



THE REPORT

IoT BASED IRRIGATION SYSTEM

M DILSHAN RODRIGO
[102867540]

1 Introduction

1.1 Topic and Background

This report presents an IoT-based smart system to manage irrigation requirements of a field using the sensing of ground parameters like soil moisture, air humidity, and temperature. Irrigation requirements control through a water pump with help of a relay module. And also include a water level sensing mechanism to notify the user about the water tank. The sensor data store in a database and also user able to access that information through a web-based interface.

I also did the first assignment related to this topic, so I decided to build an Automatic Irrigation System using Arduino Uno, NodeMCU, and other sensor modules that I previously mentioned above.

1.2 Proposed System

Main Component,

1. Arduino Uno (Collect sensor data, Send data to NodeMCU, Control water pump)
2. NodeMCU (Send and Get data from the database)
3. Soil Moisture Sensor
4. DHT11 Sensor (Humidity and Temperature)
5. Water Level Sensor
6. Relay Module and 12V Water Pump
7. LCD 16x2 display with I2C interface

1) Collect Sensor Data

This system gets threshold values from the database through NodeMCU using serial communication. After that process system begins to collect data from the sensor modules. The system collects data of four parameters using three sensor modules. That includes,

1. Water Level sensor module,
2. Soil Moisture sensor,
3. DHT11 Humidity and Temperature sensor.

The water level of the water tank is measured by the water level sensor module and it connects with the Arduino. Three led indicators are used to notify the user about the water level of the water tank.

1. Red LED – shows water level is lower and need to refill the water tank
2. Yellow LED – shows water level is medium
3. Green LED – shows water level is higher

2) Send Data to NodeMCU and Database

Sensor readings collect through the Arduino Uno board and send those readings to NodeMCU using serial communication. After receiving the sensor readings, NodeMCU sends sensor readings to the database. The system sends sensor readings to the database every specific time.

3) Control Mechanism of the Water Pump

According to the threshold values and the sensor readings, Arduino Uno control (on/off) water pump (12 V) using the help of the relay module. The water pump starts only if the required water level is available in the water tank. LCD 16x2 with I2C interface connects with Arduino Uno which displays current Soil Moisture level and the status of the water pump (is it ON or OFF).

```
void Water_Pump_Control()
{
    // check soil moisture level is less than threshold soil moisture level (soil dry)
    if(soilmoisture < tsoilmoisture)
    {
        // check temperature is greater than threshold temperature (air dry)
        if(temperature >= ttemperature)
        {
            // check humidity is greater than threshold humidity (air is more dry)
            if(humidity >= thumidity)
            {
                Pump_Process(10000);
            }
            // (air is dry)
            else
            {
                Pump_Process(5000);
            }
        }
        else // in the night or rainy day
        {
            if(humidity >= thumidity)
            {
                Pump_Process(3000);
            }
        }
    }
}
```

4) Process of the Red Button

The red color button is used in this system, when the button is pressed once, immediately collect sensor readings to control the water pump (on/off) according to the irrigation needs and send sensor reading data to the NodeMCU.

Otherwise, every 5 minutes Arduino Uno collects sensor readings from the sensor to continue the irrigation process and every 10 minutes sends the collected sensor reading data to the database using NodeMCU.

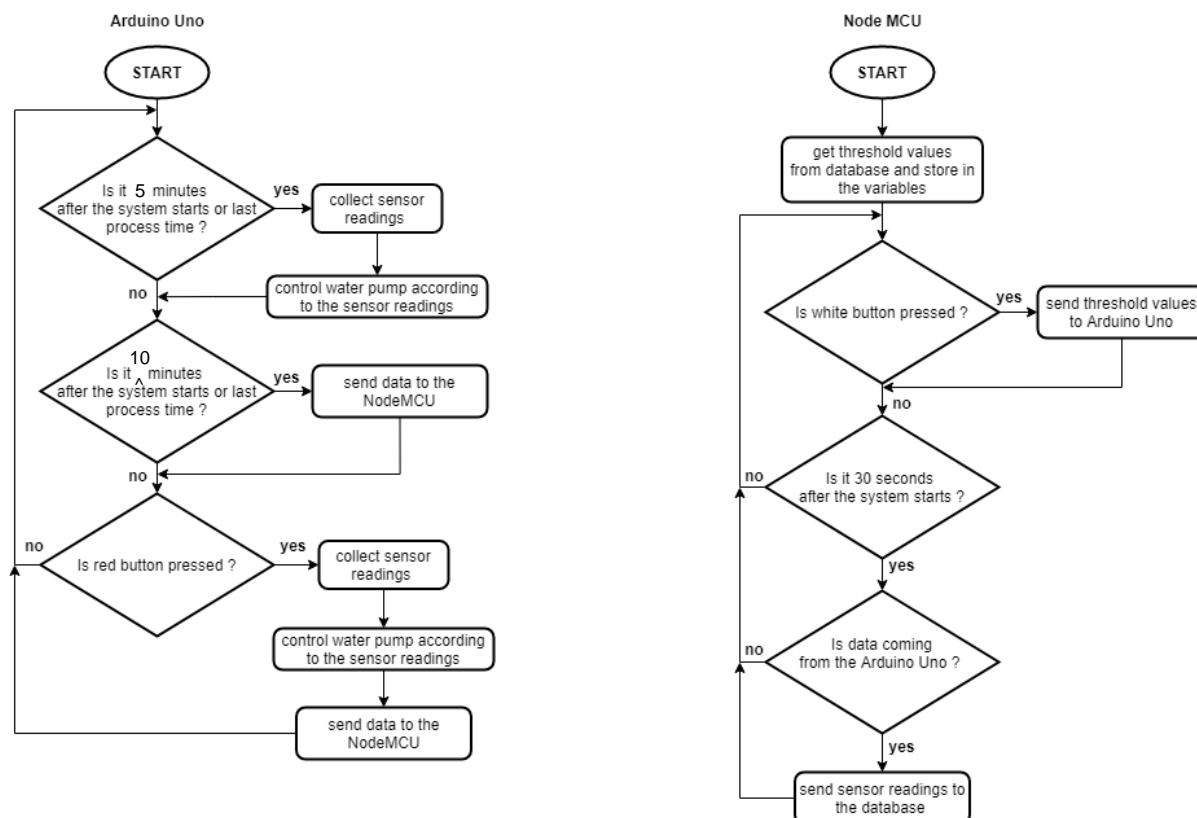
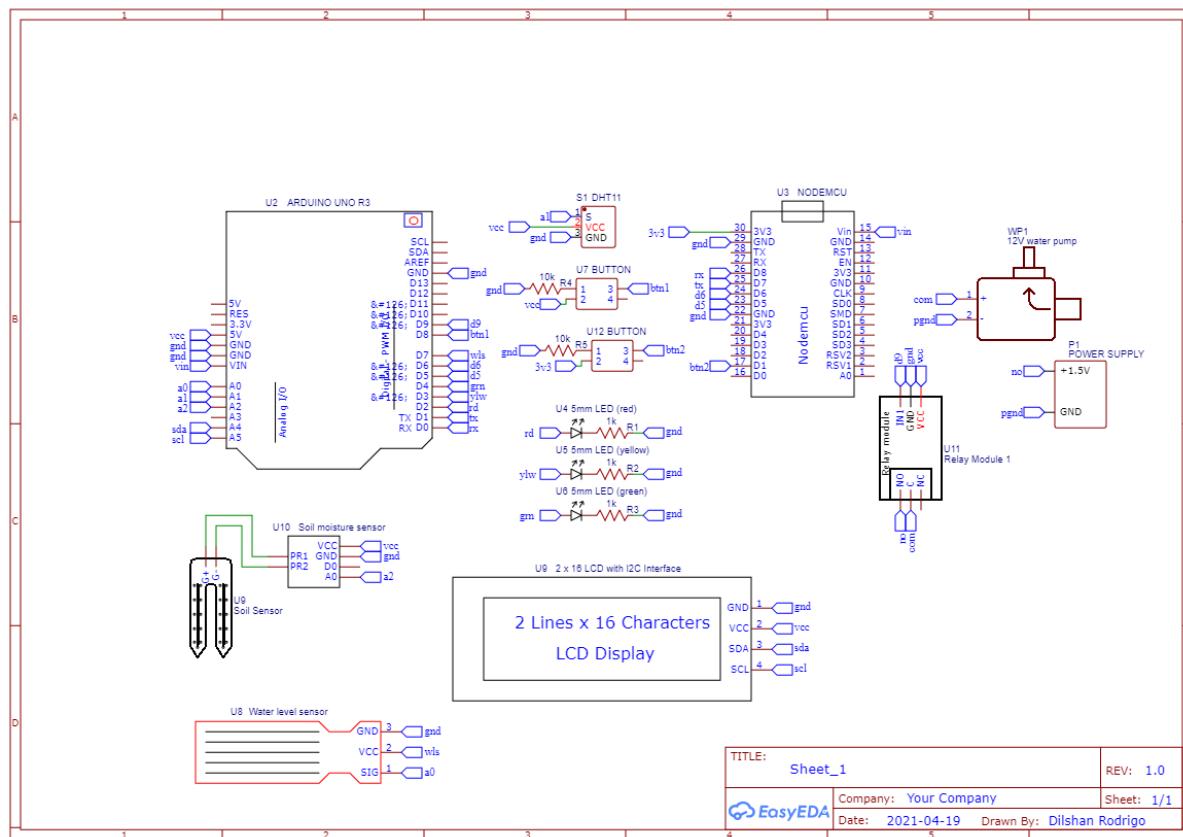
5) Process of the White Button

The white color button is used to enable the communication between NodeMCU and Arduino Uno. Simply, when the button is pressed, NodeMCU sends previously collected (from database) threshold values to Arduino Uno using serial communication.

6) User Interface and Data Visualization

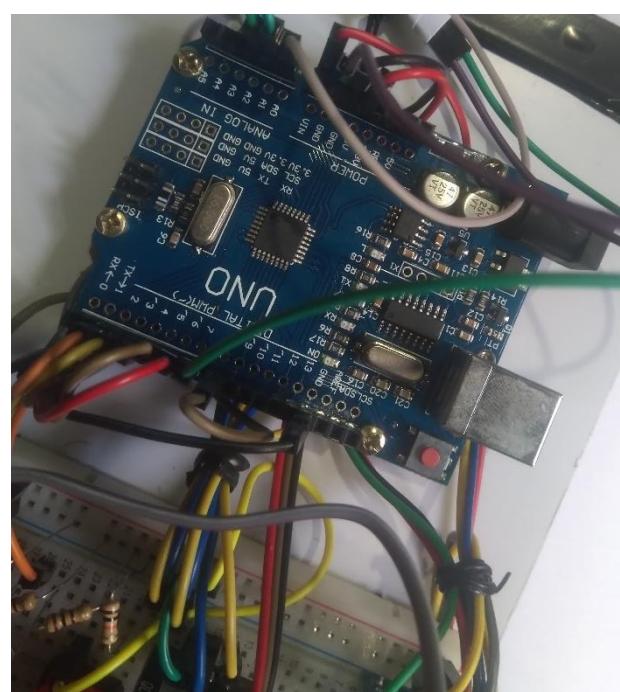
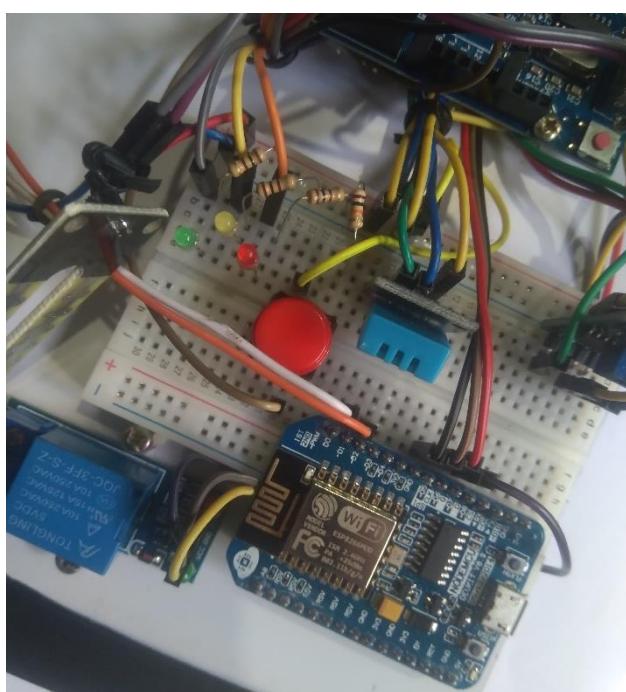
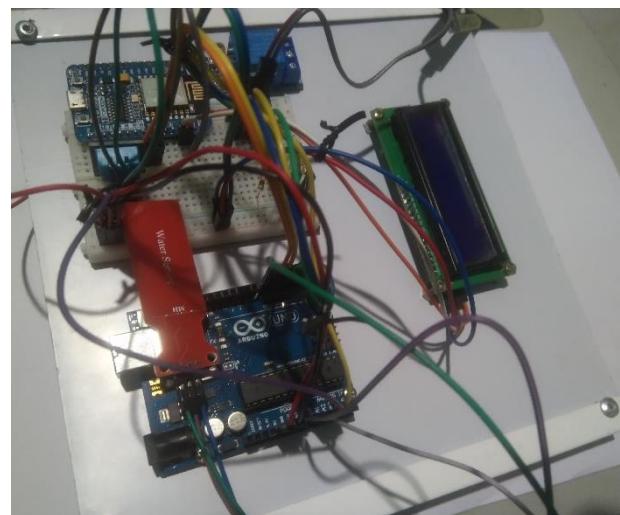
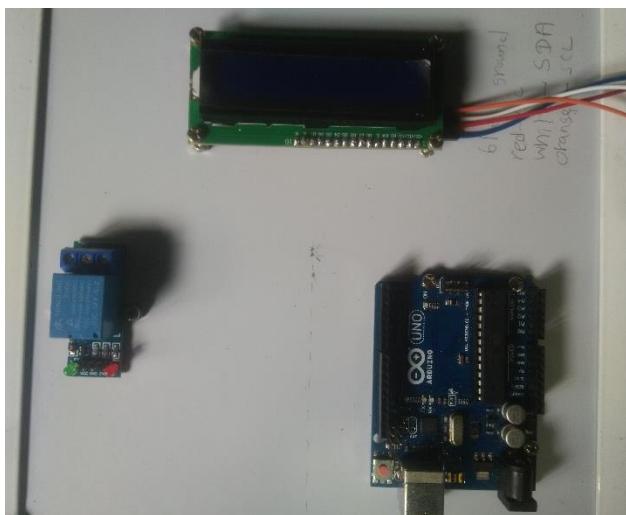
Web-based interface used to monitor the data of the Irrigation system that shows current humidity value, current temperature value, current Soil Moisture value, and the Water level of the water tank. And also that interface able to show the last twenty readings of Soil Moisture values, Humidity values, and Temperature values. The last forty readings of sensors can also be accessible through this interface. Threshold values of Soil Moisture, Humidity, and Temperature can be changed according to the plant type using the admin mode page. And that page can only access using an admin username and password, so it will protect the system from outside threats.

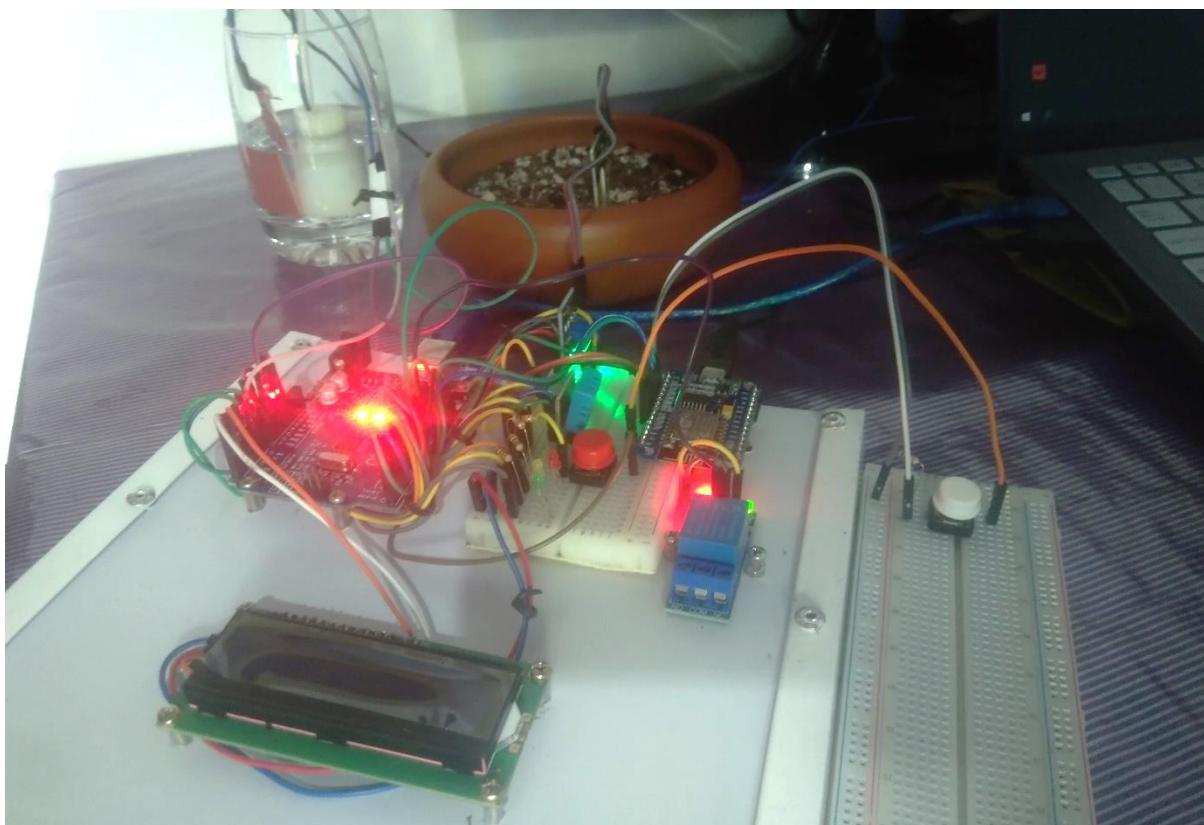
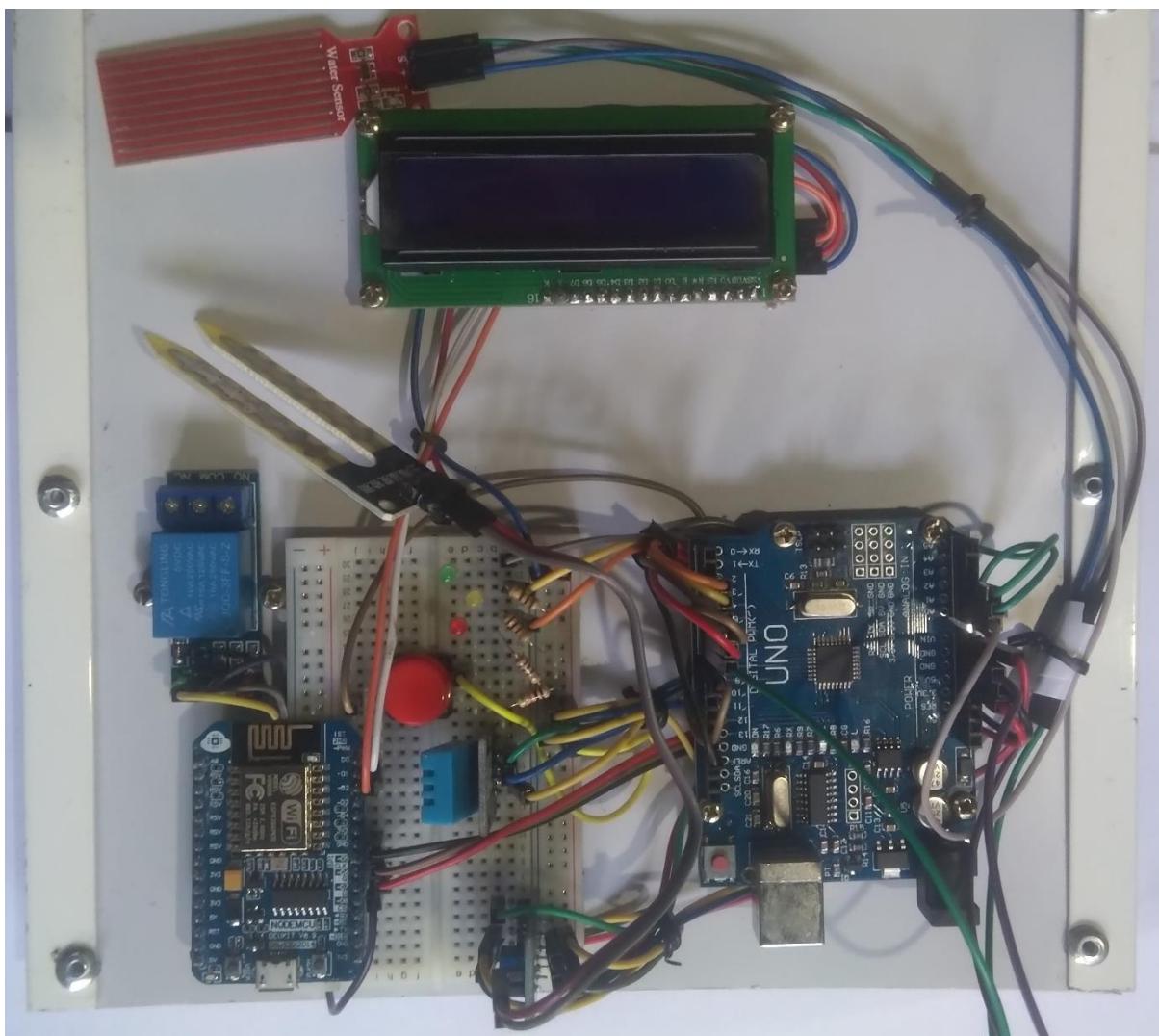
2 Conceptual Design



3 Implementation

3.1 Steps during Construction





3.2 Various States of Operation

1) Get threshold values from database and Send that data to the Arduino Uno

The image shows two side-by-side Arduino IDE windows. The left window is titled 'arduinouno' and the right window is titled 'nodemcu'. Both windows show the same code for an irrigation system.

```
arduinouno | Arduino 1.8.13
File Edit Sketch Tools Help
arduino
{
  Sending_To_NodeMCU();
  previousMilli = millis();
}
if(digitalRead(buttonPin) == HIGH)
{
  Sensor_Read();
  delay(100);
  Water_Pump();
  delay(100);
  Sending_To_NodeMCU();
}
}

void Sensor_Read()
{
  Soil_Moisture_Read();
  LCD_Display();
  delay(10);
  Humidity_And_Temp();
  delay(10);
  Water_Level_Read();
  delay(10);
}

void Soil_Moisture_Reading()
{
  soilmoisture = analogRead(soilPin);
  soilmoisture = map(soilmoisture,0,1023,90,30);
  Serial.print("Current Soil Moisture = ");
  Serial.println(soilmoisture);
}

Done uploading.

Sketch uses 15020 bytes (46%) of program storage space. Maximum is 32256 bytes.
Global variables use 1171 bytes (57%) of dynamic memory, leaving 877 bytes for local variables.
< 105
Arduino Uno on COM6

nodemcu | Arduino 1.8.13
File Edit Sketch Tools Help
nodemcu
{
  Serial.begin(115200);
  arduinouno.begin(115200);
  nodemcu.begin(115200);
  pinMode(buttonPin, INPUT_PULLUP);
  Init();
}

void loop()
{
  unsigned long currentMillis = millis();
  if(digitalRead(buttonPin) == HIGH)
  {
    Sending_To_Arduino();
    delay(100);
  }
  if(currentMillis > previousMilli)
  {
    if(Receiving_From_Arduino)
    {
      Sending_To_Data();
    }
  }
}

void Init()
{
  // connect to WiFi network
  Serial.println();
  Serial.println();
  Serial.println("Connecting to:");
  Serial.println(ssid);
}

Leaving...
Hard resetting via RTS pin...
<
```

2) Collect sensor readings and send them to the NodeMCU and then to the database

The screenshot shows two instances of the Arduino IDE running side-by-side. The left window is titled 'arduinouno | Arduino 1.8.13' and contains the code for the Arduino Uno. The right window is titled 'nodemcu | Arduino 1.8.13' and contains the code for the NodeMCU board.

Arduino Uno (Left Window):

```
arduino uno
{
  Sending_To_NodeMCU();
  previousMillis = millis();
}

if(digitalRead(buttonPin) == HIGH)
{
  Sensor_Read();
  delay(1000);
  Water_Pump();
  delay(1000);
  Sending_To_NodeMCU();
}

Sensor_Read();
delay(1000);
Water_Pump();
delay(1000);
Sending_To_NodeMCU();

Sensor_Reading();
{
  Soil_Moisture_Read();
  LCD_Display();
  delay(10);
  Humidity_And_Temp();
  delay(10);
  Water_Level_Read();
  delay(10);
}

void Soil_Moisture_Readings()
{
  soilmoisture = analogRead(soilPin);
  soilmoisture = map(soilmoisture, 0, 1023, 90, 30);
  Serial.print("Current Soil Moisture = ");
  Serial.println(soilmoisture);
}

Done uploading

Sketch uses 15020 bytes (46%) of program storage space. Maximum is 32256 bytes.
Global variables use 1171 bytes (57%) of dynamic memory, leaving 877 bytes for local variables
< 105 Arduino Uno on COM6
```

NodeMCU (Right Window):

```
nodemcu
{
  Serial.begin(115200);
  arduinouno.begin(115200);
  nodemcu.begin(115200);
  pinMode(buttonPin, INPUT_PULLUP);
  Init();
}

void loop()
{
  unsigned long currentMillis = millis();
  Received_Soil_Moisture = 60;
  Received_Humidity = 75;
  Received_Temperature = 32;
  Sensors_Data_Received();
  if(digitalRead(buttonPin) == HIGH)
  {
    Sending_To_Arduino();
    delay(1000);
  }
  if(currentMillis > serverStarted)
  {
    if(receivingFrom == 192.168.0.102)
    {
      connected = true;
      receivingFrom = 0;
      sendingToData = "GET /irrigationsystem/readings.php?";
      successful = false;
      disconnected = false;
    }
  }
}

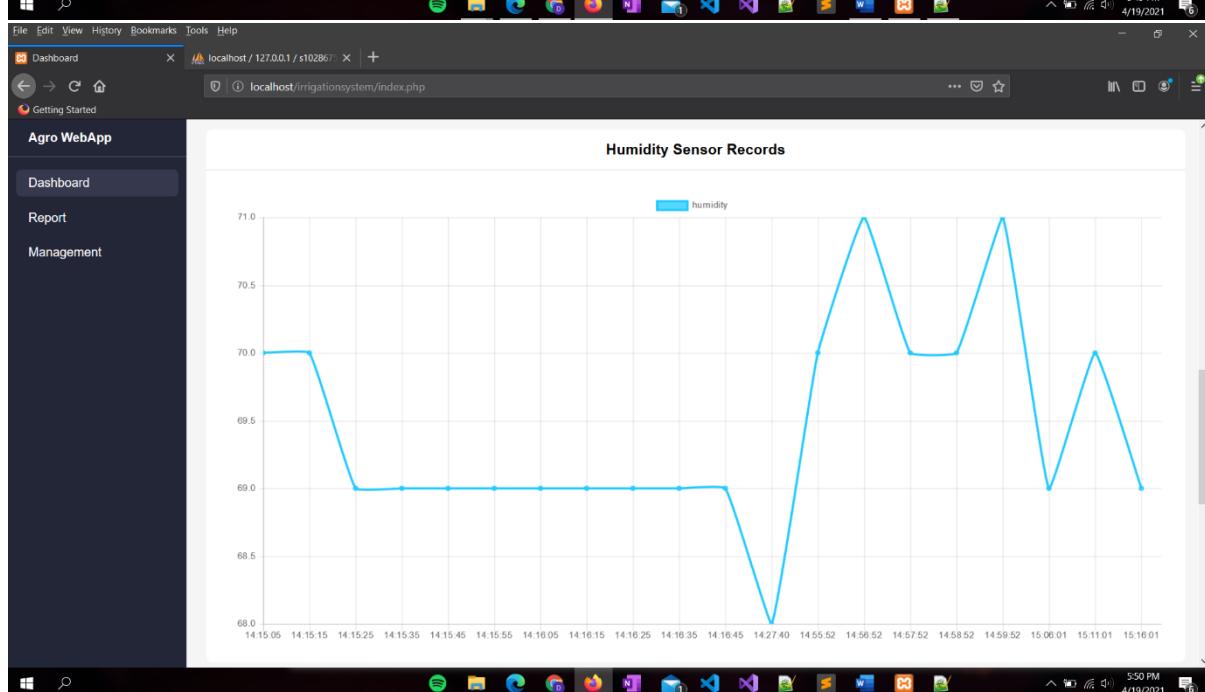
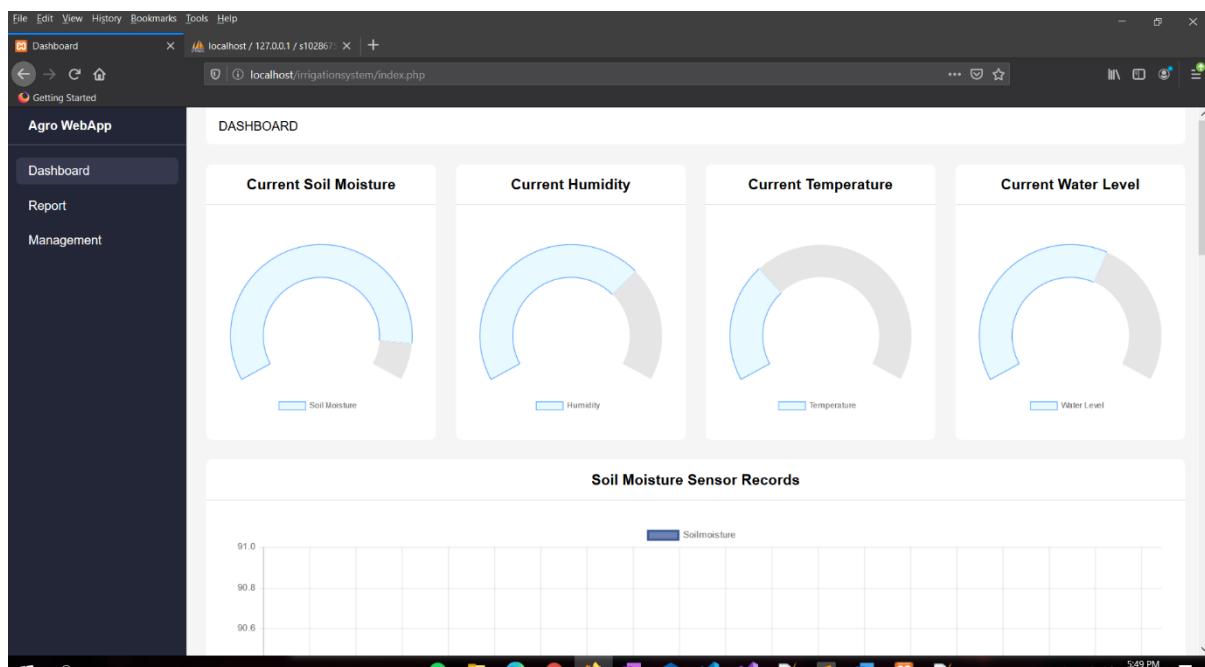
void Init()
{
  // connect to WiFi network
  Serial.println();
  Serial.println();
  Serial.println("Connecting to:");
  Serial.println(ssid);
}

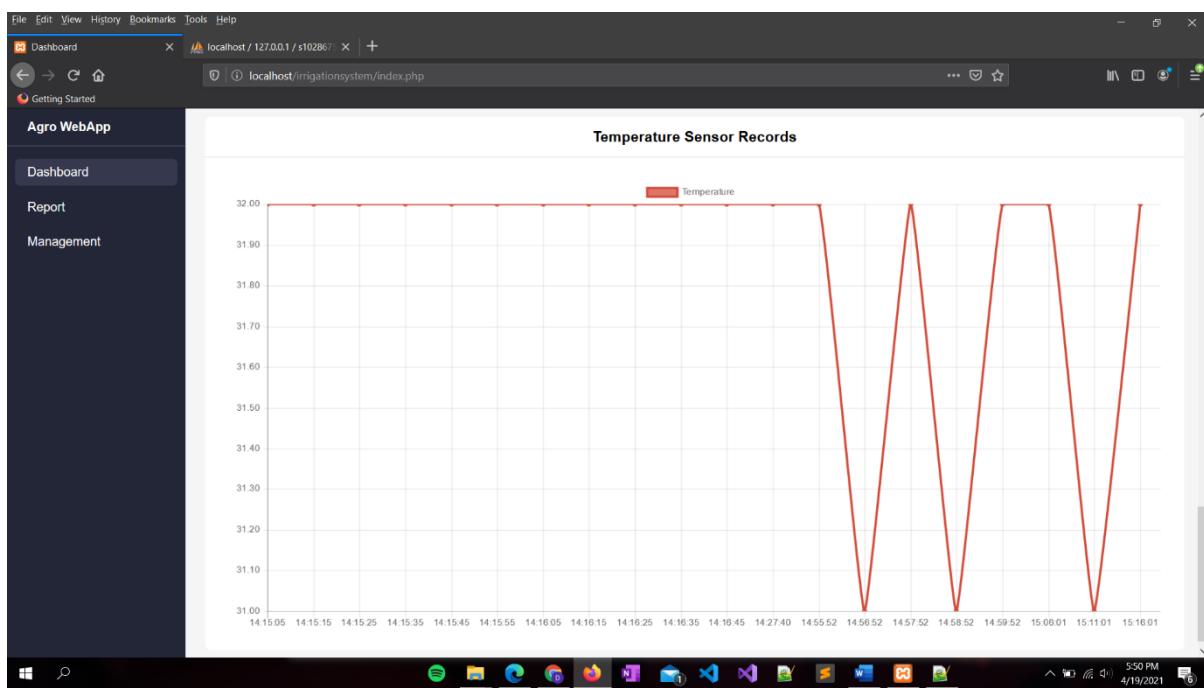
Leaving...
Hard resetting via RTS pin...
```

Both windows show the same baud rate of 115200 and the same serial port (COM6 for the Uno and COM5 for the NodeMCU). The NodeMCU window also displays the status of its connection to the Arduino Uno, indicating it is connected and successfully sending data.

3.3 Web Interface

1) index.php – Dashboard



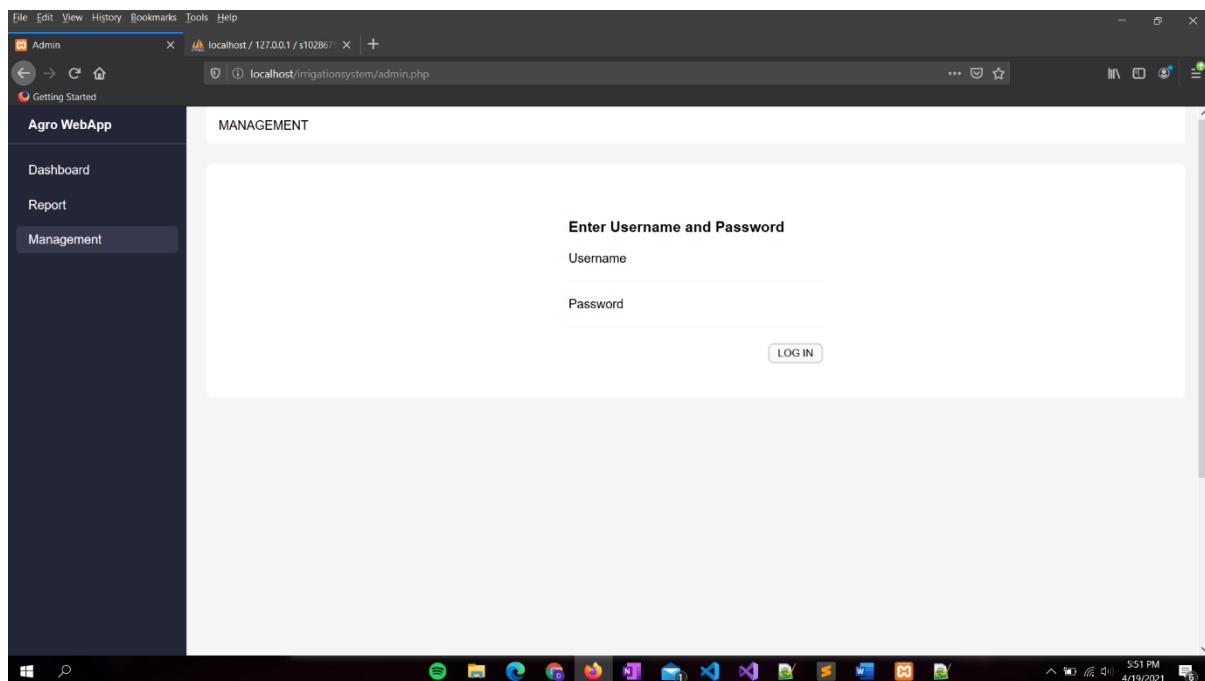


2) report.php - Report

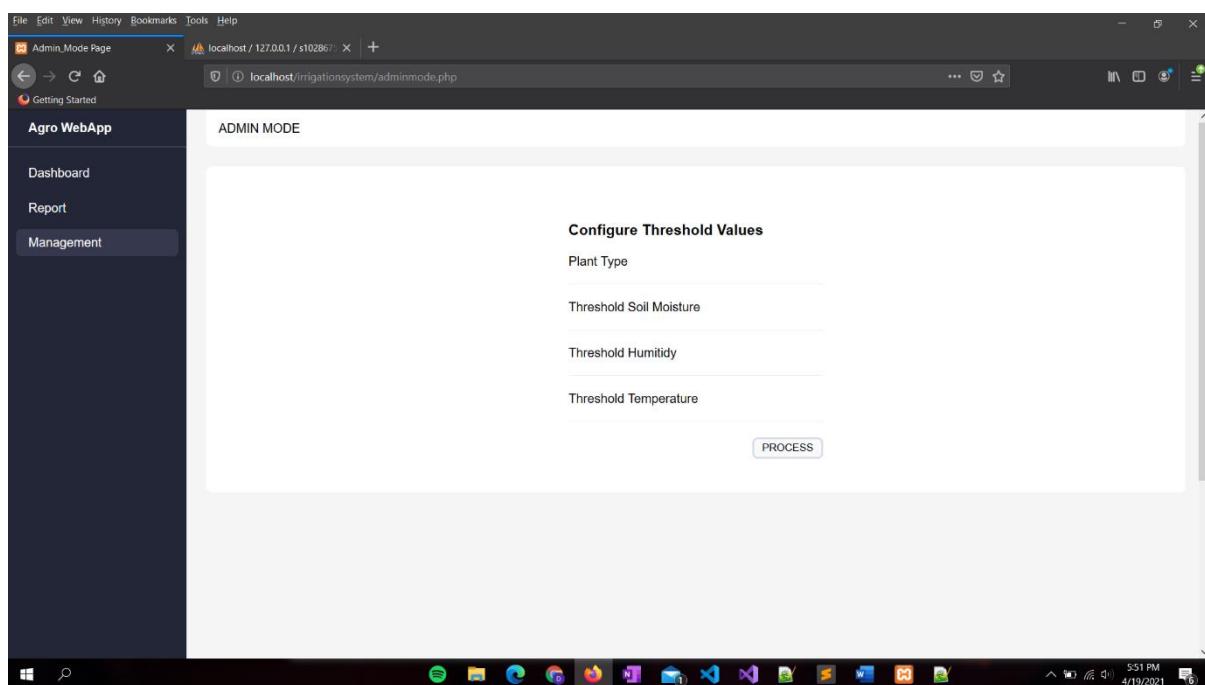
The screenshot shows a web browser window titled "localhost / 127.0.0.1 : 510286" displaying the "REPORT" page. The left sidebar has "Agro WebApp" with "Report" selected, and "Dashboard" and "Management" options. The main content area is titled "Last 40 Sensor Records" and contains a table with the following data:

Date	Soil Moisture	Humidity	Temperature	Water Level	Time
2021-04-07	100	540	20	80	13:27:37
2021-04-07	100	540	20	80	13:27:47
2021-04-07	100	540	20	80	13:27:57
2021-04-07	100	540	20	80	13:28:07
2021-04-07	100	540	20	80	13:28:17
2021-04-07	100	540	20	80	13:28:27
2021-04-07	100	540	20	80	13:30:20
2021-04-07	100	540	20	80	13:31:20
2021-04-07	100	540	20	80	13:32:20
2021-04-07	100	540	20	80	13:33:20
2021-04-07	100	540	20	80	13:34:20
2021-04-07	100	540	20	80	13:35:20
2021-04-07	100	540	20	80	13:36:20
2021-04-07	100	540	20	80	13:37:20
2021-04-07	100	540	20	80	13:38:20

3) admin.php – Management



4) adminmode.php – Admin Mode



4 Resources

4.1 Online Tutorials, Guides, and Software

- To achieve communication between NodeMCU and Arduino Uno (send sensor readings from Arduino Uno to the NodeMCU) – [1]
- To design gauge graphs for display current sensor reading values – [2], [3]
- To design line graphs for display last 20 sensor reading values – [4]
- To connect LCD 16x2 display with an I2C interface – [5]
- To connect relay module with NodeMCU – [6]
- To control water pump using relay module – [7]
- To control soil moisture sensor to read soil moisture value – [8]
- To control DHT11 to read humidity and temperature values – [9]
- To control water level sensor to read water level of the water tank – [10]

4.2 References

[1]	“Serial Communication between NodeMCU and Arduino,” <i>Engineeringprojectshub.com</i> , 19-Mar-2018. [Online]. Available: https://engineeringprojectshub.com/serial-communication-between-nodemcu-and-arduino/ . [Accessed: 19-Apr-2021].
[2]	vergissberlin, “ChartJS gauge example - CodeSandbox,” <i>Codesandbox.io</i> , 01-Dec-2018. [Online]. Available: https://codesandbox.io/s/jlonjk9zv5?file=/src/index.js:269-382 . [Accessed: 19-Apr-2021].
[3]	“ChartJS,” <i>Dyclassroom.com</i> . [Online]. Available: https://dyclassroom.com/chartjs/chartjs-how-to-create-doughnut-chart-using-data-from-mysql-mariadb-table-and-php . [Accessed: 19-Apr-2021].
[4]	“ChartJS,” <i>Dyclassroom.com</i> . [Online]. Available: https://dyclassroom.com/chartjs/chartjs-how-to-create-line-graph-using-data-from-mysql-mariadb-table-and-php . [Accessed: 19-Apr-2021].
[5]	R. T. (embedded Engineer), “How to Interface PCF8574 I2C LCD with ESP8266 NodeMCU?,” <i>Electronicshub.org</i> , 24-Feb-2021. [Online]. Available: https://www.electronicshub.org/esp8266-nodemcu-i2c-lcd/ . [Accessed: 19-Apr-2021].
[6]	Instructables.com. [Online]. Available: https://www.instructables.com/Interface-Relay-Module-With-NodeMCU/ . [Accessed: 19-Apr-2021].
[7]	“Arduino - Controls Pump,” <i>Arduinogetstarted.com</i> . [Online]. Available: https://arduinogetstarted.com/tutorials/arduino-controls-pump . [Accessed: 19-Apr-2021].
[8]	“Complete guide to use soil moisture sensor w/ examples,” <i>Arduino.cc</i> . [Online]. Available: https://create.arduino.cc/projecthub/electropeak/complete-guide-to-use-soil-moisture-sensor-w-examples-756b1f . [Accessed: 19-Apr-2021].
[9]	“DHT11 Temperature/Humidity Sensor,” <i>Arduino.cc</i> . [Online]. Available: https://create.arduino.cc/projecthub/techno_z/dht11-temperature-humidity-sensor-98b03b . [Accessed: 19-Apr-2021].
[10]	Lastminuteengineers.com. [Online]. Available: https://lastminuteengineers.com/water-level-sensor-arduino-tutorial/ . [Accessed: 19-Apr-2021].

5 Appendix

5.1 Source Codes for IoT Edge Device

1) source code of Arduino Uno

```
1 #include <SoftwareSerial.h>
2 #include <ArduinoJson.h>
3 #include <dht.h>
4 #include <LiquidCrystal_I2C.h>
5
6 // serial communication with nodemcu
7 SoftwareSerial nodemcu(5,6);
8
9 // water level sensor pins
10 #define sensorPower 7
11 #define sensorPin A0
12
13 // soil moisture sensor pin
14 #define smPin A2
15
16 // humidity and temperature sensor pin
17 #define dhtPin A1
18
19 // water level indicating led pins
20 #define redLed 2
21 #define yellowLed 3
22 #define greenLed 4
23
24 // button pin
25 #define buttonPin 8
26
27 int soilmoisture = 0,humidity = 0,temperature = 0,waterlevel = 0;
28 int tsoilmoisture = 0,thumidity = 0,ttemperature = 0;
29
30 int lowThresholdWaterLevel = 30;
31 int highThresholdWaterLevel = 45;
32
33 dht dhtsensor;
34
35 // water pump relay pin
36 #define relayPin 9
37
38 bool isReceived = false;
39
40 unsigned long previousMillis1 = 0,previousMillis2 = 0; // will store last time updated
41 const long interval = 300000;
42
43 String motorStatus = "OFF";
44
45 LiquidCrystal_I2C lcd(0x27,16,2); // set the LCD address to 0x3F for a 16 chars and 2
line display
46
47 void setup()
48 {
49   Serial.begin(115200);
50   nodemcu.begin(115200);
51   pinMode(sensorPower,OUTPUT);
52   pinMode(redLed,OUTPUT);
53   pinMode(yellowLed,OUTPUT);
54   pinMode(greenLed,OUTPUT);
55   pinMode(buttonPin,INPUT);
56
57   digitalWrite(sensorPower,LOW);
58   digitalWrite(redLed,LOW);
59   digitalWrite(yellowLed,LOW);
60   digitalWrite(greenLed,LOW);
61   delay(100);
62   pinMode(relayPin,OUTPUT);
63   lcd.init();
64   lcd.clear();
65   lcd.backlight(); // Make sure backlight is on
66   lcd.setCursor(4,0); //Set cursor to character 2 on line 0
67   lcd.print("Welcome!");
68   delay(100);
```

```

69 }
70
71 void loop()
72 {
73     if(!isReceived)
74     {
75         Receiving_From_NodeMCU();
76     }
77     else
78     {
79         unsigned long currentMillis = millis();
80
81         if (currentMillis - previousMillis1 >= interval)
82         {
83             Sensor_Readings();
84             delay(100);
85             Water_Pump_Control();
86             previousMillis1 = currentMillis;
87         }
88         if (currentMillis - previousMillis2 >= (interval*2))
89         {
90             Sending_To_NodeMCU();
91             previousMillis2 = currentMillis;
92         }
93         if(digitalRead(buttonPin) == HIGH)
94         {
95             Sensor_Readings();
96             delay(100);
97             Water_Pump_Control();
98             delay(100);
99             Sending_To_NodeMCU();
100        }
101    }
102 }
103
104 void Sensor_Readings()
105 {
106     Soil_Moisture_Readings();
107     delay(10);
108     Humidity_And_Temperature_Readings();
109     delay(10);
110     Water_Level_Readings();
111     delay(10);
112 }
113
114 void Soil_Moisture_Readings()
115 {
116     soilmoisture = analogRead(smPin);
117     soilmoisture = map(soilmoisture,0,1023,90,30);
118     Serial.print("Current Soil Moisture = ");
119     Serial.println(soilmoisture);
120 }
121
122 void Humidity_And_Temperature_Readings()
123 {
124     dhtsensor.read11(dhtPin);
125     delay(10);
126     humidity = dhtsensor.humidity;
127     temperature = dhtsensor.temperature;
128
129     if(isnan(humidity) || isnan(temperature))
130     {
131         Serial.println("Failed to read from DHT11");
132         Humidity_And_Temperature_Readings();
133     }
134     else{
135         Serial.print("Current Humidity = ");
136         Serial.print(humidity);
137         Serial.print("% ");

```

```

138     Serial.print("Temperature = ");
139     Serial.print(temperature);
140     Serial.println("C ");
141 }
143
144 void Water_Level_Readings(){
145     digitalWrite(sensorPower,HIGH);
146     delay(10);
147     waterlevel = analogRead(sensorPin);
148     waterlevel = map(waterlevel,0,1023,0,100);
149     digitalWrite(sensorPower,LOW);
150
151     if(waterlevel == 0){
152         Serial.println("Water Level: Empty");
153         digitalWrite(redLed,LOW);
154         digitalWrite(yellowLed,LOW);
155         digitalWrite(greenLed,LOW);
156     }
157     else if(waterlevel <= lowThresholdWaterLevel){
158         Serial.println("Water Level: Low");
159         digitalWrite(redLed,HIGH);
160         digitalWrite(yellowLed,LOW);
161         digitalWrite(greenLed,LOW);
162     }
163     else if(waterlevel <= highThresholdWaterLevel){
164         Serial.println("Water Level: Medium");
165         digitalWrite(redLed,LOW);
166         digitalWrite(yellowLed,HIGH);
167         digitalWrite(greenLed,LOW);
168         waterlevel += 10;
169     }
170     else if(waterlevel > highThresholdWaterLevel){
171         Serial.println("Water Level: High");
172         digitalWrite(redLed,LOW);
173         digitalWrite(yellowLed,LOW);
174         digitalWrite(greenLed,HIGH);
175         waterlevel += 45;
176     }
177     Serial.println();
178 }
179
180 void Sending_To_NodeMCU()
181 {
182     StaticJsonBuffer<500> jsonBuffer;
183     JsonObject& data = jsonBuffer.createObject();
184
185     // Assign collect data to Json Object
186     data["soilmoisture"] = soilmoisture;
187     data["humidity"] = humidity;
188     data["temperature"] = temperature;
189     data["waterlevel"] = waterlevel;
190     delay(100);
191
192     // Send data to NodeMCU
193     data.printTo(nodemcu);
194     jsonBuffer.clear();
195 }
196
197 void Receiving_From_NodeMCU()
198 {
199     StaticJsonBuffer<500> jsonBuffer;
200     JsonObject& data = jsonBuffer.parseObject(Serial);
201
202     if(data == JsonObject::invalid()){
203         jsonBuffer.clear();
204         return;
205     }
206 }
```

```

207 Serial.println("\n-----");
208 Serial.println("Sensors Data Received\n");
209 Serial.print("Received Threshold Soil Moisture: ");
210 tsoilmoisture = data["tsoilmoisture"];
211 Serial.println(tsoilmoisture);
212
213 Serial.print("Received Threshold Humidity: ");
214 thumidity = data["thumidity"];
215 Serial.println(thumidity);
216
217 Serial.print("Received Threshold Temperature: ");
218
219 ttemperature = data["ttemperature"];
220 Serial.println(ttemperature);
221 jsonBuffer.clear();
222 Serial.println("-----");
223
224 isReceived = true;
225 }
226
227 void LCD_Display()
228 {
229     lcd.clear();
230     lcd.setCursor(0,0); //Set cursor to character 2 on line 0
231     lcd.print("SoilMoisture:");
232     lcd.print(soilmoisture);
233
234     lcd.setCursor(0,1); //Move cursor to character 2 on line 1
235     lcd.print("Motor:");
236     lcd.print(motorStatus);
237 }
238
239 void Water_Pump_Control()
240 {
241     // check soil moisture level is less than threshold soil moisture level (soil dry)
242     if(soilmoisture < tsoilmoisture)
243     {
244         // check temperature is greater than threshold temperature (air dry)
245         if(temperature >= ttemperature)
246         {
247             // check humidity is greater than threshold humidity (air is more dry)
248             if(humidity >= thumidity)
249             {
250                 Pump_Process(10000);
251             }
252             // (air is dry)
253             else
254             {
255                 Pump_Process(5000);
256             }
257         }
258         else // in the night or rainy day
259         {
260             if(humidity >= thumidity)
261             {
262                 Pump_Process(3000);
263             }
264         }
265     }
266 }
267
268 void Pump_Process(int pTime)
269 {
270     // check enough water available in the water tank
271     if(waterlevel > 30)
272     {
273         motorStatus = "ON";
274         LCD_Display();
275         digitalWrite(relayPin, LOW); // turn on pump 5 seconds
276         delay(pTime);
277         motorStatus = "OFF";
278         LCD_Display();
279         digitalWrite(relayPin, HIGH); // turn off pump 5 seconds
280         delay(10);
281     }
282 }
283

```

2) source code of NodeMCU

```
1 #include <ESP8266WiFi.h>
2 #include <WiFiClient.h>
3 #include <ESP8266WebServer.h>
4 #include <ESP8266mDNS.h>
5 #include <ArduinoJson.h>
6 #include <SoftwareSerial.h>
7 SoftwareSerial nodemcu(D6,D5);
8 SoftwareSerial arduinouno(D7,D8);
9
10 // router details
11 const char* ssid = "SLT-LTE-WiFi-6317";
12 const char* password = "6HJT41F1FLY";
13
14 // host ip address
15 IPAddress server(192,168,8,101);
16
17 WiFiClient client;
18
19 int soilmoisture = 0,humidity = 0,temperature = 0,waterlevel = 0;
20 String tsm, th, tt;
21
22 #define buttonPin 5
23
24 void setup()
25 {
26     Serial.begin(115200);
27     arduinouno.begin(115200);
28     nodemcu.begin(115200);
29     pinMode(buttonPin,INPUT);
30     Init();
31 }
32
33 void loop()
34 {
35     unsigned long currentMillis = millis();
36     if(digitalRead(buttonPin) == HIGH)
37     {
38         Sending_To_ArduinoUno();
39         delay(100);
40     }
41     if(currentMillis > 30000){
42         if(Receiving_From_ArduinoUno())
43         {
44             Sending_To_Database();
45         }
46     }
47 }
48
49
50 void Init()
51 {
52     // connect to WiFi network
53     Serial.println();
54     Serial.println();
55     Serial.println("Connecting to");
56     Serial.println(ssid);
57
58     WiFi.begin(ssid,password);
59
60     while(WiFi.status() != WL_CONNECTED)
61     {
62         delay(500);
63         Serial.print(".");
64     }
65     Serial.println("");
66     Serial.println("WiFi connected");
67     delay(100);
68     Receiving_From_Database();
69     delay(100);
70 }
```

```

70 }
71
72 void Receiving_From_Database()
73 {
74     Serial.println("Server started");
75     Serial.print(WiFi.localIP());
76     delay(500);
77     Serial.println("connecting...");
78     if(client.connect(server,80))
79     {
80         Serial.println("Connected");
81         // make a HTTP request
82         Serial.println("GET /irrigationsystem/readthresholdvalues.php");
83         client.print("GET /irrigationsystem/readthresholdvalues.php");
84         client.print(" ");
85         client.print("HTTP/1.1");
86         client.println();
87         client.println("Host: 192.168.8.101");
88         client.println("Connection: close");
89         client.println();
90         delay(100);
91         String section="header";
92         while (client.connected() || client.available())
93         {
94             if (client.available())
95             {
96                 String line = client.readStringUntil('\r');
97                 //Serial.println(line);
98                 if (section=="header")
99                 {
100                     if (line=="\n")
101                         { // skips the empty space at the beginning
102                             section="json";
103                         }
104                 }
105                 else if (section=="json")
106                 {
107                     // print the good stuff
108                     section="ignore";
109                     String result = line.substring(1);
110                     int size = result.length() + 1;
111                     char json[size];
112                     result.toCharArray(json, size);
113                     StaticJsonBuffer<200> jsonBuffer;
114                     JsonObject& json_parsed = jsonBuffer.parseObject(json);
115                     if (!json_parsed.success())
116                     {
117                         Serial.println("parseObject() failed");
118                         return;
119                     }
120                     delay(100);
121                     tsm = (const char*) json_parsed["temparr"][0]["tsoilmoisture"];
122                     th = (const char*) json_parsed["temparr"][0]["thumidity"];
123                     tt = (const char*) json_parsed["temparr"][0]["ttemperature"];
124                     jsonBuffer.clear();
125                     delay(500);
126                     Serial.println("Sucessful");
127                 }
128             }
129         }
130         client.stop();
131         Serial.println("Disconnected\n");
132     }
133     else{
134         Serial.println("connection failed");
135         client.stop();
136     }
137 }
138

```

```

139 bool Receiving_From_ArduinoUno()
140 {
141     StaticJsonBuffer<500> jsonBuffer;
142     JsonObject& data = jsonBuffer.parseObject(nodemcu);
143
144     if(data == JsonObject::invalid()){
145         jsonBuffer.clear();
146         return false;
147     }
148
149     Serial.println("\n-----");
150     Serial.println("Sensors Data Received\n");
151     Serial.print("Received Soil Moisture: ");
152     soilmoisture = data["soilmoisture"];
153     Serial.println(soilmoisture);
154
155     Serial.print("Received Humidity: ");
156     humidity = data["humidity"];
157     Serial.println(humidity);
158
159     Serial.print("Received Temperature: ");
160     temperature = data["temperature"];
161     Serial.println(temperature);
162
163     Serial.print("Received Water Level: ");
164     waterlevel = data["waterlevel"];
165     Serial.println(waterlevel);
166     Serial.println("-----");
167
168     return true;
169 }
170
171 void Sending_To_ArduinoUno()
172 {
173     StaticJsonBuffer<500> jsonBuffer;
174     JsonObject& data = jsonBuffer.createObject();
175
176     // Assign collect data to Json Object
177     data["tsoilmointure"] = tsm.toInt();
178     data["thumidity"] = th.toInt();
179     data["ttemperature"] = tt.toInt();
180     delay(100);
181
182     // Send data to NodeMCU
183     data.printTo(arduinoUno);
184     jsonBuffer.clear();
185 }
186
187 void Sending_To_Database()
188 {
189     Serial.println("\nServer started");
190     Serial.print(WiFi.localIP());
191     delay(1000);
192     Serial.println("connecting...");
193     if(client.connect(server,80)){
194         Serial.println("Connected");
195         // make a HTTP request
196         Serial.println("GET /irrigationsystem/readings.php?");
197         client.print("GET /irrigationsystem/readings.php?soilmoisture=");
198         client.print(soilmoisture);
199         client.print("&humidity=");
200         client.print(humidity);
201         client.print("&temperature=");
202         client.print(temperature);
203         client.print("&waterlevel=");
204         client.print(waterlevel);
205         client.print(" ");
206         client.print("HTTP/1.1");
207         client.println();
208         client.println("Host: 192.168.8.101");
209         client.println("Connection: close");
210         client.println();
211         Serial.println("Sucessful");
212         Serial.println("Disconnected\n");
213     }
214     else{
215         Serial.println("connection failed");
216     }
217 }
218

```

5.2 Source Codes for Web Interface

1) admin.php – user has to login using username and password before accessing the admin mode

```
1  <!DOCTYPE html>
2
3  <html lang="en">
4  <head>
5      <meta charset="utf-8" />
6      <meta name="description" content="Irrigation System" />
7      <meta name="keywords" content="HTML5,irrigation system" />
8      <meta name="author" content="Dilshan Rodrigo" />
9      <meta name="viewport" content="width=device-width, initial-scale=1">
10     <link rel="stylesheet" type="text/css" href="css/style.css" />
11     <title>Admin</title>
12 </head>
13 <body>
14     <div id="vertical-nav">
15         <ul>
16             <li class="website-name"><a href="#" class="active-name" >Agro
17                 WebApp</a></li>
18             <li><a href="index.php">Dashboard</a></li>
19             <li><a href="report.php">Report</a></li>
20             <li><a href="admin.php" class="active">Management</a></li>
21         </ul>
22     </div>
23     <div style="margin-left:15.6%;padding:1px 16px;height:1000px;">
24         <div id="horizontal-nav">
25             <ul>
26                 <li><a href="#">MANAGEMENT</a></li>
27             </ul>
28         </div>
29         <?php require_once ("settings.php"); // connection info ?>
30         <div class="form-row">
31             <div class="form-sub-row">
32                 <h3>Enter Username and Password</h3>
33                 <form method="post">
34                     <div class="form-column">
35                         <label for="username">Username </label>
36                         <input type="text" name="username" id="username" />
37                     </div>
38                     <div class="form-column">
39                         <label for="password">Password </label>
40                         <input type="text" name="password" id="password" />
41                     </div>
42                     <div class="form-column">
43                         <input type="submit" name="svbtn" value="LOG IN" />
44                     </div>
45                 </form>
46             </div>
47         <?php
48             if(isset($_POST['svbtn'])) {
49                 if (isset($_POST["username"]) && isset($_POST["password"])) {
50                     $username = $_POST["username"];
51                     $password = $_POST["password"];
52
53                     $conn = @mysqli_connect ($host, $user, $pwd, $sql_db); // Checks if
54                     connection is successful
55                     if (!$conn) {
56                         // Display an error message
57                         echo "<p>Database connection failure</p>";
58                     }
59                 else {
60                     $query = "select username, password from admins_tb where username =
61                         '$username' && password = PASSWORD('$password')";
62                     $result = mysqli_query ($conn, $query);
63
64                     // checks if the execution was successful
65                     if (!$result)
66                         echo "<p>Something is wrong with ", $query, "</p>";
```

```

65
66
67
68
69
70
71
72
73
74
75
76
77
78
79

```

```

        }
        else {
            if (mysqli_num_rows($result) <= 0){
                echo '<script>alert("You cannot access to the admin
page."</script>';
            }
            else {
                header("Location: adminmode.php");
            }
        }
    }
}
?>
</body>
</html>

```

2) adminmode.php – user able to change plant type and the threshold values

```

1  <!DOCTYPE html>
2
3  <html lang="en">
4      <head>
5          <meta charset="utf-8" />
6          <meta name="description" content="Irrigation System" />
7          <meta name="keywords" content="HTML5,irrigation system" />
8          <meta name="author" content="Dilshan Rodrigo" />
9          <meta name="viewport" content="width=device-width, initial-scale=1">
10         <link rel="stylesheet" type="text/css" href="css/style.css" />
11         <title>Admin_Mode Page</title>
12     </head>
13     <body>
14         <div id="vertical-nav">
15             <ul>
16                 <li class="website-name"><a href="#" class="active-name" >Agro
WebApp</a></li>
17                 <li><a href="index.php">Dashboard</a></li>
18                 <li><a href="report.php">Report</a></li>
19                 <li><a href="admin.php" class="active">Management</a></li>
20             </ul>
21         </div>
22         <div style="margin-left:15.6%;padding:1px 16px;height:1000px;">
23             <div id="horizontal-nav">
24                 <ul>
25                     <li><a href="#">ADMIN MODE</a></li>
26                 </ul>
27             </div>
28             <?php require_once ("settings.php"); // connection info ?>
29             <div class="form-row">
30                 <div class="form-sub-row">
31                     <h3>Configure Threshold Values</h3>
32                     <form method="post">
33                         <div class="form-column">
34                             <label for="planttype">Plant Type </label>
35                             <input type="text" name="planttype" id="planttype" />
36                         </div>
37                         <div class="form-column">
38                             <label for="tsoilmoidure">Threshold Soil Moisture </label>
39                             <input type="text" name="tsoilmoidure" id="tsoilmoidure" />
40                         </div>
41                         <div class="form-column">
42                             <label for="thumidity">Threshold Humidity </label>
43                             <input type="text" name="thumidity" id="thumidity" />
44                         </div>
45                         <div class="form-column">
46                             <label for="ttemperature">Threshold Temperature </label>
47                             <input type="text" name="ttemperature" id="ttemperature" />
48                         </div>
49                         <div class="form-column">
50                             <input type="submit" name="btn" value="PROCESS" />
51                         </div>
52                     </form>
53                 </div>
54             <?php
55                 if(isset($_POST['btn'])) {
56
57                     $planttype = $_POST["planttype"];
58                     $tsoilmoidure = $_POST["tsoilmoidure"];
59                     $thumidity = $_POST["thumidity"];
60                     $ttemperature = $_POST["ttemperature"];
61
62                     $conn = @mysqli_connect ($host, $user, $pwd, $sql_db); // Checks if
connection is successful
63                     if (!$conn) {
64                         // Display an error message
65

```

```
66         echo "<p>Database connection failure</p>";
67     }
68     else {
69         $query = "select plantid,planttype from thresholdvalues_tb";
70         $result = mysqli_query ($conn, $query);
71         if (!$result) {
72             echo "<p>Something is wrong with ", $query, "</p>";
73         }
74     else {
75         if (mysqli_num_rows($result) > 0){
76             $query = "DELETE FROM thresholdvalues_tb";
77             $result = mysqli_query ($conn, $query);
78             if (!$result){
79                 echo "<p>Something is wrong with ", $query, "</p>";
80             }
81         }
82     }
83     $query = "INSERT INTO thresholdvalues_tb
84 (planttype,tsoilmoisture,thumidity,ttemperature) VALUES
85 ('$planttype','$tsoilmoisture','$thumidity','$ttemperature')";
86     $result = mysqli_query ($conn, $query);
87     if (!$result){
88         echo "<p>Something is wrong with ", $query, "</p>";
89     }
90     else {
91         header("Location: adminmode.php");
92     }
93     }
94     ?>
95 </body>
96 </html>
```

3) index.php – dashboard, which displays the last 20 sensor readings (line graphs) and current sensor readings (gauge graphs)

```
1  <!DOCTYPE html>
2
3  <html>
4  <head>
5      <meta charset="utf-8" />
6      <meta name="description" content="Irrigation System" />
7      <meta name="keywords" content="HTML5,irrigation system" />
8      <meta name="author" content="Dilshan Rodrigo" />
9      <meta name="viewport" content="width=device-width, initial-scale=1">
10     <link rel="stylesheet" type="text/css" href="css/style.css" />
11     <title>Dashboard</title>
12 </head>
13 <body>
14     <!-- navigation bar (vertical) -->
15     <div id="vertical-nav">
16         <ul>
17             <li class="website-name"><a href="#" class="active-name" >Agro
18                 WebApp</a></li>
19             <li><a href="index.php" class="active">Dashboard</a></li>
20             <li><a href="report.php">Report</a></li>
21             <li><a href="admin.php">Management</a></li>
22         </ul>
23     </div>
24     <div style="margin-left:15.6%;padding:1px 16px;height:1000px;">
25
26         <!-- navigation bar (horizontal) -->
27         <div id="horizontal-nav">
28             <ul>
29                 <li><a href="#">DASHBOARD</a></li>
30             </ul>
31         </div>
32
33         <!-- Three columns for gauge graphs -->
34         <div class="label-row">
35             <div class="label-column" style="margin-right: 25px;">
36                 <h3>Current Soil Moisture</h3>
37             </div>
38             <div class="label-column" style="margin-right: 25px;">
39                 <h3>Current Humidity</h3>
40             </div>
41             <div class="label-column" style="margin-right: 25px;">
42                 <h3>Current Temperature</h3>
43             </div>
44             <div class="label-column">
45                 <h3>Current Water Level</h3>
46             </div>
47         </div>
48         <div class="row">
49             <div class="column" style="margin-right: 25px;">
50                 <canvas id="doughnut-chartcanvas-1"></canvas>
51             </div>
52             <div class="column" style="margin-right: 25px;">
53                 <canvas id="doughnut-chartcanvas-2"></canvas>
54             </div>
55             <div class="column" style="margin-right: 25px;">
56                 <canvas id="doughnut-chartcanvas-3"></canvas>
57             </div>
58             <div class="column">
59                 <canvas id="doughnut-chartcanvas-4"></canvas>
60             </div>
61         </div>
62         <div class="line-graph-column">
63             <h3>Soil Moisture Sensor Records</h3>
64         </div>
65         <div class="line-graph">
66             <canvas id="mycanvas3"></canvas>
67         </div>
```

```
67      <div class="line-graph-column">
68          <h3>Humidity Sensor Records</h3>
69      </div>
70      <div class="line-graph">
71          <canvas id="mycanvas1"></canvas>
72      </div>
73      <div class="line-graph-column">
74          <h3>Temperature Sensor Records</h3>
75      </div>
76      <div class="line-graph">
77          <canvas id="mycanvas2"></canvas>
78      </div>
79
80      <!-- javascript -->
81      <script type="text/javascript" src="js/jquery.min.js"></script>
82      <script type="text/javascript" src="js/Chart.min.js"></script>
83      <script type="text/javascript" src="js/linegraph.js"></script>
84      <script type="text/javascript" src="js/gaugegraph.js"></script>
85  </body>
86  </html>
```

4) gaugegraph.js – design the gauge graphs

```
1 $(document).ready(function() {
2     $.ajax({
3         url : "http://localhost/irrigationsystem/readdatafromdb.php",
4         type : "GET",
5         success : function(data){
6             console.log(data);
7
8             var soilmoisture = [];
9             var humidity = [];
10            var temperature = [];
11            var waterlevel = [];
12
13            var len = data.length;
14
15            // gauge graph for soilmoisture
16            soilmoisture.push(data[0].soilmoisture);
17            soilmoisture.push(100 - data[0].soilmoisture);
18
19            // gauge graph for humidity
20            humidity.push(data[0].humidity);
21            humidity.push(100 - data[0].humidity);
22
23            // gauge graph for temperature
24            temperature.push(data[0].temperature);
25            temperature.push(100 - data[0].temperature);
26
27            // gauge graph for waterlevel
28            waterlevel.push(data[0].waterlevel);
29            waterlevel.push(100 - data[0].waterlevel);
30
31            var ctx1 = $("#doughnut-chartcanvas-1");
32            var ctx2 = $("#doughnut-chartcanvas-2");
33            var ctx3 = $("#doughnut-chartcanvas-3");
34            var ctx4 = $("#doughnut-chartcanvas-4");
35
36            var data1 = {
37                labels : ["Soil Moisture"],
38                datasets : [
39                    {
40                        label : "Current Soil Moisture",
41                        data: soilmoisture,
42                        backgroundColor: ["rgba(140, 223, 255, 0.2)"],
43                        borderColor: ["rgba(59, 147, 255,1)"],
44                        borderWidth: 1
45                    }
46                ]
47            };
48
49            var data2 = {
50                labels : ["Humidity"],
51                datasets : [
52                    {
53                        label : "Current Humidity",
54                        data: humidity,
55                        backgroundColor: ["rgba(140, 223, 255, 0.2)"],
56                        borderColor: ["rgba(59, 147, 255,1)"],
57                        borderWidth: 1
58                    }
59                ]
60            };
61
62            var data3 = {
63                labels : ["Temperature"],
64                datasets : [
65                    {
66                        label : "Current Temperature",
67                        data: temperature,
68                        backgroundColor: ["rgba(140, 223, 255, 0.2)"],
69                        borderColor: ["rgba(59, 147, 255,1)"]
70                    }
71                ]
72            };
73
74            var chart1 = new Chart(ctx1, {
75                type: 'doughnut',
76                data: data1
77            });
78
79            var chart2 = new Chart(ctx2, {
80                type: 'doughnut',
81                data: data2
82            });
83
84            var chart3 = new Chart(ctx3, {
85                type: 'doughnut',
86                data: data3
87            });
88        }
89    });
90
91});
```

```

70         borderWidth: 1
71     }
72   ]
73 };
74
75 var data4 = {
76   labels : ["Water Level"],
77   datasets : [
78     {
79       label : "Current Water Level",
80       data: waterlevel,
81       backgroundColor: ["rgba(140, 223, 255, 0.2)"],
82       borderColor: ["rgba(59, 147, 255,1)"],
83       borderWidth: 1
84     }
85   ]
86 };
87
88 var options = {
89   maintainAspectRatio: false,
90   circumference: Math.PI + 1,
91   rotation: -Math.PI - 0.5,
92   cutoutPercentage: 64,
93   legend : {
94     display : true,
95     position : "bottom"
96   }
97 };
98
99 var chart1 = new Chart( ctx1, {
100   type : "doughnut",
101   data : data1,
102   options : options
103 });
104
105 var chart2 = new Chart( ctx2, {
106   type : "doughnut",
107   data : data2,
108   options : options
109 });
110
111 var chart3 = new Chart( ctx3, {
112   type : "doughnut",
113   data : data3,
114   options : options
115 });
116
117 var chart4 = new Chart( ctx4, {
118   type : "doughnut",
119   data : data4,
120   options : options
121 });
122
123 },
124 error : function(data) {
125   console.log(data);
126 }
127 );
128
129 );

```

5) linegraph.js – design the line graphs

```
1 $(document).ready(function(){
2     $.ajax({
3         url : "http://localhost/irrigationsystem/readdatafromdb.php",
4         type : "GET",
5         success : function(data){
6             console.log(data);
7
8             var soilmoisture = [];
9             var humidity = [];
10            var temperature = [];
11            var dateandtime = [];
12
13            var len = data.length;
14
15            for(var i = len-1; i >= 0; i--) {
16                soilmoisture.push(data[i].soilmoisture);
17                humidity.push(data[i].humidity);
18                temperature.push(data[i].temperature);
19                dateandtime.push(data[i].dateandtime);
20            }
21
22            var chartdata1 = {
23                labels: dateandtime,
24                datasets: [
25                    {
26                        label: "humidity",
27                        fill: false,
28                        lineTension: 0.1,
29                        backgroundColor: "rgba(29, 202, 255, 0.75)",
30                        borderColor: "rgba(29, 202, 255, 1)",
31                        pointHoverBackgroundColor: "rgba(29, 202, 255, 1)",
32                        pointHoverBorderColor: "rgba(29, 202, 255, 1)",
33                        data: humidity
34                    }
35                ]
36            };
37
38            var chartdata2 = {
39                labels: dateandtime,
40                datasets: [
41                    {
42                        label: "Temperature",
43                        fill: false,
44                        lineTension: 0.1,
45                        backgroundColor: "rgba(211, 72, 54, 0.75)",
46                        borderColor: "rgba(211, 72, 54, 1)",
47                        pointHoverBackgroundColor: "rgba(211, 72, 54, 1)",
48                        pointHoverBorderColor: "rgba(211, 72, 54, 1)",
49                        data: temperature
50                    }
51                ]
52            };
53
54            var chartdata3 = {
55                labels: dateandtime,
56                datasets: [
57                    {
58                        label: "Soilmoisture",
59                        fill: false,
60                        lineTension: 0.1,
61                        backgroundColor: "rgba(59, 89, 152, 0.75)",
62                        borderColor: "rgba(59, 89, 152, 1)",
63                        pointHoverBackgroundColor: "rgba(59, 89, 152, 1)",
64                        pointHoverBorderColor: "rgba(59, 89, 152, 1)",
65                        data: soilmoisture
66                    }
67                ]
68            };
69        }
```

```

70         var ctx1 = $("#mycanvas1");
71
72         var LineGraph1 = new Chart(ctx1, {
73             type: 'line',
74             data: chartdata1
75         });
76
77         var ctx2 = $("#mycanvas2");
78
79         var LineGraph2 = new Chart(ctx2, {
80             type: 'line',
81             data: chartdata2
82         });
83
84         var ctx3 = $("#mycanvas3");
85
86         var LineGraph3 = new Chart(ctx3, {
87             type: 'line',
88             data: chartdata3
89         });
90     },
91     error : function(data) {
92
93     }
94 );
95 );

```

6) readdatfromdb.php – get the last 20 sensor readings from the database to an array to create the line graphs

```

1 <?php
2 //setting header to json
3 header('Content-Type: application/json');
4
5 //database
6 define('DB_HOST', 'localhost');
7 define('DB_USERNAME', 's102867540');
8 define('DB_PASSWORD', '271299');
9 define('DB_NAME', 's102867540_db');
10
11 //get connection
12 $mysqli = new mysqli(DB_HOST, DB_USERNAME, DB_PASSWORD, DB_NAME);
13
14 if(!$mysqli){
15     die("Connection failed: " . $mysqli->error);
16 }
17
18 //query to get data from the table
19 $query = sprintf("select soilmoisture, humidity, temperature, waterlevel,
cast(dateandtime as time) AS dateandtime from readvalues_tb ORDER BY id DESC LIMIT 20");
20
21 //execute query
22 $result = $mysqli->query($query);
23
24 //loop through the returned data
25 $data = array();
26 foreach ($result as $row) {
27     $data[] = $row;
28 }
29
30 //free memory associated with result
31 $result->close();
32
33 //close connection
34 $mysqli->close();
35
36 //now print the data
37 print json_encode($data);

```

7) report.php – display last 40 sensor readings as a table

```
1  <!DOCTYPE html>
2
3  <html lang="en">
4  <head>
5      <meta charset="utf-8" />
6      <meta name="description" content="Irrigation System" />
7      <meta name="keywords" content="HTML5, irrigation system" />
8      <meta name="author" content="Dilshan Rodrigo" />
9      <meta name="viewport" content="width=device-width, initial-scale=1">
10     <link rel="stylesheet" type="text/css" href="css/style.css" />
11     <title>Daily Report</title>
12 </head>
13 <body>
14     <div id="vertical-nav">
15         <ul>
16             <li class="website-name"><a href="#" class="active-name" >Agro
17                 WebApp</a></li>
18             <li><a href="index.php">Dashboard</a></li>
19             <li><a href="report.php" class="active">Report</a></li>
20             <li><a href="admin.php">Management</a></li>
21         </ul>
22     </div>
23     <div style="margin-left:15.6%;padding:1px 16px;height:1000px;">
24         <div id="horizontal-nav">
25             <ul>
26                 <li><a href="#">REPORT</a></li>
27             </ul>
28         </div>
29         <?php require_once ("settings.php"); // connection info ?>
30         <?php require_once ("createtable.php"); ?>
31         <?php
32             $conn = @mysqli_connect ($host, $user, $pwd, $sql_db); // Checks if connection
33             is successful
34             if (! $conn) {
35                 // Display an error message
36                 echo "<p>Database connection failure</p>";
37             }
38             else {
39                 $query = "select cast(dateandtime as date) AS date, soilmoisture, humidity,
40                     temperature, waterlevel, cast(dateandtime as time) AS time from
41                     readvalues_tb ORDER BY id DESC LIMIT 40";
42                 createTable($conn, $query);
43             }
44         ?>
45     </body>
46 </html>
```

8) settings.php – included database details

```
1  <?php
2      $host = "localhost";
3      $user = "s102867540"; // username
4      $pwd = "271299"; // password
5      $sql_db = "s102867540_db"; // database
6  ?>
```

9) createtable.php – extend file of the report.php file

```
1 <?php
2
3 function createTable($conn, $query) {
4
5     $result = mysqli_query($conn,$query) or die('Something is wrong with '.$query);
6
7     if (mysqli_num_rows($result) > 0)
8     {
9         echo "<div class=\"table-row-column\">\n";
10        echo "<h3>Last 40 Sensor Records</h3>";
11        echo "</div>";
12        echo "<div class ='table-row' >\n";
13        echo "<table>\n";
14        echo "<tr>\n ";
15            . "<th scope=\"col\">Date</th>\n"
16            . "<th scope=\"col\">Soil Moisture</th>\n"
17            . "<th scope=\"col\">Humidity</th>\n"
18            . "<th scope=\"col\">Temperature</th>\n"
19            . "<th scope=\"col\">Water Level</th>\n"
20            . "<th scope=\"col\">Time</th>\n"
21            . "</tr>\n";
22        /*
23        while ($row = mysqli_fetch_assoc($result)) {
24            echo "<tr>\n ";
25            echo "<td>",$row["date"], "</td>\n ";
26            echo "<td>",$row["soilmoisture"], "</td>\n ";
27            echo "<td>",$row["humidity"], "</td>\n ";
28            echo "<td>",$row["temperature"], "</td>\n ";
29            echo "<td>",$row["waterlevel"], "</td>\n ";
30            echo "<td>",$row["time"], "</td>\n ";
31            echo "</tr>\n ";
32        }*/
33
34     $date_arr = array();
35     $soilmoisture_arr = array();
36     $humidity_arr = array();
37     $temperature_arr = array();
38     $waterlevel_arr = array();
39     $time_arr = array();
40
41     $i = 39;
42
43     while ($row = mysqli_fetch_assoc($result))
44     {
45         $date_arr[$i] = $row["date"];
46         $soilmoisture_arr[$i] = $row["soilmoisture"];
47         $humidity_arr[$i] = $row["humidity"];
48         $temperature_arr[$i] = $row["temperature"];
49         $waterlevel_arr[$i] = $row["waterlevel"];
50         $time_arr[$i] = $row["time"];
51         $i--;
52     }
53
54     for($i = 0; $i < 40; $i++)
55     {
56         echo "<tr>\n ";
57         echo "<td>",$date_arr[$i], "</td>\n ";
58         echo "<td>",$soilmoisture_arr[$i], "</td>\n ";
59         echo "<td>",$humidity_arr[$i], "</td>\n ";
60         echo "<td>",$temperature_arr[$i], "</td>\n ";
61         echo "<td>",$waterlevel_arr[$i], "</td>\n ";
62         echo "<td>",$time_arr[$i], "</td>\n ";
63         echo "</tr>\n ";
64     }
65
66     echo "</table>\n ";
67     echo "</div>\n";
68
69     mysqli_free_result ($result);
70 }
71
72 ?>
```

10) readings.php – send sensor readings to the database through NodeMCU (get request)

```
1 <?php
2
3     class readings{
4         public $link = '';
5
6         function __construct($soilmoisture, $humidity, $temperature, $waterlevel){
7             $this->connect();
8             $this->storeInDB($soilmoisture, $humidity, $temperature, $waterlevel);
9         }
10
11        function connect(){
12            $this->link = mysqli_connect('localhost','s102867540','271299') or
13                die('Cannot connect to the Data Base');
14            mysqli_select_db($this->link,'s102867540_db') or die('Cannot select the
15                Data Base');
16        }
17
18        function storeInDB($soilmoisture, $humidity, $temperature, $waterlevel){
19            $query = "insert into readvalues_tb set soilmoisture = '$soilmoisture',
20                humidity = '$humidity', temperature = '$temperature', waterlevel =
21                '$waterlevel' ";
22            $result = mysqli_query($this->link,$query) or die('Something is wrong with
23                '.$query);
24        }
25    ?>
```

11) readthresholdvalues.php – get threshold values from database and send to NodeMCU

```
1 <?php
2
3     //Creating Array for JSON response
4     $response = array();
5
6     class writings{
7         public $link = '';
8
9         function __construct(){
10             $this->connect();
11             $this->storeInDB();
12         }
13
14         function connect(){
15             $this->link = mysqli_connect('localhost', 's102867540', '271299') or
16             die('Cannot connect to the Data Base');
17             mysqli_select_db($this->link, 's102867540_db') or die('Cannot select the
18             Data Base');
19         }
20
21         function storeInDB(){
22             $query = "select tsoilmoisture,thumidity,ttemperature from
23             thresholdvalues_tb";
24             $result = mysqli_query($this->link,$query) or die('Something is wrong with
25             '.$query);
26             $result = mysqli_fetch_array($result);
27
28             // temporary user array
29             $temparr = array();
30             $temparr["tsoilmoisture"] = $result["tsoilmoisture"];
31             $temparr["thumidity"] = $result["thumidity"];
32             $temparr["ttemperature"] = $result["ttemperature"];
33
34             $response["success"] = 1;
35             $response["temparr"] = array();
36
37             // Push all the items
38             array_push($response["temparr"], $temparr);
39         }
40     }
41
42     $writings = new writings();
43
44 ?>
```

- All css style files and javascript library files are included in the folder.