Project Report

on

Smart Agriculture system based on IOT

Details :-

Name: Sai Kiran Yerrangi

Email: saikiram8184994@gmail.com

Project ID: SPS_PRO_101

Internship title: Smart Agriculture system based on IOT

Category: Internet of Things

INTRODUCTION

Overview:

Smart farming is a capital-intensive and hi-tech system of growing food cleanly and sustainable for the masses. It is the application of modern ICT (Information and Communication Technologies) into agriculture.

In IoT-based smart farming, a system is built for monitoring the crop field with the help of sensors (light, humidity, temperature, soil moisture, etc.) and automating the irrigation system. The farmers can monitor the field conditions from anywhere. IoT-based smart farming is highly efficient when compared with the conventional approach.

The applications of IoT-based smart farming not only target conventional, large farming operations, but could also be new levers to uplift other growing or common trends in agricultural like organic farming, family farming (complex or small spaces, particular cattle and/or cultures, preservation of particular or high-quality varieties, etc.), and enhance highly transparent farming.

In terms of environmental issues, IoT-based smart farming can provide great benefits including more efficient water usage, or optimization of inputs and treatments. Now, let’s discuss the major applications of IoT-based smart farming that are revolutionizing agriculture.

Project Requirements:

- Project planning and kickoff
- Explore IBM cloud platform
- Explore IBM Watson platform
- Install python IDE
- connect the IOT simulator to Watson IOT platform
- configure the Node-red to get the data from IBM IOT platform & open weather API
- Building a web app
- configuring the device to receive the data from the web application & controlling the motors

Functional Requirements:

- Creating an account in IBM cloud account
- Creating a device to IBM Watson IOT Simulator

Technical Requirements:

- Building a web application to monitor weather conditions & water fields.

Software Requirements:

Python, IBM cloud account, Node red, IBM IOT Simulator https://lh3.googleusercontent.com/1kiixBEozfeG23Eodq29KP-IhqI4yKxcra_wT9C_5aC819pkSd8lLhvU_VtLjT5mAlo

Project Deliverables:

The scope of the project is, the web app display the weather conditions & even if the farmer is not present near his crop he can be able to water his crop by controlling the motors using the mobile application from anywhere.

Project Team:

Individual project

Purpose:

IOT based Smart Farming improves the entire **Agriculture** system by monitoring the field in real-time. With the help of sensors and interconnectivity, the Internet of Things in **Agriculture** has not only saved the time of the farmers but has also reduced the extravagant use of resources such as Water and Electricity

In **IOT-based smart farming**, a system is **built** for monitoring the crop field with the help of sensors (light, humidity, temperature, soil moisture, etc.) and automating the irrigation system. But in this project we are using IBM IOT Watson Simulator to configure the humidity, temperature, soil moisture. And integrate through Node-red application. The farmers can monitor the field conditions from anywhere and control the motor through the mobile app.

LITERATURE SURVEY:

Existing problem:

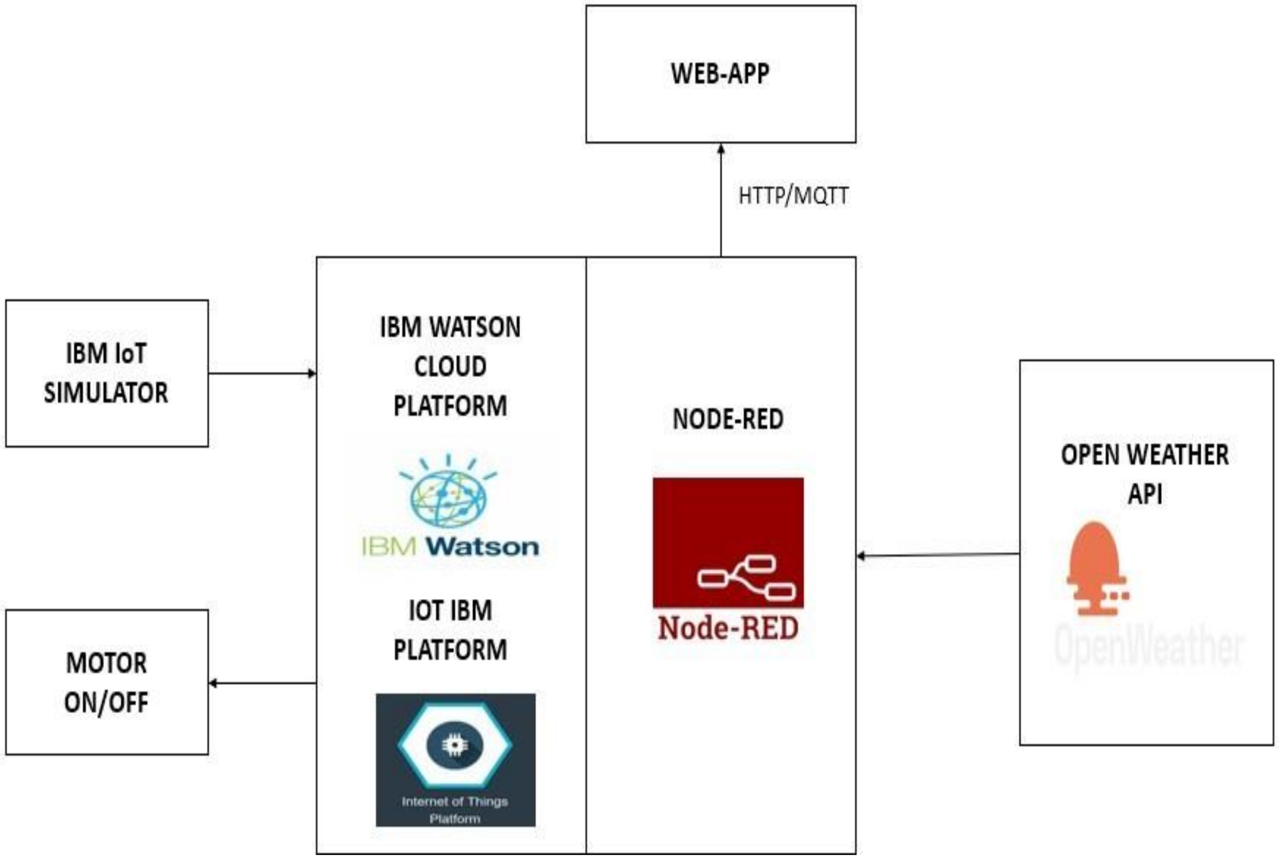
The presence of malware and data thefts is a risk in practically all types of 'connected systems', and **smart agriculture** is not an exception from that. ... Since the third-party attacks on a complex IoT system are often decentralized, detecting and removing them emerges as a big **challenge.**In some critical conditions the farmers could not able to go to their fields and not able to water their crops.

Proposed solution:

Providing **Smart Agricultural Solutions** to Farmers for Better Yielding using IOT. IOT **solution** for **Smart Farming** technology offers complete details in all the spectrums of **agriculture**. It will help provide insights and stats for crops and livestock. **A Smart Farming system uses** modern technology to increase the quantity and quality of **agricultural** products. A web application can easily solve the maintenance of the fields. Getting information from the sensors present in the fields and controlling the motor to water the crops.

THEORITICAL ANALYSIS:

Block/Flow Diagram:



Hardware / Software designing:

1. Create IBM Cloud Account

2. Create IBM Cloud services

3. Create device in IBM IOT platform

4. Create flow and configure node

5. Deploy and run Node Red app

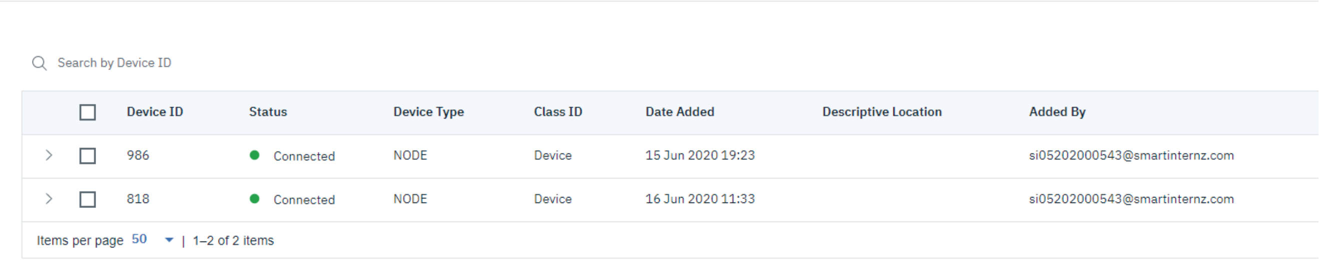
6. Install python IDLE

- firstly, create a device in IBM Cloud.

- Connect the device to IBM Simulator by giving device credentials to get the weather conditions.
- Build a Node-RED flow to display the weather conditions and control the devices for displaying data in the web application.
- Get the real time weather condition data from open weather API and integrate it in the Node-RED.
- Control the working of the created web application to the devices by python coding.

EXPERIMENTAL INVESTIGATIONS:

Device Creation:

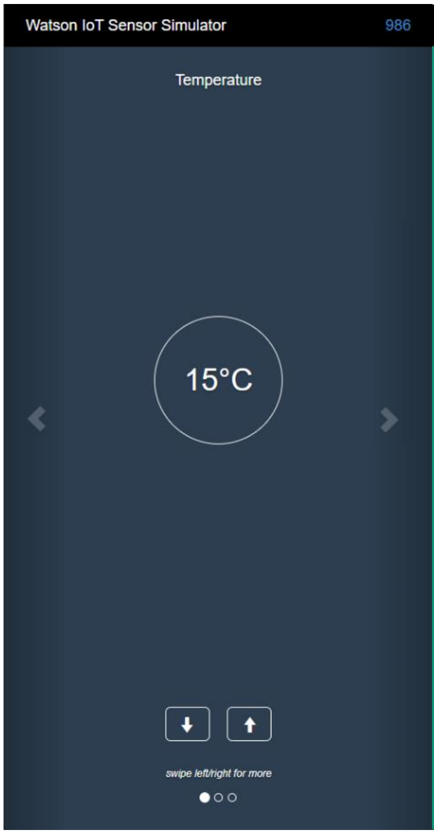


Search by Device ID

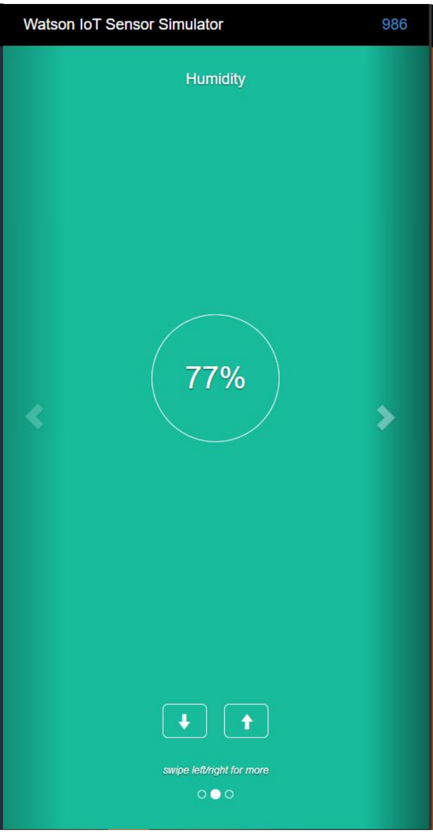
☐	Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location	Added By
☐	986	● Connected	NODE	Device	15 Jun 2020 19:23		si05202000543@smartinternz.com
☐	818	● Connected	NODE	Device	16 Jun 2020 11:33		si05202000543@smartinternz.com

Items per page 50 | 1–2 of 2 items

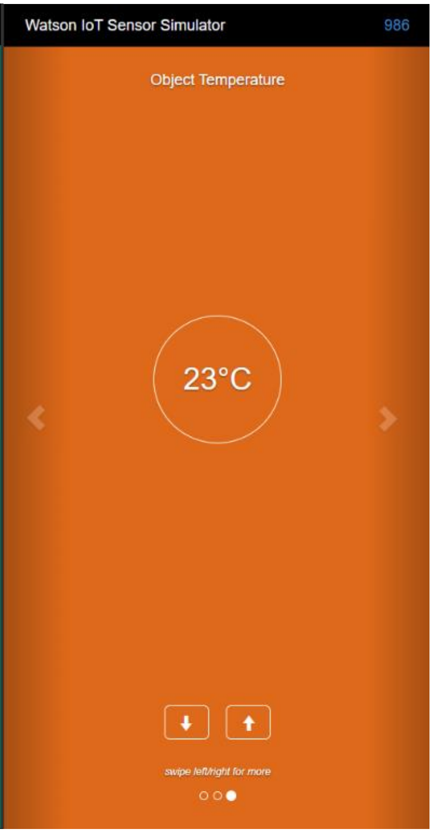
IOT Device Simulator:



Temperature

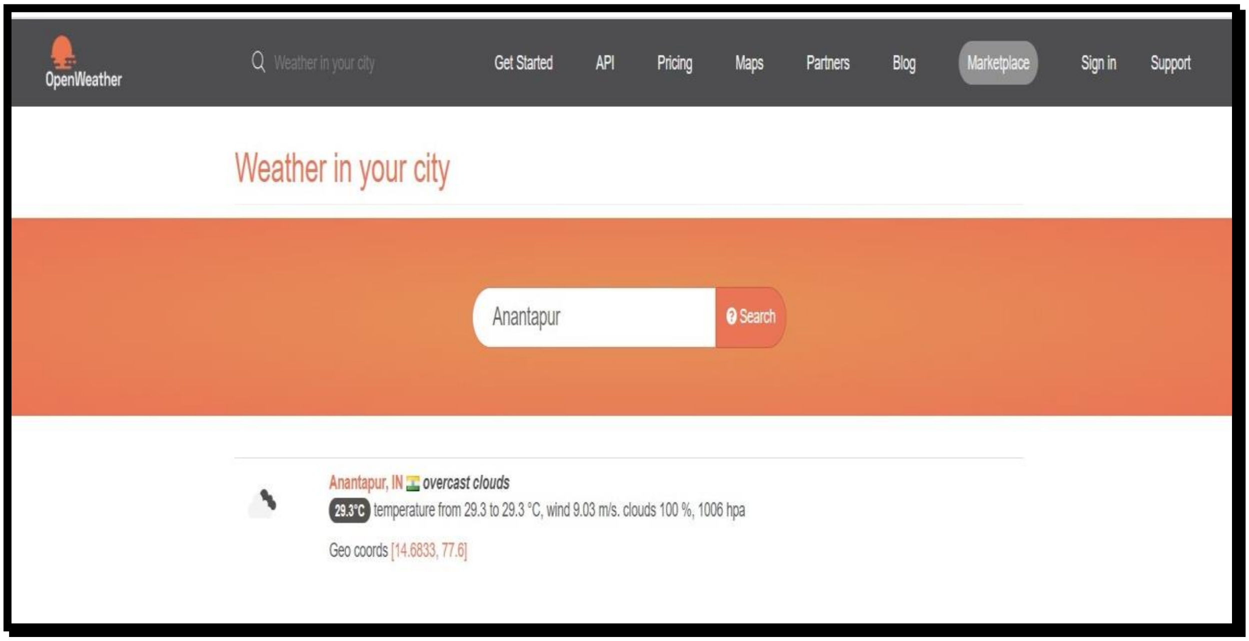


Humidity



Object-Temp

Open Weather API:



Weather in your city

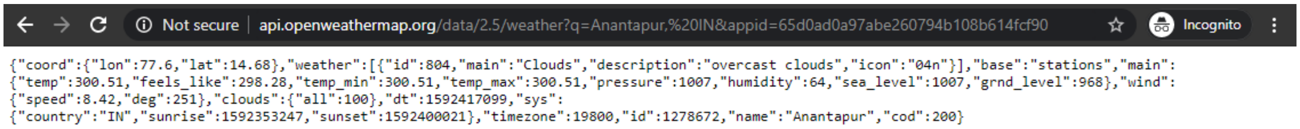
Anantapur

Anantapur, IN *overcast clouds*

29.3°C temperature from 29.3 to 29.3 °C, wind 9.03 m/s. clouds 100 %, 1006 hpa

Geo coords [14.6833, 77.6]

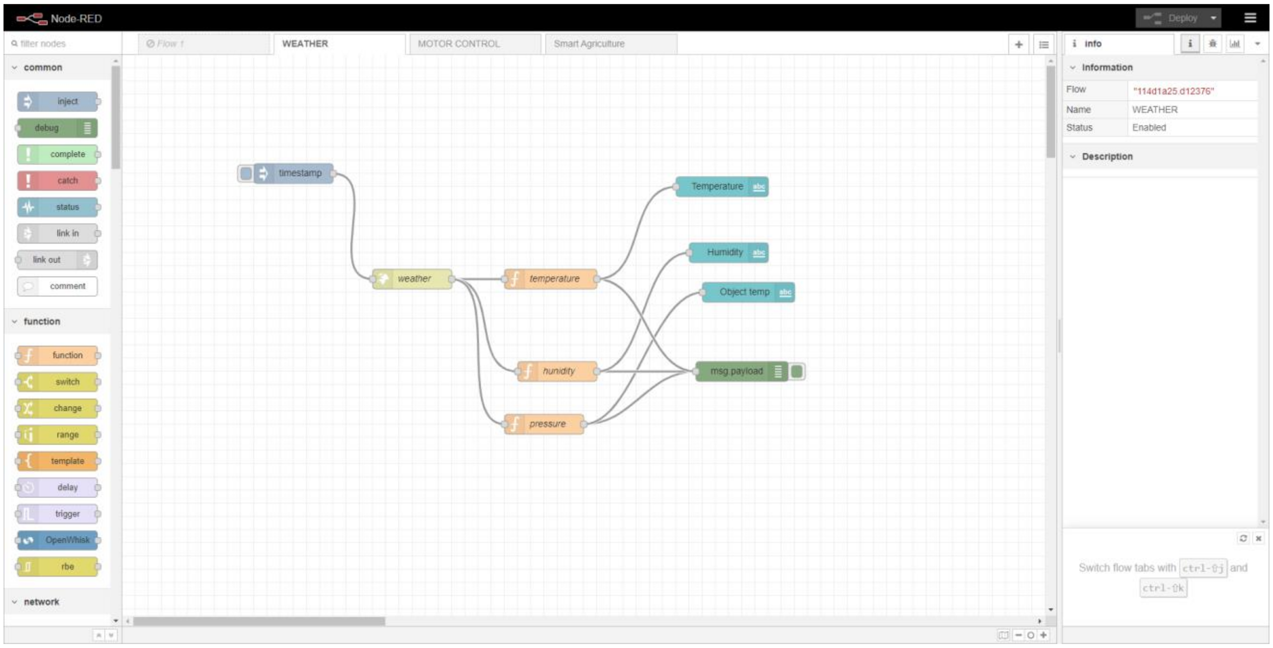
JSON data fetched from API:



api.openweathermap.org/data/2.5/weather?q=Anantapur,%20IN&appid=65d0ad0a97abe260794b108b614fcf90

```
{"coord":{"lon":77.6,"lat":14.68},"weather":[{"id":804,"main":"Clouds","description":"overcast clouds","icon":"04n"}],"base":"stations","main":
{"temp":300.51,"feels_like":298.28,"temp_min":300.51,"temp_max":300.51,"pressure":1007,"humidity":64,"sea_level":1007,"grnd_level":968},"wind":
{"speed":8.42,"deg":251},"clouds":{"all":100},"dt":1592417099,"sys":
{"country":"IN","sunrise":1592353247,"sunset":1592400021},"timezone":19800,"id":1278672,"name":"Anantapur","cod":200}
```

Node-RED HTTP Request:

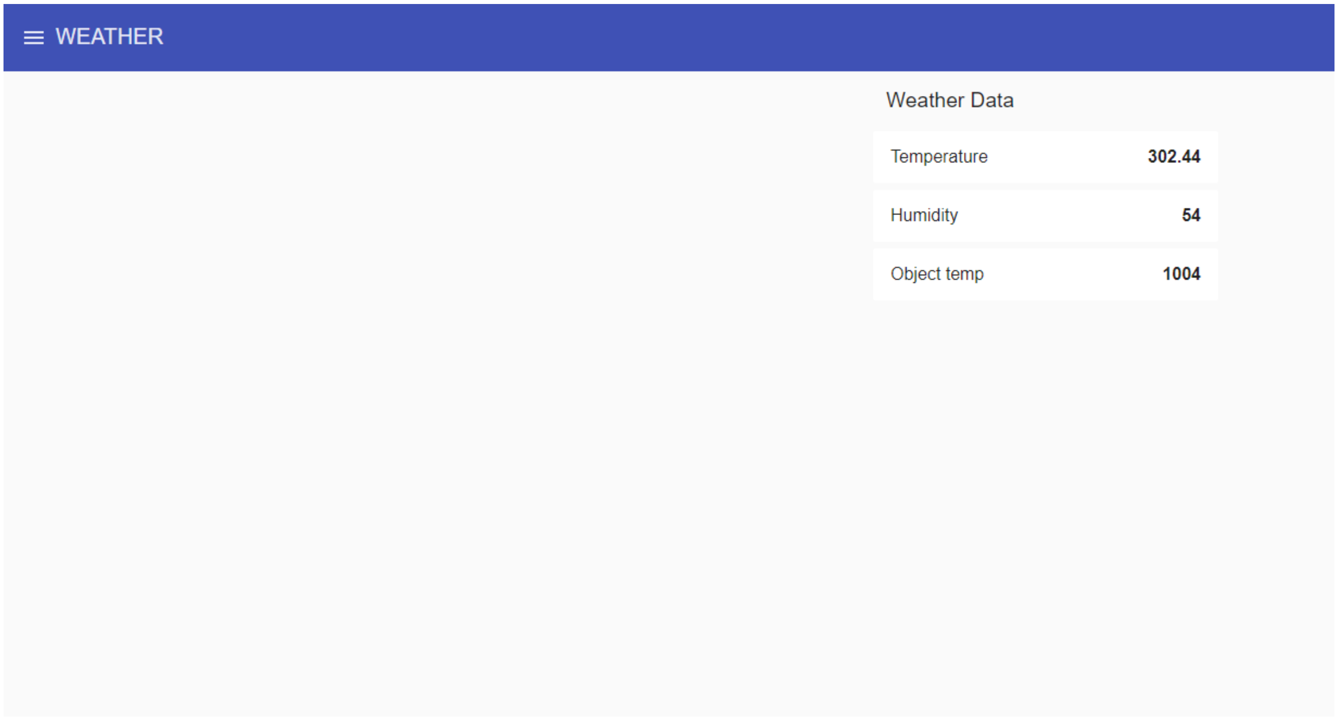


Information

Flow	"114d1a25.d12376"
Name	WEATHER
Status	Enabled

Description

Node-red UI:

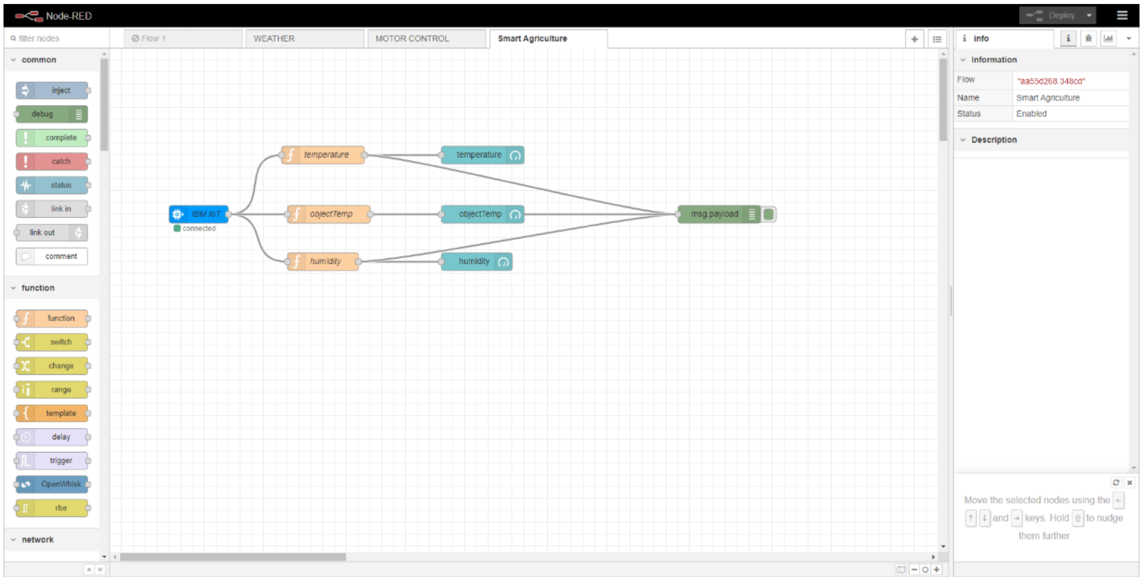


Weather Data

Temperature	302.44
Humidity	54
Object temp	1004

FLOW CHART:

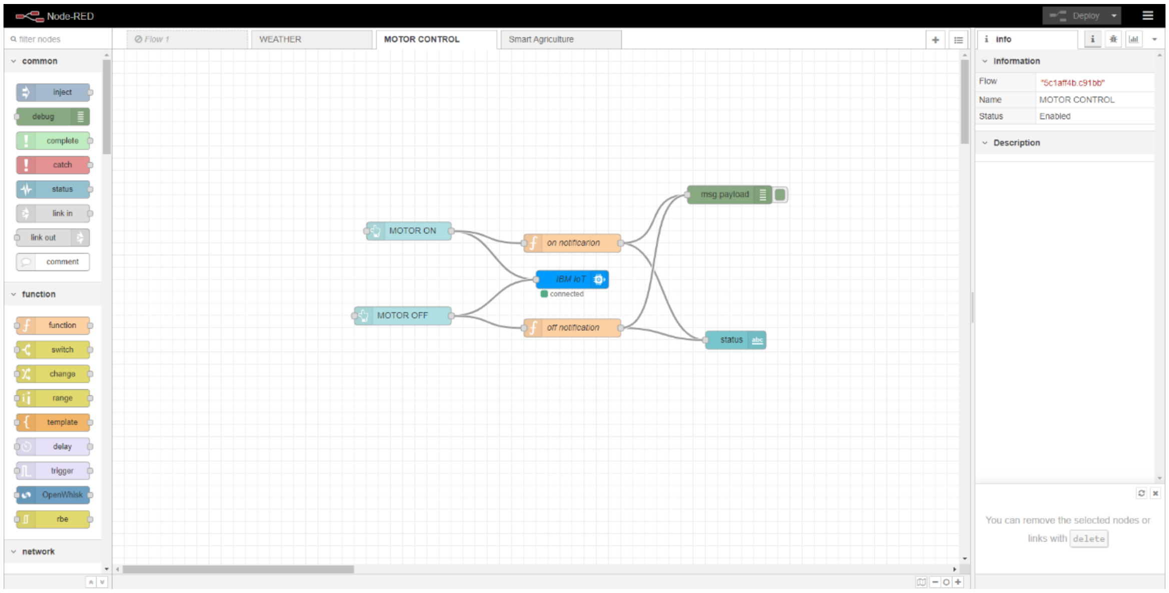
Getting values of temperature, humidity, object Temperature:



Flow	"aa55d268.348cd"
Name	Smart Agriculture
Status	Enabled

Move the selected nodes using the ← ↑ ↓ and → keys. Hold ⇧ to nudge them further

Controlling motor by web app:

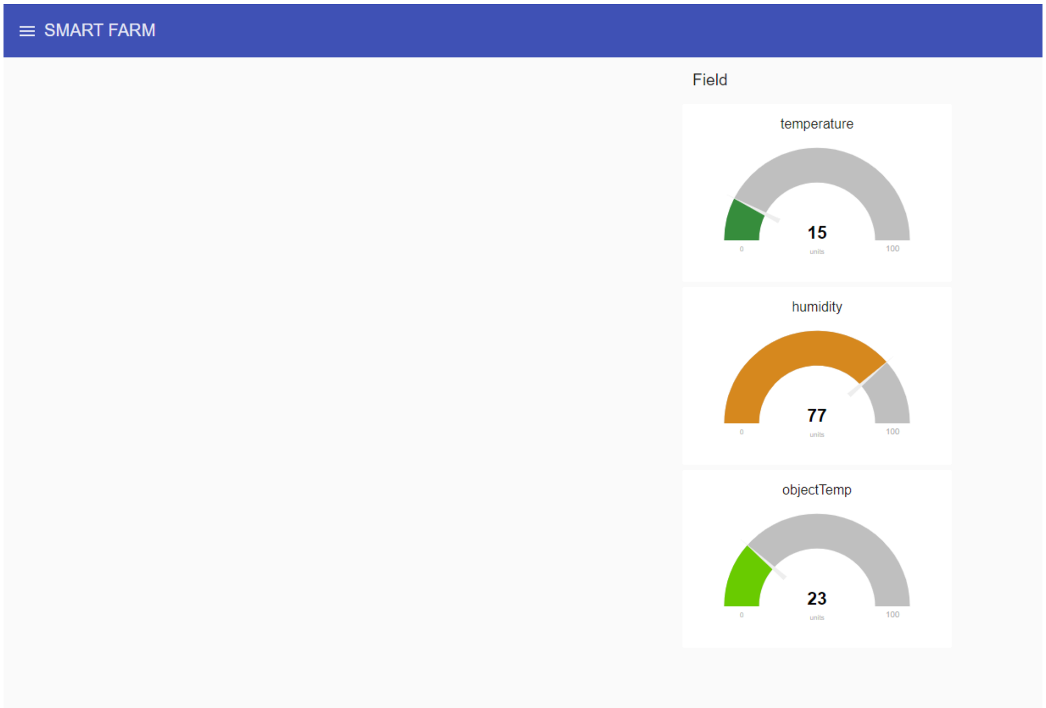


Flow	"5c1aff4b.c91bb"
Name	MOTOR CONTROL
Status	Enabled

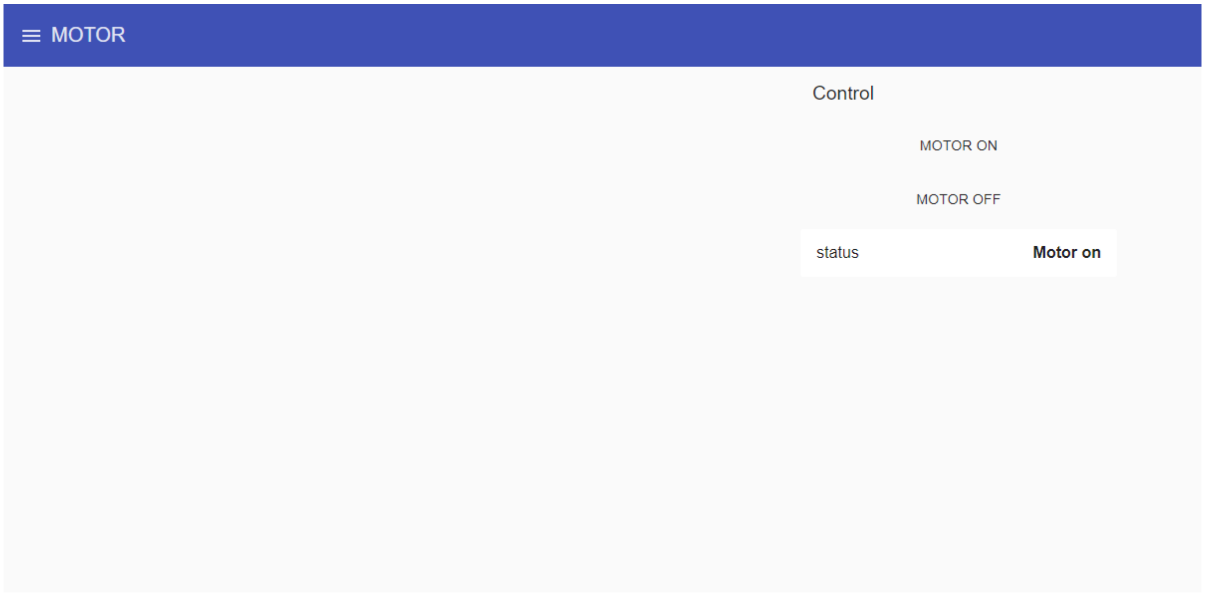
You can remove the selected nodes or links with delete

RESULT:

Node-red UI:

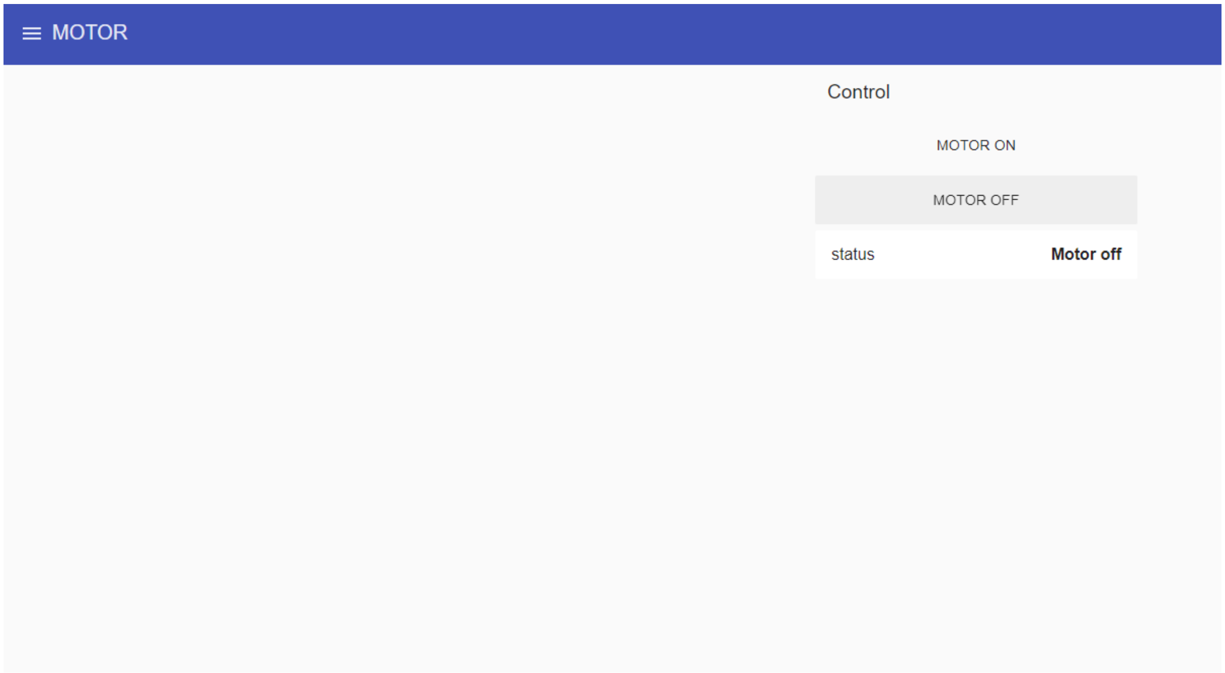


When Motor is ON:



status Motor on

When Motor is OFF:



status Motor off

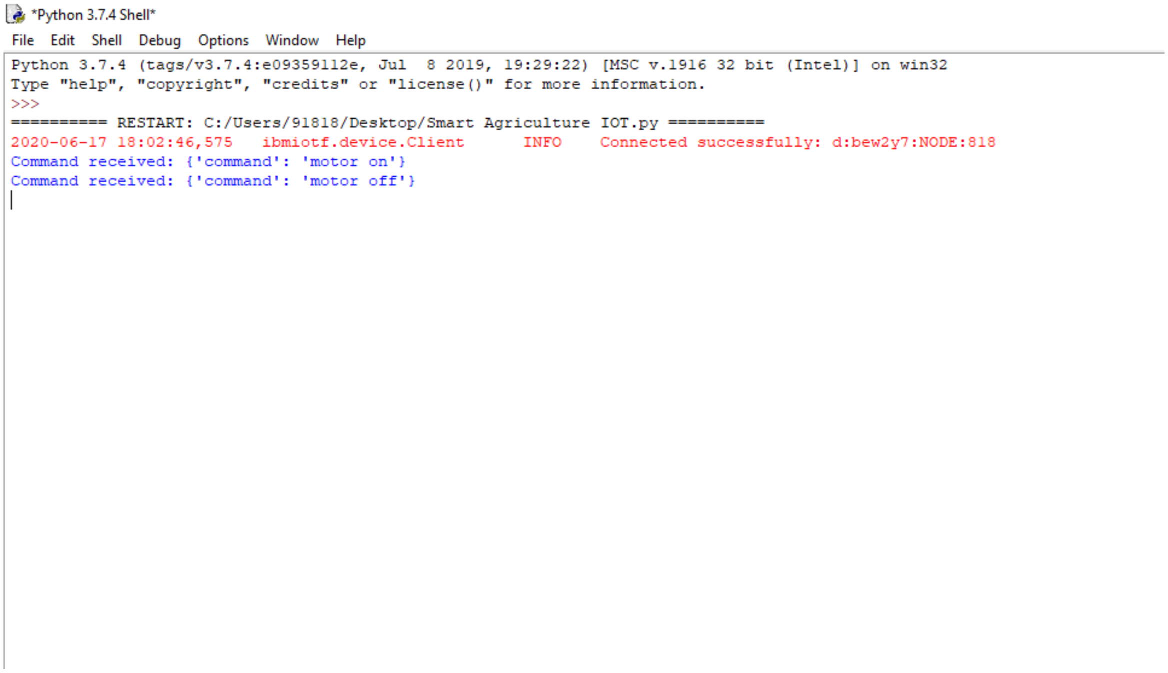
Nodes used in Web-App

- IBM IOT IN &OUT Nodes
- Function Nodes
- Debug Node
- Gauge Nodes
- Chart Nodes
- Button Nodes

- Timestamp Node
- Text NodesHttp request Node

We have successfully built a web-based UI and integrated the services using Node-RED.

By, configuring the nodes MOTOR ON/OFF we can control the motor. If we click the button MOTOR ON the motor status is MOTOR ON. If we click the button MOTOR OFF the motor status is MOTOR OFF respectively. Then the command is received to the motor & shows the output. The output can be seen on the python IDLE.



```
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul  8 2019, 19:29:22) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:/Users/91818/Desktop/Smart Agriculture IOT.py ==========
2020-06-17 18:02:46,575   ibmiotf.device.Client      INFO    Connected successfully: d:bew2y7:NODE:818
Command received: {'command': 'motor on'}
Command received: {'command': 'motor off'}
```

ADVANTAGES & DISADVANTAGE :

The agriculture sector fulfils demand of food of the nation. The agriculture which uses sensors and latest technologies such as IoT/cellular is known as smart agriculture or smart farming. The Smart Agriculture System is an IoT based device which is capable of automating the irrigation process by analysing the moisture of soil and the climate condition.

ADVANTAGES:

- It allows farmers to maximize yields using minimum resources such as water, fertilizers etc
- Mobile operated pumps save cost of electricity
- It is cost effective method.
- It delivers high quality crop production.
- Real-Time Data and Production Insight
- Improved Livestock Farming.
- Remote Monitoring.
- Efficient and saves time

DISADVANTAGES:

- The smart agriculture need availability on internet continuously.
- The smart farming-based equipment’s require
- farmers to understand and learn the use of technology.
- Lesser employment of menial staff or unskilled workers.

APPLICATIONS OF SMART AGRICULTURE SYSTEM:

Smart agriculture through the use of IOT Technologies will help farmers to reduce generated wastes and enhance productivity. It is basically a hi-tech system of growing food that is clean and is sustainable for the masses. It is the induction as well as the application of modern ICT (Information and Communication Technologies) into agriculture. Some of the applications for IOT in agriculture are:

- Precision Farming
- Agriculture Drones
- Livestock Monitoring
- Smart Greenhouses

- Wearables
- Smart City
- Smart Grids
- Industrial Internet
- Connected Car. ...
- Connected Health (Digital Health/Telehealth/ Telemedicine) ...
- Smart Retail.

CONCLUSION & FUTURE SCOPE:

CONCLUSION:

The proposed model explores the use of IOT in the agriculture sector. The objectives of this project “Smart Agriculture System” is to increase the crop production and to avoid the wastage of water. This smart agriculture system is feasible and cost effective. It also focuses on optimizing water resources which combats issues like water scarcity and ensures sustainability and monitors the environmental parameters. It also focuses on the utilization of IOT in agriculture and the solutions proposed in this paper will improve farming methods, increase productivity and lead to effective use of limited resources. A farmer should visualize his agricultural land’s moisture content and the weather conditions from time to time and water level of source is sufficient or not. The agriculture field can be monitored and controlled by an android app at user end. Hence ,will have a good production of crops and great saving of irrigation water, stronger and healthier plants.

FUTURE SCOPE:

The future scope of this project could be including variety of soil sensors like pH sensor, Rain sensor which helps in the predicting and analyzing processes more accurate. We can also monitor the life cycle of the plants. Through smart agriculture system we can analysis the crop yield and can monitor the soil moisture and plant health.

IOT sensors capable of providing farmers with information about crop yields, rainfall, pest infestation, and soil nutrition are invaluable to production and offer precise data which can be used to improve **farming** techniques over time.

Source code:

```
import time
import sys
import ibmiotf.application # to install pip install ibmiotf
import ibmiotf.device

#Provide your IBM Watson Device Credentials
organization = "bew2y7" #replace the ORG ID
deviceType = "NODE"#replace the Device type wi
deviceId = "818"#replace Device ID
authMethod = "token"
authToken = "123456789" #Replace the authtoken

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

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:

    deviceCli.commandCallback = myCommandCallback

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