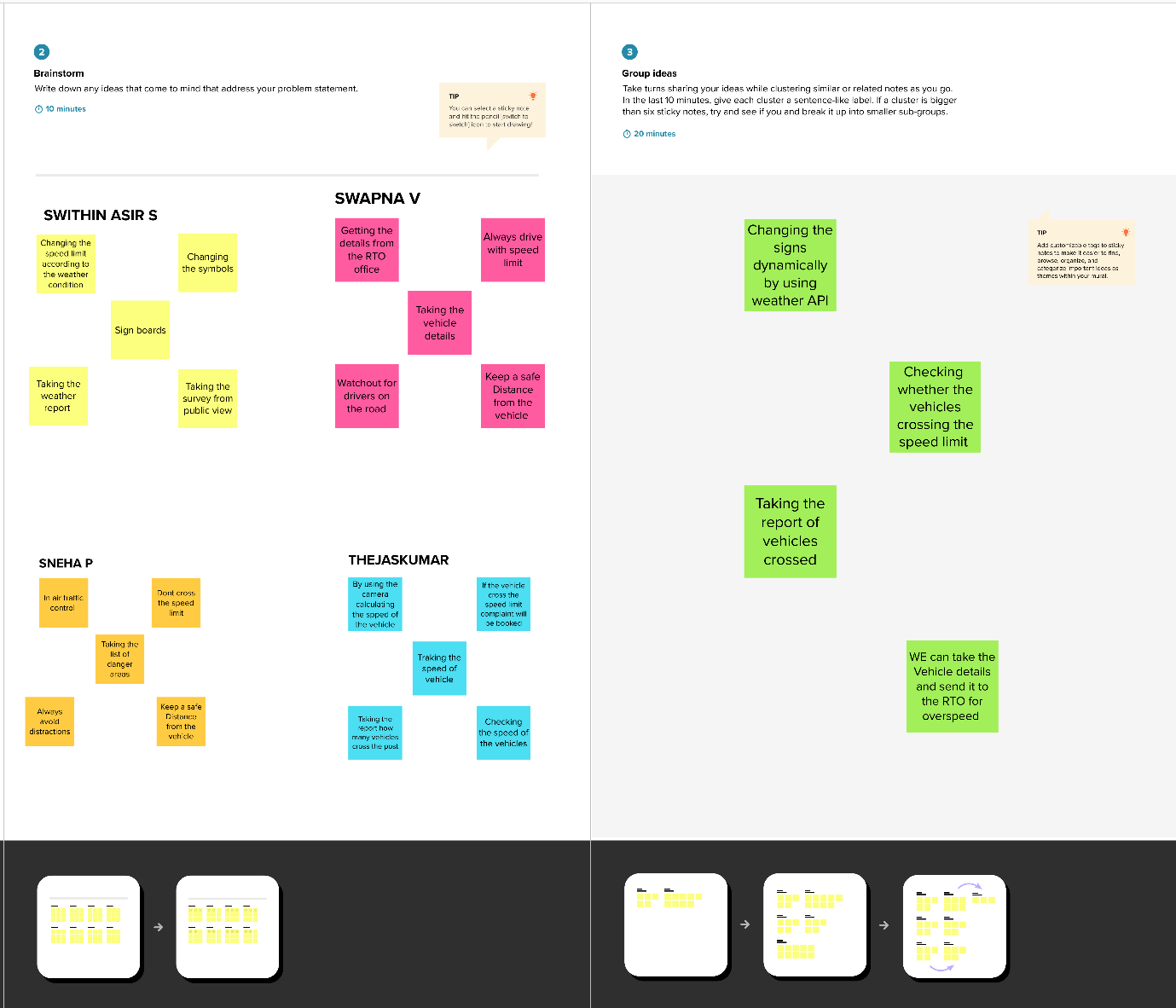
**Step-1: Team Gathering, Collaboration and Select the Problem Statement**

In this step, team members gather and provide their ideas and collaborate on those ideas and select their problem statement. The ideas should be relevant to their problem statement.



**Step-2: Brainstorm, Idea Listing and Grouping**

In this step they put their ideas and views which are prioritized based on their importance and the ideas are grouped. These ideas are categorized according to their relevant classifications.



**Step-3: Idea Prioritization**

As mentioned, idea prioritization is just a part of the idea management process. Having a structured idea management process and a systematic way of gathering, evaluating and prioritizing new ideas takes time. To make it work, the entire idea management process should be integrated into everyday ways of working.

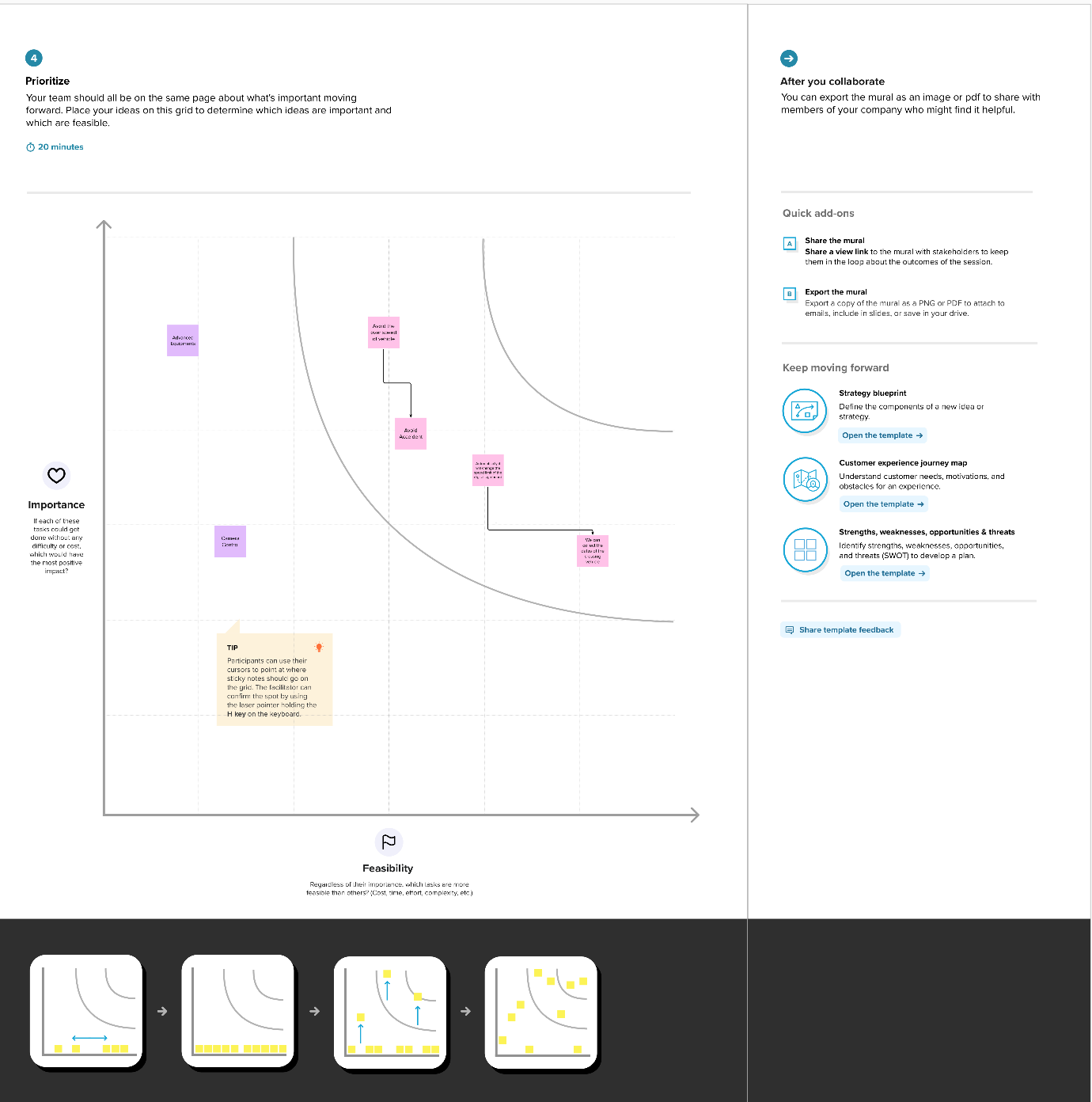
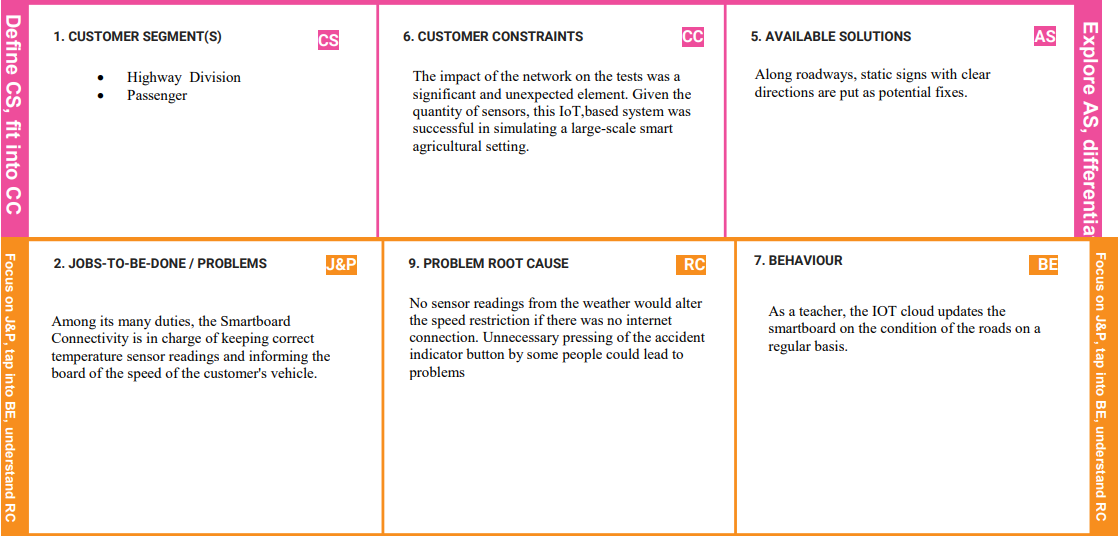


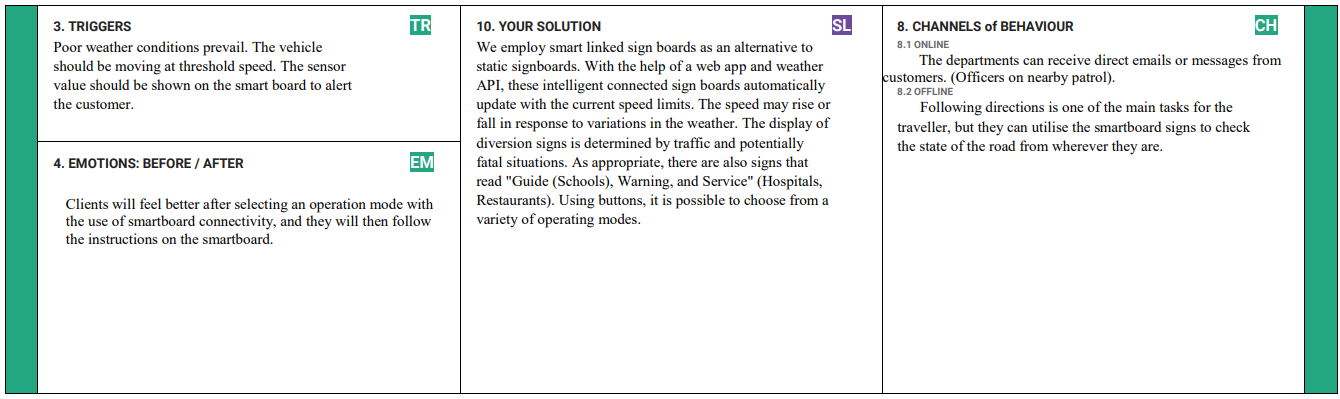
Fig 4.2 Brainstorming

**4.3 Proposed Solution:**

|  |  |  |
| --- | --- | --- |
| **S. No** | **Parameter** | **Description** |
|  | Problem Statement (Problem to be solved) | Avoid the over speed and to decrease the accidents |
|  | Idea / Solution description | * 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. |
|  | Novelty / Uniqueness | Sign boards are converted to digitals |
|  | Social Impact / Customer Satisfaction | Reduce the accidents , Control the vehicles in speed |
|  | Business Model (Revenue Model) | In this we can get good no of users ,so that the business can get profit |
|  | Scalability of the Solution | We can scalable the project by schools and colleges as our customer |

**4.4 Problem Solution fit:**





**CHAPTER 5**

**REQUIREMENT ANALYSIS**

**4.1 Functional Requirements:**

|  |  |  |
| --- | --- | --- |
| **FR No.** | **Liberation Serif Regular** | **Liberation Serif Regular** |
| FR-1 | User Registration | Sign boards should be made with LEDs that are brightly coloured and capable of attracting the attention of drivers, but they should not be too distracting or blinding as this may lead to accidents. |
| FR-2 | User Understanding | The signs should be large, clear, and legible in order for the motorist to comprehend them, and they can also incorporate images. |
| FR-3 | User Convenience | The signs should be large, clear, and legible in order for the motorist to comprehend them, and they can also incorporate images. |

**4.2 Non-Functional Requirements:**

|  |  |  |
| --- | --- | --- |
| **FR No.** | **Non-Functional Requirement** | **Description** |
| NFR-1 | **Usability** | When necessary, it should be able to upgrade and update. |
| NFR-2 | **Security** | When necessary, it should be able to upgrade and update. |
| NFR-3 | **Reliability** | It should be able to show information appropriately and without errors. |
| NFR-4 | **Performance** | It should be able to update itself automatically when a weather or traffic problem happens. |
| NFR-5 | **Availability** | It should be accessible 24 hours a day, seven days a week in order to benefit the consumer, i.e. the driver. |
| NFR-6 | **Scalability** | It should be simple to update and upgrade in response to changes in the requirements. |

**CHAPTER 6**

**SYSTEM REQUIREMENTS AND DESIGN**

* 1. **HARDWARE REQUIREMENT:**
* Raspberry pi 4
* LED Display
* Ultrasonic Sensor
* Motion Sensor
* Camera

**6.2 SOFTWARE REQUIREMENTS:**

* Node-Red
* IBM Cloud
* Python

**6.1.1 Raspberry pi 4:**

Raspberry Pi (/pa/) is a line of tiny single-board computers (SBCs) created by the Raspberry Pi Foundation in collaboration with Broadcom in the United Kingdom. The Raspberry Pi initiative began with the goal of promoting fundamental computer science education in schools and disadvantaged nations. The initial model proved to be more successful than expected, selling outside of its intended market for purposes such as robotics. Because of its inexpensive cost, versatility, and open design, it is frequently used in numerous fields, including weather monitoring. Because it adheres to the HDMI and USB standards, it is commonly used by computer and electronic enthusiasts.

The Raspberry Pi 400 is a whole personal computer housed in a small keyboard. It's the most powerful and user-friendly Raspberry Pi computer ever, with a quad-core 64-bit CPU, 4GB of RAM, WiFi networking, dual-display output, and 4K video playback, as well as a 40-pin GPIO header. A Raspberry Pi may be used for a variety of purposes. Popular usage includes converting your Raspberry Pi into a classic arcade machine, utilizing it as a web server, or using it as the brain for a robot, security system, IoT device, or specialized Android smartphone.

****

Fig 6.1.1 Raspberry Pi 4

**6.1.2 LCD DISPLAY:**

16×2 LCD is one kind of electronic device used to display the message and data. The term LCD full form is **Liquid Crystal Display**.  The display is named 16×2 LCD because it has 16 Columns and 2 Rows. it can be displayed (16×2=32) 32 characters in total and each character will be made of 5×8 Pixel Dots. These displays are mainly based on multi-segment light-emitting diodes.  There are a lot of combinations of display available in the market like 8×1, 8×2, 10×2, 16×1, etc. but the 16×2 LCD is widely used. These LCD modules are low cost, and programmer-friendly, therefore, is used in various DIY circuits, devices, and embedded projects.

These 16 x 2 LCD display modules are constant of 16 Columns and 2 Rows. The 1st row of this module has a total of 16 columns 0 to 15 and the position of the first row is 0. Also, the 2nd row has a total of 16 columns 0 to 15 and the position of the second row is position is So the total numbers of the column are 16 x 2 = 32. Its means 16 x 2 LCD module can display 32 characters at the same time.

It will be a very complicated task to handle everything with the help of a microcontroller. So an Interface IC like HD44780 is used, which is mounted on the backside of the LCD Module. The function of this IC is to get the Commands and Data from the microcontroller and process them to display meaningful information onto the LCD Screen.



Fig 6.1.2 LCD DISPLAY

**6.1.3 Ultrasonic Sensor:**

An ultrasonic sensor is an electronic device that measures the distance of a target object by emitting ultrasonic sound waves, and converts the reflected sound into an electrical signal. Ultrasonic waves travel faster than the speed of audible sound (i.e. the sound that humans can hear). Ultrasonic sensors have two main components: the transmitter (which emits the sound using piezoelectric crystals) and the receiver (which encounters the sound after it has travelled to and from the target).

In order to calculate the distance between the sensor and the object, the sensor measures the time it takes between the emission of the sound by the transmitter to its contact with the receiver. The formula for this calculation is [D = ½ T x C](https://www.arrow.com/en/research-and-events/articles/ultrasonic-sensors-how-they-work-and-how-to-use-them-with-arduino) (where D is the distance, T is the time, and C is the speed of sound ~ 343 meters/second). For example, if a scientist set up an ultrasonic sensor aimed at a box and it took 0.025 seconds for the sound to bounce back, the distance between the ultrasonic sensor and the box would be:

|  |
| --- |
| **D = 0.5 x 0.025 x 343** |

Ultrasonic sensors are used primarily as [proximity sensors](https://www.fierceelectronics.com/sensors/what-a-proximity-sensor). They can be found in automobile self-parking technology and anti-collision safety systems. Ultrasonic sensors are also used in robotic obstacle detection systems, as well as manufacturing technology. [In comparison to infrared (IR) sensors](https://www.maxbotix.com/articles/ultrasonic-or-infrared-sensors.htm) in proximity sensing applications, ultrasonic sensors are not as susceptible to interference of smoke, gas, and other airborne particles (though the physical components are still affected by variables such as heat).

Ultrasonic sensors are also used as [level sensors](https://www.fierceelectronics.com/sensors/what-a-level-sensor) to detect, monitor, and regulate liquid levels in closed containers (such as vats in chemical factories). Most notably, ultrasonic technology has enabled the medical industry to produce images of internal organs, identify tumors, and ensure the health of babies in the womb.



Fig 6.1.3 ULTRASONIC SENSOR

**6.1.4 MOTION SENSOR:**

A motion sensor (or motion detector) is an electronic device that is designed to detect and measure movement. Motion sensors are used primarily in home and business security systems, but they can also be found in phones, paper towel dispensers, game consoles, and virtual reality systems. Unlike many other types of sensors (which can be handheld and isolated), motion sensors are typically embedded systems with three major components: a [sensor](https://www.fierceelectronics.com/sensors/what-a-sensor) unit, an [embedded computer](https://www.fierceelectronics.com/embedded/what-embedded-computer), and hardware (or the mechanical component). These three parts vary in size and configuration, as motion sensors can be customized to perform highly specific functions. For example, motion sensors can be used to activate floodlights, trigger audible alarms, activate switches, and even alert the police.

There are two types of motion sensors: active motion sensors and passive motion sensors. Active sensors have both a transmitter and a receiver. This type of sensor detects motion by measuring changes in the amount of sound or radiation reflecting back into the receiver. When an object interrupts or alters the sensor’s field, an electric pulse is sent to the embedded computer, which in turn interacts with the mechanical component. The most common type of active motion detector uses [ultrasonic sensor](https://www.fierceelectronics.com/sensors/what-ultrasonic-sensor) technology; these motion sensors emit sound waves to detect the presence of objects. There are also microwave sensors (which emit microwave radiation), and tomographic sensors (which transmit and receive radio waves).

Unlike an active motion sensor, a passive motion sensor does not have a transmitter. Instead of measuring a constant reflection, the sensor detects motion based on a perceived increase of radiation in its environment. The most widely used type of passive motion sensor in home security systems is the [passive infrared (PIR) sensor](https://www.fierceelectronics.com/sensors/what-ir-sensor). The PIR sensor is designed to detect the infrared radiation emitted naturally from the human body. The receiver is contained in a filter that only allows infrared to pass through it. When a person walks into the PIR sensor’s field of detection, the difference in radiation creates a positive charge within the receiver; this perceived change causes the sensing unit to send electrical data to the embedded computer and hardware component.



Fig 6.1.4 Motion Sensor

**6.1.5 CAMERA:**

**Motion-picture camera**, also called **Movie Camera**, any of various complex photographic cameras that are designed to record a succession of images on a reel of [film](https://www.britannica.com/technology/film-photography) that is repositioned after each [exposure](https://www.britannica.com/technology/exposure-photography). Commonly, exposures are made at the rate of 24 or 30 frames per second on film that is either 8, 16, 35, or 70 mm in width.

A motion-picture [camera](https://www.britannica.com/technology/camera) essentially consists of a body, a film-transport system, [lenses](https://www.britannica.com/technology/lens-optics), shutter, and a viewing-focusing system. The motor-driven transport system is the chief element that [differentiates](https://www.merriam-webster.com/dictionary/differentiates) motion-picture cameras from still cameras. Within the camera, the unexposed [film](https://www.britannica.com/technology/film-magazine) is housed in a totally dark chamber called the forward magazine. One or both edges of the film are lined with regularly spaced perforations, or sprocket holes. Sprocket-driven gears grip these perforations, feeding the film into an enclosed exposure chamber. A mechanical claw pulls the film into position behind the shutter, locking the film momentarily in place. The shutter opens, exposes an image onto the film, and closes. Then the claw, with an automatic pulldown movement, advances the film for the next exposure. Each frame of the film comes to a complete stop for its exposure, and hence each exposure is a single still photograph, or frame. As the film moves through the camera, the exposed sections are fed into the rear magazine, which is another totally dark chamber.

**6.2.1 NODE-RED:**

Node-RED is a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways.It provides a browser-based editor that makes it easy to wire together flows using the wide range of nodes in the palette that can be deployed to its runtime in a single-click. Node-RED consists of a Node.js based runtime that you point a web browser at to access the flow editor. Within the browser you create your application by dragging nodes from your palette into a workspace and start to wire them together. With a single click, the application is deployed back to the runtime where it is run. The palette of nodes can be easily extended by installing new nodes created by the community and the flows you create can be easily shared as JSON files.

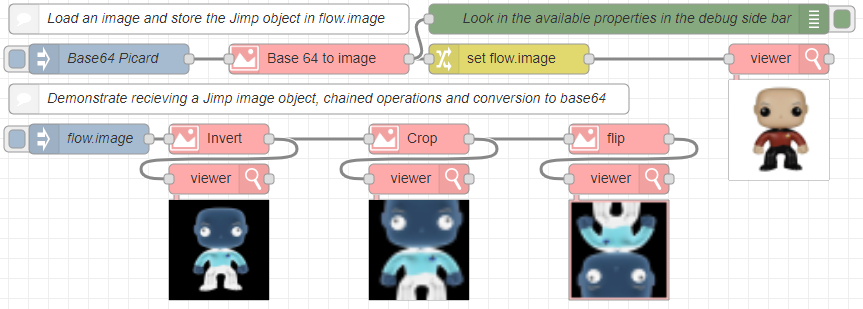


Fig 6.2.1 NODE-RED

**6.2.2 IBM CLOUD:**

IBM Cloud (previously Bluemix) is a suite of cloud computing services for businesses provided by the information technology firm IBM. IBM Cloud Paks are hybrid cloud software packages that allow you to design apps once and deploy them everywhere. VPC is a public cloud service that allows you to create your own private cloud-like computing environment on shared public cloud infrastructure. The information technology company IBM offers a variety of cloud computing services for enterprises under the name IBM Cloud (formerly Bluemix). The hybrid cloud software packages known as IBM Cloud Paks let you create apps just once and distribute them throughout the globe. With the help of the VPC public cloud service, you may build a private cloud-like computing environment on top of a shared public cloud infrastructure.

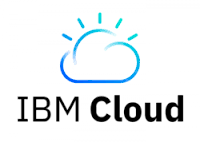
****

Fig 6.2.2 IBM CLOUD

**CHAPTER 7**

**IMPLEMENTATION AND RESULT**

**7.1 COMPONENT DESCRIPTION:**

**7.1.1 Raspberry Pi4:**

Raspberry Pi 4 is one of the recent Raspberry Pi single-board computers. In this blog, you will see Raspberry pi 4 specifications Pin Diagram and Description. You can do all kinds of basic tasks using this computer what you need one monitor, keyboard and mouse. you can play low graphics games, stream 4k video connecting on two displays, all your office work, and many more.This small computer is designed for learning purposes. It was made for those who can’t afford expensive computer hardware but still want to learn. The performance of this small computer is similar to entry-level x86 based pc systems

**7.1.2 Ultrasonic Sensor:**

An ultrasonic sensor is an electronic device that measures the distance of a target object by emitting ultrasonic sound waves, and converts the reflected sound into an electrical signal. Ultrasonic waves travel faster than the speed of audible sound (i.e. the sound that humans can hear). Ultrasonic sensors have two main components: the transmitter (which emits the sound using piezoelectric crystals) and the receiver (which encounters the sound after it has travelled to and from the target).

**7.1.3 Motion Sensor:**

A motion sensor (or motion detector) is an electronic device that is designed to detect and measure movement. Motion sensors are used primarily in home and business security systems, but they can also be found in phones, paper towel dispensers, game consoles, and virtual reality systems. Unlike many other types of sensors (which can be handheld and isolated), motion sensors are typically embedded systems with three major components: a [sensor](https://www.fierceelectronics.com/sensors/what-a-sensor) unit, an [embedded computer](https://www.fierceelectronics.com/embedded/what-embedded-computer), and hardware (or the mechanical component). These three parts vary in size and configuration, as motion sensors can be customized to perform highly specific functions. For example, motion sensors can be used to activate floodlights, trigger audible alarms, activate switches, and even alert the police.

**7.2 WORKING PRINCIPLE:**

Ultrasonic sensors work by emitting sound waves at a frequency too high for humans to hear. They then wait for the sound to be reflected back, calculating distance based on the time required. This is similar to how radar measures the time it takes a radio wave to return after hitting an object

PIR sensors can detect human or animal motion. Humans emit infrared radiation. When a human comes into the sensing range of the PIR sensor, these changes in radiations are detected by the PIR sensor which causes the output of the sensor to go high

**7.2.1 RESULT AND DISCUSSION:**

By using this system we can avoid the accidents and we can calculate the number of vehicles which has passed through it .If the vehicles go behind the speed we can generate the fine and we can send mail to them about the fine. By using this system we can give safety ness to the public who are travellinginday-to-day life.

**CHAPTER - 8**

**CONCLUSION AND FUTURE WORK**

**8.1 CONCLUSION:**

Smart connected Signs for Road Safety, These conclusions and guidelines are addressed to policy makers and private companies that are willing to use innovative solutions to decrease road-related fatalities and injuries amidst populations. Both chapter stake into account the potential users of connected technologies: individual drivers ,commercial drivers, pedestrians, cyclists and motorcyclists. The task force decided to study first the potential of connected technologies in high- and middle-income countries. Indeed middle-income countries represent 72% of the World population, 80% of road traffic deaths and 47% of registered motorized vehicles, while high income countries are leaders in development of connected vehicles

**8.2 FUTURE WORK:**

Data that helps travellers plan their routes. Sensors lining highways monitor traﬃc ﬂow and weight load, warn drivers of traﬃc jams, and automatically alert the authorities about accidents. Fibre-optic cables embedded in the road detect wear and tear, and communication between vehicles and roads can improve traﬃc management. For example, rapid ﬂow technologies use artiﬁcial intelligence (AI) to manage traﬃc lights, which respond to each other and to cars. Traditional systems were pre-programmed to optimize ﬂow around peak journey times, new technologies are able to process and optimize ﬂows in real time.

**CHAPTER 9**

**APPENDICES**

**9.1 SOURCE CODE:**

#include<ESP826h> #include<PubSubh> const char\* ssid = "SB-IOT1";

const char\*

password= "sb@iot11"; String command1,command2; #deﬁneORG "bhip5y"

#deﬁneDEVICE\_TYPE "Vamsi" #deﬁneDEVICE\_ID"8500"

#deﬁne TOKEN "8500913778"

String command;

char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; char topic[] = "iot-2/cmd/home/fmt/String";

char authMethod[] = "use-token-auth"; char token[] = TOKEN;

char clientId[] = "d:" ORG ":" DEVICE\_TYPE ":" DEVICE\_ID;

// Serial.println(clientID);

#include <Wire.h>

#include <Adafruit\_SSD1306.h> #include <Adafruit\_GFX.h> #deﬁne SSD1306\_LCDHEIGHT 64

// OLED display TWI address #deﬁne OLED\_ADDR 0x3C Adafruit\_SSD1306 display(-1);

#if (SSD1306\_LCDHEIGHT != 64)

#error("Height incorrect,please ﬁx Adafruit\_SSD1306.h!");

#endif

void callback(char\* topic,byte\* payload, unsignedint payloadLength); WiFiClient wiﬁClient;

PubSubClient client(server, 1883,callback, wiﬁClient);

void setup() {

display.begin(SSD1306\_SWITCHCAPVCC, OLED\_ADDR); Serial.begin(115200);

Serial.println(); pinMode(D1,OUTPUT); wiﬁConnect(); mqttConnect();

}

void loop() {

if (!client.loop()) { mqttConnect();

}

delay(100);

}

void wiﬁConnect() {

Serial.print("Connecting to "); Serial.print(ssid); WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) { delay(500);

Serial.print(".");

}

Serial.print("nWiFi connected, IP address: ");Serial.println(WiFi.localIP());

}

void mqttConnect() {

if (!client.connected()) {

Serial.print("Reconnecting MQTT client to "); Serial.println(server); while (!client.connect(clientId, authMethod, token)) { Serial.print(".");

delay(500);

}

initManagedDevice(); Serial.println();

}

}

void initManagedDevice() { if (client.subscribe(topic)) {

Serial.println("subscribeto cmd OK");

} else {

Serial.println("subscribe to cmd FAILED");

}

}

void callback(char\* topic, byte\* payload, unsigned int payloadLength) { Serial.print("callback invokedfor topic: ");Serial.println(topic);

for (int i = 0; i < payloadLength; i++) {

// Serial.println((char)payload[i]);

command += (char)payload[i];

}

Serial.println(command); command1=getValue(command,',',0); command2=getValue(command,',',1); if(command1=="1"){ display.clearDisplay();

// display a line of text display.setTextSize(1);

display.setTextColor(WHITE);

display.setCursor(0,10);

display.print(command);

// updatedisplay with all of the above graphicsdisplay.display();

}

commad="";

command1 ="";

command2="";

}

String getValue(String data, char separator, int index)

{

int found = 0;

int strIndex[] = { 0, -1 };

int maxIndex = data.length() - 1;

for (int i = 0; i <=maxIndex && found<= index; i++) {

if (data.charAt(i) == separator || i == maxIndex)

{

found++;

strIndex[0] = strIndex[1] + 1;

strIndex[1] = (i == maxIndex) ? i+1 : i;

}

}

return found > index ? data.substring(strIndex[0], strIndex[1]) : "";

}

//DATABASE SCHEMA

#include <ESP8266WiFi.h> #include <PubSubClient.h>

const char\* ssid = “SB-IOT1”;

const char\* password= “sb@iot11”; String command1,command2; #deﬁneORG “bhip5y”

#deﬁneDEVICE\_TYPE “Vamsi” #deﬁneDEVICE\_ID “8500”

#deﬁne TOKEN “8500913778”

String command;

char server[] = ORG “.messaging.internetofthings.ibmcloud.com”;

char topic[] = “iot-2/cmd/home/fmt/String”;

char authMethod[] = “use-token-auth”; char token[] = TOKEN;

char clientId[] = “d:” ORG “:” DEVICE\_TYPE “:” DEVICE\_ID;

/ /Serial.println(clientID); #include <Wire.h>

#include <Adafruit\_SSD1306.h> #include <Adafruit\_GFX.h> #deﬁne SSD1306\_LCDHEIGHT 64

/ OLED display TWI address #deﬁne OLED\_ADDR 0x3C Adafruit\_SSD1306 display(-1); #if (SSD1306\_LCDHEIGHT != 64)

#error(“Height incorrect, please ﬁx Adafruit\_SSD1306.h!”);

#endif

void callback(char\* topic,byte\* payload, unsignedint payloadLength); WiFiClient wiﬁClient;

PubSubClient client(server, 1883,callback, wiﬁClient); void setup() {

display.begin(SSD1306\_SWITCHCAPVCC, OLED\_ADDR); Serial.begin(115200);

Serial.println();

pinMode(D1,OUTPUT);

wiﬁConnect();

mqttConnect();

}

void loop() {

if (!client.loop()) { mqttConnect();

}

delay(100);

}

void wiﬁConnect() {

Serial.print(“Connecting to “); Serial.print(ssid); WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(“.”);

}

Serial.print(“nWiFi connected, IP address: “);Serial.println(WiFi.localIP());

}

void mqttConnect() {

if (!client.connected()) {

Serial.print(“Reconnecting MQTT client to “); Serial.println(server); while (!client.connect(clientId, authMethod, token)) { Serial.print(“.”);

delay(500);

}

initManagedDevice();

Serial.println();

}

}

void initManagedDevice() { if (client.subscribe(topic)) {

Serial.println(“subscribeto cmd OK”);

} else {

Serial.println(“subscribe to cmd FAILED”);

}

}

void callback(char\* topic, byte\* payload, unsigned int payloadLength) { Serial.print(“callback invokedfor topic: “);Serial.println(topic);

for (int I = 0; I < payloadLength; i++) {

/ Serial.println((char)payload[i]); command += (char)payload[i];

}

Serial.println(command); command1=getValue(command,’,’,0); command2=getValue(command,’,’,1); if(command1==”3”){ display.clearDisplay();

/ display a line of text display.setTextSize(1); display.setTextColor(WHITE); display.setCursor(0,10); display.print(command2);

// updatedisplay with all of the above graphicsdisplay.display();

}

command =””;

command1 =””;

command2=””;

}

String getValue(String data, char separator, int index)

{

int found = 0;

int strIndex[] = { 0, -1 };

int maxIndex = data.length() – 1;

for (int I = 0; I <=maxIndex && found<= index; i++) { if (data.charAt(i) == separator || I == maxIndex) {

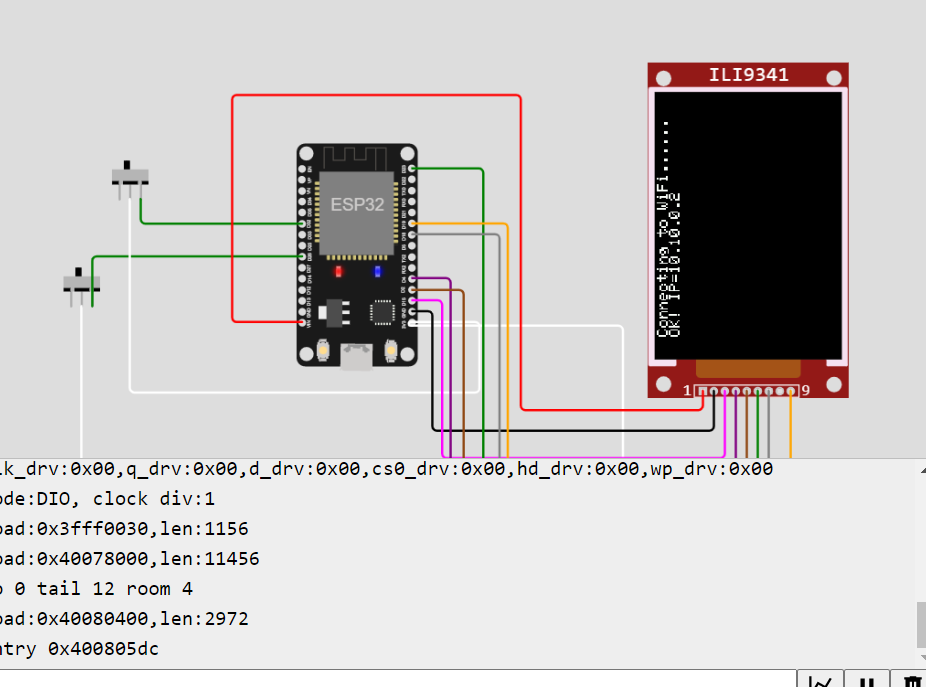
found++;

strIndex[0] = strIndex[1] + 1; strIndex[1] = (I == maxIndex) ? i+1 : I;

}

}return found > index ? data.substring(strIndex[0], strIndex[1]) : “”;

**9.2 SCREEN SHOTS:**

****

**CHAPTER 10**

**9 REFEREENCE:**

<https://www.researchgate.net/publication/328130229_Smart_Board>

<https://www.researchgate.net/publication/305365822_Poster_A_Step_Towards_Smart_Traffic_Sign_Board_by_Smart_Devices>

<https://www.researchgate.net/publication/336355855_Smart_Traffic_Sign_Boards_STSB_for_Smart_Cities>

[https://drive.google.com/file/d/1Gzr5TUW5Jb5VOSXWdFZqpFOqMo\_b3V3k/v iew?usp=share\_link](https://drive.google.com/file/d/1Gzr5TUW5Jb5VOSXWdFZqpFOqMo_b3V3k/v%20iew?usp=share_link)

<https://drive.google.com/file/d/1SroGSQaz9K0m3HOaj1ofkohoFaelSfZI/view?usp=share_link>

<https://docs.oracle.com/en/cloud/paas/app-container-cloud/python-oracle-accs/>

<https://cloud.ibm.com/docs/cloud-foundry-public?topic=cloud-foundry-public-getting-started-python>