

# SMART CONNECTED SIGNS FOR IMPROVED ROAD SAFETY

## **Introduction**

### **a)Overview:**

Roads are integral part of human civilization. They are the nervous system of our country, hence they are being laid on hill sides, narrow ridges which is a major hazard to human life. As roads play crucial role in daily routine this can be modelled in smart manner to serve us with enhanced capabilities. 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, in we can change the road signs accordingly if they are digitalized. This project purpose a system which has digital signs can be changed dynamically. If there is a 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 road diversions, accident prone areas and the information sign boards can be entered through web app. This data is retrived and displayed on the signboards.

## **Literature:**

### **a).Existing problem**

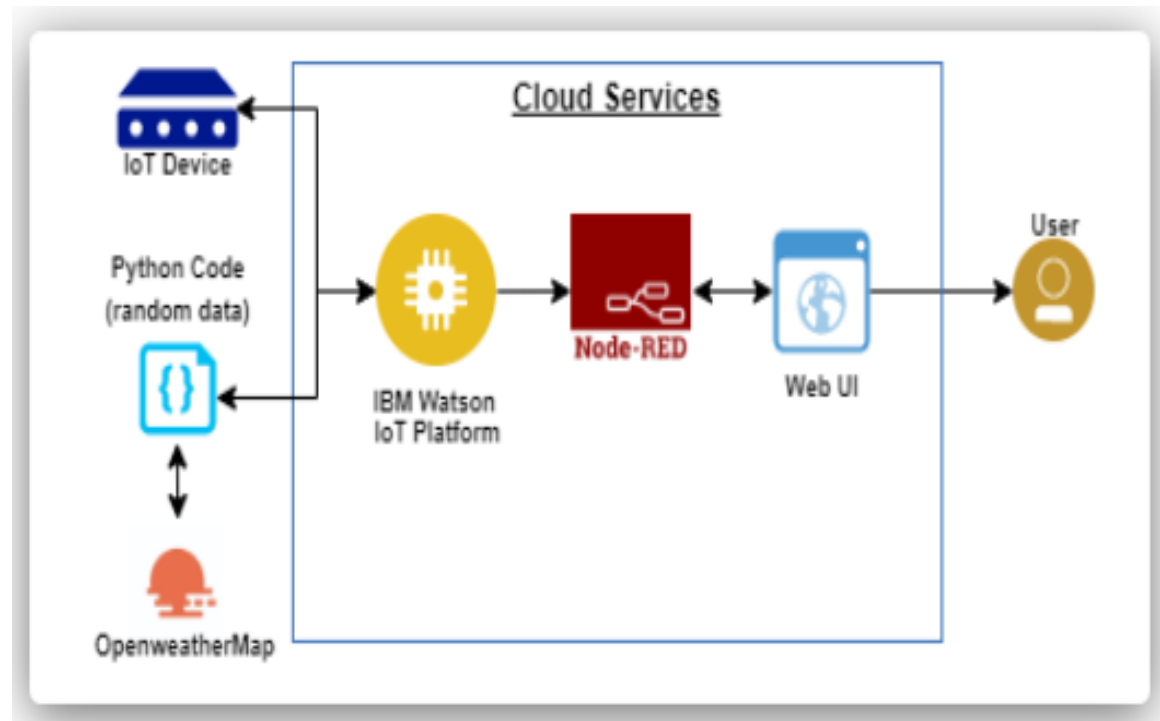
Roads are the integral part of human civilization. Mostly all the roards are displayed with static signs to limit the speed of vehicle and diversion signs. But every day and every hour weather, traffic changes with time . Static sign boards doesn't change with time ,so if there is a rain or wind the person who drives thinks that it is a safety speed but it causes the accidents due to slippery roads caused by the rains.

### **b).Proposed solution**

In this project static boards are replaced by dynamically digitalized boards. In this we used open weather map to get weather Api. We got the current weather report like windspeed, humidity, clouds etc from open weather based on that we are changing the speed limit using nodered in a particular area. With help of sensor we will get to know traffic information based on that information we can change our way this we will save our time. By using digitalized boards we can display safety speed of the vehicle based on weather conditions. This prevents the accidents and saves the persons life. We will alert the driver if there is school zone or hospital to reduce their speed.

## Theoretical Analysis

### Block Diagram



### Software designing

Firstly, created account in IBM cloud. Then created the Node-Red application and created the account in openweathermap then generated api key .With help of importing requests in python we can extracted the weather data by using URL and APIKEY . IBM device should be created then it should be launched by using ibm iot watson . Then use this credentials to import the required values into ibm cloud. Then made the flows for required output with help of function nodes.To export the values to python code we used IBMout node. For providing information in web ui we used http nodes i.e http in, http out.Output variables are made into global variabes to use in http node flow with help of global.set. To display the variables or strings in web page we get values by using global.get in the function node of http flow.By appending /data and /ui in the url we get data and pictorial representation respectively.

### Experimental Investigation

Intially started with python code as hardware is not present.We imported the data into ibm cloud by using ibm iot device.

cloud.ibm.com login - Bing | IBM Cloud | Node-RED: amulya123.eu-gb.m... | https://amulya123.eu-gb.mybluemix.net/red/#flow/48008f33.138b8

Node-RED

Flow 2

debug

humidity: 35  
windspeed: 5.91  
clouds: "overcast clouds"  
traffic: 93  
accident: 1  
zones: 1

6/22/2021, 5:31:50 AM node: ccecbdbd07f48  
iot-2/type/externshipid/9876/ev/status/fmt/json :  
msg.payload: Object

▼ object  
humidity: 35  
windspeed: 5.91  
clouds: "overcast clouds"  
traffic: 35  
accident: 1  
zones: 3

6/22/2021, 5:31:52 AM node: ccecbdbd07f48  
iot-2/type/externshipid/9876/ev/status/fmt/json :  
msg.payload: Object

▶ { humidity: 35, windspeed: 5.91, clouds: "overcast clouds", traffic: 36, accident: 0 - }

6/22/2021, 5:31:55 AM node: ccecbdbd07f48

cloud.ibm.com login - Bing | Application Details - IBM Cloud | Node-RED: amulya123.eu-gb.m... | https://amulya123.eu-gb.mybluemix.net/red/#flow/48008f33.138b8

Node-RED

Flow 2

Edit function node

Name: vechiles-count

Setup | On Start | On Message | On Stop

```
1 global.set('t',global)
2 p=msg.payload.traffic
3 if(p>=70){
4   global.p={"overtraffic":"better choose other way" }
5   msg.payload={"overtraffic":"better choose other way" }
6 } else if(p>=40){
7   global.p={"moderatetraffic":"you can try" }
8   msg.payload={"moderatetraffic":"you can try"}
9 } else{
10  global.p={"No traffic" : "best way"}
11  msg.payload={"No trafic":"best way"}
12 }
13 return msg;
```

Node: "322c1698.45178a"  
Type: function

Pressing enter will edit the first node in the current selection

29°C Haze 7:49 AM 6/23/2021

cloud.ibm.com login - Bing | Application Details - IBM Cloud | Node-RED: amulya123.eu-gb.mybluemix.net

https://amulya123.eu-gb.mybluemix.net/red/#flow/48008f33.138b8

### Node-RED

filter nodes

Flow 2

common

- inject
- debug
- complete
- catch
- status
- link in
- link out
- comment

function

- function
- switch
- change

IBM IoT

connected

#### Edit function node

Delete Cancel Done

Properties

Name fatal situation

Setup On Start On Message On Stop

```
1 global.set("a",global)
2 accident=msg.payload.accident
3 if(accident==1){
4   global.a={"accident prone area": "Drive carefully" }
5   msg.payload={"accident prone area": "Drive carefully" }
6 }else{
7   global.a={"msg":"happy journey"}
8   msg.payload={"msg":"happy journey"}
9 }
10
11
12 return msg;
```

Enabled

info

Search flows

Flows

- Flow 2

Subflows

Global Configuration Nodes

fatal situation

Node "7bf4590a.f84598"

Type function

show more

Search for nodes using **ctrl-f**

29°C Haze 7:50 AM 6/23/2021

cloud.ibm.com login - Bing | Application Details - IBM Cloud | Node-RED: amulya123.eu-gb.mybluemix.net

https://amulya123.eu-gb.mybluemix.net/red/#flow/48008f33.138b8

### Node-RED

filter nodes

Flow 2

common

- inject
- debug
- complete
- catch
- status
- link in
- link out
- comment

function

- function
- switch
- change

IBM IoT

connected

#### Edit function node

Delete Cancel Done

Properties

Name zones

Setup On Start On Message On Stop

```
1 global.set("z",global)
2 p=msg.payload.zones
3 if(p==1){
4   global.z={"schoolzone": "go slow" }
5   msg.payload={"schoolzone": "go slow" }
6 }else if(p==2){
7   global.z={"hospital": "hospital is near by place"}
8   msg.payload={"hospital": "hospital is near by place"}
9 }else if(p==3){
10    global.z={"reasturant": "come and enjoy the food"}
11    msg.payload={"reasturant": "come and enjoy the food"}
12 }else{
13    global.z= {"road": "don't drink and drive"}
14    msg.payload={"road": "don't drink and drive"}
15 }
16 return msg;
```

Enabled

info

Search flows

Flows

- Flow 2

Subflows

Global Configuration Nodes

zones

Node "99d78c9e.20ca3"

Type function

show more

Import a flow by dragging its JSON into the editor, or with **ctrl-i**

29°C Haze 7:51 AM 6/23/2021

cloud.ibm.com login - Bing | Application Details - IBM Cloud | Node-RED: amulya123.eu-gb.mybluemix.net | +

https://amulya123.eu-gb.mybluemix.net/red/#flow/48008f33.138b8

### Node-RED

filter nodes | Flow 2

**common**

- inject
- debug
- complete
- catch
- status
- link in
- link out
- comment

**function**

- function
- switch
- change

**IBM IoT** | connected

**Edit function node**

Delete | Cancel | Done

**Properties**

Name: vechiles-count

Setup | On Start | **On Message** | On Stop

```
1 global.set('t',global)
2 p=msg.payload.traffic
3 if(p>=70){
4   global.p={"overtraffic":"better choose other way"}
5   msg.payload={"overtraffic":"better choose other way"}
6 }else if(p>=40){
7   global.p={"moderatetraffic":"you can try"}
8   msg.payload={"moderatetraffic":"you can try"}
9 }else{
10  global.p={"No traffic":"best way"}
11  msg.payload={"No traffic":"best way"}
12 }
13 return msg;
```

☐ Enabled

**info**

Search flows

Flows

- Flow 2

Subflows

Global Configuration Nodes

**vechiles-count**

Node: "322c1698.45178a"

Type: function

show more

Pressing **enter** will edit the first node in the current selection

29°C Haze | 7:49 AM | 6/23/2021

cloud.ibm.com login - Bing | Application Details - IBM Cloud | Node-RED: amulya123.eu-gb.mybluemix.net | +

https://amulya123.eu-gb.mybluemix.net/red/#flow/48008f33.138b8

### Node-RED

filter nodes | Flow 2

**common**

- inject
- debug
- complete
- catch
- status
- link in
- link out
- comment

**function**

- function
- switch
- change

**IBM IoT** | connected

**Edit function node**

Delete | Cancel | Done

**Properties**

Name: weboutput

Setup | On Start | **On Message** | On Stop

```
1 msg.payload={"traffic alert":global.get('s'),"traffic information":global.get('t')}
2 return msg;
```

☐ Enabled

**info**

Search flows

Flows

- Flow 2

Subflows

Global Configuration Nodes

**weboutput**

Node: "9c2d967e.dcb38"

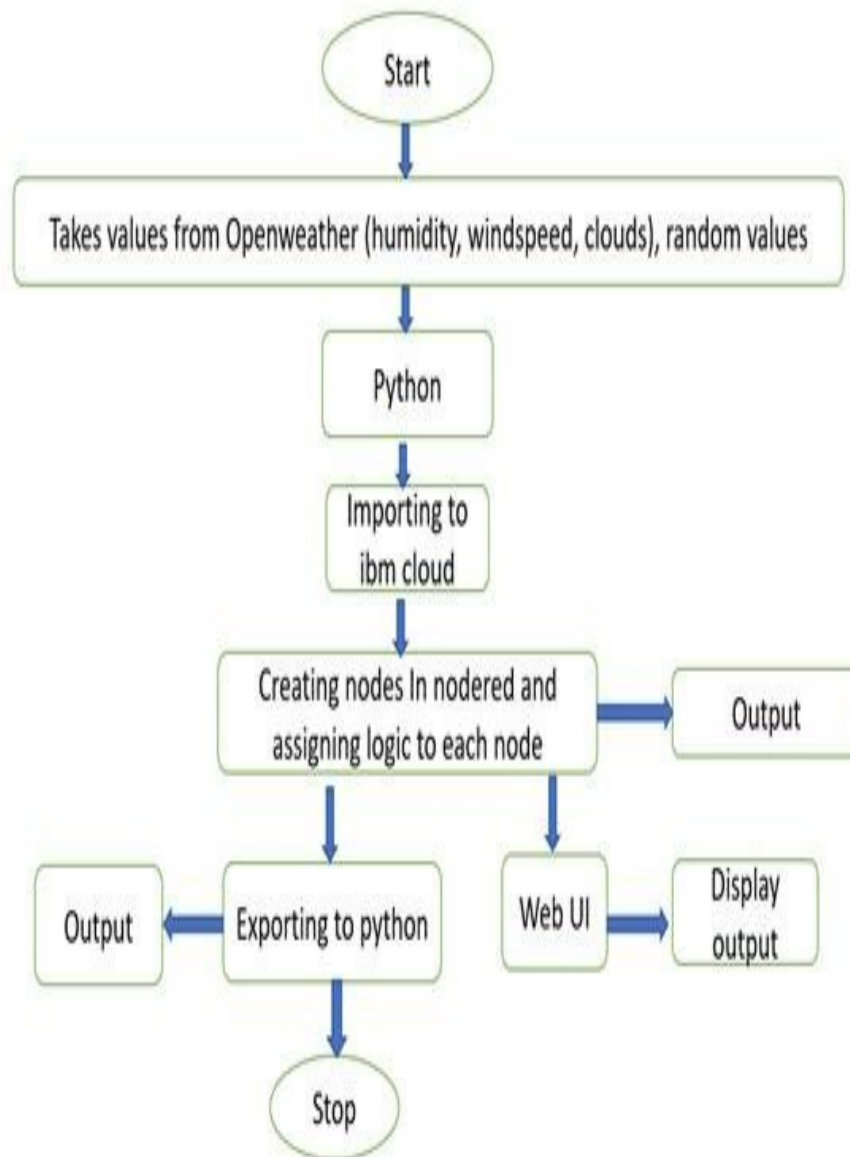
Type: function

show more

You can confirm your changes in the node edit tray with **ctrl+enter** or cancel them with **ctrl+escape**

29°C Haze | 7:53 AM | 6/23/2021

## Flow chart



## Results:

### python output

```
Python Shell 3.9.5*
File Edit Shell Debug Options Window Help

{'hospital': 'hospital is near by place'}
{'msg': 'happy journey'}
{'overtraffic': 'better choose other way'}
{'maximumspeed': 20}
{'schoolzone': 'go slow'}
{'msg': 'happy journey'}
{'moderatetraffic': 'you can try'}
{'maximumspeed': 20}
{'road': 'don't drink and drive'}
{'msg': 'happy journey'}
{'No traffic': 'best way'}
{'maximumspeed': 20}
{'schoolzone': 'go slow'}
{'msg': 'happy journey'}
{'No traffic': 'best way'}
{'maximumspeed': 20}
{'reasturant': 'come and enjoy the food'}
{'accident prone area': 'Drive carefully'}
{'No traffic': 'best way'}
{'maximumspeed': 20}
{'road': 'don't drink and drive'}
{'msg': 'happy journey'}
{'No traffic': 'best way'}
{'maximumspeed': 20}
{'schoolzone': 'go slow'}
{'accident prone area': 'Drive carefully'}
{'overtraffic': 'better choose other way'}
{'maximumspeed': 20}
{'hospital': 'hospital is near by place'}
{'msg': 'happy journey'}
{'overtraffic': 'better choose other way'}
{'maximumspeed': 20}
{'reasturant': 'come and enjoy the food'}
{'msg': 'happy journey'}
{'No traffic': 'best way'}
{'maximumspeed': 20}
{'schoolzone': 'go slow'}
{'msg': 'happy journey'}
{'overtraffic': 'better choose other way'}
{'maximumspeed': 20}
```

### Node-Red output

Node-RED interface showing a flow for processing IoT data. The flow starts with an IBM IoT node (connected) feeding into a function node (f) that splits the data into four parallel paths: zones, fatal situation, vehicles-count, and weather-input. These paths then feed into a function node (f) that outputs four parallel streams: location information, fatal information, traffic information, and traffic alert. The bottom path of the flow is a sequence of [get]/data, weboutput, and http nodes.

The debug console shows the following log entries:

```
6/22/2021, 10:42:30 AM node: 8e1a15c6 24e5e8
iot-2/type/externshipid/9876/ev/status/fmt/json :
msg.payload: Object
> { accident prone area: "Drive carefully" }


6/22/2021, 10:42:30 AM node: 8e1a15c6 24e5e8
iot-2/type/externshipid/9876/ev/status/fmt/json :
msg.payload: Object
> { overtraffic: "better choose other way" }

6/22/2021, 10:42:30 AM node: 8e1a15c6 24e5e8
iot-2/type/externshipid/9876/ev/status/fmt/json :
msg.payload: Object
> { maximumspeed: 20 }

6/22/2021, 10:42:30 AM node: 8e1a15c6 24e5e8
iot-2/type/externshipid/9876/ev/status/fmt/json :
msg.payload: Object
> { reasturant: "come and enjoy the food" }

6/22/2021, 10:42:30 AM node: 8e1a15c6 24e5e8
iot-2/type/externshipid/9876/ev/status/fmt/json :
msg.payload: Object
> { accident prone area: "Drive carefully" }
```

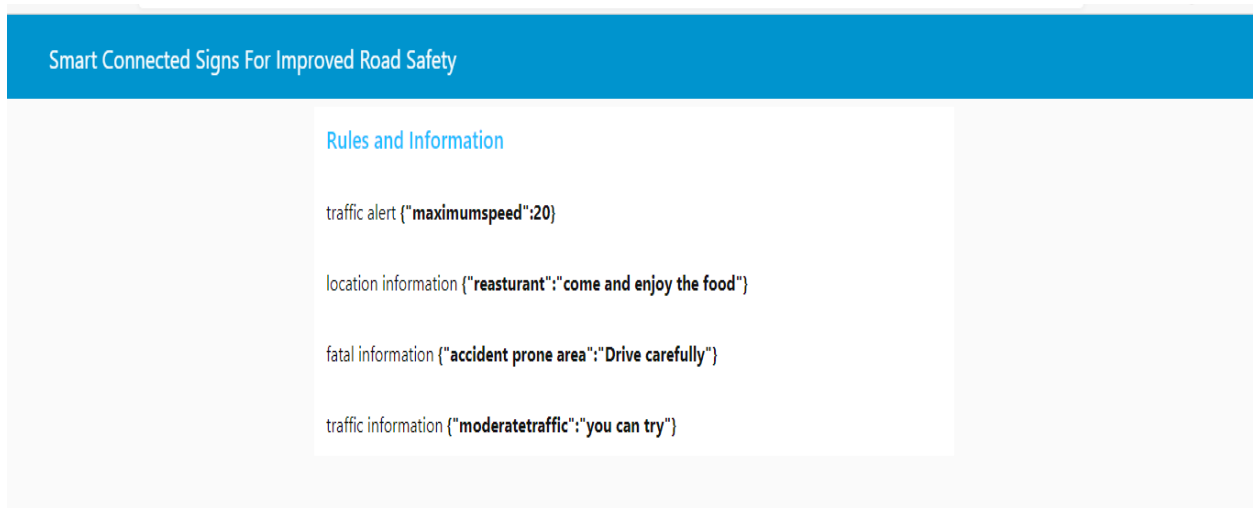
## web data output



The screenshot shows a web browser with multiple tabs. The active tab is a REST client interface for the URL `https://amulya123.eu-gb.mybluemix.net/data`. The response body contains the following JSON data:

```
{
  "traffic alert": {
    "speedlimit": {
      "maximum speed": 20
    }
  },
  "traffic information": {
    "p": {
      "overttraffic": "better choose other way"
    },
    "suggestion": {
      "a": {
        "msg": "happy journey"
      }
    },
    "zones": {
      "z": {
        "schoolzone": "go slow"
      }
    }
  }
}
```

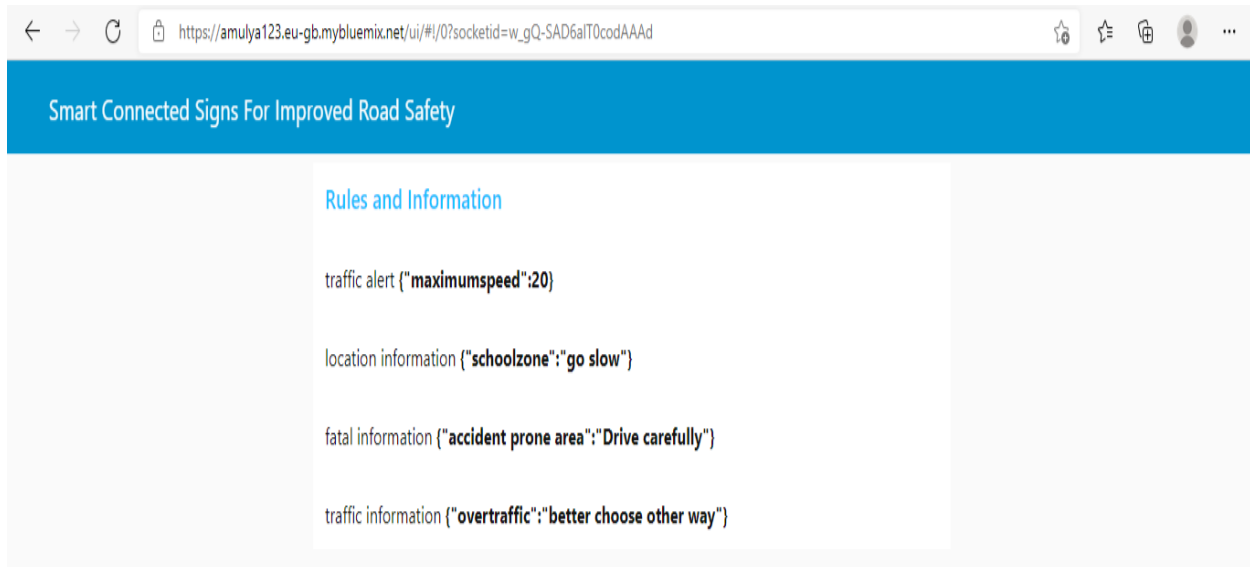
## Web ui output



The screenshot displays a web application interface with a blue header bar containing the text "Smart Connected Signs For Improved Road Safety". Below the header, there is a section titled "Rules and Information" in blue text. This section contains four lines of JSON-like data, each representing a different type of traffic information:

- traffic alert {"maximumspeed":20}
- location information {"reasturant":"come and enjoy the food"}
- fatal information {"accident prone area":"Drive carefully"}
- traffic information {"moderatetraffic":"you can try"}





## Advantages

- Message boards can contribute to safer conditions on the road for drivers.
- It provides real-time traffic information so that drivers can adjust their routes.
- Accidents or harsh conditions are also displayed so that it encourages drivers to slow down and keep an eye out for hazards on the road which results in safer conditions for everyone.
- It suggests the safe speed for the drivers according to climatic conditions.

## Applications

- Smart city

## Conclusion

This project integrates with openweather, traffic and displays the instructions for safe journey. This boards helps to reduce road accidents and also reduces traffic by displaying traffic information. It gives the information about schools, hospitals, reasturants for new visitors. These are designed to be used to inform drivers, so they are made with durability ,reliability and visibility in mind.

## Future Scope

In future we can implement this in smart cities for saving lives and time.

## Bibiliography

- Bilal Ghazal, Khaled ElKhatib, Khaled Chahine, Mohamad Kherfan," Smart traffic

light control systems", IEEE 2016

- Abishek C, Mukul Kumar and Kumar Padmandh "City Traffic Congestion Control in Indian Scenario using wireless sensors network"

## Appendix:

### a.Source code

File Edit Format Run Options Window Help

```
import wiotp.sdk.device
import time
import requests, json
import random
myConfig = {
    "identity": {
        "orgId": "c2f7f6",
        "typeId": "externship",
        "deviceId": "9876"
    },
    "auth": {
        "token": "12345678"
    }
}
def myCommandCallback(cmd):
    print(" %s" % cmd.data)
client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()
while True:
    BASE_URL = "https://api.openweathermap.org/data/2.5/weather?"
    CITY = "nalgonda"
    API_KEY = "84e0ea50e8626893742002b8319d7ec3"
    URL = BASE_URL + "q=" + CITY + "&appid=" + API_KEY
    response = requests.get(URL)
    if response.status_code == 200:
        data = response.json()
        main = data['main']
        p = data['wind']
        humidity = main['humidity']
        windspeed = p['speed']
        report = data['weather']
        vehicles = random.randint(0,100)
        accident= random.randint(0,1)
        zones= random.randint(0,3)
        myData={'humidity':humidity, 'windspeed':windspeed,'clouds':report[0]['description'],'traffic':vehicles,'accident':accident,'zones':zones}
        client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
        client.commandCallback = myCommandCallback
        time.sleep(2)
client.disconnect()
```

### b) UI OUTPUT

## Smart Connected Signs For Improved Road Safety

### Rules and Information

traffic alert {"maximumspeed":40}

location information {"hospital":"hospital is near by place"}

fatal information {"msg":"happy journey"}

traffic information {"No traffic":"best way"}