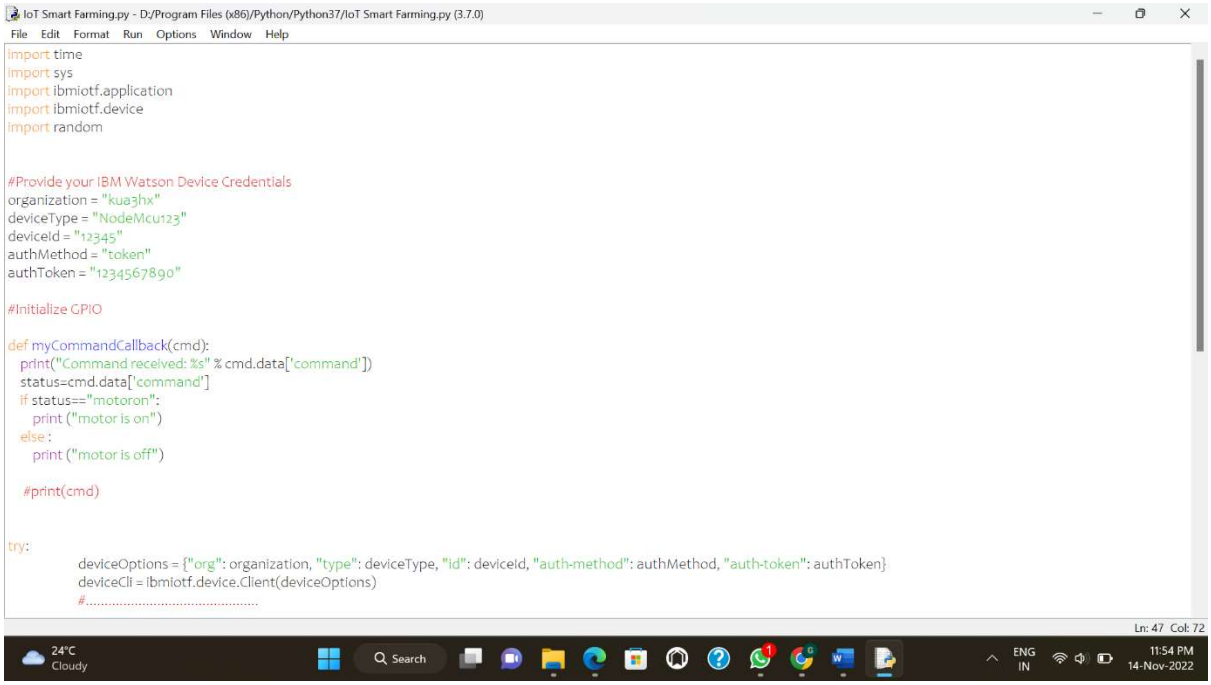
Python Code

TITLE	**Smart Farmer-IOT Enabled Smart Farming Application**
DOMAIN NAME	INTERNET OF THINGS
TEAM ID	PNT2022TMID23830
Leader Name	GOWSALYA L
Team Members Name	DEEPIKA B K MEGAVARSHINI G MONISHA N
MENTOR NAME	THIRUPPATHI M

CODE:

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

#Provide your IBM Watson Device Credentials
organization = "kua3hx"
deviceType = "NodeMcu123"
deviceId = "12345"
authMethod = "token"
authToken = "1234567890"

#Initialize GPIO

def myCommandCallback(cmd):
   print("Command received: %s" % cmd.data['command'])
   status=cmd.data['command']
   if status=="Motor ON":
      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)
        Moist=random.randint(0,100)

        data = { 'temperature' : temp, 'humidity': Humid , 'moisture': Moist}
        #print data
        def myOnPublishCallback():
            print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % Humid, "Moisture = %s %%" % Moist, "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
#Disconnect the device and the application from the cloud
deviceCli.disconnect()
```



Receiving commands from IBM cloud using Python program:

Observations & Results: