OUTPUT – SPRINT 2

DATE	04 NOV 2022
Team ID	PNT2022TMID17071
Project Tittle	Signs with smart connectivity for better road safety

main.py - C:\Users\Dell\Desktop\Project\Project Development Phase\Sprint 2\main.py (3.9.7)

File Edit Format Run Options Window Help

```
# Python code

# IMPORT SECTION STARTS

import brain

# IMPORT SECTION ENDS
# ------------------------------------------------
# USER INPUT SECTION STARTS

myLocation = "Chennai,IN"
APIKEY = "9cd610e5fd400c74212074c7ace0d62c"

localityInfo = {
    "schools" : {
        "schoolZone" : True,
        "activeTime" : ["7:00","17:30"] # schools active from 7 AM till 5:30 PM
        },
    "hospitalsNearby" : False,
    "usualSpeedLimit" : 40 # in km/hr
}

# USER INPUT SECTION ENDS
# ------------------------------------------------
# MICRO-CONTROLLER CODE STARTS

print(brain.processConditions(myLocation,APIKEY,localityInfo))

'''
MICRO CONTROLLER CODE WILL BE ADDED IN SPRINT 2 AS PER OUR PLANNED SPRINT SCHEDU
'''

# MICRO-CONTROLLER CODE ENDS
```

brain.py - C:\Users\Dell\Desktop\Project\Project Development Phase\Sprint 2\brain.py (3.9.7)

File Edit Format Run Options Window Help

```
from datetime import datetime as dt
from publishData import logData2Cloud as log2cloud
import weather

# IMPORT SECTION ENDS
# ------------------------------------------------
# UTILITY LOGIC SECTION STARTS
def processConditions(myLocation,APIKEY,localityInfo):
    weatherData = weather.get(myLocation,APIKEY)

    log2cloud(myLocation,weatherData["temperature"],weatherData["visibility"])

    finalSpeed = localityInfo["usualSpeedLimit"] if "rain" not in weatherData el
    finalSpeed = finalSpeed if weatherData["visibility"]>35 else finalSpeed/2

    if(localityInfo["hospitalsNearby"]):
        # hospital zone
        doNotHonk = True
    else:
        if(localityInfo["schools"]["schoolZone"]==False):
            # neither school nor hospital zone
            doNotHonk = False
        else:
            # school zone
            now = [dt.now().hour,dt.now().minute]
            activeTime = [list(map(int,_.split(":"))) for _ in localityInfo["sch
            doNotHonk = activeTime[0][0]<=now[0]<=activeTime[1][0] and activeTim

    return({
        "speed" : finalSpeed,
        "doNotHonk" : doNotHonk
    })

# UTILITY LOGIC SECTION ENDS
```

Ln: 1 Col: 0

publishData.py - C:\Users\Dell\Desktop\Project\Project Development Phase\Sprint 2\publish...

File Edit Format Run Options Window Help

```python
# Python code

# IMPORT SECTION STARTS

import wiotp.sdk.device # python -m pip install wiotp
import time

# IMPORT SECTION ENDS
# ---------------------------------------------
# API CONFIG SECTION STARTS

myConfig = {
    "identity" : {
        "orgId" : "f59trs",
        "typeId" : "testdevice",
        "deviceId" : "device1"
    },
    "auth" : {
        "token" : "Jrwa7c8Os2Zpq)WW18"
    }
}

# API CONFIG SECTION ENDS
# ---------------------------------------------
# FUNCTIONS SECTION STARTS

def myCommandCallback(cmd):
    print("recieved cmd : ",cmd)


def logData2Cloud(location,temperature,visibility):
    client = wiotp.sdk.device.DeviceClient(config=myConfig,logHandlers=None)
    client.connect()
    client.publishEvent(eventId="status",msgFormat="json",data={
        "temperature" : temperature,
        "visibility" : visibility,
        "location" : location
    },qos=0,onPublish=None)
    client.commandCallback = myCommandCallback
    client.disconnect()
```

Ln: 1 Col: 0

weather.py - C:\Users\Dell\Desktop\Project\Project Development Phase\Sprint 2\weather.py ...

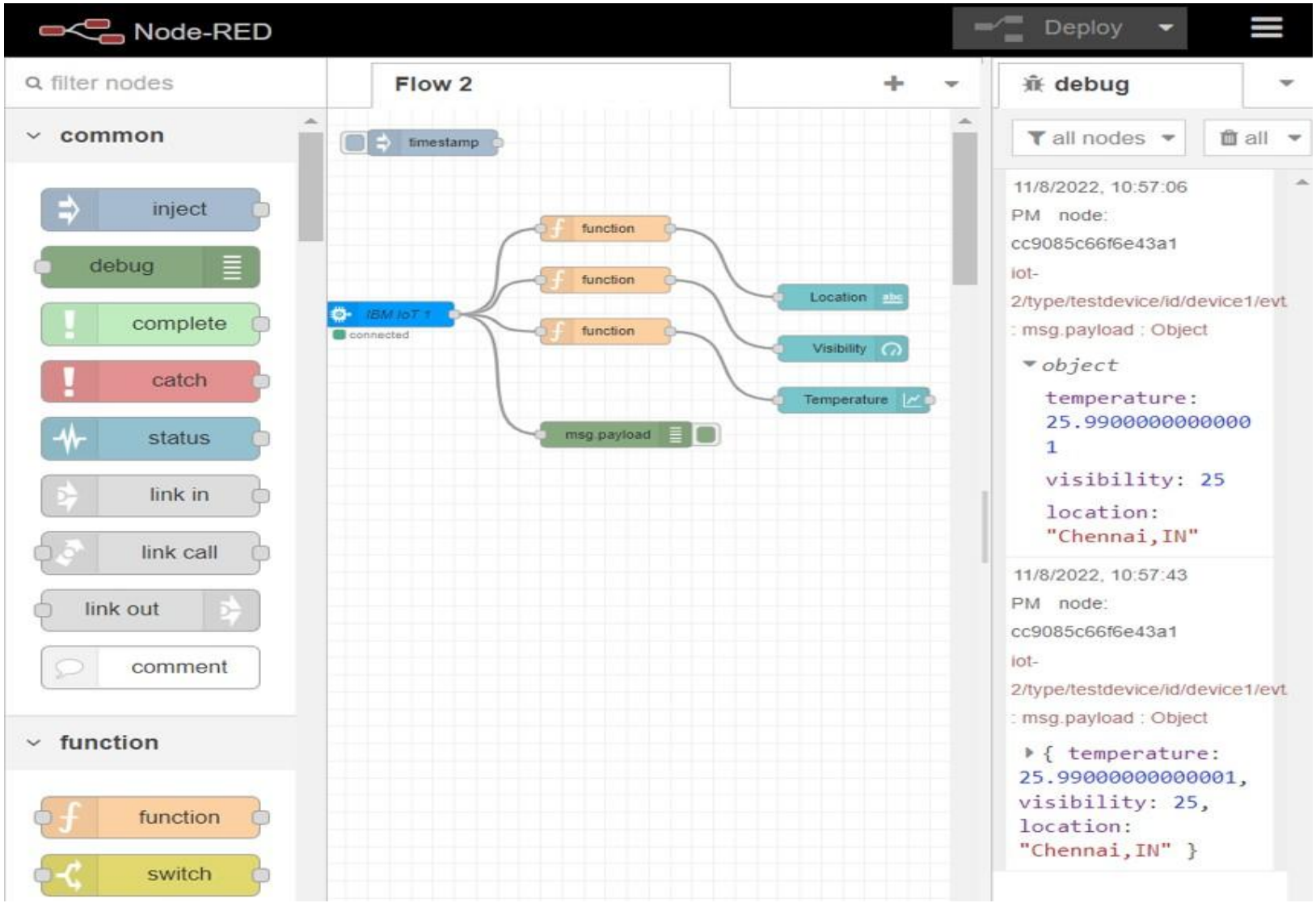
File Edit Format Run Options Window Help

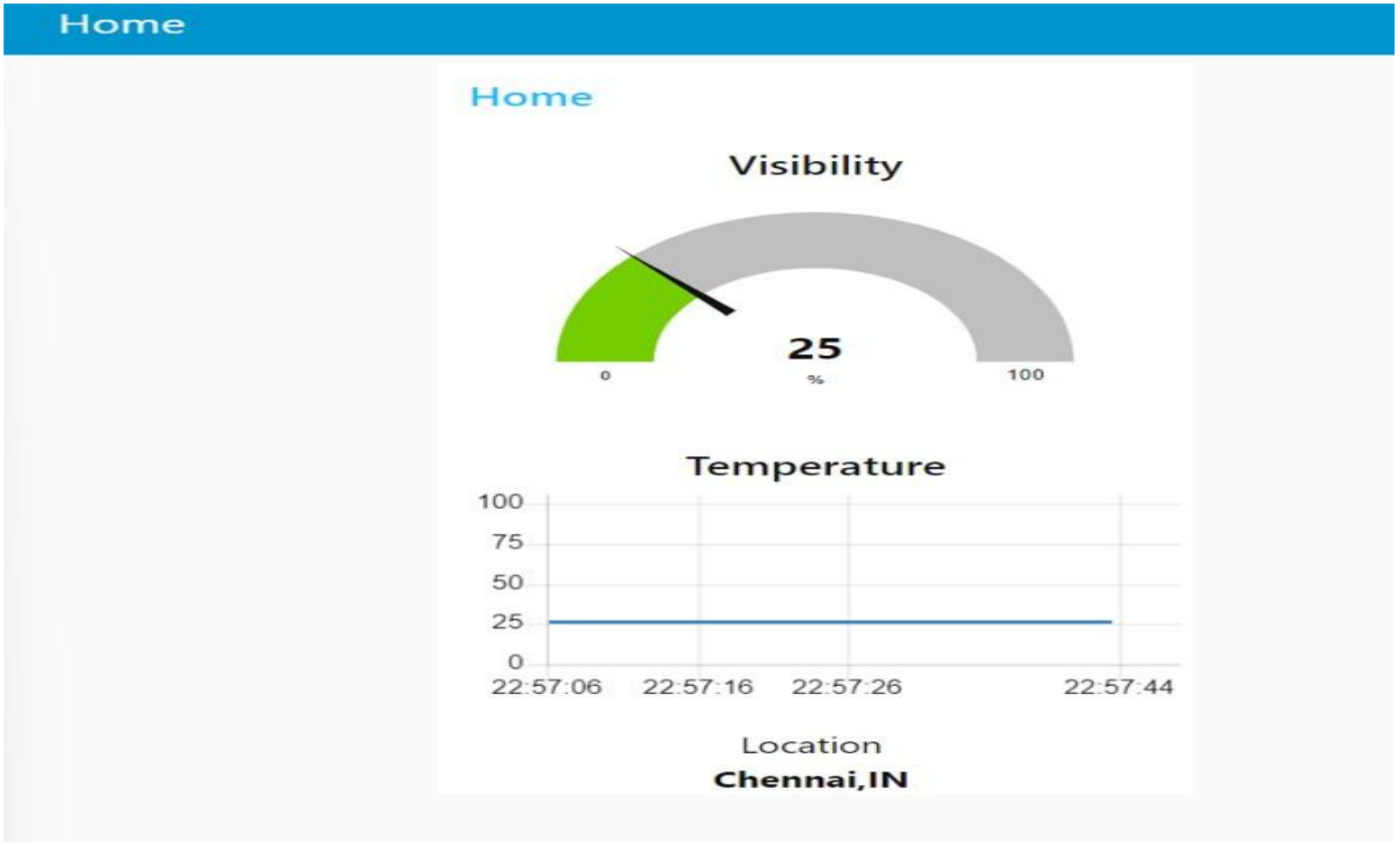
```
# Python code

import requests as reqs

def get(myLocation,APIKEY):
    apiURL = f"https://api.openweathermap.org/data/2.5/weather?q={myLocation}&ap
    responseJSON = (reqs.get(apiURL)).json()
    returnObject = {
        "temperature" : responseJSON['main']['temp'] - 273.15,
        "weather" : [responseJSON['weather'][_]['main'].lower() for _ in range(l
        "visibility" : responseJSON['visibility']/100, # visibility in percentag
    }
    if("rain" in responseJSON):
        returnObject["rain"] = [responseJSON["rain"][key] for key in responseJSO
    return(returnObject)
```

Ln: 1 Col: 0


Node-RED
Deploy
filter nodes
Flow 2
debug
common
inject
debug
complete
catch
status
link in
link call
link out
comment
function
function
switch
timestamp
IBM IoT 1
connected
function
function
function
msg payload
Location
Visibility
Temperature
all nodes
all
11/8/2022, 10:57:06 PM node: cc9085c66f6e43a1
iot-2/type/testdevice/id/device1/evt : msg.payload : Object
object
temperature: 25.990000000000001
visibility: 25
location: "Chennai,IN"
11/8/2022, 10:57:43 PM node: cc9085c66f6e43a1
iot-2/type/testdevice/id/device1/evt : msg.payload : Object
{ temperature: 25.99000000000001, visibility: 25, location: "Chennai,IN" }


Home
Home
Visibility
25
%
0
100
Temperature
100
75
50
25
0
22:57:06
22:57:16
22:57:26
22:57:44
Location
Chennai,IN