Project Scope

- **Project Summary**

    - Smart Agriculture System based on IoT can monitor soil moisture and climatic conditions to grow and yield a good crop.

    - A Farmer is provided a mobile app using which he can monitor the temperature, humidity and soil moisture parameters along with weather forecasting details.

    - Based on all the parameters he can water his crop by controlling the motors using the mobile

- **Project Team**

- Anushka Rai

- **Project Requirements**

    - IOT Application Development

    - IOT Cloud Platform

- **Software Requirements**

    - GIT tool and Nodered

    - Python IDE

- **Project Deliverables**

        - Explore IBM Cloud Platform

    - Connect The IOT Simulator To Watson IOT Platform

- Configure The Nodered To Get The Data From IBM IOT Platform And Open Weather API
- Building A Web App
- Configure Device To Receive The Data From The Web Application And Control Motors

Project Report

Smart Agriculture System based on IOT

Name : Anushka Rai

Internship : Smart Agriculture system based on IoT

Contents

1. Introduction

1 .1 Overview

1. Agriculture is a major source of income for the largest population in India and is a major contributor to the Indian economy.

2. In the past decade it is observed that there is not much crop development in the agriculture sector.

3. Food prices are continuously increasing because crop rate declined.

4. There are number of factor which handles this, it may be due to water waste, low soil fertility, Fertilizer abuse, climate change or diseases etc.

5. It is very essential to make effective intervention in agriculture and

the solution is IOT in integration with wireless sensor network.

6. The Internet of things (IOT) is a method of connecting everything to the internet - it is connecting objects or things (such as car, home, electronic devices, etc.) which are previously not connected with each other.

7. The main purpose of IOT is to ensure the delivery of right information to right people at the right time.

8. In agriculture irrigation is the important factor as the monsoon rain falls are unpredictable and uncertain.

2. Literature Survey

2 .1Proposed Solution

IoT is based on SMART AGRICULTURE SYSTEM is regarded as the IoT gadget focusing on Live Monitoring of Environmental data in terms of Temperature, Moisture and Humidity of atmosphere and the plant/crop. The system provides the concept of “Plug and Sense” in which farmers can directly implement smart farming by such as pung the System on the field and geng Live Data feeds on various electronic devices using Web Application. Moreover, the data generated via sensors can be easily shared and viewed by agriculture consultants anywhere remotely via Cloud Compung

technology integration. The system allows manually to turn the pumping motor ON and OFF on sensing the moisture content of the soil.

3. Project Description

1. Smart Agriculture System based on IoT can monitor soil moisture and climatic conditions to grow and yield a good crop.

2. The farmer can also get the realtime weather forecasting data by using external platforms like Open Weather API.

3. Farmer is provided a mobile app using which he can monitor the temperature,humidity and soil moisture parameters along with weather

forecasting details.

4. Based on all the parameters he can water his crop by controlling the motors using the mobile application.

5. Even if the farmer is not present near his crop he can water his crop by controlling the motors using the mobile application from anywhere.

6. Here we are using the Online IoT simulator for getting the Temperature,Humidity and Soil Moisture values.

4. Project Scope

4.1 Project Summary

1. Smart Agriculture System based on IoT can monitor soil moisture and climatic conditions to grow and yield a good crop.

2. A Farmer is provided a mobile app using which he can

monitor the temperature,humidity and soil moisture parameters along with weather forecasting details.

3. Based on all the parameters he can water his crop by controlling the motors using the mobile application

a. Project Requirements

1. IOT Application Development

2. IOT Cloud Platform

4.3 Software Requirements

1. GIT tool and Nodered

2. Python IDE

4.4 Project Deliverables

1. Explore IBM Cloud Platform

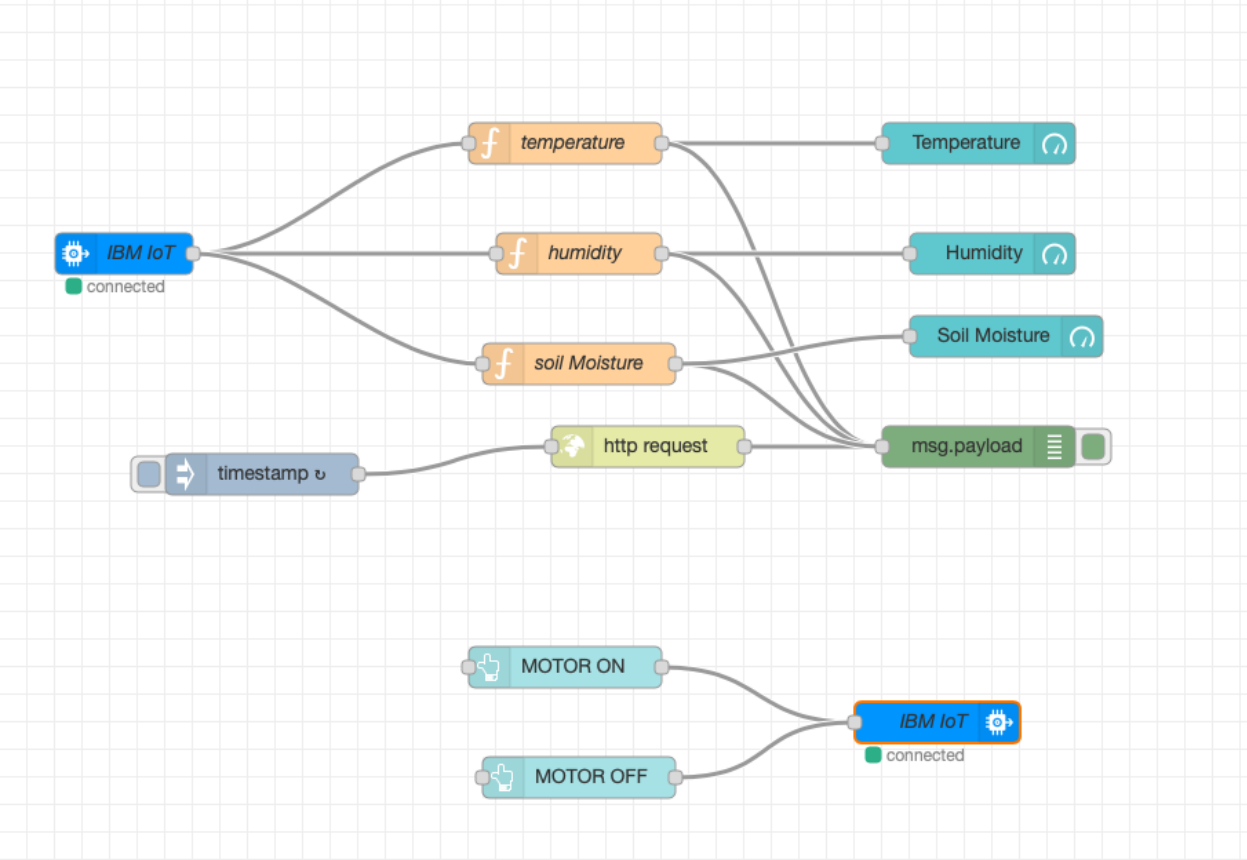
2. Connect The IOT Simulator To Watson IOT Platform

3. Configure The Nodered To Get The Data From IBM IOT Platform And Open Weather API

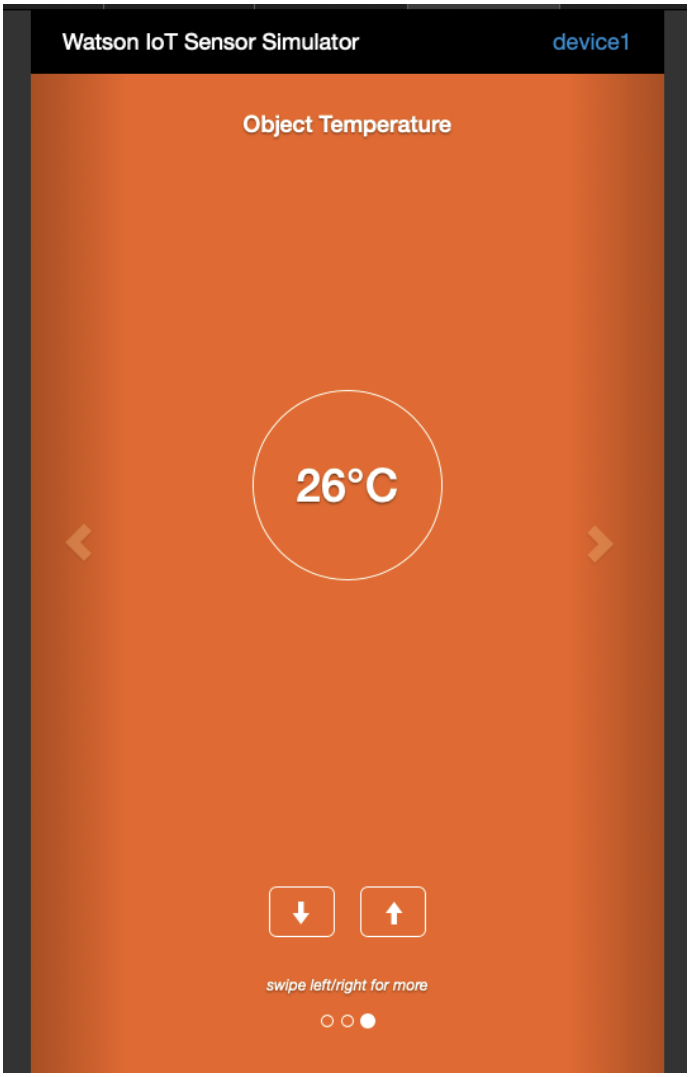
4. Building A Web App

5. Configure Device To Receive The Data From The Web Application And Control Motors

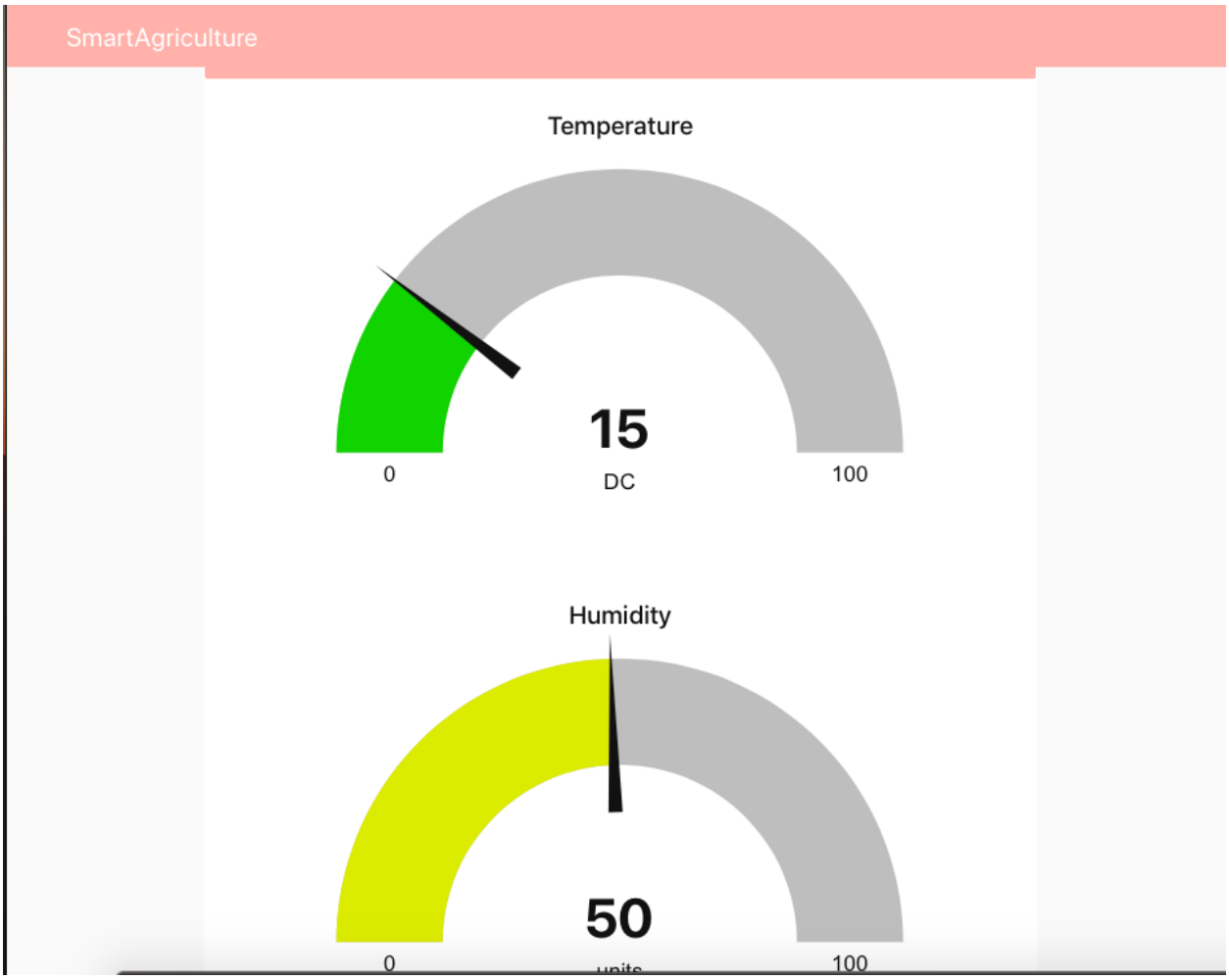
5.Node-Red Flow



6. Watson IOT Sensor



7. Node-Red UI Dashboard



8. Advantages and Disadvantages

8 .1 Advantages

1. Small sized customer support team is enough as only a low volume of queries are redirected.

2. Cost efficient.

3. Only a small amount of queries need representatives, hence the customer i s also satisfied with quick and easy solution.

8.2 Disadvantages

1. The IBM Watson Discovery service returns wrong results if not properly configured.

2. Same answers might be given for different queries

9. Applications

1. Using the IoT concept in the agriculture field will help farmers not only reduce waste but also increase in yield production varying from the quantity of fertilizer utilized to the quality of the production achieved.

2. These days IoT has also been implemented in these following practices.

    a. Crop Monitoring: Using IoT technique we can monitor the quality of crop.

    b. Precision Farming: Precision farming is a farming practice that is more accurate and controlled. It deals with production of crop along with raising livestock.

    c. Livestock Monitoring: With the help of sensor, health of the livestock can be monitored which will directly help in the yield production of good produced from them.

    d. Agricultural drones: It is a good example for farming and in order to improve the various agricultural practices.

10. Conclusion

By following the above-menoned steps, we create a basic chatbot which can help

us to answer the basic quesons of the customer or user related to location of the office, working hours and the information about the product. We have successfully created the intelligent help desk smart chatbot using IBM Watson Assistant, IBM Cloud Function, IBM Watson Discovery and Node-Red.

11. Future Scope

In the future, the bot can be improvised with live UI. Also, features like similar information, helpline contacts, Watson text to audio, extended user requests, a database to store and display recent chats, etc.

12.Source Code

12.1 Node-RED Flow (json)

[{"id":"63c4097.d1b20f8","type":"tab","label":"Flow
1","disabled":**false**,"info":""},{"id":"cf604055.a51c","type":"debug","z":"63c4097.d1b20f8","name":"","active":**true**,"tosidebar":
true,"console":**false**,"tostatus":**false**,"complete":"payload","targetType":"msg","x":730,"y":440,"wires":[]},{"id":"3498946b.22
ccac","type":"function","z":"63c4097.d1b20f8","name":"temperature","func":"msg.payload=msg.payload.d.temperature;\
nreturn
msg;","outputs":1,"noerr":0,"x":430,"y":220,"wires":[["e68ff193.2acf9","cf604055.a51c"]]},{"id":"e68ff193.2acf9","type":"ui_g
auge","z":"63c4097.d1b20f8","name":"","group":"4d20de1e.b8d6a8","order":4,"width":12,"height":7,"gtype":"gage","title":"Te
mperature","label":"DC","format":"{{value}}","min":0,"max":"100","colors":["#00b500","#e6e600","#ca3838"],"seg1":"","seg2":
"","x":730,"y":220,"wires":[]},{"id":"9d1e271c.a993d8","type":"http
request","z":"63c4097.d1b20f8","name":"","method":"GET","ret":"obj","paytoqs":**false**,"url":"api.openweathermap.org/data/
2.5/weather?q=Hyderabad,
IN&appid=3239928487eccbfc7fb7c22c822d2637","tls":"","persist":**false**,"proxy":"","authType":"","x":490,"y":440,"wires":[["c
f604055.a51c"]]},{"id":"e33f35d4.2b7428","type":"inject","z":"63c4097.d1b20f8","name":"","topic":"","payload":"","payloadTy
pe":"date","repeat":"10","crontab":"","once":**false**,"onceDelay":0.1,"x":210,"y":460,"wires":[["9d1e271c.a993d8"]]},{"id":"8e0fb
159.635078","type":"function","z":"63c4097.d1b20f8","name":"humidity","func":"msg.payload=msg.payload.d.humidity\n
return
msg;","outputs":1,"noerr":0,"x":440,"y":300,"wires":[["ac3a71d4.9b87d","cf604055.a51c"]]},{"id":"99065915.a59678","type":
"ui_button","z":"63c4097.d1b20f8","name":"","group":"4d20de1e.b8d6a8","order":2,"width":0,"height":0,"passthru":**false**,"la
bel":"MOTOR
ON","tooltip":"","color":"","bgcolor":"","icon":"","payload":"{\"command\":\"motoron\"}","payloadType":"json","topic":"","x":430,"
y":600,"wires":[["fbf2de1a.6ae398"]]},{"id":"ef412860.c97d4","type":"ui_button","z":"63c4097.d1b20f8","name":"","group":"4
d20de1e.b8d6a8","order":3,"width":0,"height":0,"passthru":**false**,"label":"MOTOR
OFF","tooltip":"","color":"","bgcolor":"","icon":"","payload":"{\"command\":\"motoroff\"}","payloadType":"json","topic":"","x":44
0,"y":680,"wires":[["fbf2de1a.6ae398"]]},{"id":"ac3a71d4.9b87d","type":"ui_gauge","z":"63c4097.d1b20f8","name":"","group"
:"4d20de1e.b8d6a8","order":6,"width":0,"height":0,"gtype":"gage","title":"Humidity","label":"units","format":"{{value}}","min":
0,"max":"100","colors":["#00b500","#e6e600","#ca3838"],"seg1":"","seg2":"","x":740,"y":300,"wires":[]},{"id":"ec4b27b4.3047
18","type":"function","z":"63c4097.d1b20f8","name":"soil
Moisture","func":"msg.payload=msg.payload.d.objectTemp;\nreturn
msg;","outputs":1,"noerr":0,"x":440,"y":380,"wires":[["cf604055.a51c","2b2f2804.8fce"]]},{"id":"2b2f2804.8fce","type":"ui_ga
uge","z":"63c4097.d1b20f8","name":"","group":"4d20de1e.b8d6a8","order":1,"width":"0","height":"0","gtype":"gage","title":"S
oil
Moisture","label":"units","format":"{{value}}","min":0,"max":"100","colors":["#00b500","#e6e600","#ca3838"],"seg1":"","seg2":
"","x":750,"y":360,"wires":[]},{"id":"170c801b.0e3e38","type":"ibmiot
in","z":"63c4097.d1b20f8","authentication":"quickstart","apiKey":"20b68643.13e922","inputType":"evt","logicalInterface":""
,"ruleId":"","deviceId":"device1","applicationId":"","deviceType":"+","eventType":"+","commandType":"","format":"json","name"
:"IBM
IoT","service":"quickstart","allDevices":"","allApplications":"","allDeviceTypes":**true**,"allLogicalInterfaces":"","allEvents":**true**,
"allCommands":"","allFormats":"","qos":0,"x":110,"y":300,"wires":[["3498946b.22ccac","8e0fb159.635078","ec4b27b4.3047
18"]]},{"id":"fbf2de1a.6ae398","type":"ibmiot
out","z":"63c4097.d1b20f8","authentication":"quickstart","apiKey":"20b68643.13e922","outputType":"evt","deviceId":"devi
ce1","deviceType":"1.0.6","eventCommandType":"agri","format":"json","data":"data points","qos":0,"name":"IBM
IoT","service":"quickstart","x":700,"y":640,"wires":[]},{"id":"4d20de1e.b8d6a8","type":"ui_group","z":"","name":"IOT","tab":"f98
e1edb.972628","order":1,"disp":**true**,"width":"12","collapse":**true**},{"id":"20b68643.13e922","type":"ibmiot","z":"","name":"","k
eepalive":"60","serverName":"https://api.eu-gb.discovery.watson.cloud.ibm.com/instances/88761352-79af-4e3c-bfc4-
ad33fea6b989","cleansession":**true**,"appId":"","shared":**false**},{"id":"f98e1edb.972628","type":"ui_tab","z":"","name":"Smart
Agriculture","icon":"dashboard","disabled":**false**,"hidden":**false**}]

12.2 Python Code

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

#Provide your IBM Watson Device Credentials
organization = "do7gzk" # repalce it with organization ID
deviceType = "IOTdevice" #replace it with device type
deviceId = "device1" #repalce with device id
authMethod = "token"
authToken = "anushkarai1"#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)

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()
```