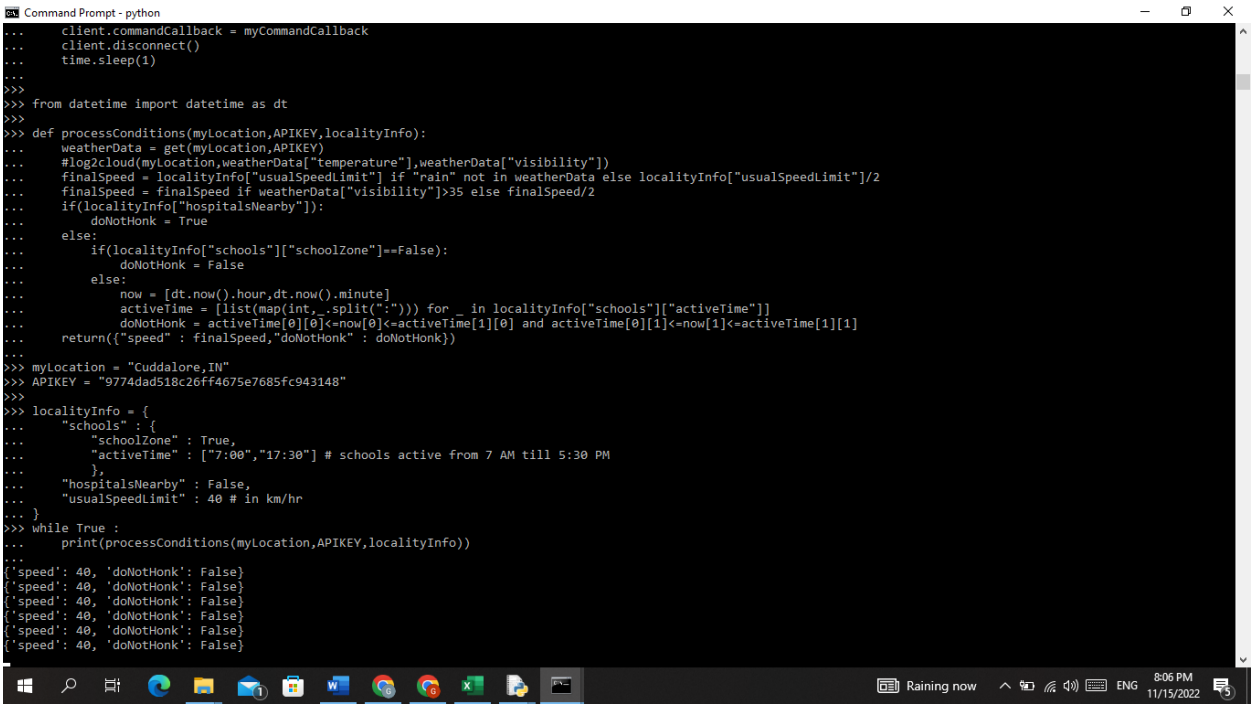
#PYTHON CODE

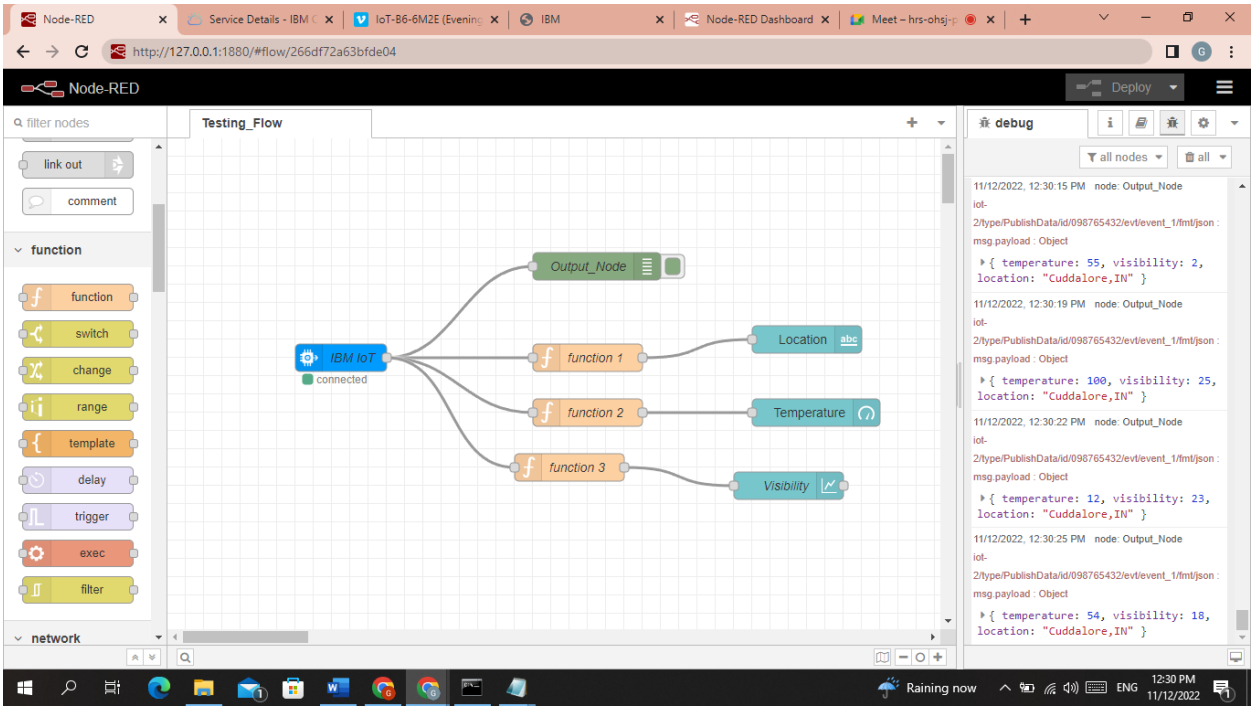
```
import requests as reqs
import wiotp.sdk.device # python -m pip install wiotp
import time
def get(myLocation,APIKEY):
    apiURL =
f"https://api.openweathermap.org/data/2.5/weather?q={myLocation}&appid={APIKEY}"
    responseJSON = (reqs.get(apiURL)).json()
    returnObject = {
        "temperature" : responseJSON['main']['temp'] - 273.15,
        "weather" : [responseJSON['weather'][_]['main'].lower() for _ in
range(len(responseJSON['weather']))],
        "visibility" : responseJSON['visibility']/100, # visibility in percentage where 10km is 100%
and 0km is 0%
    }
    if("rain" in responseJSON):
        returnObject["rain"] = [responseJSON["rain"][key] for key in responseJSON["rain"]]
    return(returnObject)



myConfig = {
    "identity" : {
        "orgId" : "f59trs",
        "typeId" : "testdevice",
        "deviceId" : "device1"
    },
```

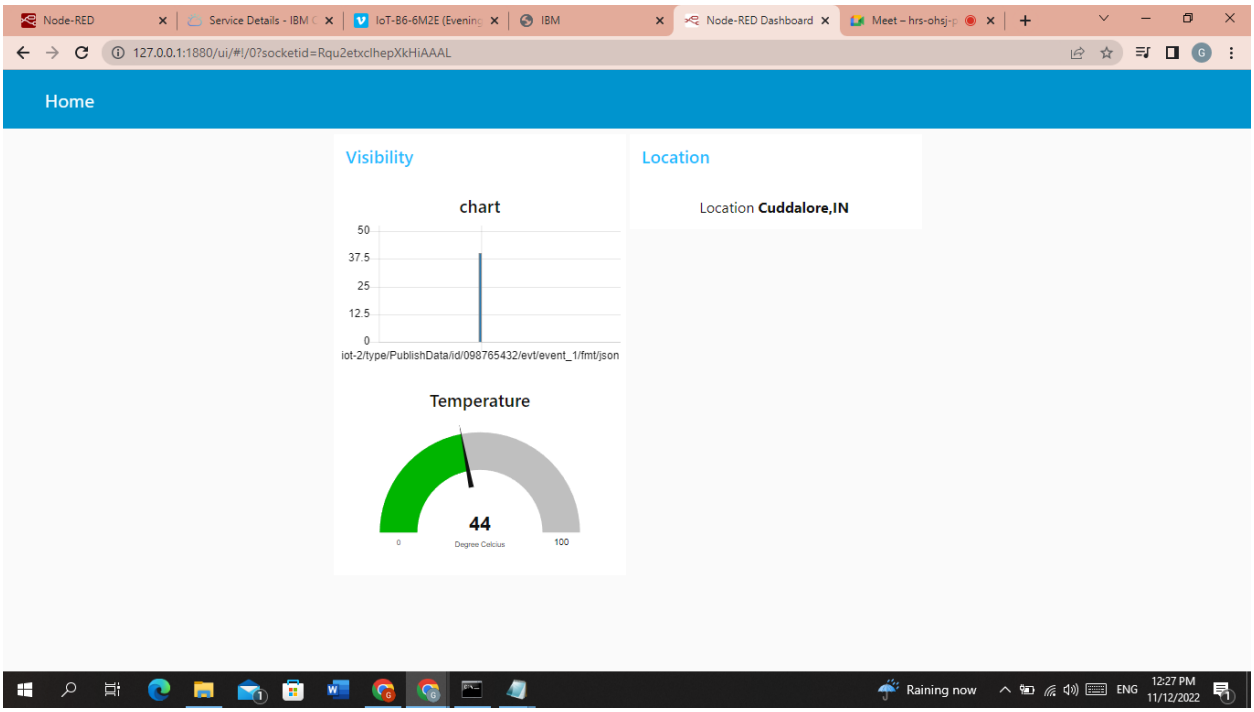
```
    "auth" : {
        "token" : "Jrwa7c8Os2Zpq)WW18"
    }
}
def myCommandCallback(cmd):
    print("recieved cmd : ",cmd)



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()
    time.sleep(1)



from datetime import datetime as dt


def processConditions(myLocation,APIKEY,localityInfo):
    weatherData = get(myLocation,APIKEY)
    #log2cloud(myLocation,weatherData["temperature"],weatherData["visibility"])
    finalSpeed = localityInfo["usualSpeedLimit"] if "rain" not in weatherData else
localityInfo["usualSpeedLimit"]/2
    finalSpeed = finalSpeed if weatherData["visibility"]>35 else finalSpeed/2
    if(localityInfo["hospitalsNearby"]):
        doNotHonk = True
```

```
    else:
        if(localityInfo["schools"]["schoolZone"]==False):
            doNotHonk = False
        else:
            now = [dt.now().hour,dt.now().minute]
            activeTime = [list(map(int,_.split(":"))) for _ in localityInfo["schools"]["activeTime"]]
            doNotHonk = activeTime[0][0]<=now[0]<=activeTime[1][0] and
activeTime[0][1]<=now[1]<=activeTime[1][1]
    return({"speed" : finalSpeed,"doNotHonk" : doNotHonk})


myLocation = "Cuddalore,IN"
APIKEY = "9774dad518c26ff4675e7685fc943148"


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
}
while True :
    print(processConditions(myLocation,APIKEY,localityInfo))
```

OUTPUT:


Command Prompt - python
client.commandCallback = myCommandCallback
client.disconnect()
time.sleep(1)
>>> from datetime import datetime as dt
>>> def processConditions(myLocation,APIKEY,localityInfo):
weatherData = get(myLocation,APIKEY)
#log2cloud(myLocation,weatherData["temperature"],weatherData["visibility"])
finalSpeed = localityInfo["usualSpeedLimit"] if "rain" not in weatherData else localityInfo["usualSpeedLimit"]/2
finalSpeed = finalSpeed if weatherData["visibility"]>35 else finalSpeed/2
if(localityInfo["hospitalsNearby"]):
doNotHonk = True
else:
if(localityInfo["schools"]["schoolZone"]==False):
doNotHonk = False
else:
now = [dt.now().hour,dt.now().minute]
activeTime = [list(map(int,_.split(":"))) for _ in localityInfo["schools"]["activeTime"]]
doNotHonk = activeTime[0][0]<=now[0]<=activeTime[1][0] and activeTime[0][1]<=now[1]<=activeTime[1][1]
return({"speed" : finalSpeed,"doNotHonk" : doNotHonk})
>>> myLocation = "Cuddalore,IN"
>>> APIKEY = "9774dad518c26ff4675e7685fc943148"
>>> 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
}
>>> while True :
print(processConditions(myLocation,APIKEY,localityInfo))
{'speed': 40, 'doNotHonk': False}
{'speed': 40, 'doNotHonk': False}
{'speed': 40, 'doNotHonk': False}
{'speed': 40, 'doNotHonk': False}
{'speed': 40, 'doNotHonk': False}
{'speed': 40, 'doNotHonk': False}
Raining now
ENG
8:06 PM
11/15/2022


Node-RED
Service Details - IBM
IoT-B6-6M2E (Evening
IBM
Node-RED Dashboard
Meet – hrs-ohsj-p
http://127.0.0.1:1880/#flow/266df72a63bfde04
Node-RED
Deploy
filter nodes
Testing_Flow
link out
comment
function
function
switch
change
range
template
delay
trigger
exec
filter
network
Output_Node
IBM IoT
connected
function 1
function 2
function 3
Location
Temperature
Visibility
debug
all nodes
all
11/12/2022, 12:30:15 PM node: Output_Node
iot-2/type/PublishData/id/098765432/evt/event_1/fmt/json : msg.payload : Object
{ temperature: 55, visibility: 2, location: "Cuddalore,IN" }
11/12/2022, 12:30:19 PM node: Output_Node
iot-2/type/PublishData/id/098765432/evt/event_1/fmt/json : msg.payload : Object
{ temperature: 100, visibility: 25, location: "Cuddalore,IN" }
11/12/2022, 12:30:22 PM node: Output_Node
iot-2/type/PublishData/id/098765432/evt/event_1/fmt/json : msg.payload : Object
{ temperature: 12, visibility: 23, location: "Cuddalore,IN" }
11/12/2022, 12:30:25 PM node: Output_Node
iot-2/type/PublishData/id/098765432/evt/event_1/fmt/json : msg.payload : Object
{ temperature: 54, visibility: 18, location: "Cuddalore,IN" }
Raining now
ENG
12:30 PM
11/12/2022


Node-RED
Service Details - IBM
IoT-B6-6M2E (Evening
IBM
Node-RED Dashboard
Meet – hrs-ohsj-p
127.0.0.1:1880/ui/#!/0?socketid=Rqu2etxclhepXkHiAAAL
Home
Visibility
chart
50
37.5
25
12.5
0
iot-2/type/PublishData/id/098765432/evt/event_1/fmt/json
Temperature
44
Degree Celcius
0
100
Location
Location Cuddalore,IN
Raining now
ENG
12:27 PM
11/12/2022