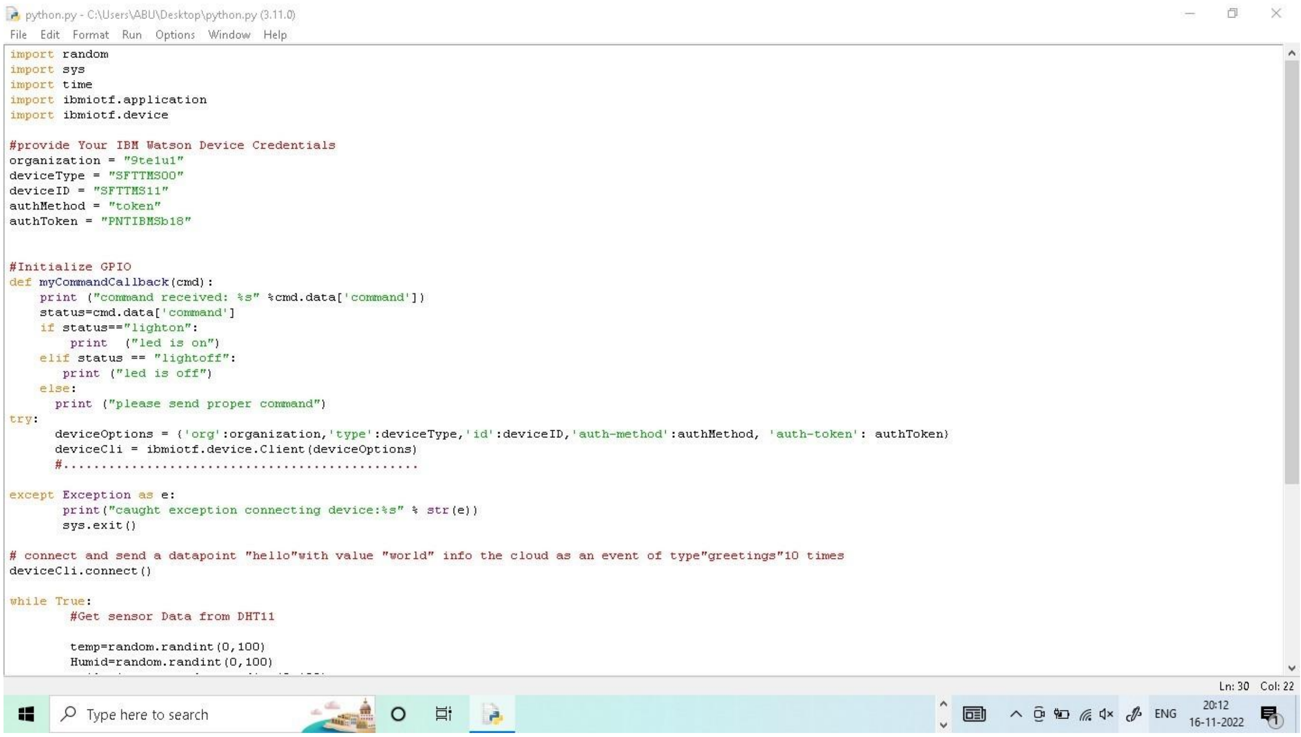
IOT ENABLED SMART FARMING APPLICATION

SPRINT DELIVERY – 4

TEAM ID : PNT2022TMID29852

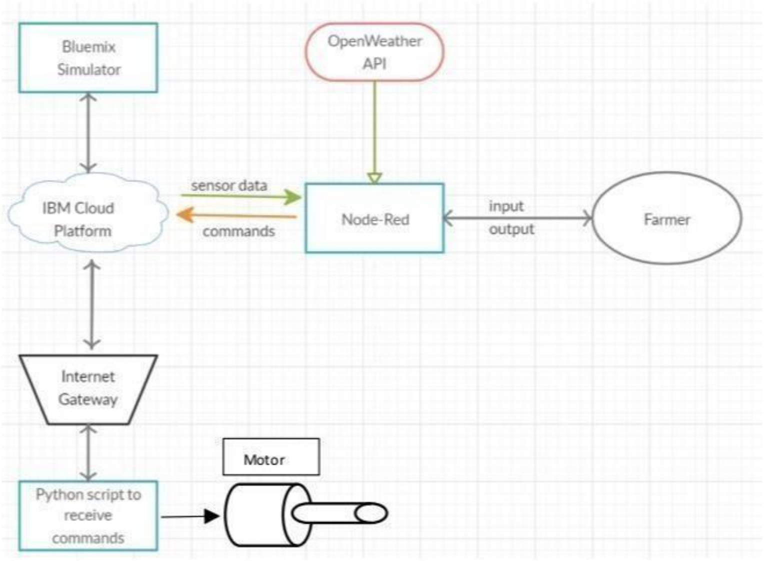
5.5 Receiving commands from IBM cloud using Python program

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

#provide Your IBM Watson Device Credentials
organization = "9te1u1"
deviceType = "SFTTMS00"
deviceID = "SFTTMS11"
authMethod = "token"
authToken = "PNTIBMSb18"


#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" info the cloud
as an event of type"greetings"10 times
deviceCli.connect()

while True:
        #Get sensor Data from DHT11

        temp=random.randint(0,100)
        Humid=random.randint(0,100)
        soilmoisture=random.randint(0,100)

        data = { 'temp' : temp, 'Humid': Humid, 'soilmoisture':soilmoisture}
        #print data
        def myOnPublishCallback():
            print ("published Temperature = %s C" %  temp, "Humidity = is %s
%%" % Humid, "soilmoisture= is %s %%" % soilmoisture,"to IBM
Watson")


        success = deviceCli.publishEvent("IOTSensor",
"json",data,qos=0,on_publish=myOnPublishCallback)
        if not success:
            print("Not connected to IOTF")
        time.sleep(5)


        deviceCli.commandCallback = myCommandCallback

# Disconnect the device and application from the cloud
deviceCli.disconnect()
```

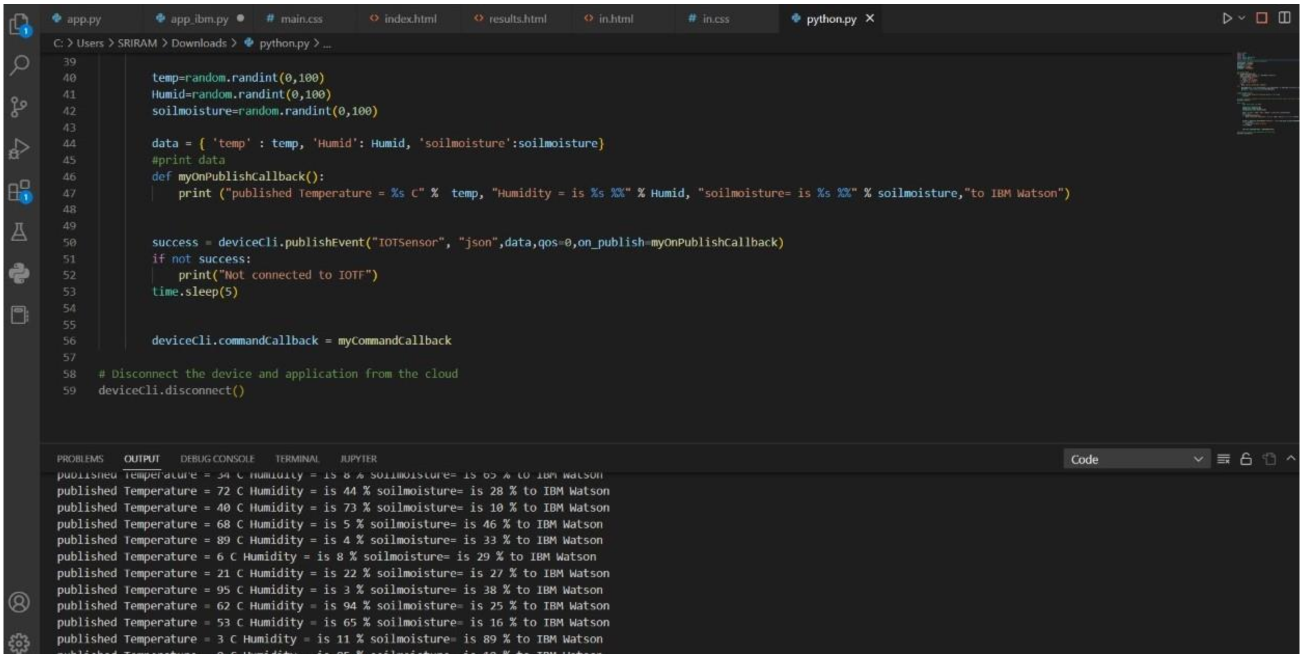

python.py - C:\Users\ABU\Desktop\python.py (3.11.0)
File Edit Format Run Options Window Help
Ln: 30 Col: 22
Type here to search
ENG
20:12
16-11-2022

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

#provide Your IBM Watson Device Credentials
organization = "9te1u1"
deviceType = "SFTTMS00"
deviceID = "SFTTMS11"
authMethod = "token"
authToken = "PNTIBMSb18"


#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" info the cloud as an event of type"greetings"10 times
deviceCli.connect()

while True:
        #Get sensor Data from DHT11

        temp=random.randint(0,100)
        Humid=random.randint(0,100)
```

FLOW CHART

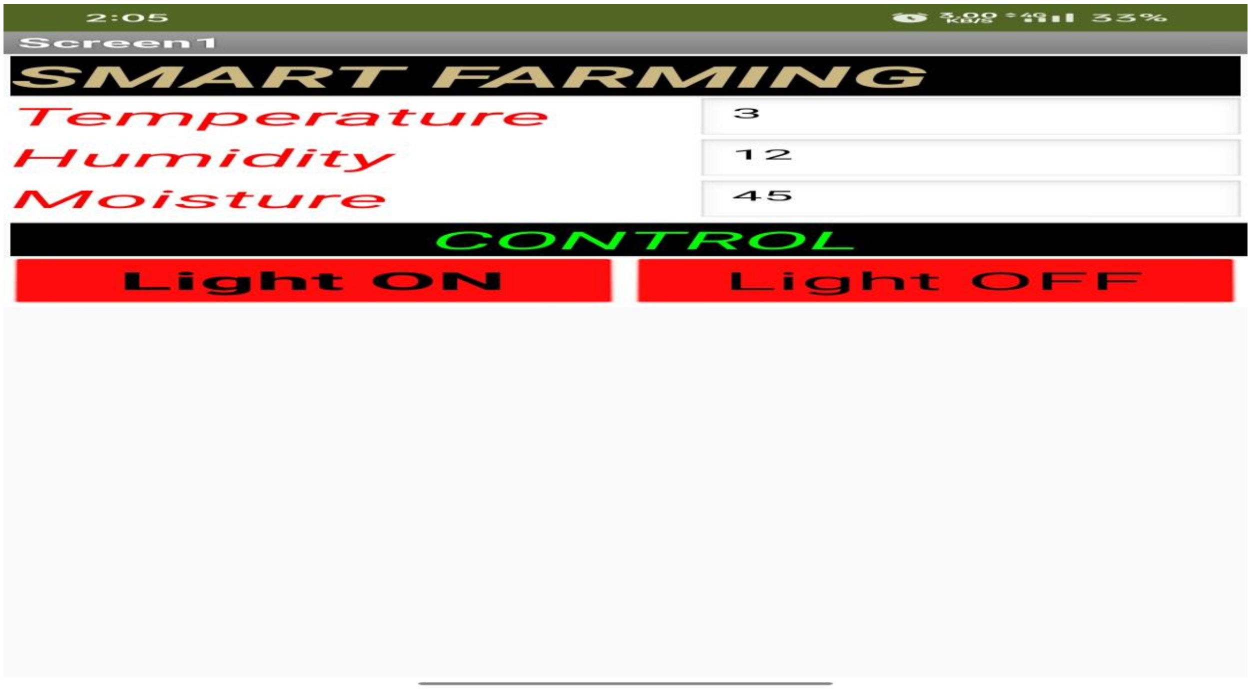


6. Observations & Results



```python

        temp=random.randint(0,100)
        Humid=random.randint(0,100)
        soilmoisture=random.randint(0,100)

        data = { 'temp' : temp, 'Humid': Humid, 'soilmoisture':soilmoisture}
        #print data
        def myOnPublishCallback():
            print ("published Temperature = %s C" %  temp, "Humidity = is %s %%" % Humid, "soilmoisture= is %s %%" % soilmoisture,"to IBM Watson")


        success = deviceCli.publishEvent("IOTSensor", "json",data,qos=0,on_publish=myOnPublishCallback)
        if not success:
            print("Not connected to IOTF")
        time.sleep(5)


        deviceCli.commandCallback = myCommandCallback

# Disconnect the device and application from the cloud
deviceCli.disconnect()
```

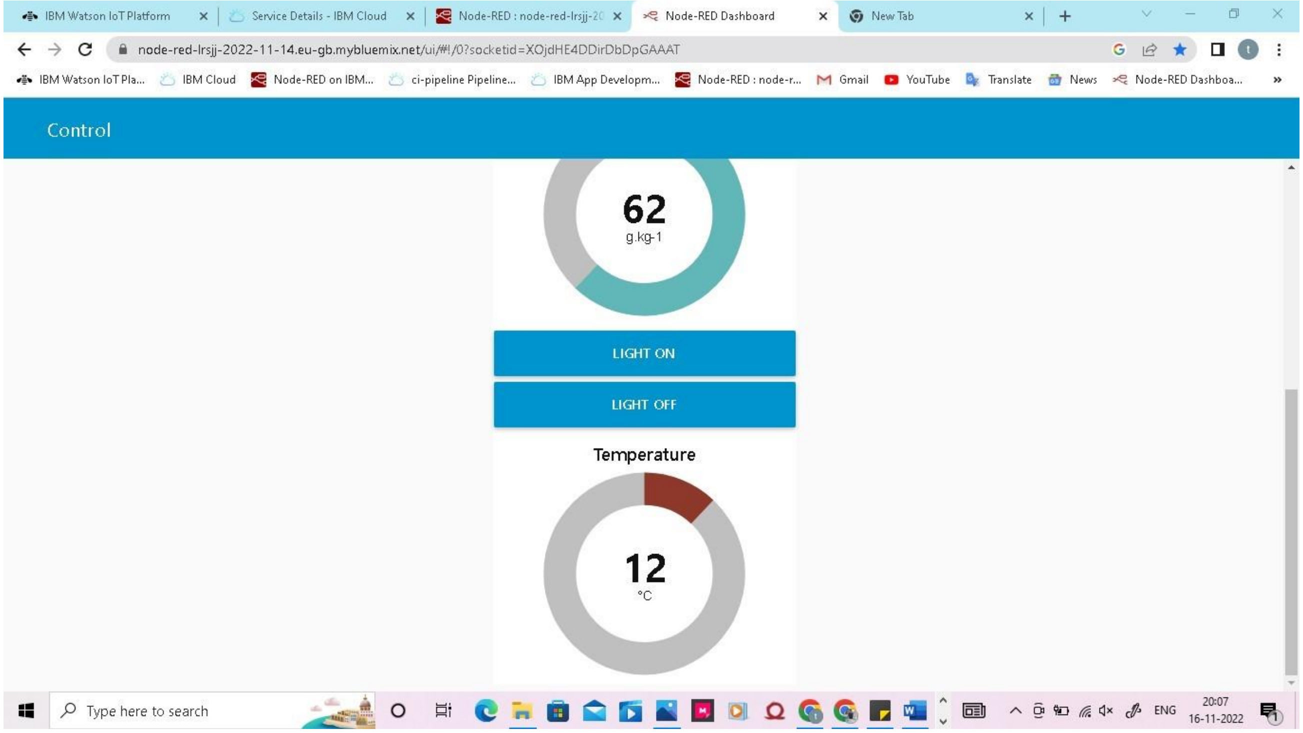
```
published Temperature = 34 C Humidity = is 8 % soilmoisture= is 65 % to IBM Watson
published Temperature = 72 C Humidity = is 44 % soilmoisture= is 28 % to IBM Watson
published Temperature = 40 C Humidity = is 73 % soilmoisture= is 10 % to IBM Watson
published Temperature = 68 C Humidity = is 5 % soilmoisture= is 46 % to IBM Watson
published Temperature = 89 C Humidity = is 4 % soilmoisture= is 33 % to IBM Watson
published Temperature = 6 C Humidity = is 8 % soilmoisture= is 29 % to IBM Watson
published Temperature = 21 C Humidity = is 22 % soilmoisture= is 27 % to IBM Watson
published Temperature = 95 C Humidity = is 3 % soilmoisture= is 38 % to IBM Watson
published Temperature = 62 C Humidity = is 94 % soilmoisture= is 25 % to IBM Watson
published Temperature = 53 C Humidity = is 65 % soilmoisture= is 16 % to IBM Watson
published Temperature = 3 C Humidity = is 11 % soilmoisture= is 89 % to IBM Watson
```



SMART FARMING

Temperature	3
Humidity	12
Moisture	45

CONTROL



node-red-lrsjj-2022-11-14.eu-gb.mybluemix.net/ui/#!/0?socketid=XOjdHE4DDirDbDpGAAAT

62
g.kg-1

Temperature

12
°C

7. 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.

9.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.

10.Bibliography

IBM cloud reference: https://cloud.ibm.com/

IoT simulator : https://watson-iot-sensor-simulator.mybluemix.net/

OpenWeather : https://openweathermap.org/