PROJECT REPORT

INTRODUCTION

Overview: Smart Agriculture System Based on IOT

Purpose: To develop a Smart Agriculture System based on IoT that can monitor soil moisture and climatic conditions to grow and yield a good crop.

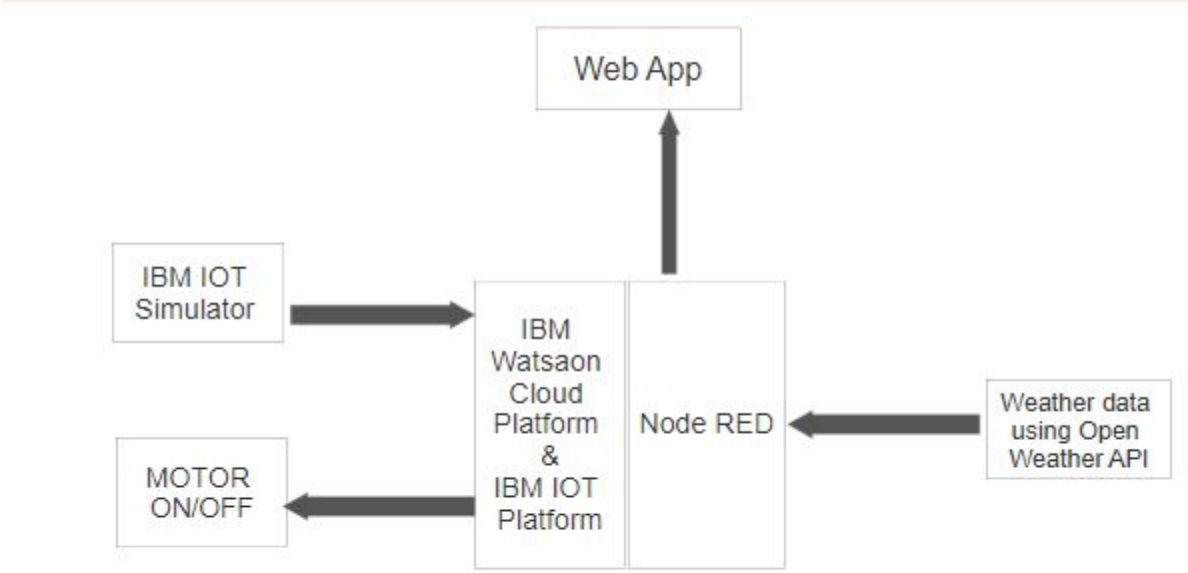
LITERATURE SURVEY

2.1 Existing problem : Problem is to monitor temperature, humidity and soil moisture in an agriculture system and keep farmers updated.

2.2 Proposed solution : Solution is to develop a “smart” agriculture that tells all the factors mentioned above through an IoTDevice which is connected to an IBM simulator and modifies the data as per OpenWeather API information.

THEORETICAL ANALYSIS

3.1 Block diagram



3.2 Hardware / Software designing

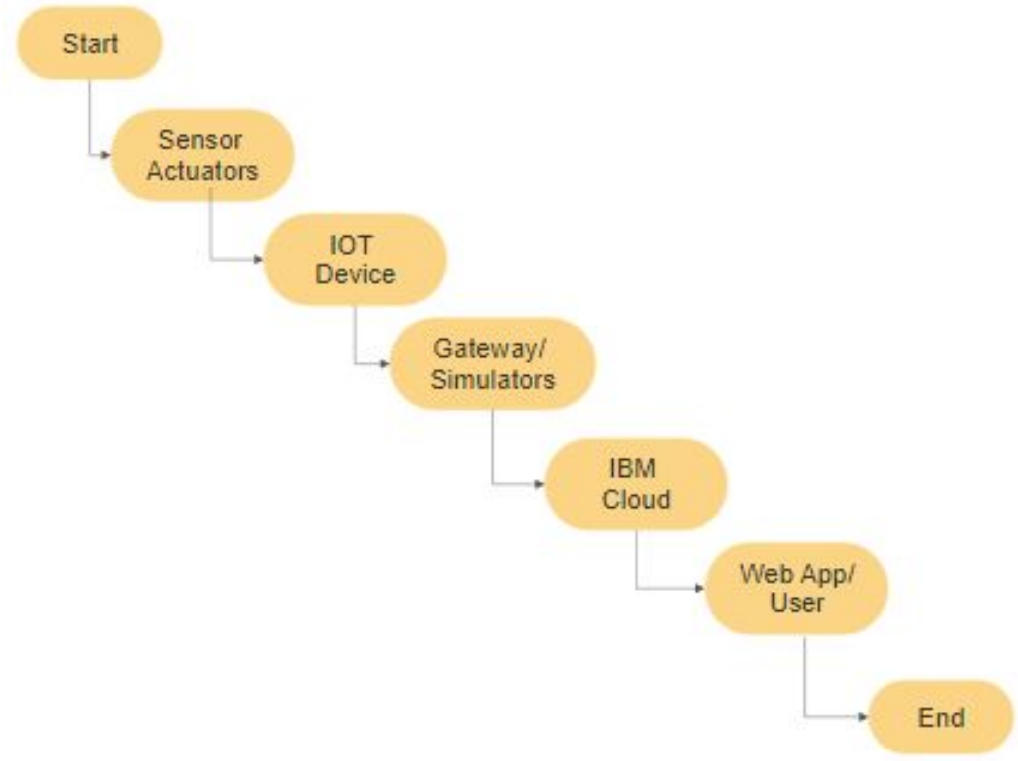
We are working on software designing only in this project. Software used here is Node-Red. Node-RED is a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways.

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.Node- red designing included installing node-red dashboard and ibmiot, working with flows , making webapp using nodes and debugging to see the output. Along with that functionalities to design webapp are also provided in various nodes and the data can be depicted in different types.

EXPERIMENTAL INVESTIGATIONS

- Insights into Node-Red
- Insights of Internet of things
- Insights of IBM Watson Platform
- Insights into IBM Cloud
- Knowing about HTTP/ MQTT protocols.
- Information about HTML, CSS, Json

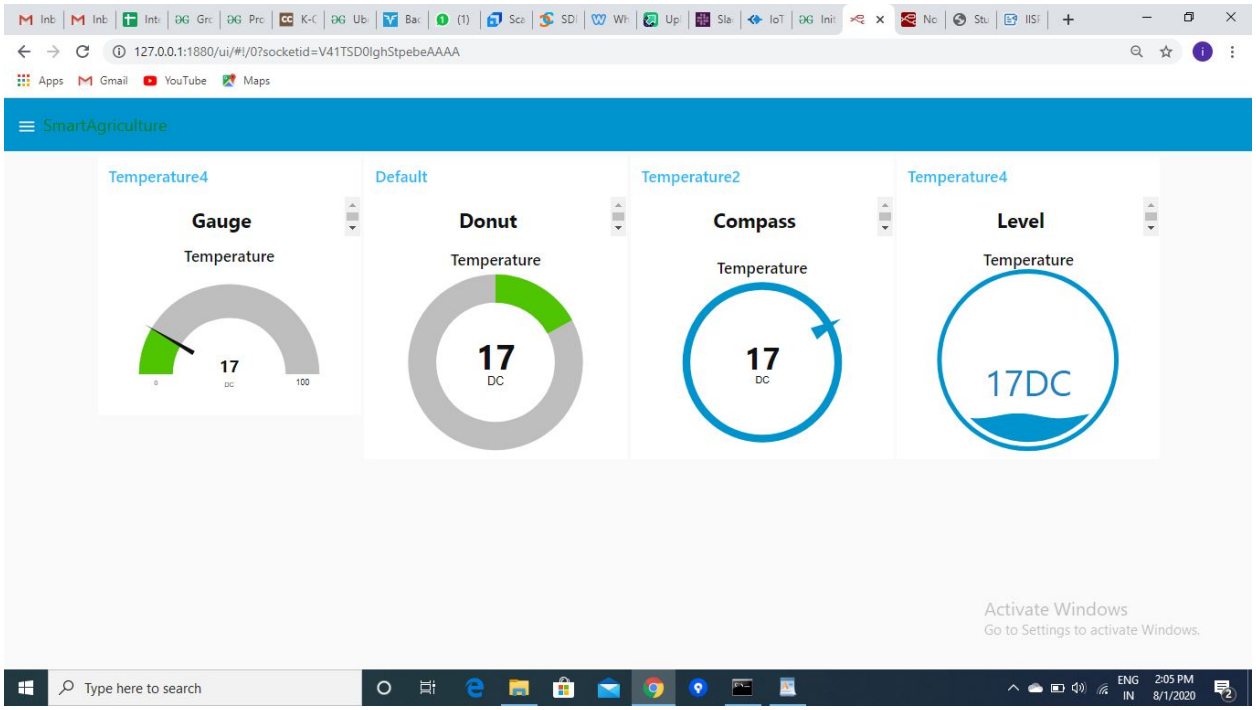
FLOWCHART



RESULT

We developed a Smart Agriculture System based on IOT that displays current climatic conditions and help farmers know how to proceed with watering or giving fertilisers quantitatively.

This is the node-red UI developed:

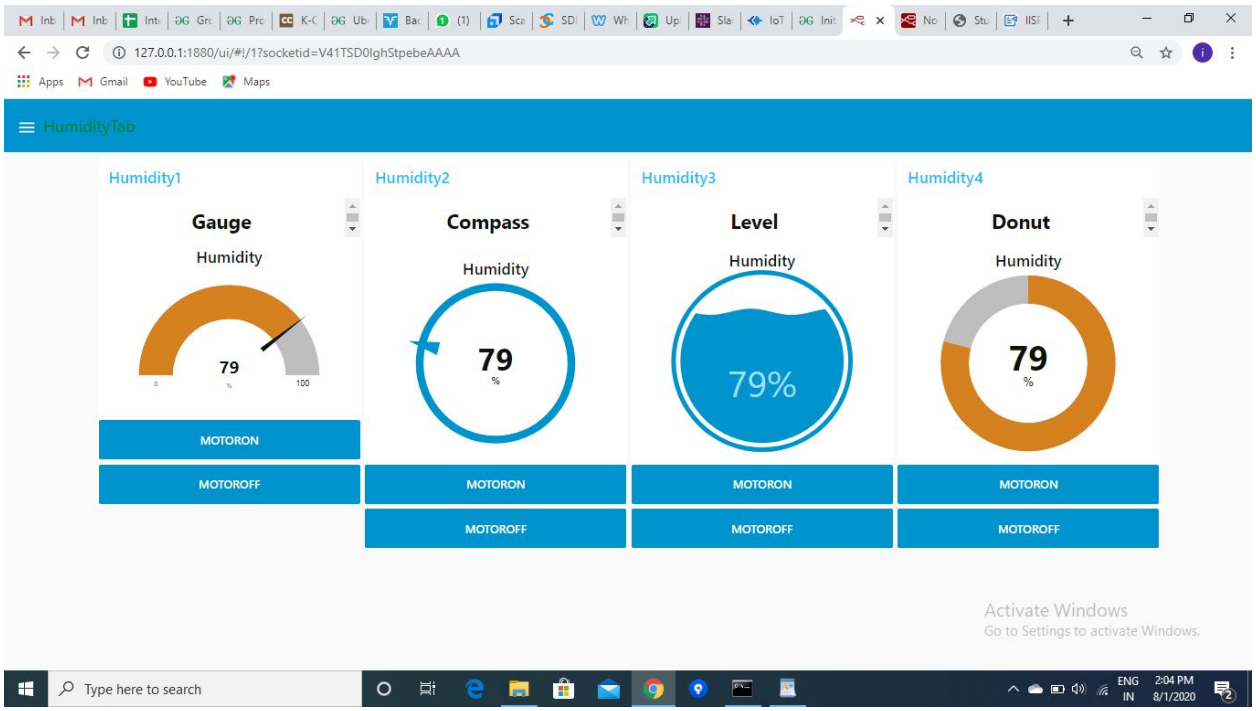


This is temperature tab . It displays 4 different types of gauge nodes.

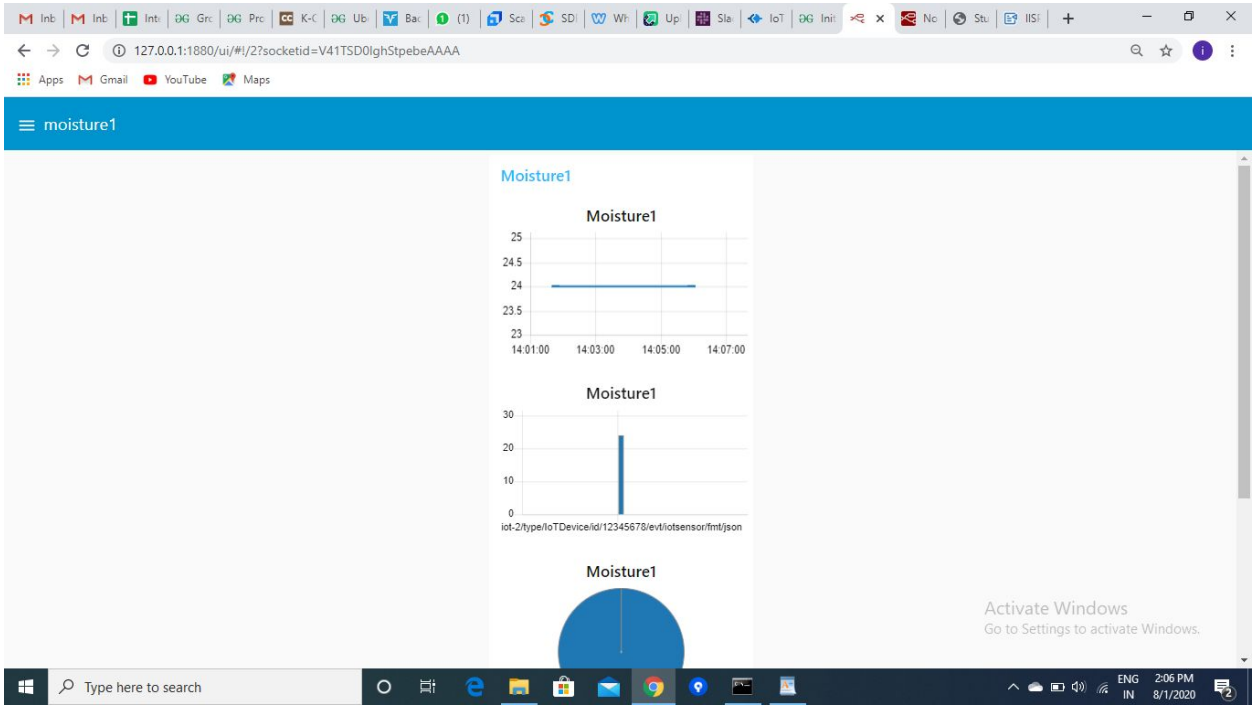
Gauge node is basically used to display temperature in four different graphical form.

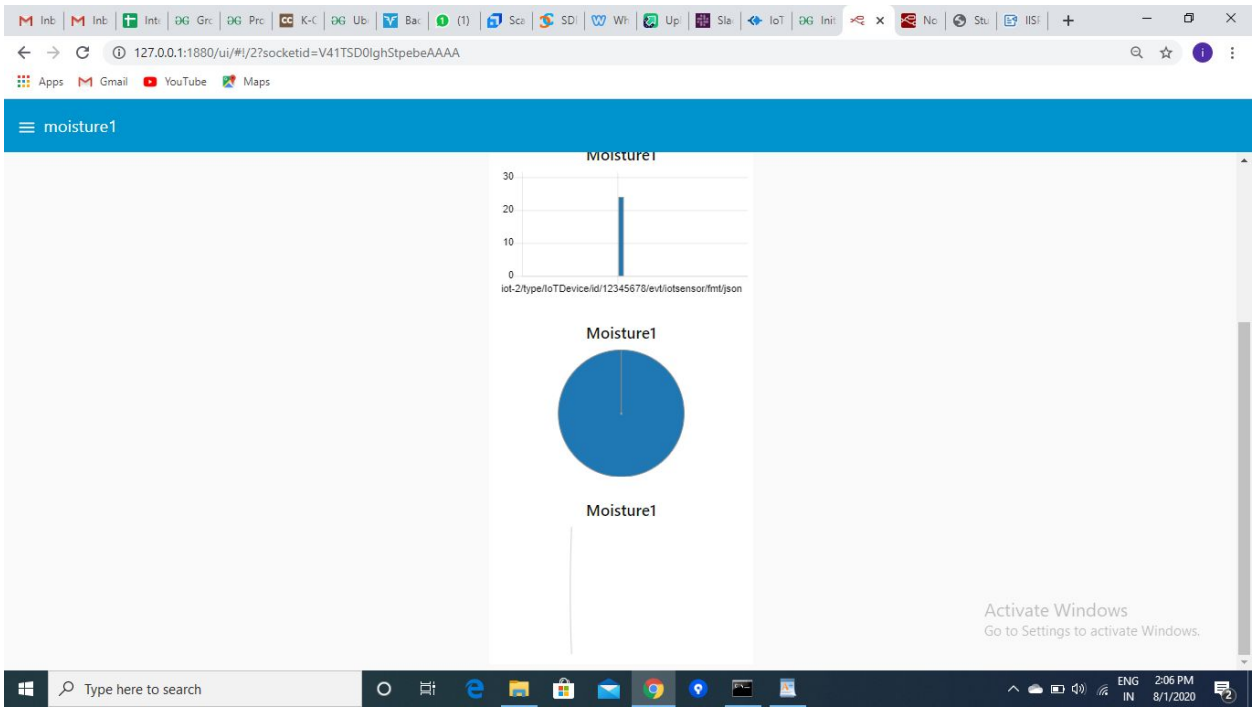
The temperature increased or decreased in IOT simulator will be displayed here.

These are displayed in single line by giving different names to groups and tags.



This is humidity tab . It consists of two buttons "motoroff" and "motoron". Clicking on these would give command at backend for motoroff and motoron and it will display output accordingly. These are added in order to allow farmer to control motor buttons even when he is not near the field.





The moisture is represented through graph nodes and I have used 4 different types of graph to display moisture. These are intentionally left in vertical order so as to depict how same group names behave.

ADVANTAGES & DISADVANTAGES

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 real time 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.

APPLICATIONS

1. Webapps determining climatic factors.
2. Pollution monitoring systems.
3. Webapps determining traffic management system.
4. Smart Street Lights
5. Robots and Artificial Intelligence
6. Smart Wheelchair, Alarm clocks etc.

CONCLUSION

Based on all the parameters,the farmer can water his crop by controlling the motors using the mobile application even if he is not present near his crop. Thus using IOT simulator one can develop smart projects and thus lead to smart India.

FUTURE SCOPE

IOT proves to have a huge scope as it provides a unique opportunity for businesses to turn data into insights. There are a number of contributing factors as well that drive the adoption of IOT such as improved sensors, device connections, the evolution of lifestyle and mobility. IOT projects once developed can prove to be a boon for masses.

BIBLIOGRAPHY

- IBM Watson Platform and IBM Cloud
- Wikipedia
- IoT Agenda Website
- https://nodered.org/docs/
- https://searchnetworking.techtarget.com/definition/protocol
- https://ieeexplore.ieee.org/document/8079928

Source code

```
import time

import sys

import ibmiotf.application

import ibmiotf.device


#Provide your IBM Watson Device Credentials

organization = "qb5ct7" # replace it with organization ID

deviceType = "IoTDevice" #replace it with device type

deviceId = "12345678" #replace with device id

authMethod = "use-token-auth"

authToken = "12345678"#replace with token


def myCommandCallback(cmd):

        print("Command received: %s" % cmd.data)

        if cmd.data['command']=='motoron':

                print("MOTOR ON")

        elif cmd.data['command'] == 'motoroff':

            print("MOTOR OFF")


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()


deviceCli.connect()


while True:

    T=50;

    H=32;

    #Send Temperature & Humidity to IBM Watson

    data = { 'Temperature' : T, 'Humidity': H }

    #print data

    def myOnPublishCallback():

        print ("Published Temperature = %s C" % T, "Humidity = %s %%" % H, "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
# Disconnect the device and application from the cloud
deviceCli.disconnect()
```

SUBMITTED TO:

Sir Durga Prasad

Smart Internz, Mentor

SUBMITTED BY:

Ishika Bansal

IGDTUW

Smart Internz, Intern