

SPRINT 3

Team ID	PNT2022TMID52137
Project Name	Signs with Smart Connectivity for Better Road Safety

SPRINT	FUNCTIONAL REQUIREMENT (EPIC)	USER STORY/TASK	STORY POINTS	PRIORITY	TEAM MEMBERS
Sprint-3	Resources Initialization	Create and initialize accounts in various public APIs like OpenWeather API.	1	LOW	Ravichandran.T Pooja.E Robinson.S Rooba.P

STEP 1: PYTHON CODE STIMULATION

```
File Edit Format Run Options Window Help
message="SLOW DOWN, SCHOOL IS NEAR"
elif msg==2:
    message="NEED HELP, POLICE STATION AHEAD"
elif msg==3:
    message="EMERGENCY, HOSPITAL NEARBY"
elif msg==4:
    message="DINE IN, RESTAURENT AVAILABLE"
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==3:
    signMsg="Left Diversion"
elif sign==5:
    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.publish(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
#PUBLISHING TO IOT WATSON
print("Published data Successfully: %s", myData)
client.disconnect()
```

PYTHON CODE:

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

myConfig = { #Configuration
    "identity": {
        "orgId": "d5zx56",
        "typeId": "Connectivity123", "deviceId": "ESP32"},
    #API Key
    "auth": {
        "token": "9514598766"
    }
}

#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.commandCallback= myCommandCallback
client.connect()

#OpenWeatherMap Credentials
BASE_URL="https://api.openweathermap.org/data/2.5/weather?"
CITY = "Chennai"
URL = BASE_URL + "q=" + CITY + "&units=metric"+"&appid=" +
"9cca583812b638930cefd580106f6c58"

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="SLOW DOWN, SCHOOL IS NEAR"
elif msg==2:
    message="NEED HELP, POLICE STATION AHED"
elif msg==3:
    message="EMERGENCY, HOSPITAL NEARBY"
elif msg==4:
    message="DINE IN, RESTAURENT AVAILABLE"
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==3:
    signMsg="Left Diversion"
elif sign==5:
    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: %s", myData)

client.disconnect()

```

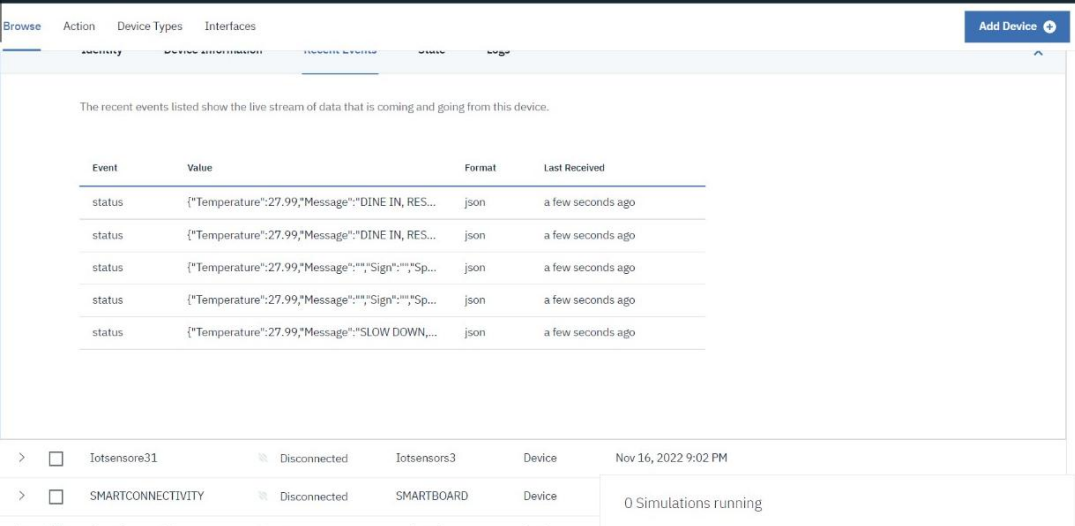
OUTPUT OF PYTHON CODE:

[illegible]

STEP 2:

IOT DEVICE- IOT PLATFORM

By running the code in python IDLE ,the data is published in IBM cloud.



The screenshot displays the IBM Cloud IoT Platform interface. At the top, there are navigation tabs: 'Browse', 'Action', 'Device Types', and 'Interfaces'. A blue 'Add Device' button is located in the top right corner. Below the tabs, a message states: 'The recent events listed show the live stream of data that is coming and going from this device.' Below this message is a table with the following data:

Event	Value	Format	Last Received
status	{"Temperature":27.99,"Message":"DINE IN, RES..."}	json	a few seconds ago
status	{"Temperature":27.99,"Message":"DINE IN, RES..."}	json	a few seconds ago
status	{"Temperature":27.99,"Message":"","Sign":"","Sp..."}	json	a few seconds ago
status	{"Temperature":27.99,"Message":"","Sign":"","Sp..."}	json	a few seconds ago
status	{"Temperature":27.99,"Message":"SLOW DOWN,..."}	json	a few seconds ago

Below the table, there is a list of devices. The first device is 'Iotsensore31', which is 'Disconnected' and has a status of 'Iotsensors3'. The second device is 'SMARTCONNECTIVITY', which is also 'Disconnected' and has a status of 'SMARTBOARD'. The interface indicates that there are '0 Simulations running'.