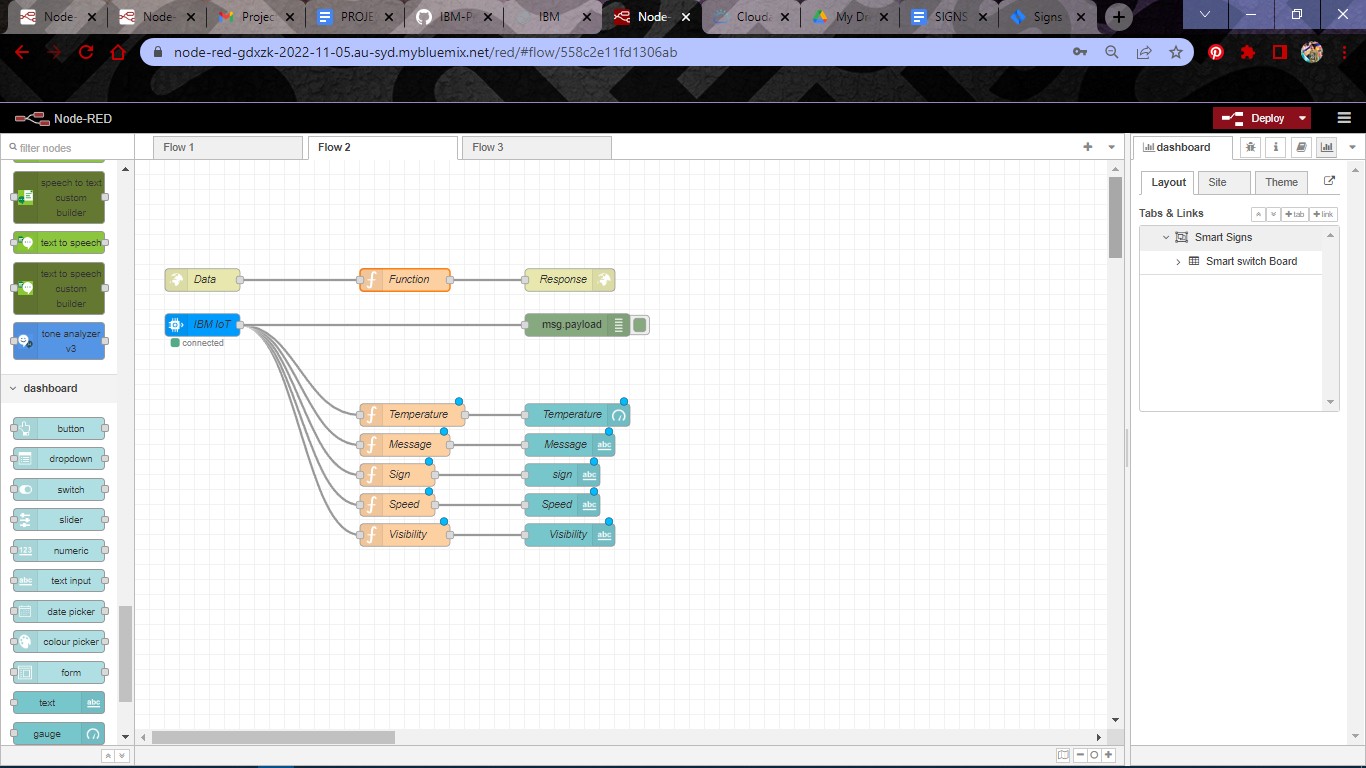
PROJECT DEVELOPMENT PHASE

Sprint-4

|  |  |
| --- | --- |
| **TEAM ID** | **PNT2022TMID277860** |
| **PROJECT NAME** | **SIGNS WITH SMART CONNECTIVITY FOR BETTER ROAD SAFETY** |

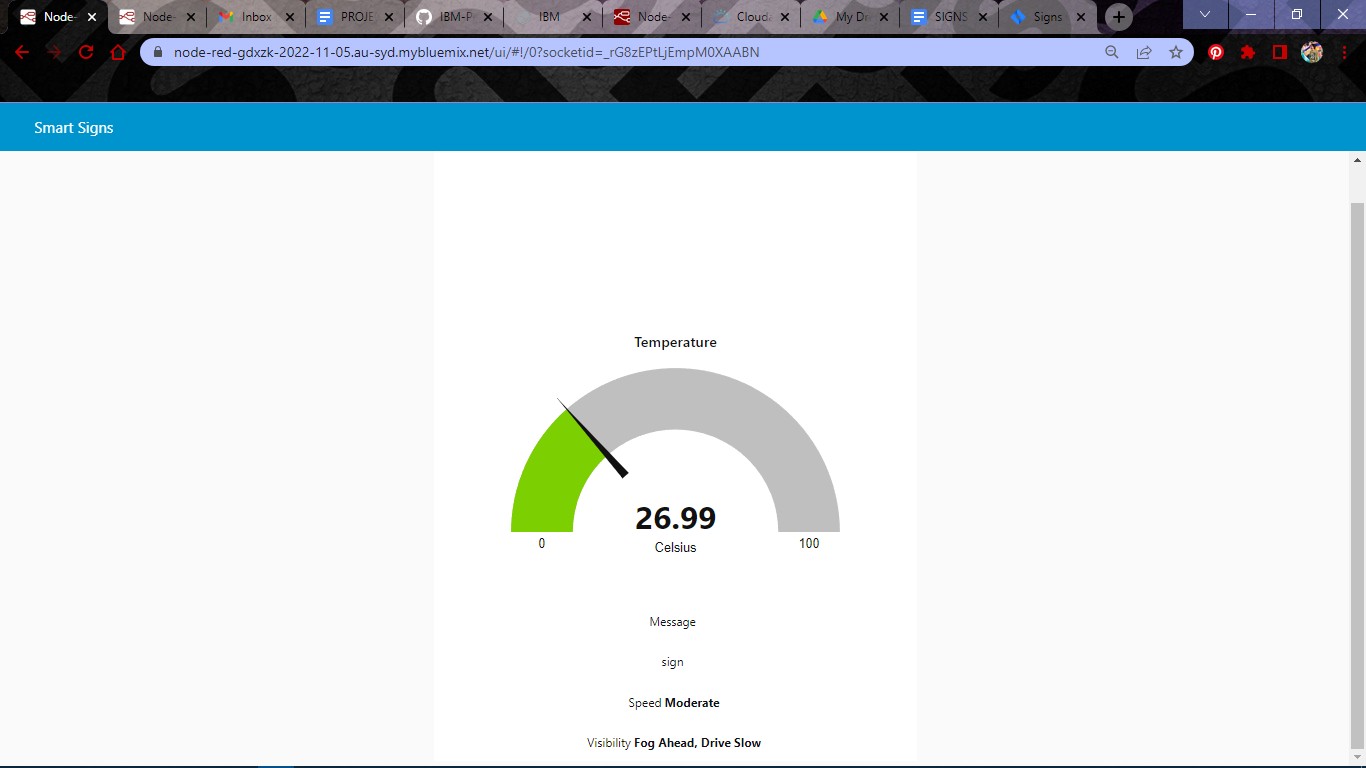
DEVELOPING ROUTE BASED ON THE PROGRAM:

Here based on the project the routing is developed by using appropriate nodes.

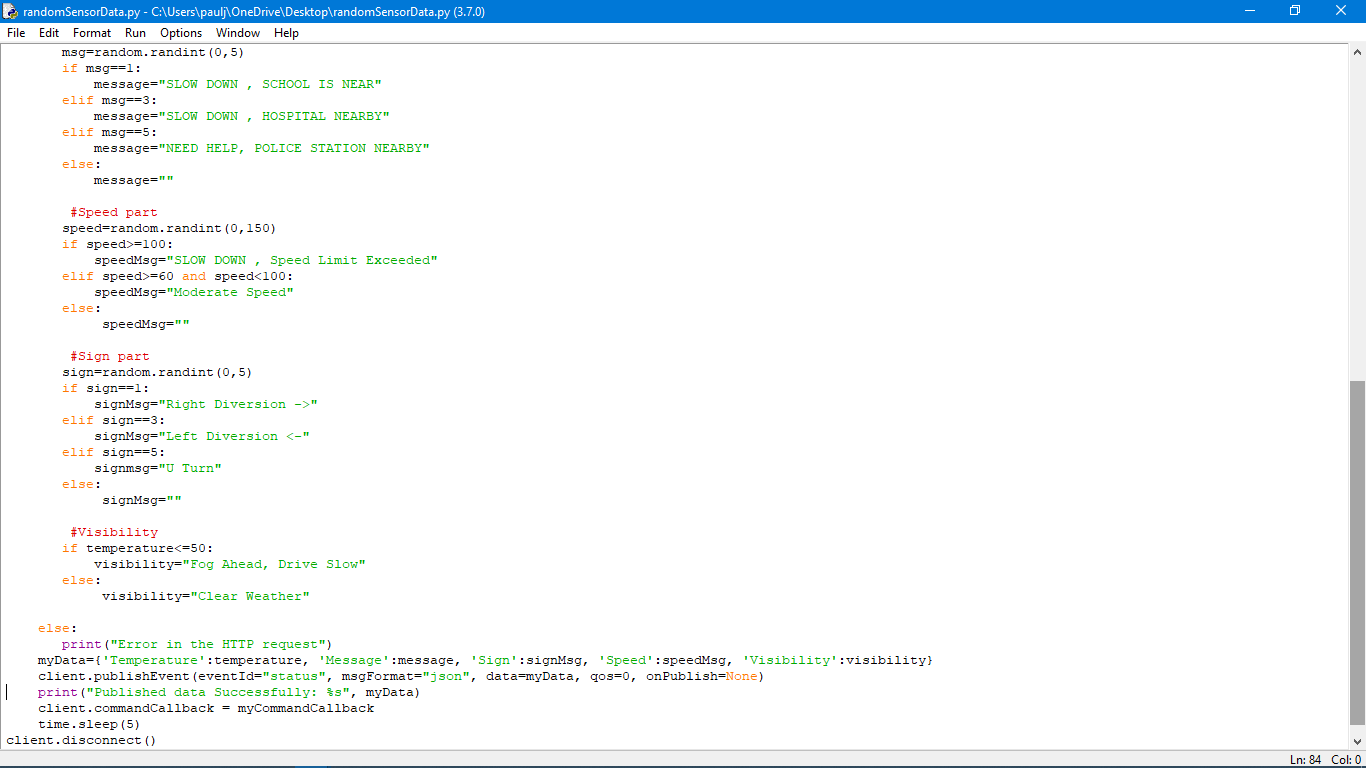


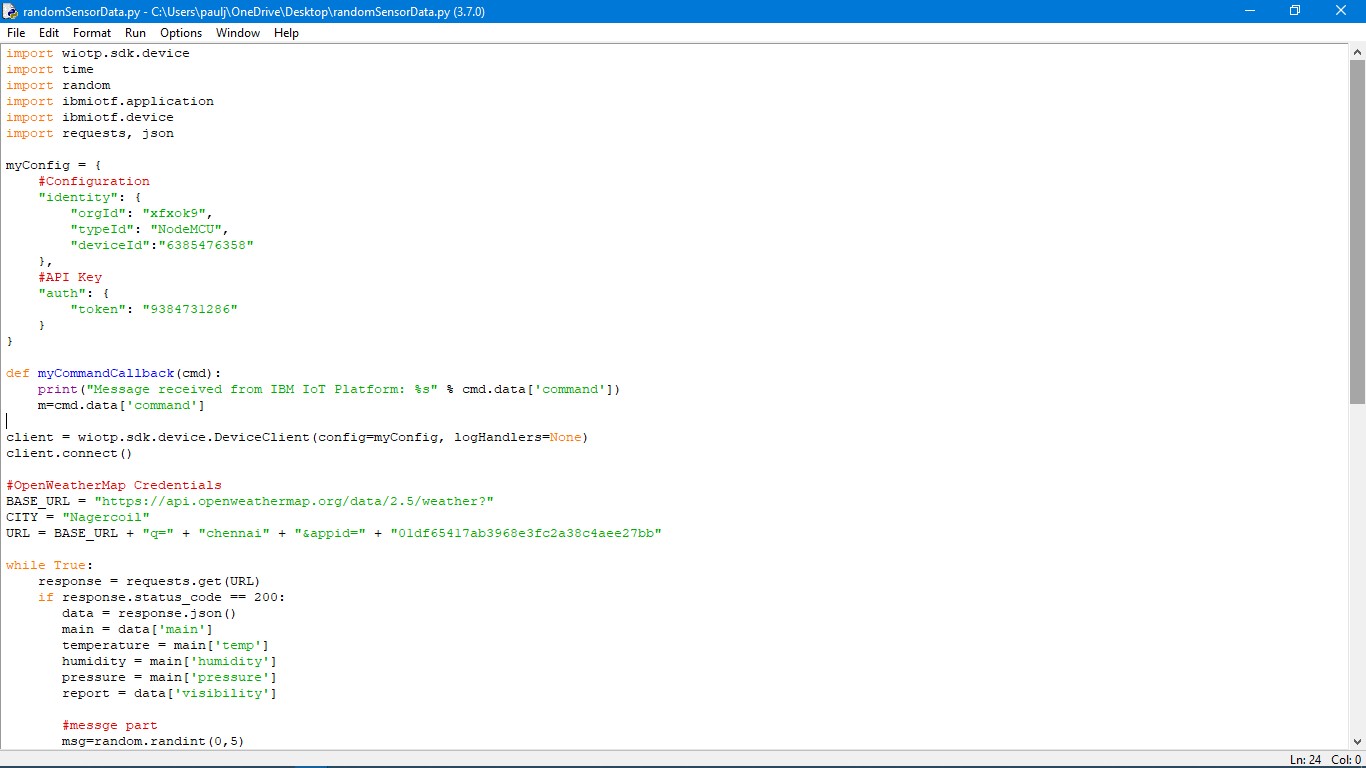
OUTPUT FOR NODE RED:

After making the proper connection between nodes the deploy button is enabled and the result is displayed on the node-red dashboard.

It shows the result in a diagrammatic structure.

**CODE IN PYTHON IDLE:**

**PROGRAM:**



**Program used in the code:**

import wiotp.sdk.device import time

import random

import ibmiotf.application import ibmiotf.device import requests, json

myConfig = { #Configuration "identity": {

"orgId": "xfxok9",

"typeId": "NodeMCU", "deviceId":"6385476358"

},

#API Key

"auth": {

"token": "9384731286"

}

}

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 = "Chennai"

URL = BASE\_URL + "q=" + "chennai" + "&appid=" + "01df65417ab3968e3fc2a38c4aee27bb"

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']

#messge part msg=random.randint(0,5) if msg==1:

message="SLOW DOWN , SCHOOL IS NEAR"

elif msg==3:

message="SLOW DOWN , HOSPITAL NEARBY"

elif msg==5:

message="NEED HELP, POLICE STATION NEARBY"

else:

message=""

#Speed part speed=random.randint(0,150) if speed>=100:

speedMsg="SLOW DOWN , Speed Limit Exceeded" elif

speed>=60 and speed<100:

speedMsg="Moderate Speed" else:

speedMsg=""

#Sign part sign=random.randint(0,5) if sign==1:

signMsg="Right Diversion ->" elif sign==3:

signMsg="Left Diversion <-" elif sign==5:

signmsg="U Turn" else:

signMsg=""

#Visibility

if temperature<=50:

visibility="Fog Ahead, Drive Slow" else:

visibility="Clear Weather"

else:

print("Error in the HTTP request") myData={'Temperature':temperature, 'Message':message,

'Sign':signMsg, 'Speed':speedMsg, 'Visibility':visibility} 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()

**Output displayed in Python Idle:**

The output of the code was displayed in python idle shell mode.