Project Development Phase
Sprint-3

Date	18thNovember2022
Team ID	**PNT2022TMID17542**
Project Name	Project: Signs with Smart Connectivity for Better Road Safety.
Marks	20 Marks

Sprint	functional requirement	User Story Number	User Story/Task	Story Points	Priority	Team members
Sprint-3		US-1	Develop a python script to publish random sensordata such as temperate, humidity, and visibility to the IBM IoT Platform.	7	High	PradeepKumar S Arunpandiyan R SudhanKarthick K KarthicBabu KG
Sprint-3		US-2	After developing the python code, commands are received and the statement representing the devices' control are printed.	5	Medium	PradeepKumar S Arunpandiyan R SudhanKarthick K KarthicBabu KG
Sprint-3		US-3	Publish Data tothe IBM cloud.	8	High	PradeepKumar S Arunpandiyan R SudhanKarthick K KarthicBabu KG

US-
1Developapythonscripttopublishrandomsensordatasuchastemperature,humidity, and visibility to the IBM IoT Platform

```
import time
import sys
import ibmiotf.application
import ibmiotf. device
import random

#ProvideyourIBMWatsonDeviceCredentials
organization="33lnun"
deviceType="PNT2022TMID38378"deviceId
="PNT2022TMID38378"
authMethod="token" authToken=
"F*IPB)eaQ5+CqaU(Gj"


#IntializeGPIO

defmyCommandCallback(cmd):
    print("Commandreceived:%s%cmd.data['command']")status=cmd.data['command]
    ifstatus=="lighton":print
        ("ledison")
    else:
        print("ledisoff")

    #print(cmd)

try:
    deviceOptions={"org":organization,"type":deviceType,"id":deviceId,"auth-
method":authMethod,"auth-token":authToken}
    deviceCli=ibmiotf.device.Client(deviceOptions)#.....................
    ...

exceptExceptionase:
    print("Caughtexceptionconnectingdevice:%s"%str(e))sys.exit()

    #Connectandsendadatapoint" hello"with value" world"into the cloudasanevent of type
"greeting" 10 times
    deviceCli.connect()

whileTrue:
    #Get Sensor Data from DHT11

    temp=random.randint(0,100)
    humid=random.randint(0,100)
```

```
    visi=random.randint(0,100)

    data={'temperature'=temp,'humidity'=humid,'visibility'=visi}
    #printdata
    defmyOnPublishCallback(): print("Publishedtemperature=%sC"%temp,"humidity=%s%%"
%humid,"visibility=%s%%"%visi,"IoTBMWatson")

        success=deviceCli.publishEvent("IoTSensor","json",data,qos=0,on_publish=my
OnPublishCallback)
        ifnotsuccess:
            print("NotconnectedtoIoTF")time.sleep(1)

            deviceCli.commandCallback=myCommandCallback

            #Disconnect the device and application from the cloud
            deviceCli.disconnect()
```