Project development Phase sprint-4

Team id	PNT2022TMID33212
Project name	Signs with smart connectivity for
	better road safety

Code for (random temperature, road signs, speed limit, message):

```
import wiotp.sdk.device
import time
import random
import ibmiotf.application
import ibmiotf.device
import requests, json
myConfig = {
  #Configuration
  "identity": {
    "orgId": "o3esI6",
    "typeId": "HAYAN24",
    "deviceId":"HAYAN24id"
    },
  #API Key
  "auth": {
    "token": "hayanthika2411"
    }
}
```

```
#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 = "Dindigul, IN"
URL = BASE_URL + "q=" + CITY + "&units=metric"+"&appid=" + "f58e4720c739a54c439aba9b05176839"
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']
    #messge part
    msg=random.randint(0,5)
    if msg==1:
      message="GO SLOW, SCHOOL ZONE AHEAD"
    elif msg==2:
      message="NEED HELP, POLICE STATION AHEAD"
    elif msg==3:
      message="EMERGENCY, HOSPITAL NEARBY"
```

```
elif msg==4:
  message="DINE IN, RESTAURENT AVAILABLE"
elif msg==5:
 message="PETROL BUNK NEARBY"
else:
 message=""
#Speed Limit part
speed=random.randint(0,150)
if speed>=100:
  speedMsg=" Limit Exceeded"
elif speed>=60 and speed<100:
  speedMsg="Moderate"
else:
  speedMsg="Slow"
#Diversion part
sign=random.randint(0,5)
if sign==1:
 signMsg="Right Diversion"
elif sign==2:
 signMsg="Speed Breaker"
elif sign==3:
 signMsg="Left Diversion"
elif sign==4:
  signmsg="U Turn"
else:
 signMsg=""
#Visibility
```

```
if temperature < 24:
      visibility="Fog Ahead, Drive Slow"
    elif temperature < 20:
      visibility="Bad Weather"
    else:
      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: ", myData)
    print("")
    client.commandCallback = myCommandCallback
    time.sleep(5)
client.disconnect()
```

python simulation:

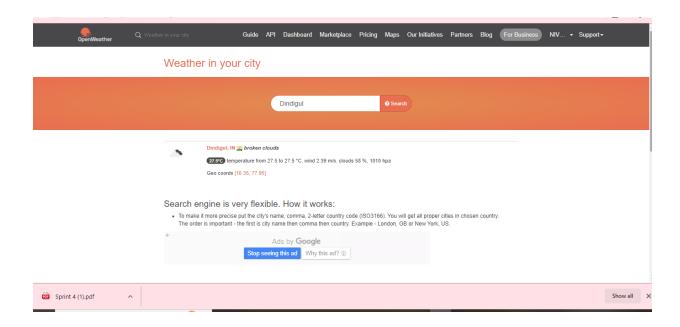
```
File Edit Format Run Options Window Help
Lamport time
Lamport time
Lamport intentification
Lamport intentification
Lamport intentification
Lamport intentification
Lamport intentification
Lamport requests, yaon
Myconfiguration
Lamport intentification
Lamport requests, yaon
Myconfiguration
Lamport intentification
Lamport intentification
Lamport requests, yaon
Myconfiguration
Lamport intentification
Lamport intent
```

Import wiotp-sdk and ibm iot:

```
Ricrosoft Windows [Version 18.8.19844.2251]
(c) Microsoft Corporation. All rights reserved.

CilversIMPLAppdata\locallyporgams\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\Python\P
```

OpenWeatherMap - (Ex : Dindigul,IN):



Python IDLE Output:

