SMART IRRIGATION

IU1641150012 Harshil Maniar

IU1641150020 Juhi Kamdar

IU1641150023 Kaushil Kundalia

IU1641150026 Joy Mehta

Table of Contents

1. Introduction:

In the age of water scarcity, it is important that the water wastage is of minimal amount. One of the major domain where fresh water is used, is agriculture domain. In a case, where the soil is already moist, plant may not require more water. In such a case, watering the plants is useless. So, a solution for it, was to check the moisture content in the soil, and then add water. To make this autonomous, water pump was used for actuation. An additional sensor, temperature and humidity sensor is also added, so as to understand the moisture content in the air too, which can provide a naturally humid atmosphere. The values of the sensor are also uploaded on the cloud, so as to help monitor the conditions as well as for taking intelligent business decisions.

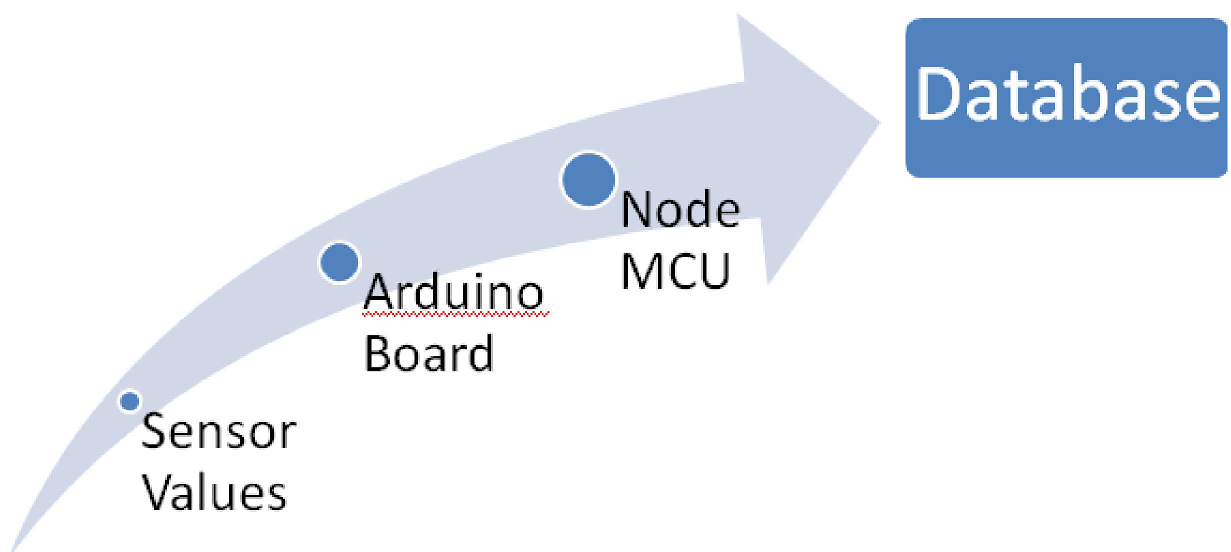


Fig.1 System Architecture

<u>1.1 System Functionalities</u>

The sensor attached to Arduino Board sense the values. These values are used by the Arduino
i) Check threshold values and perform Actuation of Water Pump
ii) Sending data to NodeMCU

The NodeMCU takes the values from Arduino Board, binds it into JSON object, and sends the data through POST request to a port of the server.

The connection Diagram can be given as below.

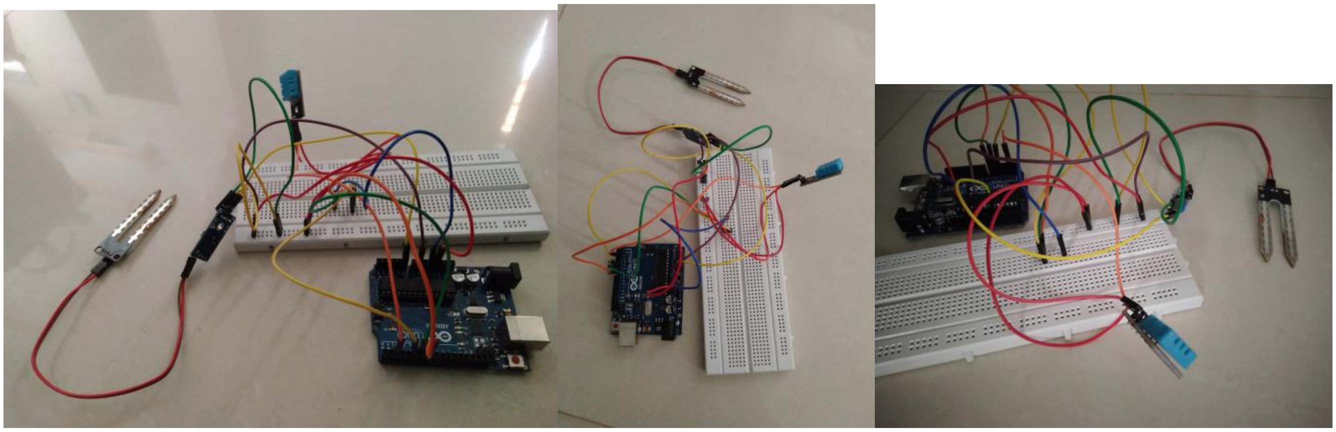


Fig. 2 Sensors' Connection

1.2 Project Overview:

- User Interface: Website
- Communication Interface: ESP8266 Node MCU
- Memory Storage: Cloud (on server by Digital Ocean)
- Design Constraints: Constant battery supply needed for the system to work. The server needs to be working all the time in order to store the values.

1.3 Technical Details:

1. *Communication Model:*
   Sensor to Arduino: Producer-Consumer Model
   Arduino to NodeMCU: Producer-COnsumer Mode
   NodeMCU to cloud: Publisher-broker-subscriber
   Cloud to User: Request-Response Model
2. *Connectivity Model:*
   Sensors to Arduino: Device to device
   Arduino to NodeMCU: Device to Gateway
   NodeMCU to Cloud: Gate to cloud
   Cloud to User: Backend

1.4 Functionalities:

- Autonomously water the plant
- Remote access to the conditions on the farm
- Water saving capabilities

1.5 Applications:

- Vast farmlands wherein it is tedious task for humans to reach everywhere
- Watering plants at home
- Watering the plants in public garden

2. Hardware Design and Description

2.1 Hardware Requirements:

a) Soil Moisture Sensor FC-28
b) Temperature and Humidity Sensor DHT-11
c) Mini Submersible Water Pump 3-6V
d) Arduino UNO board
e) Node MCU

Soil Moisture Sensor FC-28

Fig. 3 Soil Moisture Sensor

This is a simple water sensor can be used to detect soil moisture. It provides its reading in analog form. It has two electrodes made up of Fiber Glass. These probes sense the value and send to the module for further processing.

Temperature and Humidity Sensor DHT-11

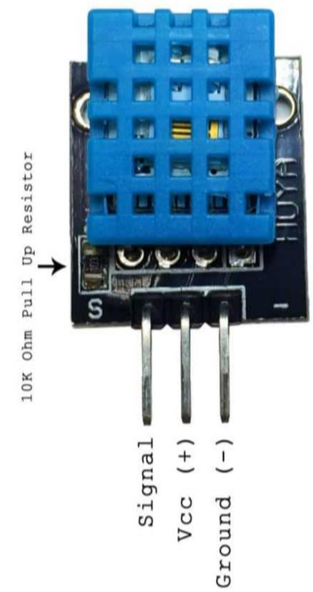


Fig. 4 Temperature and Humidity Sensor

Vcc: Power supply 3.5V to 5.5V
Data: Outputs both Temperature and Humidity through serial Data
Ground: Connected to the ground of the circuit

Mini Submersible Water Pump 3-6V

Fig. 5 Submersible Pump

This is a low cost, small size Submersible Pump Motor which can be operated from a 2.5 ~ 6V power supply. It can take up to 120 liters per hour with very low current consumption of 220mA. Just connect tube pipe to the motor outlet, submerge it in water and power it. We need to make sure that the water level is always higher than the motor. Dry run may damage the motor due to heating and it will also produce noise.

Arduino UNO board

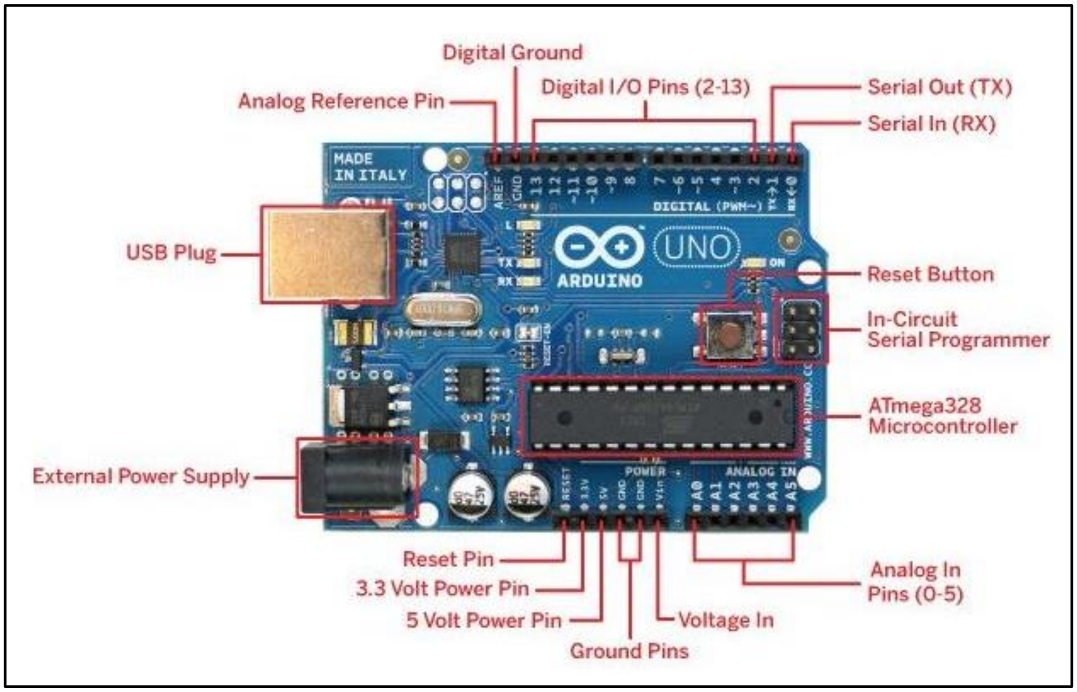


Fig. 6 Arduino Uni R3 Board and Its Pins

Power Supply: The power supply of the Arduino can be done with the help of an exterior power supply otherwise USB connection.

Vin: The input voltage or Vin to the Arduino while it is using an exterior power supply opposite to volts from the connection of USB or else RPS (regulated power supply).

5Volts: The RPS can be used to give the power supply to the microcontroller as well as components which are used on the Arduino board. This can approach from the input voltage through a regulator.

3.3V: A 3.3 supply voltage can be generated with the onboard regulator, and the highest draw current will be 50 mA.

GND: GND (ground) pins

Memory: The memory of an ATmega328 microcontroller includes 32 KB and 0.5 KB memory is utilized for the Boot loader, and also it includes SRAM-2 KB as well as EEPROM-1KB.

Serial Pins:The serial pins of an Arduino board are TX (1) and RX (0) pins and these pins can be used to transfer the TTL serial data.

External Interrupt Pins: The external interrupt pins of the board are 2 & 3, and these pins can be arranged to activate an interrupt on a rising otherwise falling edge, a low-value otherwise a modify in value

*PWM Pins:*The PWM pins of an Arduino are 3, 5, 6, 9, 10, & 11, and gives an output of an 8-bit PWM with the function analog Write ().

UART: An Arduino Uno uses the two functions like the transmitter digital pin1 and the receiver digital pin0. These pins are mainly used in UART TTL serial communication.

Arduino Cable

- Use it to connect Arduino Uno, Arduino Mega 2560, Arduino 101 or any board with the USB female A port of your computer.
- Cable length is approximately 178cm. Cable color and shape may vary slightly from image as our stock rotates.
- If you want to have a closer look to USB cables and standards check the USB cable pinouts referral page on pinouts.ru
- USB cable type A/B Standard USB 2.0 cable.

Node MCU

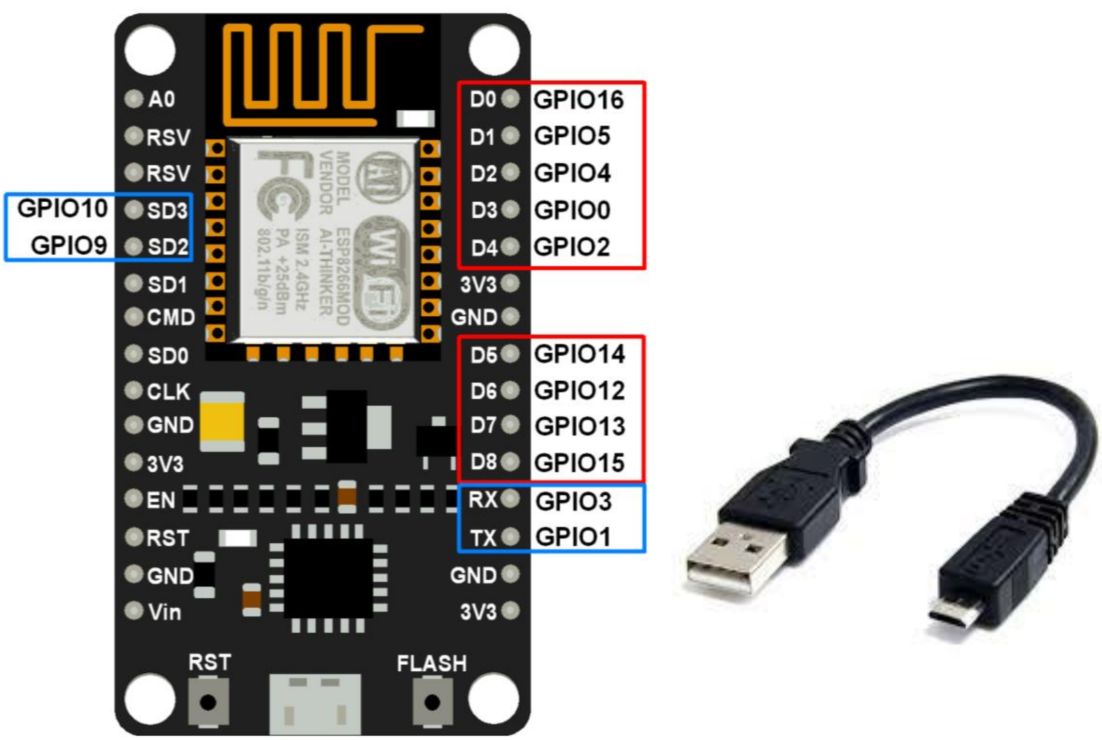


Fig. 7 Node MCU and its connecting wire(Micro USb)

ESP8266 is a system on a chip (SoC) design with components like the processor chip. The processor has around 16 GPIO lines, some of which are used internally to interface with other components of the SoC, like flash memory.

Since several lines are used internally within the ESP8266 SoC, we have about 11 GPIO pins remaining for GPIO purpose.

Now again 2 pins out of 11 are generally reserved for RX and TX in order to communicate with a host PC from which compiled object code is downloaded.

Hence finally, this leaves just 9 general purpose I/O pins i.e. D0 to D8.

As shown in the above figure of NodeMCU Dev Kit. We can see RX, TX, SD2, SD3 pins are not mostly used as GPIOs since they are used for other internal process. But we can try with SD3 (D12) pin which mostly like to respond for GPIO/PWM/interrupt like functions.

Note that D0/GPIO16 pin can be only used as GPIO read/write, no special functions are supported on it.

2.2 Component Connection Diagram:

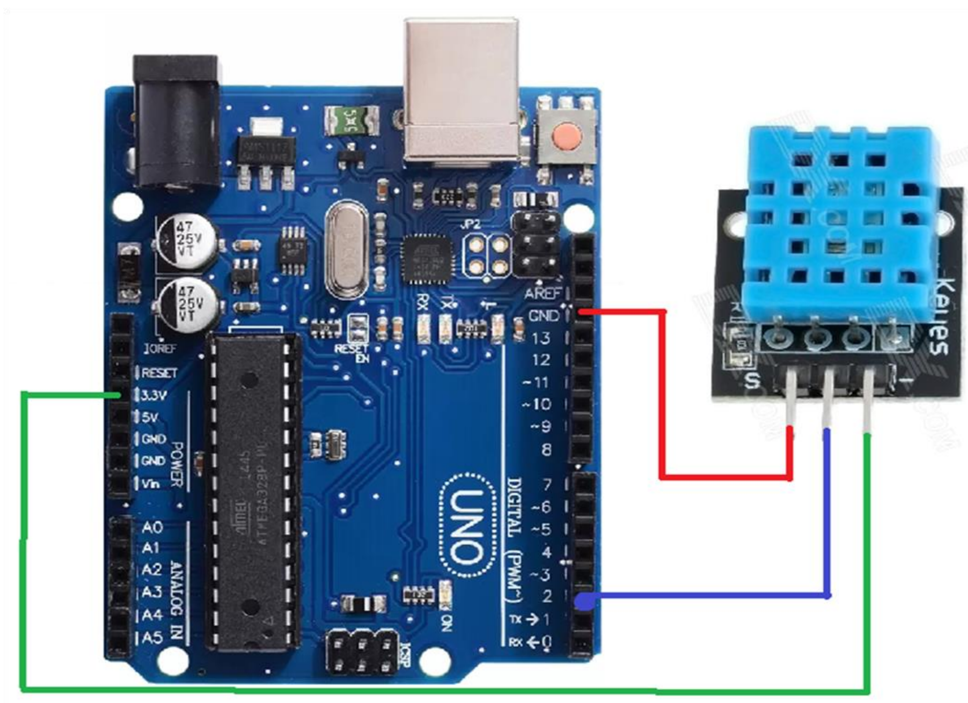


Fig .8 Arduino to Temperature and humidity Sensor Connection

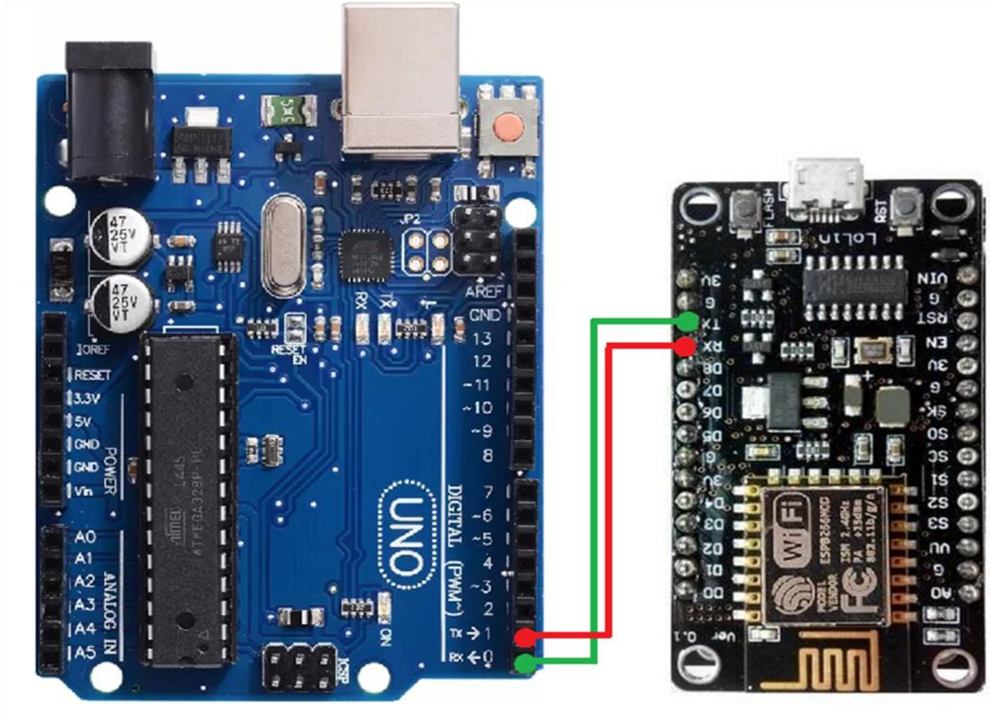


Fig. 9 Arduino to Temperature and Node MCU Connection

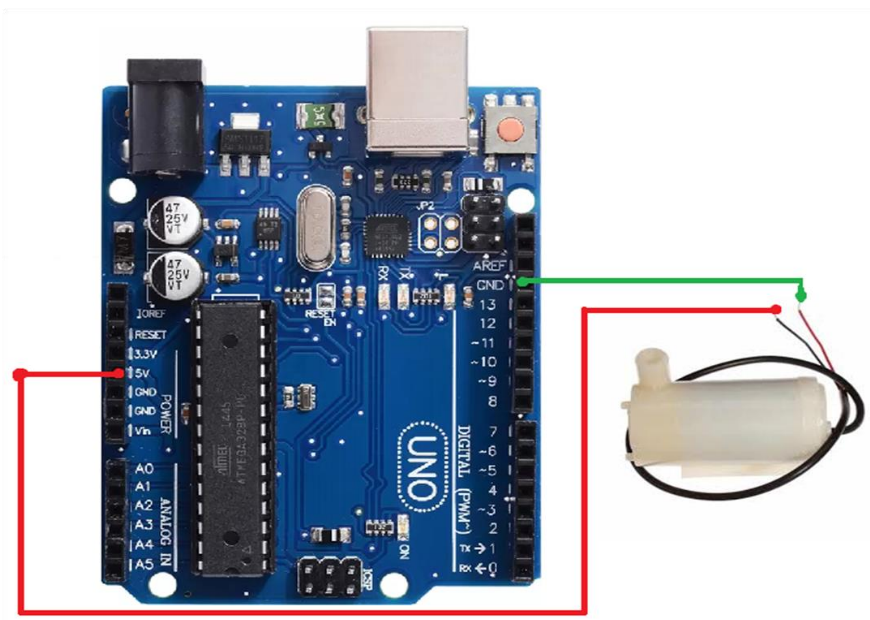


Fig. 10 Arduino to Water pump connection

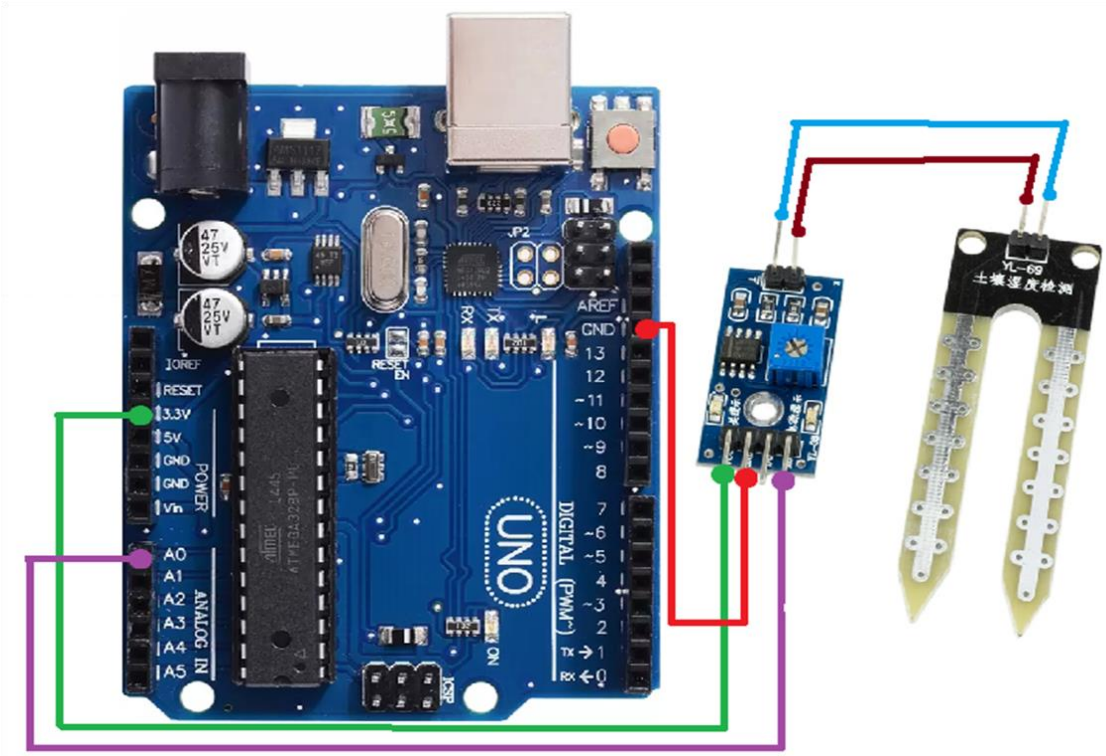


Fig. 11 Arduino to soil moisture Sensor connection

3. Software Design and Description

This chapter describes the software that is being used in the project.

3.1 Software Requirements

- Arduino Code Editor (Arduino IDE)

3.2 Algorithm for programming the arduino board

Algorithm:

Step 1: Get the values of all the sensors

Step 2: Print the values by sensors

Check for threshold, perform actuation of pump

Send the values to NodeMCU

3.3 Algorithm for programming the NodeMCU

Algorithm:

Step 1: Connect with WiFi

Step 2: Get the values from Arduino

Step 2: Pack the data into JSON object

Step 3: Send the JSON object to cloud using a POST request

3.4 Algorithm for programming the Cloud

Algorithm:

Step 1: Accept POST Request on Cloud

Step 2: Extract the values from the URL and commit the data on the SQLite database

Step 3: Display the database values when the user asks for it through the website

3.5 Software Design

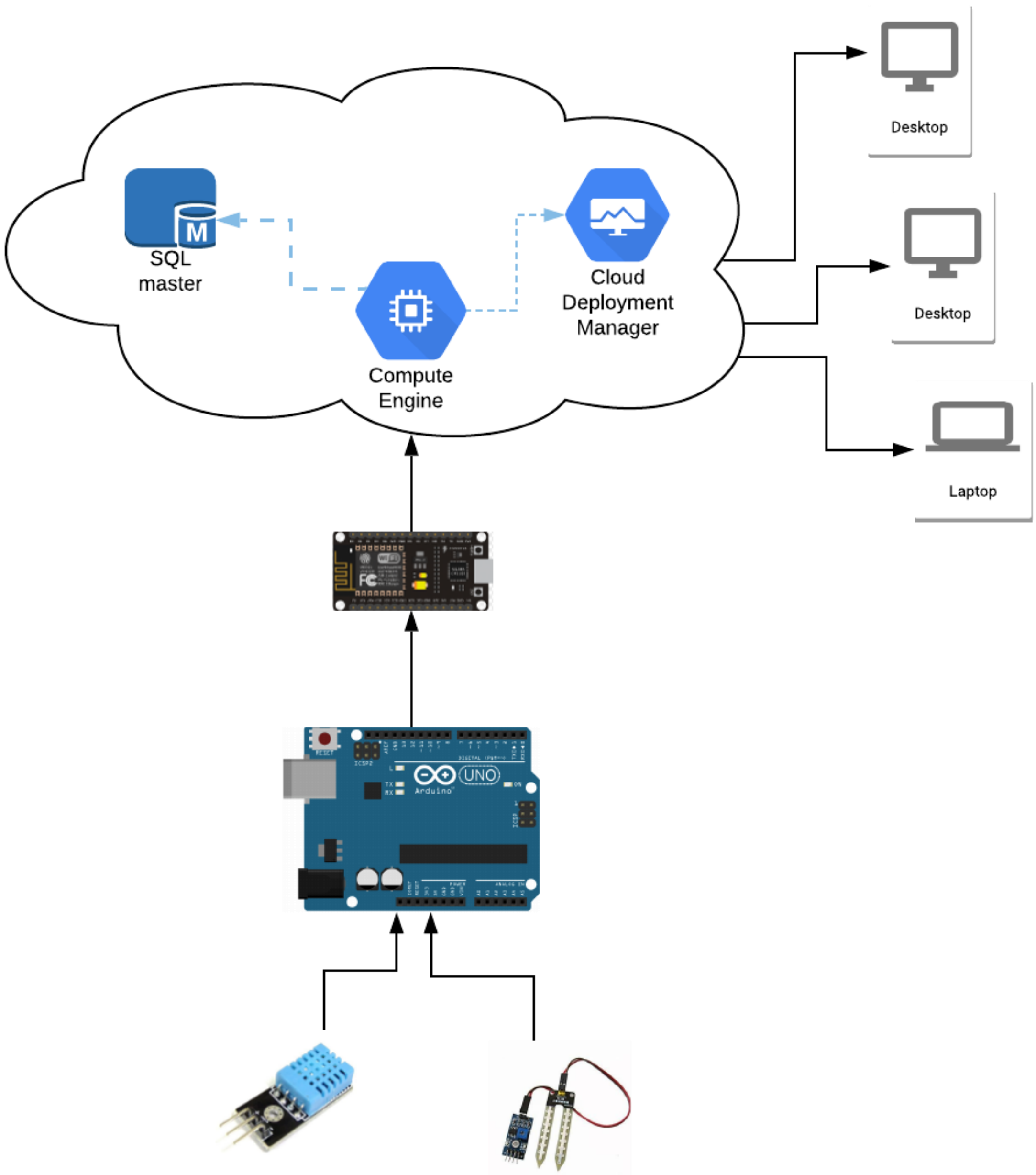


Fig. 12 System Design

4. Implementation

The Soil moisture sensor, and temperature and humidity sensor, are connected to the Arduino UNO R3 board. The Board is connected to the Node MCU. The power supply to both, Arduino and NodeMCU is provided using USB cable. Now, the data received by NodeMCU is uploaded to the cloud.

4.1 Hardware Implementation

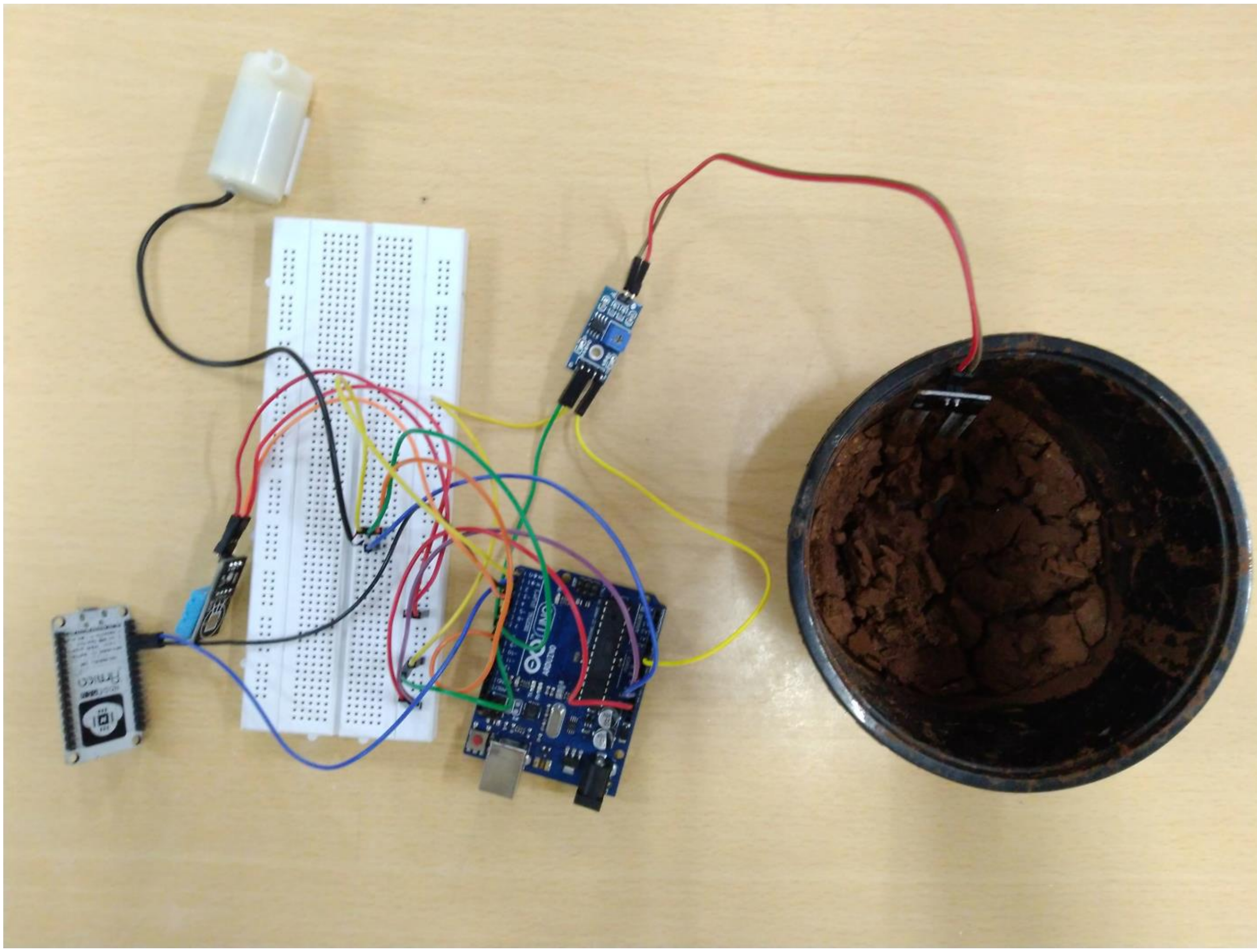


Fig. 13 Complete Circuit

4.2 User Interface

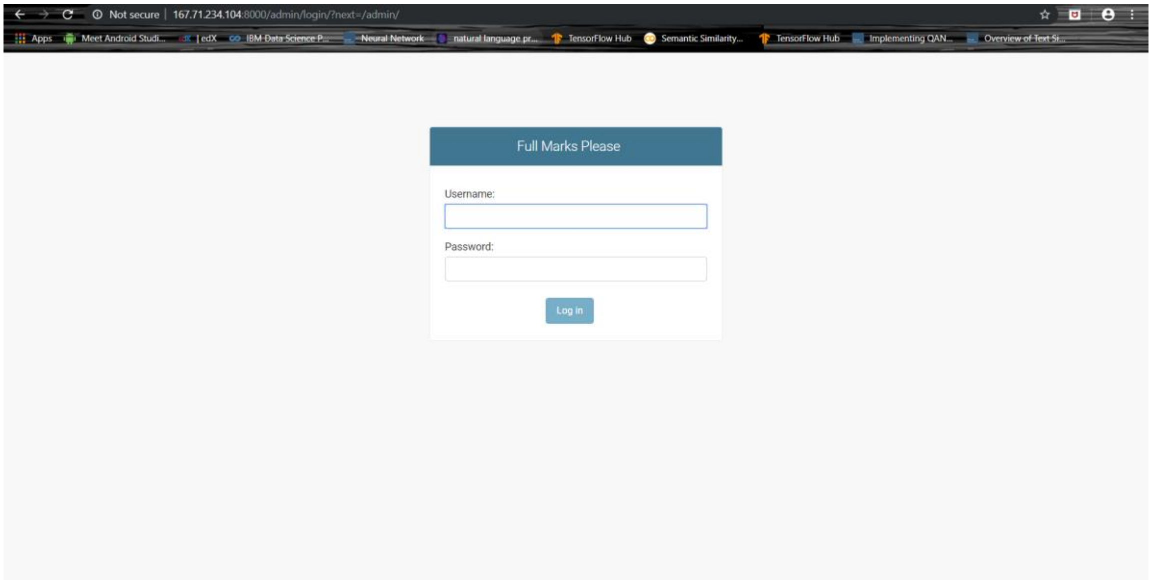


Fig. 14 Login Page on Desktop

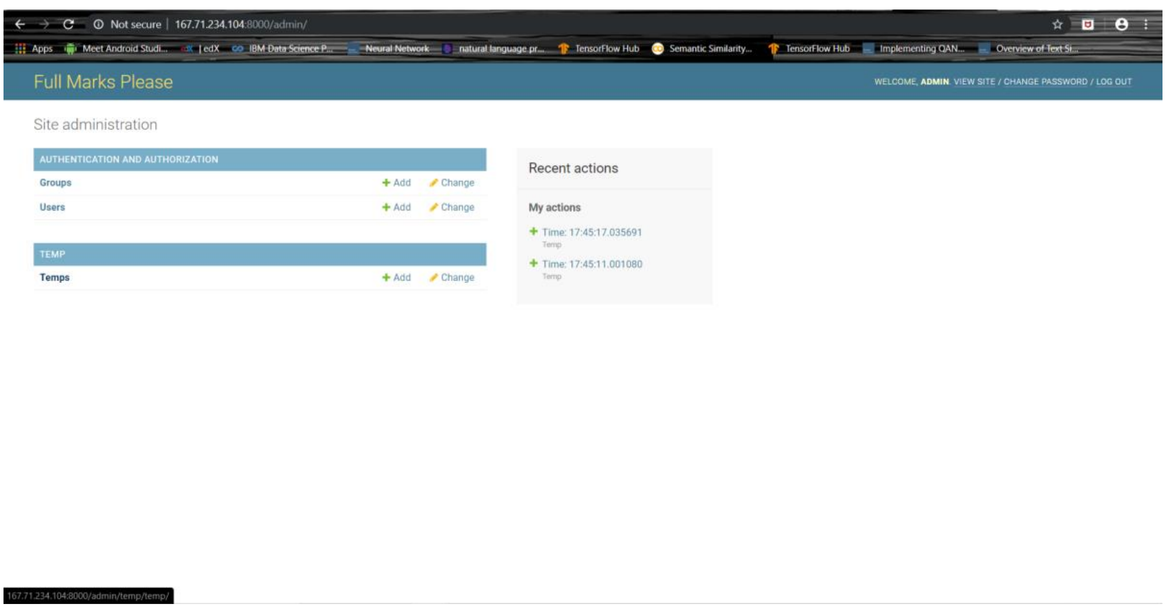


Fig. 15 Home Page on Desktop

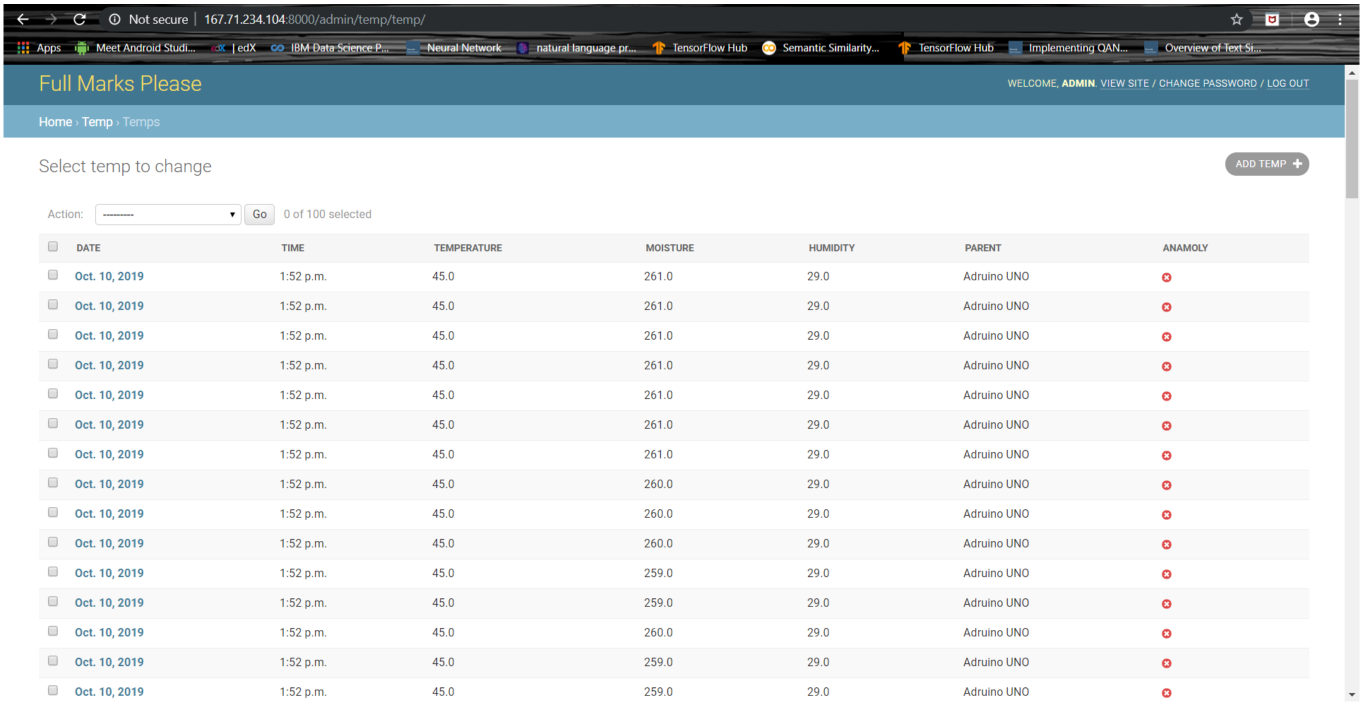


Fig. 16 Database View Page on Desktop

Accessing the Website from Phone:

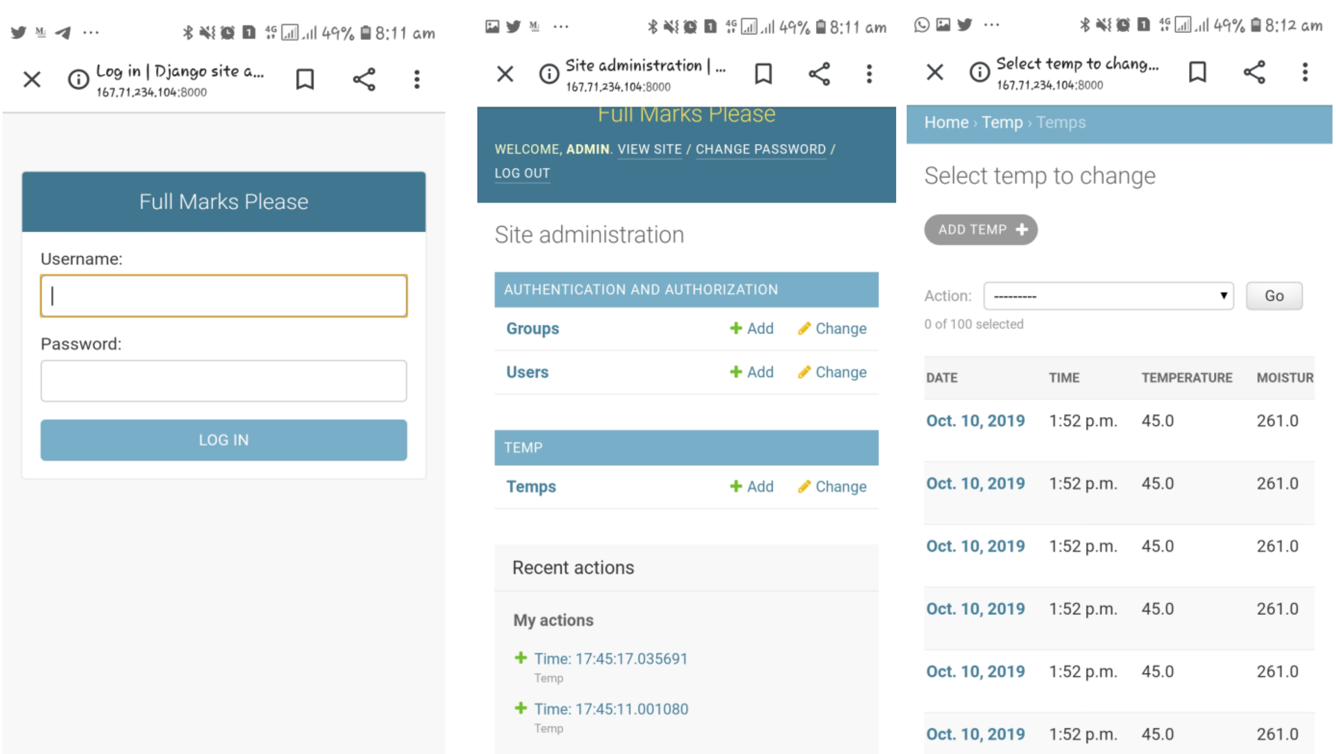
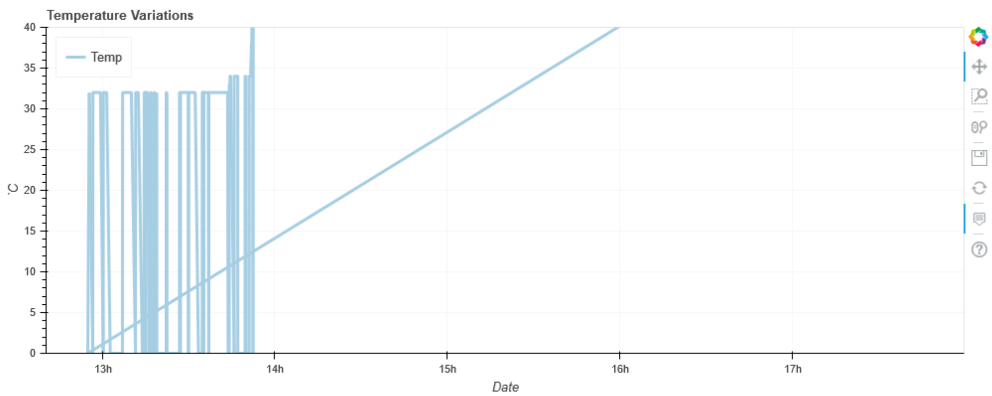


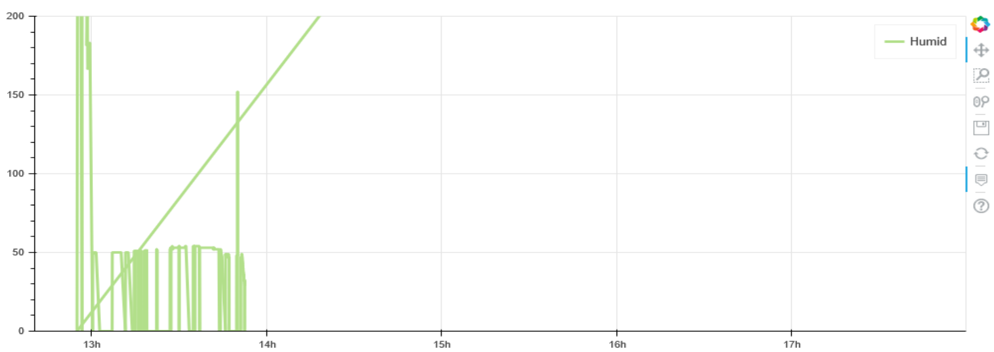
Fig. 17 Various Views from Phone

5. Analysis

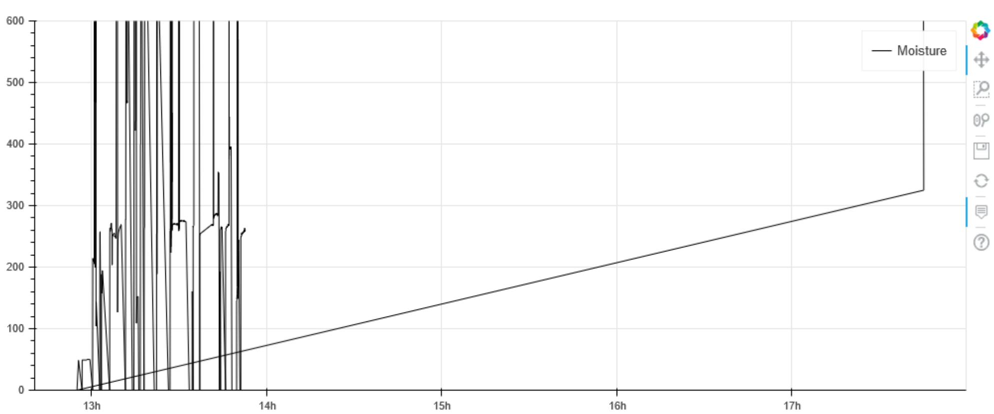
Temperature Analysis



Humidity Analysis



Soil Moisture Analysis



6. Challenges

- Getting the perfect WiFi module
- Debugging the code without errors being displayed
- Assembling the different hardware’s into a single circuit
- Sending JSON file from Arduino to NodeMCU
- Keeping all the components in the bread board, once the assembling was done
- Making the backend of the Application

7. Scalability and Future Scope

- The project is scalable to n number of sensors connected
- The backend is also scalable
- In future, this model will have to use various security features and network protocol, if more than one arduinos are connected to gateway nodes.
- Analysis of the data can be done using the data of various nodes, and water usage can be estimated
- Based on the temperature and humidity conditions also, which crops can be better suited, can be estimated

8. References

- https://www.instructables.com/id/Connecting-Arduino-WiFi-to-the-Cloud-Using-ESP8266/
- https://mybtechprojects.tech/serial-communication-between-nodemcu-and-arduino/
- https://www.google.com/amp/s/techtutorialsx.com/2017/01/08/esp8266-posting-json-data-to-a-flask-server-on-the-cloud/amp/
- https://forum.arduino.cc/
- https://www.arduino.cc/
- https://docs.djangoproject.com/en/2.2/