**SMART IRRIGATION SYSTEM**



**Project Based Synopsis**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.no.** | **Name** | **UID** | **Group** | **Branch** |
| **1** | **Mayank Joshi** | **20BCS3971** | **1** | **CSE Big Data** |

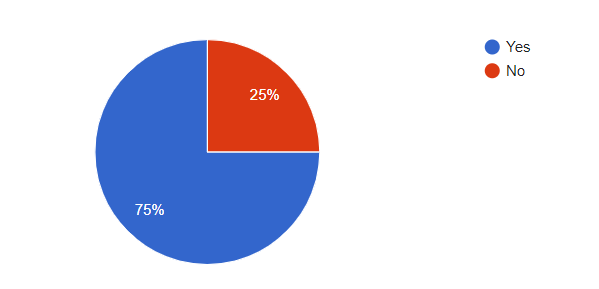
**Project Problem Statement**

**In India, agriculture in villages plays an essential role in developing the country. Basically, agriculture depends on the monsoons which have not enough water source. Many Farmers are facing problem in growing crops due to unpredictable weather conditions. Also when they have water they supply uncalculated water to the field which leads to over water . So we have to find a solution which would predict the weather conditions and supply right amount of water to the fields.**

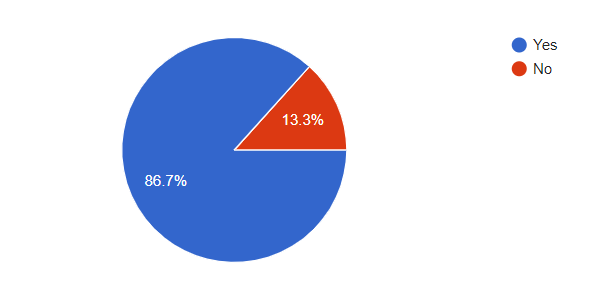
**Survey**

**We got a report in survey on smart irrigation system**

**Are you a Farmer**



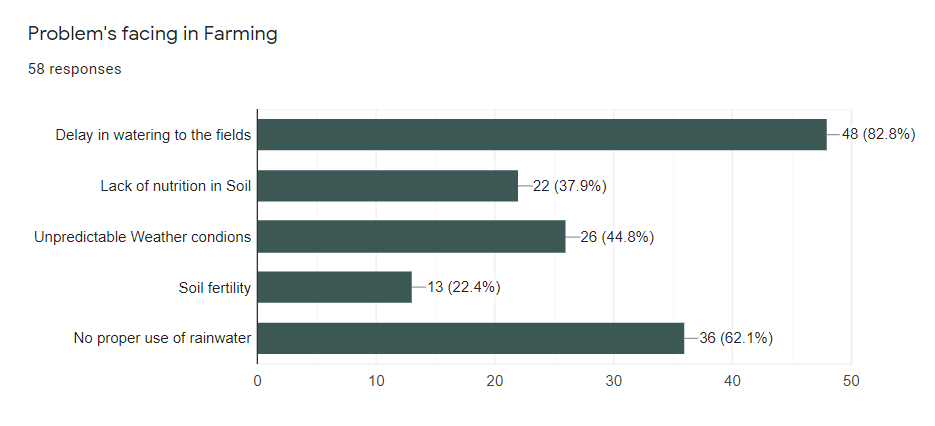
**Is Smart Irrigation System a good solution**



**Household or landlord**

****

**Problems facing in farming**



**Solution**

**To solve this problem we are going to design a Smart Irrigation System that will be completely IOT based which will capable of analyzing the soil moisture, climate condition as well as wind speed. We will be able to predict weather conditions. The data of sensors will be displayed on Blot cloud page. It will command the Arduino through a webpage to control the motor and the rest of whole irrigation process will be automatic.**

**Key features/Benefits**

* **Cheap and flexibility**
* **Cloud based system**
* **Water saving**
* **Time saving**
* **Arduino/328p microcontroller is used to control the motor that supplies water.**
* **It will automatically stop if it is raining.**

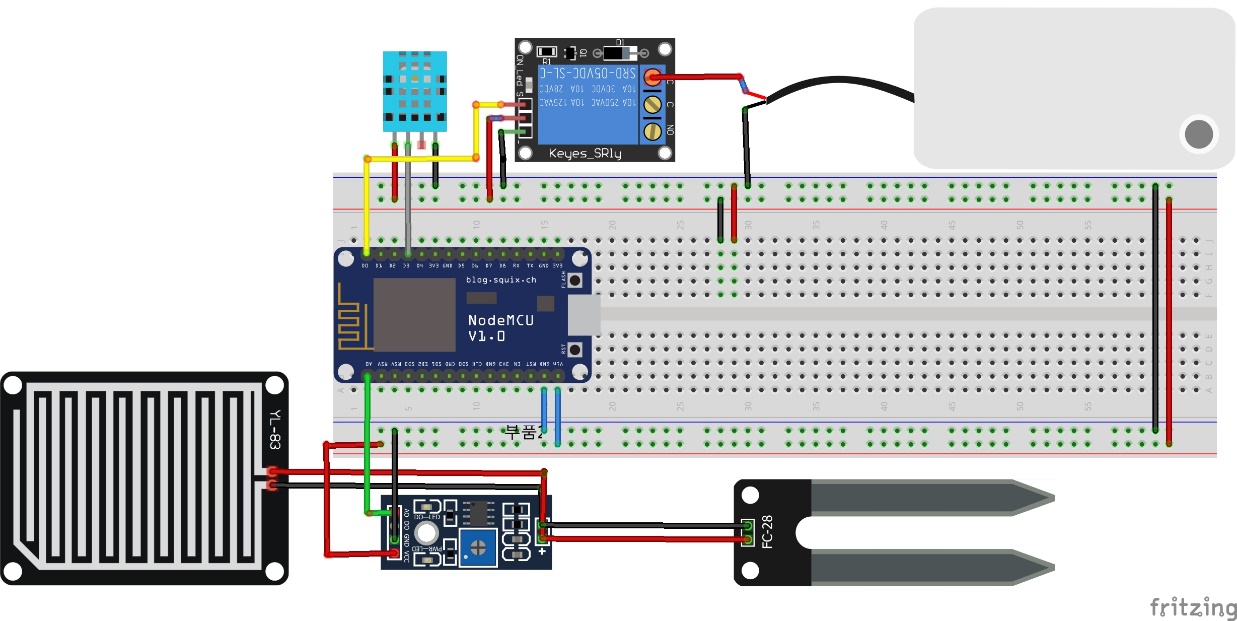
**List of software used**

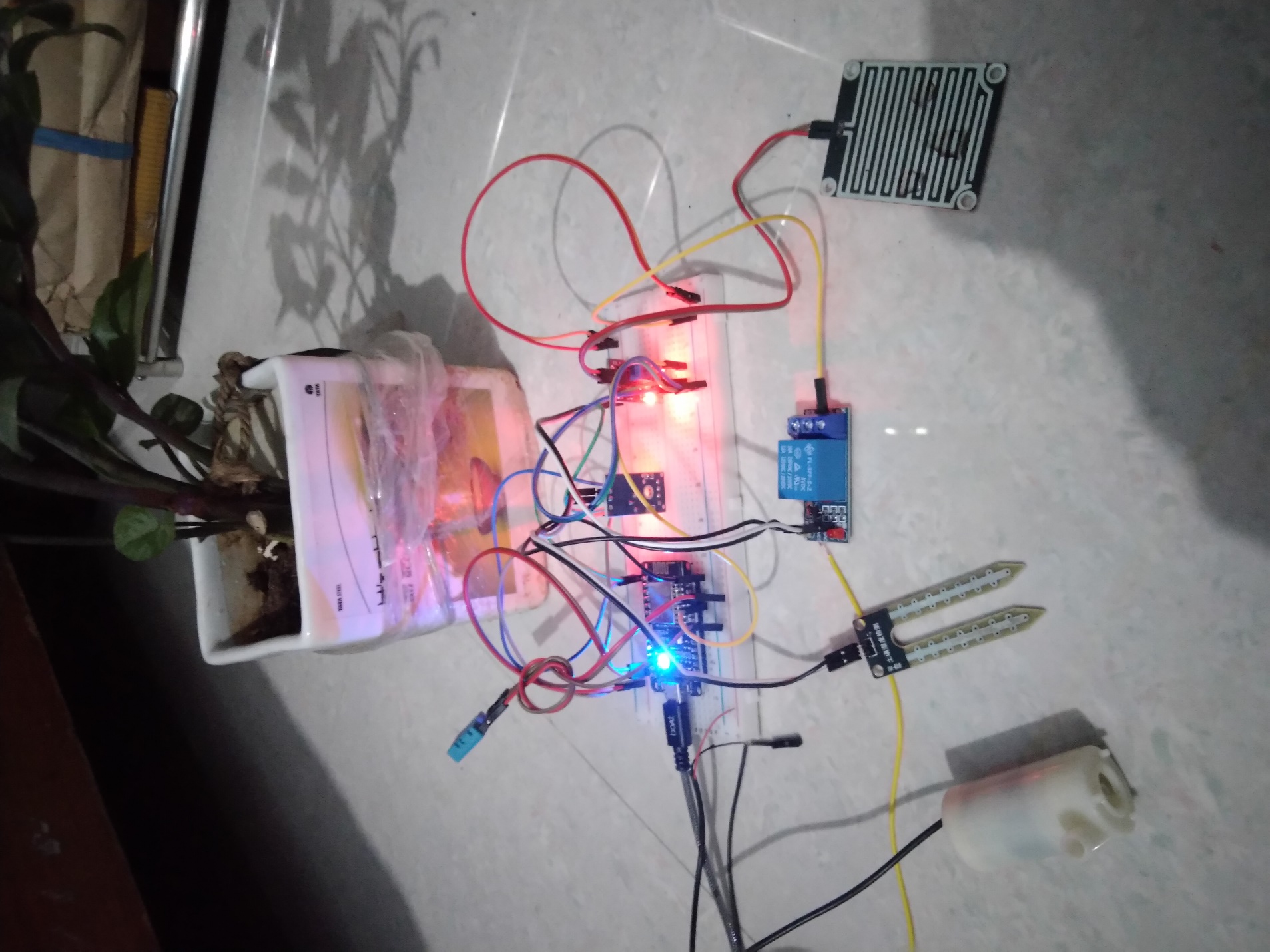
* **Arduino**
* **Fritzing**
* **Thingspeak**
* **IFTTT**

**List of Hardware used**

* **Bread bord**
* **ESP8266 NODE MCU**
* **BMP 280 pressure and temperature sensor**
* **Rain Drop Sensor Module**
* **5V 10A Relay Module**
* **3-12V Mini DC Submersible Pump**
* **DHT11 Temperature and Humidity sensor**
* **Soil Module sensor Module**

**Circuit**

****

****

**Code:**

#include <ESP8266WiFi.h>

#include <Wire.h>

String apiKey = "9GXWHK5YOVCOMBUN"; // Enter your Write API key here

const char\* server = "api.thingspeak.com";

const char \*ssid = "moto g(9) 8875"; // Enter your WiFi Name

const char \*pass = "maya2000"; // Enter your WiFi Password

#define DHTPIN D3 // GPIO Pin where the dht11 is connected

DHT dht(DHTPIN, DHT11);

WiFiClient client;

const int moisturePin = A0; // moisture sensor pin

const int motorPin = D0;

unsigned long interval = 10000;

unsigned long previousMillis = 0;

unsigned long interval1 = 1000;

unsigned long previousMillis1 = 0;

float moisturePercentage; //moisture reading

float h; // humidity reading

float t; //temperature reading

void setup()

{

Serial.begin(115200);

delay(200);

pinMode(motorPin, OUTPUT);

digitalWrite(motorPin, LOW); // keep motor off initally

dht.begin();

Serial.println("Connecting to ");

Serial.println(ssid);

WiFi.begin(ssid, pass);

while (WiFi.status() != WL\_CONNECTED)

{

delay(500);

Serial.print("."); // print ... till not connected

}

Serial.println("");

Serial.println("WiFi connected");

}

void loop()

{

unsigned long currentMillis = millis(); // grab current time

h = dht.readHumidity(); // read humiduty

t = dht.readTemperature(); // read temperature

if (isnan(h) || isnan(t))

{

Serial.println("Failed to read from DHT sensor!");

return;

}

moisturePercentage = ( 100.00 - ( (analogRead(moisturePin) / 1023.00) \* 100.00 ) );

if ((unsigned long)(currentMillis - previousMillis1) >= interval1) {

Serial.print("Soil Moisture is = ");

Serial.print(moisturePercentage);

Serial.println("%");

previousMillis1 = millis();

}

if (moisturePercentage < 50) {

digitalWrite(motorPin, HIGH); // tun on motor

}

if (moisturePercentage > 50 && moisturePercentage < 55) {

digitalWrite(motorPin, HIGH); //turn on motor pump

}

if (moisturePercentage > 56) {

digitalWrite(motorPin, LOW); // turn off motor

}

if ((unsigned long)(currentMillis - previousMillis) >= interval) {

sendThingspeak(); //send data to thing speak

previousMillis = millis();

client.stop();

}

}

void sendThingspeak() {

if (client.connect(server, 80))

{

String postStr = apiKey; // add api key in the postStr string

postStr += "&field1=";

postStr += String(moisturePercentage); // add mositure reading

postStr += "&field2=";

postStr += String(t); // add temperature reading

postStr += "&field3=";

postStr += String(h); // add humidity reading

postStr += "\r\n\r\n";

client.print("POST /update HTTP/1.1\n");

client.print("Host: api.thingspeak.com\n");

client.print("Connection: close\n");

client.print("X-THINGSPEAKAPIKEY: " + apiKey + "\n");

client.print("Content-Type: application/x-www-form-urlencoded\n");

client.print("Content-Length: ");

client.print(postStr.length()); //send length of the string

client.print("\n\n");

client.print(postStr); // send complete string

Serial.print("Moisture Percentage: ");

Serial.print(moisturePercentage);

Serial.print("%. Temperature: ");

Serial.print(t);

Serial.print(" C, Humidity: ");

Serial.print(h);\*/

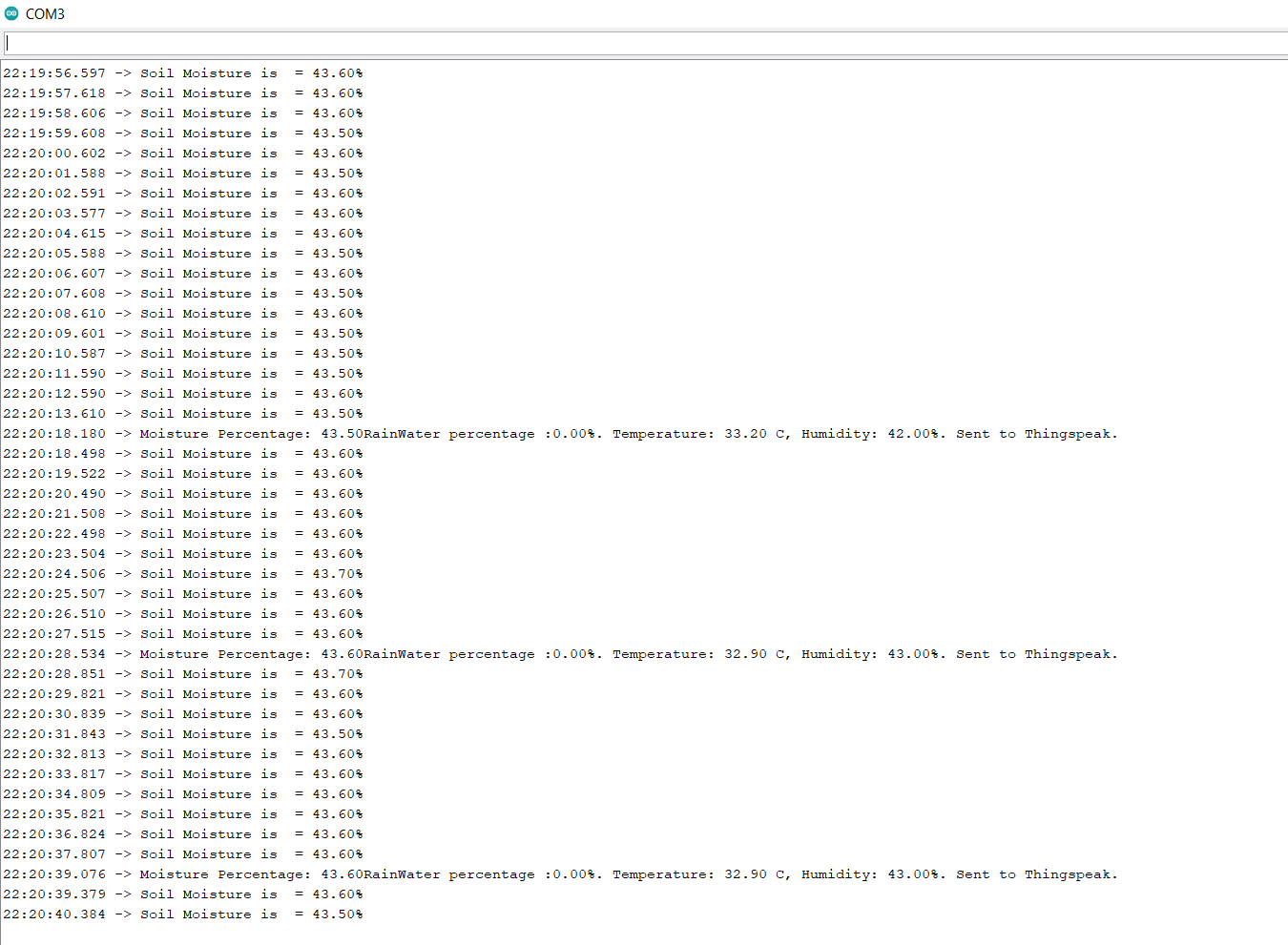
Serial.println("%. Sent to Thingspeak.");

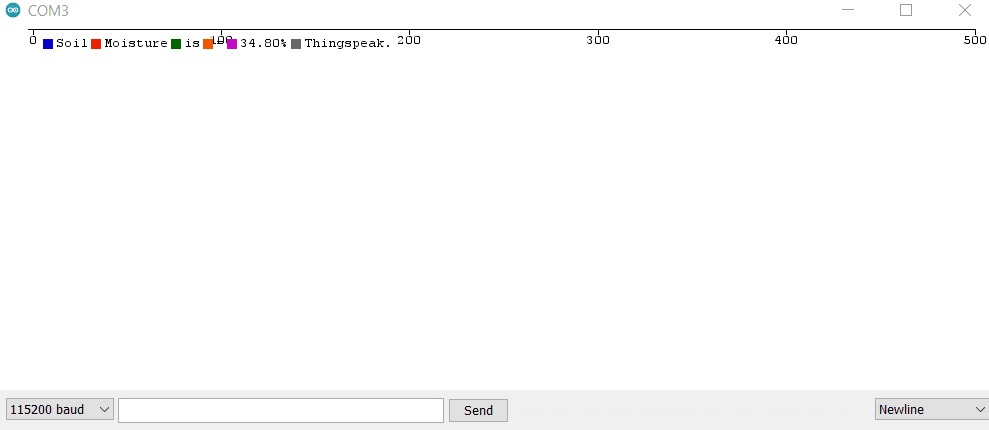
}

}

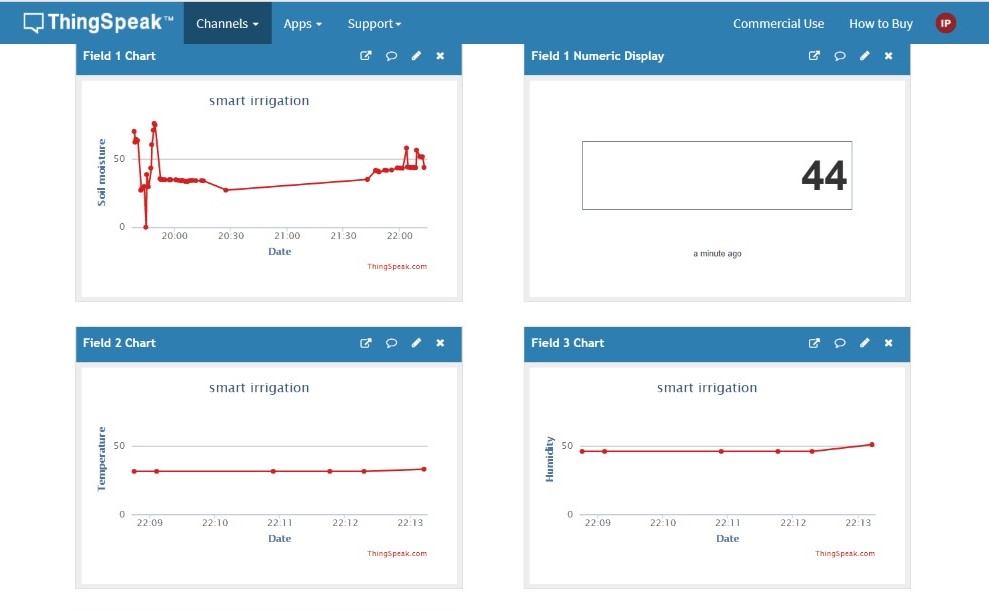
**Outcomes:**

**The data collected from serial monitor(Arduino)**

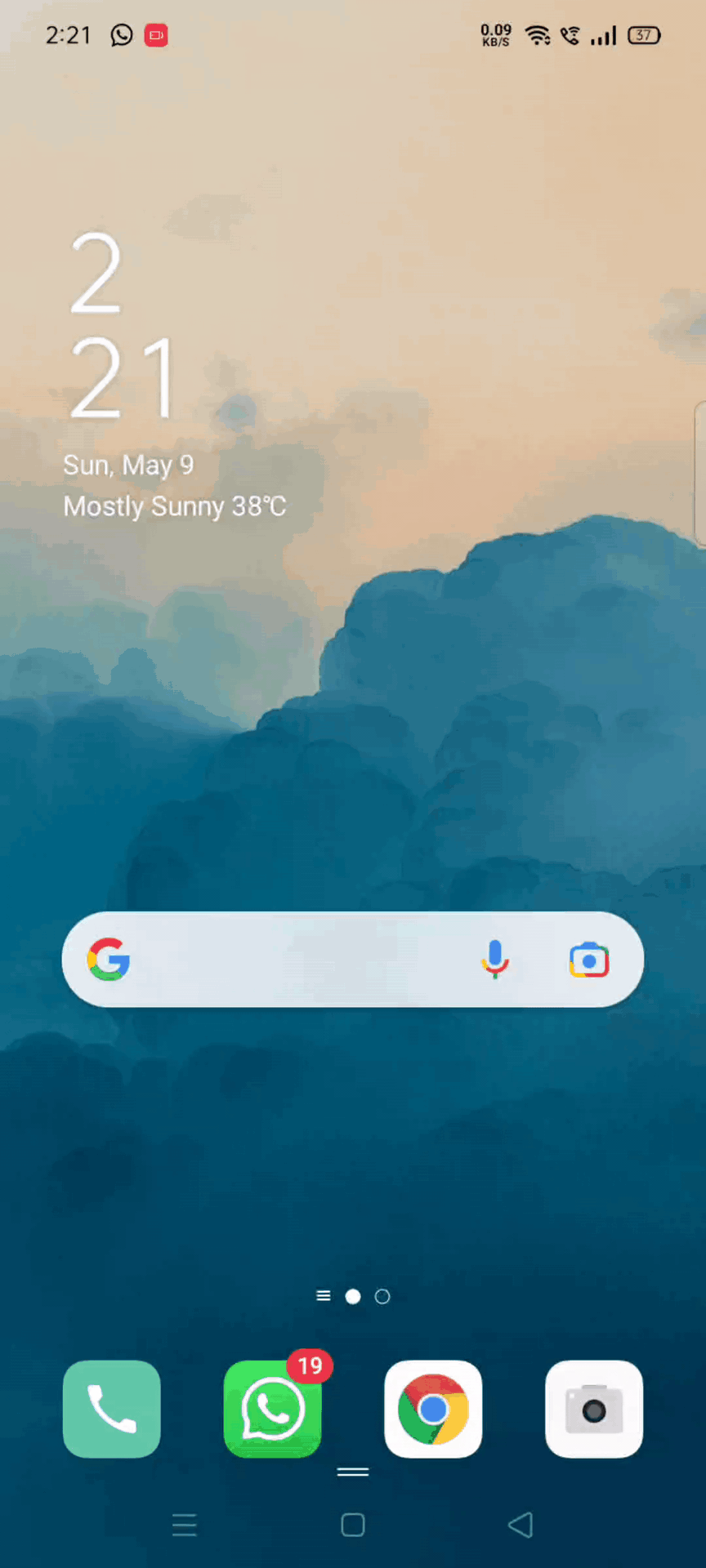
****



**The data collected from ThingSpeak**



**Notification from IFTTT:**



**Deliverables**

* **ppt**
* **document**
* **video**
* **circuit**
* **survey**

**Future Deliverables:**

* **To get the accurate values of soil moisture, Humidity and Temperature with approximate precision.**
* **This project is dealing with the soil moisture, so there will be specific and suitable soil moisture values for each crop type for better yielding. By this project we will get to know the soil moisture and thus we can suggest farmers which is suitable crop for their soil.**
* **Also right amount of water according to the need of different crops would be provided.**