SPRINT-2

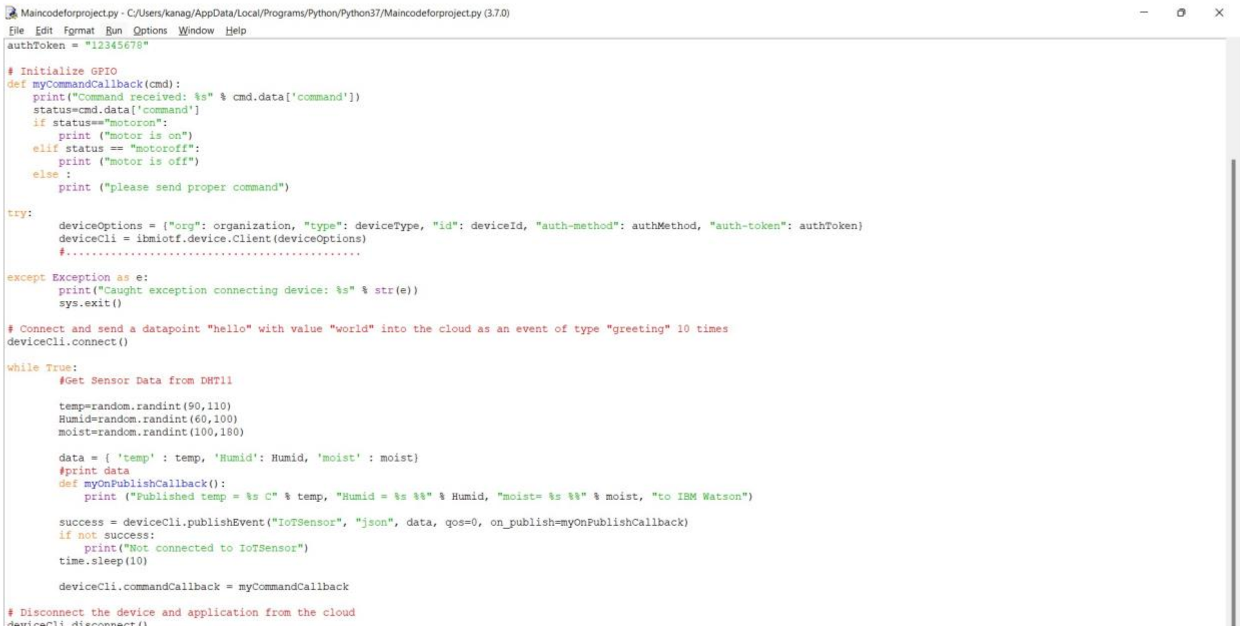
Team Id	PNT2022TMID16510
Project Name	Smart Farmer-IoT enabled smart farming application
TEAM	MUTHUKUMAR.V(TL) SURESH BABU G.S(™) KARTHICK (TM) MANOJ KUMAR(™)

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

Code:

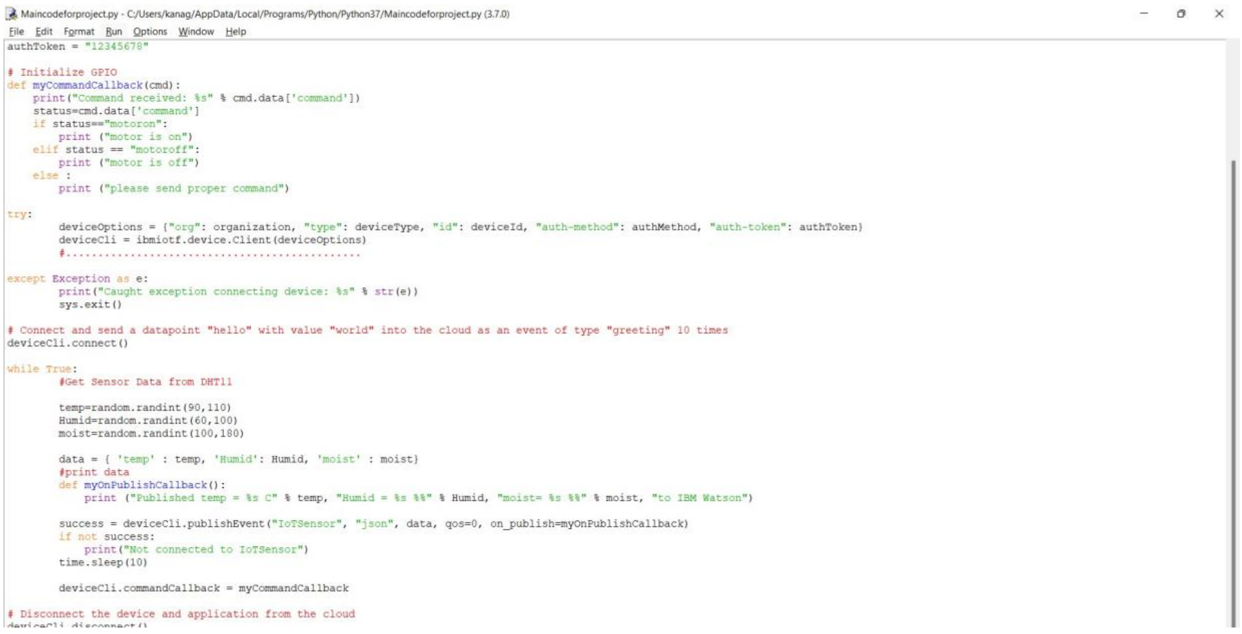
```
import time import sys
import ibmiotf.application
import ibmiotf.device
import random

#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)

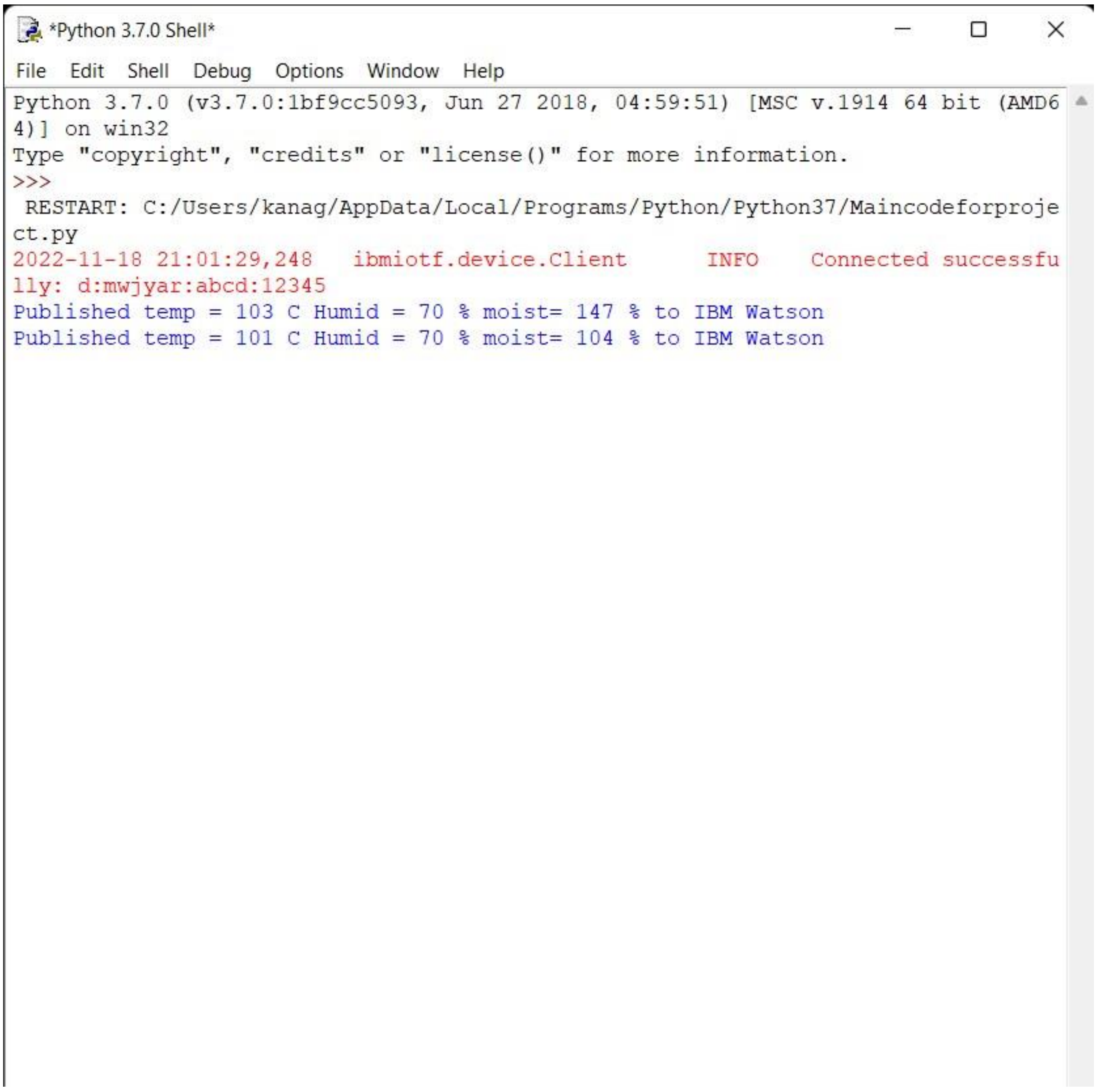
```
authToken = "12345678"

# 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()
```

…



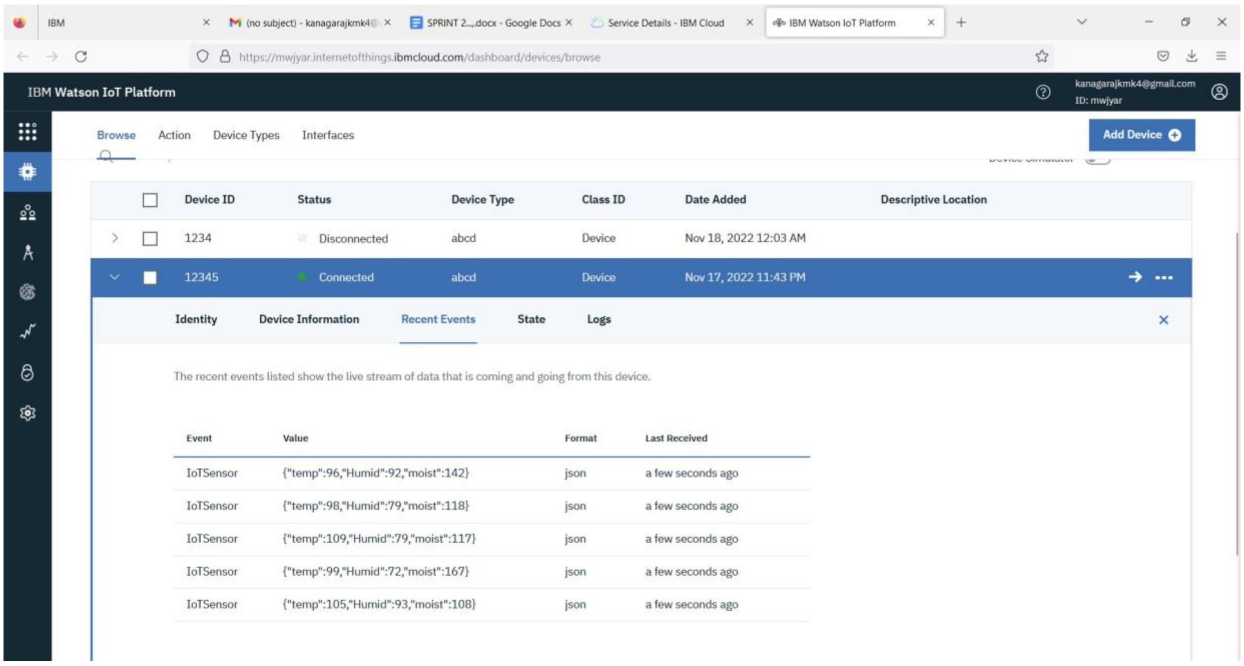
```
authToken = "12345678"

# 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:



```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD6
4)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 
 RESTART: C:/Users/kanag/AppData/Local/Programs/Python/Python37/Maincodeforproje
ct.py
2022-11-18 21:01:29,248   ibmiotf.device.Client      INFO    Connected successfu
lly: d:mwjyar:abcd:12345
Published temp = 103 C Humid = 70 % moist= 147 % to IBM Watson
Published temp = 101 C Humid = 70 % moist= 104 % to IBM Watson
```

IBM WATSON IoT PLATFORM:



	Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
	1234	Disconnected	abcd	Device	Nov 18, 2022 12:03 AM	
	12345	Connected	abcd	Device	Nov 17, 2022 11:43 PM	

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
IoTSensor	{"temp":96,"Humid":92,"moist":142}	json	a few seconds ago
IoTSensor	{"temp":98,"Humid":79,"moist":118}	json	a few seconds ago
IoTSensor	{"temp":109,"Humid":79,"moist":117}	json	a few seconds ago
IoTSensor	{"temp":99,"Humid":72,"moist":167}	json	a few seconds ago
IoTSensor	{"temp":105,"Humid":93,"moist":108}	json	a few seconds ago

Our code is running Successfully…..