

Project Development Phase

Delivery Of Sprint-4

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

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 = "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()

```

```
IoT Smart Farming.py - D:/Program Files (x86)/Python/Python37/IoT Smart Farming.py (3.7.0)
File Edit Format Run Options Window Help

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

#Provide your IBM Watson Device Credentials
organization = "kua3hix"
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=="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)
    #.....

Ln: 47 Col: 72
```

```
IoT Sm 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: D:/Program Files (x86)/Python/Python37/IoT Smart Farming.py ====
2022-11-14 23:55:04.929 ibmiotf.device.Client INFO Connected successfully: dkua3hix:NodeMcu123:12345
Published Temperature = 98 C Humidity = 71 % Moisture = 26 % to IBM Watson
Published Temperature = 78 C Humidity = 81 % Moisture = 30 % to IBM Watson
Published Temperature = 47 C Humidity = 61 % Moisture = 34 % to IBM Watson
Published Temperature = 44 C Humidity = 37 % Moisture = 90 % to IBM Watson
Published Temperature = 86 C Humidity = 10 % Moisture = 74 % to IBM Watson
Published Temperature = 84 C Humidity = 88 % Moisture = 27 % to IBM Watson
Published Temperature = 72 C Humidity = 25 % Moisture = 22 % to IBM Watson
Published Temperature = 81 C Humidity = 10 % Moisture = 61 % to IBM Watson
Published Temperature = 100 C Humidity = 54 % Moisture = 23 % to IBM Watson
Published Temperature = 42 C Humidity = 87 % Moisture = 91 % to IBM Watson
Published Temperature = 18 C Humidity = 61 % Moisture = 36 % to IBM Watson
Published Temperature = 53 C Humidity = 86 % Moisture = 37 % to IBM Watson
Published Temperature = 49 C Humidity = 51 % Moisture = 61 % to IBM Watson
Published Temperature = 97 C Humidity = 57 % Moisture = 68 % to IBM Watson
Published Temperature = 72 C Humidity = 6 % Moisture = 20 % to IBM Watson

#Provide
organiza
deviceT
deviceId
authMe
authTok

#Initializ

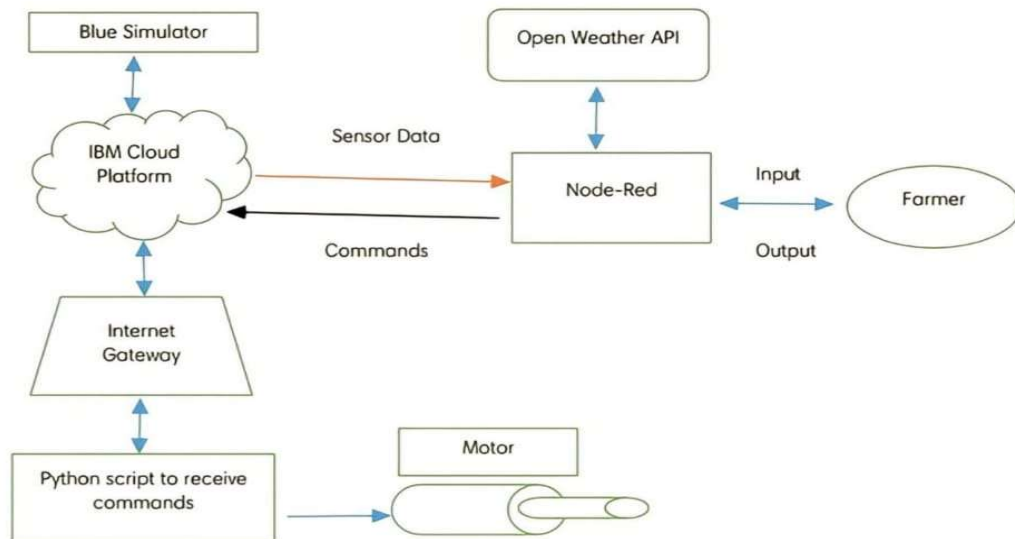
def myC
print(
status
if stat
prin
else:
prin
#prin

try:

en": authToken}
```

Flow Chart:

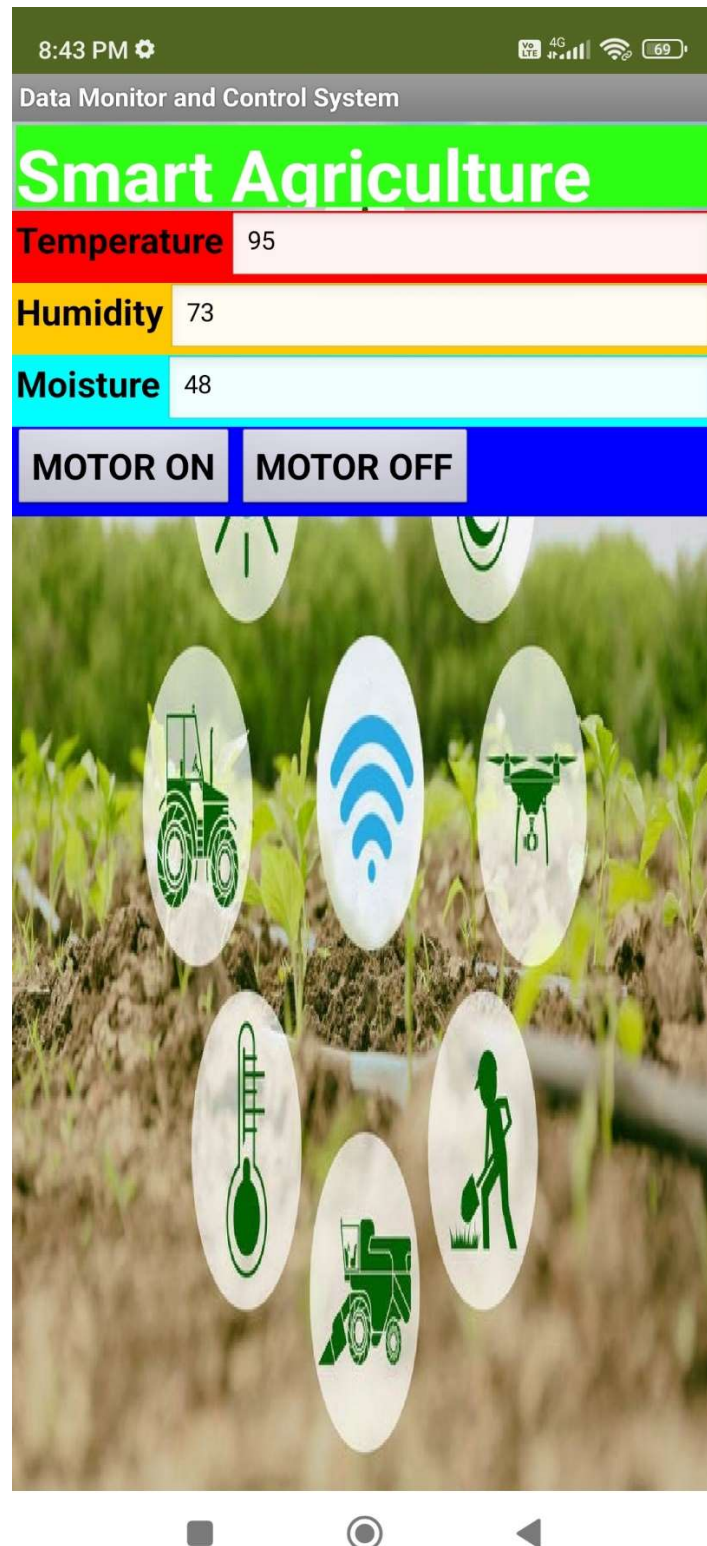
FLOW CHART



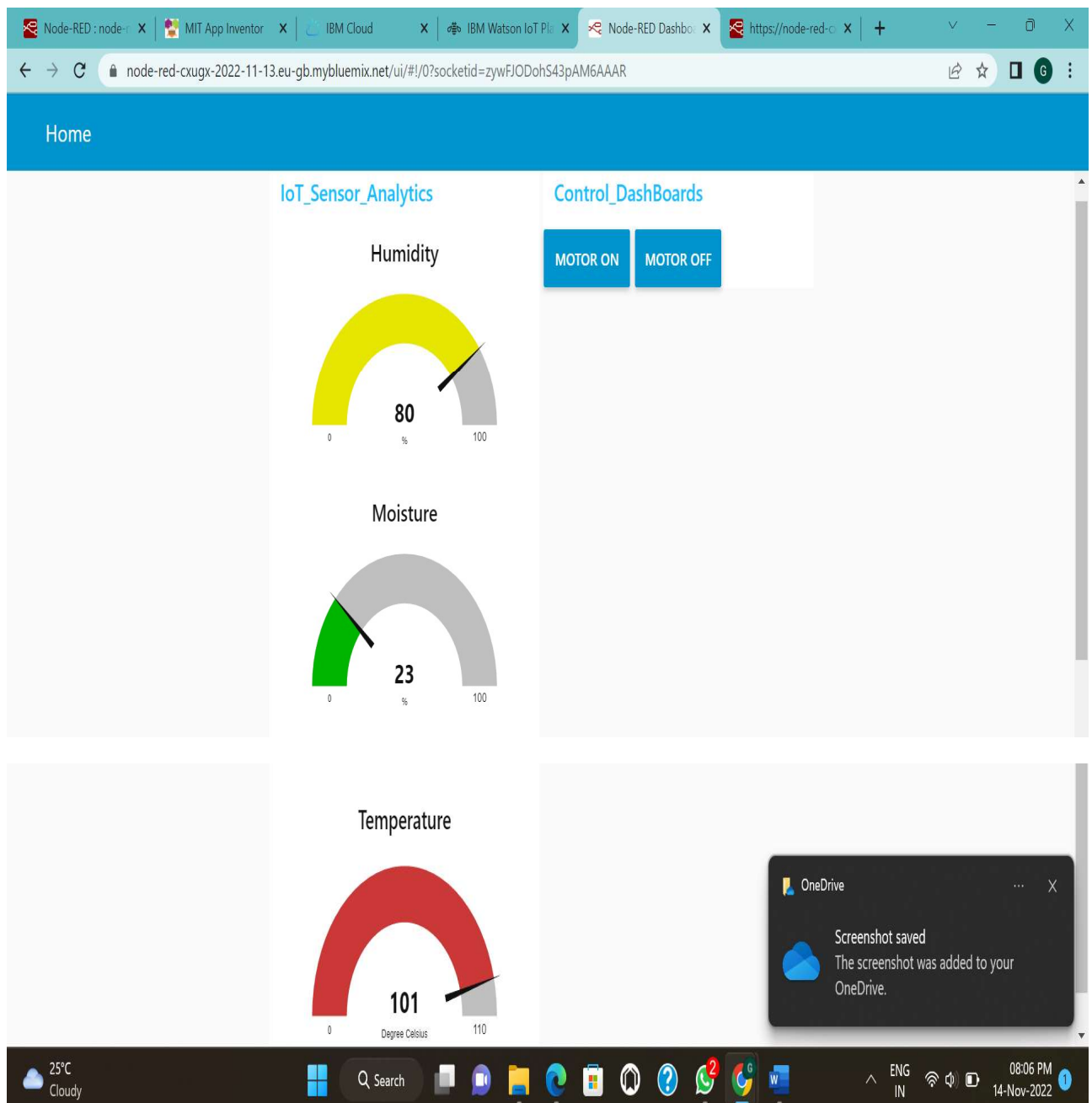
Observations & Results:

```
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: D:\Program Files (x86)\Python\Python37\IoT Smart Farming.py ====
2022-11-15 00:44:25,821 ibmiotf.device.Client INFO Connected successfully: dkuag3hx:NodeMcu123:12345
Published Temperature = 14 C Humidity = 98 % Moisture = 25 % to IBM Watson
Published Temperature = 46 C Humidity = 64 % Moisture = 94 % to IBM Watson
Published Temperature = 67 C Humidity = 7 % Moisture = 2 % to IBM Watson
Published Temperature = 33 C Humidity = 78 % Moisture = 97 % to IBM Watson
Published Temperature = 36 C Humidity = 69 % Moisture = 48 % to IBM Watson
Published Temperature = 87 C Humidity = 37 % Moisture = 28 % to IBM Watson
Command received: Motor OFF
Motor is OFF
Published Temperature = 45 C Humidity = 39 % Moisture = 98 % to IBM Watson
Published Temperature = 60 C Humidity = 73 % Moisture = 90 % to IBM Watson
Command received: Motor ON
Motor is ON
Published Temperature = 20 C Humidity = 15 % Moisture = 15 % to IBM Watson
Published Temperature = 84 C Humidity = 7 % Moisture = 0 % to IBM Watson
Published Temperature = 50 C Humidity = 24 % Moisture = 83 % to IBM Watson
Published Temperature = 100 C Humidity = 93 % Moisture = 32 % to IBM Watson
Command received: Motor ON
Motor is ON
Published Temperature = 32 C Humidity = 66 % Moisture = 29 % to IBM Watson
Command received: Motor OFF
Motor is OFF
Published Temperature = 49 C Humidity = 3 % Moisture = 98 % to IBM Watson
Published Temperature = 84 C Humidity = 88 % Moisture = 51 % to IBM Watson
Published Temperature = 8 C Humidity = 66 % Moisture = 88 % to IBM Watson
Published Temperature = 8 C Humidity = 88 % Moisture = 18 % to IBM Watson
Published Temperature = 69 C Humidity = 18 % Moisture = 49 % to IBM Watson
Published Temperature = 69 C Humidity = 18 % Moisture = 49 % to IBM Watson
```

APP DEVELOPED:



WEB UI DESIGNED:



Advantages:

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

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.