

Smart Agriculture System Based on IoT

REPORT

CONTENT

1 INTRODUCTION

1.1 Overview

1.2 Purpose

2 LITERATURE SURVEY

2.1 Existing problem

2.2 Proposed solution

3 THEORITICAL ANALYSIS

3.1 Block diagram

3.2 Hardware / Software designing

4 EXPERIMENTAL INVESTIGATIONS

5 FLOWCHART

6 RESULT

7 ADVANTAGES & DISADVANTAGES

8 APPLICATIONS

9 CONCLUSION

10 FUTURE SCOPE

11 BIBILOGRAPHY

APPENDIX

A. Source code

1. INTRODUCTION

1.1 Overview

In India, agriculture plays an important role for development in food production. In our country, agriculture depends on the monsoons which are not a sufficient source of water. So irrigation is used in agriculture field. Internet of Things (IoT) is a milestone in the evolution of technology. IOT plays an important role in many fields, one of that is Agriculture by which it can feed billions of people on Earth in future.

1.2 Purpose

The objective of this project is to overcome the challenges in the irrigation system of agricultural activities. The system can be operated from remote location through wireless transmission so there is no need to concern about irrigation timing as per crop or soil condition. The farmer is notified with the information regarding field condition through mobile periodically. This system will be more useful in areas where there is scarcity of water and will be worth efficient with satisfying its requirements. The objective of the system is to

- a) conserve energy & water resources
- b) handle the system manually and automatically.

2. LITERATURE SURVEY

2.1 Existing problem

Due to unplanned use of water the ground water level is decreasing day by day, lack of rains and scarcity of land water also results in decrement in volume of water on earth. Nowadays, water shortage is becoming one of the biggest problems in the world. We need water in each and every field. In our day to day life also water is essential. Agriculture is one of fields where water is required in tremendous quantity. Wastage of water is the major problem in agriculture. Every time excess of water is give to the fields. There are many techniques to save or to control wastage of water in agriculture. If the field is a far away land, the farmer needs to travel along distance daily to monitor and irrigate his crops. This brings in the dire need of a system that proposes effective usage of water, without wasting even a drop and also a system that helps a farmer in controlling his crops even if he is away.

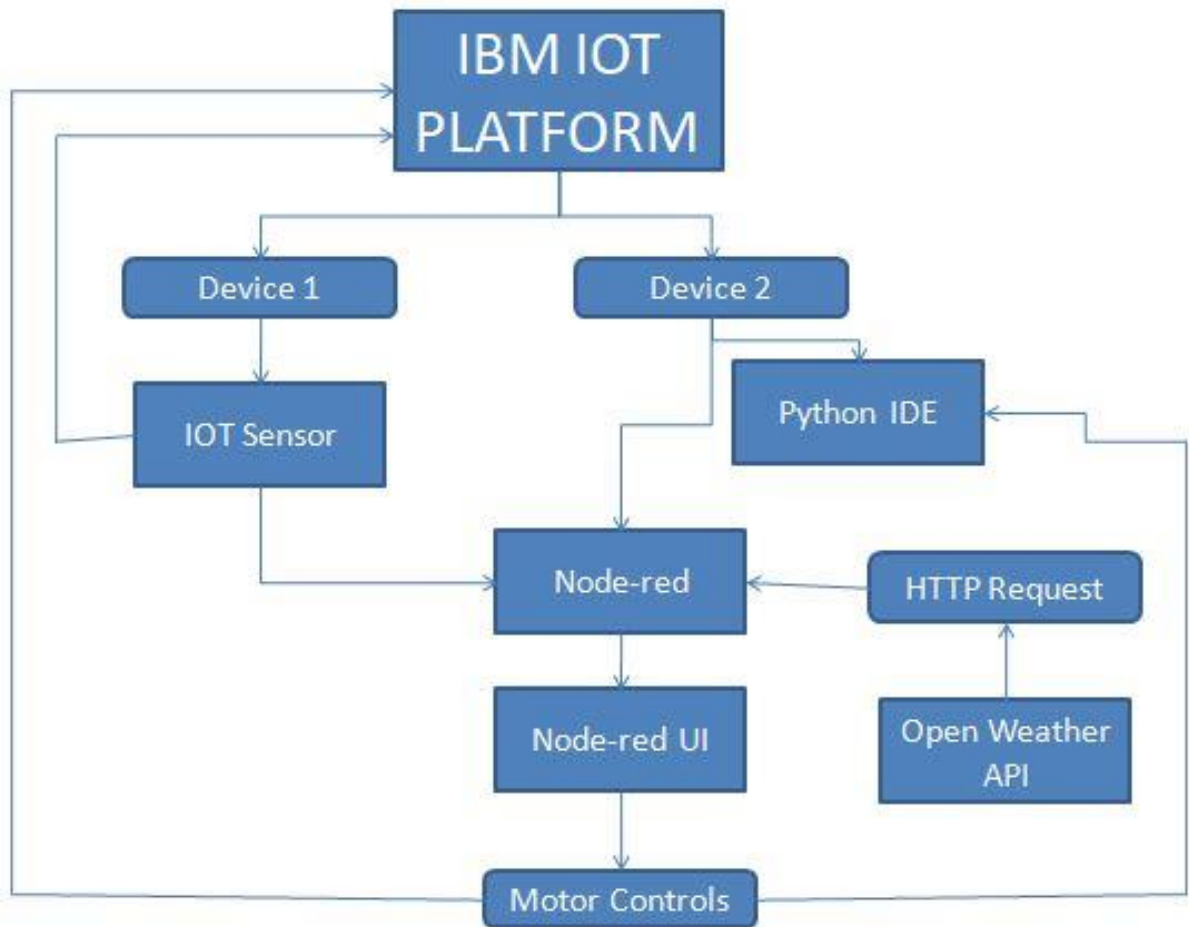
2.2 Proposed solution

Smart Agriculture System based on IoT helps a farmer yeild a good crop. The objective of the system is to a) conserve energy & water resources b) handles the system manually and automatically.

The farmer can watch his crops from anywhere using the mobile application. The app provides him details about soil moisture, climatic conditions, temperature, humidity and weather forecasting and other parameters with which the farmer can decide when and how much amount of water to be irrigated to his crops. He can irrigate the crops by controlling the motor using the application even when he is not near the field.

3. THEORETICAL ANALYSIS

3.1 Block diagram



3.2 Hardware/Software analysis

The main softwares used in this project are:

- IBM IOT Watson Platform
An IOT platform is created where all the sensor data and device data is displayed. 2 separate devices are created for IOT sensor and Control Panel.
- Node-Red
A work flow based virtual programming tool that is relying more on input-output with json as a data exchange format.
- Node-Red UI
An user interface where we are given the motor control options and the data on filed parameters, weather conditions etc.
- Open Weather API
Open Weather provides current weather forecasting data locally. Weather conditions are accessed using the API Key issued after logging in.
- IBM IOT Sensor
Used to connect and simulate the devices created from IBM IOT Platform to generate necessary data.
- Python IDE
The source code is written in python. The device credentials are provided in the python program so as to connect it to IOT Platform.

4 EXPERIMENTAL INVESTIGATIONS

• SmartInternz Dashboard

The screenshot shows the SmartInternz dashboard for a student. The top navigation bar includes 'SMARTINTERNZ', 'IoT App Developer', 'CAREER BASIC', 'STUDENTS', 'COMPANIES', 'EVENTS', 'MORE', and 'SETTINGS'. The left sidebar lists 'DASHBOARD', 'INTERNSHIPS', 'CHALLENGES', 'PROFILE', 'LEARNING PATH', 'CERTIFICATES', and 'CHANGE PASSWORD'. The main content area displays internship details for 'Smart Agriculture system based on IoT - SB37013'. It includes fields for Project ID (SPS_PRO_101), Project Title, Duration (16.2 Days), and an Internship Description link. Two progress indicators show 'Overall Project Progress' and 'Assigned Tasks Progress' at 100%. A team member 'J1' is listed. Below this, a 'MENTOR INSTRUCTIONS' section provides a list of guidelines for the internship, such as workspace access, duration, attendance, and deliverables. A 'HIDE' button is present next to the instructions.

SMARTINTERNZ | IoT App Developer | CAREER BASIC | STUDENTS | COMPANIES | EVENTS | MORE | SETTINGS

DASHBOARD | INTERNSHIPS | CHALLENGES | PROFILE | LEARNING PATH | CERTIFICATES | CHANGE PASSWORD

Internship Title : Smart Agriculture system based on IoT - SB37013
Project ID : SPS_PRO_101
Project Title : Smart Agriculture system based on IoT
Duration : 16.2 Days
Internship Description : [Click here to view the Internship Description](#)
Team : [View Team](#) | J1

Overall Project Progress: 100%
Assigned Tasks Progress: 100%

★ MENTOR INSTRUCTIONS | HIDE

- Click on Go to Workspace Option to access the Project Workspace
- Total Internship duration is 1 month, within this time you have to complete project with expected outcome
- References & Learning resources are provided for every activity
- Your login and logouts to the workspace are monitored, it is mandatory to maintain 5-days a week attendance.
- All the project deliverables shall be pushed to GitHub Repository & daily work status shall be updated to mentor via Slack Channel
- Use Zoho Writer to update the project documentation regularly
- Individual activity status shall be updated in the Kanban Board without fail
- Use commenting option on activity card to communicate with mentor in case of any query, Mentor replies can be accessed from Mentor View tab
- Once mentor approves all activities, you have to capture a project demonstration video and upload to the GitHub
- Your profile shall be filled completely to get the Internship Certificate, you can access the certificate anytime from the dashboard
- We wish you all the best!!

Activate Windows | Go to Settings to activate Windows.

• IBM IOT Watson Platform showing the connected status of devices

The screenshot shows the IBM Watson IoT Platform dashboard. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. The left sidebar lists 'All Devices', 'Diagnose', and other options. The main content area displays 'Browse Devices' with a table of connected devices. The table has columns for Device ID, Status, Device Type, Class ID, and Date Added. Two devices are listed: 'iot' and '1234', both with a status of 'Connected'. A 'Device Simulator' toggle is visible. The bottom of the page shows pagination information and a 'Cookie Preferences' link.

IBM Watson IoT Platform | si0520200494@smartinternz.com | ID: br0u70

Browse | Action | Device Types | Interfaces | Add Device +

Browse Devices | All Devices | Diagnose

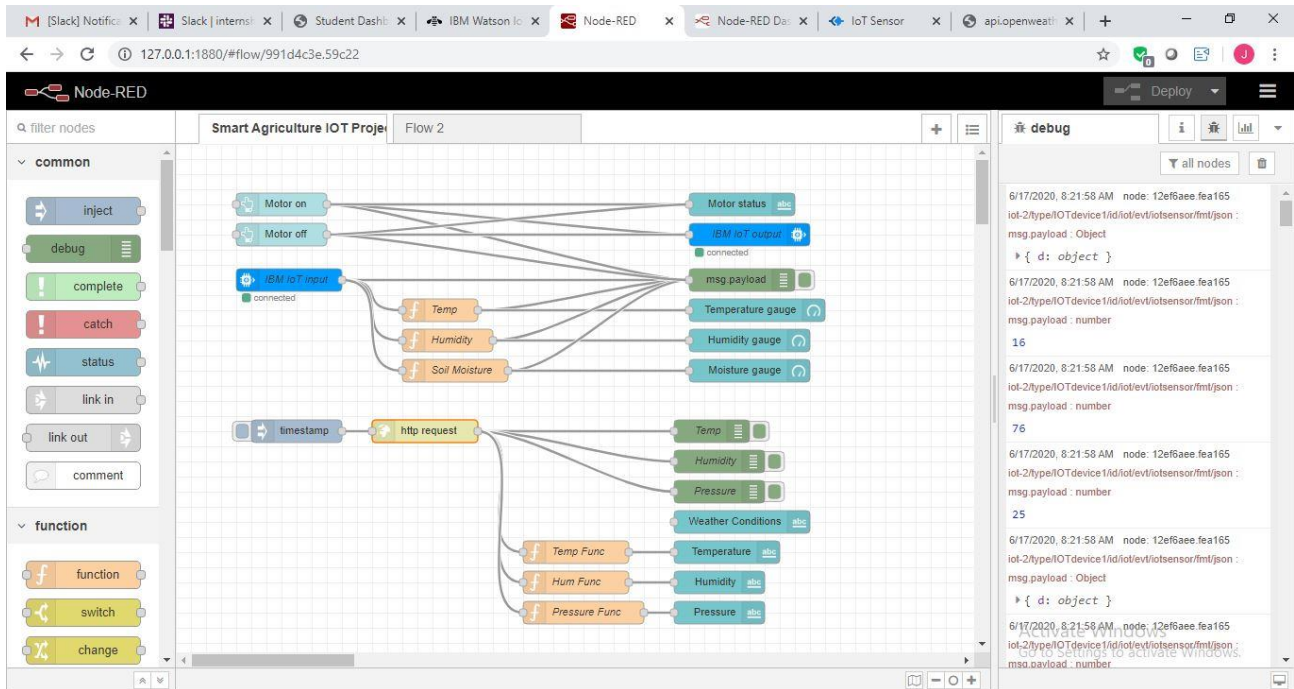
This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.

Search by Device ID | Device Simulator | 101 | Filter

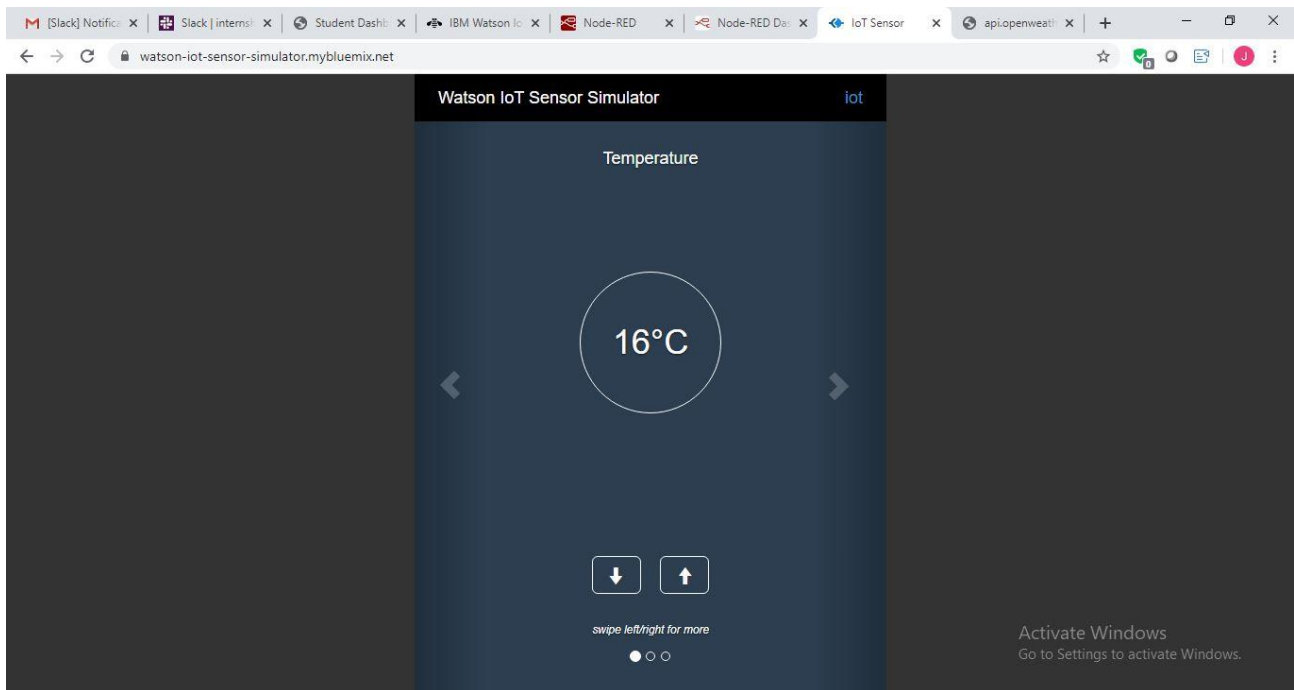
	Device ID	Status	Device Type	Class ID	Date Added
>	iot	Connected	IOTdevice1	Device	Jun 3, 2020 9:33 AM
>	1234	Connected	IOTdevice1	Device	Jun 10, 2020 4:00 PM

Items per page: 50 | 1-2 of 2 items | 1 of 1 page | Go to Settings to activate Windows. | Cookie Preferences

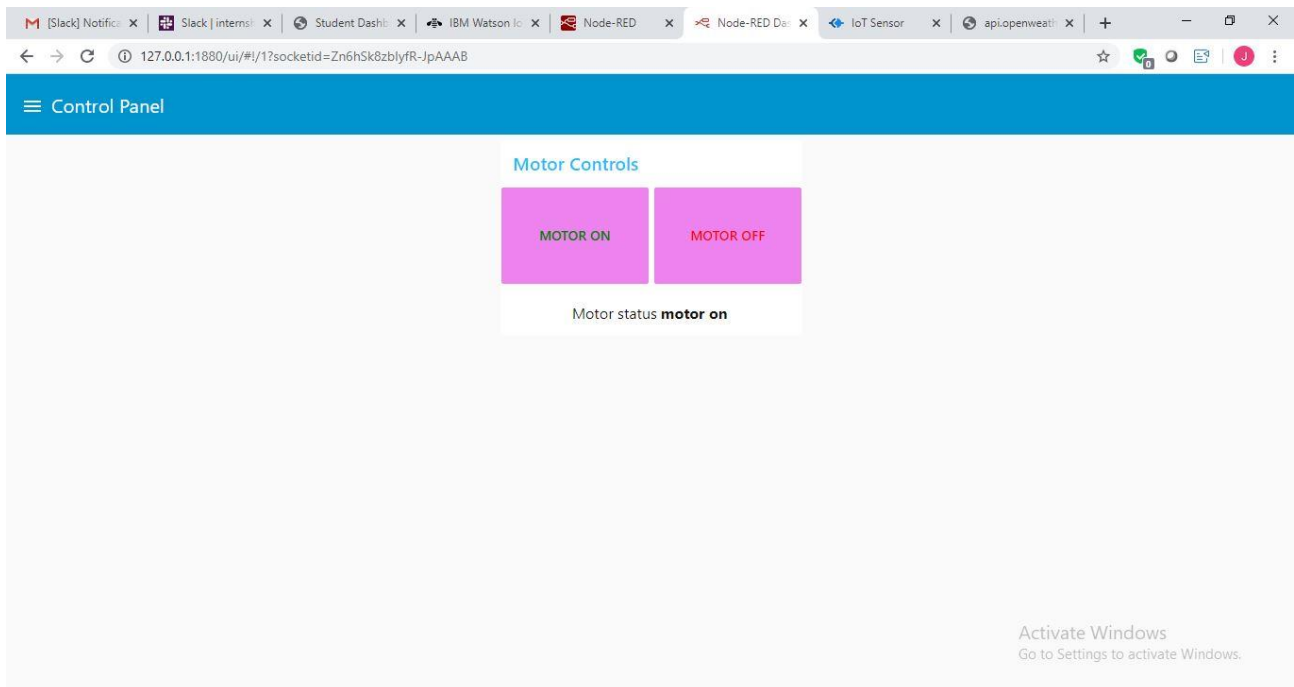
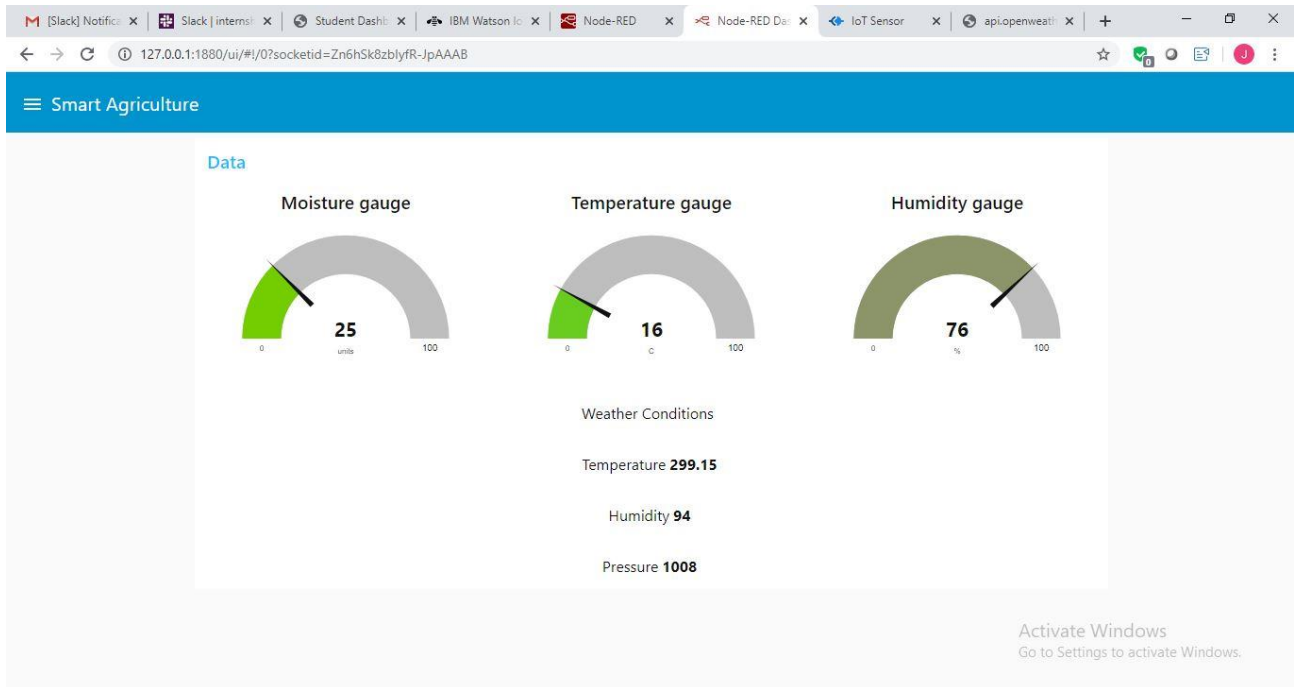
• Node-RED Flow



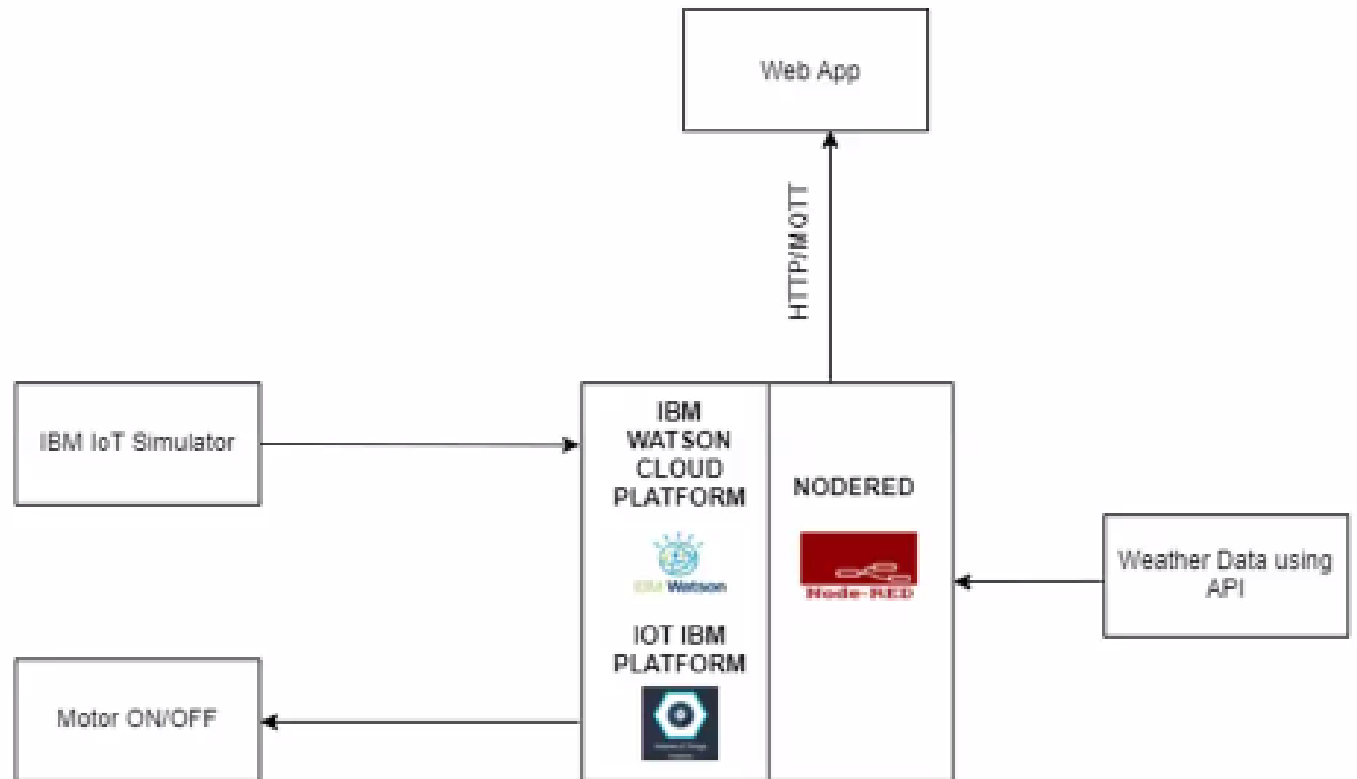
• IOT Sensor



• Node-RED UI



5 FLOWCHART



6 RESULT

Developed a web application that displays weather conditions of a particular area (the area selected by the user) and also temperature, humidity and soil moisture of the crops. User can control the motor operations and irrigate the crops after analyzing the environmental conditions of his crops.

7 ADVANTAGES AND DISADVANTAGES

Advantages:

- Conserve energy & water resources
- System can be handled manually and automatically
- System can be operated from a distance

Disadvantages:

- Over irrigation can occur if the farmer forgets to turn off the motor which will badly affect the crops. Therefore a timer mechanism should be implemented so as to turn off the motor after a specified interval of time if he forgets to turn off it.

8 APPLICATIONS

The application has found its base on agricultural field but can be extended to other areas too like constructional activities. It currently helps in:

- Displaying environmental conditions of the crops
- Controlling the motor operation
- Monitoring the crops even far away from the place

The application can be modified to meet new requirements and expectations.

9 CONCLUSION

The automated irrigation system implemented was found to be feasible and cost effective for optimizing water resources for agriculture production. This irrigation system allows cultivation in places with water scarcity thereby improving sustainability. The irrigation system helps the farmer by making his work smarter. As the demand for water increases, along with the need to protect aquatic habitats, water conservation practices for irrigation need to be effective and affordable. As multiple sensors are used, water can be provided only to the required area of land. This system reduces the water consumption to greater extent. It needs minimal maintenance. The power consumption has been reduced very much. The crop productivity increases and the wastage of crops are very much reduced.

10 FUTURE WORK

The extension work is to make user interface much simpler by just using SMS messages for notifications and to operate the switches.

Since different regions have different weather conditions, another feature that can be added to the system is displaying or providing information on which crop is suitable for which weather conditions.

Different crops need different amount of water. The amount of water required by each crop should also be made available so that the farmer can turn on the motor for the required moisture content.

11 BIBLIOGRAPHY

1. Joaquín Gutiérrez, Juan Francisco Villa-Medina, Alejandra NietoGaribay, and Miguel Ángel Porta- Gándara” Automated Irrigation System Using a Wireless Sensor Network and GPRS module” , Ieee Transactions OnInstrumentation And Measurement, Vol. 63, No. 1, January 2014.
2. “An Automatic Irrigation System using ZigBee in Wireless Sensor Network” 2015 International Conference on Pervasive Computing (ICPC)- IEEE 2015 by Pravina B. Chikankar, Deepak Mehetre , Soumitra Das Computer Engineering Department K J College of Engineering Management Research, Pune, India

APPENDIX

A. Source Code

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

#Provide your IBM Watson Device Credentials
organization = "br0u70" #replace the ORG ID
deviceType = "IoTdevice1"#replace the Device type wi
deviceId = "1234"#replace Device ID
authMethod = "token"
authToken = "12345678" #Replace the authtoken

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()
```