MINI-PROJECT REPORT

ON

‘IOT BASED CROP MONITORING SYSTEM’

SUBMITTED BY

ADITI OAK **(C. No.:** C 22019111217 **)**

ISHWARI PATIL **(C. No.:** C 22019111228 **)**

AARYA PHANSALKAR **(C. No. :**C 22019111235 **)**

UNDER THE GUIDANCE OF

PROF. SHASHIKANT SAHARE

MKSSS's
Cummins
COLLEGE OF ENGINEERING
FOR WOMEN

DEPARTMENT OF

ELECTRONICS AND TELECOMMUNICATION ENGINEERING

MKSSS’s

Cummins College of Engineering for Women, Pune

(An Autonomous Institute Affiliated to Savitribai Phule Pune University)

(2021-2022)

IOT BASED CROP MONITORING SYSTEM

ADITI OAK (C. No.: C 22019111217)
ISHWARI PATIL (C. No.: C 22019111228)
AARYA PHANSALKAR (C. No.: C 22019111235)

Under the Guidance of

PROF. SHASHIKANT SAHARE

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION ENGINEERING

MAHARSHI KARVE STREE SHIKSHAN SAMSTHA'S

CUMMINS COLLEGE OF ENGINEERING FOR WOMEN
KARVE-NAGAR, PUNE-411 052. (INDIA)

(An Autonomous Institute Affiliated to Savitribai Phule Pune University)

<u>2021-2022</u>

MAHARSHI KARVE STREE SHIKSHAN SAMSTHA'S

CUMMINS COLLEGE OF ENGINEERING FOR WOMEN

KARVE NAGAR, PUNE-411 052. (INDIA)

(An Autonomous Institute Affiliated to Savitribai Phule Pune University)

CERTIFICATE

This is to certify that the Mini Project work entitled

'IOT BASED CROP MONITORING SYSTEM'

is a bonafide record of the project work carried out in this institute

by

ADITI OAK (C. No. C 22019111217 **)**

ISHWARI PATIL (C. No. C 22019111228 **)**

AARYA PHANSALKAR (C. No. C 22019111235 **)**

in partial completion of the term work for the Third Year B.Tech.

in

Electronics and Telecommunication Engineering

in the academic year **2021-2022.**

This Mini-Project Report is a record of their own work carried out under our supervision and guidance.

Prof. S. G. Dube	**Dr. Prachi Mukherji**	**Dr. M. B. Khambete**
Internal Guide	**Head of Department (E&Tc)**	**Principal, CCOEW, Pune-52.**

ACKNOWLEDGEMENT

We would like to express our sincere gratitude towards our mini-project guide **Prof. Shashikant Sahare** for his constant encouragement and valuable guidance during the completion of this mini-project work.

We would also like to thank **Dr. Prachi Mukherji (H.O.D.,E&Tc)** for her continuous valuable guidance, support, valuable suggestions and her precious time in every possible way inspite of her busy schedule throughout our project activity.

We take this opportunity to express our sincere thanks to all the staff members of E&Tc. Department for their constant help whenever required. Finally; we express our sincere thanks to all those who helped us directly or indirectly in many ways in completion of this mini-project work.

Student Names

1) Aditi Oak, C. No. C- 22019111217

2) Ishwari Patil, C. No. C- 22019111228

3) Aarya Phansalkar, C. No. C- 22019111235

TABLE OF CONTENTS

1. ABSTRACT

As new technologies have been introduced and utilized in the modern world, there is a need to bring advancement in the field of agriculture also. Various kinds of research have been undergone to improve crop cultivation and have been widely used. In order to improve the crop productivity efficiently, it is necessary to monitor the environmental conditions in and around the field. Some of the parameters that have to be properly monitored to enhance the yield are soil characteristics, weather conditions, moisture, temperature, etc. Internet of Things (IOT) is being used in several real time applications. The introduction of IOT along with the sensor network in agriculture refurbishes the traditional way of farming. Online crop monitoring using IOT helps the farmers to stay connected to their field from anywhere and anytime. In this project, the system monitoring of soil moisture, temperature and light has been proposed by using NodeMCU ESP8266, Arduino Uno, sensors, namely Soil Moisture Sensor, Photodiode and DHT11 sensor and the Cloud. Further, displaying of the data log is done on the web which is accessible via mobile phones or other smart devices. Additionally, this data can be used to perform analysis like predict whether the current atmospheric conditions are ideal for the crop and whether the crop will survive or also to adjust the moisture content based on soil moisture sensor output.

2. INTRODUCTION

India is a largely agrarian society, with almost 55% of its population relying on agriculture for their livelihood. The agriculture sector in India employs close to 263 million people in what can only be deemed as the world’s largest agricultural land. With agriculture being of such prime value in India, ensuring the conditions surrounding the crops remain ideal is of utmost importance. Conditions such as temperature, humidity, soil moisture and the amount of light reaching the crops dictates if the crop is growing in an environment which facilitates or impedes its growth.

The crop monitoring systems currently at place are extremely time consuming and require large manpower. Both these problems are not ones that can be neglected, as in the current Indian landscape, manpower available for farming and its allied needs is plummeting rapidly. This has in turn put pressure on the existing farmers due to which they would not opt for any technological systems that would take a large amount of time of their day. With 82 percent of farmers being owners of small and marginal lands, the need for the crop monitoring system to be cost-effective is paramount and was a factor we took into consideration during the design and manufacturing phase.

Taking into consideration the requirements of the farmers, we have created an IoT based system for crop monitoring to ensure the crops are growing in a suitable environment and to inform the farmers in a timely fashion so that changes can be made in its surroundings if the conditions are not favourable. Our project is a system which measures the values of temperature, humidity, soil moisture surrounding the crops and the amount of light reaching the crops and

With the majority of Indian farmers being illiterate, it is important for the user interface to be encouraging. Hence, we have designed our UI in a way such that the farmers can access the data easily and can implement the changes in their farm accordingly.

During the ideation of our project topic, it was of prime importance for us to work on a project that would have technology for social good at its core. We have tried maintaining that thought process throughout our project timeline keeping in mind the needs of our end user, the farmers, and devising a system which will have actual on ground applications.

3. LITERATURE SURVEY

Sr. no.	Research Paper Title	Publication year	Summary	Link
1	Remote sensing for agricultural applications: A meta-review	November 2019	1. Illustrates recent research that would strengthen the role of remote sensing in providing operational, efficient and long-term services for agricultural applications. 2. Also talks about near real time models of farms using multiple unique datasets	Link to the research paper
2	IOT Based Smart Crop Monitoring in Farm Land	January 2018 (Imperial Journal of Interdisciplinary Research (IJIR))	1. Proposed an intelligent agriculture greenhouse monitoring system. The system performs data acquisition, processing, transmission and reception functions. 2. Realize the greenhouse environment system, where the aim is system efficiency and reduce the money, farming cost and save energy.	Link to the research paper
3	Review of research progress on soil moisture sensor technology	August 2021 (International Journal of Agricultural and Biological Engineering)	1. This review research is aimed at providing a certain perspective in the process of selecting soil moisture sensor products and measuring soil moisture. 2. Comparative study of different sensors and techniques	Link to the research paper
4	Enhancement of Plant Monitoring Using IoT	August 2018 (International	1. Proposed suitable methods for finding water level and taking needful actions if below threshold 2. Discussed different approaches of	Link to the research paper

		Journal of Engineering & Technology)	using cloud, Raspberry Pi, Arduino etc from IoT perspective	

4. PROPOSED METHODOLOGY

a) The major objective to be achieved through this project is timely monitoring of crops and its surrounding environment which is an important aspect of farming.

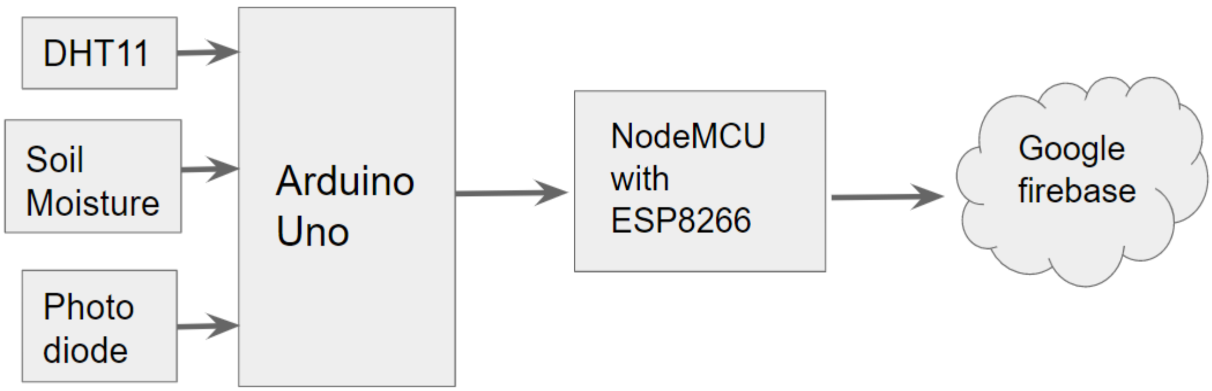
b) Continuous monitoring of crops becomes a tedious job for farmers and requires a lot of manpower. If neglected, this may cause losses in terms of yield.

c) The goal is to provide all the necessary information about various parameters like soil moisture, light intensity, humidity etc. handy to the farmers and accessible remotely. For this cloud technology is put to use.

d) The data acquisition is done from various sensors and transmitted to the microcontroller unit Arduino Uno. From microcontroller data is then pushed on cloud

e) Usage of cloud makes it feasible to store and analyze data remotely making it easy to take prompt corrective measures to better quality yield.

5. BLOCK DIAGRAM



5.1. BLOCK DIAGRAM EXPLANATION

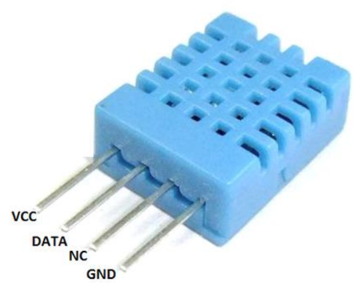
- Sensor blocks:
  - DHT 11 sensor senses the ambient temperature and humidity in the area surrounding the crop.
  - Photodiode senses the light intensity when connected in reverse bias.
  - Soil moisture sensor is a capacitive type sensor where the 2 probes act as capacitor plates and as moisture changes the capacitance changes are noted

- Arduino:
  - Acts as the microcontroller unit where all above sensors are interfaced, and data acquired from sensors is analysed.
  - Arduino IDE used for writing code

- NodeMCU:
  - For connecting hardware components that is the microcontroller unit to Google firebase.
  - Has the ESP8266 which is the actual wifi microchip, embedded in it

- Cloud:
  - Made use of Google Firebase Realtime Database
  - Data is stored realtime for further computations or displaying

5.2. SENSORS & COMPONENTS OF BLOCK DIAGRAM

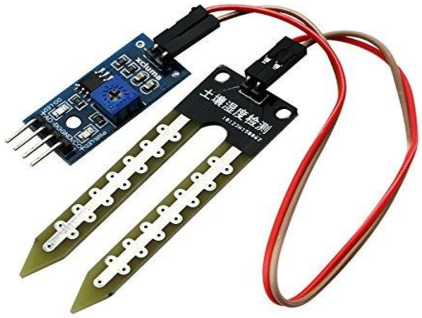


DHT11 Temperature and Humidity Sensor

- Operating Voltage: 3.5V to 5.5V
- Temperature Range: 0°C to 50°C
- Humidity Range: 20% to 90%
- Resolution: Temperature and Humidity both are 16

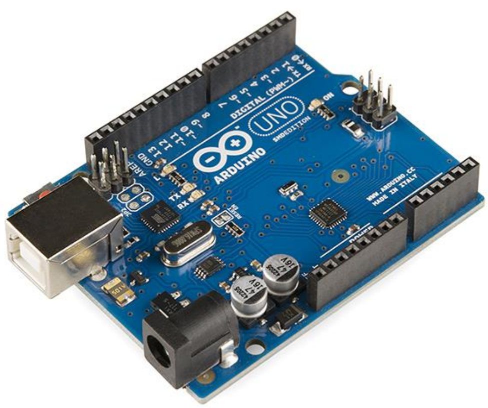
Photodiode Sensor

- Reverse Light current: 40Ma
- Reverse Dark current: 5nA
- Rise Time/ Fall Time: 45/45nS



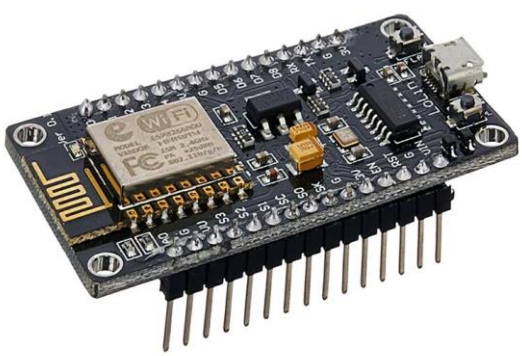
Soil Moisture Sensor

- Operating Voltage: 3.3V to 5V DC
- LM393 based design
- Moisture sensor pins, moisture detect pins



Arduino Uno

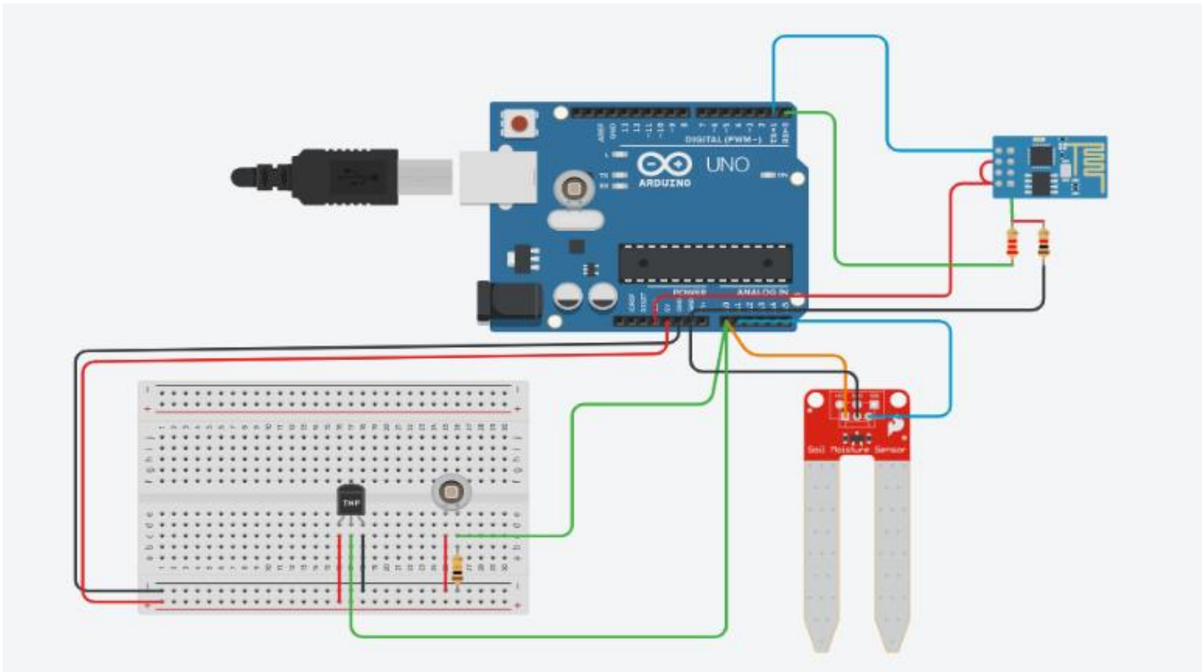
- OPERATING VOLTAGE - 5V
- DIGITAL I/O PINS - 14 (of which 6 provide PWM output)
- ANALOG INPUT PINS - 6
- FLASH MEMORY - 32 KB



ESP8266 NodeMCU

- ESP8266 is a microcontroller with WiFi capability while NodeMcu is a development board with ESP8266
- Wi-Fi Direct (P2P), soft-AP
- Integrated TCP/IP protocol stack
- Integrated temperature sensor
- General-purpose input/output (16 GPIO)
- Analog-to-digital conversion (10-bit ADC)

6. CIRCUIT DIAGRAM



7. ALGORITHM

a) Include DHT11 library

b) Define and initialise soil moisture sensor, photodiode and DHT11 pins connected to Arduino Uno

c) Set all sensors modes as output

d) Read soil moisture sensor value and photodiode values using analogRead() and convert the values in percentage

e) Record DHT11 sensor outputs using readHumidity(), readTemperature() and readTemperature() and computeHeatIndex() functions

f) Establish serial communication between Arduino Uno and Esp8266

g) Include ESP8266WIFI, Firebase, Software Serial libraries

h) Enter firebase projects credentials and set ssid and password of the WIFI router

i) Adjust baud rates and extract the values printed on the serial monitor

j) Push the extracted sensor values to Firebase

k) Make a HTML file to build the webpage and include css and script in it

l) Configure firebase in the script by adding the apikey, databaseURL, etc

m) Initialize firebase in our webapp

n) Hit the database using the api and extract the needful key, values from the database

o) Make divisions (divs) in HTML for every information that is to be displayed

p) Add styling to the page

q) Copy the path to the HTML file and run it locally

8. CODE

```
#include "DHT.h"

#define DHTPIN 2

#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

//photodiode sensor

 int pd=3;

 int senRead=A1;
```

```
//Soil moisture sensor

#define sensorPower 7

#define sensorPin A0

  //  This function returns the analog soil moisture measurement

int readSensor() {

  digitalWrite(sensorPower, HIGH);  // Turn the sensor ON

  delay(10);              // Allow power to settle

  int val = analogRead(sensorPin);  // Read the analog value form sensor

  digitalWrite(sensorPower, LOW);   // Turn the sensor OFF

  return val;             // Return analog moisture value

}

void setup() {

  // put your setup code here, to run once:

  Serial.begin(9600);

//  Serial.println(F("DHTxx test!"));

  dht.begin();

  pinMode(pd,OUTPUT);

  digitalWrite(pd,HIGH);

  //soil moisture

   pinMode(sensorPower, OUTPUT);

  // Initially keep the sensor OFF

  digitalWrite(sensorPower, LOW);

}
```

```
void loop() {
  // put your main code here, to run repeatedly:
  delay(2000);
  // Reading temperature or humidity takes about 250 milliseconds!
  // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
  float h = dht.readHumidity();
  // Read temperature as Celsius (the default)
  float t = dht.readTemperature();
  // Read temperature as Fahrenheit (isFahrenheit = true)
  float f = dht.readTemperature(true);
  // Check if any reads failed and exit early (to try again).
  if (isnan(h) || isnan(t) || isnan(f)) {
    Serial.println(F("Failed to read from DHT sensor!"));
    return;
  }
  // Compute heat index in Fahrenheit (the default)
  float hif = dht.computeHeatIndex(f, h);
  // Compute heat index in Celsius (isFahreheit = false)
  float hic = dht.computeHeatIndex(t, h, false);
//  Serial.print(F("Humidity: "));
  Serial.print(h);
  Serial.print(" ");
//  Serial.print(F("%  Temperature: "));
  Serial.print(t);
```

```
  Serial.print(" ");
//  Serial.print(F("°C "));
  Serial.print(f);
  Serial.print(" ");
//  Serial.print(F("°F  Heat index: "));
  Serial.print(hic);
  Serial.print(" ");
//  Serial.print(F("°C "));
  Serial.print(hif);
  Serial.print(" ");
//  Serial.println(F("°F"));
  //Photodiode
   float val=analogRead(senRead);  //variable to store values from the photodiode
   val=(val/1023)*100;
//   float ValInV=val*(5.0/1023.0);
//  Serial.print("Photodiode value is: ");          // prints the values from the sensor in serial
monitor
 Serial.print(val);
// Serial.println("V");
  //SoilMoisture
    //get the reading from the function below and print it
//  Serial.print("Analog output of soil moisture sensor is: ");
Serial.print(" ");
  Serial.print((float(readSensor())/1023)*100);
```

```
  Serial.print("\n");
  delay(2000);
}
```

NodeMCU:

```
#include <ESP8266WiFi.h>
#include <FirebaseArduino.h>
#include<SoftwareSerial.h>
SoftwareSerial espSerial(D1,D2);
#define FIREBASE_HOST "plantmonitoringsystem-da869-default-rtdb.firebaseio.com"
#define FIREBASE_AUTH "qCo3Mqk4Ksz5JLzEyQaPkjJUBMYR2CUcs4rdLs6K"
const char *ssid =  "";     // replace with your wifi ssid and wpa2 key
const char *pass =  "";
WiFiClient client;
String getValue(String data, char separator, int index)
  {
    int found = 0;
     int strIndex[] = {0, -1};
     int maxIndex = data.length()-1;
    for(int i=0; i<=maxIndex && found<=index; i++){
     if(data.charAt(i)==separator || i==maxIndex){
          found++;
          strIndex[0] = strIndex[1]+1;
          strIndex[1] = (i == maxIndex) ? i+1 : i;
```

```
      }
    }
    return found>index ? data.substring(strIndex[0], strIndex[1]) : "";
  }
void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
      WiFi.begin(ssid, pass);
     while (WiFi.status() != WL_CONNECTED)
      {
        delay(500);
        Serial.print(".");
      }
    Serial.println("");
    Serial.println("WiFi connected");
Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
  espSerial.begin(9600);
}
void loop() {
   //put your main code here, to run repeatedly:
    String msg=espSerial.readStringUntil('\n');
    String humidity = getValue(msg,' ',0);
    String TempC = getValue(msg,' ',1);
    String TempF = getValue(msg,' ',2);
```

```
   String HeatC = getValue(msg,' ',3);
   String HeatF = getValue(msg,' ',4);
   String Photodiode = getValue(msg,' ',5);
   String SoilM = getValue(msg,' ',6);
  if(humidity!="")
   Firebase.pushString("/SensorOutput/Humidity", humidity);          //setup path to send
Humidity readings
    if(TempC!="")
   Firebase.pushString("/SensorOutput/TemperatureinC", TempC);        //setup path to
send Temperature readings
//    Firebase.pushString("/SensorOutput/TemperatureinF", TempF);
 if(HeatC!="")
   Firebase.pushString("/SensorOutput/HeatIndexinC", HeatC);
//    Firebase.pushString("/SensorOutput/HeatIndexinF", HeatF);
 if(Photodiode!="")
   Firebase.pushString("/SensorOutput/PhotodiodeOutput", Photodiode);
    if(SoilM!="")
   Firebase.pushString("/SensorOutput/SoilMoistureOutput", SoilM);
   if (Firebase.failed())
   {
     Serial.print("pushing /logs failed:");
     Serial.println(Firebase.error());
     return;
  }
```

```
if ((humidity!="") && (TempC!="") && (HeatC!="") && (Photodiode!="") &&
(SoilM!=""))
{
//  Firebase.pushString("/SensorOutput", "");
  Firebase.setString("/LiveOutput/LiveData", msg);
  Serial.print("Humidity: ");
  Serial.println(humidity);
  Serial.print("Temperature in C: ");
  Serial.println(TempC);
  Serial.print("Heat Index in C: ");
  Serial.println(HeatC);
  Serial.print("Photo sensor Output in %: ");
  Serial.println(Photodiode);
  Serial.print("Soil Moisture Output in %: ");
  Serial.println(SoilM);
  Serial.print("\n");
}
```

Webpage:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Document</title>
  <script src="https://www.gstatic.com/firebasejs/7.2.2/firebase-app.js"></script>
  <script src="https://www.gstatic.com/firebasejs/3.1.0/firebase-database.js"></script>
</head>
<body
  style="
    background-image: linear-gradient(to right, #00ced1 , #90ee90);
    font-weight: bold;"
>
  <div id="root">
    <div style="font-size: 2.5rem; padding-left: 30%; padding-top: 1rem">
      CROP MONITORING SYSTEM
    </div>
    <div
      style="
        font-size: 1.8rem;
        padding-left: 40%;
        padding-top: 2rem;
        padding-bottom: 3rem;
      "
    >
      Real Time Sensor Data
    </div>
```

```
<div class="float-container">
 <div
  class="float-child"
  style="width: 10%; float: left; font-size: 1.2rem"
 >
  Timestamp
 </div>
 <div
  class="float-child"
  style="width: 18%; float: left; font-size: 1.2rem"
 >
  Humidity
 </div>
 <div
  class="float-child"
  style="width: 18%; float: left; font-size: 1.2rem"
 >
  Temperature
 </div>
 <div
  class="float-child"
  style="width: 18%; float: left; font-size: 1.2rem"
 >
  Heat Index
```

```
  </div>
  <div
    class="float-child"
    style="width: 18%; float: left; font-size: 1.2rem"
  >
    Photodiode
  </div>
  <div
    class="float-child"
    style="width: 18%; float: left; font-size: 1.2rem"
  >
    Soil Moisture
  </div>
</div>
<div style="height: 1.6rem"></div>
<div class="float-container">
  <div
    id="time"
    class="float-child"
    style="width: 10%; float: left"
  ></div>
  <div
    id="humidity"
    class="float-child"
```

```
      style="width: 18%; float: left"
    ></div>
    <div
      id="temp"
      class="float-child"
      style="width: 18%; float: left"
    ></div>
    <div
      id="heat"
      class="float-child"
      style="width: 18%; float: left"
    ></div>
    <div
      id="photo"
      class="float-child"
      style="width: 18%; float: left"
    ></div>
    <div
      id="soil"
      class="float-child"
      style="width: 18%; float: left"
    ></div>
  </div>
</div>
```

```
  <script>
    var firebaseConfig = {
      apiKey: "",
 authDomain: "",
 databaseURL: "",
 projectId: "",
 storageBucket: "",
 messagingSenderId: "",
 appId: "",
 measurementId: ""
    };
    firebase.initializeApp(firebaseConfig);
    var firebaseRef = firebase.database().ref().child("LiveOutput");
    firebaseRef.on("value", function (snapshot) {
      snapshot.forEach(function (element) {
        var myEle0 = document.getElementById("time");
        if (myEle0) {
          var today = new Date();
          var time =
            today.getHours() +
            ":" +
            today.getMinutes() +
            ":" +
            today.getSeconds();
```

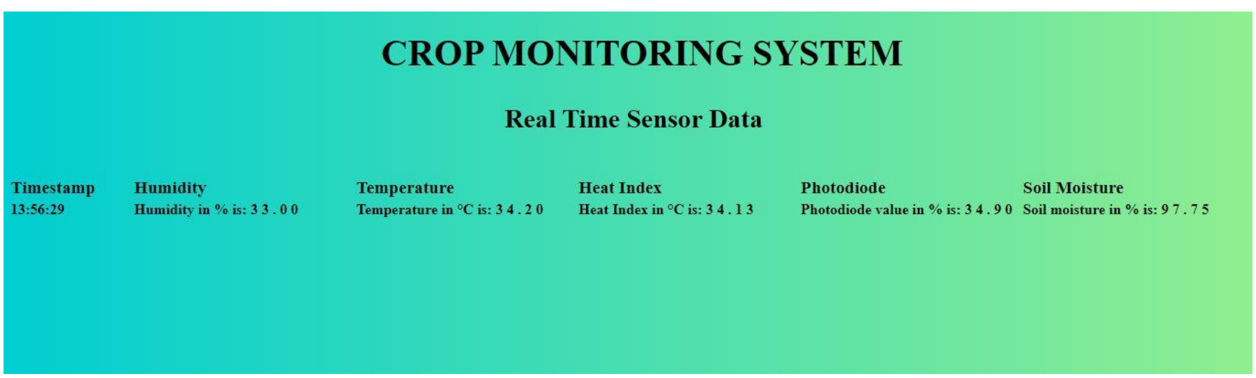
```
  document.getElementById("time").innerHTML = time;
}
var myEle1 = document.getElementById("humidity");
if (myEle1) {
  var myEle0 = document.getElementById("time");
if (myEle0) {
  var today = new Date();
  var time =
    today.getHours() +
    ":" +
    today.getMinutes() +
    ":" +
    today.getSeconds();
  document.getElementById("time").innerHTML = time;
}
  document.querySelector("#humidity").innerHTML += `
<div>
  Humidity in % is:
  ${element.val()[0]}
  ${element.val()[1]}
  ${element.val()[2]}
  ${element.val()[3]}
  ${element.val()[4]}
  </div>
```

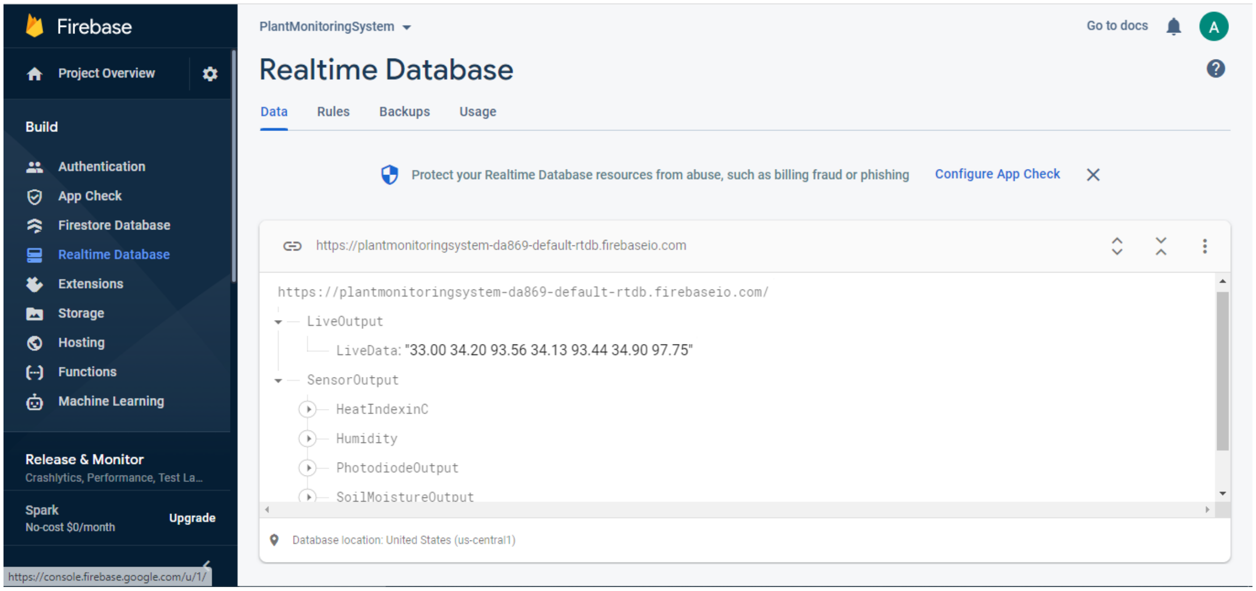
```
`;
}
var myEle2 = document.getElementById("temp");
if (myEle2) {
  document.querySelector("#temp").innerHTML += `
<div>
  Temperature in °C is:
  ${element.val()[6]}
  ${element.val()[7]}
  ${element.val()[8]}
  ${element.val()[9]}
  ${element.val()[10]}
  </div>
`;
}
var myEle3 = document.getElementById("heat");
if (myEle3) {
  document.querySelector("#heat").innerHTML += `
<div>
  Heat Index in °C is:
  ${element.val()[18]}
  ${element.val()[19]}
  ${element.val()[20]}
  ${element.val()[21]}
```

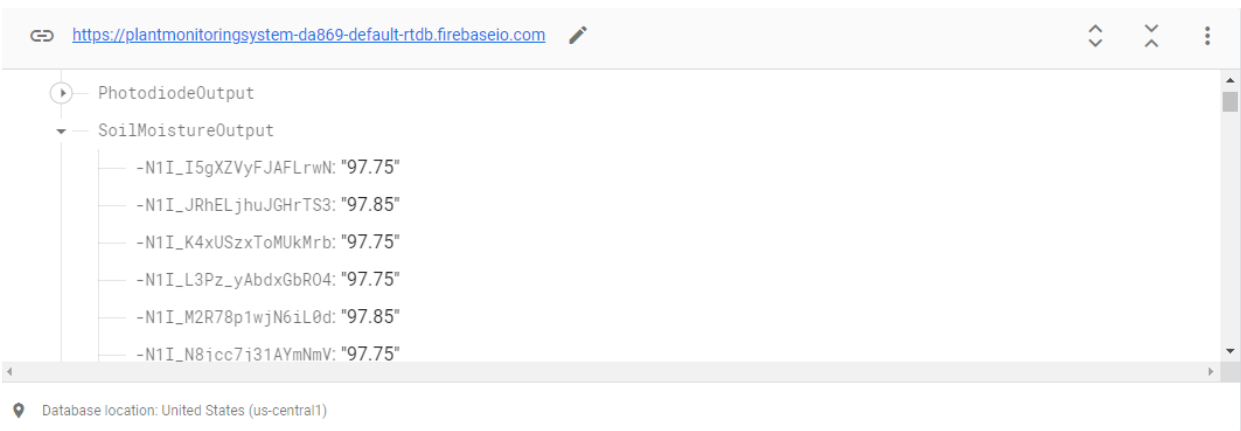
```
  ${element.val()[22]}
  </div>
`;
}
var myEle4 = document.getElementById("photo");
if (myEle4) {
  document.querySelector("#photo").innerHTML += `
<div>
  Photodiode value in % is:
  ${element.val()[30]}
  ${element.val()[31]}
  ${element.val()[32]}
  ${element.val()[33]}
  ${element.val()[34]}
  </div>
`;
}
var myEle5 = document.getElementById("soil");
if (myEle5) {
  document.querySelector("#soil").innerHTML += `
<div>
  Soil moisture in % is:
  ${element.val()[36]}
  ${element.val()[37]}
```

```
        ${element.val()[38]}

        ${element.val()[39]}

        ${element.val()[40]}

        </div>

       `;

       }

     });

   });

   window.setTimeout(function () {

     window.location.reload();

   }, 20000);

  </script>

 </body>

</html>
```

9. RESULTS







10. OBSERVATIONS / INTERPRETATIONS

By the use of various sensors, we could get the values of different parameters around the crop. These sensors were interfaced with Arduino UNO and data was acquired. The temperature and humidity values were recorded using the DHT 11 sensor, and then after some calculations on the temperature we could also get the heat index. We have used a photodiode to record the amount of light in the surroundings of the crop. The soil moisture value was acquired using the soil moisture sensor.

All of the data that we collected from the sensors was pushed to cloud (google firebase) using NodeMCU and stored there. We then extracted that data and it was displayed on a web page with a user friendly, minimalistic UI. Analyzing the data, the farmer can come to conclusions regarding the crop’s overall health and its growth conditions. These results will enable the farmer to do the needful actions like watering the plants if the soil moisture is too low, etc.

As an extension of the project, we aim at applying machine learning models to the data that we have acquired and predict the health of the crop, its survival chances in its current environment, predict the yield, etc.

11. CONCLUSION

With the introduction of this robust system, a direct impact can be observed in the lives of the farmers. The data acquired through the system enables the farmer to implement timely interventions to ensure optimal growth of crops. With the changing landscape of the Indian economy, the youth of India are migrating to urban areas, thus any technology for the farmlands must be designed in a way such that it requires low manpower for its functioning. With the use of technological advancements in farming still looked as secondary by most farmers, it is important for the crop monitoring system to be time-efficient as well so that the farmers would be willing to spend their time on it. With our country relying so heavily on agriculture, a system as such integrating crop management with IoT is truly the need of the hour.

12. COST OF PROJECT

Bill of Materials

Component	**Total Cost (Rs)**
Arduino Uno	750
NodeMCU ESP8266	300
Breadboard	26
Soil Moisture Sensor	69
Photodiode	5
DHT11 sensor	100
USB to micro USB (ESP8266 Power)	200
USB A to USB B cable	50
10k ohm resistor (DHT11)	1
3.3k (Photodiode)	1
Jumper Wires	60
Net Total	1562

13. REFERENCES

1) https://www.makerguides.com/nodemcu-esp8266-vs-arduino-uno-board/#:~:text=The%20NodeMCU%20ESP8266%20is%20a,at%2016%20MHz%20clock%20speed.

2) https://create.arduino.cc/projecthub/pulasthi-Narada/connecting-esp8266-to-firebase-to-send-receive-data-4adf66

3) https://www.engineersgarage.com/articles-arduino-dht11-humidity-temperature-sensor-interfacing/

14. DATASHEETS

1) DHT11:
   https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf

2) Soil Moisture Sensor:
   https://components101.com/modules/soil-moisture-sensor-module

3) Arduino Uno:
   https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf

4) NodeMCU Esp8266:
   https://espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf