SOURCE CODE

Date	25 NOV 2022
Team ID	PNT2022TMID08084
Project Name	Project -Smart farmer-IOT enabled smartFarming Application

PYTHON CODE:

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

# Provide your IBM Watson Device Credentials
organization = "x0fxss"  # replace the ORG ID
deviceType = "smartfarmapplication"  # replace the Device type wi
deviceId = "98712345"  # replace Device ID
authMethod = "token"
authToken = "1234567890"  # Replace the authtoken
# Initialize GPIO

# Receives Command from Node-red


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")
    elif status == "motorthirty":
        print("motor is on for 30 minutes")
        print("motor Started")
        for i in range(1,31):
            print("%d minutes to stop"%(30-i))  # use time.sleep(60) for delay of
one minute in each iteration
        print("motor stopped")


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 like
"{'temp:45, 'Humid':57, 'soilmoisture':76}"
with value in the name of event "IoTSensor"'

deviceCli.connect()

while True:
    # Get Sensor Data from DHT11
    # Get Sensor Data from Soil Moisture Sensor

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

    data = {'temp': temp, 'Humid': Humid, 'soilmoisture': soilmoisture}
    # print data

    def myOnPublishCallback():
        print("Published Temperature = %s C" % temp, "Humidity = %s %%"
%
              Humid, "soilmoisture = %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)  # sends a datapoint with delay of 5 seconds

    deviceCli.commandCallback = myCommandCallback

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