Smart Farmer-IOT Enabled Smart Farming Application

SPRINT DELIVERY- 4

TITLE	Smart Farmer-IOT Enabled Smart Farming Application
DOMAIN NAME	INTERNET OF THINGS
TEAM ID	PNT2022TMID21357

Receiving commands from IBM cloud using Python program

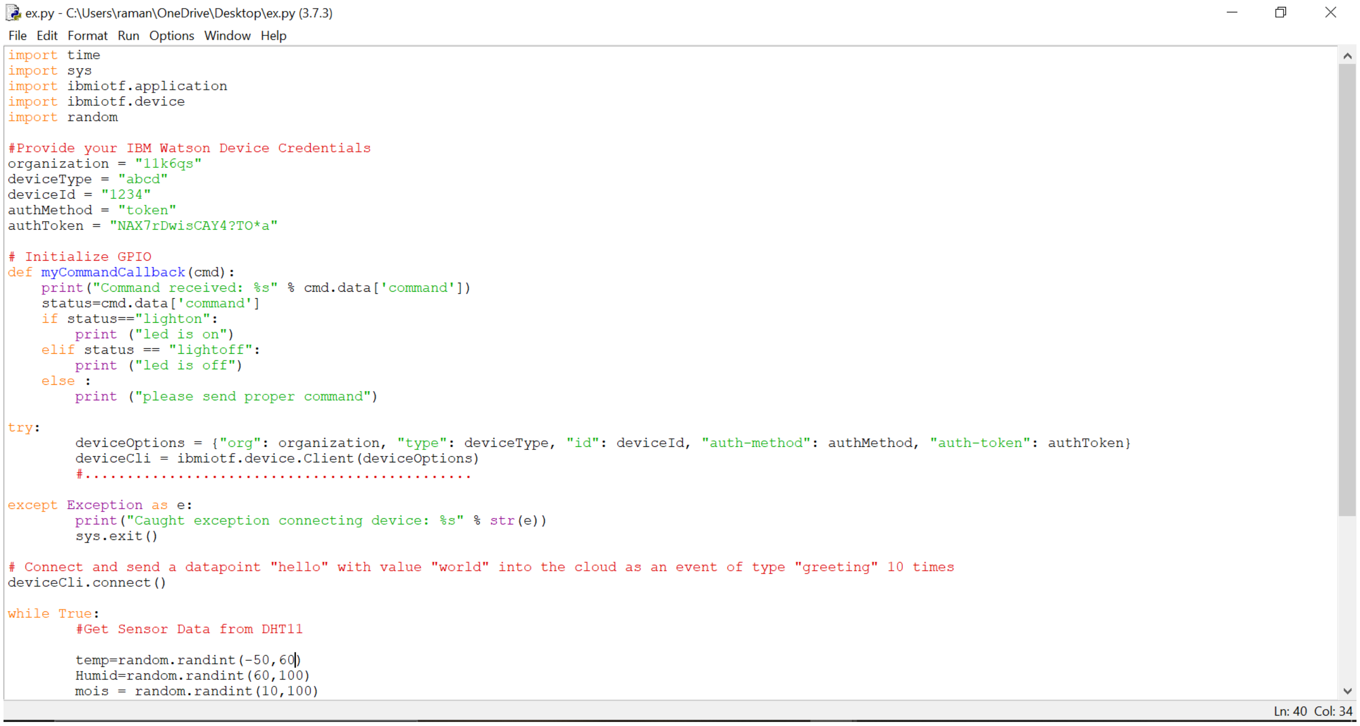
```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random

#Provide your IBM Watson Device Credentials
organization = "11k6qs"
deviceType = "abcd"
deviceId = "1234"
authMethod = "token"
authToken = "NAX7rDwisCAY4?TO*a"

# 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")
    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(-50,60)
    Humid=random.randint(60,100)
    mois = random.randint(10,100)

    data = { 'temp' : temp, 'Humid': Humid ,'mois' : mois}
    #print data
    def myOnPublishCallback():
        print ("Published Temperature = %s C" % temp, "Humidity = %s " % Humid, "Moisture
= %s %%" % mois + "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)
```

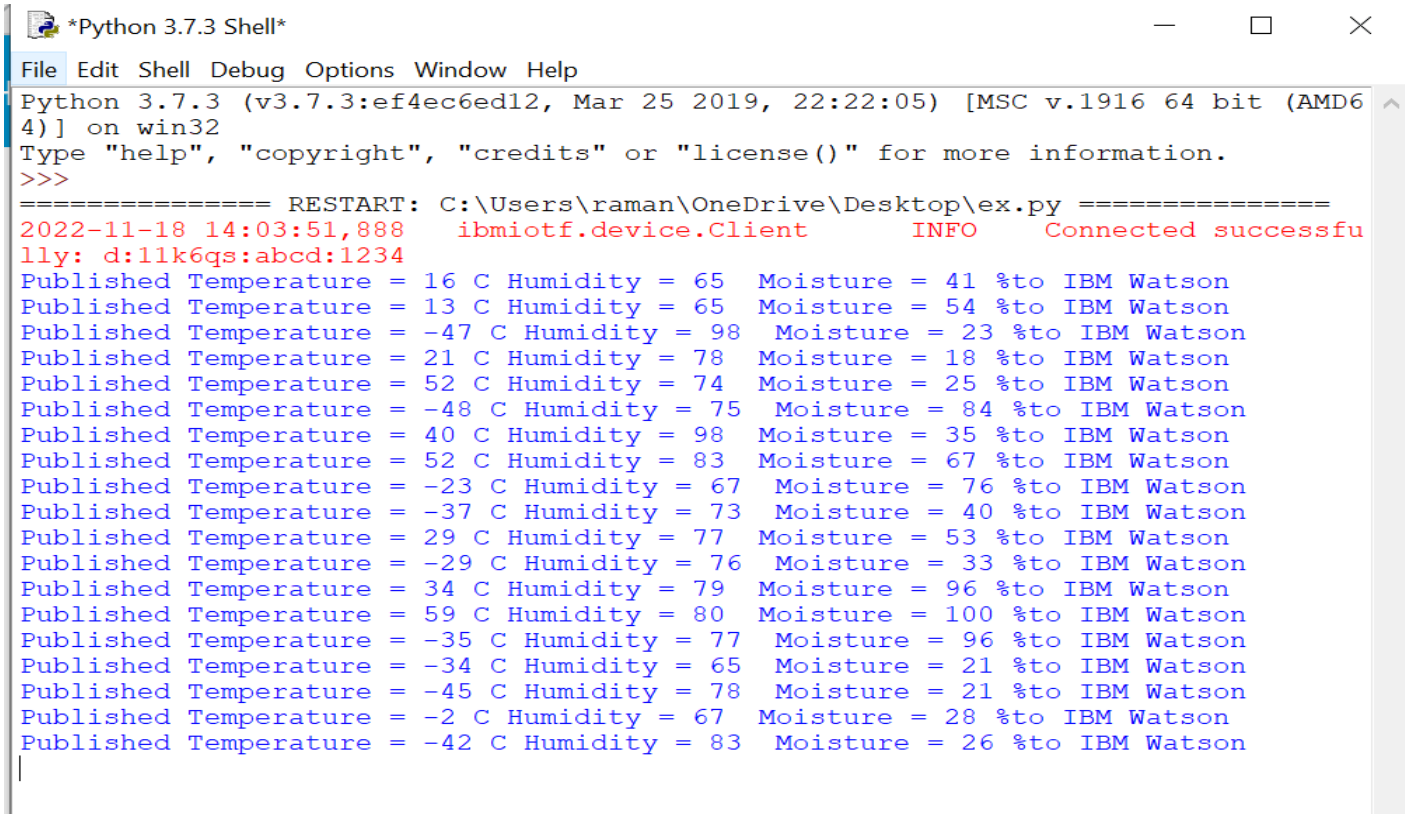
deviceCli.commandCallback = myCommandCallback

# Disconnect the device and application from the cloud

deviceCli.disconnect()

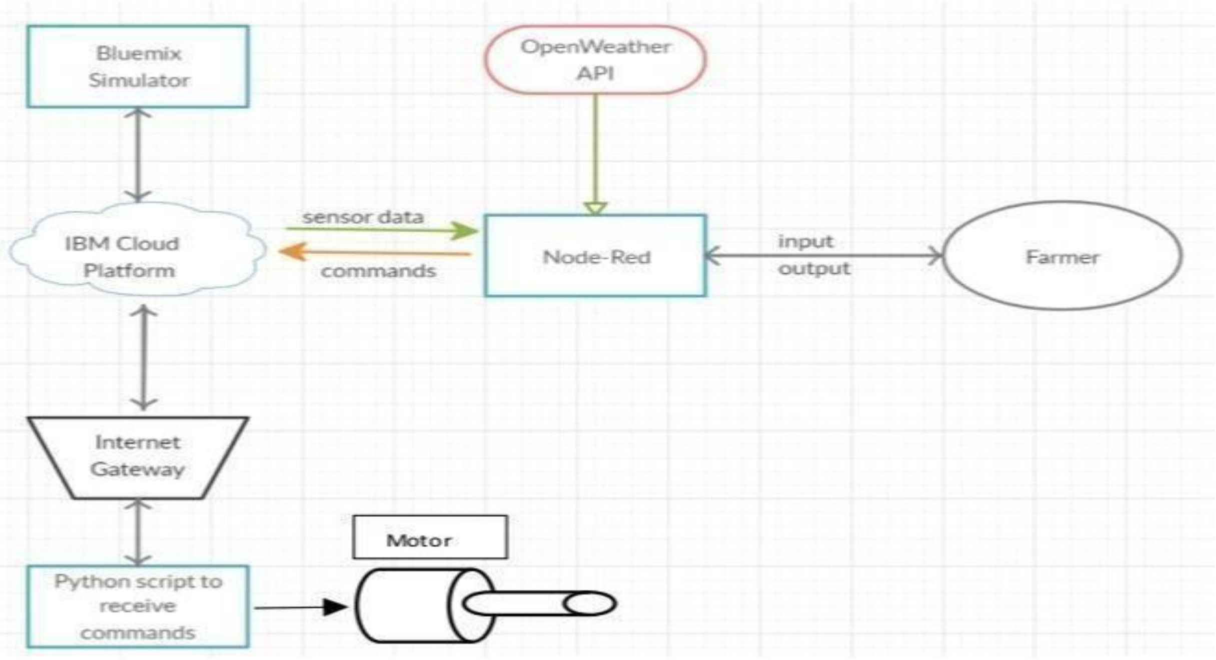

ex.py - C:\Users\raman\OneDrive\Desktop\ex.py (3.7.3)
File Edit Format Run Options Window Help
Ln: 40 Col: 34

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

#Provide your IBM Watson Device Credentials
organization = "11k6qs"
deviceType = "abcd"
deviceId = "1234"
authMethod = "token"
authToken = "NAX7rDwisCAY4?TO*a"

# 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")
    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(-50,60)
        Humid=random.randint(60,100)
        mois = random.randint(10,100)
```

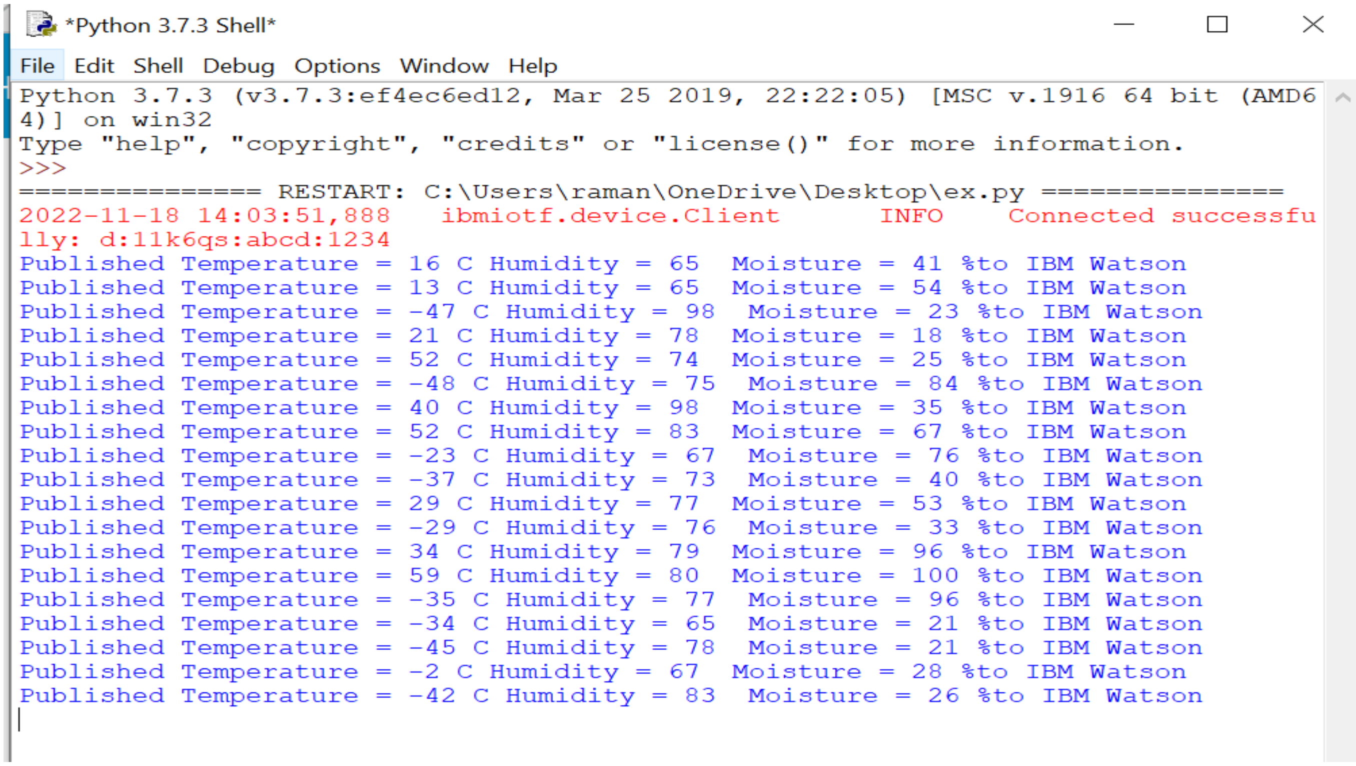


```
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 22:22:05) [MSC v.1916 64 bit (AMD6
4)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=============== RESTART: C:\Users\raman\OneDrive\Desktop\ex.py ===============
2022-11-18 14:03:51,888   ibmiotf.device.Client      INFO    Connected successfu
lly: d:11k6qs:abcd:1234
Published Temperature = 16 C Humidity = 65  Moisture = 41 %to IBM Watson
Published Temperature = 13 C Humidity = 65  Moisture = 54 %to IBM Watson
Published Temperature = -47 C Humidity = 98  Moisture = 23 %to IBM Watson
Published Temperature = 21 C Humidity = 78  Moisture = 18 %to IBM Watson
Published Temperature = 52 C Humidity = 74  Moisture = 25 %to IBM Watson
Published Temperature = -48 C Humidity = 75  Moisture = 84 %to IBM Watson
Published Temperature = 40 C Humidity = 98  Moisture = 35 %to IBM Watson
Published Temperature = 52 C Humidity = 83  Moisture = 67 %to IBM Watson
Published Temperature = -23 C Humidity = 67  Moisture = 76 %to IBM Watson
Published Temperature = -37 C Humidity = 73  Moisture = 40 %to IBM Watson
Published Temperature = 29 C Humidity = 77  Moisture = 53 %to IBM Watson
Published Temperature = -29 C Humidity = 76  Moisture = 33 %to IBM Watson
Published Temperature = 34 C Humidity = 79  Moisture = 96 %to IBM Watson
Published Temperature = 59 C Humidity = 80  Moisture = 100 %to IBM Watson
Published Temperature = -35 C Humidity = 77  Moisture = 96 %to IBM Watson
Published Temperature = -34 C Humidity = 65  Moisture = 21 %to IBM Watson
Published Temperature = -45 C Humidity = 78  Moisture = 21 %to IBM Watson
Published Temperature = -2 C Humidity = 67  Moisture = 28 %to IBM Watson
Published Temperature = -42 C Humidity = 83  Moisture = 26 %to IBM Watson
```

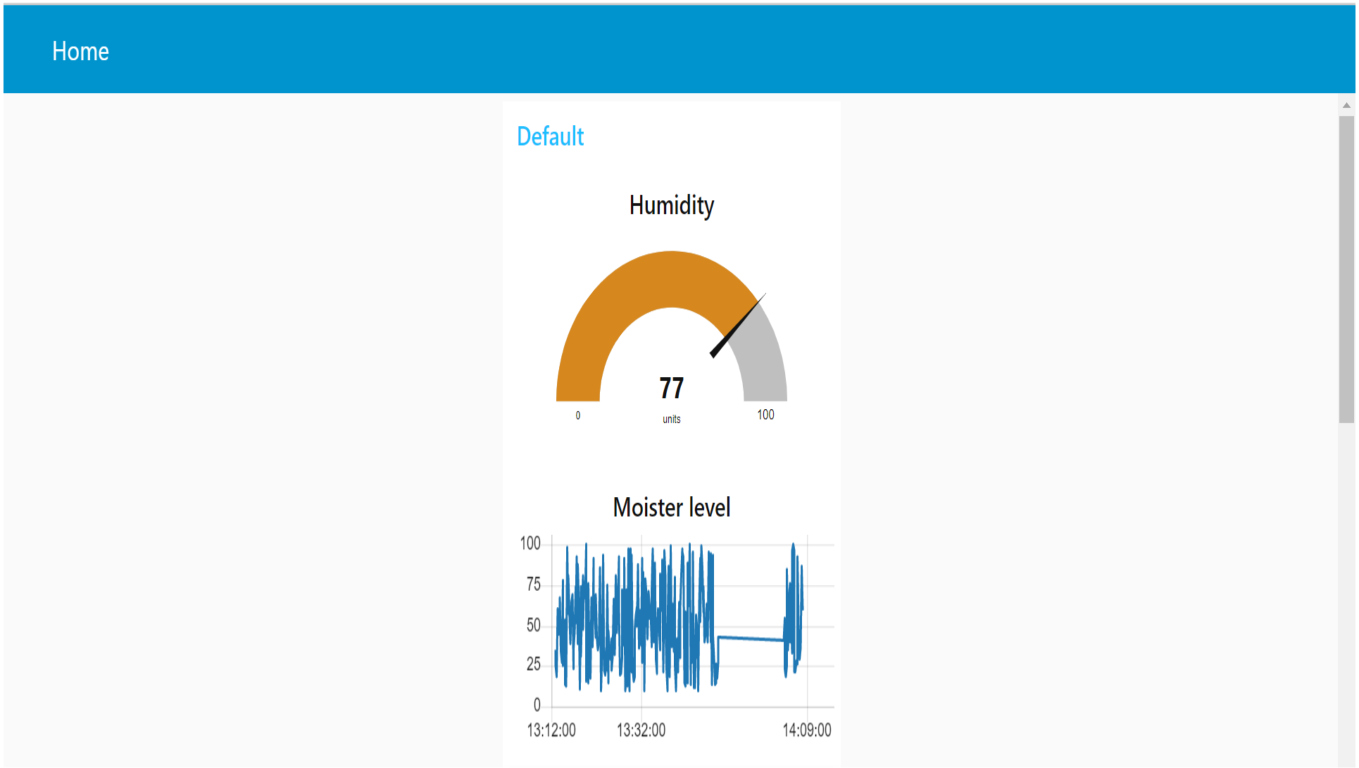
Flow Chart

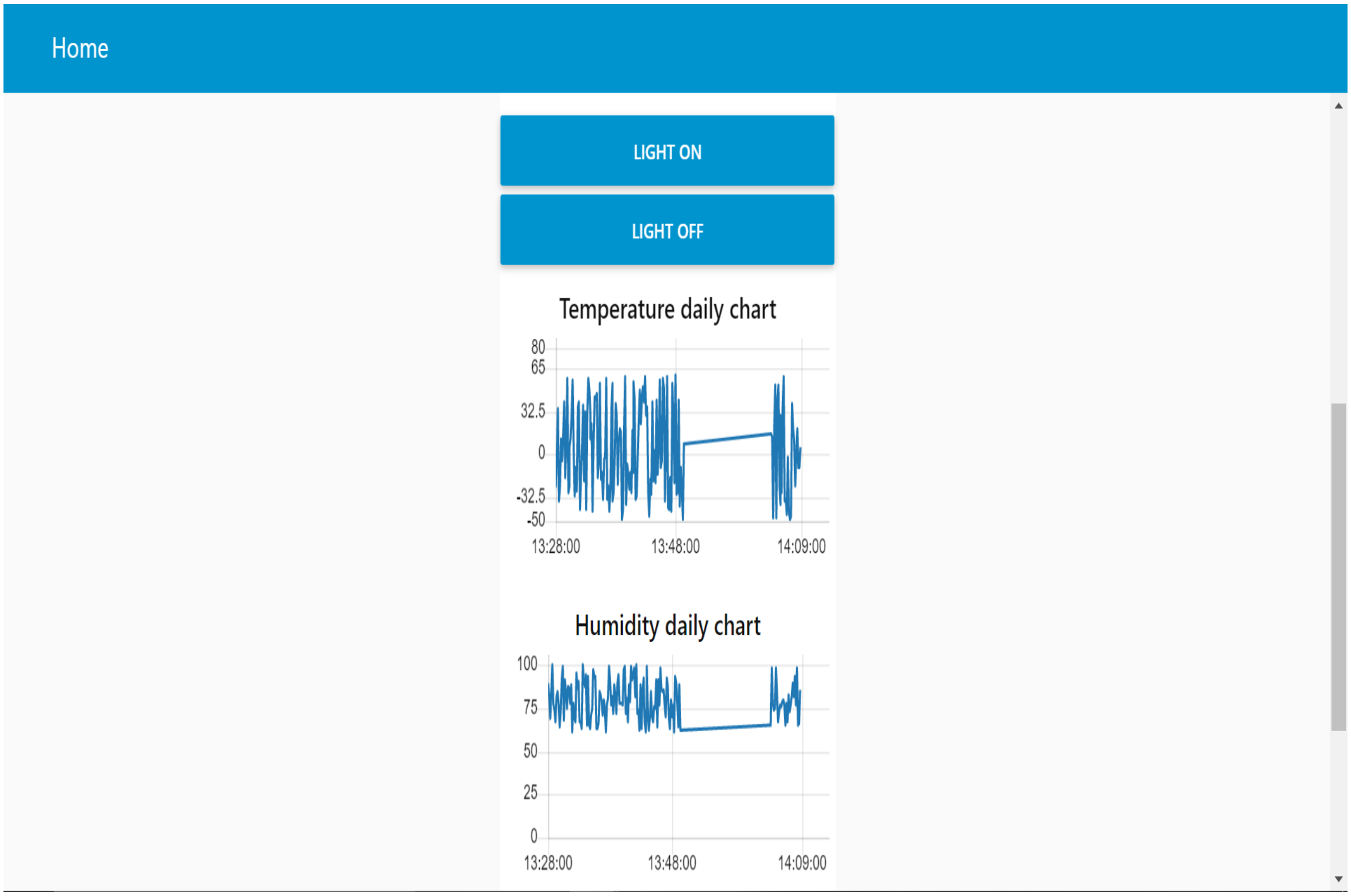


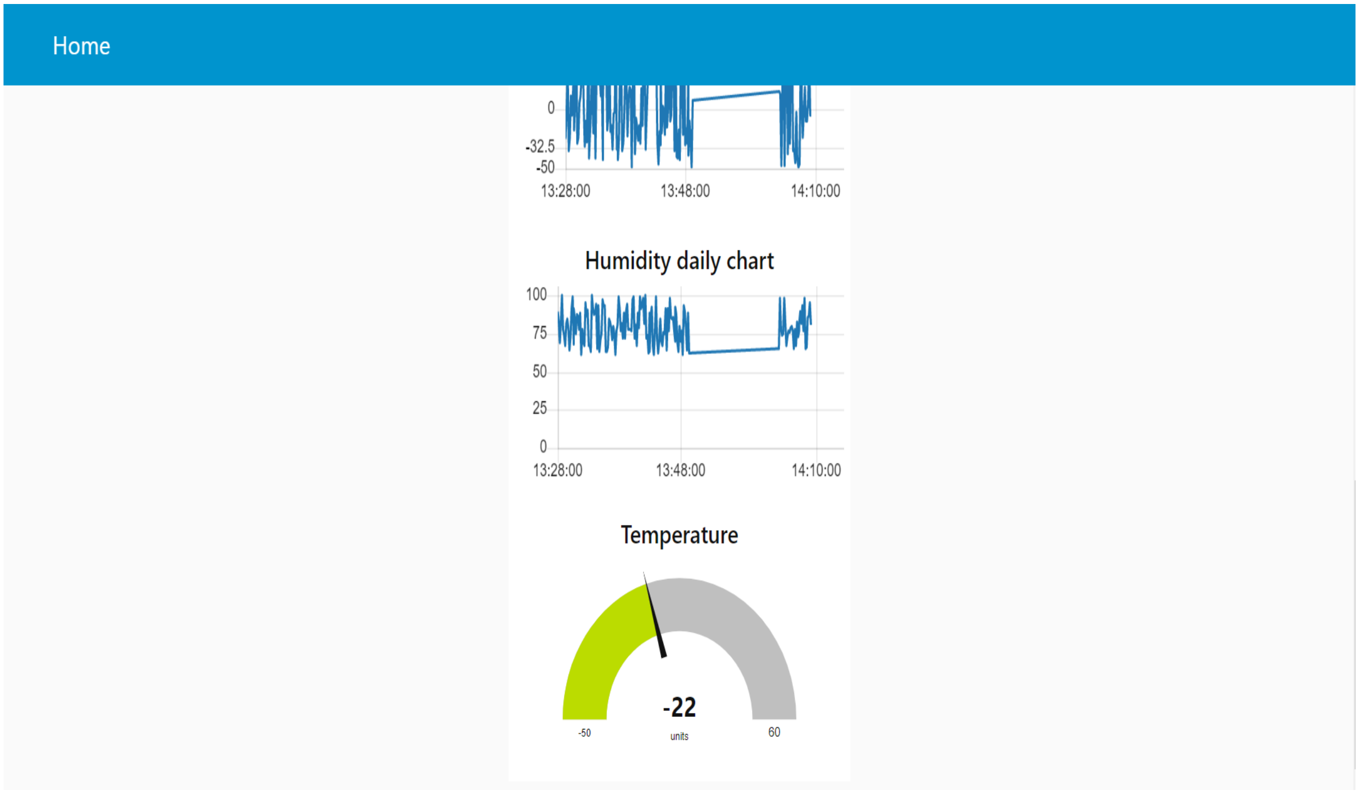
Observations & Results



```
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 22:22:05) [MSC v.1916 64 bit (AMD6
4)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=============== RESTART: C:\Users\raman\OneDrive\Desktop\ex.py ===============
2022-11-18 14:03:51,888   ibmiotf.device.Client      INFO    Connected successfu
lly: d:11k6qs:abcd:1234
Published Temperature = 16 C Humidity = 65  Moisture = 41 %to IBM Watson
Published Temperature = 13 C Humidity = 65  Moisture = 54 %to IBM Watson
Published Temperature = -47 C Humidity = 98  Moisture = 23 %to IBM Watson
Published Temperature = 21 C Humidity = 78  Moisture = 18 %to IBM Watson
Published Temperature = 52 C Humidity = 74  Moisture = 25 %to IBM Watson
Published Temperature = -48 C Humidity = 75  Moisture = 84 %to IBM Watson
Published Temperature = 40 C Humidity = 98  Moisture = 35 %to IBM Watson
Published Temperature = 52 C Humidity = 83  Moisture = 67 %to IBM Watson
Published Temperature = -23 C Humidity = 67  Moisture = 76 %to IBM Watson
Published Temperature = -37 C Humidity = 73  Moisture = 40 %to IBM Watson
Published Temperature = 29 C Humidity = 77  Moisture = 53 %to IBM Watson
Published Temperature = -29 C Humidity = 76  Moisture = 33 %to IBM Watson
Published Temperature = 34 C Humidity = 79  Moisture = 96 %to IBM Watson
Published Temperature = 59 C Humidity = 80  Moisture = 100 %to IBM Watson
Published Temperature = -35 C Humidity = 77  Moisture = 96 %to IBM Watson
Published Temperature = -34 C Humidity = 65  Moisture = 21 %to IBM Watson
Published Temperature = -45 C Humidity = 78  Moisture = 21 %to IBM Watson
Published Temperature = -2 C Humidity = 67  Moisture = 28 %to IBM Watson
Published Temperature = -42 C Humidity = 83  Moisture = 26 %to IBM Watson
```


Home
Default
Humidity
77
units
0
100
Moister level
100
75
50
25
0
13:12:00
13:32:00
14:09:00


Home
LIGHT ON
LIGHT OFF
Temperature daily chart
80
65
32.5
0
-32.5
-50
13:28:00
13:48:00
14:09:00
Humidity daily chart
100
75
50
25
0
13:28:00
13:48:00
14:09:00



Advantages & Disadvantages Advantages:

- Farms can be monitored and controlled remotely.
- Increase in convenience to farmers.
- Less labor cost.
- Better standards of living.

Disadvantages:

- Lack of internet/connectivity issues.
- Added cost of internet and internet gateway infrastructure.
- Farmers wanted to adapt the use of Mobile App.

Conclusion

Thus, the objective of the project to implement an IoT system in order to help farmers to control and monitor their farms has been implemented successfully.