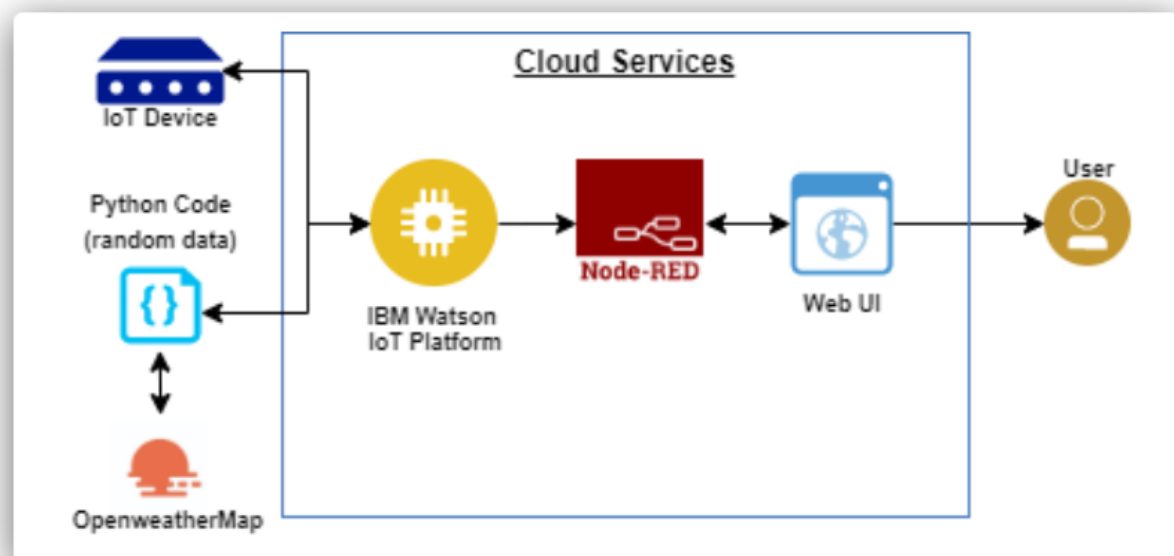


PROBLEM:

## Smart Connected Signs For Improved Road Safety Using IBM Cloud

### Features:

- 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 updates 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.
- Different modes of operations can be selected with the help of buttons



SOLUTION:

Python Program:

```
import wiotp.sdk.device
import time
import random
import requests, json
```

```
myConfig = {
    "identity": {
        "orgId": "I0lyjx",
        "typeId": "AApp1",
        "deviceId": "12345"
    },
    "auth": {
```

```

    "token": "12345678"
}
}

```

```

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

```

```

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

```

```

BASE_URL = "https://api.openweathermap.org/data/2.5/weather?"
CITY = "Hyderabad"
URL = BASE_URL + "q=" + "delhi" + "&appid=" + "c4aa755540f66e8c800cbfd67df6ddcb"

```

```

while True:
    response = requests.get(URL)
    if response.status_code == 200:
        data = response.json()
        main = data['main']
        temperature = main['temp']
        humidity = main['humidity']
        pressure = main['pressure']
        report = data['visibility']
        repo=random.randint(0,5)
        if repo==1:
            prt="SLOW DOWN , SCHOOL IS NEAR"
        elif repo==3:
            prt="SLOW DOWN , HOSPITAL NEARBY"
        elif repo==5:
            prt="NEED HELP, POLICE STATION NEARBY"
        else:
            prt=""
        speed=random.randint(0,150)
        if speed>=100:
            prt3="SLOW DOWN , Speed Limit Exceeded"
        elif speed>=60 and speed<100:
            prt3="Moderate Speed"
        else:
            prt3=""
        sign=random.randint(0,5)
        if sign==1:
            prt2="Right Diversion ->"
        elif sign==3:
            prt2="Left Diversion <-"
        elif sign==5:
            prt2="U Turn"
        else:

```

```

prt2=""
if temperature<=50:
    prt4="Fog Ahead, Drive Slow"
else:
    prt4="Clear Weather"

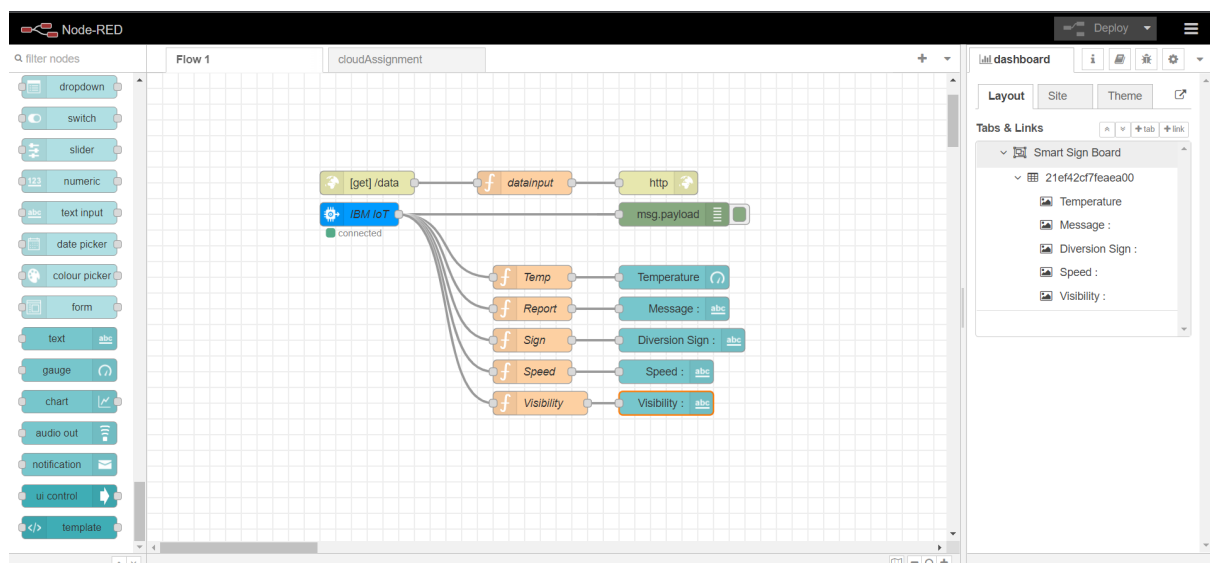
else:
    print("Error in the HTTP request")
myData={'Temperature':temperature, 'Message':prt, 'Sign':prt2, 'Speed':prt3,
'Visibility':prt4}
client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,
onPublish=None)
print("Published data Successfully: %s", myData)
client.commandCallback = myCommandCallback
time.sleep(5)
client.disconnect()

```

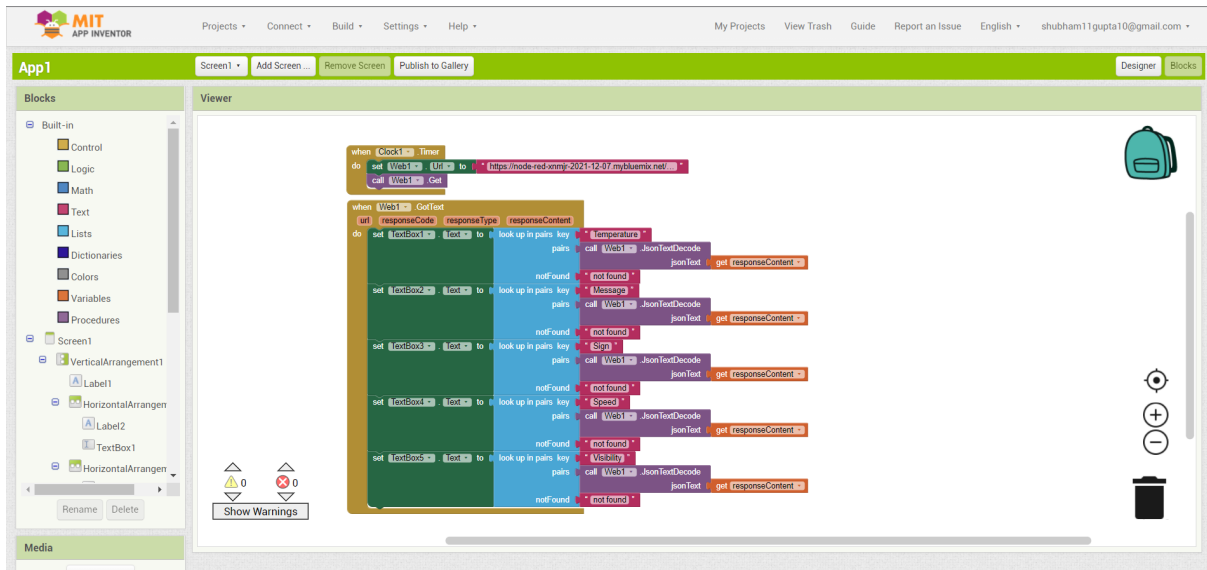
## CODE EXPLANATION-

Here we have extracted weather data from OpenWeatherMap using the API call and using the data to display the temperature and define the visibility to provide an alert. Next, we have given the readings of speed, nearby places and diversion signs and display related messages on the screen while the temperature on the gauge.

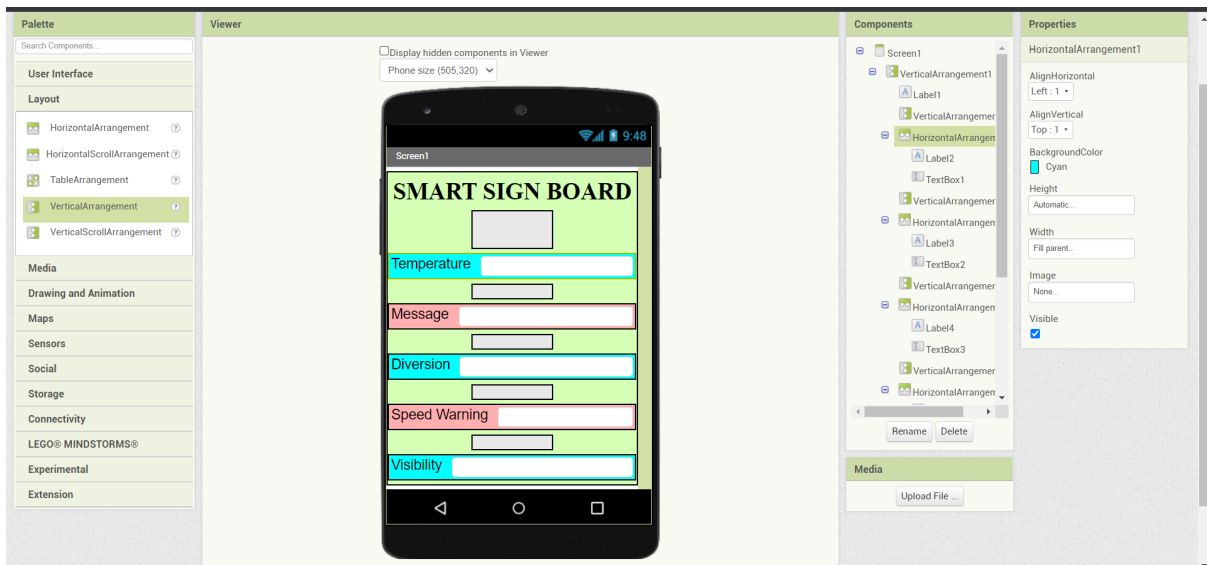
Node-Red Structure:



MIT App Structure Backend:



MIT App Structure Frontend:



Android app display output:

# SMART SIGN BOARD

Temperature 283.2

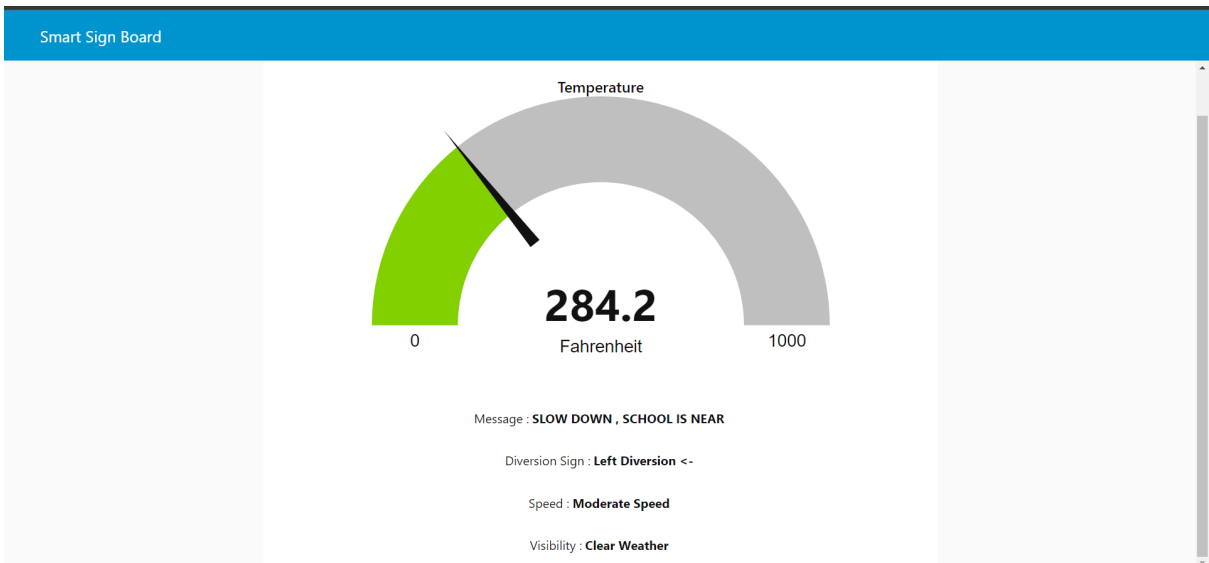
Message SLOW DOWN , SCHOOL IS

Diversion Right Diversion ->

Speed Warning SLOW DOWN , Speed

Visibility Clear Weather

Sign board display output:



IBM Cloud output:

The IBM Watson IoT Platform interface displays a list of devices. The first device is 12251, which is disconnected. The second device is 12345, which is also disconnected. The interface shows the device identity, device information, recent events, state, and logs. The recent events section shows a live stream of data coming and going from the device. The events are listed in a table with columns for Event, Value, Format, and Last Received. The events are status updates with a temperature of 284.2 and a message about a school near. The events are received in json format and are received a few seconds ago.

Event	Value	Format	Last Received
status	{"Temperature":284.2,"Message":"","Sign":"","Sp..."}	json	a few seconds ago
status	{"Temperature":284.2,"Message":"","Sign":"","Sp..."}	json	a few seconds ago
status	{"Temperature":284.2,"Message":"SLOW DOWN ..."	json	a few seconds ago
status	{"Temperature":284.2,"Message":"","Sign":"","Sp..."}	json	a few seconds ago
status	{"Temperature":284.2,"Message":"","Sign":"","Sp..."}	json	a few seconds ago

```
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\DOCS\pycode\calC.py =====
2021-12-21 02:09:22,192 wiotp.sdk.device.client.DeviceClient INFO Connected successfully: d:l0lyjx
:AApp1:12345
Published data Successfully: %s {'Temperature': 283.2, 'Message': '', 'Sign': '', 'Speed': 'Moderate Spe
ed', 'Visibility': 'Clear Weather'}
Published data Successfully: %s {'Temperature': 283.2, 'Message': '', 'Sign': 'Left Diversion <-', 'Spee
d': '', 'Visibility': 'Clear Weather'}
Published data Successfully: %s {'Temperature': 283.2, 'Message': '', 'Sign': 'U Turn', 'Speed': 'SLOW D
OWN , Speed Limit Exceeded', 'Visibility': 'Clear Weather'}
Published data Successfully: %s {'Temperature': 283.2, 'Message': '', 'Sign': '', 'Speed': '', 'Visibili
ty': 'Clear Weather'}
Published data Successfully: %s {'Temperature': 283.2, 'Message': '', 'Sign': '', 'Speed': 'SLOW DOWN ,
Speed Limit Exceeded', 'Visibility': 'Clear Weather'}
|
```