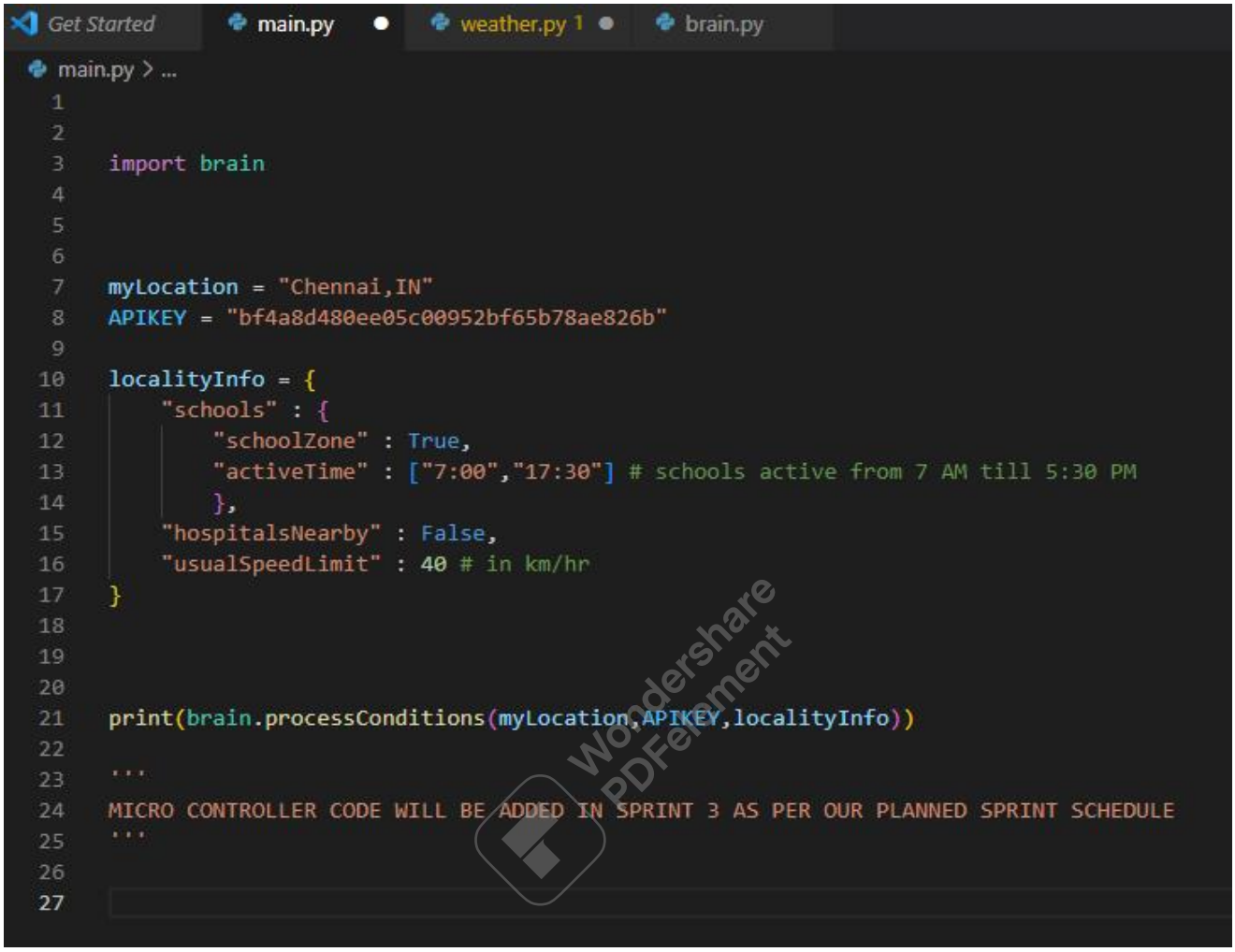
Develpement Phase Sprint 2

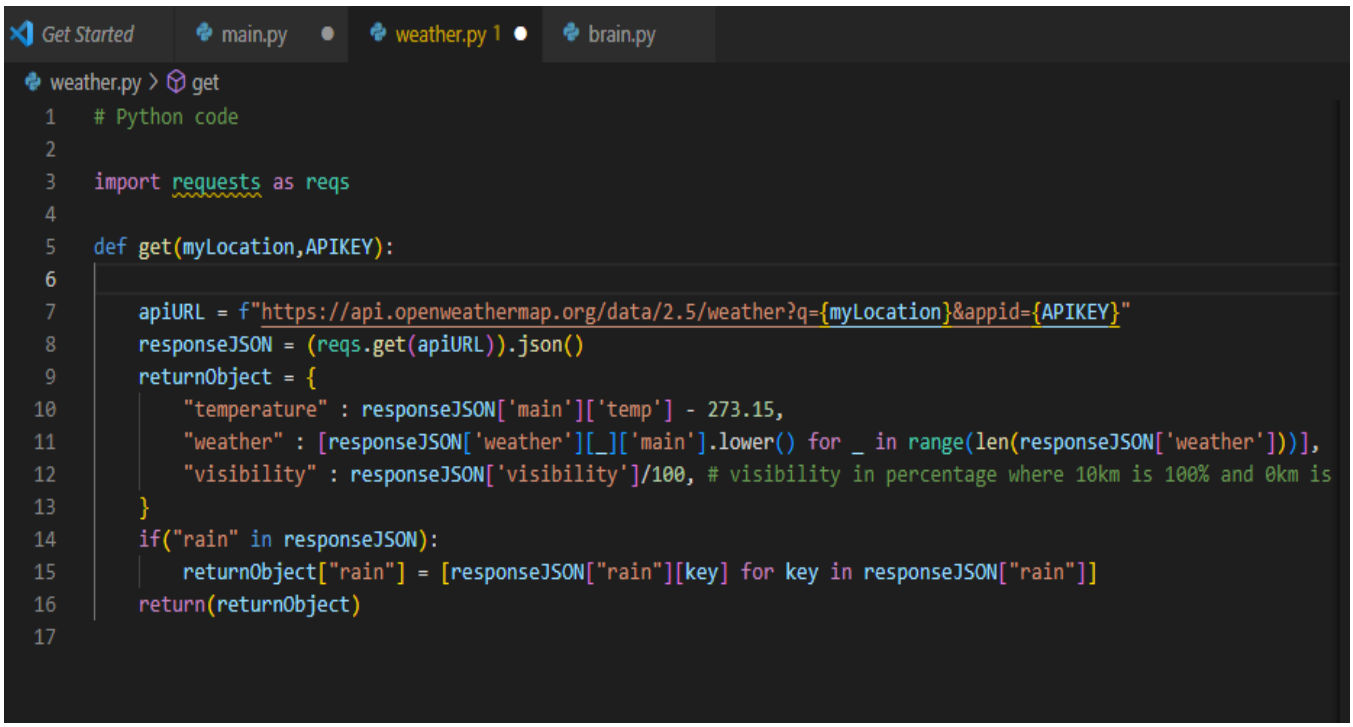
Team ID : PNT2022TMID22921

Project Name: Project- Signs with Smart Connectivity for Better Safety

Main.py




weather.py



publishData.py

main.py ● weather.py 1 ● brain.py ● publishData.py 1 ●

publishData.py > ...

```python

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

myConfig = {
    "identity" : {
        "orgId" : "epmoec",
        "typeId" : "testDevice",
        "deviceId" : "device0"
    },
    "auth" : {
        "token" : "?-KDXUPMvDo_TK2&b1"
    }
}

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)

```

## brain.py

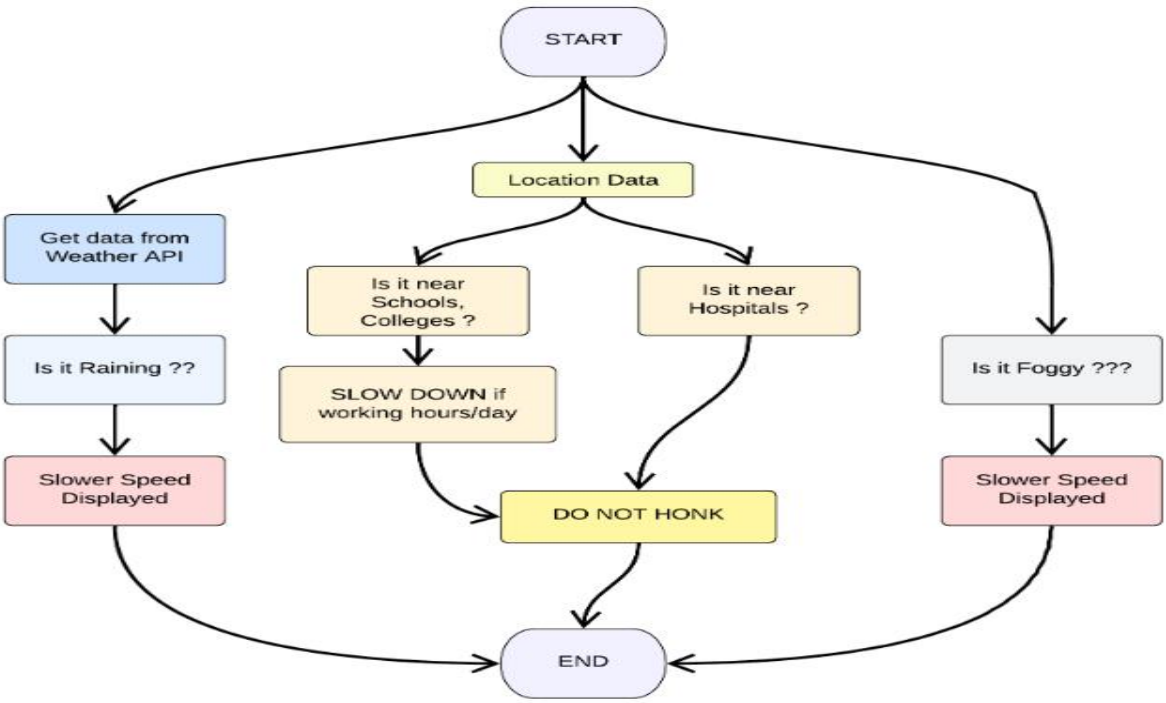
Get Started | main.py | weather.py 1 | brain.py

brain.py > ...

```python
import weather
from datetime import datetime as dt

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

    finalSpeed = localityInfo["usualSpeedLimit"] if "rain" not in weatherData else localityInfo["usualSpeedLimit
    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["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
    })

# UTILITY LOGIC SECTION ENDS
```



## Code Flow:



output:

```
# Code Output
2022-11-06 21:38:33,452   wiotp.sdk.device.client.DeviceClient  INFO
Connected successfully: d:epmoec:testDevice:device0
2022-11-06 21:38:33,452   wiotp.sdk.device.client.DeviceClient  INFO
Disconnected from the IBM Watson IoT Platform
2022-11-06 21:38:33,452   wiotp.sdk.device.client.DeviceClient  INFO
 Closed connection to the IBM Watson IoT Platform
{'speed': 40, 'doNotHonk': False}
2022-11-06 21:38:35,631   wiotp.sdk.device.client.DeviceClient  INFO
Connected successfully: d:epmoec:testDevice:device0
2022-11-06 21:38:35,631   wiotp.sdk.device.client.DeviceClient  INFO
Disconnected from the IBM Watson IoT Platform
2022-11-06 21:38:35,631   wiotp.sdk.device.client.DeviceClient  INFO
  Closed connection to the IBM Watson IoT Platform
{'speed': 40, 'doNotHonk': False}

... repeats every 1 sec
```

output image: