

## Project Milestone and Tasks

### Develop A Python Script

Date	24 October 2022
Team ID	PNT2022TMID41539
Project Name	Project-Signs with Smart Connectivity for Better Road Safety
Maximum marks	4 Marks

### Extract weather data from OpenWeatherMap using APIs

The screenshot shows the OpenWeatherMap API website. The browser's address bar displays 'openweathermap.org/api'. The website's navigation bar includes links for 'Weather in your city', 'Guide', 'API', 'Dashboard', 'Marketplace', 'Pricing', 'Maps', 'Our Initiatives', 'Partners', 'Blog', 'For Business', 'Sign in', and 'Support'. The main heading is 'Weather API'. Below this, a paragraph encourages users to sign up for the fast and easy-to-work weather APIs, recommending the 'One Call API 3.0'. A 'Subscribe' button is visible. The 'One Call API 3.0' is highlighted as 'NEW'. A list of features is provided: Minute forecast for 1 hour, Hourly forecast for 48 hours, Daily forecast for 8 days, Historical data for 40+ years back by timestamp, and National weather alerts. The pricing is 'Pay as you call' at '1,000 API calls per day for free' and '0.0012 GBP per API call over the daily limit'. A 'Subscribe to One Call by Call' button is present. A footer section mentions 'Professional collections' and 'Allow all' / 'Manage cookies' options. The taskbar at the bottom shows various application icons and the system clock indicating 00:01 on 19-11-2022.

IBM Weather AP Service Det IBM Watson Node-RED Node-RED MIT App In MIT App In Inbox (24)

openweathermap.org/api

OpenWeather Weather in your city Guide API Dashboard Marketplace Pricing Maps Our Initiatives Partners Blog For Business Sign in Support

## Weather API

Home / Weather API

Please, **sign up** to use our fast and easy-to-work weather APIs. As a start to use OpenWeather products, we recommend our **One Call API 3.0**. For more functionality, please consider our products, which are included in **professional collections**.

### One Call API 3.0 **NEW**

API doc **Subscribe**

Make one API call and receive all essential weather data in one response:

- Minute forecast for 1 hour
- Hourly forecast for 48 hours
- Daily forecast for 8 days
- Historical data for 40+ years back by timestamp
- National weather alerts

Read more about this API and subscription plan in the **FAQ**.

This is a separate subscription plan, which include only One Call API.

**Pay as you call**

**1,000 API calls per day for free**  
**0.0012 GBP** per API call over the daily limit

**Subscribe to One Call by Call**

We use cookies which are essential for the site to work. We also use non-essential cookies to help us improve our services. Any data collected is anonymised. You can allow all cookies or manage them individually.

**Professional collections**

Allow all Manage cookies

Waiting for tpc.google syndication.com...

00:01  
19-11-2022

## Python Script Developed

```
Python code.py - C:\Users\Oomsithan\Desktop\PROJECT\Python code.py (3.7.0)
File Edit Format Run Options Window Help

import wiotp.sdk.device
import time
import random
import ibmiotf.application
import ibmiotf.device
import requests, json

myConfig = {
    #Configuration
    "identity": {
        "orgId": "74k16v",
        "typeId": "Board",
        "deviceId": "20082001"
    },
    #API Key
    "auth": {
        "token": "12345678"
    }
}

#Receiving callbacks from IBM IOT platform
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 = "Nagercoil"
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']
```

Ln: 1 Col: 0

00:03  
19-11-2022

```
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']

#message part
msg=random.randint(0,5)
if msg==1:
    message="MESSAGE: SLOW DOWN, SCHOOL IS NEAR"
elif msg==3:
    message="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="SPEED MESSAGE: SLOW DOWN, speed Limit Exceeded"
elif speed>=60 and speed<100:
    speedMsg="SPEED MESSAGE: Moderate"
else:
    speedMsg="SPEED MESSAGE: Slow"

#Sign part
sign=random.randint(0,5)
if sign==1:
    signMsg="SIGN: Right Diversion"
elif sign==3:
    signMsg="SIGN: Left Diversion"
elif sign==5:
    signMsg="SIGN: U Turn"
else:
    signMsg=""

#Visibility
if temperature < 50:
```



```
Python code.py - C:\Users\Oomsithan\Desktop\PROJECT\Python code.py (3.7.0)
File Edit Format Run Options Window Help
message= 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="SPEED MESSAGE: SLOW DOWN, speed Limit Exceeded"
elif speed>=60 and speed<100:
    speedMsg="SPEED MESSAGE: Moderate"
else:
    speedMsg="SPEED MESSAGE: Slow"

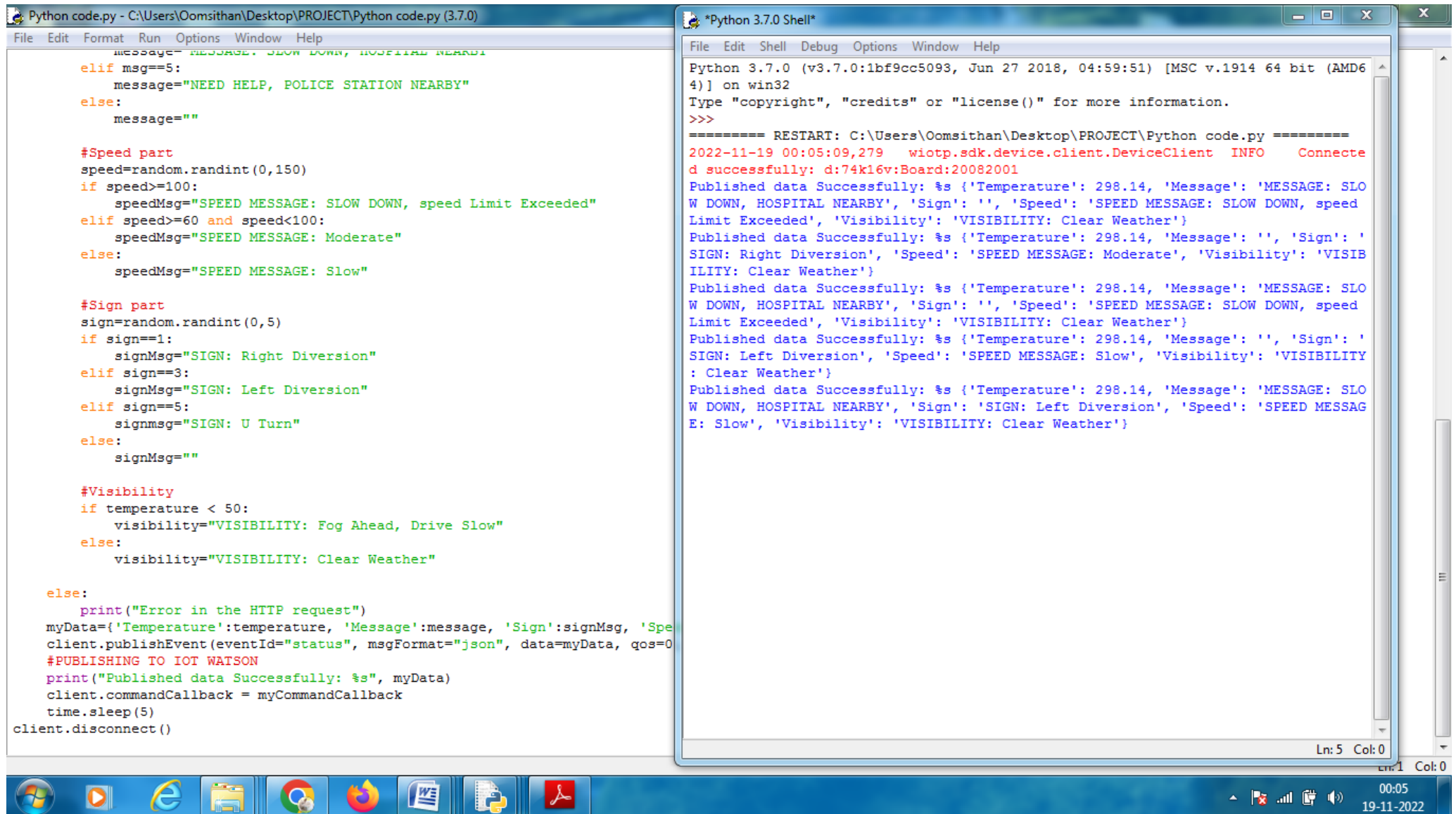
#Sign part
sign=random.randint(0,5)
if sign==1:
    signMsg="SIGN: Right Diversion"
elif sign==3:
    signMsg="SIGN: Left Diversion"
elif sign==5:
    signMsg="SIGN: U Turn"
else:
    signMsg=""

#Visibility
if temperature < 50:
    visibility="VISIBILITY: Fog Ahead, Drive Slow"
else:
    visibility="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)
#PUBLISHING TO IOT WATSON
print("Published data Successfully: %s", myData)
client.commandCallback = myCommandCallback
time.sleep(5)
client.disconnect()
```

Send the extracted data to the cloud by running the code

## Receive data from the cloud and view it in the python compiler( Python code output)



The image shows a screenshot of a Python IDE with two windows. The left window, titled 'Python code.py - C:\Users\Oomsithan\Desktop\PROJECT\Python code.py (3.7.0)', contains Python code that generates a JSON message based on speed, sign, and visibility. The right window, titled '\*Python 3.7.0 Shell\*', shows the output of the code execution, including a restart message and several 'Published data Successfully' logs with JSON data.

```
Python code.py - C:\Users\Oomsithan\Desktop\PROJECT\Python code.py (3.7.0)
File Edit Format Run Options Window Help
message="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="SPEED MESSAGE: SLOW DOWN, speed Limit Exceeded"
elif speed>=60 and speed<100:
    speedMsg="SPEED MESSAGE: Moderate"
else:
    speedMsg="SPEED MESSAGE: Slow"

#Sign part
sign=random.randint(0,5)
if sign==1:
    signMsg="SIGN: Right Diversion"
elif sign==3:
    signMsg="SIGN: Left Diversion"
elif sign==5:
    signMsg="SIGN: U Turn"
else:
    signMsg=""

#Visibility
if temperature < 50:
    visibility="VISIBILITY: Fog Ahead, Drive Slow"
else:
    visibility="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)
#PUBLISHING TO IOT WATSON
print("Published data Successfully: %s", myData)
client.commandCallback = myCommandCallback
time.sleep(5)
client.disconnect()
```

```
*Python 3.7.0 Shell*
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\Oomsithan\Desktop\PROJECT\Python code.py =====
2022-11-19 00:05:09,279 wiotp.sdk.device.client.DeviceClient INFO Connected successfully: d:74k16v:Board:20082001
Published data Successfully: %s {'Temperature': 298.14, 'Message': 'MESSAGE: SLOW DOWN, HOSPITAL NEARBY', 'Sign': '', 'Speed': 'SPEED MESSAGE: SLOW DOWN, speed Limit Exceeded', 'Visibility': 'VISIBILITY: Clear Weather'}
Published data Successfully: %s {'Temperature': 298.14, 'Message': '', 'Sign': 'SIGN: Right Diversion', 'Speed': 'SPEED MESSAGE: Moderate', 'Visibility': 'VISIBILITY: Clear Weather'}
Published data Successfully: %s {'Temperature': 298.14, 'Message': 'MESSAGE: SLOW DOWN, HOSPITAL NEARBY', 'Sign': '', 'Speed': 'SPEED MESSAGE: SLOW DOWN, speed Limit Exceeded', 'Visibility': 'VISIBILITY: Clear Weather'}
Published data Successfully: %s {'Temperature': 298.14, 'Message': '', 'Sign': 'SIGN: Left Diversion', 'Speed': 'SPEED MESSAGE: Slow', 'Visibility': 'VISIBILITY: Clear Weather'}
Published data Successfully: %s {'Temperature': 298.14, 'Message': 'MESSAGE: SLOW DOWN, HOSPITAL NEARBY', 'Sign': 'SIGN: Left Diversion', 'Speed': 'SPEED MESSAGE: Slow', 'Visibility': 'VISIBILITY: Clear Weather'}
Ln: 5 Col: 0
```