A Project Report

on

Smart Agriculture System Based on IoT

by

NITIN KUMAR PATEL
Internship id - SB43119

Submitted to : Smartinternz (The Smartbridge)

SMARTBRIDGE
Let's Bridge the Gap

ACKNOWLEDGEMENT

I am a student of 1st year, Industrial and production Engineering of Jabalpur Engineering College JABALPUR (M.P.), have completed the project under the guidance of smartinternz instructors.

I would like to thank smartinternz for providing me this remote internship in this difficult situation. I have learnt a lot through this internship program.

NITIN KUMAR PATEL
SB43119

INDEX

Project Scope

Project title - **Smart Agriculture system based on IoT - SB43119**	Project Id - **SPS_PRO_101**
Company - **The Smartbridge**	Duration - **30 days**
Kickoff Date - **14-05-2020**	Estimated Completion - **14-06-2020**

- **Scope Desciption :-**

1. A smart agriculture system to monitor farm land weather and control motor pumps.
2. Setup of the smart agriculture system based on IoT.
3. understand the working of IBM cloud,Node-Reda and receiving data through API call.

- **Scope Deliverables :-**

1. An User Interface displaying temperature, humidity, soil temperature, pressure and wind speed of the farm land.
2. According to forecast farmers can irrigate the farm land.

- **Stake holders:-** The project is designed for the farmers, with the help of user interface they may gain their corps. By adding a large number of farmers with this interface, project could be success.

- **Project Member :- NITIN KUMAR PATEL**

- **Project Requirements : -**

1. IBM Cloud,Watson IoT platform,IoT IBM simulator sensor
2. Node-Red Flow editor
3. Python IDLE
4. Open weather API

- **Project Schedule**:-

Week 1	Project scope, schedule , team and deliverable, setup the Development Environment, creation of IBM cloud account
Week 2	Node-Red installation, python IDLE, connected IOT simulator to Watson IoT platform
Week 3	Configured the node red to get the data from IBM IoT platform and open weather API
Week 3	Built a web app, configured device to receive data from the web app and controlled motor

★★★★★

Project Report

Project Name :- **Smart Agriculture System Based on IoT**
Kick-off Date :- **14/05/2020**
Company :- **Smartinternz (The Smartbridge)**

1. INTRODUCTION :-

1.1 Overview - This report details a REMOTE SUMMER INTERNSHIP at smartinternz (The Smartbridge). I worked in the Smart Agriculture system based on IoT project. The product proposed in this paper uses IBM Watson IoT platform, Node-Red flow editor, IBM IoT sensor simulator, Openweather API and live data feed can be monitored on Node-Red Dashboard.

1.2 Purpose - The aim is to propose a technology which can generate messages on different platforms to notify farmers. The product will assist farmers by getting live data (Temperature, humidity, soil moisture) from the farmland to take necessary steps to enable them to do smart farming by also increasing their crop yields and saving resources .

2. LITERATURE SURVEY :-

2.1 Existing Problem - This is the project from the motivation of the farmers working in the farm lands are solely dependent on the rains and bore wells for irrigation of their land. In recent times, farmers have been using irrigation technique through manual control in which the farmers irrigate the land at regular intervals by turning the water-pump ON/OFF when required. They may have to travel so far for SWITCHING ON/OFF motor. They may be suffering from hot Sun, rain and night time too. Sometimes they know about the weather forecast but due to some reasons they can't go to the farm lands for irrigation. After reaching their farm, they found that there is no power, so they quietly disappointed to it.

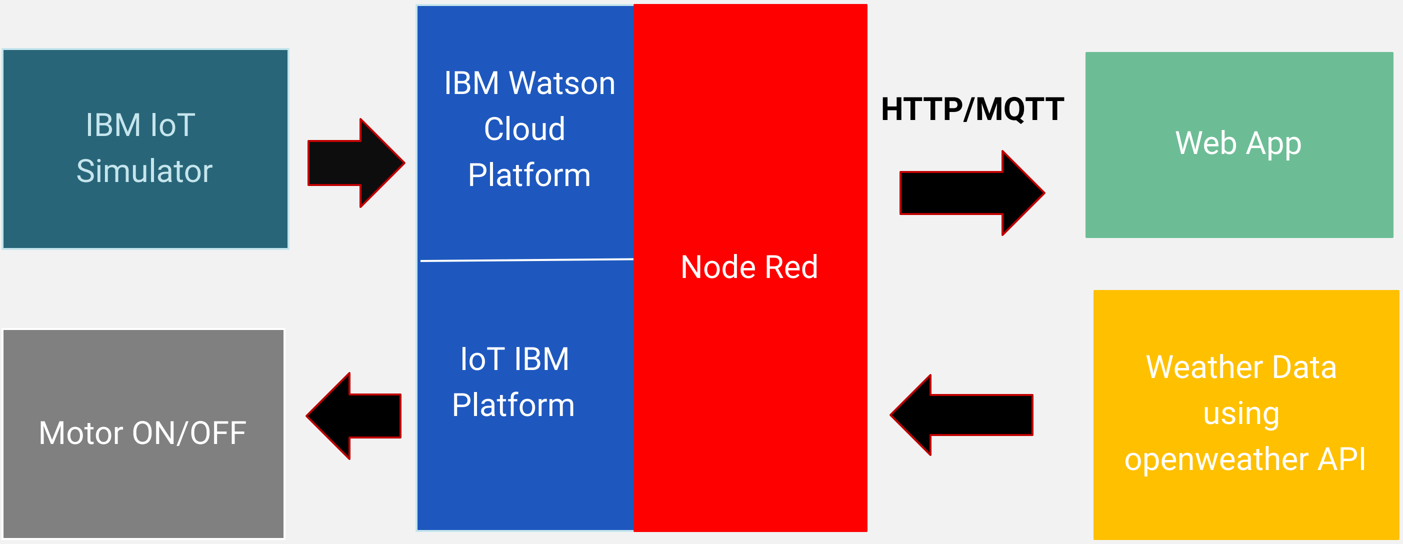
2.2 Proposed Solution - Today IoT has started touching people everywhere and from the point of normal use, IoT is laying the foundation of development of various products like smart health services, smart living, smart education in schools and automation. And commercially it is being used in manufacturing, transportation, agriculture and business management and many other fields. The most researched area of IoT is agriculture. Because it is really crucial sector to ensure the food security as global population is increasing rapidly. Agriculture products need applications like soil moisture monitoring, environmental condition monitoring for temperature, moisture, supply chain management and infrastructure management.

IoT technologies can improve crop yield by eliminating waste, driving operational efficiently, and establish secured food supply chain. This solution can monitor the plant environment 24/7 in real-time, analyzing sensor data from soil and environment, getting information such as temperature, humidity, plant soil moisture, rainfall, air, humidity and more.

In this project, IBM cloud will act as a base. We will receive simulator data through IBM IoT simulator sensor and openweather API data through node red to the IBM IoT Watson cloud platform. In Node red we will design a UI to display and analyze the weather conditions and forecast of the farm land. According to the weather conditions we can control the motor pump from anywhere.

3. THEORTICAL ANALYSIS :-

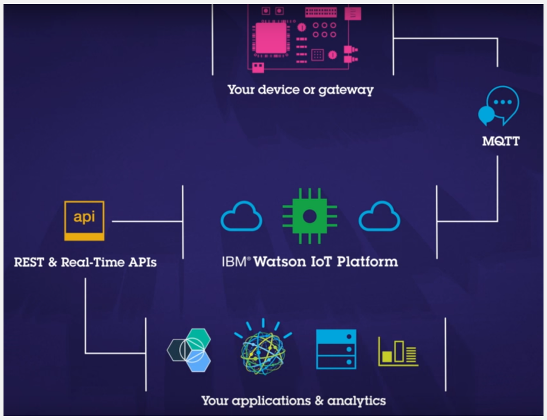
3.1 Block Diagram -

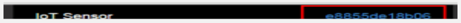


3.2 Software Designing - we have not integral any hardware device in this project. it's developed and processed by software devices. There are so many software devices which we have used in this project, are following:-

- **IBM cloud** - IBM Cloud is a set of cloud computing services for businesses offered by the information technology company IBM. It combines platform as a service (PaaS)with infrastructure as a service (IaaS). The platform scales and supports both small development teams and organizations, and large enterprise businesses. It is globally deployed across data centers around the world.

- **IBM Watson IoT platform** - IBM Watson IoT Platform is a managed, cloud-hosted service designed to make it simple to derive value from IoT devices. Watson IoT Platform and its additional add on services - Blockchain service and analytic service - enable organizations to capture and explore data for devices, equipment, and machines, and discover insights that can drive better decision-making.




IoT Sensor e8855de18b06

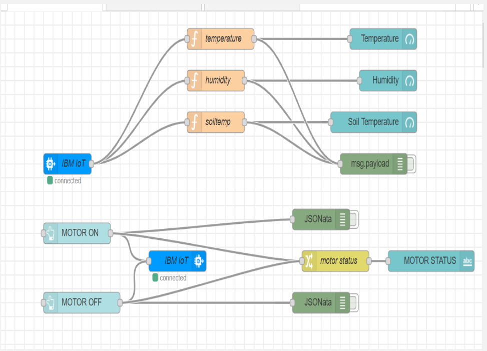
- **Watson IBM IoT sensor simulator** - The IBM Watson IoT platform device simulator to set up simulated events for devices to learn about, test, and demonstrate fully functioning Watson IoT Platform features without having to register and connect actual devices.

- **Git Hub** - GitHub is a Git repository hosting service, but it adds many of its own features. While Git is a command line tool, GitHub provides a Web -based graphical interface. It also provides access control and several collaboration feature such as a wikis and basic task management tools for every project.

- **Slack** - Slack is a proprietary business communication platform developed by American software company Slack Technologies. Slack offers many IRC-style features, including persistent chat rooms (channels) organized by topic, private groups, and direct messaging.

- **Zoho writer** - Zoho writer is a fully - featured word processor on the cloud, designed for collaborative work that gives you everything you need to create powerful documents. with a clean UI, intuitve interface and document modes, it introduces you to a new way of writing. zoho is an efficient word processor, document editor that helps your teams collaborate better.

- **Node Red** - Node Red is a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways. It provides a browser-based editor that makes it easy to wire together flows using the wide range of nodes in the palette that can be deployed to its runtime in a single click.

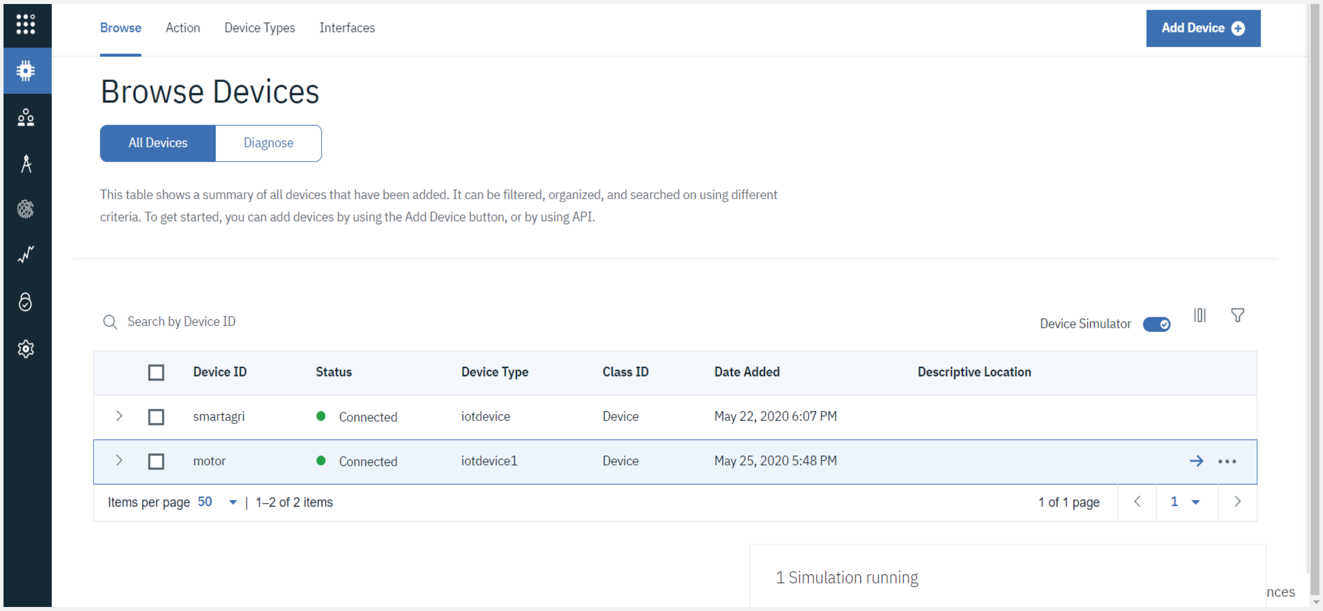


- **Open weather API** - They support multiple languages, units of measurement and data formats. Additionally, the **OpenWeatherMap** service allows any users to get basic **weather** data on the company's website.

4. EXPERIMENTAL INVESTIGATIONS :-

4.1 IBM cloud and Watson IoT platform - I have created an account on IBM cloud which provides so many type of services such as AI, ML, IoT, app development etc. our project is IoT based so I have launched it. In this platform we can connect our hardware and software devices and can receive events or data and commands also we can visualize our data in this platform. In this project I have created two devices one for receiving weather data through API and other is for receiving commands from the Node-Red UI.

For creating a device in the Watson IBM IoT platform we have to write device type and device id in the given field and authentication token for security purpose.



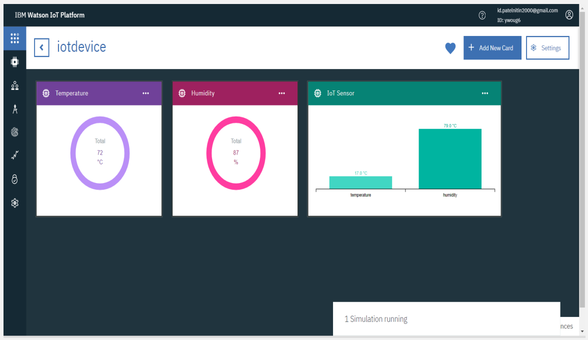
Browse Devices

This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.

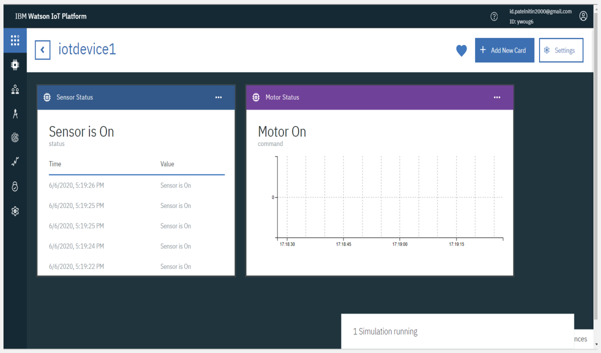
□	Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
□	smartagri	● Connected	iotdevice	Device	May 22, 2020 6:07 PM	
□	motor	● Connected	iotdevice1	Device	May 25, 2020 5:48 PM	

Created device named as smartagri and motor

I have created two boards to display IoT sensor data and commands from the Node-red UI.



IBM IoT simulator sensor Board



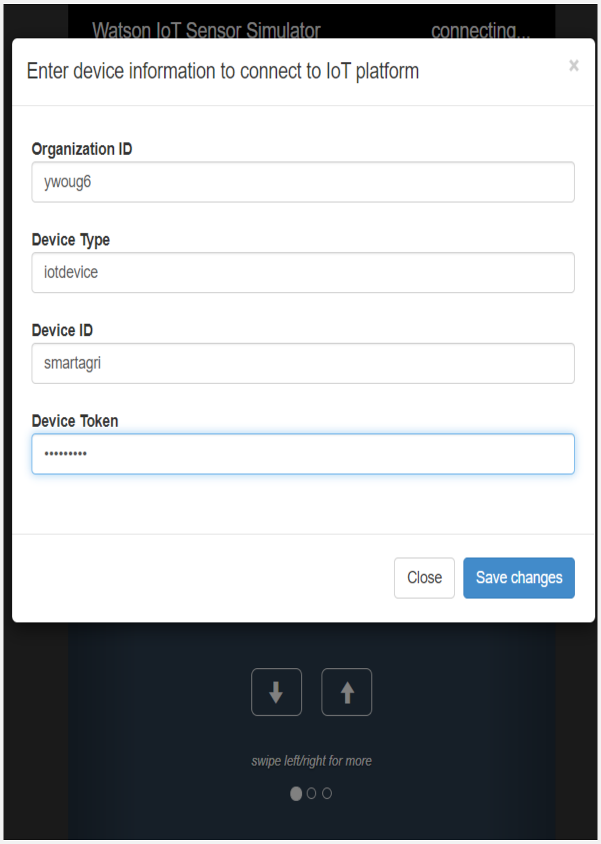
Sensor is On
status

Time	Value
6/6/2020, 5:19:26 PM	Sensor is On
6/6/2020, 5:19:25 PM	Sensor is On
6/6/2020, 5:19:25 PM	Sensor is On
6/6/2020, 5:19:24 PM	Sensor is On
6/6/2020, 5:19:22 PM	Sensor is On

Motor On
command

Motor command board

4.2 IBM IoT simulator sensor - It is a sensor which collects soil temperature, humidity and atmosphere temperature and send to the Ibm Watson platform. To get this data I have connected this sensor with my first device "smartagri". connecting procedure is given below in the images :-



Enter device information to connect to IoT platform

Organization ID: ywoug6

Device Type: iotdevice

Device ID: smartagri

Device Token: •••••••••

Fig.1

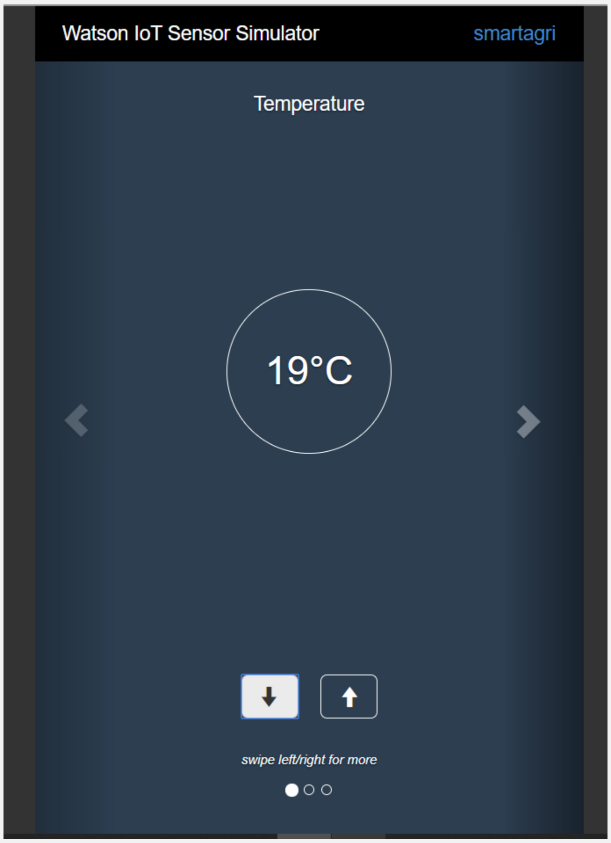


Fig. 2

In the figure 1 I have filled "smartagri" device credential and in the figure 2 is connected with the device.

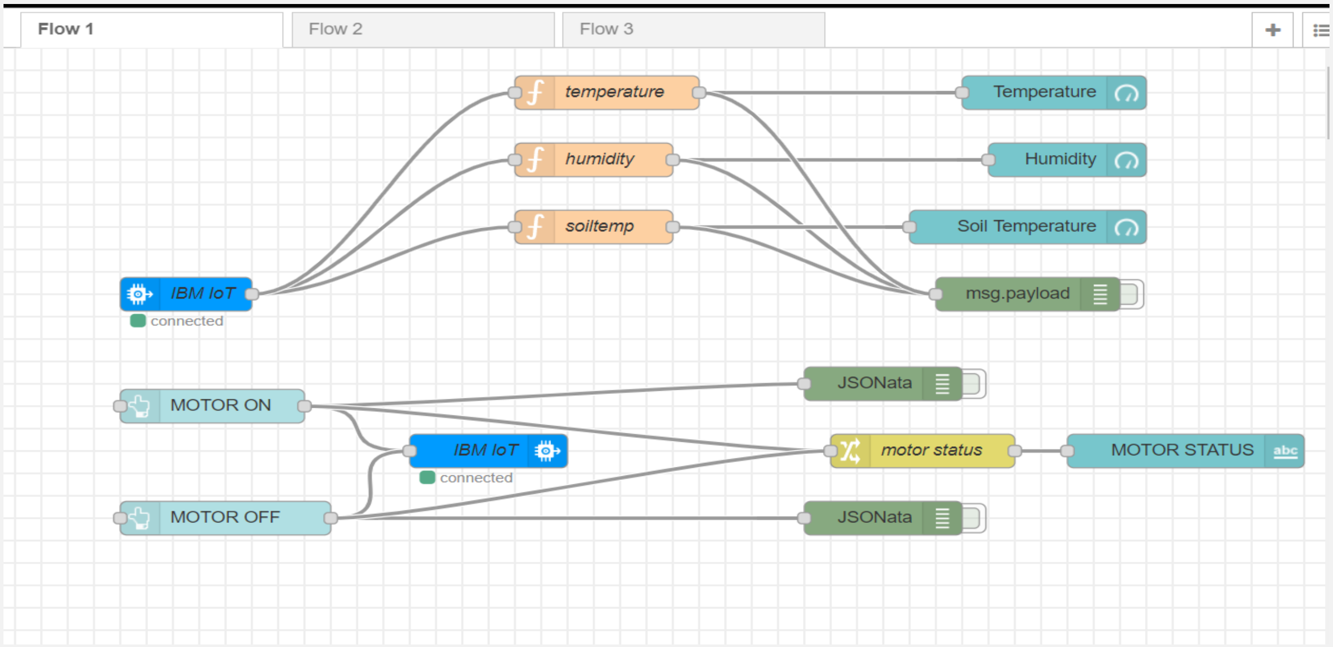
4.3 Node- Red - Node red is a flow editor to make flows and send or collect data through API calls or clouds. Here I have made three node flows to generate expected UI.

In the first flow I have made two connections first is to retrieve sensor data using IBM input node,function node and gauge node to show the output in the UI.

second connection is to create motor ON/OFF button to control motor pumps using button node, IBm output node and change node is used to show motor status.

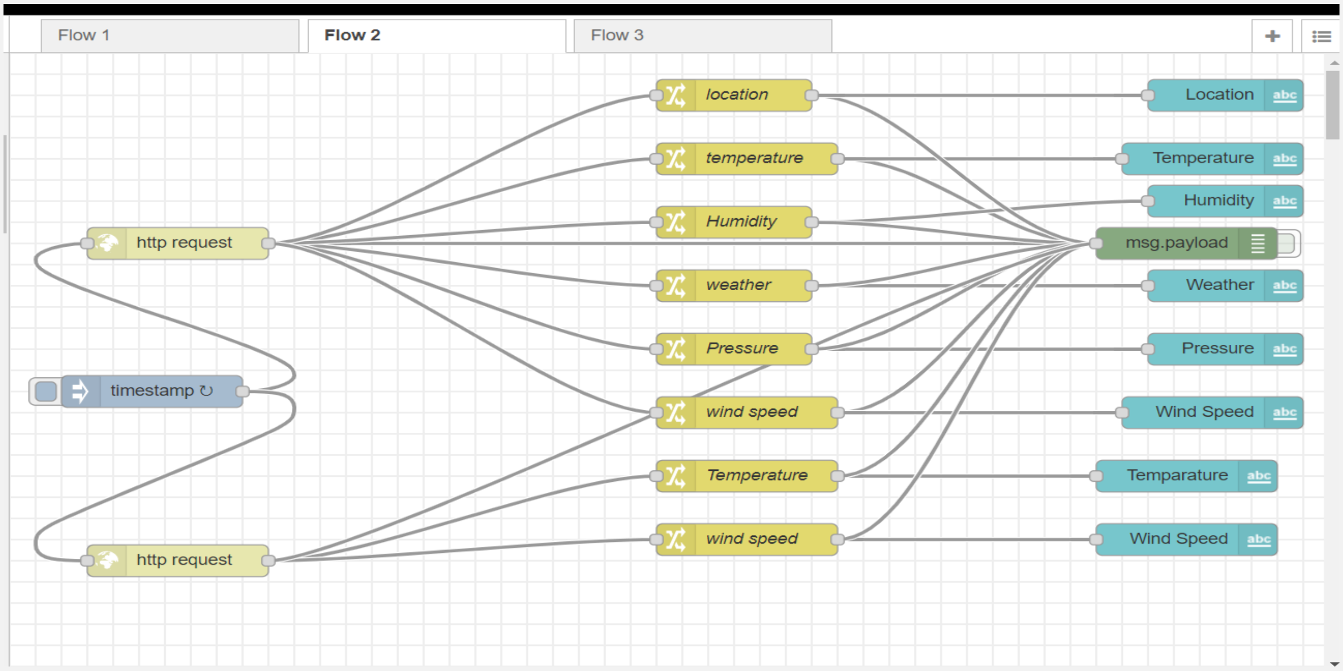
Debug node is used to show the output which we will receive in the UI.

Flow 1 image is shown in the next page...



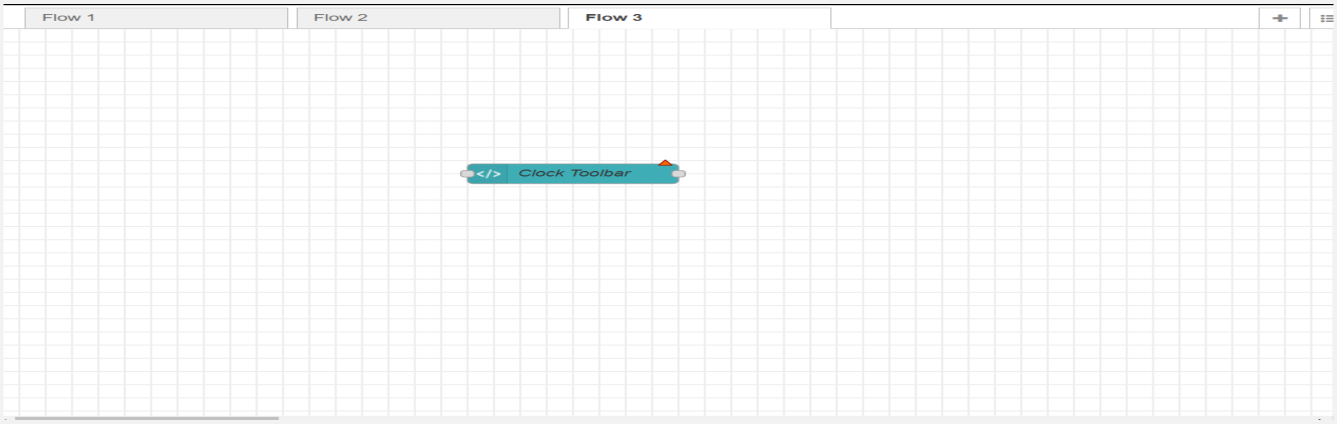
Flow 1

In the flow 2 we are getting open weather API data which is provides us current weather data and forecasting of a specific location using API. Here I have used inject node to inject data in a specific time interval , http request node to connect with openweather API, switch node changes format of a message or a text in different formats. flow 2 image is given below :-



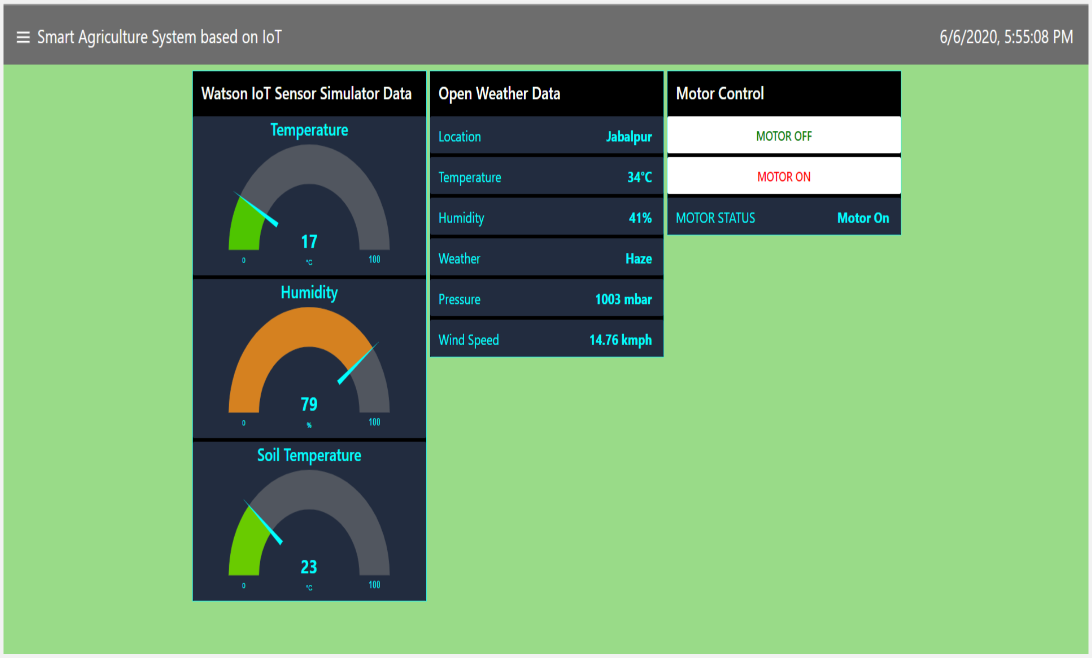
Flow 2

In the flow 3 I have imported clock template to add a clock in the head section of the UI. flow 3 image is given below :-



Flow 3

4.4 Node-Red Dashboard (UI) - After completion of node connection we can achieve dashboard using dashboard node and setting up groups and other styles. In this Project I have created a simple dashboard which is shown below in the image :_



☰ Smart Agriculture System based on IoT 6/6/2020, 5:55:08 PM

Watson IoT Sensor Simulator Data

Temperature 17 °C (0–100)

Humidity 79 % (0–100)

Soil Temperature 23 °C (0–100)

Open Weather Data

Location	Jabalpur
Temperature	34°C
Humidity	41%
Weather	Haze
Pressure	1003 mbar
Wind Speed	14.76 kmph

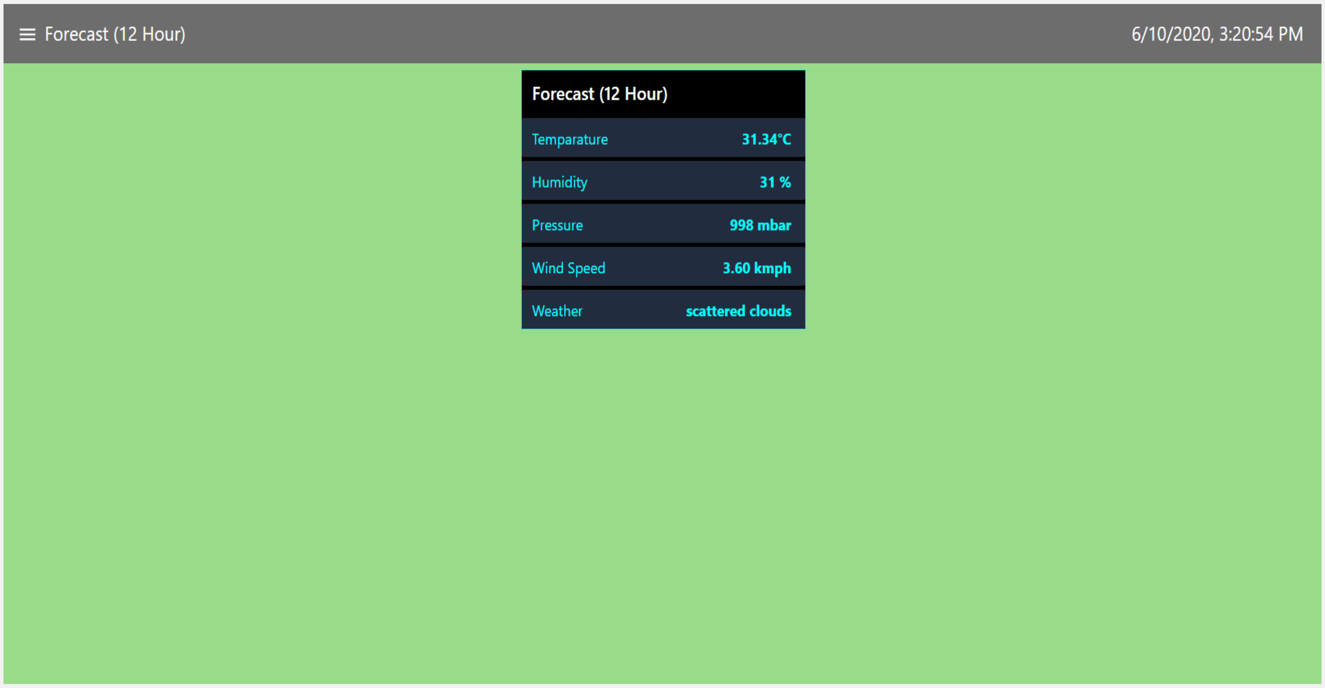
Motor Control

MOTOR OFF

MOTOR ON

MOTOR STATUS Motor On

Tab 1

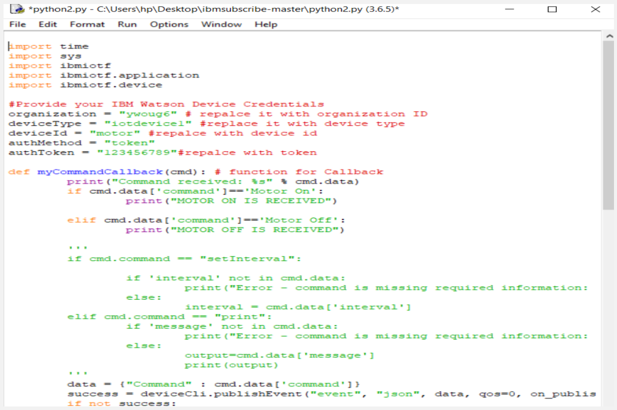


☰ Forecast (12 Hour) 6/10/2020, 3:20:54 PM

Forecast (12 Hour)	
Temparature	31.34°C
Humidity	31 %
Pressure	998 mbar
Wind Speed	3.60 kmph
Weather	scattered clouds

Tab 2

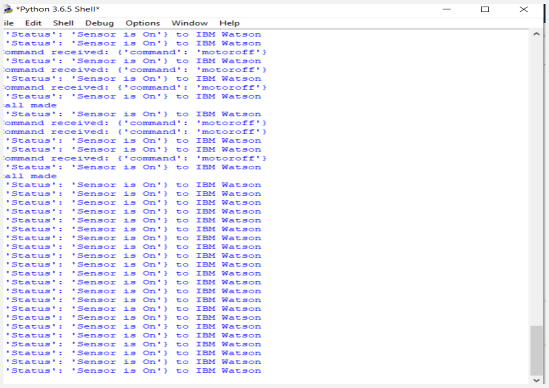
4.5 Subscribing Python code - I have imported the given python code from the github and replaced second device credentials named as "motor" in it. This code is for getting motor pump commands and sending them to the Watson IBM IoT platform. After changing the credentials we have to run this code then we can see a message that device is connected as well as in the Watson IoT platform it will show connected. If we will press a motor ON/OFF button in the dashboard we will receive a message in the python shell that motor is on or off. Python shell image is given below :-



python2.py - C:\Users\hp\Desktop\ibmsubscribe-master\python2.py (3.6.5)
File Edit Format Run Options Window Help

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

#Provide your IBM Watson Device Credentials
organization = "ywoug6" # repalce it with organization ID
deviceType = "iotdevice1" #replace it with device type
deviceId = "motor" #repalce with device id
authMethod = "token"
authToken = "123456789"#repalce with token

def myCommandCallback(cmd): # function for Callback
        print("Command received: %s" % cmd.data)
        if cmd.data['command']=='Motor On':
                print("MOTOR ON IS RECEIVED")

        elif cmd.data['command']=='Motor Off':
                print("MOTOR OFF IS RECEIVED")

        '''
        if cmd.command == "setInterval":

                if 'interval' not in cmd.data:
                        print("Error - command is missing required information:
                else:
                        interval = cmd.data['interval']
        elif cmd.command == "print":
                if 'message' not in cmd.data:
                        print("Error - command is missing required information:
                else:
                        output=cmd.data['message']
                        print(output)
        '''
        data = {"Command" : cmd.data['command']}
        success = deviceCli.publishEvent("event", "json", data, qos=0, on_publis
        if not success:
```

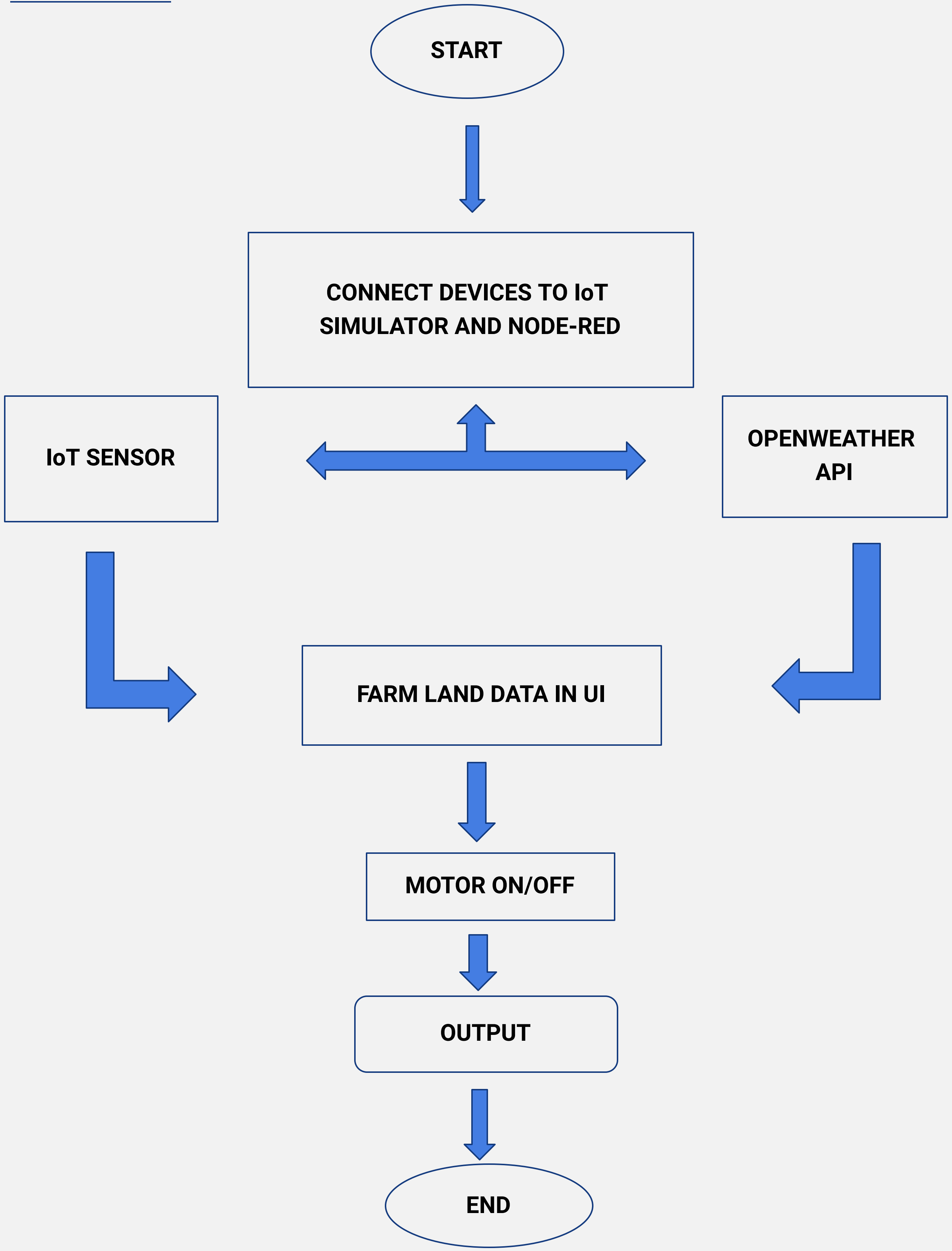
Python code



```
'Status': 'Sensor is On'} to IBM Watson
'Status': 'Sensor is On'} to IBM Watson
ommand received: {'command': 'motoroff'}
'Status': 'Sensor is On'} to IBM Watson
ommand received: {'command': 'motoroff'}
'Status': 'Sensor is On'} to IBM Watson
ommand received: {'command': 'motoroff'}
'Status': 'Sensor is On'} to IBM Watson
all made
'Status': 'Sensor is On'} to IBM Watson
ommand received: {'command': 'motoroff'}
ommand received: {'command': 'motoroff'}
'Status': 'Sensor is On'} to IBM Watson
'Status': 'Sensor is On'} to IBM Watson
ommand received: {'command': 'motoroff'}
'Status': 'Sensor is On'} to IBM Watson
all made
'Status': 'Sensor is On'} to IBM Watson
'Status': 'Sensor is On'} to IBM Watson
'Status': 'Sensor is On'} to IBM Watson
'Status': 'Sensor is On'} to IBM Watson
'Status': 'Sensor is On'} to IBM Watson
'Status': 'Sensor is On'} to IBM Watson
'Status': 'Sensor is On'} to IBM Watson
'Status': 'Sensor is On'} to IBM Watson
'Status': 'Sensor is On'} to IBM Watson
'Status': 'Sensor is On'} to IBM Watson
'Status': 'Sensor is On'} to IBM Watson
'Status': 'Sensor is On'} to IBM Watson
'Status': 'Sensor is On'} to IBM Watson
'Status': 'Sensor is On'} to IBM Watson
'Status': 'Sensor is On'} to IBM Watson
'Status': 'Sensor is On'} to IBM Watson
'Status': 'Sensor is On'} to IBM Watson
'Status': 'Sensor is On'} to IBM Watson
'Status': 'Sensor is On'} to IBM Watson
'Status': 'Sensor is On'} to IBM Watson
'Status': 'Sensor is On'} to IBM Watson
'Status': 'Sensor is On'} to IBM Watson
```

Received Commands

5. FLOWCHART :-


START
CONNECT DEVICES TO IoT SIMULATOR AND NODE-RED
IoT SENSOR
OPENWEATHER API
FARM LAND DATA IN UI
MOTOR ON/OFF
OUTPUT
END

6.RESULTS:-

We have a web app which is an user interface having temperature, humidity, soil temperature and open weather API data. The web app displays current weather 12 hours forecast having wind speed, pressure and weather description (rain prediction) by which farmer can turn ON/OFF the motor pump which is implemented in farm land.

7. ADVANTAGES AND DISADVANTAGES :-

ADVANTAGES :-

i. It is cost effective method.
ii. it provides wireless motor controlling so from anywhere motor can be] controlled.
iii. Hard ware device is not required.
iv. easy to use and user-friendly.

DISDVANTAGES :-

i. The smart agriculture availability on internet continuously. rural part of the developing countries did not fulfill this requirements. Moreover internet is slower.
ii. Fault sensor or data processing engines can cause faulty decisions which may lead to over use of water, fertilizers and other wastage of resource.

8. APPLICATIONS :-

PRECISION FARMING:- Precision agriculture is one of the most famous applications of IoT in the agricultural sector and numerous organizations are leveraging this technique around the world.

SMART GREENHOUSES :- For controlling the environment in a smart greenhouse, different sensors that measure the environmental parameters according to the plant requirement are used. We can create a cloud server for remotely accessing the system when it is connected using IoT.

9. CONCLUSION :-

IoT based smart agriculture system for live monitoring of temperature, soil temperature, humidity and openweather forecast has been proposed using IBM cloud. The system has high efficiency and accuracy in fetching the live data of temperature and soil humidity. The IoT based smart agriculture system being proposed via this report will assist farmers in increasing the agriculture yield and take efficient care of food production as the system will always provide helping hand to farmers for getting accurate live feed of environmental temperature and soil humidity and soil temperature.

10. FUTURE SCOPE:-

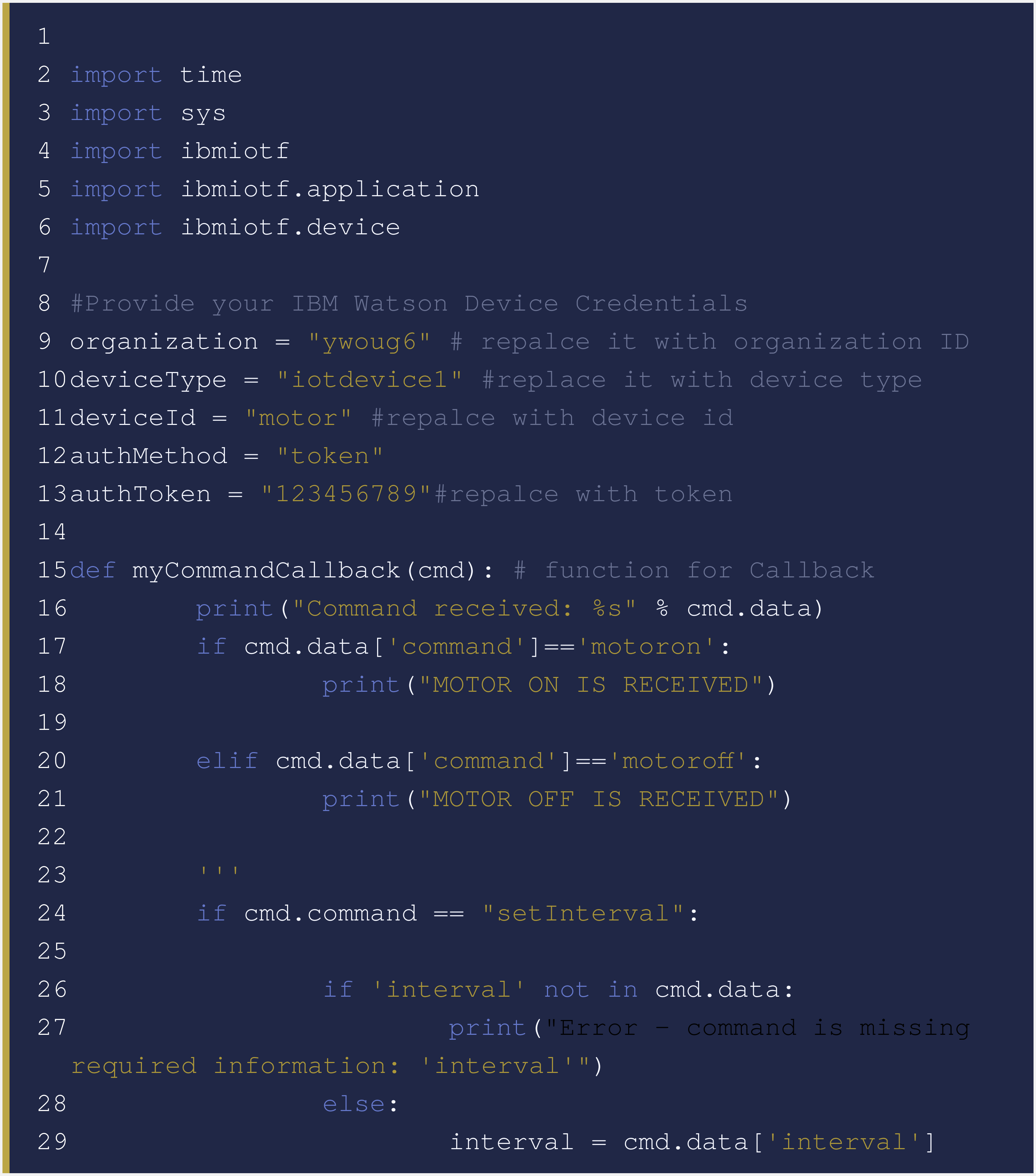
Smart farming is a concept quickly catching on in the agricultural business. Offering high-precision crop control, useful data collection, and automated farming techniques, there are clearly many advantages a networked farm has to offer. With a future of efficient, data-driven, highly-precise farming methods, it is definitely safe to call this type of farming smart. We can expect IoT will forever change the way we grow food.

11. BIBILOGRAPHY:-

- Reference link for installing nodes-
  https://smartinternz.com/assets/docs/Smart%20Home%20Automation%20using%20IBM%20cloud%20Services%20(1).pdf
- Reference link to get simulator-
  https://smartinternz.com/assets/docs/Smart%20Home%20Automation%20using%20IBM%20cloud%20Services%20(1).pdf
- Link to create an open weather API account -
  https://openweathermap.org/
- Reference link to configure open weather API platform-
  https://smartinternz.com/assets/docs/Sending%20Http%20request%20to%20Open%20weather%20map%20website%20to%20get%20the%20weather%20forecast.pdf
- Reference for HTTP request-
  https://www.youtube.com/watch?v=cicTw4SEdxk
- Reference link to create UI-
  https://smartinternz.com/assets/docs/Smart%20Home%20Automation%20using%20IBM%20cloud%20Services%20(1).pdf
- Code to retrieve command from IBM IoT
  https://github.com/rachuriharish23/ibmsubscribe

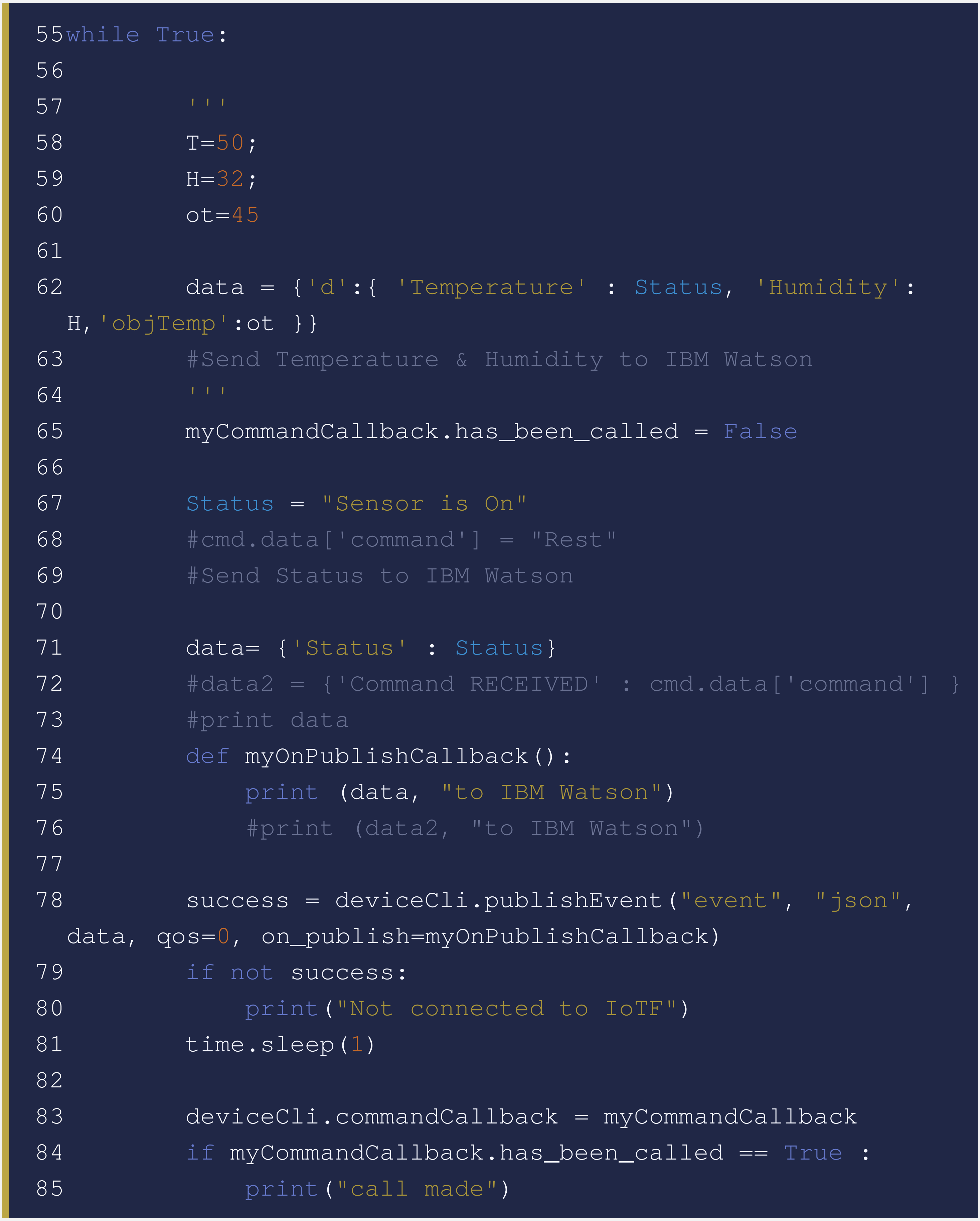
12. **APENDIX :-**

A. SOURCE CODE :- Python code to retrieve command from web app



```

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

#Provide your IBM Watson Device Credentials
organization = "ywoug6" # repalce it with organization ID
deviceType = "iotdevice1" #replace it with device type
deviceId = "motor" #repalce with device id
authMethod = "token"
authToken = "123456789"#repalce with token

def myCommandCallback(cmd): # function for Callback
        print("Command received: %s" % cmd.data)
        if cmd.data['command']=='motoron':
                print("MOTOR ON IS RECEIVED")

        elif cmd.data['command']=='motoroff':
                print("MOTOR OFF IS RECEIVED")

        '''
        if cmd.command == "setInterval":

                if 'interval' not in cmd.data:
                        print("Error - command is missing required information: 'interval'")
                else:
                        interval = cmd.data['interval']
```

```
    elif cmd.command == "print":
        if 'message' not in cmd.data:
            print("Error - command is missing required information: 'message'")
        else:
            output=cmd.data['message']
            print(output)
    '''
    data = {"Command" : cmd.data['command']}
    success = deviceCli.publishEvent("event", "json", data, qos=0, on_publish=myOnPublishCallback)
    if not success:
      print("Not connected to IoTF")

    myCommandCallback.has_been_called = True
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:

        '''
        T=50;
        H=32;
        ot=45

        data = {'d':{ 'Temperature' : Status, 'Humidity': H,'objTemp':ot }}
        #Send Temperature & Humidity to IBM Watson
        '''
        myCommandCallback.has_been_called = False

        Status = "Sensor is On"
        #cmd.data['command'] = "Rest"
        #Send Status to IBM Watson

        data= {'Status' : Status}
        #data2 = {'Command RECEIVED' : cmd.data['command'] }
        #print data
        def myOnPublishCallback():
            print (data, "to IBM Watson")
            #print (data2, "to IBM Watson")

        success = deviceCli.publishEvent("event", "json", data, qos=0, on_publish=myOnPublishCallback)
        if not success:
            print("Not connected to IoTF")
        time.sleep(1)

        deviceCli.commandCallback = myCommandCallback
        if myCommandCallback.has_been_called == True :
            print("call made")
```

```


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