SPRINT-2

Team Id	**PNT2022TMID16477**
Project Name	**Smart Farmer-IoT enabled smart farming application**
TEAM	**KANAGARAJ.P(TL)** **AKASH.R(™)** **MANOJKUMAR.R(™)** **MOUNIESH.M.K(™)**

1.Python to generate random numbers for the Temperature ,Humidity and Soil_Moisture.

Code:

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

#Provide your IBM Watson Device Credentials
organization = "mwjyar"
deviceType = "abcd"
deviceId = "12345"
authMethod = "token"
authToken = "12345678"

# Initialize GPIO
```

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


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(90,110)
        Humid=random.randint(60,100)
```

```
        moist=random.randint(100,180)

        data = { 'temp' : temp, 'Humid': Humid, 'moist' :
moist}
        #print data
        def myOnPublishCallback():
                print ("Published temp = %s C" % temp, "Humid
= %s %%" % Humid, "moist= %s %%" % moist, "to
IBM Watson")

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

        deviceCli.commandCallback = myCommandCallback

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

PYTHON CODE :

Maincodeforproject.py - C:/Users/kanag/AppData/Local/Programs/Python/Python37/Maincodeforproject.py (3.7.0)

File Edit Format Run Options Window Help

```
authToken = "12345678"

# Initialize GPIO
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="motoron":
        print ("motor is on")
    elif status == "motoroff":
        print ("motor is off")
    else :
        print ("please send proper command")

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(90,110)
        Humid=random.randint(60,100)
        moist=random.randint(100,180)

        data = { 'temp' : temp, 'Humid': Humid, 'moist' : moist}
        #print data
        def myOnPublishCallback():
            print ("Published temp = %s C" % temp, "Humid = %s %%" % Humid, "moist= %s %%" % moist, "to IBM Watson")

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

        deviceCli.commandCallback = myCommandCallback

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

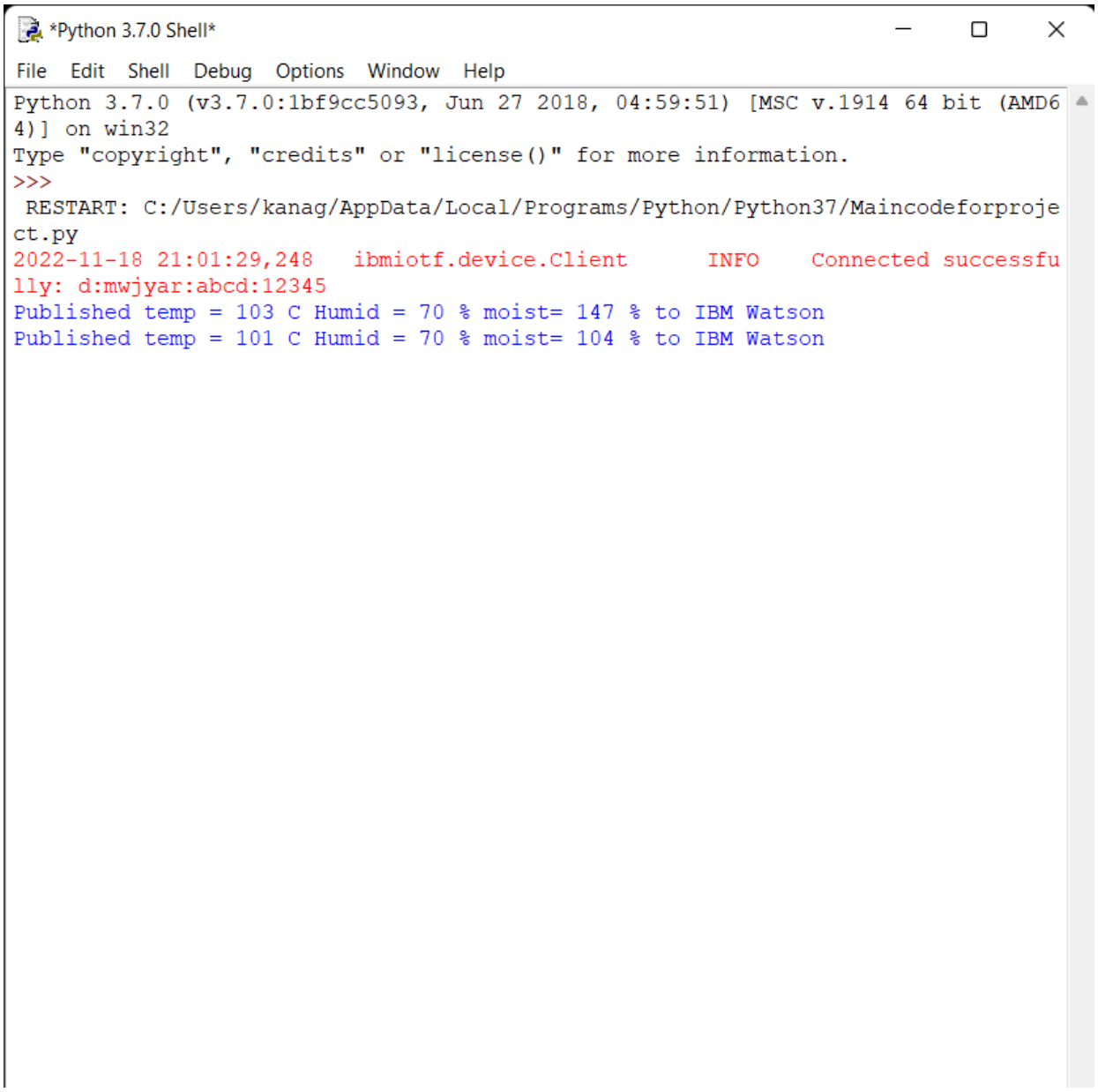
• • •

Maincodeforproject.py - C:/Users/kanag/AppData/Local/Programs/Python/Python37/Maincodeforproject.py (3.7.0)

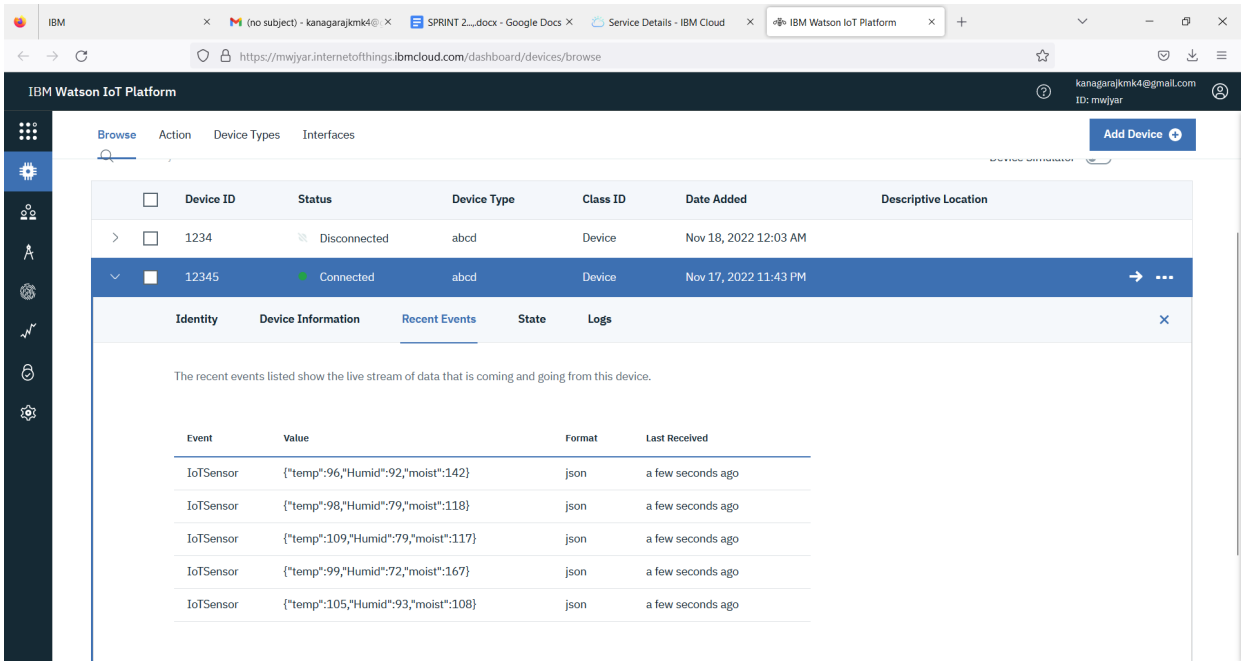
File Edit Format Run Options Window Help

```
authToken = "12345678"

# Initialize GPIO
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="motoron":
        print ("motor is on")
    elif status == "motoroff":
        print ("motor is off")
    else :
        print ("please send proper command")

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(90,110)
        Humid=random.randint(60,100)
        moist=random.randint(100,180)

        data = { 'temp' : temp, 'Humid': Humid, 'moist' : moist}
        #print data
        def myOnPublishCallback():
            print ("Published temp = %s C" % temp, "Humid = %s %%" % Humid, "moist= %s %%" % moist, "to IBM Watson")

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

        deviceCli.commandCallback = myCommandCallback

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

RESULT:



IBM WATSON IoT PLATFORM:



Our code is running Successfully…..