Module Code & Module Title
CS5068NI– Cloud Computing & IoT
Smart Farm

Assessment Type
10% Proposal Report / 50% Group Report

Semester
2024 Spring

Group Members

London Met ID	Student Name
23047335	David Ramovich Mandal
23047339	Srina Mahat
23047330	Shubham Lamichanne
23047555	Saurav Timilsina
23048517	Rahul Uprety
23047523	Hriday Bikram Kand

Assignment Due Date: 15th May 2025
Assignment Submission Date: 15th May 2025
Submitted to: Mr. Sugat Man Shakya
Word Count: 3313

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

Document Details

Submission ID

trn:oid:::3618:95992605

Submission Date

May 15, 2025, 11:49 AM GMT+5:45

Download Date

May 15, 2025, 11:54 AM GMT+5:45

File Name

Document 2.docx

File Size

21.5 KB

25 Pages

3,313 Words

17,837 Characters

10% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Match Groups

- **39** Not Cited or Quoted 10%
  Matches with neither in-text citation nor quotation marks
- **0** Missing Quotations 0%
  Matches that are still very similar to source material
- **0** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

2%	Internet sources
2%	Publications
8%	Submitted works (Student Papers)

Acknowledgement

We would like to thank our module leader, Mr. Sugat Man Shakya and tutor Mr. Ayush Bhakta Pradhanang for his selfless and consistent support as well as his guidance from this project inception. His valuable and constructive criticism and suggestions helped us to shape where we're going and his words of encouragement have been invaluable in driving us forward. We are also extremely thankful that he devoted his time and dedication to this work.

We also want to thank Islington College for giving us the resources, infrastructure and academic environment that helped us to finish this project. The college has given us the support and opportunities to test and implement new cutting-edge technologies in the work that we do. Without the encouragement and technical tools available to them, the college could not have allowed this project to happen.

We also want to express our gratitude to our team who committed hard work and worked with their best efforts. This project was one which every person who played a role in it, from research and design through to implementation and testing, contributed greatly. This project was successful and rewarding because they were able to work together in order to solve the problems at hand, as well as dedicate themselves to the success of the same.

Abstract

Agriculture forms a key part of Nepal’s economy, employing a large proportion of Nepal’s population and contributing much to its GDP. And yet conventional farming practices still dominate the sector, often resulting in unproductive yields, unpredictable crop yields and mismanagement of resources. In a country in which most farming is heavily dependent on adverse weather and manual labour, it is vital that smart, sustainable and technology-based solutions be developed which can help growers improve plant reproduction yield and conservation of resources. A new and innovative solution will solve these problems, and it's titled "Smart Farm" (cloud computing using the Internet of things) based on integration of sensors, automated irrigation for agricultural crops, real time environmental monitoring and data analytics that make it possible for farmers to make informed decisions, maximizing their water consumption and improving crop health. Using sensors such as soil moisture sensors, DHT11 temperature/humidity sensors and water pumps connected to Arduino boards the system will automate and optimize the farming process. By applying the technology on local conditions in the country’s agriculture context (mainly rural areas where there are low resources and farming is labour intensive), it demonstrates how innovation can contribute to food security and sustainability; ultimately the project is aimed at building a framework to scale up technologies that can empower local farmers and lower their dependence on manual work and encourage technology based agriculture across the country.

Table of Contents

Table of Figures

Table of tables :

1. Introduction

The Internet of Things (IoT) is a rapidly growing technology in the field of IT that plays a vital role in different industries by enabling smart solutions. The system finds multiple uses throughout different sectors to develop useful intelligent applications. Through IoT we are experiencing many changes in both daily living and work practices which further serve the goals of technology and society and economic development. The IoT system in smart agriculture features internet-enabled sensors with microcontrollers that function autonomously. Farmers can collect important data through the system which enables them to make decisions for automation purposes to grow additional crops. The experts has predicted the world economy will benefit from more than $11 trillion dollars from IoT devices reaching over 100 billion devices by 2025 since they believe this technology has tremendous agricultural potential.

The **Smart Farm** project based on Internet of Things seeks to resolve agricultural challenges by providing farmers with efficient and secure farming management systems. This system allows farmers to track soil temperatures together with soil moisture levels and additional critical parameters. The system depends on various devices encompassing sensors together with microcontrollers and others for monitoring and controlling agricultural processes. The internet connection on these devices enables farmers to monitor real-time data which lets them decide from any faraway location.

1.1 Current Scenario

Agriculture is a traditional income source for many Nepalese with more than 61% of the population engaged in agricultural production. Approximately 24 to 27% of Nepal's Gross Domestic Product (GDP) comes from agriculture reflecting the large disparity between the amount of labor put into crops and the economic return on investment. One of the major constraints is the lack of efficient farming practices which still depend heavily on manual labor largely due to limited rural access to technology and more effective, simpler technologies that could enhance production levels and save farmers' labor costs. At the end of 2023 only about 33% of Nepal's cultivated land was used for farming and only 40% of all an area irrigated had access to water through the year. Farmers rely heavily on monsoon rainfall which has become increasingly variable because of climate change affecting production. Reports have also suggested that climate change could dramatically reduce wheat yields in western Nepal by the 2020s proving how much we must do to adapt to climate change, which is one of the objectives of the project.

In addition, nearly half of Nepal's farmers are smallholders who lack access to markets and finance, Internet access and technical training to access or operate modern farming systems particularly in rural regions where agriculture is the dominant occupation. Even after government initiatives to boost the adoption of technology (such as the Agriculture Development Strategy (2015–2035), the extent to which smart farming tools have been integrated in farmer practices on the ground is relatively low. Hence, there is a huge opportunity for accessible, cost-effective solutions tailored to the grassroots agricultural landscape of Nepal.

1.2 Problem Statement and Project as a Solution

While there has been considerable emphasis in Nepalese government policy on upgrading agricultural infrastructure, most of Nepal's farmers have remained very reliant on traditional hand harvesting practices, whose automation is inefficient, time-consuming, and inconvenient for workers. Some of the major issues cited are lack of timely environmental information and difficulties in anticipating adverse weather conditions. Insufficient information on the soil's moisture level can result in excessive consumption or stress of plant and crop resources which has consequences for yield quality and quantity.Moreover, farmers can spend too much time performing repetitive activities such as hand watering or monitoring temperature or controlling pests. Without accurate and timely forecasting based on data, uneven harvests and restrictions to small scale farming continue to persist. Analysis of rainfall patterns and climate change cycles over a century ago highlight that farmers must respond quickly to climate change.

The market of smart farming is growing due to the rising demand for food , thanks to the rising population and availability of advanced of technology. The adoption of technologies such as Internet of Things (IoT) and AI is driving the growth of the market. According to a report by market.us , the smart farming market is projected to surpass USD 53 billion by 2032 (GlobeNewswire, 2023).

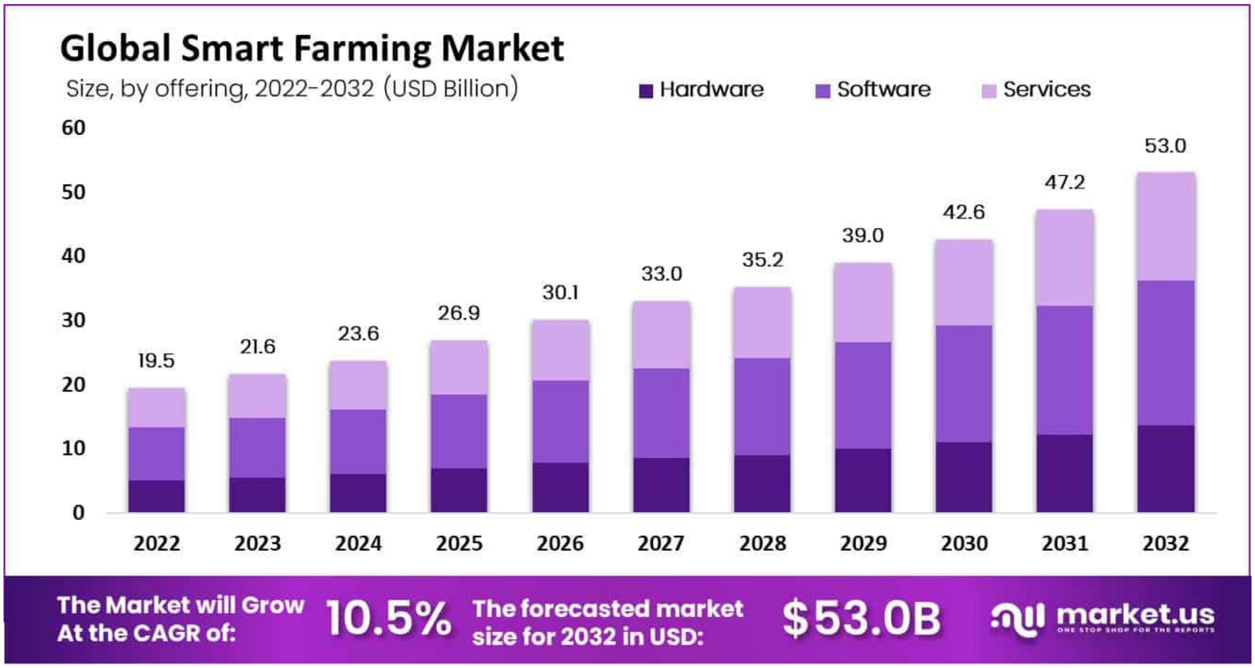

Figure 1: Smart Farming Market Prediction

We have developed a project called Smart Farm to directly address the problems mentioned above by adopting Internet of Things (IoT) technology and cloud computing into farming practices. Smart Farm comprised low-cost sensors to measure the soil moisture, temperature and humidity where a Arduino based controller automates irrigation according to real-time data and a cloud platform that stores and analyses environmental information. Components like Arduino boards, DHT11 sensors and basic relay modules are cheap and easy to buy in Nepali market. Thus, it is very suitable in resource-poor setting. By automating these critical processes and giving farmers real-time access to information about their fields, Smart Farm promotes better precision and eliminates the need for guess work and excessive manual labour. It also provides a cost-effective, sustainable and user-friendly solution to smallholder farmers, particularly in rural districts across Nepal where access to digital technologies is especially limited. So Smart Farm not only provides a practical response to present agricultural challenges but also fits Nepal’s long term goal of climate resilience, data-driven and inclusive growth in agriculture.

1.3 Aim and Objectives

1.3.1 Aim

The aim of this project is to design a smart farming system using IoT that features automated irrigation, smart door, and smart lighting to enhance both productivity and security.

1.3.2 Objectives

- Utilize IoT sensors for real-time monitoring of soil moisture, temperature, and other environmental conditions.
- Implement an automated irrigation system that activates based on current soil moisture levels.
- Install remotely controlled smart doors for enhanced farm security.
- Set up a smart lighting system driven by motion detection and ambient light levels for energy efficiency and safety.
- Develop a centralized platform for collecting, storing, and visualizing data from all IoT devices to support effective monitoring and decision-making.

2. Background

2.1 System Overview

Smart Farm system is a technology driven approach that combines Internet of Things (loT) hardware, automation techniques and cloud computing to maximize efficiency in farming practices. The general goal is to remotely automate irrigation based on real-time environmental data to allow farmers to make accurate decisions without constant manual supervision.

The system runs at the bottom of the stack and is handled by an Arduino microcontroller that acts as the main processor and receives data from all the sensors in the field. The sensors are a soil moisture sensor, a DHT11 temperature and humidity sensor light and sensor. But once the soil moisture drops below a specified threshold, it prompts the relay module to activate a water pump, which automatically starts irrigation. After the soil reaches the desired moisture level, it shuts off the pump, saving water and avoiding overwatering of plants. All the sensor data are stored and analysed in the cloud services, allowing remote access and historical observation. At the farm level, farmers or users have access to a simple web (or mobile) dashboard of the farm area that displays temperature, humidity and soil conditions in real time. Through this data-based approach, not only can irrigation be automated but also the agricultural activities are forecast and planned more well.

2.2 Design Diagrams

2.2.1 System Architecture

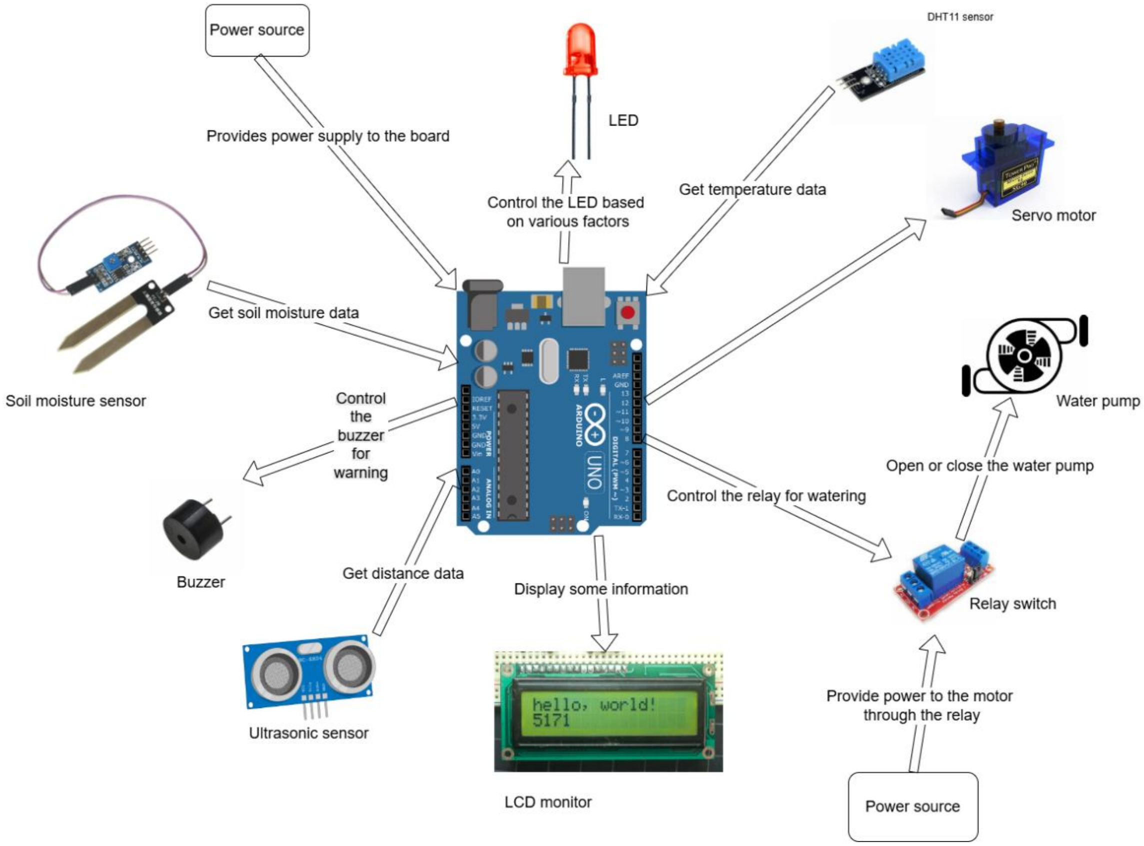

Figure 2: System Architecture

2.2.2 Flowchart

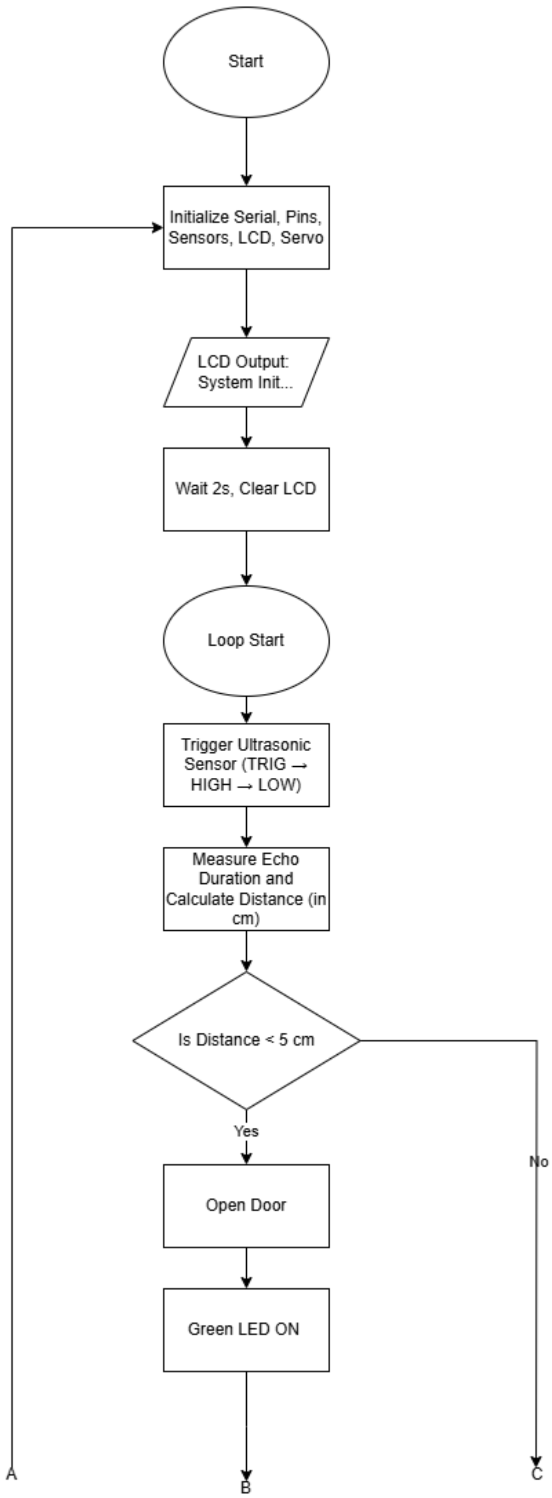

Figure 3: Flowhchart-1

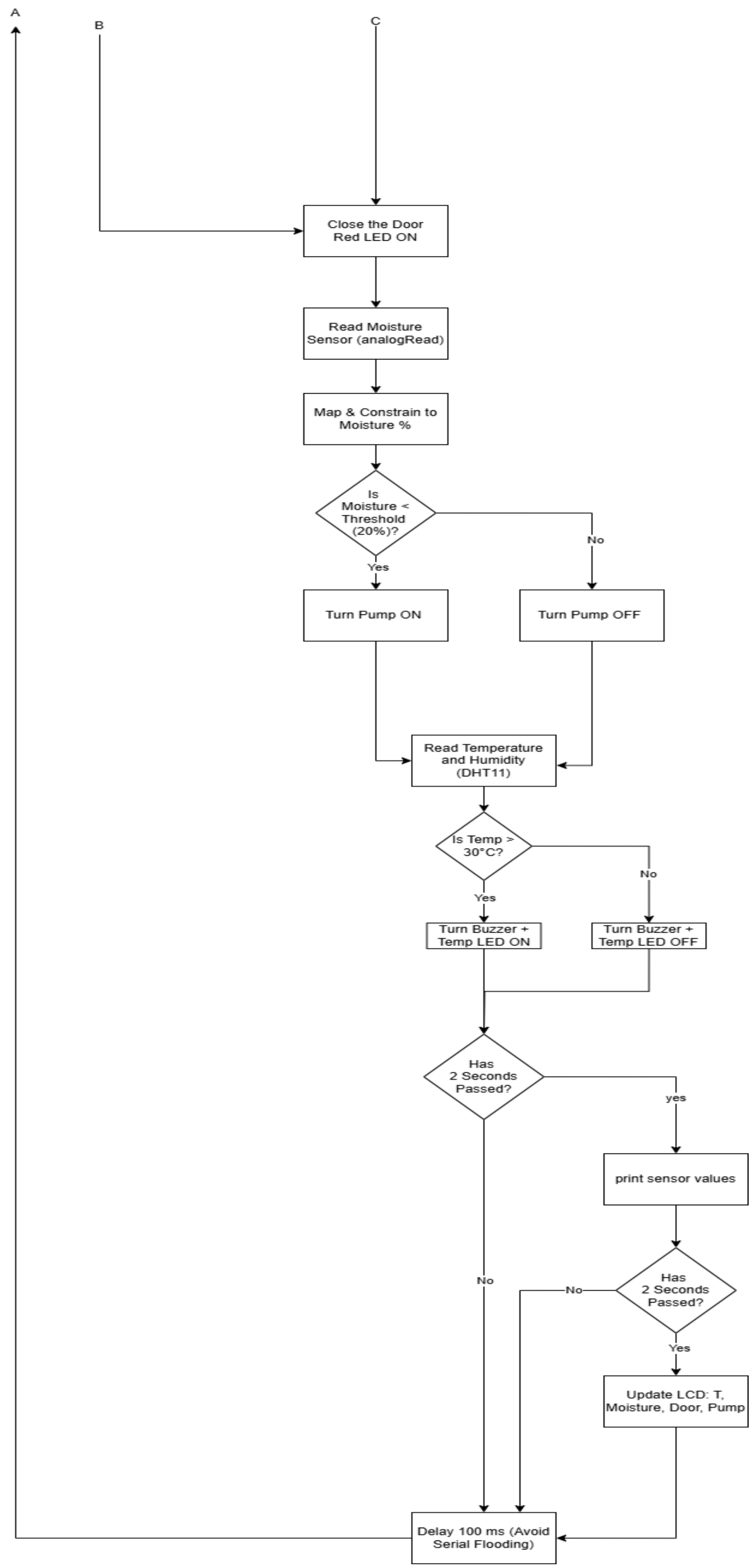

Figure 4: Flowchart-2

2.2.3 Circuit Diagram

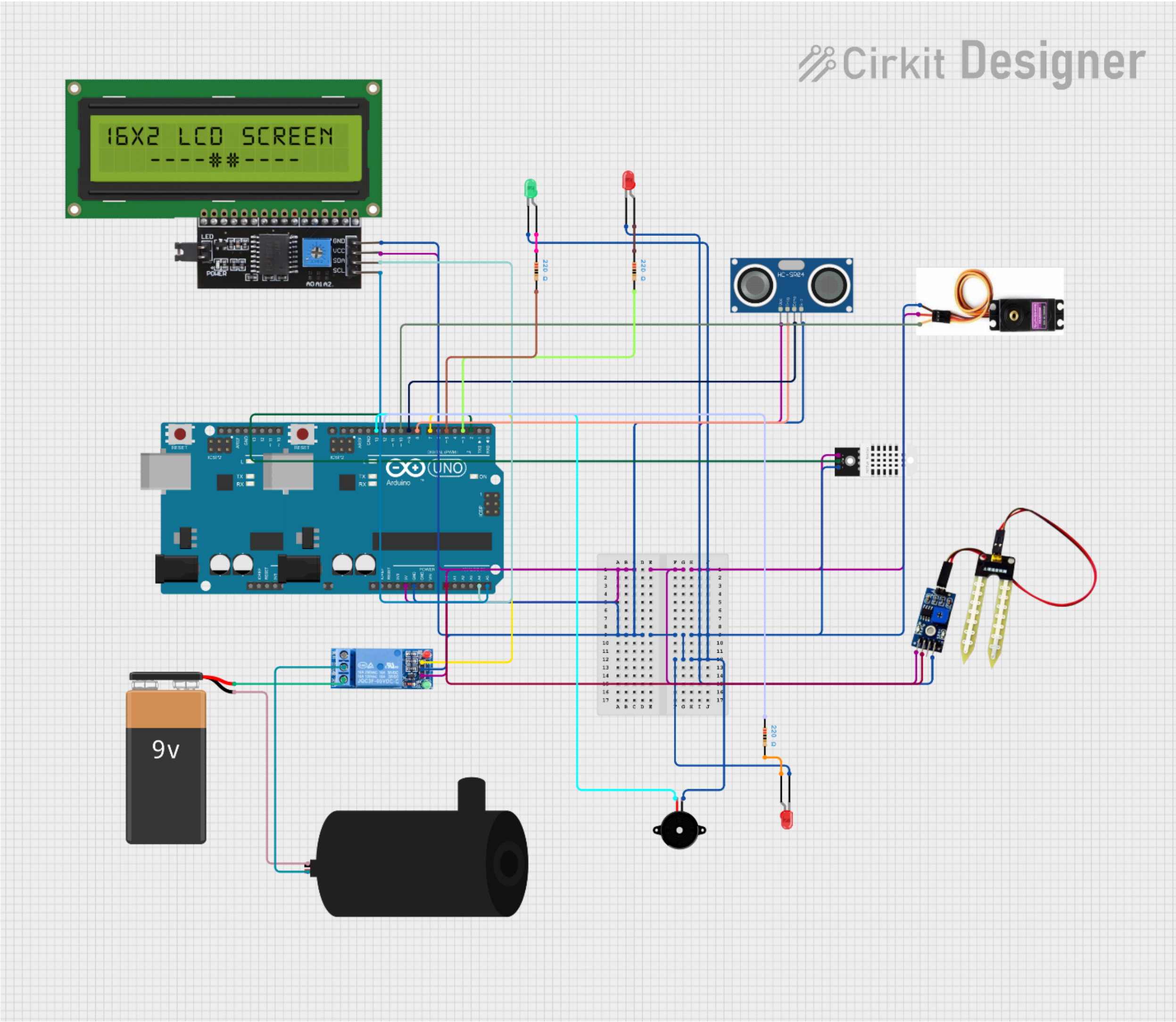

Figure 5:Circuit Diagram

2.2.4 Block diagram

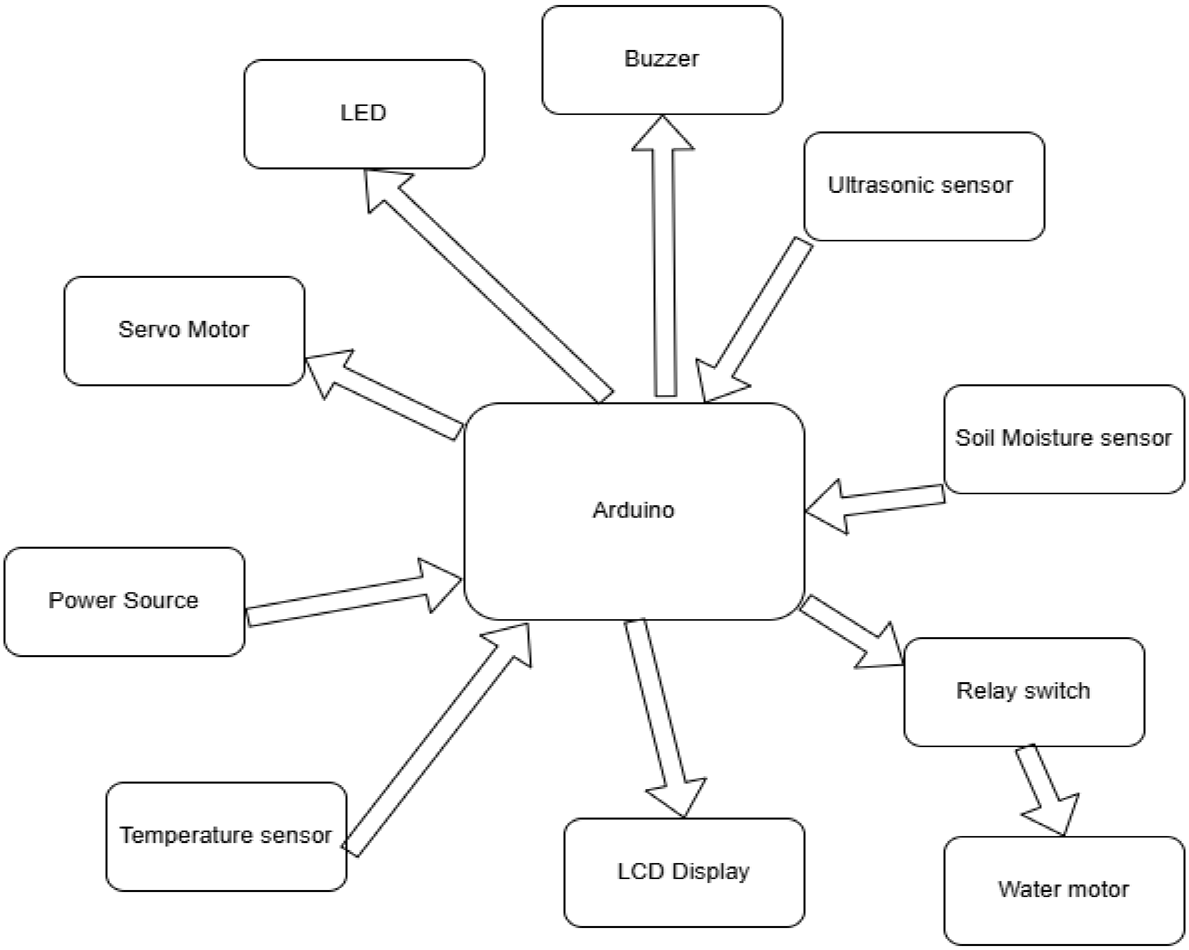

Figure 6: Block Diagram

2.2.5 Schematics

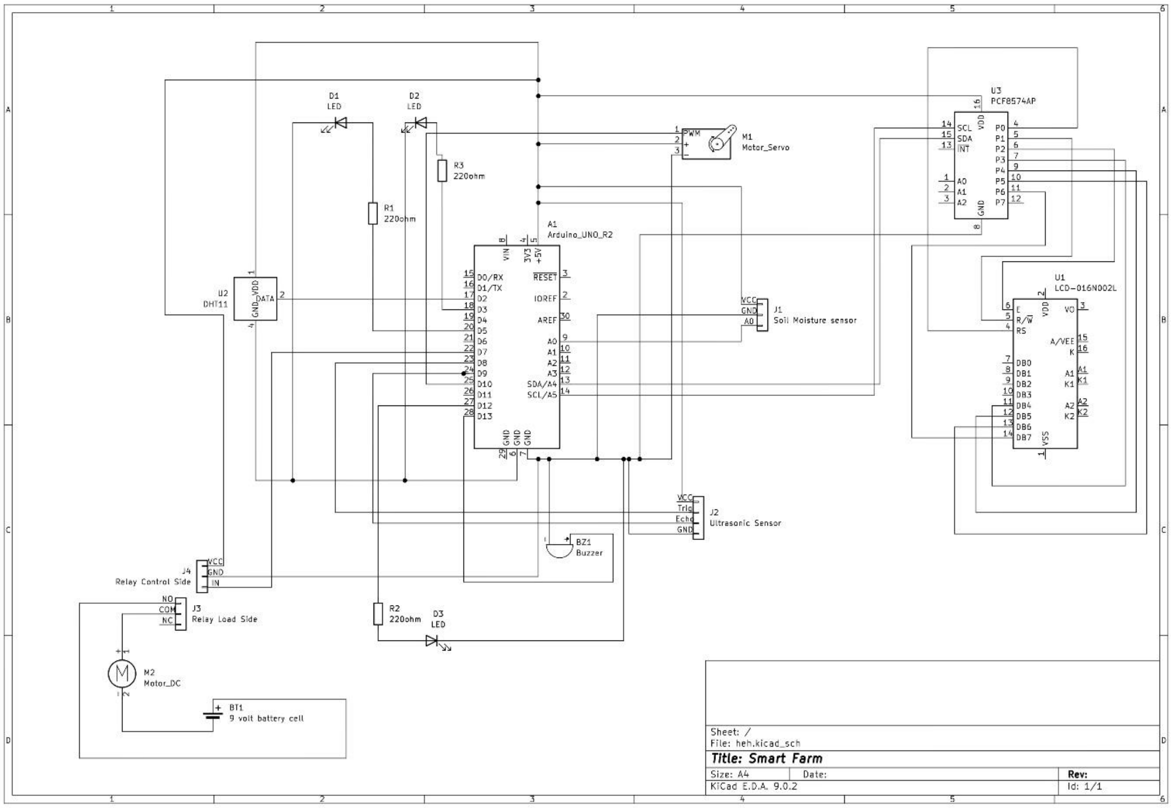

Figure 7: Schematic Diagram

2.3 Requirement Analysis

2.3.1 Hardware Components

The description of the core hardware components used in the report is in the components part in the appendix section.

- Arduino Board
- Servo Motors
- Ultrasonic Sensor
- Soil Moisture Sensor
- Water Pump
- LCD Display
- Resistors
- Jumper Wire
- Breadboard
- LED
- Buzzer
- DHT11
- Battery

2.3.2 Software Components

The description of the software components used in the report is in the components part in the appendix section.

- Arduino IDE
- Microsoft Word
- Kicad
- Draw.io

3. Development

3.1 Resource Collection

We obtained components required for building the Smart Farm prototype by acquiring items from our college's IT Resource Department along with purchasing them from local electronics suppliers. We submitted a formal request letter to the department to obtain available hardware. The resources collected from the IT Resource Department and local electronics suppliers are:

- Arduino Uno
- Breadboard
- Jumper Wires
- Relay Module
- Servo Motor
- I2C LCD Display
- Soil Moisture Sensor (Analog Type)
- DHT11 Temperature and Humidity Sensor
- HC-SR04 Ultrasonic Sensor
- Additional jumper wires and resistors if required
- LED light
- Buzzer
- Resistors
- Battery 9 volt
- Water motor

The equipment components underwent full compatibility tests before operation. The advance collection of required materials created conditions for smooth hardware setup and testing operations.

3.2 System Development

Step 1: Connecting the soil moisture sensor

This step involved linking the VCC pin to the Arduino’s 5V and GND pin to ground while using analog pin A0 for output. The VCC pin of the soil moisture sensor received power from the Arduino's 5V pin, while its GND pin joined the Arduino GND and its analog output connected to analog pin A0. The connection allowed us to obtain precise reading measurements from the soil moisture levels.

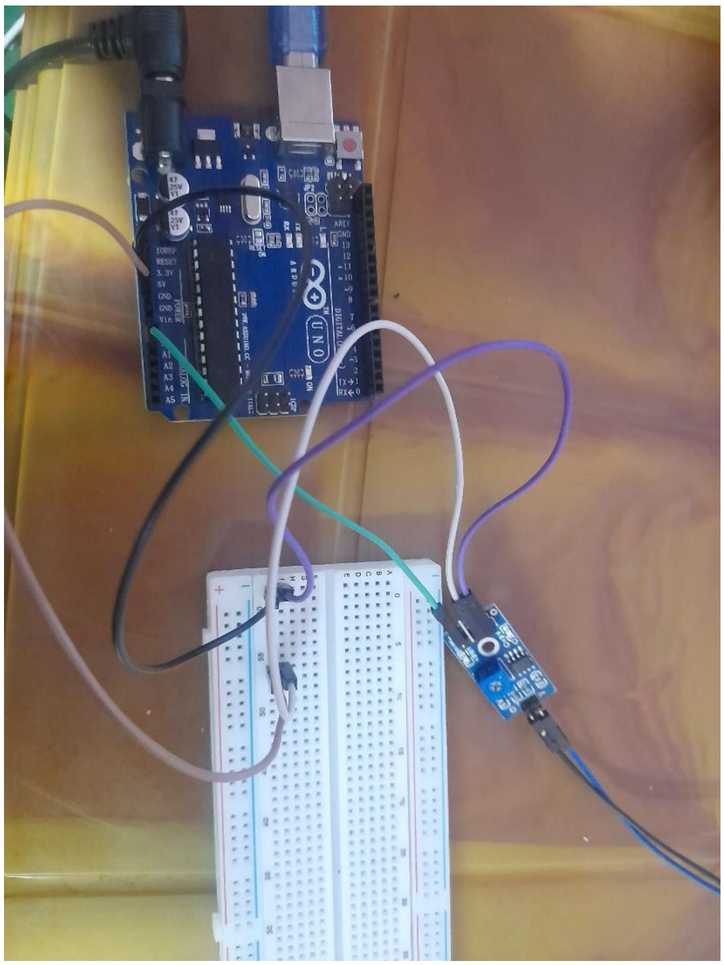

Figure 8: Step 1

Step 2: Connecting the DHT11 Sensor

The VCC of DHT11 sensor received power from 5V while connecting its GND to GND and DATA to digital pin D2. The device functioned as an environmental data collecting system which plays a vital role in understanding plant health.

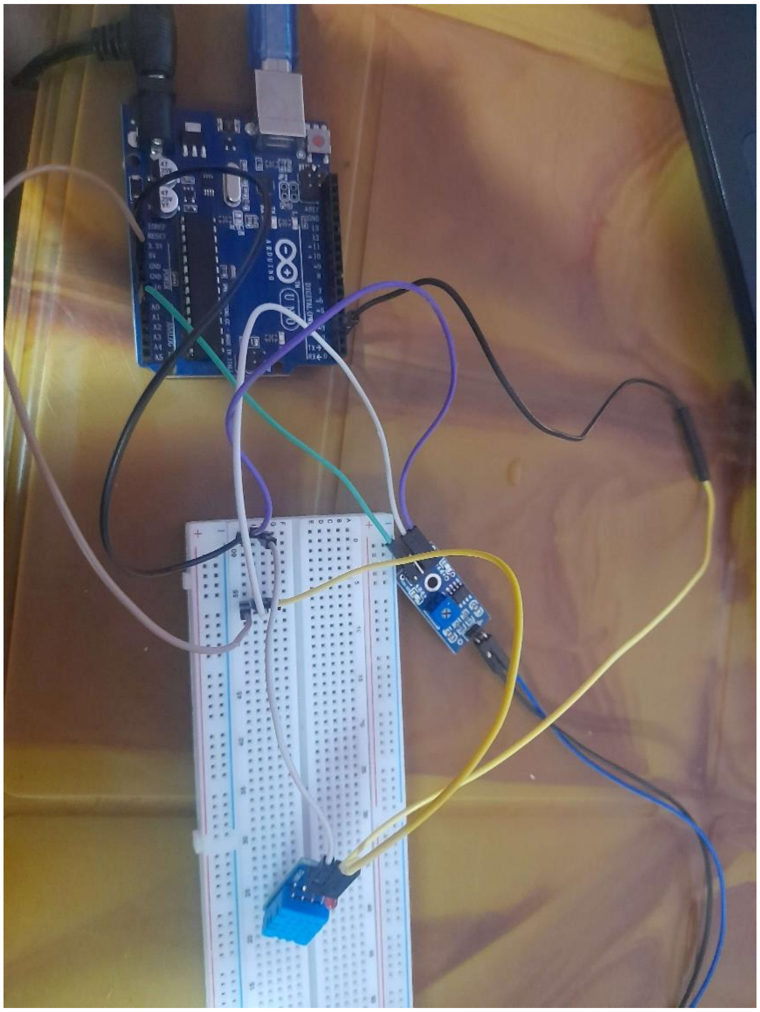

Figure 9: Step 2

Step 3: Connecting the Ultrasonic Sensor (HC-SR04)

The ultrasonic sensor received its VCC from 5V while GND went to GND and TRIG connected to D8 and ECHO went to D9. The sensor operated to identify nearby objects within its range.

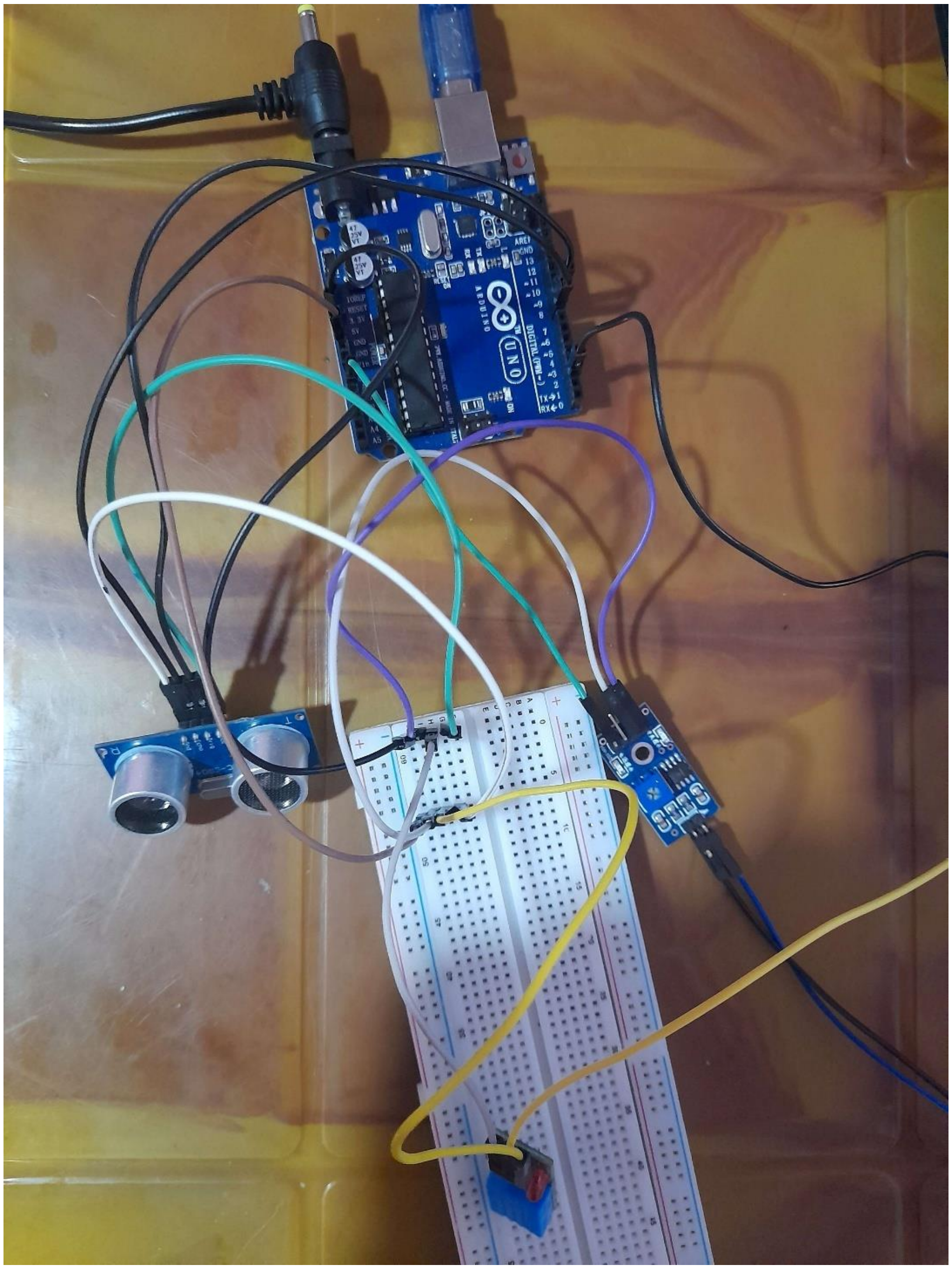

Figure 10: Step 3

Step 4: Connecting the Servo Motor

The servo motor received its power source from 5V at its red wire (VCC) along with its brown wire (GND) connected to GND and its orange wire (signal) linked to digital pin D10. The servo would turn according to sensor proximity measurement.

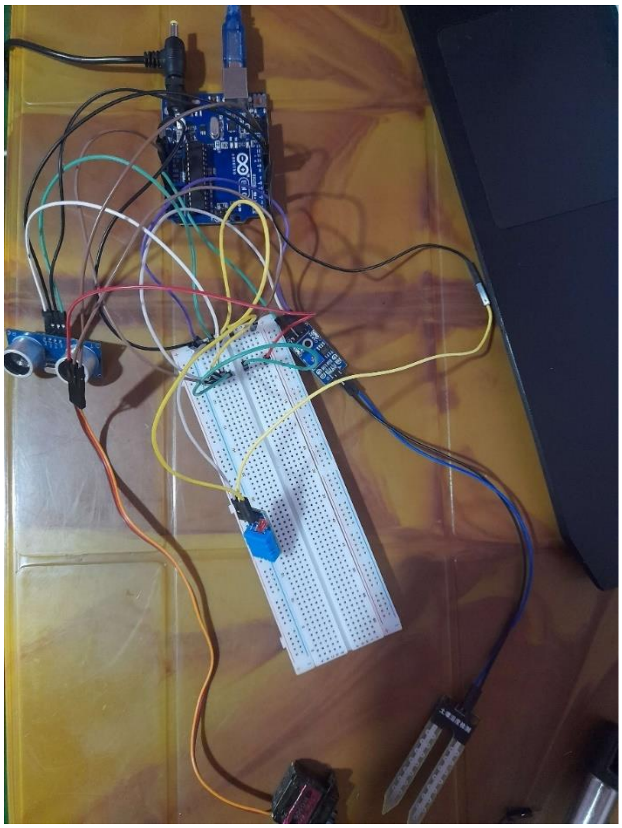

Figure 11: Step 4

Step 5: Connecting the Relay Module

The one-channel relay acted through the Arduino device to regulate power output for attached external devices including pumps or fans. The connection between the VCC pin of the relay module and 5V power was established along with connecting GND to GND and the IN control to D7 on the Arduino.

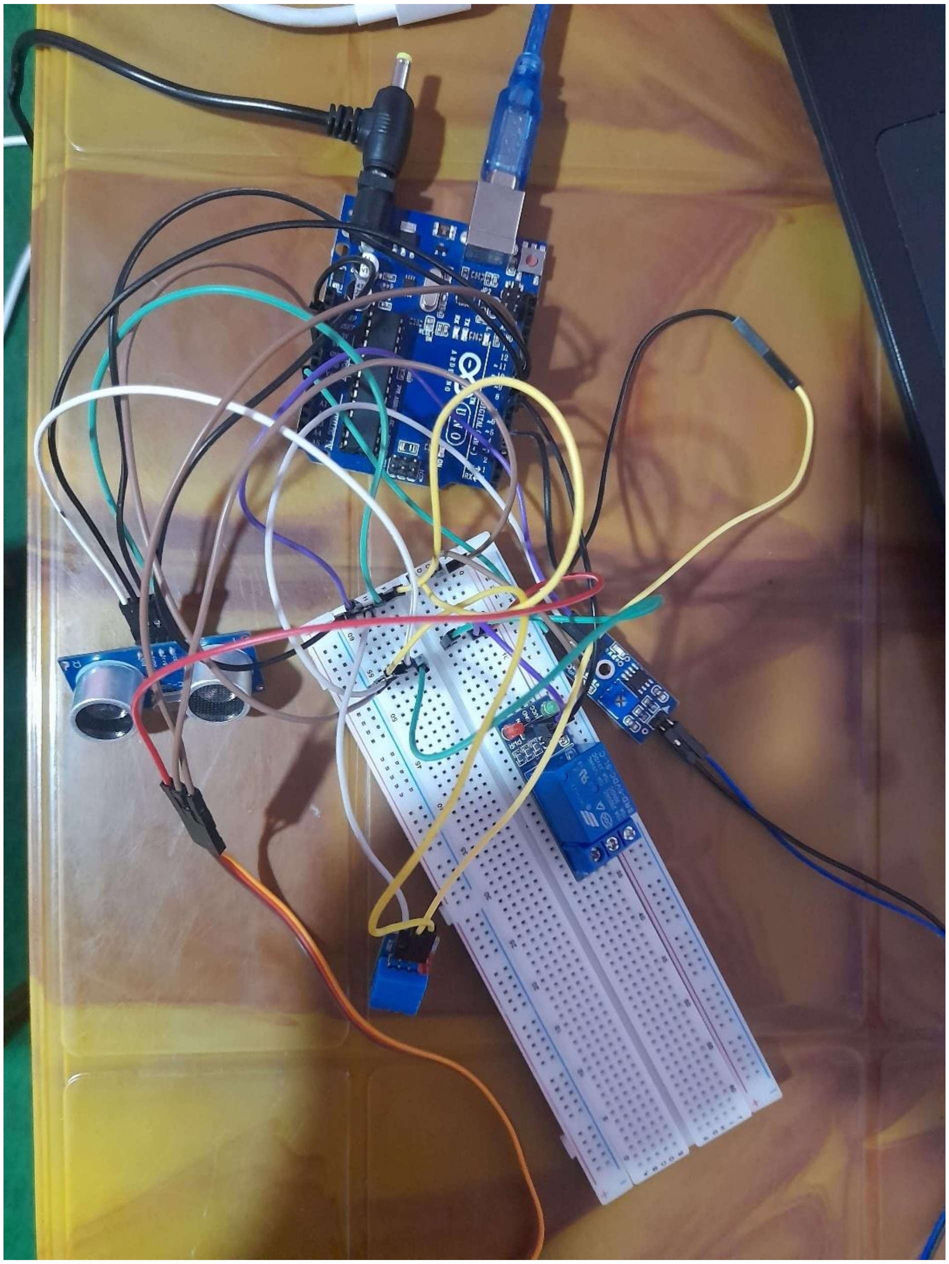

Figure 12: Step 5

Step 6: Connecting the I2C LCD Display

The I2C LCD displayed in real-time the readings from sensor measurements. We connected GND of LCD to GND while VCC went to 5V and SDA attached to A4 analog pin and SCL directed to A5.

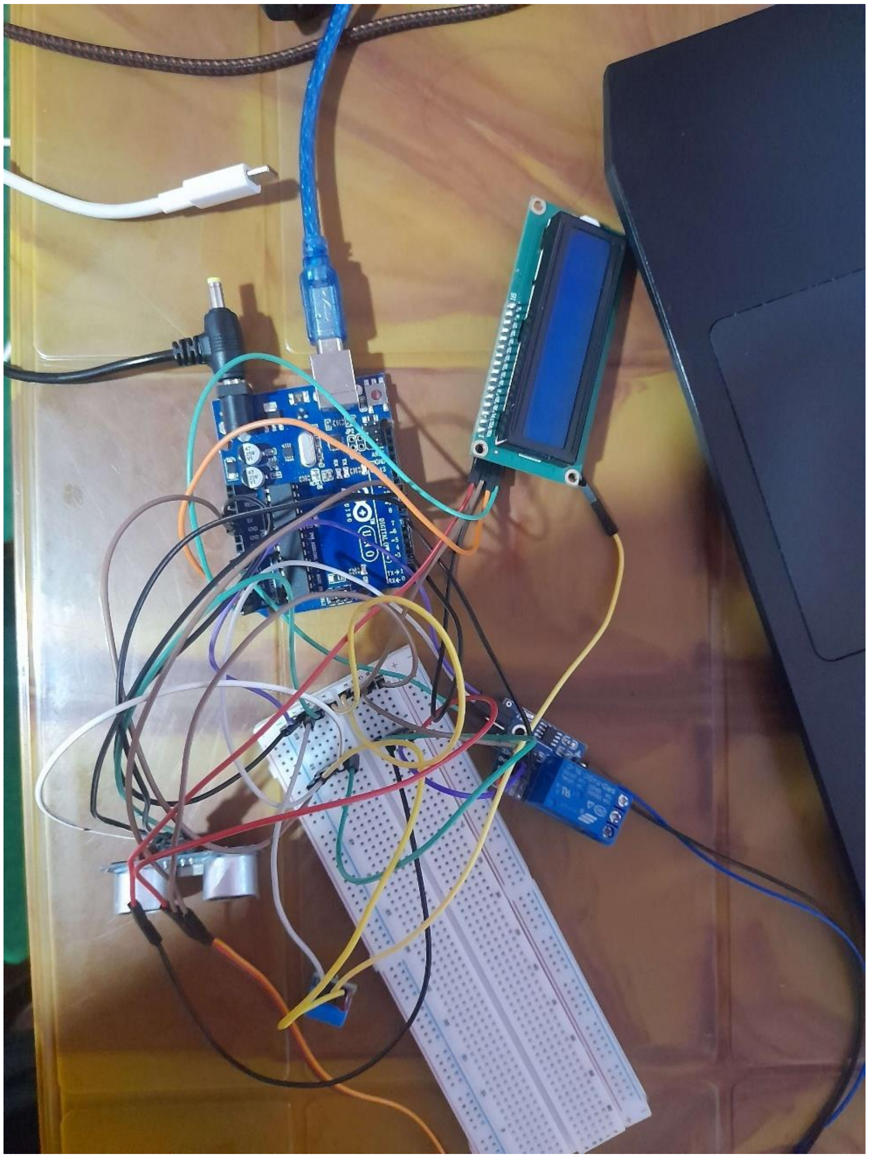

Figure 13: Step 6

Step 7 : Buzzer, Temperature, and Door Status LED Connections

The buzzer was connected to digital pin D13, while the temperature indicator LED was connected to pin D12. For door status indication, the red LED was wired to digital pin D3 and the green LED to pin D5.

Step 8: Library Installation and Initialization

We installed the following libraries using the Arduino IDE:

- `DHT` and `Adafruit Unified Sensor` for the DHT11 sensor
- `Servo` for servo control
- `LiquidCrystal_I2C` for I2C LCD display support

We then set the objects , constants , globals and the pin configurations

```
#include <Servo.h>
#include <DHT.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

// === Pin Definitions ===
#define DHTPIN 2
#define DHTTYPE DHT11
#define MOISTURE_PIN A0
#define RELAY_PIN 7
#define TRIG_PIN 8
#define ECHO_PIN 9
#define SERVO_PIN 10
#define BUZZER_PIN 13
#define TEMP_LED_PIN 12
#define DOOR_RED_PIN 3
#define DOOR_GREEN_PIN 5

// === Constants ===
const int MOISTURE_THRESHOLD = 20;
const int DETECTION_DISTANCE_CM = 5;
const int DOOR_OPEN_ANGLE = 90;
const int DOOR_CLOSED_ANGLE = 0;
unsigned long previousSerialMillis = 0;
const unsigned long serialInterval = 2000;

// === Objects ===
Servo doorServo;
DHT dht(DHTPIN, DHTTYPE);
LiquidCrystal_I2C lcd(0x27, 16, 2);

// === Globals ===
long duration;
int distanceCm;
unsigned long previousLcdMillis = 0;
const unsigned long lcdInterval = 2500;
```

Figure 14: Step 8

Step 9: Initialising the setup() function

In the `setup()` function, we initialized all components including:

- Setting appropriate `pinMode()` for input/output devices
- Starting serial communication for debugging
- Initializing the LCD with backlight
- Calibrating the servo to its default (closed) position
- Turning off the relay and all output indicators (buzzer, LEDs)

A startup message was briefly displayed on the LCD before clearing it for dynamic data.

```
void setup() {
  Serial.begin(9600);

  pinMode(MOISTURE_PIN, INPUT);
  pinMode(RELAY_PIN, OUTPUT);
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  pinMode(BUZZER_PIN, OUTPUT);
  pinMode(TEMP_LED_PIN, OUTPUT);
  pinMode(DOOR_RED_PIN, OUTPUT);
  pinMode(DOOR_GREEN_PIN, OUTPUT);

  digitalWrite(BUZZER_PIN, LOW);
  digitalWrite(TEMP_LED_PIN, LOW);
  digitalWrite(DOOR_RED_PIN, LOW);
  digitalWrite(DOOR_GREEN_PIN, LOW);

  doorServo.attach(SERVO_PIN);
  doorServo.write(DOOR_CLOSED_ANGLE);
  digitalWrite(RELAY_PIN, HIGH); // Pump OFF

  dht.begin();
  lcd.init();
  lcd.backlight();

  lcd.setCursor(0, 0);
  lcd.print("System Init...");
  delay(2000);
  lcd.clear();
}
```

Figure 15: Step 9

Step 10: Loop Logic and Functional Implementation

In the `loop()` function, we implemented the following modules:

Proximity-Based Door Control

The ultrasonic sensor measures the distance of nearby objects.

- If an object is closer than 5 cm, the servo opens the door (90°), the green LED lights up, and the red LED turns off.
- Otherwise, the door stays closed (0°), with the red LED on and the green LED off.

```
if (distanceCm < DETECTION_DISTANCE_CM) {
  doorServo.write(DOOR_OPEN_ANGLE);
  digitalWrite(DOOR_GREEN_PIN, HIGH);
  digitalWrite(DOOR_RED_PIN, LOW);
} else {
  doorServo.write(DOOR_CLOSED_ANGLE);
  digitalWrite(DOOR_GREEN_PIN, LOW);
  digitalWrite(DOOR_RED_PIN, HIGH);
}
```

Figure 16: Step 10.1

Soil Moisture Monitoring and Pump Activation

- Moisture readings are taken via analog input from the sensor.
- The raw analog value is mapped to a percentage (0–100%).
- If the moisture level is below 20%, the relay is activated (pump ON). Otherwise, it remains off.

```
// ====== Soil Moisture ======
int raw_moisture = analogRead(MOISTURE_PIN);
int moisture_percent = map(raw_moisture, 1023, 500, 0, 100);
moisture_percent = constrain(moisture_percent, 0, 100);

digitalWrite(RELAY_PIN, (moisture_percent < MOISTURE_THRESHOLD) ? LOW : HIGH);

// ====== DHT Sensor ======
```

Figure 17: Step 10.2

Temperature Monitoring with Alerts

- Temperature and humidity readings are taken from the DHT11 sensor.
- If the temperature exceeds 30°C, both the buzzer and temperature warning LED are activated. Otherwise, they stay off.

```

// ====== DHT Sensor ======
float humidity = dht.readHumidity();
float temperature = dht.readTemperature();

// ====== Temperature-based buzzer and LED ======
if (temperature > 30.0) {
  digitalWrite(BUZZER_PIN, HIGH);
  digitalWrite(TEMP_LED_PIN, HIGH);
} else {
  digitalWrite(BUZZER_PIN, LOW);
  digitalWrite(TEMP_LED_PIN, LOW);
}
```

Figure 18: Step 10.3

Data Display and Debug Output

- Every 2.5 seconds, the LCD is updated to display the current temperature, soil moisture level, door status (Open/Close), and pump status (ON/OFF).
- Simultaneously, serial output provides a real-time log of the same parameters for

debugging purposes.

```
unsigned long currentMillis = millis();
// ====== Serial Output for Debugging ======
if (currentMillis - previousSerialMillis >= serialInterval) {
previousSerialMillis = currentMillis;

  Serial.print("Distance: ");
  Serial.print(distanceCm);
  Serial.print(" cm, Temperature: ");
  Serial.print(temperature);
  Serial.print(" C, Moisture: ");
  Serial.print(moisture_percent);
  Serial.print(" %, Pump: ");
  Serial.print(moisture_percent < MOISTURE_THRESHOLD ? "ON" : "OFF");
  Serial.print(", Door: ");
  Serial.println(distanceCm < DETECTION_DISTANCE_CM ? "OPEN" : "CLOSED");
}
```

Figure 19: Step 10.4(1)

```
  // ====== LCD Display ======
//  unsigned long currentMillis = millis();
  if (currentMillis - previousLcdMillis >= lcdInterval) {
    previousLcdMillis = currentMillis;

    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("T:");
    lcd.print((int)temperature);
    lcd.print(" P:");
    lcd.print(moisture_percent);
    lcd.print("%");

    lcd.setCursor(0, 1);
    lcd.print("D:");
    lcd.print((distanceCm < DETECTION_DISTANCE_CM) ? "Open " : "Close");
    lcd.print(" Pu:");
    lcd.print(moisture_percent < MOISTURE_THRESHOLD ? "ON " : "OFF");
  }

  delay(100);
}
```

Figure 20: Step 10.4(2)

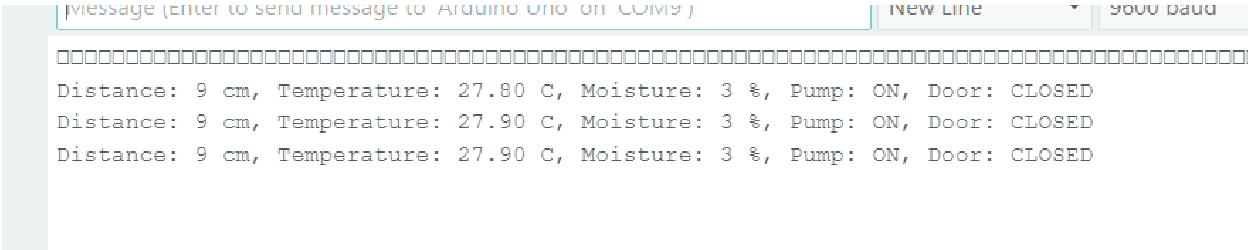

Figure 21: Step 10.4(3)

Step 13: Final Setup and Field Testing

After confirming functionality, we installed components in realistic conditions—soil sensor in soil, ultrasonic at a fixed angle, relay connected to an appliance, and LCD in a visible spot. The Arduino was powered via USB. Final testing confirmed the system ran smoothly, responded to environmental changes, and interacted with devices correctly.

4. Result and Findings

4.1 Results

Test No	**1**
Objective	To test the working of LCD.
Action	Observed the LCD for a minute to check if all the information was being displayed or not.
Expected Result	The LCD should show all the details as required.
Actual Result	The LCD showed all the details as required.
Conclusion	The test was successful.

Table 1: Test 1

Figure 22: Test 1

4.2. Test 2 : Working of door

Test No	**2**
Objective	To find if the smart door is working or not
Action	Palm was kept near the ultrasonic sensor and the activity of the servo motor was observed.
Expected Result	The door should open and close after being sensed.
Actual Result	The door open and closes as expected.
Conclusion	The test was successful.

Table 2: Test 2

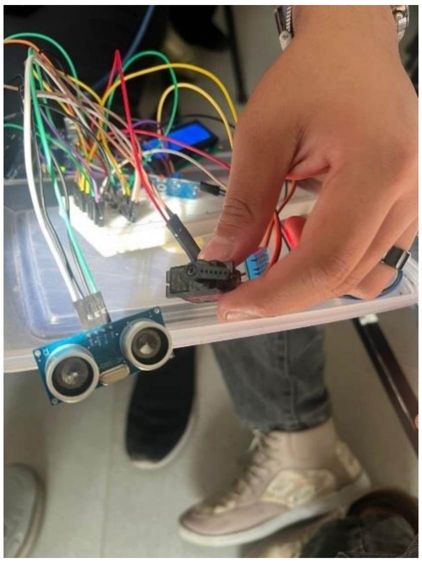

Figure 23: Test 2(1)

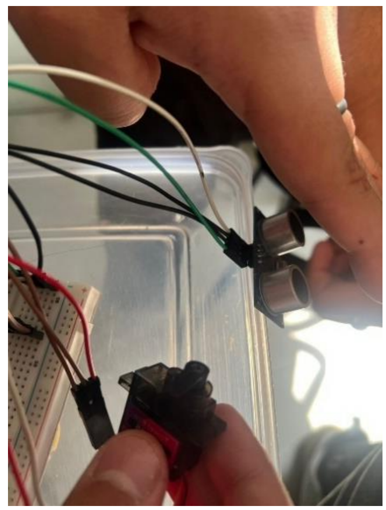

Figure 24: Test 1(2)

Test 3 : Working of soil moisture

Test No	**3**
Objective	To test the soil moisture sensor is working or not
Action	We dipped the soil moisture sensor in water to see if it can detect the water or not
Expected Result	The soil moisture sensor detects the water
Actual Result	Initially, the soil moisture sensor failed to detect water accurately.
Analysis and	To troubleshoot, we tested the sensor using a separate sensor testing sketch. The output consistently displayed a fixed "Raw soil moisture" value, indicating a flaw in the code rather than a hardware issue. Upon review, we found that the map() function used to convert the raw analog values to percentage was not properly calibrated for our sensor's range.
Mitigation	After adjusting the mapping and constraining values appropriately, the sensor began responding correctly to changes in soil moisture.
Conclusion	The soil moisture sensor code was faulty.

Table 3: Test 3

```
// ====== Soil Moisture ======
int raw_moisture = analogRead(MOISTURE_PIN);
int moisture_percent = map(raw_moisture, 550, 10, 0, 100);
moisture_percent = constrain(moisture_percent, 0, 100);

```

Figure 25: Test 3(1)

```
// ====== Soil Moisture ======
int raw_moisture = analogRead(MOISTURE_PIN);
int moisture_percent = map(raw_moisture, 1023, 500, 0, 100);
moisture_percent = constrain(moisture_percent, 0, 100);

```

Figure 26: Test 3(2)

Test 4 : Working of heat sensor

Test No	**4**
Objective	To test the working of heat sensor
Action	The temperature was observed while at rest and the temperature sensor was covered with hand to generate heat.
Expected Result	When the sensor is covered with hand, the temperature should gradually increase.
Actual Result	When the sensor is covered with hand, the temperature has gradually increase.
Conclusion	The test was successful.

Table 4: Test 4

```
Moisture: 53% | Temp: 26.80C | Humidity: 39.70% | Distance: 55cm | Door: Closed | Pump: OFF
Moisture: 54% | Temp: 26.80C | Humidity: 44.00% | Distance: 60cm | Door: Closed | Pump: OFF
Moisture: 54% | Temp: 26.90C | Humidity: 49.10% | Distance: 58cm | Door: Closed | Pump: OFF
Moisture: 52% | Temp: 27.00C | Humidity: 53.80% | Distance: 54cm | Door: Closed | Pump: OFF
Moisture: 52% | Temp: 27.30C | Humidity: 59.30% | Distance: 65cm | Door: Closed | Pump: OFF
Moisture: 53% | Temp: 27.50C | Humidity: 62.80% | Distance: 54cm | Door: Closed | Pump: OFF
Moisture: 53% | Temp: 27.80C | Humidity: 66.60% | Distance: 67cm | Door: Closed | Pump: OFF
Moisture: 53% | Temp: 28.00C | Humidity: 68.40% | Distance: 54cm | Door: Closed | Pump: OFF
Moisture: 53% | Temp: 28.10C | Humidity: 69.50% | Distance: 70cm | Door: Closed | Pump: OFF
Moisture: 53% | Temp: 28.50C | Humidity: 72.40% | Distance: 70cm | Door: Closed | Pump: OFF
```

Figure 27:Test 4

Test 5 : Working of the warning light and buzzer

Test No	**5**
Objective	To observe if the red LED and buzzer gets activated when the temperature rises.
Action	The temperature is increased by holding the sensor in our hand.
Expected Result	When the temperature is above 30 degree Celsius, the LED and the buzzer should be activated.
Actual Result	When the temperature is above 30 degree Celsius, the LED and the buzzer was activated.
Conclusion	The test was successful.

Table 5: Test 5

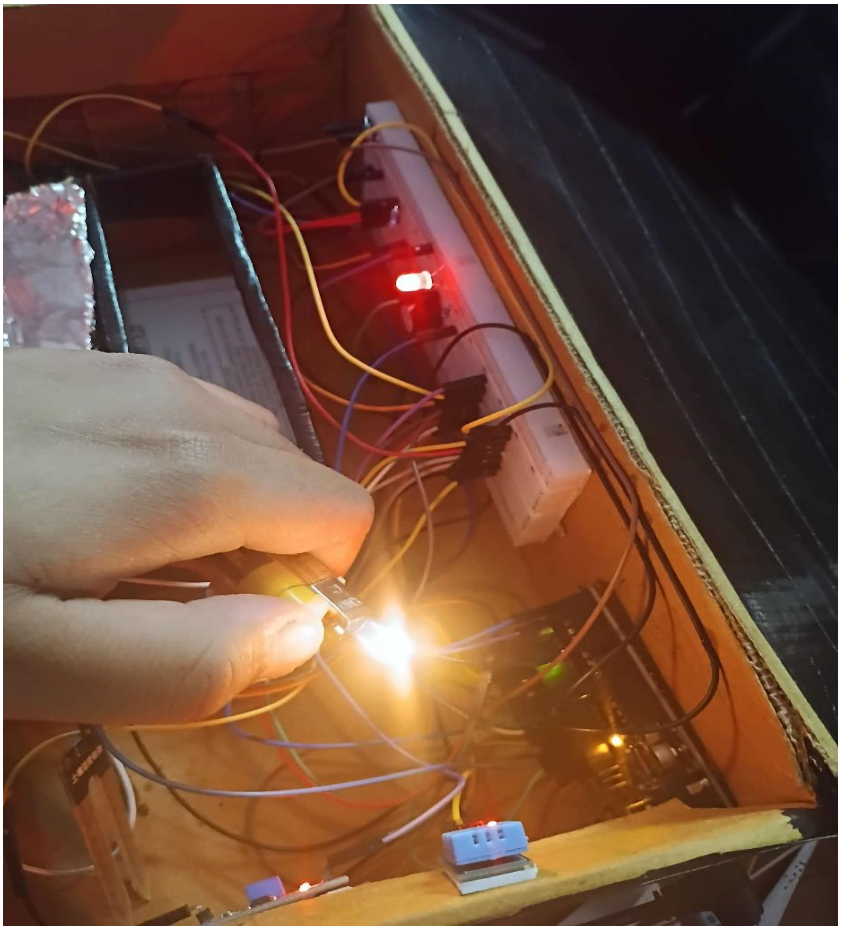

Figure 28: Test 5

5. Future Works

The Smart Farm project has successfully resolved main agricultural challenges via IoT technology, however there are multiple areas of development and improvement of the overall system. The subsequent works will be based on the following.

Integration of Weather Prediction Systems: The system relies right now on real time soil moisture and temperature data. In the future, the integrating of weather prediction systems will allow for weather prediction and adjustment of irrigation schedules according to the weather changes and in turn, reduce water waste and increase crop health.

Machine Learning for Predictive Analytics: The machine learning algorithms could be used by the system to analyze historical data and predict crop yields, predict disease outbreaks and other things while suggesting the optimal planting and harvesting times. It would have a huge impact on making decisions and increasing agricultural output.

Expansion of Smart Devices: Further optimization of farm management would involve introducing more smart devices like an automated pest control, crop health monitoring cameras, as well as soil nutrient sensors. Further expansion of the monitored parameters allows farmers to increase the precision in farm management to even a higher degree.

Mobile App Development: It can be developed as mobile application to enable the farmer being able to monitor and control the system remotely using his or her Smartphone without being in the farm. Such app could offer real time notifications, status of the system, troubleshooting tips and the system could therefore be more accessible to farmers who are in rural parts.

Scalability for Larger Farms: The current design of the system is for smallholder farms. The system could be expanded to serve several farms and many sensors coupled with rapid processing of large volumes of data in real time to accommodate the needs of the larger agricultural operation. Increase in data load can be handled using cloud based solutions and edge computing.

Integration with Government Platforms: Better agricultural health monitoring, resource allocation and policy making can be possible if government agencies collaborates with Smart Farm to integrate Smart Farm data into national agricultural database. In addition, this integration could form a basis for disaster response and planning in the midst of severe weather events.

6. Conclusion

The Smart Farm IoT system developed is an effective and innovative solution to the problems confronting the Nepalese farmers. IoT enabled us in automating some routine farming work such as irrigation, temperature monitoring and security, we provide farmers with real time data and control over their farming operation. Apart from this, it saves a lot of manual work as well as resources such as water hence can make our farming practice sustainable.

The system has shown it’s potential to increase the productivity and security on farms while providing farmers with an inexpensive, easy to use alternative that can be adapted for use in rural Nepal. Additionally, the Smart Farm project deals with the issue of climate change and resource depletion at a larger level which is also a part of the goal goal that the climate resilient agriculture in Nepal.

Nevertheless, although this project shows a potentially good solution, there exist many ways to further expand and optimize the work. The Smart Farm system can be further improved by the incorporation of weather prediction models, machine learning, and increasing the range of smart devices. By continuously scaling this project up and building out this framework over time, it could be a very transformative project to enable this pattern of sustainable, data driven agriculture in Nepal, and beyond.

Smart Farm can be considered a vital step in the transition of the modernizer of agriculture to Nepal, contributing to the livelihoods of smallholder farmers and promoting the movement of smart, sustainable agriculture at the global level.

7. References

Academy, T. I., 2023. *What is an Ultrasonic Sensor, How Does It Work, and What are Its Applications?.* [Online]
Available at: https://www.theiotacademy.co/blog/ultrasonic-sensor/
[Accessed 4 May 2025].

Adams, S., 2025. *What is Microsoft Word: Everything You Need to Know*. [Online]
Available at: https://www.theknowledgeacademy.com/blog/what-is-microsoft-word/
[Accessed 4 May 2025].

Aljundi, L., 2024. *Using the Arduino Software (IDE).* [Online]
Available at: https://docs.arduino.cc/learn/starting-guide/the-arduino-software-ide/
[Accessed 4 May 2025].

Analytics, E. D., 2025. *Soil Moisture Sensors: Smart Tool For Precision Farming.* [Online]
Available at: https://eos.com/blog/soil-moisture-sensor/
[Accessed 4 May 2025].

Arduino, 2018. *What is Arduino?.* [Online]
Available at: https://www.arduino.cc/en/Guide/Introduction/
[Accessed 4 May 2025].

CircuitBread, 2019. *What is a Breadboard?.* [Online]
Available at: https://www.circuitbread.com/ee-faq/what-is-a-breadboard
[Accessed 4 May 2025].

Crompton, 2023. *What are Water Pumps and How do They Work?.* [Online]
Available at: https://www.crompton.co.in/blogs/pumps-guide/what-are-water-pumps-and-how-do-they-work#:~:text=A%20water%20pump%20is%20used,beds%2C%20residential%20buildings%2C%20etc.
[Accessed 4 May 2025].

elprocus, 2022. *What is an LCD Display : Construction & Its Working.* [Online]
Available at: https://www.elprocus.com/ever-wondered-lcd-works/
[Accessed 4 May 2025].

Giencke, A., 2024. *What Is Tinkercad?.* [Online]
Available at: https://all3dp.com/2/what-is-tinkercad-simply-explained/
[Accessed 4 May 2024].

GlobeNewswire, 2023. *GlobeNewswire.* [Online]
Available at: https://www.globenewswire.com/en/news-release/2023/04/18/2649521/0/en/Smart-Farming-Market-Poised-for-Substantial-Growth-Forecasted-to-Surpass-USD-53-Billion-by-2032-Market-us-Report.html
[Accessed 2025].

Hope, C., 2024. *Draw.io.* [Online]
Available at: https://www.computerhope.com/jargon/d/drawio.htm
[Accessed 4 May 2025].

KiCad, 2025. *KiCad.* [Online]
Available at: https://docs.kicad.org/master/en/kicad/kicad.html
[Accessed 2025].

Magazine, E. D., 2024. *Servo Motor: Definition, Working Principle, and Applications.* [Online]
Available at: https://www.electrical4u.com/what-is-servo-motor/
[Accessed 4 May 2024].

Mahalingam, R., 2022. *Resistor Definition, Uses & Examples.* [Online]
Available at: https://study.com/academy/lesson/what-is-a-resistor-definition-uses.html
[Accessed 4 May 2025].

William, 2022. *What is a Relay Module & How Does It Work? (Types & Uses).* [Online]
Available at: What is a Relay Module & How Does It Work? (Types & Uses)
[Accessed 4 May 2025].

Wiltronics, 2022. *What Are Jumper Wires: Know by Colour, Types and Uses.* [Online] Available at: https://www.wiltronics.com.au/wiltronics-knowledge-base/what-are-jumper-wires/ [Accessed 4 May 2025].

8. Appendix

8.1 Source Code

```
#include <Servo.h>
#include <DHT.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

// === Pin Definitions ===
#define DHTPIN 2
#define DHTTYPE DHT11
#define MOISTURE_PIN A0
#define RELAY_PIN 7
#define TRIG_PIN 8
#define ECHO_PIN 9
#define SERVO_PIN 10
#define BUZZER_PIN 13
#define TEMP_LED_PIN 12
#define DOOR_RED_PIN 3
#define DOOR_GREEN_PIN 5

// === Constants ===
const int MOISTURE_THRESHOLD = 20;
const int DETECTION_DISTANCE_CM = 5;
const int DOOR_OPEN_ANGLE = 90;
const int DOOR_CLOSED_ANGLE = 0;
unsigned long previousSerialMillis = 0;
const unsigned long serialInterval = 2000;

// === Objects ===
Servo doorServo;
DHT dht(DHTPIN, DHTTYPE);
LiquidCrystal_I2C lcd(0x27, 16, 2);

// === Globals ===
long duration;
int distanceCm;
unsigned long previousLcdMillis = 0;
const unsigned long lcdInterval = 2500;

void setup() {
  Serial.begin(9600);

  pinMode(MOISTURE_PIN, INPUT);
  pinMode(RELAY_PIN, OUTPUT);
```

```
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  pinMode(BUZZER_PIN, OUTPUT);
  pinMode(TEMP_LED_PIN, OUTPUT);
  pinMode(DOOR_RED_PIN, OUTPUT);
  pinMode(DOOR_GREEN_PIN, OUTPUT);

  digitalWrite(BUZZER_PIN, LOW);
  digitalWrite(TEMP_LED_PIN, LOW);
  digitalWrite(DOOR_RED_PIN, LOW);
  digitalWrite(DOOR_GREEN_PIN, LOW);

  doorServo.attach(SERVO_PIN);
  doorServo.write(DOOR_CLOSED_ANGLE);
  digitalWrite(RELAY_PIN, HIGH); // Pump OFF

  dht.begin();
  lcd.init();
  lcd.backlight();

  lcd.setCursor(0, 0);
  lcd.print("System Init...");
  delay(2000);
  lcd.clear();
}

void loop() {
  // ====== Door Logic ======
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  duration = pulseIn(ECHO_PIN, HIGH);
  distanceCm = duration * 0.034 / 2;


  if (distanceCm < DETECTION_DISTANCE_CM) {
    doorServo.write(DOOR_OPEN_ANGLE);
    digitalWrite(DOOR_GREEN_PIN, HIGH);
    digitalWrite(DOOR_RED_PIN, LOW);
  } else {
    doorServo.write(DOOR_CLOSED_ANGLE);
    digitalWrite(DOOR_GREEN_PIN, LOW);
    digitalWrite(DOOR_RED_PIN, HIGH);
```

```
  }

  // ====== Soil Moisture ======
  int raw_moisture = analogRead(MOISTURE_PIN);
  int moisture_percent = map(raw_moisture, 1023, 500, 0, 100);
  moisture_percent = constrain(moisture_percent, 0, 100);

  digitalWrite(RELAY_PIN, (moisture_percent < MOISTURE_THRESHOLD) ? LOW : HIGH);

  // ====== DHT Sensor ======
  float humidity = dht.readHumidity();
  float temperature = dht.readTemperature();

  // ====== Temperature-based buzzer and LED ======
  if (temperature > 30.0) {
    digitalWrite(BUZZER_PIN, HIGH);
    digitalWrite(TEMP_LED_PIN, HIGH);
  } else {
    digitalWrite(BUZZER_PIN, LOW);
    digitalWrite(TEMP_LED_PIN, LOW);
  }
  unsigned long currentMillis = millis();
  // ====== Serial Output for Debugging ======
  if (currentMillis - previousSerialMillis >= serialInterval) {
  previousSerialMillis = currentMillis;

    Serial.print("Distance: ");
    Serial.print(distanceCm);
    Serial.print(" cm, Temperature: ");
    Serial.print(temperature);
    Serial.print(" C, Moisture: ");
    Serial.print(moisture_percent);
    Serial.print(" %, Pump: ");
    Serial.print(moisture_percent < MOISTURE_THRESHOLD ? "ON" : "OFF");
    Serial.print(", Door: ");
    Serial.println(distanceCm < DETECTION_DISTANCE_CM ? "OPEN" : "CLOSED");
}

  // ====== LCD Display ======
//  unsigned long currentMillis = millis();
  if (currentMillis - previousLcdMillis >= lcdInterval) {
    previousLcdMillis = currentMillis;

    lcd.clear();
    lcd.setCursor(0, 0);
```

```
    lcd.print("T:");
    lcd.print((int)temperature);
    lcd.print(" P:");
    lcd.print(moisture_percent);
    lcd.print("%");

    lcd.setCursor(0, 1);
    lcd.print("D:");
    lcd.print((distanceCm < DETECTION_DISTANCE_CM) ? "Open " : "Close");
    lcd.print(" Pu:");
    lcd.print(moisture_percent < MOISTURE_THRESHOLD ? "ON " : "OFF");
  }

  delay(100);
}
```

8.2 Components

2.3.1 Hardware Components

Arduino Board

Arduino is an open-source electronics platform based on simple hardware and software. Arduino boards can take in a ray of light on a sensor, a finger on a button, or a post on Twitter, and translate it into an output spin a motor, turn on an LED, or post on the web. (Arduino, 2018)

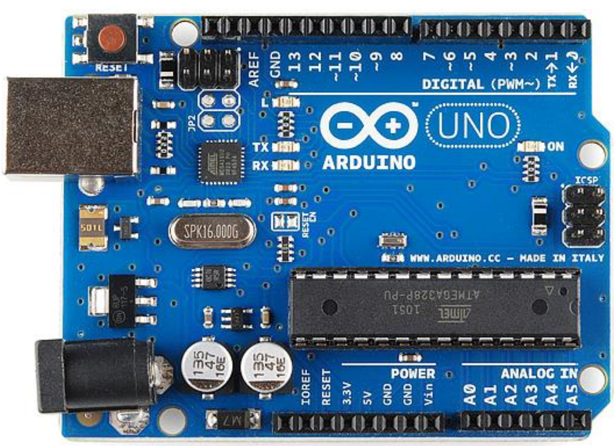

Figure 29:Arduino Board

Servo Motors

A servo motor is defined as an electric motor that allows for precise control of angular or linear position, speed, and torque. It has a proper motor with a position feedback sensor and a controller that manages the motion of the motor in accordance with a desired setpoint. (Magazine, 2024)

Figure 30:Servo Motor

Ultrasonic Sensor

Ultrasonic sensors are electronic devices that determine a target's distance. They work by emitting ultrasonic sound waves and converting those waves into electrical signals. Ultrasonic travel at a faster rate than audible sounds. Therefore, ultrasonic sensor work involves sound waves to find the distance to an item. (Academy, 2023)

Figure 31: Ultrasonic Sensor

Soil Moisture Sensor

A soil moisture sensor, as its name implies, is an appliance for monitoring the moisture levels in the soil. Integrating this tool in the irrigation system allows for more precise watering scheduling than is possible with past information or weather forecasts. (Analytics, 2025)

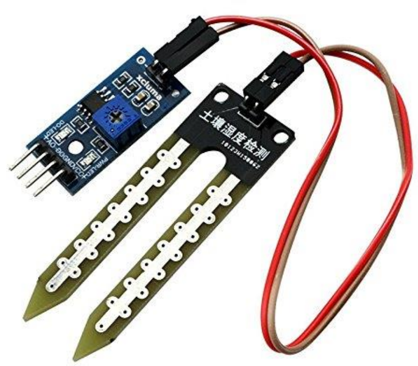

Figure 32: Soil Moisture Sensor

Water Pump

A water pump is used to move, compress, or transfer water from a lower level to a higher one. The main purpose of a water pump is to transfer water between two points and to get rid of excess water. (Crompton, 2023)

Figure 33:Water Pump

LCD Display

A liquid crystal display or LCD draws its definition from its name itself. It is a combination of two states of matter, the solid and the liquid. LCD uses a liquid crystal to produce a visible image. Liquid crystal displays are super-thin technology display screens that are generally used in laptop computer screens, TVs, cell phones, and portable video games. (elprocus, 2022)

Figure 34:LCD Display

Resistors

A resistor is a common component in most electric circuits. Resistors are essential components in electrical networks and electronic circuits, and although may seem to be unusual, their ability to limit the flow of current has numerous applications. (Mahalingam, 2022)

Figure 35: Resistors

Relay module

A relay is an electrical switch that can be used to control devices and systems that use higher voltages. The relay module input voltage is usually DC. However, the electrical load that a relay will control can be either AC or DC, but essentially within the limit levels that the relay is designed for. (William, 2022)

Figure 36:Relay Module

Jumper Wire

Jumper wires are electrical wires with connector pins at each end. They have two or more connection points, which regulate an electrical circuit board. A jumper can be set to enable or disable it. (Wiltronics, 2022)

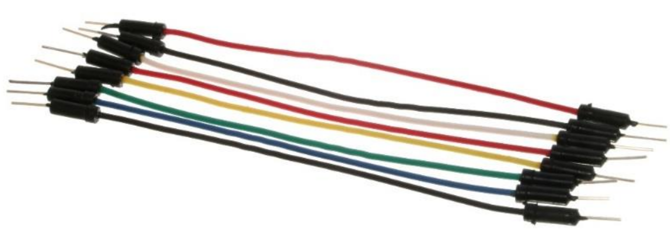

Figure 37:Jumper Wire

Breadboard

A Breadboard is simply a board for prototyping or building circuits on. It allows you to place components and connections on the board to make circuits without soldering. (CircuitBread, 2019)

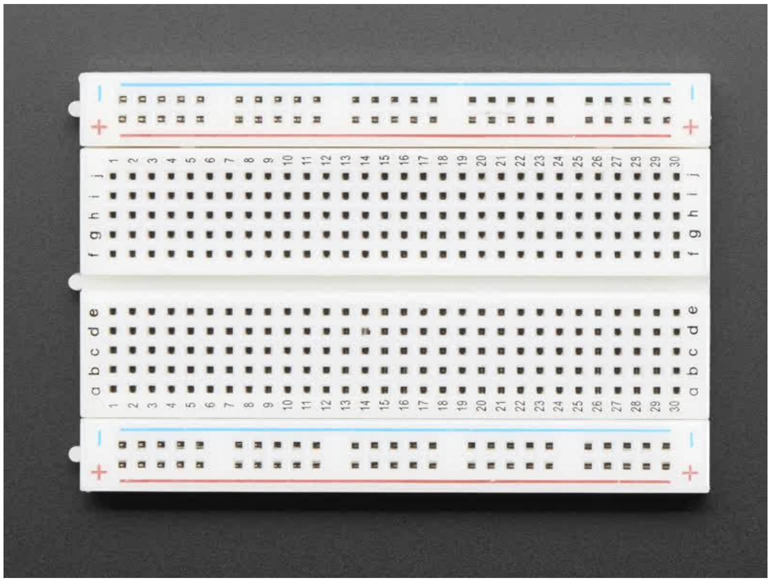

Figure 38:Breadboard

2.3.2 Software Components

Arduino IDE

The Arduino Software (IDE) makes it easy to write code and upload it to the board offline. We recommend it for users with poor or no internet connection. This software can be used with any Arduino board. (Aljundi, 2024)

Figure 39: Arduino IDE

KiCad :

KiCad is an open source software that helps in the creation of electronic circuit schematics and PCBs (KiCad, 2025).

Microsoft Word

Microsoft Word is an element of Microsoft Office which helps with the creation, editing, and formatting of documents. It includes facilities for spell and grammar checking, as well as text formatting. (Adams, 2025)

Figure 40:MS Word

Draw.io

draw.io is proprietary software for making diagrams and charts. The software lets you choose from an automatic layout function, or create a custom layout. (Hope, 2024)

Figure 41: draw.io

8.3 Individual Contribution Table

Student Name	Role	Contribution (in percentage)
David Ramovich Mandal	**Proposal :** -Requirement analysis and individual contribution plan **System development report :** - Test cases **Application Implementation :** - Reviewing the code and improving it. **Presentation :** Exception of the project.	16.666
Rahul Uprety	**Proposal** : - Conclusion and Acknowledgement **System Development Report :** - Procedure and working mechanism of the project **Application Implementation :** - Code to execute the connection between different components **Presentation :** - Working mechanism of the code	16.666
Saurav Timilsina	**Proposal :** Abstract , Introduction , aims and objectives of the proposal **System Development Report :** - Design and Diagram of the flowchart **Application Implementation :** -Setting up the arduino IDE	16.666

	Presentation : -Purpose and objectives of the project	
Hriday Kand	**System Development Report :** - Circuit Design **Application Implementation :** - Making connection between different components **Presentation :** - Purpose and objectives of the project	16.666
Shubam Lamichhane	**Proposal :** - Expected Outcomes and deliverables **System Development Report :** Hardware and Software Evaluation **Application Implementation :** - Setting up the Arduino IDE **Presentation :** - Working mechanism of the project	16.666
Srina Mahat	**Proposal :** - Conclusion and Acknowledgement **System Development Report :** - Test Cases **Application Implementation :** - Reviewong the code and improving **Presentation :** - Exception of the project	16. .666

Table 6: Individual Contribution Table

8.3. Work distribution

Team members :	David Ramovich Mandal **(Team Leader)**	Srina Mahat	Hriday Kand	Shubham Lamichhane	Saurav Timilsina	Rahul Uprety
Proposal	Requirement analysis and individual contribution plan	Conclusion and Acknowledgement	Circuit Design	Expected Outcomes and deliverables	Abstract , Introduction , aims and objectives of the proposal	Conclusion and Acknowledgement
System development report :	Procedure and working mechanism of the project	Design and Diagram of the flowchart	N/A	Hardware and Software Evaluation	- Test cases	- Test cases
Application Implementation	Reviewing the code and improving it.	Reviewing the code and improving	Making connection between different components	Setting up the Arduino IDE	Setting up the arduino IDE	Code to execute the connection between different components
Presentation	Exception of the project.	Exception of the project	Purpose and objectives of the project	Working mechanism of the project	Purpose and objectives of the project	Working mechanism of the code

Figure 42: Work Distribution Plan