

IBM NALAIYATHIRAN
SMART FARMER-IOT ENABLED SMART FARMING
APPLICATION

SPRINT-4

Title	Smart farmer-IoT enabled smart farming application
Domain	Internet of Things
Team ID	PNT2022TMID44170
Project Name	Project – Smart Farmer-IoT Enabled smartFarming Application

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 = "48qubr" deviceType =
"hasnarahah09" deviceId = "hasna09" authMethod
= "token" authToken = "
glo4Y*)BMQqN8HR9T9"
```

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)
    Mois=random. Randint(20,120)    data =
    { 'temp' : temp, 'Humid': Humid , 'Mois':
    Mois}

    #print data    def
myOnPublishCallback():

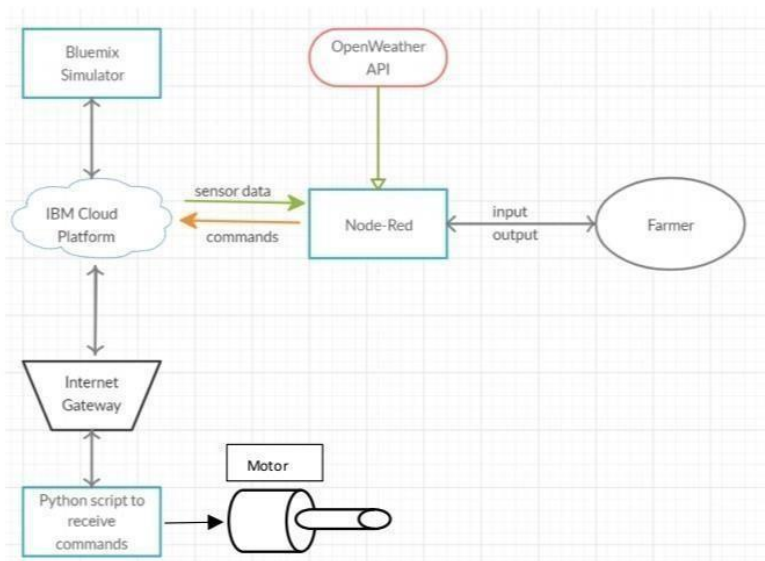
    print ("Published Temperature = %s C" % temp, "Humidity = %s
%%" % Humid, "Moisture =%s deg c" % Mois "to IBM Watson")
    success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,
on_publish=myOnPublishCallback)    if not success:
    print("Not connected to IoTF")
    time.sleep(10)

    deviceCli.commandCallback = myCommandCallback #
Disconnect the device and application from the cloud
deviceCli.disconnect()

```

```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\ELCOT\Downloads\ibmiotpublishsubscribe.py =====
2022-11-07 20:01:24,074 ibmiotf.device.Client INFO Connected successfully: d:157uf3:abcd:7654321
Published Moisture = 90 deg C Temperature = 96 C Humidity = 76 % to IBM Watson
Published Moisture = 102 deg C Temperature = 110 C Humidity = 68 % to IBM Watson
Published Moisture = 45 deg C Temperature = 99 C Humidity = 100 % to IBM Watson
Command received: motoron
motor is on
Published Moisture = 77 deg C Temperature = 91 C Humidity = 85 % to IBM Watson
Published Moisture = 73 deg C Temperature = 94 C Humidity = 86 % to IBM Watson
Command received: motoroff
motor is off
Published Moisture = 101 deg C Temperature = 104 C Humidity = 87 % to IBM Watson
```

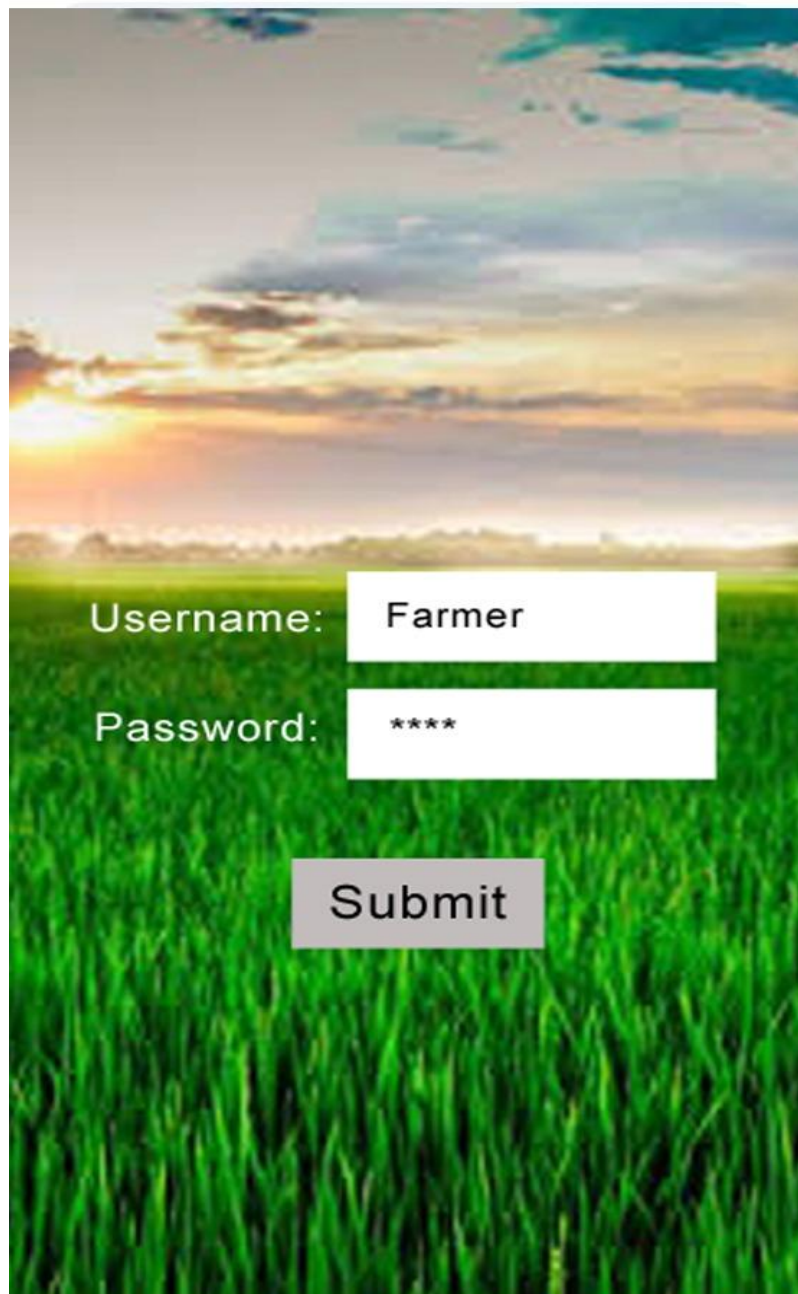
FLOW OF THE PROJECT



RESULT

11:12 AM

Bluetooth, Vibration, 4G, Signal, Wi-Fi, 60%



11:12 AM



Measured Data

Moisture : 59

Temperature(c) : 90

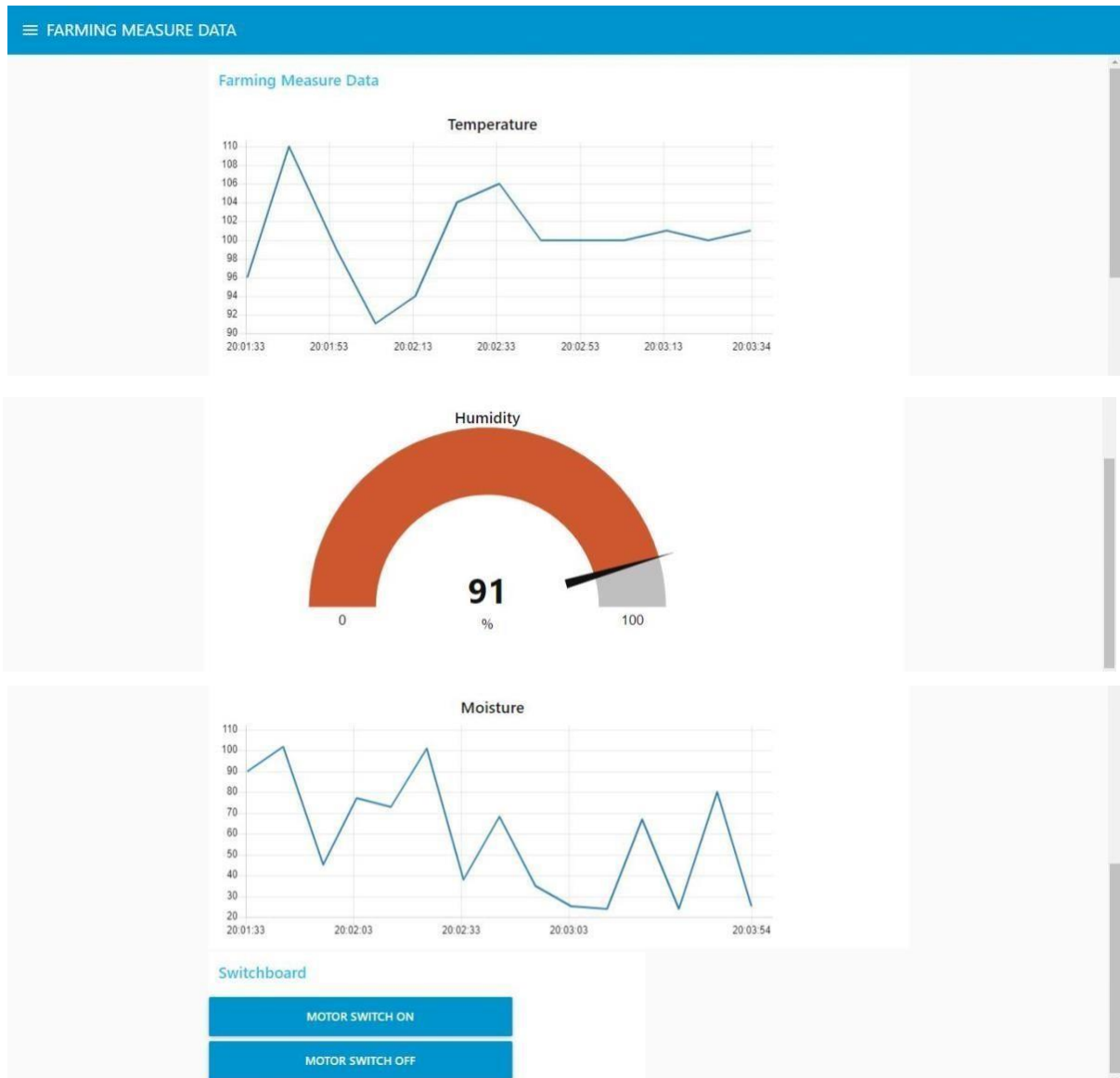
Humidity (%) : 85

Switchboard

Motor on

Motor off

OUTPUT ON THE IOT PLATFORM



ADVANTAGES

- Increased Production
- Water Conservation
- Real-Time Data and Production Insight
- Lowered Operation Costs
- Accurate Farm and Field Evaluation
- Improved Livestock Farming
- Reduced Environmental Footprint
- Remote Monitoring
- Equipment Monitoring

DISADVANTAGES

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

CONCLUSION

Smart farming is a wonderful option if you want to save the cost of electricity. It allows for the use of solar-powered tools like pumps that save your expenditure. It is cost-effective as it somewhat reduces the spending usually generated by farmers in maintaining their capital intensive techs. Smart agriculture makes use of IOT platform to improve the process of wireless monitoring, regulation and data collection. With these inputs on your farm, all thanks to smart farming, you can be sure of high-quality crop production and delivery.

BIBLIOGRAPHY

- ✚ IBM cloud reference: <https://cloud.ibm.com/>
- ✚ IoT simulator : <https://watson-iot-sensor-simulator.mybluemix.net/>
- ✚ OpenWeather : <https://openweathermap.org/>