Develop A Python Script

To

Publish AndSubscribe

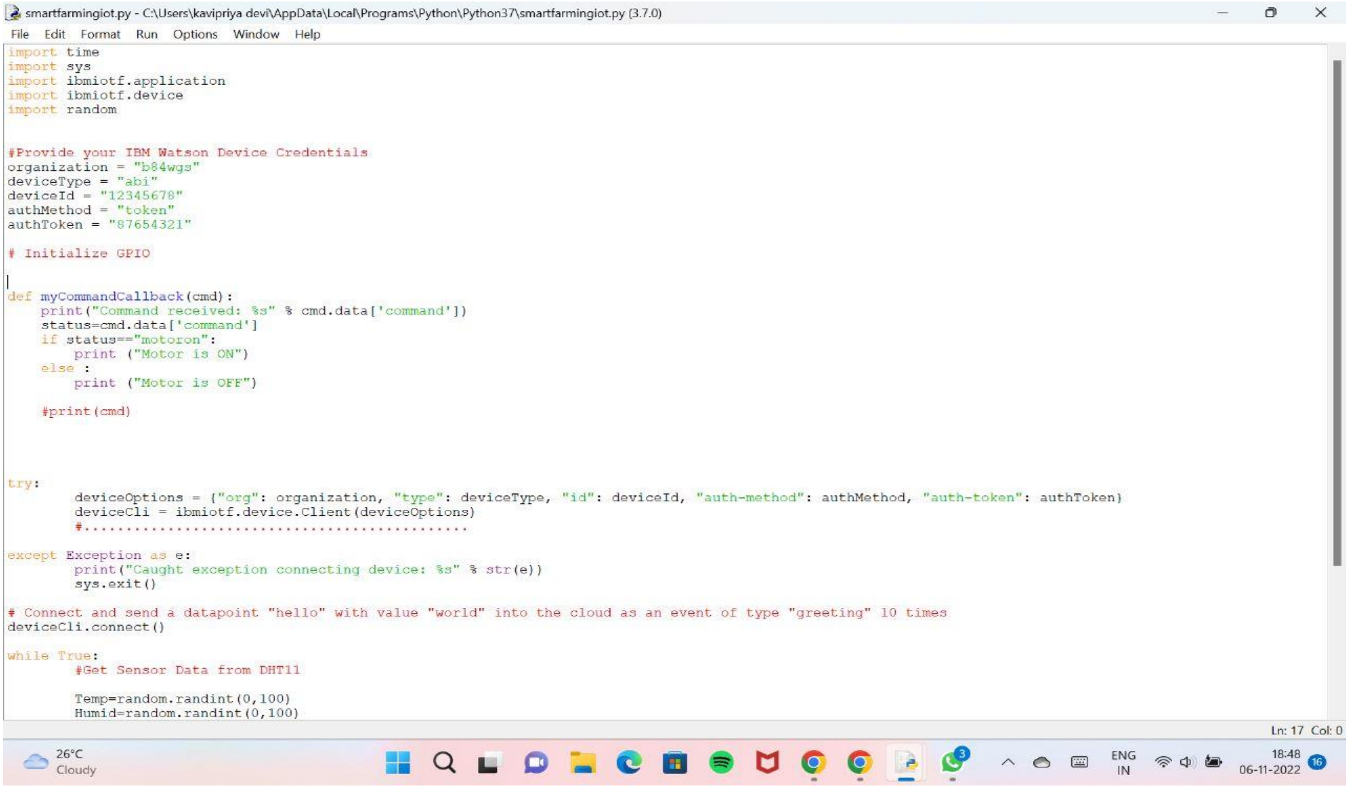
To IBM IoT Platform

TEAM ID: PNT2022TMID34551

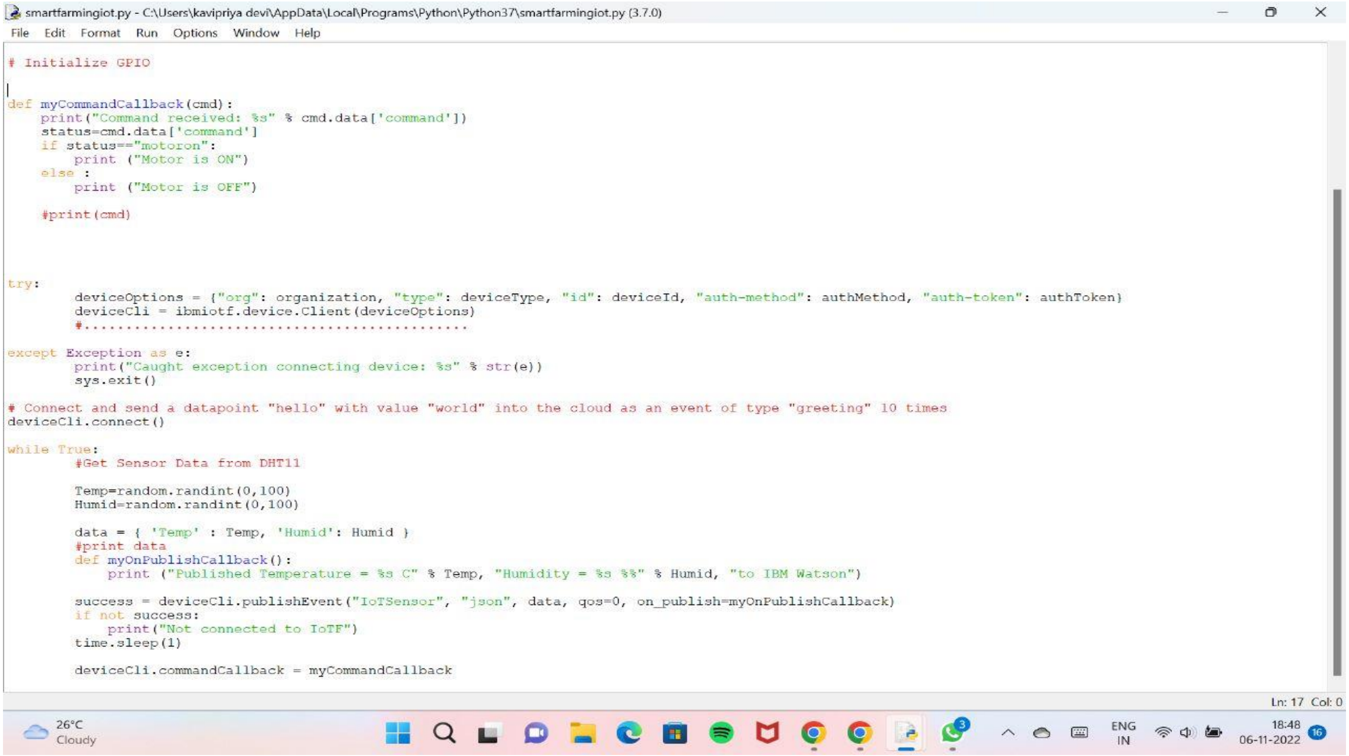
Team Leader :	ANUSUYA M	: (961819106301)
Team member :	ABIGAIL DERALSHYA S	: (961819106002)
Team member :	BAVITHRA S	: (961819106010)
Team member :	KAVIPRIYA DEVI V	: (961819106032)

Code in Python IDLE:

PROGRAM:


smartfarmingiot.py - C:\Users\kavipriya devi\AppData\Local\Programs\Python\Python37\smartfarmingiot.py (3.7.0)
File Edit Format Run Options Window Help
Ln: 17 Col: 0
26°C
Cloudy
ENG
IN
18:48
06-11-2022

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


#Provide your IBM Watson Device Credentials
organization = "b84wgs"
deviceType = "abi"
deviceId = "12345678"
authMethod = "token"
authToken = "87654321"

# Initialize GPIO


def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="motoron":
        print ("Motor is ON")
    else :
        print ("Motor is OFF")

    #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)
```


smartfarmingiot.py - C:\Users\kavipriya devi\AppData\Local\Programs\Python\Python37\smartfarmingiot.py (3.7.0)
File Edit Format Run Options Window Help
Ln: 17 Col: 0
26°C
Cloudy
ENG
IN
18:48
06-11-2022

```
# Initialize GPIO


def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="motoron":
        print ("Motor is ON")
    else :
        print ("Motor is OFF")

    #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)

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

        success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=myOnPublishCallback)
        if not success:
            print("Not connected to IoTF")
        time.sleep(1)

        deviceCli.commandCallback = myCommandCallback
```

Program used in the code:

```
import time

import sys

import ibmiotf.application

import ibmiotf.device

import random

#Provide your IBM Watson Device Credentials

organization = "b84wgs"

deviceType = "abi"

deviceId = "12345678"

authMethod = "token"

authToken = "87654321"


# Initialize GPIO

def myCommandCallback(cmd):

    print("Command received: %s" % cmd.data['command'])

    status=cmd.data['command']

    if status=="lighton":

        print ("led is on")

    else :

        print ("led is off")

    #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)

    data = { 'temp' : temp, 'Humid': Humid }

    #print data

    def myOnPublishCallback():

        print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % Humid, "to IBM Watson")

    success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=myOnPublishCallback)

    if not success:

        print("Not connected to IoTF")

    time.sleep

    deviceCli.commandCallback = myCommandCallback
```