

PROJECT DEVELOPMENT PHASE SPRINT – 2 (USN-2)

TEAM ID	PNT2022TMID22305
PROJECT NAME	SIGNS WITH SMART CONNECTIVITY FOR BETTER ROAD SAFETY

OPEN WEATHER MAP

#OPENWEATHER MAP(SPRINT 2)

import wiotp.sdk.device #importing library files for connecting with CLOUD, sdk=software development kit

import requests #for API request

import json #converting it to json(key:values) myConfig

```
= {
    "identity": {
        "orgId": "vrpc8b",
        "typeId": "Ecedevice",    #configuration with CLOUD, finding identity
        "deviceId": "123456"
    },
    "auth": {
        "token": "Mukil@12"    #authenticating with cloud device
    }
}
```

client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)

#initialising device client with above myconfig detail

client.connect()

while True:

```
    print("=====")
```

```
    weatherData =
```

```
requests.get('https://api.openweathermap.org/data/2.5/weather?q=Coimbatore
&appid=7d04e779249c800fe47641f63937b58c&units=metric')
```

```
    a=weatherData.text
```

```
    b=json.loads(a)
```

```
    temp = b["main"]["temp"]
```

```
    humi = b["main"]["humidity"]
```

```
    main = b["weather"][0]["main"]    #0th index is taken from the object
```

```
    description = b["weather"][0]["description"]
```

```
    Visibility = b["visibility"]
```

```
    TemperatureRecommendation = ""
```

```
    SpeedRecommendation = ""
```

```
    RecommendationForVisibility = ""
```

```
    #print("Temperature(celcius) :", b["main"]["temp"]) if
```

```
    (temp>33):
```

```

    TemperatureRecommendation="Temperature is higher than ideal value"
    #print("Temperature is higher than ideal value")elif
(temp<19):
    TemperatureRecommendation="Temperature is lower than ideal value"
    #print("Temperature is lower than ideal value")
else:
    TemperatureRecommendation="Temperature is ideal"
    #print("Temperature is ideal ")
    #print("Humidity :",b["main"]["humidity"])
    #print("WeatherCondition",(b["weather"][0]["main"]))if
(main == "Rain"):
    rain = b["rain"]["1h"]
    SpeedRecommendation = "30KM/HR ,ROAD WILL BE SLIPPERY"
    #print("Rain:",b["rain"]["1h"])
    #print("SPEED RECOMMENDATION : 30KM/HR ,ROAD WILL BE
SLIPPERY")
elif (main == "Drizzle"):
    SpeedRecommendation = "30KM/HR"
    #print("SPEED RECOMMENDATION : 30KM/HR")
elif (main == "Mist"):
    SpeedRecommendation = "30KM/HR and switch on the headlight" #print("SPEED
RECOMMENDATION : 30KM/HR and switch on the
Headlight")
    #print("Description of weather :", (b["weather"][0]["description"]))
    #print("visibility", (b["visibility"]))
if (Visibility<1000):
    RecommendationForVisibilty = "SPEED RECOMMENDATION : 30KM/HRand
SWITCH ON THE HEAD LIGHT"
else:
    RecommendationForVisibilty = "Visibility range is ideal for vechicles" #print("SPEED
RECOMMENDATION : 30KM/HR and SWITCH ON THE
HEAD LIGHT")
mydata={"temperature":temp,
"TemperatureRecommendation":TemperatureRecommendation,"humidity":hum
i,"WeatherCondition":main,"SpeedRecommendation":SpeedRecommendation
,"DescriptionOfWeather":description,"Visibility":Visibility,"RecommendationForV
isibilty":RecommendationForVisibilty}
print(mydata)
client.publishEvent("12345","json",mydata)

```

CODE IN PYTHON IDLE

```
finalpy - C:\Users\hemant\Desktop\finalpy\finalpy (3.7.0)
File Edit Format Run Options Window Help

import wiotp.sdk.device #importing library files for connecting with CLOUD,sdk=software development kit
import requests #for API request
import json #converting it to json(key:values)
import sys

myConfig = {
    "identity": {
        "orgId": "c0mbt9",
        "typeId": "SmartSigns", #configuration wit CLOUD,finding identity
        "deviceId": "SS"
    },
    "auth": {
        "token": "Hrtme!0y*PQT-s@Hxz" #authenticating with cloud device
    }
}

#TRAFFIC AND FATAL SITUATION ALERT MESSAGE DISPLAYING IN WEB UI WHEN THE
client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None) #initialising device client with above myconfig detail
client.connect()
ALERT=""
NOTIFY=""

def myCommandCallback(cmd):
    print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
    m=cmd.data['command']
    #THIS IF CONDITION BLOCK IS FOR TRAFFIC AND FATAL SITUATION ALERT MESSAGE DISPLAYING IN WEB UI WHEN THE MESSAGE WAS RECEIVED FROM THE ROAD SAFETY OFFICE
    ALERT=""
    NOTIFY=""
    if (m=="TRAFFIC"):
        ALERT="TRAFFIC - PLEASE WAIT OR PREFER ANOTHER ROUTE"
        print("*****//PLEASE WAIT OR PREFER ANOTHER ROUTE//*****")
    elif (m=="ACCIDENT"):
        ALERT="ACCIDENT - TAKE DIVERSION"
        print("*****//TAKE DIVERSION//*****")
    elif (m=="MESSAGE"):
        ALERT="HAVE A NICE DAY!"
        print("HAVE A NICE DAY!")
    #THE BELOW CONDITION BLOCK IS TO DISPLAY HOSPITAL ,SCHOOL, AND RESTAURANT REGIONED AREA AND SPEED RECOMMENDATION
    if (m=="SCHOOL"):
        NOTIFY="SCHOOL REGION MAINTAIN SPEED LIMIT BELOW 40KM/HR"
        print("SCHOOL REGION MAINTAIN SPEED LIMIT BELOW 40KM/HR")
    elif (m=="HOSPITAL"):
        NOTIFY="HOSPITAL REGION DONT USE HORN"
        print("HOSPITAL REGION DONT USE HORN")
    elif (m=="RESTAURANT"):
        NOTIFY="CROWDED AREA PLEASE MAINTAIN SPEED LIMIT"
        print("CROWDED AREA PLEASE MAINTAIN SPEED LIMIT")
    mydata={}
    if (m=="TRAFFIC" or m=="ACCIDENT" or m=="MESSAGE"):
        mydata={"CAUTION":ALERT}
    elif (m=="SCHOOL" or m=="HOSPITAL" or m=="RESTAURANT"):
        mydata={"CAUTION":NOTIFY}
    client.publishEvent("SS",json,mydata)
while True:
    print("=====")
    AREA = "Chennai, IN"
    weatherData = requests.get("https://api.openweathermap.org/data/2.5/weather?q=" + AREA + "&appid=c023e4f9eaf0ba505b859624415b4ae&units=metric")
    a=weatherData.text
    b=json.loads(a)
    temp = b["main"]["temp"]
    humi = b["main"]["humidity"]
    main = b["weather"][0]["main"] #0th index is taken from the object
    description = b["weather"][0]["description"]
    visibility = b["visibility"]
    windspeed = b["wind"]["speed"]

    TemperatureRecommendation=""
    SpeedRecommendation=""

    RecommendationForVisibility=""
    #print("Temperature(celcius) :",b["main"]["temp"])
    if (temp>33):
        TemperatureRecommendation="Temperature is higher than ideal value"
        #print("Temperature is higher than ideal value")
    elif (temp<19):
        TemperatureRecommendation="Temperature is lower than ideal value"
        #print("Temperature is lower than ideal value")
    else:
        TemperatureRecommendation="Temperature is ideal"
        #print("Temperature is ideal ")
    #print("Humidity :",b["main"]["humidity"])
    #print("WeatherCondition", (b["weather"][0]["main"]))
    if (main == "Rain"):
        rain = b["rain"]["rh"]
        SpeedRecommendation = "30KM/HR ,ROAD WILL BE SLIPPERY"
        #print("Rain:",b["rain"]["rh"])
        #print("SPEED RECOMMENDATION : 30KM/HR ,ROAD WILL BE SLIPPERY")
    elif (main == "Drizzle"):
        SpeedRecommendation = "30KM/HR"
        #print("SPEED RECOMMENDATION : 30KM/HR")
    elif (main == "Mist"):
        SpeedRecommendation = "30KM/HR and switch on the headlight"
        #print("SPEED RECOMMENDATION : 30KM/HR and switch on the Headlight")
    elif (main == "Thunderstorm"):
        SpeedRecommendation = "30KM/HR and stay away in the open place"
        #print("SPEED RECOMMENDATION : 30KM/HR and stay away in the open place")
    elif (main == "Clouds"):
        SpeedRecommendation = "MAINTAIN NORMAL SPEED LIMIT UPTO 50 KM/HR"
        #print("SPEED RECOMMENDATION : 30KM/HR and stay away in the open place")
    #print("Description of weather :", (b["weather"][0]["description"]))
    #print("visibility", (b["visibility"]))
    if (visibility<1000):
        RecommendationForVisibility = "SPEED RECOMMENDATION : 30KM/HR and SWITCH ON THE HEAD LIGHT"
    else:
        RecommendationForVisibility = "visibility range is ideal for vehicles"
    #print("SPEED RECOMMENDATION : 30KM/HR and SWITCH ON THE HEAD LIGHT")
    mydata["TemperatureRecommendation":TemperatureRecommendation,"humidity":humi,"WeatherCondition":main,"SpeedRecommendation":SpeedRecommendation,"DescriptionOfWeather":description,"visibility":visibility,"Recommen"
    print(mydata)
    client.publishEvent("SS",json,mydata)
    client.commandCallback = myCommandCallback
```

```
finalpy - C:\Users\hemant\Desktop\finalpy\finalpy (3.7.0)
File Edit Format Run Options Window Help

mydata={"CAUTION":NOTIFY}
client.publishEvent("SS",json,mydata)
while True:
    print("=====")
    AREA = "Chennai, IN"
    weatherData = requests.get("https://api.openweathermap.org/data/2.5/weather?q=" + AREA + "&appid=c023e4f9eaf0ba505b859624415b4ae&units=metric")
    a=weatherData.text
    b=json.loads(a)
    temp = b["main"]["temp"]
    humi = b["main"]["humidity"]
    main = b["weather"][0]["main"] #0th index is taken from the object
    description = b["weather"][0]["description"]
    visibility = b["visibility"]
    windspeed = b["wind"]["speed"]

    TemperatureRecommendation=""
    SpeedRecommendation=""

    RecommendationForVisibility=""
    #print("Temperature(celcius) :",b["main"]["temp"])
    if (temp>33):
        TemperatureRecommendation="Temperature is higher than ideal value"
        #print("Temperature is higher than ideal value")
    elif (temp<19):
        TemperatureRecommendation="Temperature is lower than ideal value"
        #print("Temperature is lower than ideal value")
    else:
        TemperatureRecommendation="Temperature is ideal"
        #print("Temperature is ideal ")
    #print("Humidity :",b["main"]["humidity"])
    #print("WeatherCondition", (b["weather"][0]["main"]))
    if (main == "Rain"):
        rain = b["rain"]["rh"]
        SpeedRecommendation = "30KM/HR ,ROAD WILL BE SLIPPERY"
        #print("Rain:",b["rain"]["rh"])
        #print("SPEED RECOMMENDATION : 30KM/HR ,ROAD WILL BE SLIPPERY")
    elif (main == "Drizzle"):
        SpeedRecommendation = "30KM/HR"
        #print("SPEED RECOMMENDATION : 30KM/HR")
    elif (main == "Mist"):
        SpeedRecommendation = "30KM/HR and switch on the headlight"
        #print("SPEED RECOMMENDATION : 30KM/HR and switch on the Headlight")
    elif (main == "Thunderstorm"):
        SpeedRecommendation = "30KM/HR and stay away in the open place"
        #print("SPEED RECOMMENDATION : 30KM/HR and stay away in the open place")
    elif (main == "Clouds"):
        SpeedRecommendation = "MAINTAIN NORMAL SPEED LIMIT UPTO 50 KM/HR"
        #print("SPEED RECOMMENDATION : 30KM/HR and stay away in the open place")
    #print("Description of weather :", (b["weather"][0]["description"]))
    #print("visibility", (b["visibility"]))
    if (visibility<1000):
        RecommendationForVisibility = "SPEED RECOMMENDATION : 30KM/HR and SWITCH ON THE HEAD LIGHT"
    else:
        RecommendationForVisibility = "visibility range is ideal for vehicles"
    #print("SPEED RECOMMENDATION : 30KM/HR and SWITCH ON THE HEAD LIGHT")
    mydata["TemperatureRecommendation":TemperatureRecommendation,"humidity":humi,"WeatherCondition":main,"SpeedRecommendation":SpeedRecommendation,"DescriptionOfWeather":description,"visibility":visibility,"Recommen"
    print(mydata)
    client.publishEvent("SS",json,mydata)
    client.commandCallback = myCommandCallback
```

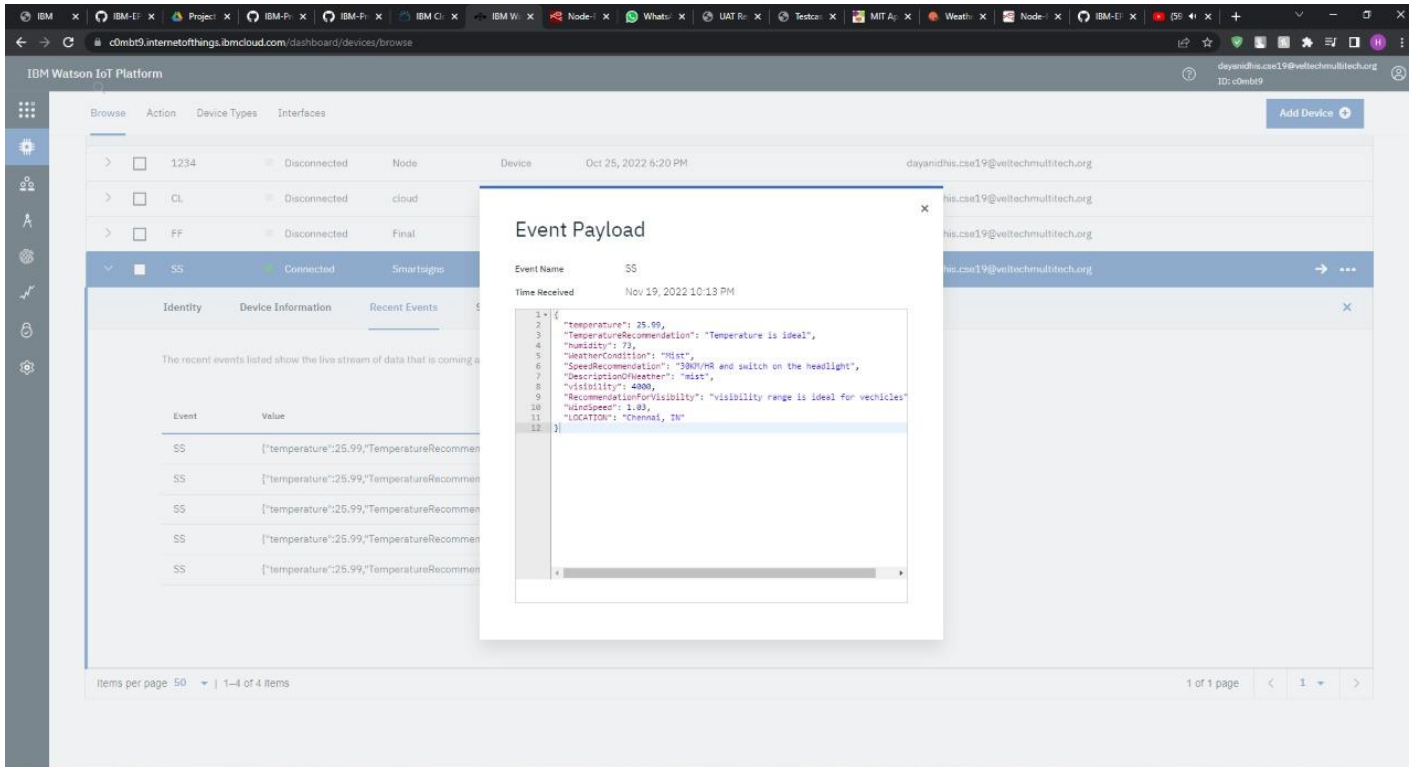
PYTHON OUTPUT:

[illegible]

ESTABLISHING THE OPENWEATHERMAP CODE TO CLOUD TO EASY ACCESSING BYCREATING A DEVICE IN IBM WATSON WITH THE CONFIGURATION DETAILS:

The screenshot displays the IBM Watson IoT Platform interface. At the top, a navigation bar includes tabs for 'Browse', 'Action', 'Device Types', and 'Interfaces'. A sidebar on the left contains icons for various platform functions. The main content area shows a list of devices with columns for status, type, name, and last update time. The 'SS' device is highlighted, and a modal window titled 'Identity' is open, showing 'Recent Events' for this device. The events table lists temperature recommendations with their values, formats, and timestamps. At the bottom, pagination controls show '1 of 1 page' and '1 - 4 of 4 items'.

Event	Value	Format	Last Received
SS	["temperature":25.99,"TemperatureRecommend...	json	a few seconds ago
SS	["temperature":25.99,"TemperatureRecommend...	json	a few seconds ago
SS	["temperature":25.99,"TemperatureRecommend...	json	a few seconds ago
SS	["temperature":25.99,"TemperatureRecommend...	json	a few seconds ago
SS	["temperature":25.99,"TemperatureRecommend...	json	a few seconds ago



COLLECTING THE DATA FROM THE IBM WATSON CLOUD THROUGH AUTHENTICATION IN NODE RED SERVICE:

