IoT Based Smart Farming System

A Project Report

Submitted by:

NIHAR SHAH(1641054)
PURVANG SHAH(1641061)
PARTH PATEL(1641062)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

INFORMATION AND COMMUNICATION TECHNOLOGY(ICT)

at

Ahmedabad University

School of Engineering and Applied Sciences (SEAS) Ahmedabad, Gujarat

April ‘2020

DECLARATION

We hereby declare that the project entitled "*IoT Based Smart Farming System*" submitted for the B. Tech. (ICT) degree is our original work and the project has not formed the basis for the award of any other degree, diploma, fellowship or any other similar titles.

Place: Ahmedabad
Date: 30th April, 2020

Nihar Shah
(AU1641054)
30th April, 2020

Purvang Shah
(AU1641061)
30th April, 2020

Parth Patel
(AU1641062)
30th April, 2020

CERTIFICATE

This is to certify that the project titled "IoT Based Smart Farming System" is the bonafide work carried out by Nihar Shah, Purvang Shah, and Parth Patel, students of B Tech (ICT) of School of Engineering and Applied Sciences at Ahmedabad University during the academic year 2019-2020, in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology (Information and Communication Technology) and that the project has not formed the basis for the award previously of any other degree, diploma, fellowship or any other similar title.

Place: School of Engineering and Applied Science, Ahmedabad University
Date: 30th April 2020

Prof. Anurag Lakhlani,
Senior Lecturer,
SEAS, AU

Abstract

For the last half-century, technology has developed exponentially and it is still growing but the agriculture sector has not grown globally compared to all other sectors. Developing countries are still using traditional and backward methods for agriculture for activities. We are exploring ways to solve the problems faced by these farmers and will give possible solutions to make the current farming system into a more efficient and sophisticated farming system. We have divided this project into three modules. In the first module, we will test the soil and surrounding conditions and we would suggest which crop will be most productive based on the data. In the second module, we would maintain the crops through a smart rooftop and automated irrigation system. As heavy rains can damage the crops so with the help of the rooftop we will cover the crops and prevent crops from damaging. In the third module, we will detect disease on the crops using the machine learning algorithm and suggest possible solutions for the disease. And finally, we will make an application to integrate all these modules so that the farmer can keep track of all the activities on his fingertips.

Acknowledgment

It gives us immense pleasure submitting this project report towards the successful completion of our study. This project has given us an opportunity to think, implement, and interact with various aspects of technology in the field of computer science.

We want to thank our supervisor, Prof. Anurah Lakhlani, for his guidance. Without his directions, we might not have been able to reach this far. We would like to thank our university, Ahmedabad University, and our college, School of Engineering and Applied Science, for giving us the opportunity to explore and innovate in this project.

Lastly, we can never be excessively appreciative of our adoring guardians for their hard work. We will always remain in debt with what our parents have provided us. They have always encouraged us to gain knowledge. It is because of them and their efforts we have reached this far.

List of Figures

Figure 1: Gantt Chart	vii
Figure 2.1: Block Diagram of the system	6
Figure 3.1: Image of Arduino Mega 2560[4]	9
Figure 3.2: Image of LDR[5]	10
Figure 3.3: Image of Humidity Sensor[6]	10
Figure 3.4: Image of Temperature Sensor[7]	11
Figure 3.5: Image of Raindrop detection[8]	11
Figure 3.6: Image of NodeMCU [9]	12
Figure 3.7: Image of Soil Moisture[10]	13
Figure 3.8: Flowchart of Crop Suggestion	14
Figure 3.9: Flowchart of Application(Module 2)	15
Figure 3.10: Flowchart of Disease Detection	16
Figure 3.11: Sensors Configured with Arduino	17
Figure 3.12: Use Case Diagram	18
Figure 3.13: Sequence Diagram	19
Figure 4.1: Farmers' Data	21
Figure 4.2: Sensors' data which are installed on farms with timestamps.	21
Figure 4.3: Disease Detection result stored on server storage.	22
Figure 4.4: Application Login Page	23
Figure 4.5: Application Signup Page	24
Figure 4.6: Application Reset Password Page	25
Figure 4.7: Application Home Page	26
Figure 4.8: Application Charts	27
Figure 4.9: Application Vision Page	28
Figure 4.10: Application Pump Values Chart	29
Figure 4.11: Application Temperature History	30
Figure 4.12: Application Crop Prediction	31
Figure 4.13: Application Profile Page	32

Timeline/Gantt Chart

IoT Based Smart Farming System				
	Start Date	End Date	Timeline	Status
IoT Based Smart Farming System	Dec 9, 2019	Apr 30, 2020		
Project Title and Definition	Dec 9, 2019	Dec 25, 2019		Complete
Market survey and components orders	Dec 25, 2019	Jan 20, 2020		Complete
Testing all the Individual Sensors	Jan 20, 2020	Feb 10, 2020		Complete
Integration of all the sensors and codes	Jan 20, 2020	Feb 20, 2020		Complete
Debugging	Jan 20, 2020	Feb 26, 2020		Complete
Module 1 Completion	Jan 20, 2020	Mar 15, 2020		Complete
Module 3 Completion	Mar 1, 2020	Apr 20, 2020		Complete
		Burndown		

Figure 1: Gantt Chart

Table of Contents

Page No.

DECLARATION	ii
CERTIFICATE	iii
Abstract	iv
Acknowledgment	v
List of Figures	vi
Timeline/Gantt Chart	vii
1 Introduction	1
1.1 Problem Definition	1
1.2 Project Overview	1
1.3 Hardware Specification	2
1.4 Software Specification	3
2 Literature Review	5
2.1 Existing System	5
2.2 Proposed System	5
2.3 Feasibility Study	6
2.3.1 Technical Feasibility	7
2.3.2 Economic Feasibility	7
3 System Analysis & Design	9
3.1 Requirement Specification	9
3.1.1 Arduino Mega 2560	9
3.1.2 LDR Sensor	10
3.1.3 DHT22 Sensor	10
3.1.4 MLX90614 Temperature Sensor	11
3.1.5 Robodo Raindrop Sensor	11
3.1.6 NodeMCU	12

3.1.7 Soil Moisture Sensor 13

3.1.8 Arduino 1.8.11 13

3.1.9 Firebase 14

3.2 Flowcharts 14

3.3 Design and Test Steps/Criteria 17

3.4 Algorithms and Pseudo Code 19

3.5 Testing Process 20

4 Results/Outputs 21

5 Conclusions/Recommendations 33

6 References 35

1 Introduction

1.1 Problem Definition

Create an application that can suggest which crop can be grown on the given land based on the soil and surroundings condition and also detect disease at an early stage if there is any. Moreover, the rooftop system can be controlled automatically or manually based on the requirement of the farmer using the application.

1.2 Project Overview

The smart farming system is a solution for the farmers who don't have enough knowledge about which crop to grow and whether a crop is affected by any disease, by using IoT enabled devices.

At the client-side:

The proposed system uses sensors like the MLX90614 temperature sensor, DHT22 humidity sensor, LDR sensor, Robodo raindrop sensor, and soil moisture sensor. These sensors mainly involve in measuring the soil and surrounding condition. The goal of the project is to analyze which crop yields maximum production based on the soil and surrounding conditions of the land. Arduino Mega and NodeMCU are used to collect the data from the sensors and send the data to the database. Once the data is available in the database, data will be processed and analysis will be done and the result will be shown the application. Moreover, the registered user can upload a photo using the camera or select one photo from the gallery of the crop and the system will be able to detect any disease if there is any using Machine Learning algorithm. Also, the user would be able to control rooftop and irrigation facilities manually as well as both the facilities can be automated.

At the server-side:

The data of the sensor would be sent to NodeMCU from the Arduino Mega using serial communication. Once, the data is received by NodeMCU, it is uploaded to the Firebase database. Once the data is acquired, processing and analyzing of the data takes place on the backend.

1.3 Hardware Specification

For the implementation of the smart farming system, we require multiple IoT devices to be installed all across the farm. Data is acquired in the Arduino Mega, send over to the NodeMCU using serial communication, is stored in the database, and processed in the application.

Soil Moisture Sensor: Soil moisture sensor helps measure the volumetric water content in the soil. The data of the sensor is read using the Arduino Mega and sent to the database using NodeMCU.

Temperature Sensor: Using the temperature sensor, we can measure the temperature of the surroundings of the crop. The temperature of the surrounding is one of the factors for deciding which crop will be most suitable for growing on the given land. Similar to the soil moisture sensor, the data of the temperature sensor is sent to the database using NodeMCU.

Humidity Sensor: With the help of the humidity sensor we can measure the humidity of the surroundings of the crop. Similar to the temperature, humidity is also one of the key factors for deciding which crop will be most suitable for growing on the given land. The data of the humidity sensor is also sent to the database using NodeMCU.

Robodo Raindrop Sensor: The sensor helps in detecting the rain. So, in case of the heavy rainfall necessary steps can be taken to avoid the damage to the crops. When the water is exposed to the sensor, electric variations are produced due to change in the water volume.

NodeMCU: NodeMCU is a wi-fi sensor that is connected to the Arduino Mega and it receives the data of the sensors from the Arduino Mega using serial communication. Once the data is received from the Arduino Mega, data is sent to the database once the connection is established to the wi-fi connection available with internet connectivity.

Motor: Motor is used to cover the crops with a rooftop(a mechanical system designed to avoid the damage to the crops from the heavy rainfall) once the rainfall is detected.

1.4 Software Specification

- Platform: Windows, Ubuntu
- Database/Server: Firebase
- Languages: Python, C, Dart
- Framework: Flutter
- APIs: Flask
- Software: Android Studio, Arduino 1.8.11

2 Literature Review

2.1 Existing System

For the last half-century, technology has developed exponentially and it is still growing but the agriculture sector has not grown globally compared to all other sectors. Developing countries are still using traditional and backward methods for agriculture for activities. Now, researchers are trying to solve the Maximum Production Gain problem of agriculture using BIG DATA and IoT devices. There are many countries that have actually implemented these technologies and improved the productivity of the farm significantly. There are software and technologies globally such as Mavrx, Spensa Technologies, 360 Yield center, which helps you to increase productivity with the optimal use of the available resources. These technologies are very efficient in increasing the production of the product.[1][2]

Many times crops are affected by diseases and farmers are unaware till the crop is degraded completely. So, with the help of image processing, diseases in plants can be detected if there are any. There already exist systems where diseases are detected at a very early stage. These systems mostly use Machine Learning(ML) algorithms to detect diseases in crops. ML algorithms like backpropagation algorithm, Convolutional Neural Network(CNN), etc. Existing technologies are using drones to capture images of the farm. Once images are captured, image processing is done on images and the result is obtained whether the crop is having a disease or not.[3]

2.2 Proposed System

IoT based Smart Farming System combines the existing approaches and some new approaches to observe farms. As we have seen that there already exist systems that can improve the productivity of the crops. But these systems are highly expensive and most of the farmers cannot afford it. Also, we know that there exist systems for detecting diseases in crops. But these technologies use drones to capture images of the farms and many times images of crops get blurred because of the motion.

So, we are proposing a system that can be cost-efficient and most of the farmers can afford it. We have proposed a system that can solve the three most important problems of farmers which are:

1. Which crop to grow on the given land?

2. How to overcome the problem of excessive or restrained watering?
3. How to detect disease in plants at an early stage?

We have divided our project into three modules. In the first module, we will suggest farmers about which crop to grow based on the conditions of the soil/land and based on the condition of the surroundings. In the second module, we are making a rooftop system that will be helpful when there are a flood and a smart irrigation system that will decide when to water the farms based on the moisture of the soil. In the third module, we will implement a system that can detect diseases in plants at an early stage.

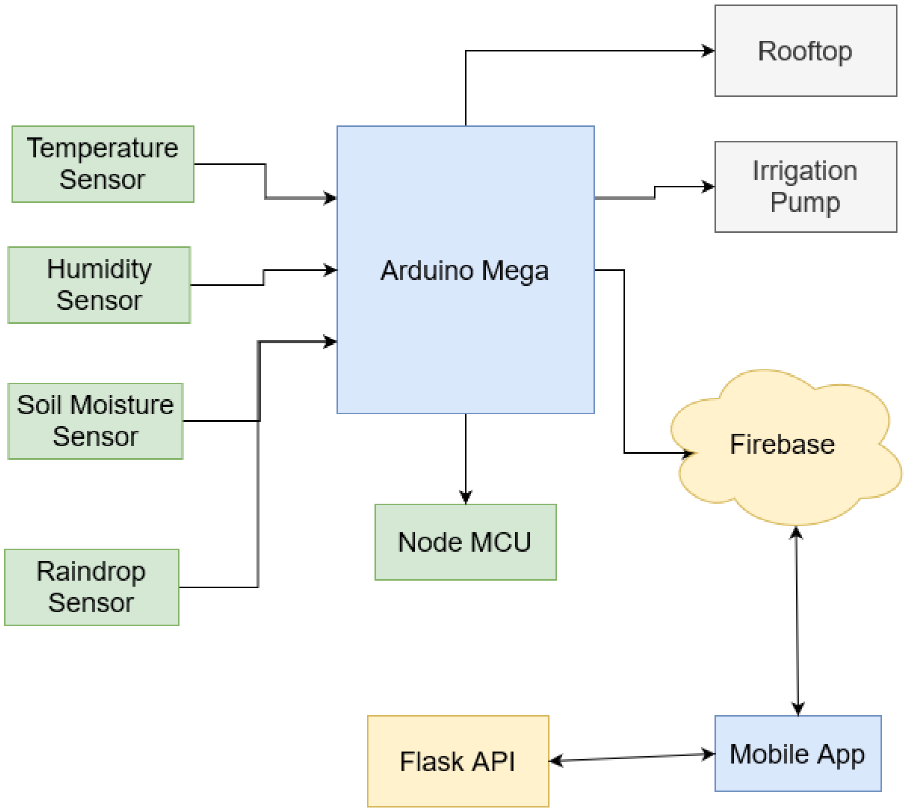

Figure 2.1: Block Diagram of the system

2.3 Feasibility Study

The smart farming system is independent and easy to use. The system can be easily installed at small farms and can be used easily. In our country, many farmers don't have proper knowledge about which crops to grow. By using this system the farmers would be able to maximize the production of the crops.

2.3.1 *Technical Feasibility*

For our system, we have used the Robodo raindrop sensor, soil moisture sensor, DHT22 moisture sensor, MLX90614 temperature sensor, and LDR sensor. While selecting the sensors, we had done research that which sensors had the best accuracy and selected the sensors accordingly.

2.3.2 *Economic Feasibility*

Apart from researching which sensors are the best for our project, we had also done research about the cost of the sensors. The system is cost-efficient along with its accuracy which is necessary as our system is for the farmers and the farmers cannot afford a costly system to improve their production.

3 System Analysis & Design

3.1 Requirement Specification

There are some hardware requirements that are required for our project.

3.1.1 *Arduino Mega 2560*

Figure 3.1: Image of Arduino Mega 2560[4]

Arduino is the most utilized microcontroller, particularly by the general population who like to try varieties of project ideas and some use it just for their side interest also. Arduino Mega is an open-source platform based on easy to use hardware. A microcontroller carries out the responsibility of a little PC. So essentially a microcontroller is a PC integrated into a single chip. It can lead to the activity of an embedded system. It comprises a processor center, memory, and a few input-output peripherals. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. It is cost-effective for its hardware and the software. Arduino Mega can be connected with various sensors or other electronic components to build a device, which will be able to do certain functionality.

It has 54 digital I/O pins (of which 15 provide PWM output), 16 analog inputs. It operates on 5V and DC current per I/O pin is 20 mA. The Arduino boards are inexpensive compared to other microcontrollers and run on Mac, Windows and Linux based operating systems.

3.1.2 *LDR Sensor*

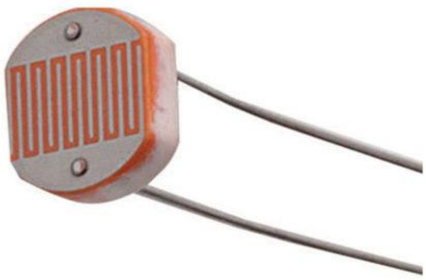

Figure 3.2: Image of LDR[5]

LDR sensor is basically a photocell that works on the principle of photoconductivity. When the intensity of light falling on the sensor increases, the resistance value of the resistor decreases.

3.1.3 *DHT22 Sensor*

Figure 3.3: Image of Humidity Sensor[6]

The DHT22 sensor is a low-cost humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air and gives a digital signal as an output on the data pin. The sensor is calibrated and doesn’t require extra components so you can get the right to measuring relative humidity and temperature. The sensor gives an output(new data) once every two seconds.

3.1.4 *MLX90614 Temperature Sensor*

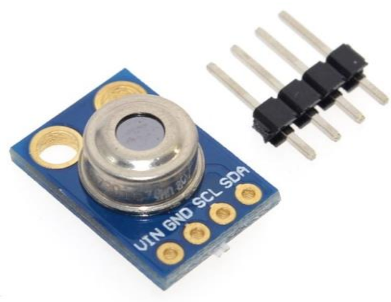

Figure 3.4: Image of Temperature Sensor[7]

The MLX90614 Temperature sensor is an infrared sensor for non-contact temperature measurements. The sensor is configured to continuously transmit the measured temperature in a range of -20 to 120℃, with an output resolution of 0.14℃. The main benefits of this sensor are that it is small in size, low cost, and easy to integrate.

3.1.5 *Robodo Raindrop Sensor*

Figure 3.5: Image of Raindrop detection[8]

The Raindrop detection sensor is an easy-to-use and low cost drop recognition sensor. The sensor works through a series of exposed parallel traces on board which produces electrical variations when drops or water volume changes. Once the analog output is obtained, it is easy to convert it into digital output. The sensor supports both digital and analog output.

Technical Specifications:

Operating voltage: 3.3-5V

Operating current: Less than 20mA

Output format: Digital output(0 and 1), Analog output

3.1.6 *NodeMCU*

Figure 3.6: Image of NodeMCU [9]

- Voltage: 3.3V
- Wi-Fi Direct(P2P), soft-AP
- Current consumption: 10μA~170mA
- Flash memory attachable: 16MB max(512K normal)
- Integrated TCP/IP protocol stack
- Processor: Tensilica L106 32-bit
- Processor speed: 80~160MHz
- RAM: 32K + 80K
- GPIOs: 17(multiplexed with other functions)
- Analog to Digital: 1 input with 1024 step resolution
- 802.11 support: b/g/n

- Maximum concurrent TCP connections: 5

3.1.7 *Soil Moisture Sensor*

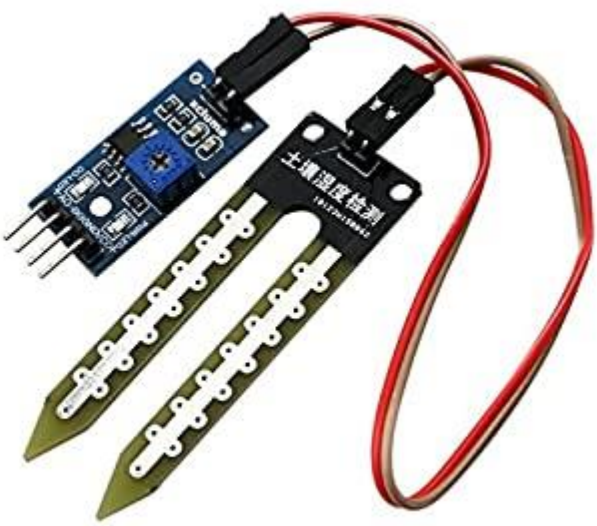

Figure 3.7: Image of Soil Moisture[10]

Soil moisture sensor measures the water content in the soil. Measuring soil moisture is important in agriculture to help farmers manage their irrigation systems more efficiently. Not only are farmers able to generally use less water to grow a crop, but they are also able to increase yields and the quality of the crop by better management of soil moisture during critical plant growth stages.

The sensor has a 3-Pin male header. The pins are as follows VCC(external 3.3V-5V), GND(external GND), and DO-board digital output interface(0 and 1).

There are also some software specifications that are required for this project.

3.1.8 *Arduino 1.8.11*

The open-source Arduino software makes it easy to write code and upload it to the board. It runs on Windows, Mac OS, and Linux. The environment is written in Java and is based on processing and other open-source software. The Arduino IDE supports the C and C ++ languages using special code structuring rules

3.1.9 *Firebase*

Firebase is a mobile and web application development platform. This platform provides varieties of functionality. We use this platform as our database server to store all sensors' data and farmers' data and disease detection results. Also, we use it to authenticate users while they log in and signup.

3.2 Flowcharts

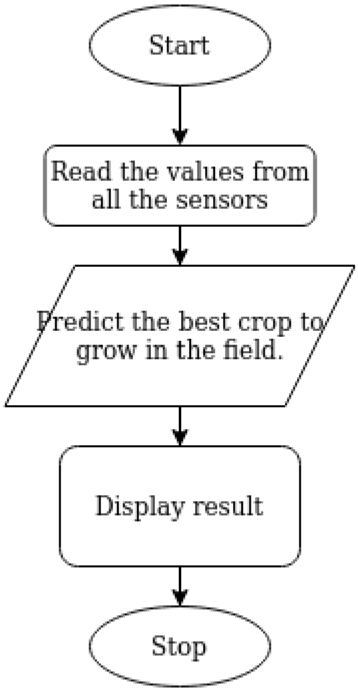

Figure 3.8: Flowchart of Crop Suggestion

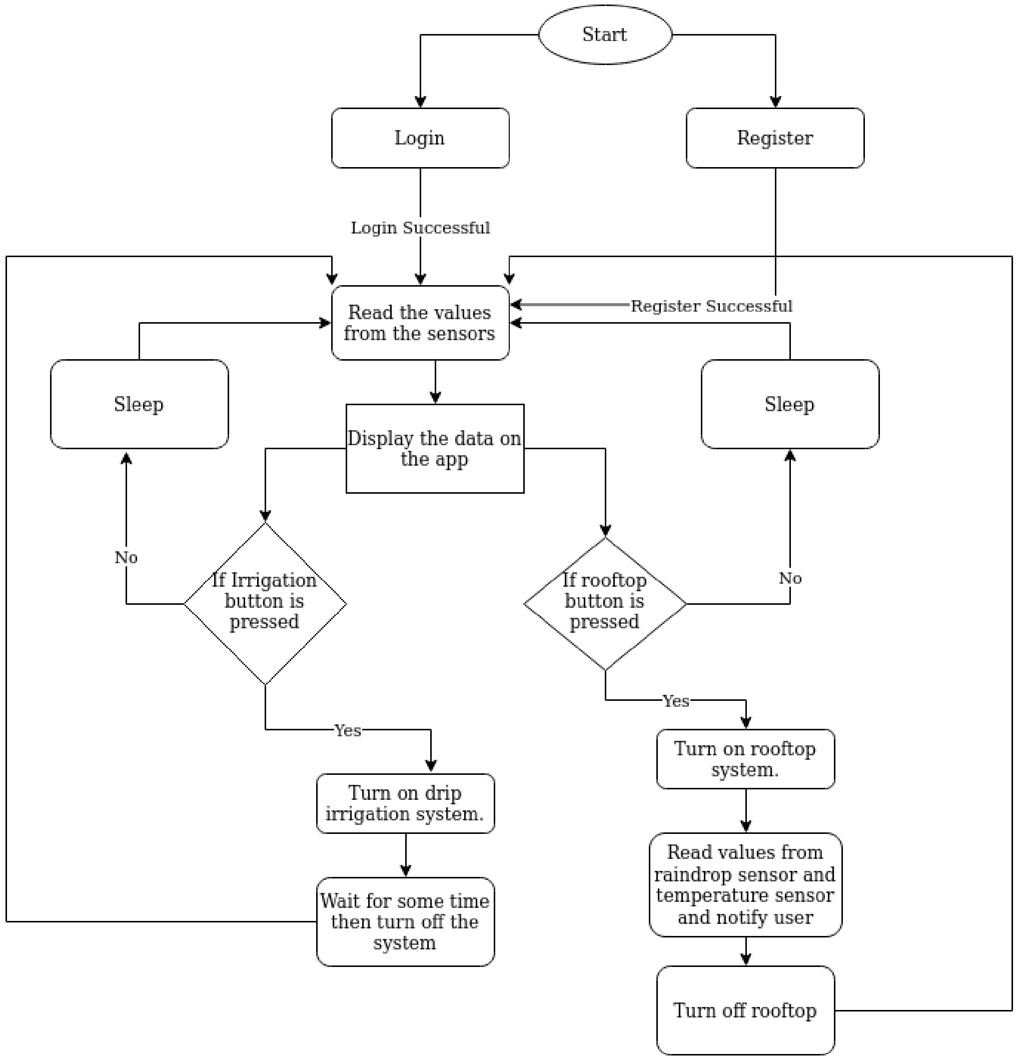

Figure 3.9: Flowchart of Application(Module 2)

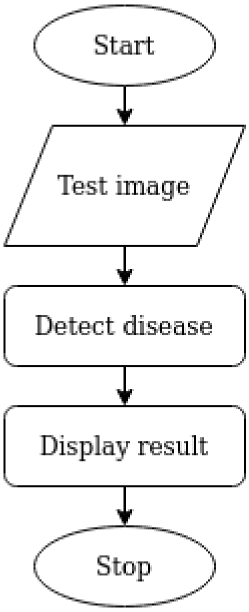

Figure 3.10: Flowchart of Disease Detection

3.3 Design and Test Steps/Criteria

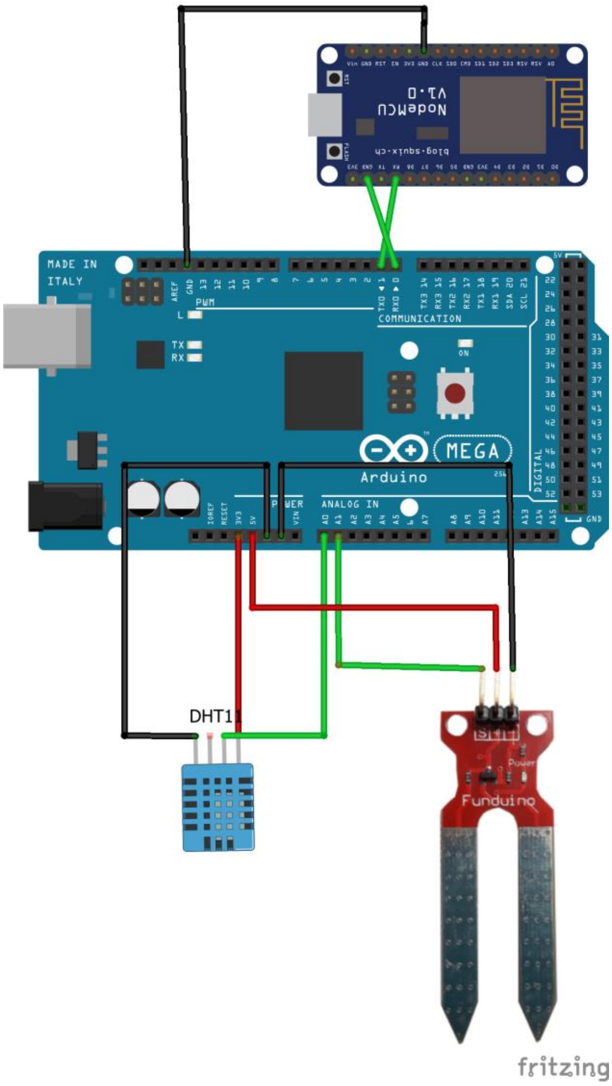

Figure 3.11: Sensors Configured with Arduino
*[Above diagram is drawn using **fritzing** software. Source: https://fritzing.org/home/]*

Firstly, sensors like DHT11 and Soil moisture are connected to the Arduino Mega to get the data of the surroundings. Then the NodeMCU is also connected to the Arduino Mega to send the data of the sensor to the database.

The DHT11 humidity sensor has 4 pins(Vcc, GND, Analog, and Digital). The first pin is connected to Vcc which gives a 5V power supply, the second pin is connected to A0 of the Arduino, and the last pin is connected to the GND.

The soil moisture sensor has 4 pins(Vcc, GND, Analog, and Digital). The first pin is connected to Vcc which gives a 5V power supply, the second pin is connected to A0 of the Arduino, and the last pin is connected to the GND.

The NodeMCU Wi-Fi module is connected to the Arduino as described below:
Rx(NodeMCU) - Tx(Arduino)

Tx(NodeMCU) - Rx(Arduino)
GND(NodeMCU) - GND(NodeMCU)

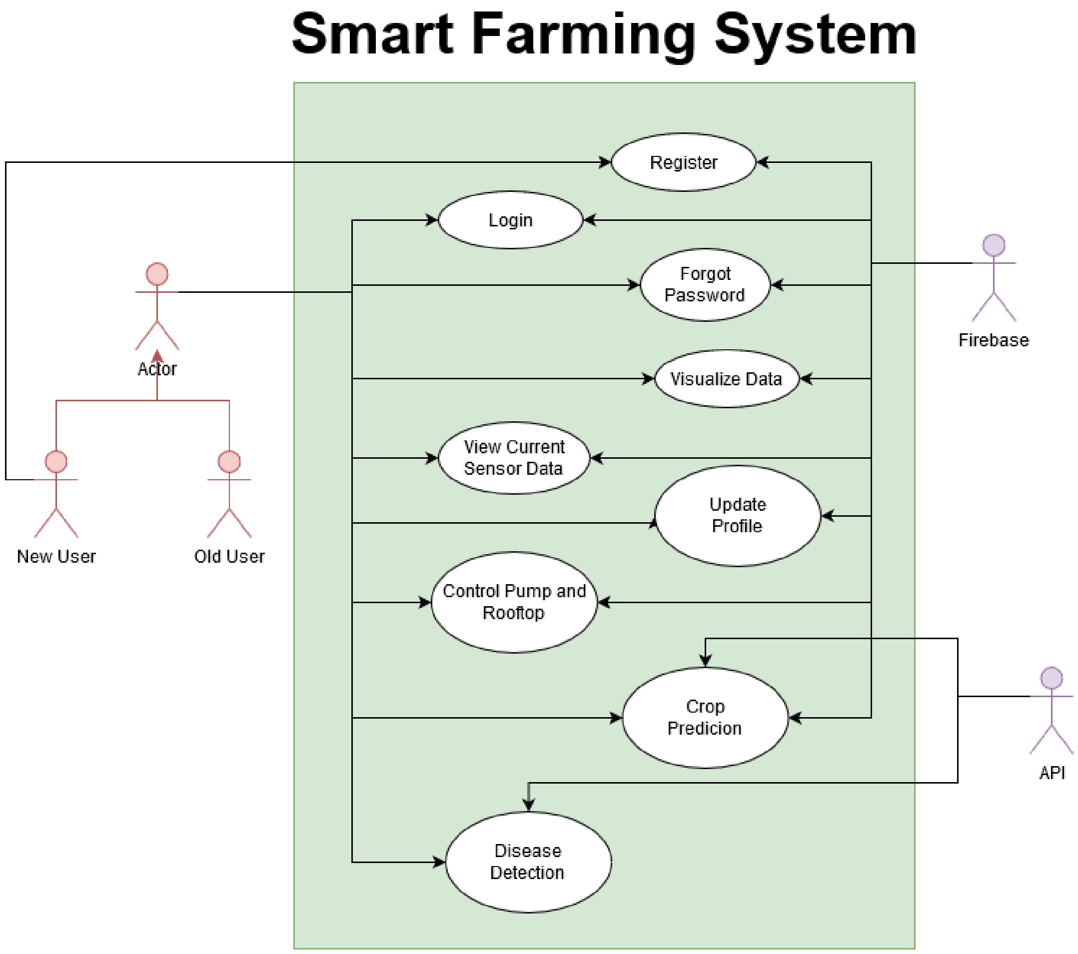

Figure 3.12: Use Case Diagram

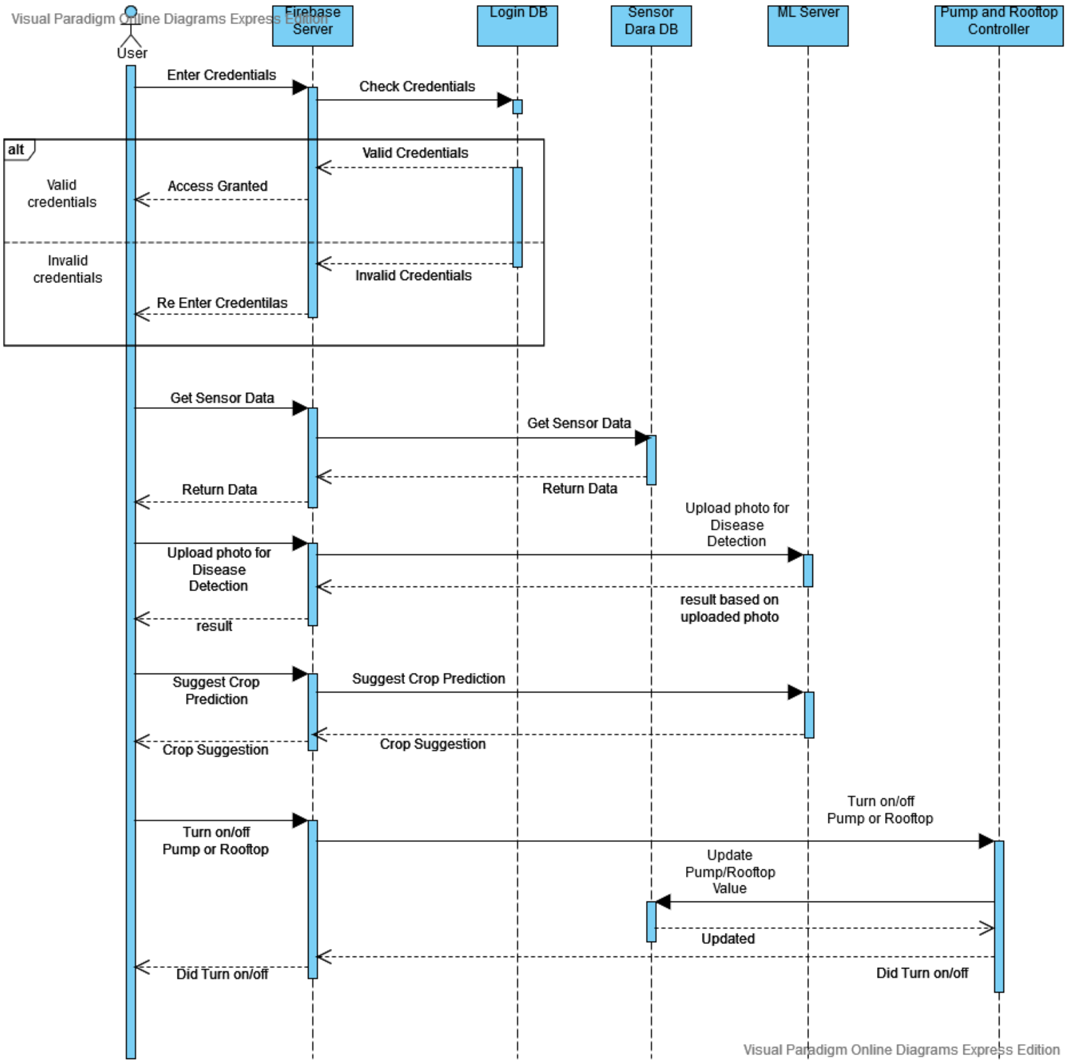

Figure 3.13: Sequence Diagram

3.4 Algorithms and Pseudo Code

For Crop Suggestion

Step 1: Start

Step 2: Read values from sensors

Step 3: Send that data to flask API

Step 4: Flask API will predict the crop which can be grown for maximum production

Step 5: Flask API return reply

Step 6: Display the results

Step 7: End

For Disease Detection

Step 1: Start

Step 2: Upload or take photos of a leaf of crop

Step 3: Send that image to flask API

Step 4: Flask API will predict whether the plant has a disease or not based on the given image

Step 5: Flask API return reply

Step 6: Display the result

Step 7: Upload that image with prediction result on firebase

Step 8: End

3.5 Testing Process

We have tested the android application on a various basis like it should not crash while feeding it with wrong data. Secondly, APIs should work properly for crop prediction and disease detection. If the server is not available then it should show that as well. The third point is data of the user should be in sync with the database and the farmer should able to see all real-time as well as previous some values of the sensors installed on the farm. Forth is whenever there is no data available of the farm sensors then it should not show the real-time sensor value page.

4 Results/Outputs

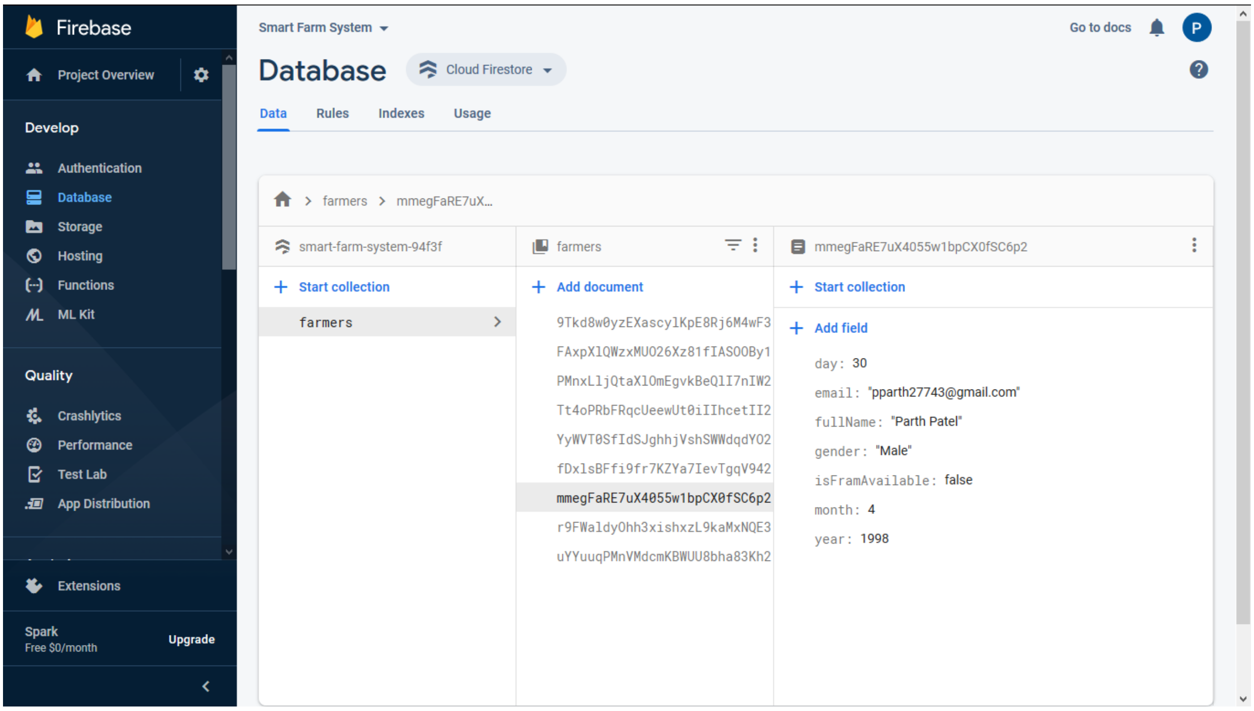

Figure 4.1: Farmers' Data

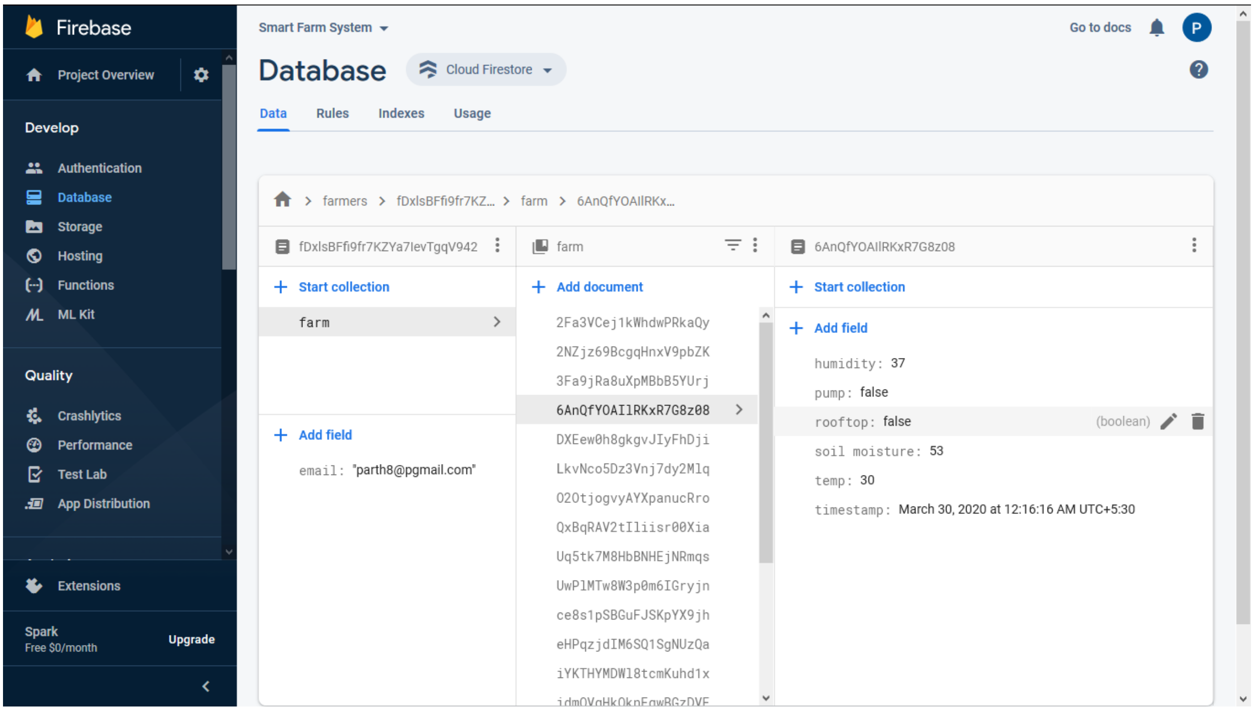

Figure 4.2: Sensors' data which are installed on farms with timestamps.

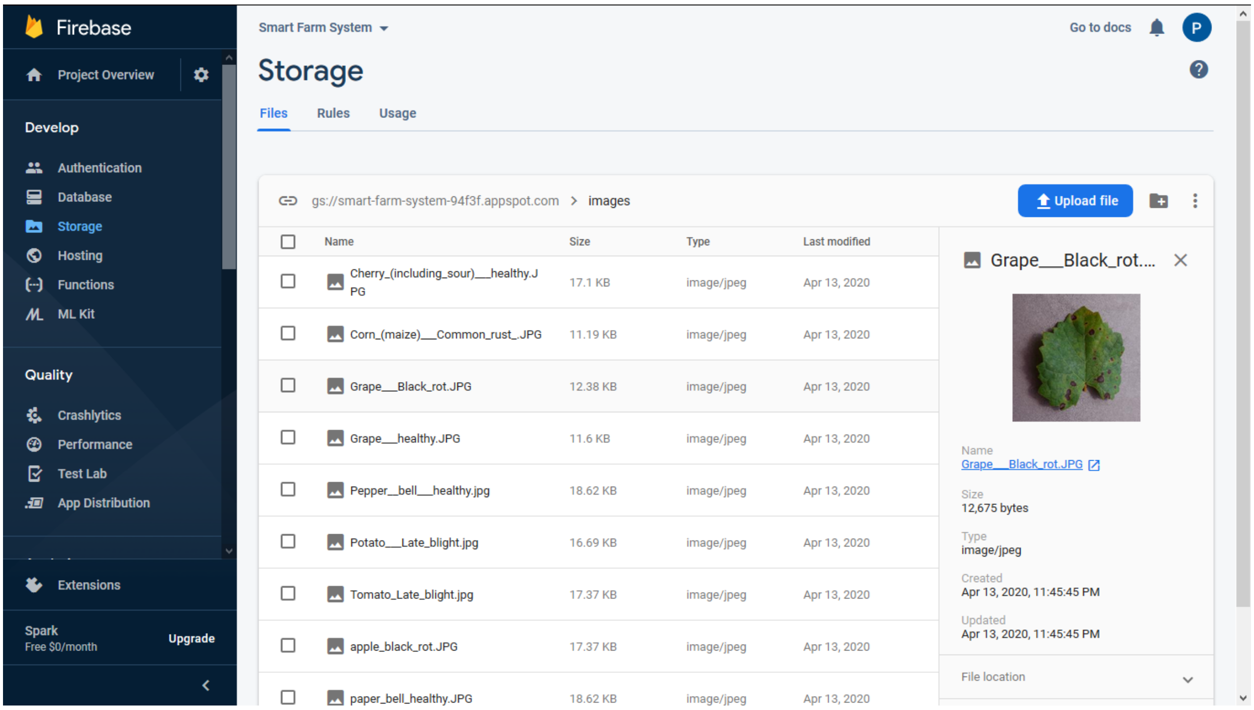

Figure 4.3: Disease Detection result stored on server storage.

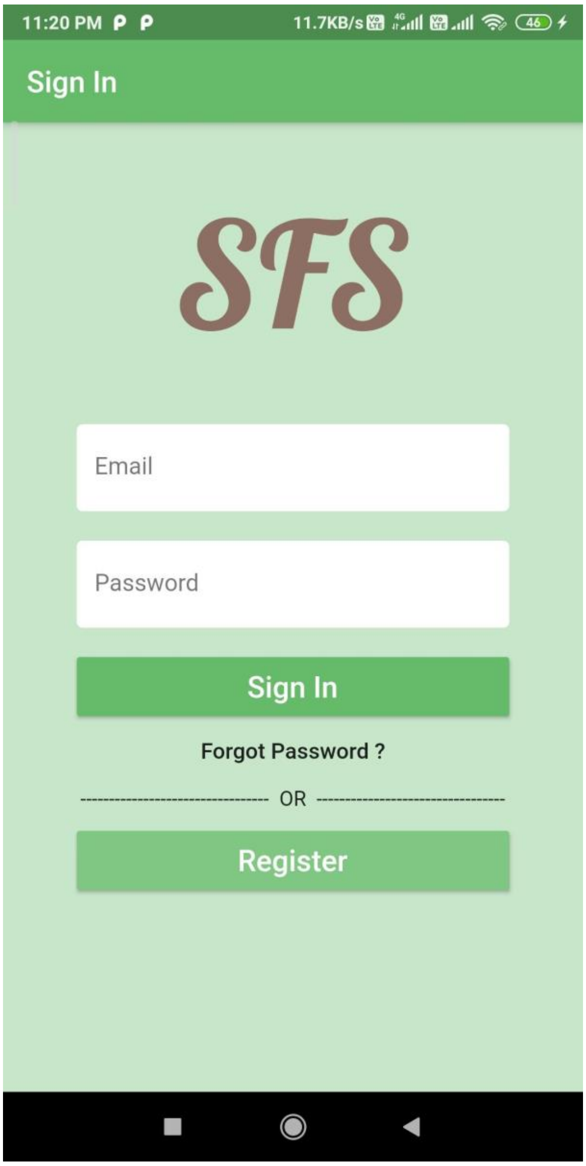

Figure 4.4: Application Login Page

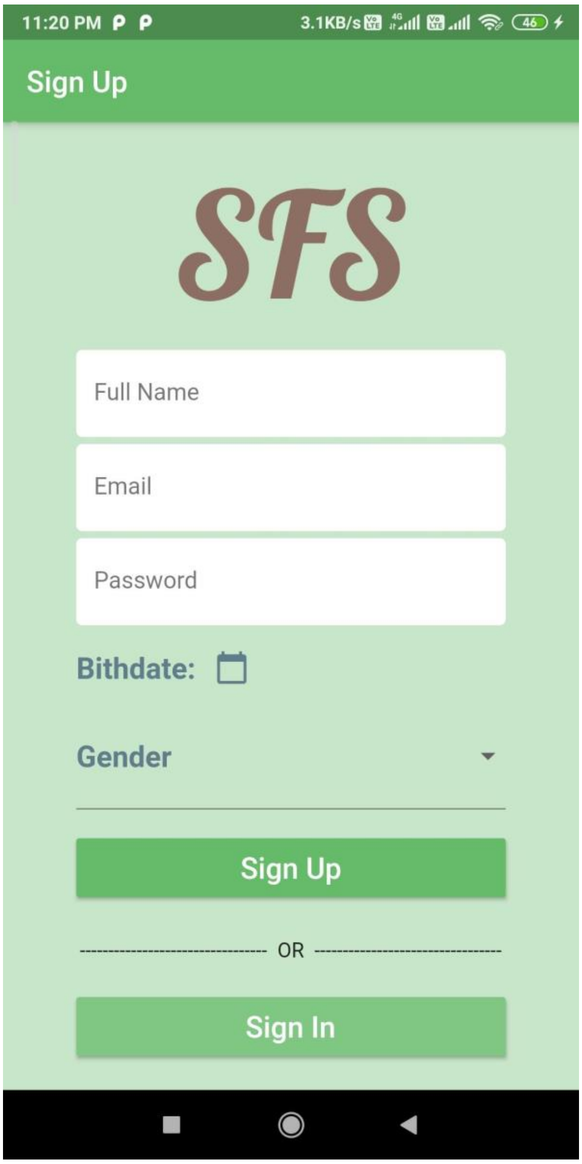

Figure 4.5: Application Signup Page

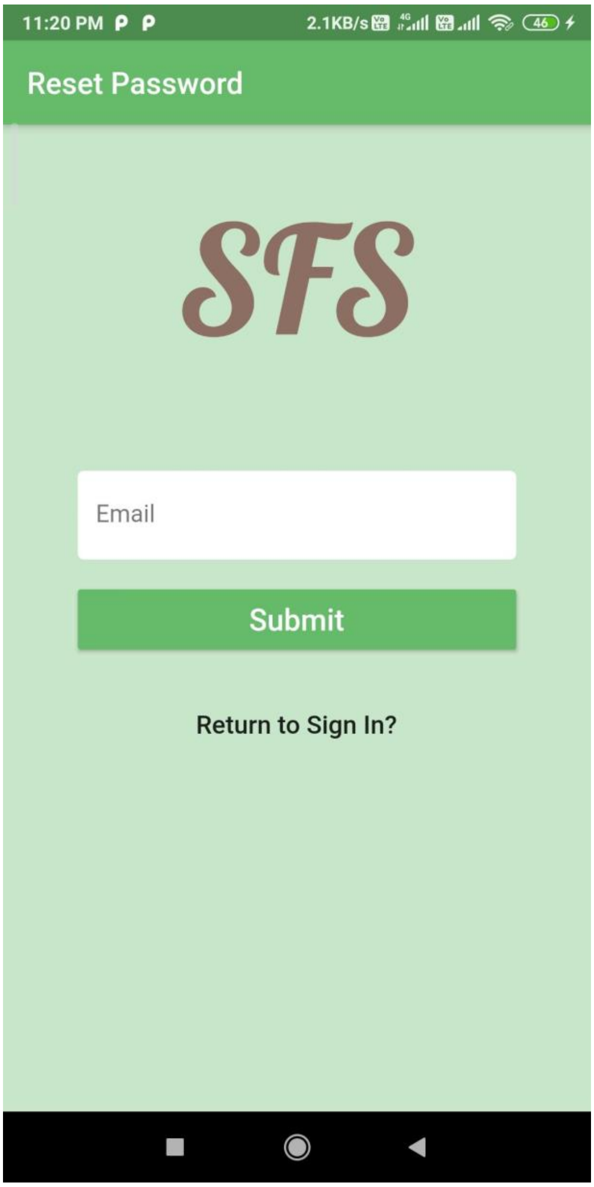

Figure 4.6: Application Reset Password Page

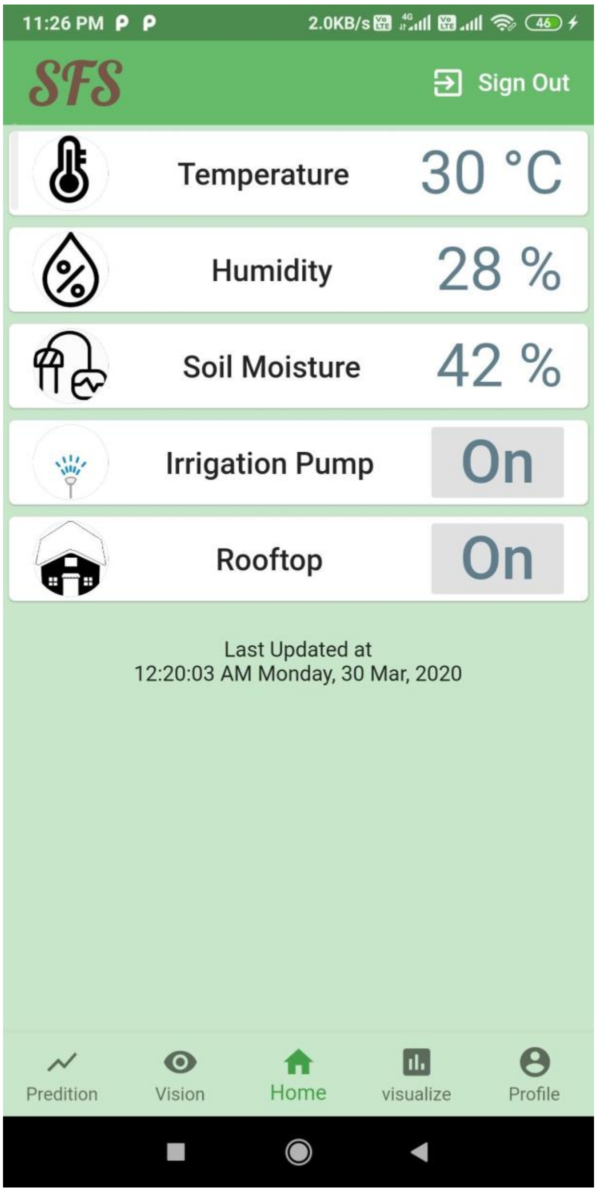

Figure 4.7: Application Home Page

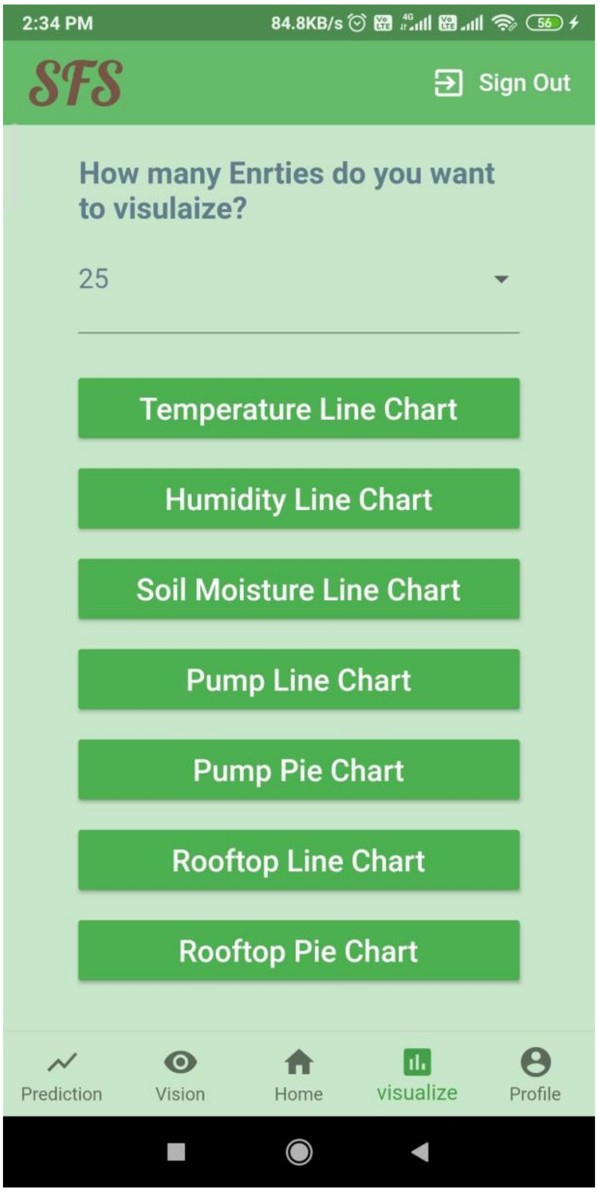

Figure 4.8: Application Charts

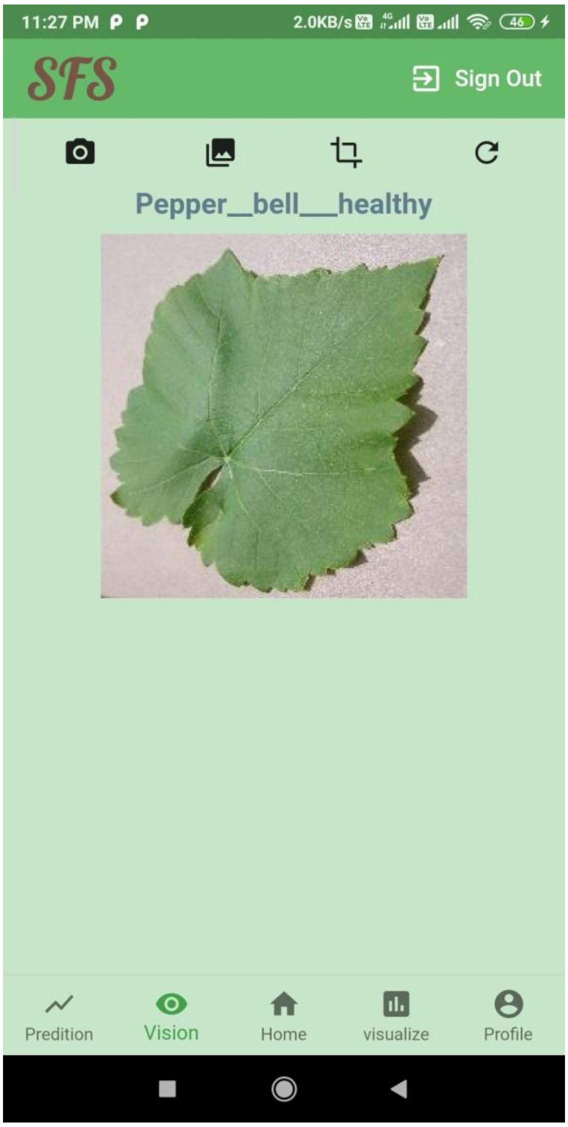

Figure 4.9: Application Vision Page

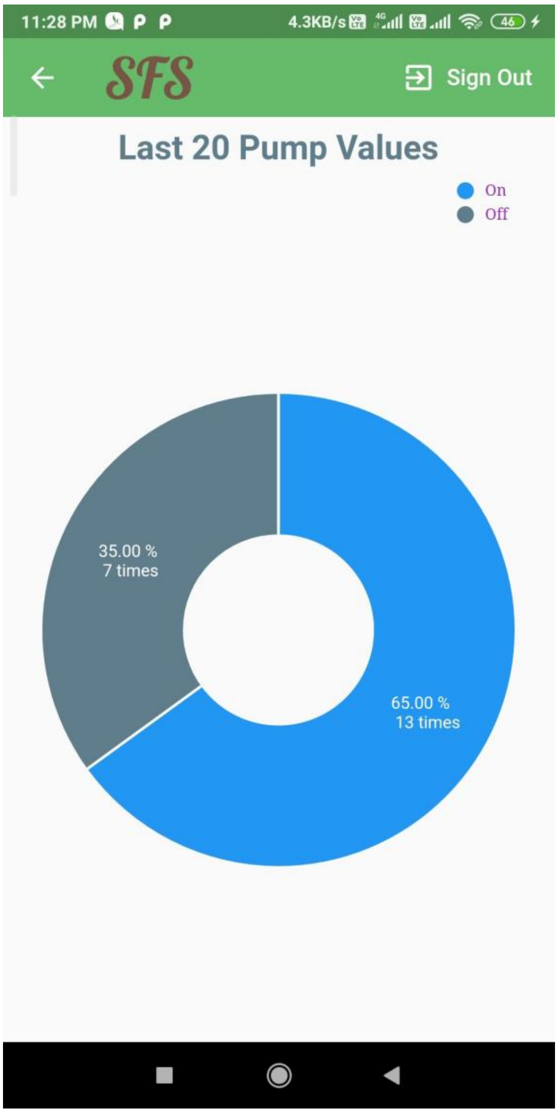

Figure 4.10: Application Pump Values Chart

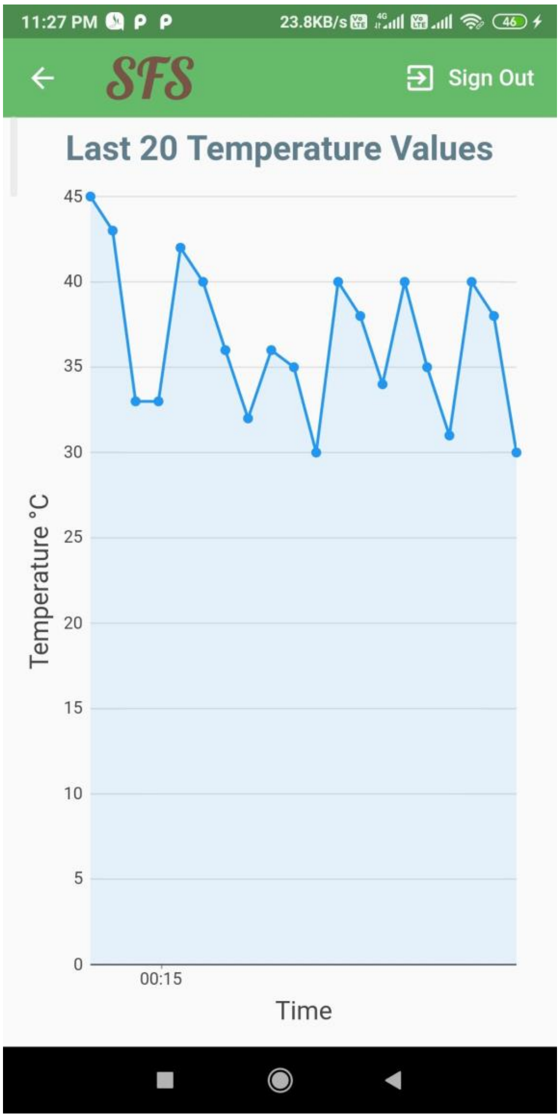

Figure 4.11: Application Temperature History

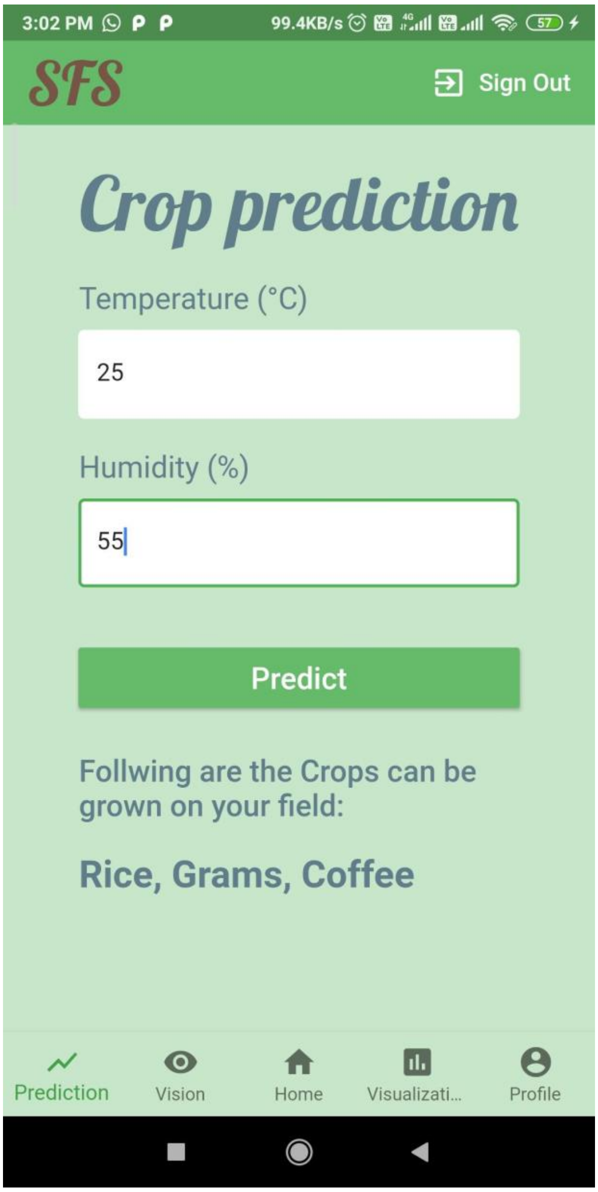

Figure 4.12: Application Crop Prediction

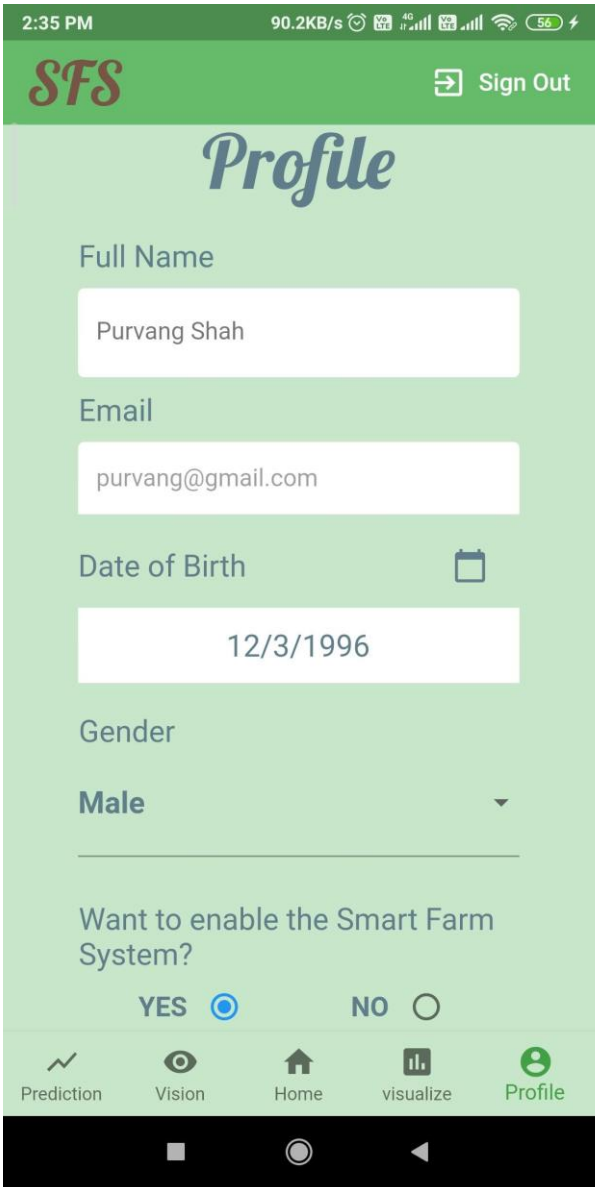

Figure 4.13: Application Profile Page

5 Conclusions/Recommendations

The focus on more intelligent, better, and progressively productive harvesting procedures are required to satisfy the increasing need of the food in the essence of the ever-contracting arable land. The entire system gives live monitoring of temperature humidity and Soil Moisture, predicts and gives suggestions of the best suitable crop according to the land and surroundings to the farmer, protects the crops automatically in circumstances beyond one's control, decreases the farmers work by functioning fully automatic that can also be controlled manually. Moreover, the application also detects certain types of diseases in the leaves that can damage crop production. For this purpose, wireless sensors, Cloud-computing, communication technologies, Big Data Solutions, and Machine learning algorithms are discussed thoroughly. It can be concluded that every inch of farmland is vital to maximize crop production. However, to deal with every inch accordingly, the use of sustainable IoT-based sensors and communication technologies is not optional; it is necessary.

Future works

Implementation of working rooftop system and irrigation system is remaining due to unavailability of the components. Drone images or surveillance camera images can be used instead of manually clicking photos for leaf disease detection. Moreover the website of the application can be built.

6 References

1. “Precision Farming: 10 Companies to Watch in 2018.” *Disruptor Daily*, 9 Mar. 2018, www.disruptordaily.com/precision-farming-10-companies-watch-2018/.
2. Wolfert, Sjaak, et al. "Big data in smart farming–a review." Agricultural Systems 153(2017): 69-80.
3. “Smart Farm Technologies Are Here in India, but Available to Just a Few.” *Hindustan Times*, 6 Aug. 2018, www.hindustantimes.com/india-news/smart-farm-technologies-are-here-in-india-but-available-to-just-a-few/story-XcJwjq6w8w3kncw43iTKBN.html.
4. Image of Arduino mega 2560, Retrieved from Arduino: https://store.arduino.cc/usa/mega-2560-r3
5. LDR-light-dependent-resistor (n.d.) Retrieved from https://5.imimg.com/data5/LF/JE/MY-21614932/ldr-light-dependent-resistor-500x500.jpg
6. DHT22 Sensor image Retrieved from https://robu.in/wp-content/uploads/2017/06/robu-4-7-768x768.jpg
7. MLX90614 temperature sensor for arduino image Retrieved From https://storage.googleapis.com/stateless-www-faranux-com/2018/02/f90ec438-mlx90614-contactless-temperature-sensor-module-for-arduino.jpg
8. ROBODO Rain drop sensor module image Retrieved from https://5.imimg.com/data5/VV/VB/MY-9380557/rain-drop-sensor-500x500.jpg
9. ESP8266 Serial wireless module nodemcu wifi internet image Retrieved from https://5.imimg.com/data5/DW/KO/MY-43948449/esp8266-serial-wireless-module-nodemcu-v3-lua-wifi-internet-500x500.jpg
10. Soil moisture sensor for arduino image Retrieved from https://images-na.ssl-images-amazon.com/images/I/51CEr20GRdL._SX425_.jpg