



REVIEW-3

Intelligent Garden using IoT



TEAM MEMBERS

19BCE1704 - SHOURYA GUPTA

19BCE1697 - RIDHIKA SAHNI



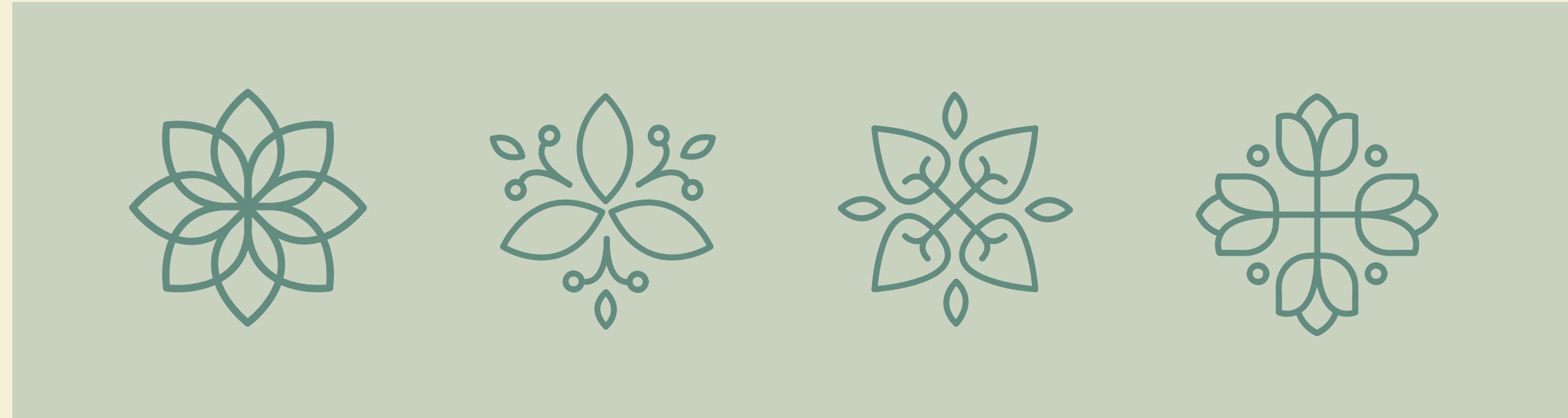
WHY THIS PROJECT?

Most people around the world are getting more and more interested in living a long and healthy life nowadays. And many studies have proven that growing house plants, as long as being a trend, improves health. The truth is, we are so busy with work, and even when we get home, that we do not have enough time to properly water our plants when they need it.

WHAT INSPIRED US?



Problem Statement



We are making an IoT Smart Plant Monitoring System. It will display soil moisture, temperature, humidity, and will also give us an option to control the water flow on our smartphone - which can be operated from anywhere in the world.

For future planning, we can make an alert system, so if the soil has less moisture it will pop up an alert message on our smartphone to water our plants.

Components Required



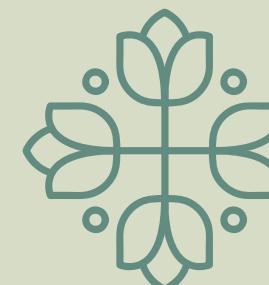
Microcontroller - Node MCU



Temperature Sensor - DHT11

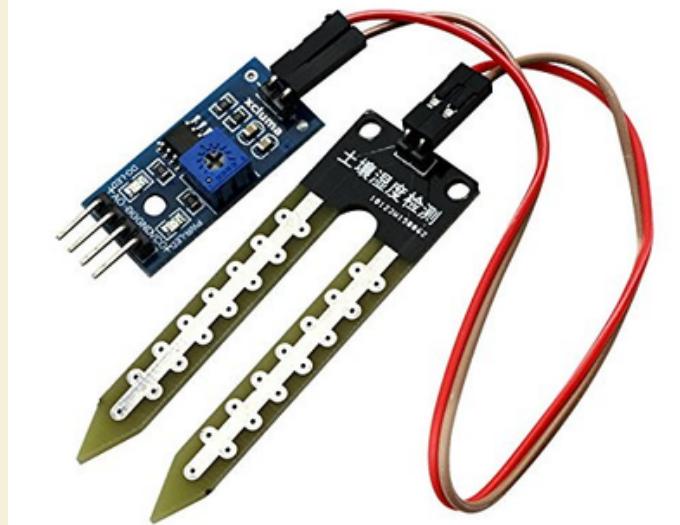
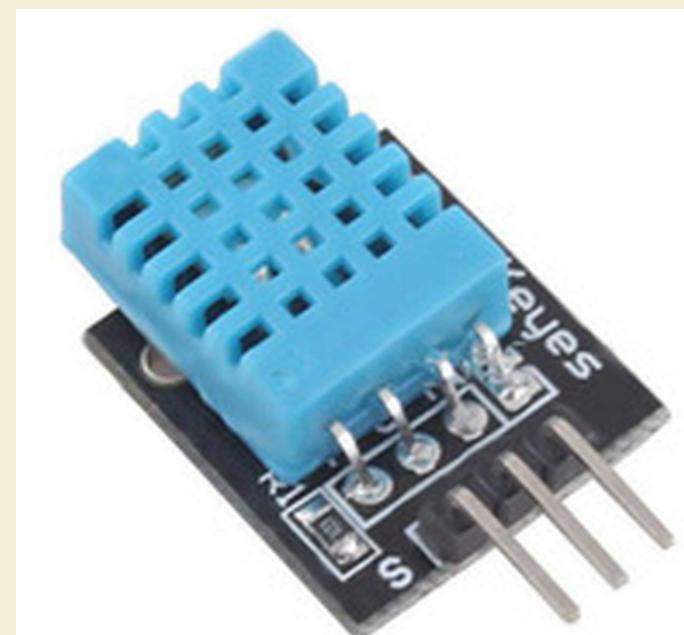
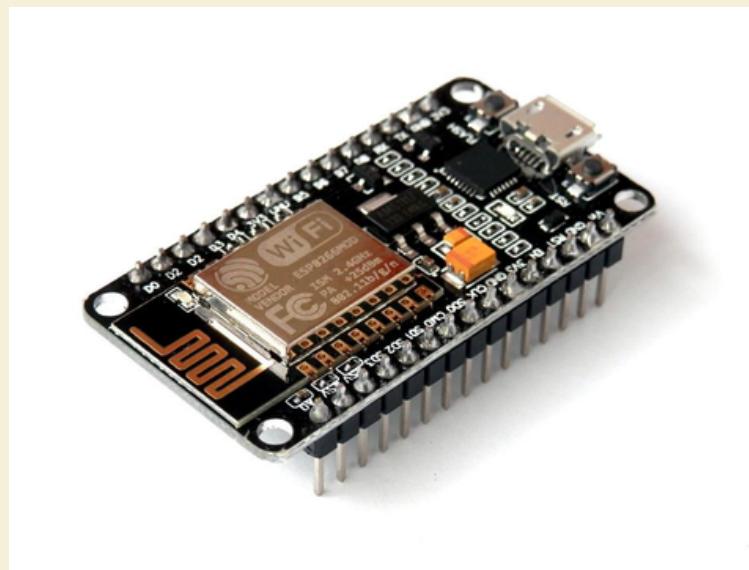


Soil Moisture Sensor - YL69



Water Pump, Relay, Wires,
Battery, etc

Components bought !!



Software Component

Arduino IDE - to run our code

Blynk IoT for Arduino, ESP8266/32

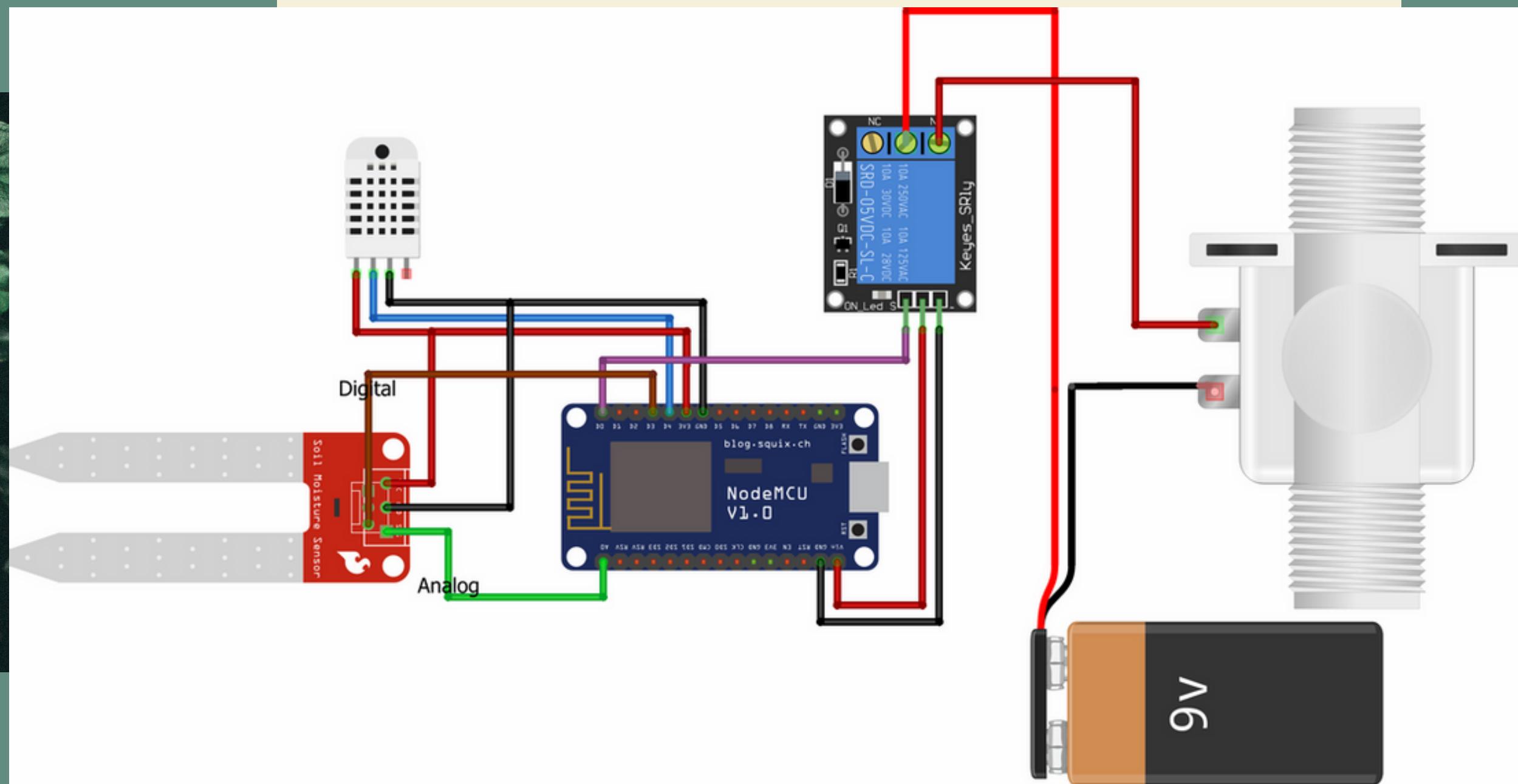


BRING NATURE
INTO YOUR HOME

System Architecture



We have referred several youtube videos, and GitHub projects - by certain analysis done. We have made circuit like the below one.





IMPLEMENTATION

We can install our IoT device in our garden, so that everytime we are notified about the less moisture in our soil, we can simply click on the button "ON" to supply water, from any place in the world. For user interface, we will use an application (Blynk) to make it more user-friendly and easy to use.

CODE

iot_try §

```
#define BLYNK_PRINT Serial
#include <OneWire.h>
#include <SPI.h>
#include <BlynkSimpleEsp8266.h>
#include <DHT.h>
#include <DallasTemperature.h>
#define ONE_WIRE_BUS D2

OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);

//authorization key taken from the blynk app
char auth[] = "yeQ2H9_LKVLAGVSQzk3aLA6J75L25uKd";

//my wifi name
char ssid[] = "Airtel_9810156712";

//my wifi pwd
char pass[] = "air22503";

#define DHTPIN 2
#define DHTTYPE DHT11
```

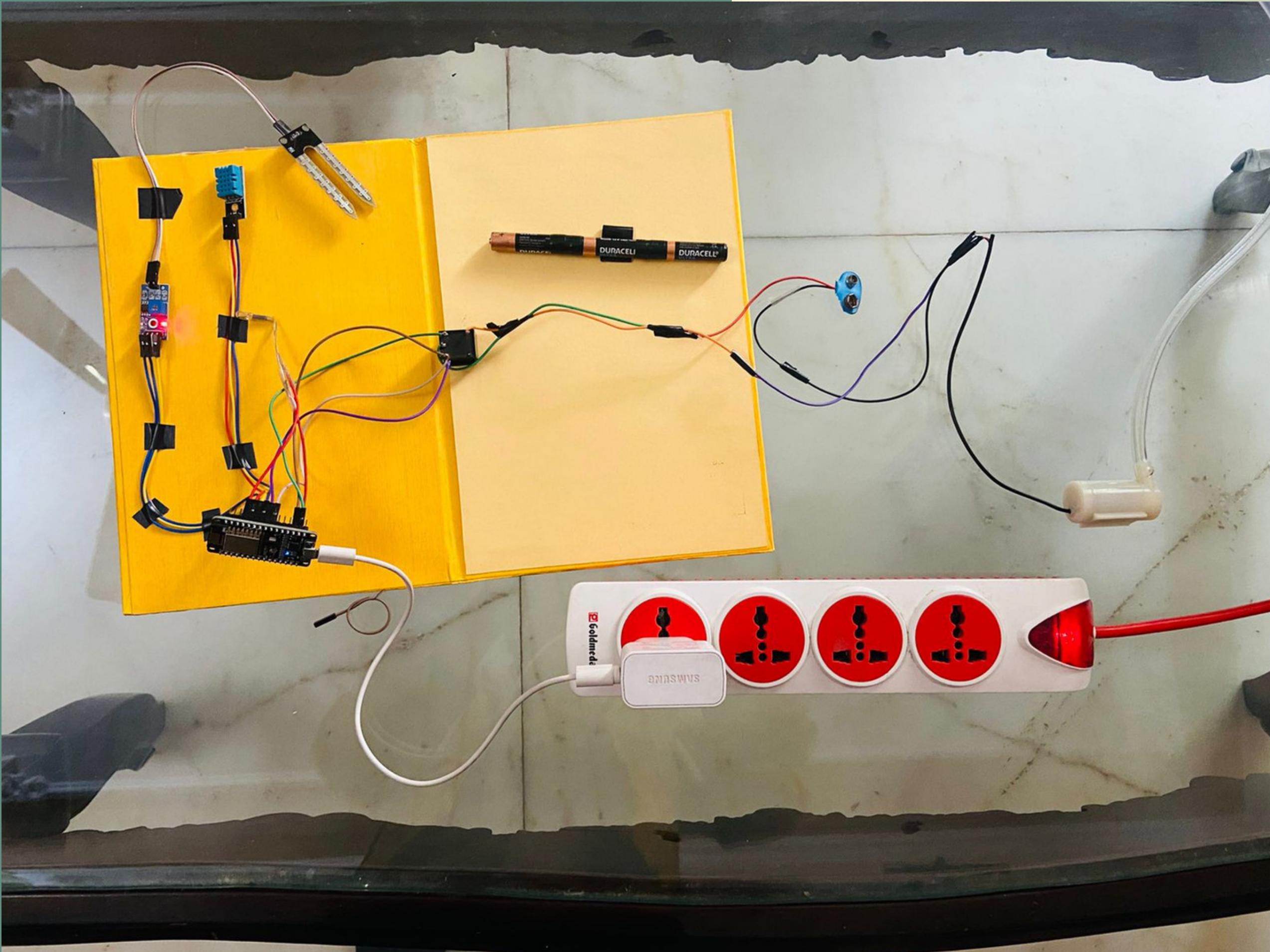
CODE

```
iot_try $  
  
DHT dht(DHTPIN, DHTTYPE);  
SimpleTimer timer;  
void sendSensor()  
{  
    float h = dht.readHumidity(); //reads humidity  
    float t = dht.readTemperature(); //reads temp  
  
    if (isnan(h) || isnan(t)) { //not null  
        Serial.println("Failed to read from DHT sensor!");  
        return;  
    }  
  
    Blynk.virtualWrite(V5, h); //V5 (virtual pin) is for Humidity  
    Blynk.virtualWrite(V6, t); //V6 (virtual pin) is for Temperature  
}  
  
void setup()  
{  
    Serial.begin(9600); //baud rate  
    dht.begin();  
  
    timer.setInterval(1000L, sendSensor); //each second it will check the established connection  
    Blynk.begin(auth, ssid, pass); //connected Blynk application  
    sensors.begin();  
}
```

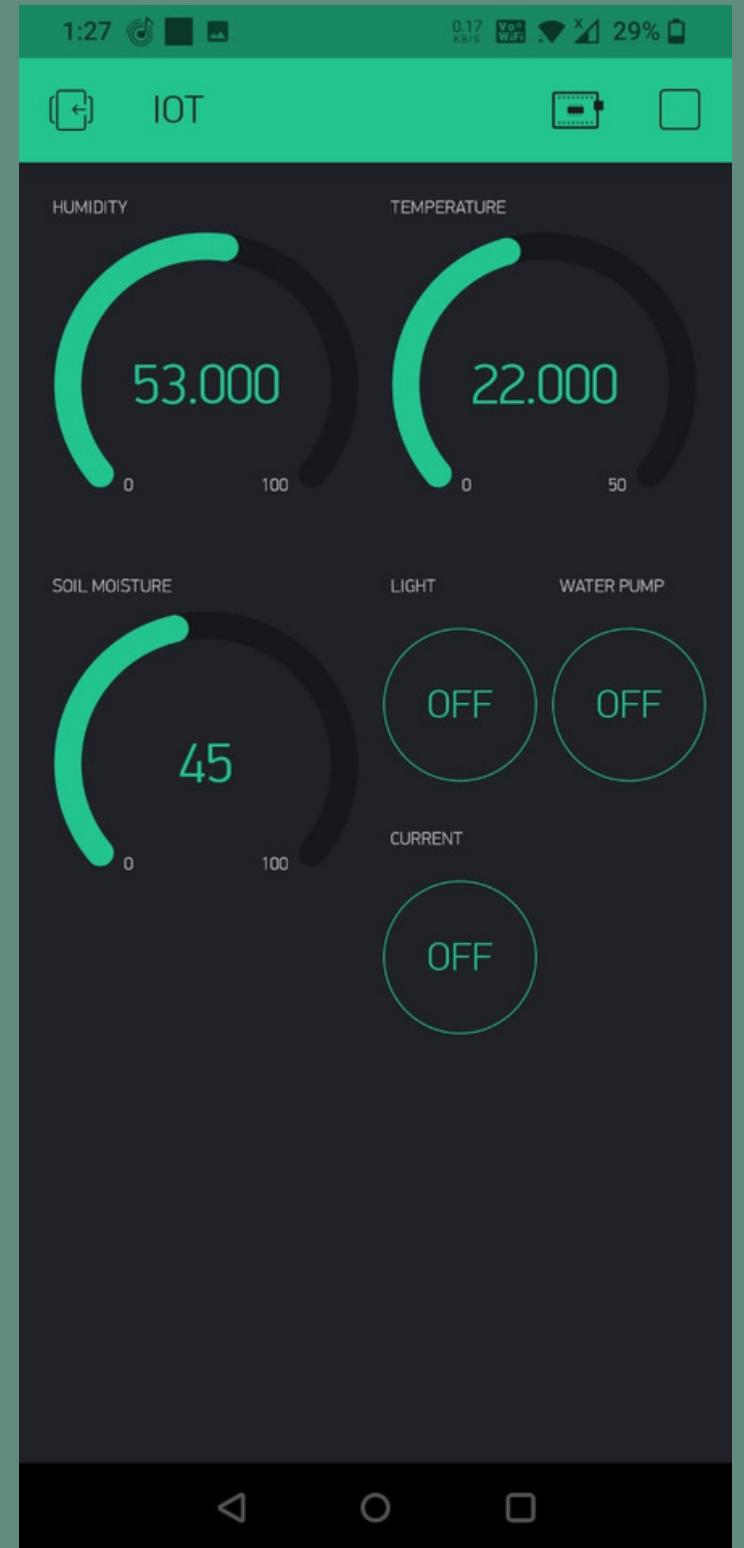
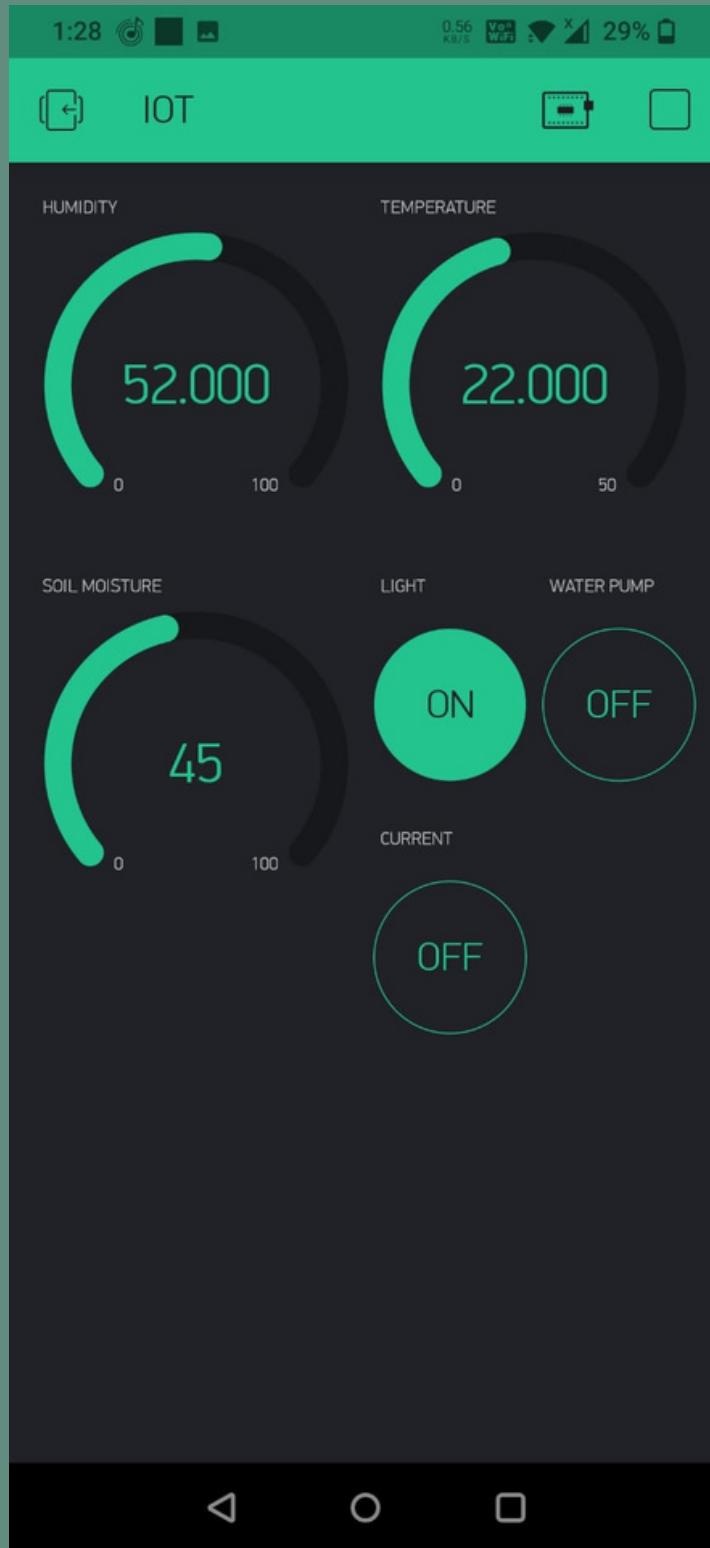
CODE

```
iot_try §  
int sensor=0;  
int output=0;  
  
void sendTemps()  
{  
    sensor=analogRead(A0);  
    output=(145-map(sensor,0,1023,0,100)); //in place 145 there is 100(it change with the change in sensor)  
    delay(1000);  
    sensors.requestTemperatures();  
    float temp = sensors.getTempCByIndex(0);  
  
    //prints on serial monitor  
    Serial.println(temp);  
    Serial.print("moisture = ");  
    Serial.print(output);  
    Serial.println("%");  
    Blynk.virtualWrite(V1, temp);  
    Blynk.virtualWrite(V2, output);  
    delay(1000);  
}  
  
void loop()  
{  
    Blynk.run();  
    timer.run();  
    sendTemps();  
}
```

CIRCUIT



OUTPUT



Future implementation

We can add threshold value to soil moisture so that it will turn on and off automatically instead of turning it on manually. This is make our project more efficient.



