DEVELOP A PYTHON SCRIPT

Date	15 November 2022
Team ID	PNT2022TMID34030
Project Name	Smart Waste Management System For Metropolitan Cities
Maximum Marks	4 Marks

PYTHON CODE

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

#Provide your IBM Watson Device Credentials
organization = "nw3318"
deviceType = "123"
deviceId = "1234567"
authMethod = "token"
authToken = "12345678"

# Initialize GPIO

def myCommandCallback(cmd):
print("Command received: %s" % cmd.data['command'])
status=cmd.data['command']
if status=="lighton":
print ("led is on")
elif status == "lightoff":
print("led is off")
elif status == "motoron":
print("motor is on")
elif status == "motoroff":
```

```
print("motor is off")
else :
print ("please send proper command")

#print(cmd)


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

except Exception as e:
print("Caught exception connecting device: %s" % str(e))
sys.exit()

# Connect and send a datapoint "hello" with value "world" into the cloud as an event of type "greeting" 10 times
deviceCli.connect()

while True:
#Get Sensor Data from DHT11

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

data = { 'temp' : temp, 'humid': humid, 'soilmoist': soilmoist }
#print data
def myOnPublishCallback():
print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % humid,"Soilmoisture = %s %%" % soilmoist, "to IBM Watson")

success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=myOnPublishCallback)
if not success:
print("Not connected to IoTF")
time.sleep(10)
```

```
deviceCli.commandCallback = myCommandCallback

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