The present work was submitted to
the German-Mongolian Institute of Resource and Technology

DESIGN OF PROTOTYPE INDOOR SMART FARM FOR MONGOLIAN HOUSEHOLD

Bachelor's Thesis

By

JAVKHLAN Munkhjargal

Study program: Mechatronics Engineering
Student ID: B2100290

1^{st} Supervisor/Examiner: Ph.D. SUNGCHIL Lee
2^{nd} Supervisor/Examiner: M.E. MYAGMARJAV Bold

Ulaanbaatar/Nalaikh

2025

The present work was submitted to
the German-Mongolian Institute of Resource and Technology

DESIGN OF PROTOTYPE INDOOR SMART FARM FOR MONGOLIAN HOUSEHOLD

Bachelor's Thesis

By

JAVKHLAN Munkhjargal

Study program: Mechatronics Engineering

Student ID: B2100290

1^{st} Supervisor/Examiner: Ph.D. SUNGCHIL Lee

2^{nd}Supervisor/Examiner: M.E. MYAGMARJAV Bold

Ulaanbaatar/Nalaikh

2025

Statutory Declaration

Javkhlan Munkhjargal	B2100290
Last Name, First Name	Student ID Number

I hereby affirm in lieu of an oath that I provided the submitted bachelor thesis

DESIGN OF PROTOTYPE INDOOR SMART FARM FOR MONGOLIAN HOUSEHOLD

I did not use any sources other than those stated. In case that the work is additionally submitted on a data medium, I declare that the written and the electronic form are completely identical. The work was not submitted in the same or similar form to any examination authority.

2025/05/02	Жавклан
Place, Date	Signature

Acknowledgement

I would like to begin by expressing my sincere appreciation to Dr. Sungchil Lee, my primary supervisor, for granting me the opportunity to pursue the thesis topic titled "Design of a Prototype Indoor Smart Farm for Mongolian Households". I am particularly grateful for his support in initiating and assigning this topic, which aligned closely with my academic focus and professional interests. His guidance provided a solid foundation upon which this research was built.

I extend my deep gratitude to Mr. Myagmarjav Bold, my secondary supervisor, whose expertise in web application development was instrumental to this research. His technical advice and continuous feedback played a key role in the successful design and implementation of the system's web interface, helping me resolve numerous challenges encountered during the development phase.

I also wish to thank the faculty and staff of the German-Mongolian Institute for Resources and Technology (GMIT) for fostering a supportive academic environment and for providing the resources necessary to complete this thesis.
My heartfelt thanks go to my family for their unwavering encouragement, patience, and belief in me throughout my academic journey. Their support has been essential to my success.

Finally, I am thankful to my classmates and friends for their collaboration, motivation, and encouragement during the research and implementation stages of this work.

I am truly grateful to everyone who contributed to the completion of this thesis.

Abstract

This thesis presents the design, integration, and validation of a sensor-based indoor smart farming prototype utilizing an Arduino R4 Wi-Fi microcontroller, aimed at enabling efficient, small-scale agricultural automation within constrained household environments in Mongolia. The prototype integrates several key components, including a soil moisture sensor, LDR (Light Dependent Resistor), DHT22 (temperature and humidity sensor) (1), BH1750 (ambient light sensor) (2), UV LED for supplemental plant lighting, and a mini pump controlled by a relay module, all managed through an Arduino R4 Wi-Fi microcontroller (3).

A custom web application was developed to enable real-time environmental monitoring, device control, and system management from remote locations. This application provides users with up-to-date data on soil moisture, light intensity, temperature, and humidity, while also offering manual control options for lighting and irrigation.

In the future, the system is designed to be extended with additional modules such as a fan system for active temperature control and gas sensors (e.g., CO sensors) (4) to monitor air quality, ensuring an even healthier environment for plant growth. Other planned extensions include integrating a water flow sensor and more advanced automation features.

Overall, this research successfully demonstrates a flexible, modular, and practical prototype for indoor smart farming, offering significant potential to support sustainable agricultural practices for Mongolian households.

Table of Contents

List of Appendices

List of Figures

1. Introduction

1.1 Problem Statement

Mongolia faces unique agricultural challenges due to its harsh continental climate, with long winters and short growing seasons (5). Traditional outdoor farming is feasible only for a few months annually, limiting the availability of fresh produce and increasing dependency on imports (6). As a result, a large portion of vegetables and fruits consumed in Mongolia are imported, leading to high dependency on foreign markets, increased costs, and reduced food security.

For Mongolian households, especially those living in urban areas with limited land access, there are few options for growing fresh, healthy food independently throughout the year. Indoor farming technologies offer a potential solution by creating controlled environments where plants can thrive regardless of external weather conditions. However, most commercially available smart farming systems are expensive, complex, and not designed for small-scale or household-level use (7).

Therefore, there is a clear need for an affordable, easy-to-use, and scalable indoor farming system that is specifically designed for Mongolian households. The system should be capable of monitoring key environmental factors such as soil moisture, light intensity, temperature, and air quality, and it should automate responses like irrigation, ventilation, and lighting adjustments. Additionally, the system must be accessible remotely through a user-friendly web interface to provide maximum convenience to users.

This thesis aims to address this need by designing and developing a prototype indoor smart farm system that integrates affordable sensors and actuators with an Arduino R4 Wi-Fi module and a custom-built website, offering a practical solution for year-round home-based food production in Mongolia.

1.2 Research Question

The main research question addressed in this thesis is:

How can a cost-effective and modular indoor smart farm system be designed and implemented to enable Mongolian households to monitor and control key environmental conditions for year-round plant cultivation?

To explore this question, the following sub-questions are also considered:

- What types of sensors and actuators are most suitable for monitoring and controlling soil moisture, light intensity, temperature, and air quality in a household indoor farm environment?
- How can these components be integrated into a reliable and user-friendly system using an Arduino R4 Wi-Fi microcontroller?
- What methods can be used to design an intuitive web-based platform for real-time monitoring, control, and management of the smart farm system?
- How can the system be made scalable and adaptable to include additional environmental controls, such as water flow monitoring and CO gas detection, for future improvements?

By answering these questions, the research aims to develop a prototype that not only addresses the immediate needs of Mongolian households but also provides a foundation for future expansion and further automation in smart farming practices.

2. Literature review

2.1 Smart Farming Systems

Smart farming, often referred to as precision agriculture or digital agriculture, represents a new generation of farming practices that utilize modern technologies to enhance crop productivity, resource efficiency, and sustainability. By integrating Internet of Things (IoT) devices, sensor networks, data analytics, and automation systems, smart farming allows farmers to monitor and control environmental conditions with greater accuracy and minimal human intervention.

The concept of smart farming has evolved significantly over the past decade. Initially focused on large-scale agricultural operations, recent advances in technology have made it possible to apply smart farming principles to smaller-scale farms and even individual households. Key technologies driving this transition include affordable microcontrollers (such as Arduino and Raspberry Pi), low-cost environmental sensors, wireless communication modules, and cloud-based data platforms (3).

Smart farming systems typically monitor parameters such as soil moisture, ambient light, temperature, humidity, and air quality. Based on real-time data, the system can automatically adjust irrigation schedules, activate artificial lighting, or control ventilation to optimize growing conditions. Additionally, remote access via smartphones or web applications allows users to monitor their farm status and intervene manually if needed.

Several studies highlight the benefits of adopting smart farming solutions. According to Wolfert et al. (2017), smart farming can significantly reduce water usage, optimize fertilizer application, and improve crop yields. In urban environments, smart indoor farming systems provide a practical solution to food security challenges, allowing fresh produce to be grown locally and sustainably throughout the year (7).

In the context of Mongolia, smart farming offers a particularly valuable opportunity. Given the country’s severe weather patterns and dependence on food imports, indoor smart farms could help households produce fresh vegetables independently, reducing reliance on external markets and enhancing food resilience (7).

The design of such systems, however, must carefully balance cost, simplicity, and functionality to ensure accessibility for typical households. This thesis builds upon existing smart farming concepts and adapts them to meet the specific needs of Mongolian users by focusing on affordability, ease of use, and modularity.

2.2 Indoor Farming Methods

Indoor farming refers to the practice of growing crops within controlled environments, such as greenhouses, vertical farms, and specialized indoor rooms. These systems are designed to create optimal growing conditions independently of external weather, allowing year-round production of fresh food (7). There are several main types of indoor farming methods, each with its own advantages and challenges.

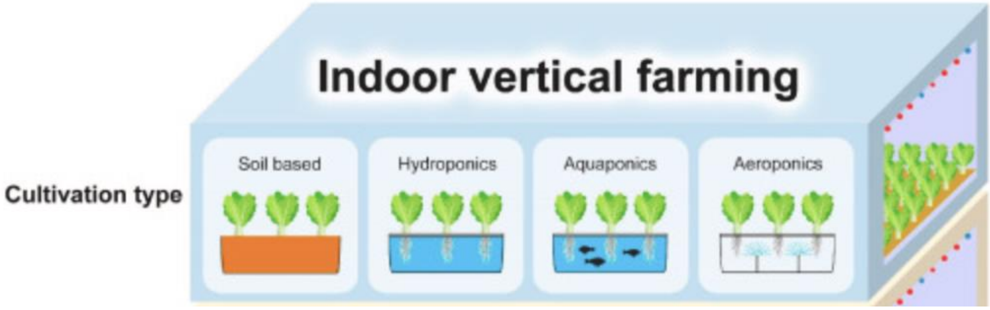


Figure 1. Indoor smart farming cultivation type *(7)*

One common method is hydroponics, which involves growing plants in a nutrient-rich water solution without the use of soil. Hydroponic systems offer high efficiency in water usage and nutrient delivery, and they can achieve faster plant growth rates compared to traditional methods. However, hydroponics typically requires precise management of nutrient concentrations, pH levels, and water quality, making it more technically demanding for beginners (8).

Another advanced method is aeroponics, where plant roots are suspended in the air and periodically misted with nutrient solutions. Aeroponics can maximize oxygen availability to plant roots, resulting in rapid growth, but the system is expensive to set up and requires sophisticated control equipment, making it less practical for small-scale household use (9).

Soil-based indoor farming, by contrast, remains the most accessible and widely adopted method for home growers. In soil-based systems, plants are grown in containers filled with high-quality soil or soil substitutes. This method requires fewer technological inputs and is more forgiving of minor mistakes compared to hydroponic or aeroponic systems. Soil naturally buffers nutrients and retains moisture, reducing the complexity of system management.

Given the context of Mongolian households, soil-based indoor farming is the most suitable approach for this research. It combines low setup cost, simplicity, and user familiarity, making it an ideal foundation for an indoor smart farm prototype. In this system, environmental parameters such as soil moisture, temperature, humidity, light intensity, and air quality must be monitored and adjusted to maintain optimal plant health.

Thus, this thesis focuses on designing a soil-based indoor smart farming system with integrated sensors and automated control, aiming to make indoor farming accessible, affordable, and effective for Mongolian households.

2.3 Schematic Diagram

The design of the indoor smart farm system follows a modular and scalable structure, allowing various sensors and actuators to work together under centralized control. The core of the system is an Arduino R4 Wi-Fi microcontroller (3), which collects sensor data and triggers actuator responses based on predefined conditions.

The key components of the schematic design include:

- Sensors:
    - Soil Moisture Sensor to monitor soil water content.
    - LDR (Light Dependent Resistor) to detect ambient light levels.
    - DHT22 to measure temperature and humidity (1).
    - BH1750 for accurate ambient light intensity (lux) measurement (2).
    - (Future integration) CO Sensor to monitor air quality inside the farm environment.
- Actuators:
    - Mini Water Pump connected through a Relay Module to automate irrigation based on soil moisture readings.
    - UV LED to provide supplemental lighting when natural or indoor light is insufficient.
    - Ventilation Fan to maintain optimal temperature and air circulation when high temperatures are detected.
- Controller:
    - Arduino R4 Wi-Fi Module reads sensor inputs, processes the data according to control logic, and activates actuators as needed (3).
    - It also sends real-time data to the web server via Wi-Fi.

- Web Application:
  - A Node.js-based website displays real-time sensor data (10) (soil moisture, temperature, humidity, light intensity).
  - Users can manually control the pump, fan, and lighting remotely through the dashboard.

The schematic diagram is structured to enable easy maintenance, future sensor expansion (such as water flow monitoring), and smooth communication between the hardware system and the online platform.

A simplified block diagram of the system is shown in Figure 3, illustrating the connection between sensors, actuators, controller, and the web server.

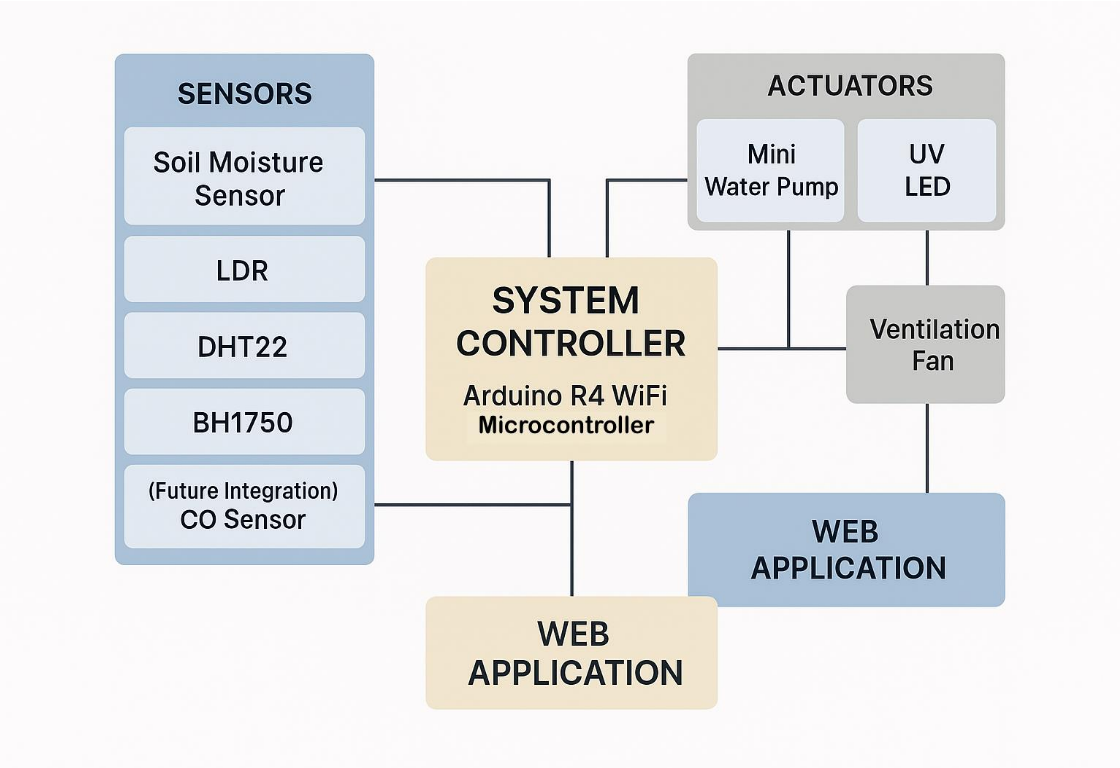


Figure 2. Self-illustrated simplified block diagram of the system.

In addition to displaying real-time sensor values such as soil moisture, temperature, humidity, and light intensity, the web application also provides interactive graph views, allowing users to visualize environmental changes over time. Graphical data representation helps users easily monitor trends, detect abnormalities, and make better-informed decisions regarding plant care.

This modular and extensible schematic design ensures that the indoor smart farm system can adapt to evolving needs and technological improvements in the future.

2.4 Commonly Grown Plants in Indoor Farms

Indoor farming systems provide a controlled environment that is suitable for growing a wide variety of plants year-round, even in regions with extreme external weather conditions such as Mongolia. Successful indoor cultivation typically depends on selecting plants that can thrive in relatively compact spaces and controlled lighting, temperature, and humidity conditions (7).

Some of the most commonly grown plants in indoor farms include (7):

- Leafy Greens:
  - Lettuce (Lactuca sativa)
  - Spinach (Spinacia oleracea)
  - Kale (Brassica oleracea var. sabellica)
- Herbs:
  - Basil (Ocimum basilicum)
  - Mint (Mentha spp.)
  - Parsley (Petroselinum crispum)
  - Cilantro (Coriandrum sativum)
- Vegetables:
  - Cherry tomatoes (Solanum lycopersicum var. cerasiforme)
  - Chili peppers (Capsicum spp.)
  - Radishes (Raphanus sativus)
- Strawberries:
  - Fragaria × ananassa — a popular fruit choice for indoor farming due to its compact size and high value.

These plants are ideal for indoor farming because they have relatively short growing cycles, do not require excessive root space, and respond well to artificial lighting. Additionally, leafy greens and herbs are tolerant to moderate variations in temperature and humidity, making them easier to manage in small-scale systems (11).

For the purpose of this prototype, focusing on leafy greens and herbs provides a practical starting point for Mongolian households, offering nutritious produce with minimal technical complexity. As the system becomes more advanced (e.g., with better light, ventilation, and CO2 control) (3), it can later be adapted to support a wider range of vegetables and fruits.

2.5 Temperature and Humidity Management

Temperature and humidity are two of the most critical environmental factors influencing plant growth, development, and productivity. Maintaining optimal ranges for these parameters inside an indoor farm is essential for ensuring healthy crops and maximizing yields.

Different plants have specific temperature and humidity preferences, but in general, most leafy greens and herbs thrive at temperatures between 18°C to 24°C and relative humidity levels between 50% to 70%. Deviations outside these ranges can cause stress to the plants, leading to slower growth, reduced quality, and increased susceptibility to pests and diseases (1).

In the context of an indoor smart farm for Mongolian households, controlling temperature and humidity becomes even more important due to extreme seasonal temperature variations. During winter, indoor air can become excessively dry, while in summer, indoor temperatures may rise significantly without proper ventilation.

To address these challenges, this prototype integrates:

- A DHT22 sensor to continuously monitor ambient temperature and humidity (1).
- A ventilation fan that is automatically activated when the temperature exceeds a set threshold to maintain a stable growing environment.
- Future planning for humidifiers or dehumidifiers, if needed, to further balance humidity levels for more sensitive crops.

Real-time monitoring through the web application also allows users to view current temperature and humidity values and observe their trends over time through graphical representations. This makes it easier for users to identify patterns and adjust environmental control strategies proactively.

By automating temperature and humidity management, the system reduces the need for constant human supervision, increases energy efficiency, and ensures that plants can grow in an optimal and stress-free environment throughout the year.

2.6 Air Quality and Ventilation Control

Air quality is another vital factor affecting the success of indoor farming. Plants require not only light, water, and nutrients but also a supply of fresh, clean air rich in carbon dioxide (CO_2) for photosynthesis (3). Poor air circulation can lead to stagnant environments where temperature and humidity become uneven, increasing the risk of mold, mildew, and plant diseases.

In a closed indoor environment, ventilation plays a key role in maintaining healthy air quality by:

- Removing excess heat generated by lighting or ambient conditions.
- Reducing humidity buildup, which can otherwise promote fungal growth.
- Providing fresh air to replenish CO_2 levels, which is essential for plant photosynthesis.

In the prototype design of this smart farm system:

- A ventilation fan is controlled automatically based on temperature readings from the DHT22 sensor (1).
- When the ambient temperature rises above a specified threshold, the fan is activated to expel warm air and draw in cooler, fresher air.
- This simple control mechanism helps stabilize both temperature and air quality.

In future system upgrades, the addition of a CO or CO_2 gas sensor is planned.

- A CO (carbon monoxide) sensor can help detect harmful air quality conditions, ensuring a safe environment for both plants and users.
- A CO_2 sensor, if added, would allow for even more precise air management, optimizing photosynthesis rates by maintaining ideal carbon dioxide concentrations.

Table 1 Air Quality Table

#	Element	Density (g/L)	Percentage in Air (%)
1	Carbon Dioxide (CO_2)	1.977	~0.04
2	Oxygen (O_2)	1.429	20.95

3	Nitrogen (N_2)	1.25	78.08
4	Carbon Monoxide (CO)	1.145	<0.001
5	Methane (CH_4)	0.656	<0.0002

Real-time air quality data could also be integrated into the web application, allowing users to receive alerts or automatically trigger fans or ventilation systems based on gas concentration thresholds.

By prioritizing proper air quality and ventilation control, the indoor smart farm system creates a healthier, more stable environment that supports better plant growth and minimizes the risk of environmental stress.

2.7 Light Color, UV, and Photoperiod

Light is one of the most important environmental factors affecting plant growth, influencing photosynthesis, flowering, and overall plant development. In indoor farming systems, where natural sunlight is limited or inconsistent, artificial lighting must be carefully managed to replicate the ideal conditions for plant health (12).

Light Color (Wavelength):

Different wavelengths of light have varying effects on plant processes (7):

- Blue light (around 450–495 nm) promotes vegetative growth, including leaf and stem development.
- Red light (around 620–750 nm) encourages flowering and fruit production.
- Ultraviolet (UV) light, particularly UV-A (315–400 nm), has been found to stimulate secondary metabolite production in some plants, improving flavor, color, and resistance to pests.

To support these needs, this prototype integrates a UV LED system alongside monitoring ambient light levels using LDR and BH1750 sensors (12).

- The LDR provides a basic indication of overall light availability.
- The BH1750 measures precise ambient light intensity in lux, offering better control decisions regarding supplemental lighting needs (2).

Photoperiod (Light Duration):

The photoperiod, or the daily duration of light exposure, plays a crucial role in plant growth stages (12).

- Short-day plants (e.g., lettuce, spinach) require shorter periods of light and longer darkness to flower.
- Long-day plants (e.g., basil, mint) thrive when exposed to extended light periods.

In the smart farm prototype, manual control of the UV LED lights through the web application allows users to adjust lighting schedules based on the specific needs of the plants being grown. In future developments, this system could be automated to simulate natural day-night cycles automatically according to plant species.

Importance for Mongolian Context:

During Mongolia's long, harsh winters, when natural sunlight hours are very short, supplemental UV and artificial lighting become essential for maintaining continuous indoor farming operations.

By carefully managing light color, intensity, and photoperiod, the indoor smart farm ensures that plants receive optimal lighting conditions year-round, leading to healthier growth, better yields, and more efficient resource use.

3. Case Study: Indoor Smart Farm Prototype

This chapter presents the case study of the indoor smart farm prototype developed for Mongolian households. The prototype was designed to demonstrate how affordable, modular technology can enable efficient indoor farming under Mongolia's harsh climate conditions, where outdoor cultivation is often impractical for much of the year.

The primary objective of the prototype was to create a small-scale, self-monitored, and semi-automated farming system that could be easily used by non-expert household users. The system integrates several environmental sensors and control modules managed by an Arduino R4 Wi-Fi microcontroller and connected to a custom-built web application for real-time monitoring and manual control (3).

3.1 Key System Components:

- Sensors:
  - Soil Moisture Sensor: Monitors the water content of the soil to optimize irrigation.
  - DHT22 Sensor: Measures ambient temperature and humidity levels.
  - LDR (Light Dependent Resistor): Detects changes in ambient light conditions.
  - BH1750 Sensor: Provides precise light intensity measurements in lux.
  - CO or CO_2 Sensor: To monitor air quality.
- Actuators:
  - Mini Water Pump with Relay: Activated automatically when soil moisture falls below a defined threshold.
  - UV LED Light: Supplementary lighting for periods of low ambient light, manually controllable via the website.
  - Ventilation Fan: Automatically activated when temperature exceeds a safe limit to maintain ideal growing conditions.
- Controller:
  - Arduino R4 Wi-Fi Module: Serves as the central control unit, reading sensor inputs, processing decisions, activating actuators, and sending data to the web server.
- Web Application:
  - Developed using Node.js and Express framework.

  - Displays real-time sensor data (soil moisture, temperature, humidity, light intensity).
  - Offers graphical visualization of environmental trends.
  - Provides manual control for the pump, UV LED, and fan via a user-friendly dashboard.

3.2 Prototype Features:

- Automated Soil Moisture Control:
  When the soil moisture sensor detects low water content, the mini pump is activated automatically until optimal moisture is restored.
- Temperature-Driven Fan Activation:
  If the DHT22 sensor reports a temperature above the defined threshold (e.g., 26°C), the ventilation fan automatically switches on to cool the environment.
- Light Monitoring and Control:
  The LDR and BH1750 monitor ambient light levels. UV LED lighting can be manually turned on through the website when necessary, especially during darker winter months.
- Real-Time Web Monitoring and Control:
  Users can log in to the web application to monitor live sensor data, view historical trends through graphs, and manually trigger devices as needed.
- Expandability:
  The modular design allows easy integration of future sensors such as water flow meters or air quality sensors (CO or CO_2) to further enhance system capabilities.

This indoor smart farm prototype successfully demonstrates a scalable, low-cost solution for household-level farming, offering Mongolian families the opportunity to grow fresh, healthy produce year-round with minimal technical complexity.

4. Design and Methodology

4.1 Control Functions

The control functions of the indoor smart farm prototype are engineered to autonomously manage critical environmental parameters essential for optimal plant growth, while simultaneously allowing manual override through a web-based user interface. The system's functionality is based on continuous acquisition of real-time sensor data, logical comparison against predefined setpoints, and corresponding actuator responses executed either automatically or manually by the user.

The primary environmental control mechanisms implemented are as follows:

Air Quality-Based Ventilation Control

While temperature-based fan control is already integrated, the prototype is also designed to support future enhancements using air quality sensors (CO or CO_2). The goal is to ensure a safe and CO_2-enriched environment to support healthy plant respiration and optimal photosynthesis.

In the enhanced version of the system, two air quality sensors are planned — one installed at the top and one at the bottom of the plant chamber. These sensors detect vertical CO_2 concentration differences caused by stagnant air. The control logic is as follows:

If measured CO_2 concentration exceeds threshold limit (e.g., 1000 ppm)
Then activate **ventilation fan** to purge CO_2-rich air and draw in fresh air
Else deactivate fan once levels normalize

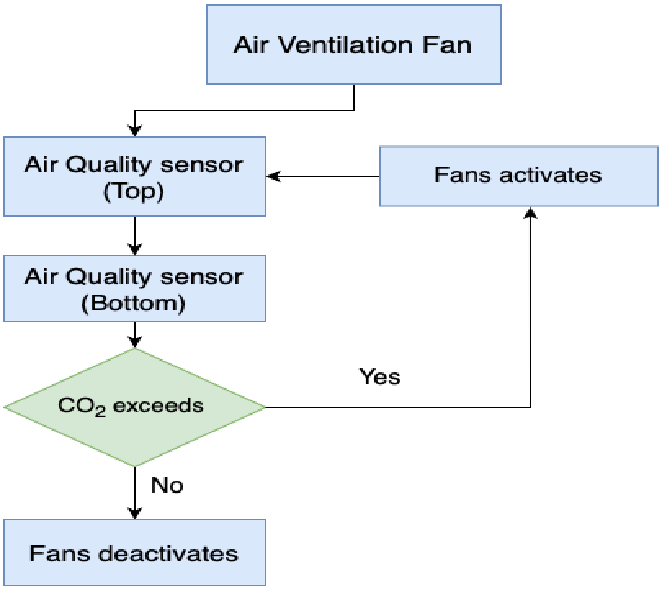


Figure 3. Air Ventilation Control Block Diagram

This mechanism is illustrated in Figure 4: Air Ventilation Control Block Diagram. The fan remains active until both top and bottom sensor readings fall within acceptable limits, ensuring proper air mixing throughout the chamber.

Such a setup helps:

- Remove excess carbon dioxide or airborne contaminants
- Introduce fresh oxygenated air
- Prevent localized high humidity or temperature zones
- Maintain uniform atmospheric composition

While current implementation is limited to temperature-based fan activation, the system architecture is designed to support this CO_2-based ventilation logic in future versions through software updates and additional sensor integration.

Soil Moisture-Based Irrigation Control

The soil moisture sensor continuously monitors the volumetric water content (VWC) of the soil.

An operational threshold, θ_{min}, is established (e.g., 30% VWC). When the measured soil moisture, θ, falls below this threshold:

Condition:
If $\theta < \theta_{min}$, then **Activate Pump**.

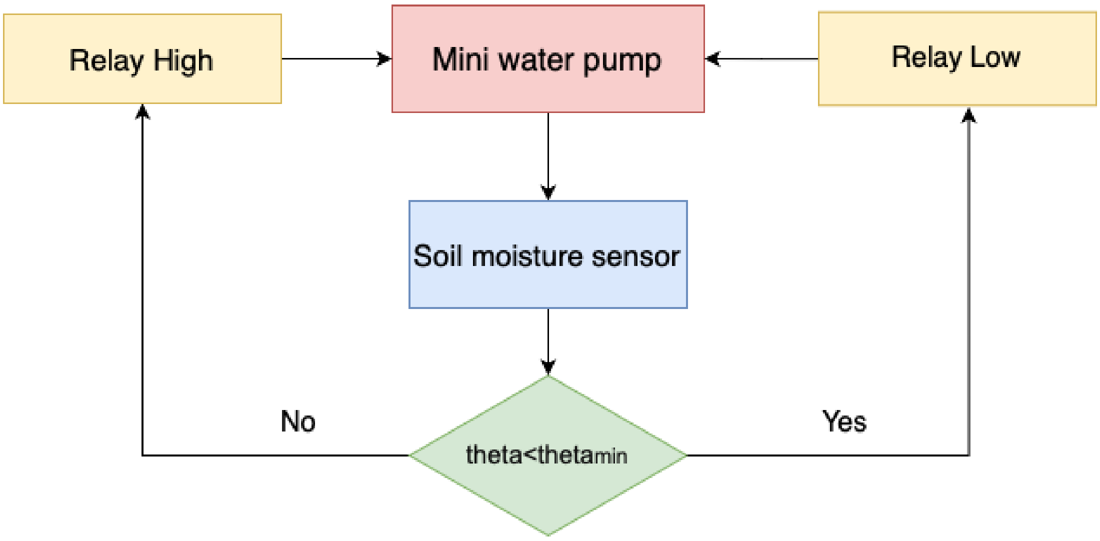


Figure 4. Water Pump Control Block Diagram

The mini water pump, controlled via a relay module, is energized to deliver irrigation water until the sensor reading indicates sufficient rehydration. This closed-loop control approach prevents both over-irrigation and water stress, optimizing water usage efficiency and promoting healthy plant growth.

Temperature-Based Ventilation Control

Ambient temperature is continuously measured by the DHT22 sensor. A temperature threshold, T_{max}, is defined (e.g., 26°C). The ventilation control logic follows:

Condition:
If $T > T_{max}$, then **Activate Ventilation Fan**.
If $T \leq Tmax$, then **Deactivate Ventilation Fan**.

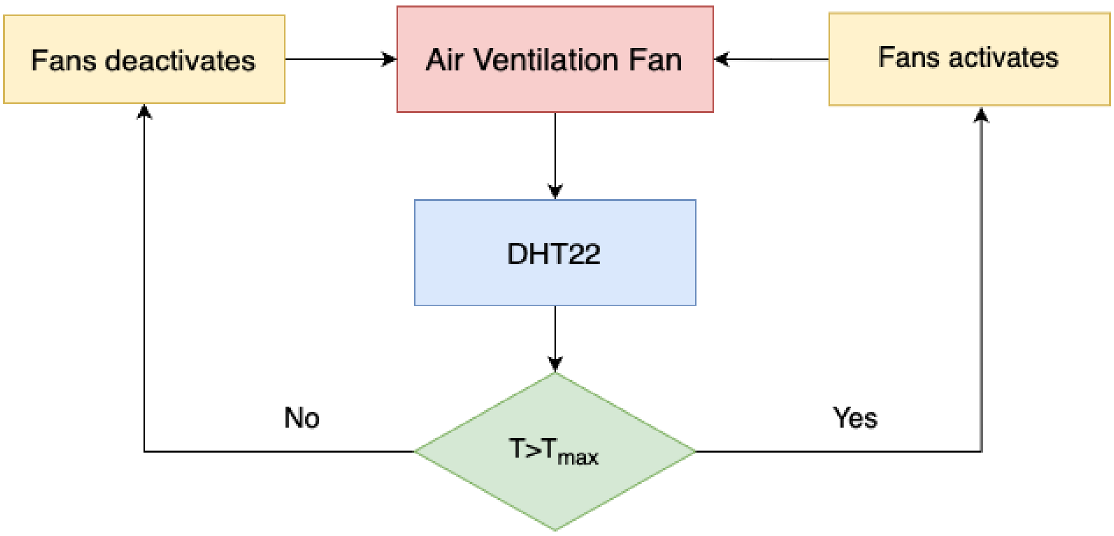


Figure 5. Air Ventilation Fan Control Block Diagram

This ensures the indoor environment remains within biologically favorable temperature ranges, preventing thermal stress to the plants.

Light Intensity Monitoring and Lighting Control

Ambient light levels are assessed through a combination of the LDR and BH1750 sensors. If the measured light intensity, L, falls below a defined threshold lux value, L_{min} , supplemental lighting can be activated:

Condition:
If $L < L_{min}$, **User may activate UV LED Lighting** via web interface.

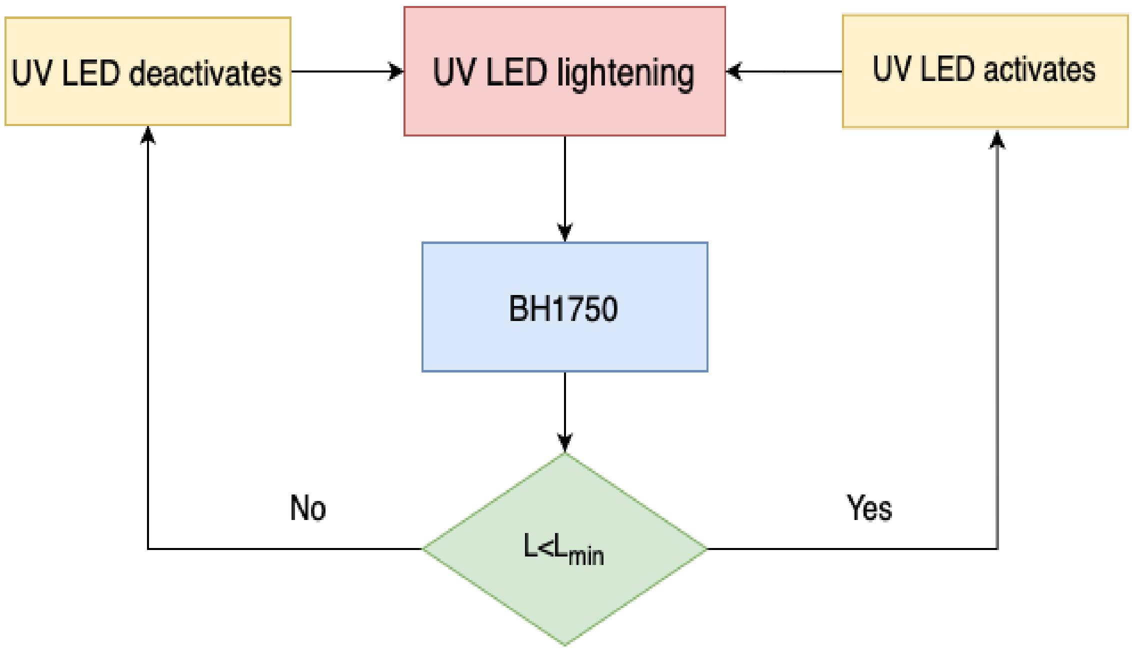


Figure 6. Light Control Block Diagram

While the current system relies on manual activation of UV LED lighting, future versions could automate this process using dynamic lux threshold comparisons.

Web-Based Manual Control

To enhance flexibility, a web-based dashboard interface allows users to manually override the automatic system. Manual controls include:

- Switching the mini water pump ON/OFF.
- Switching the UV LED lighting ON/OFF.
- Switching the ventilation fan ON/OFF.

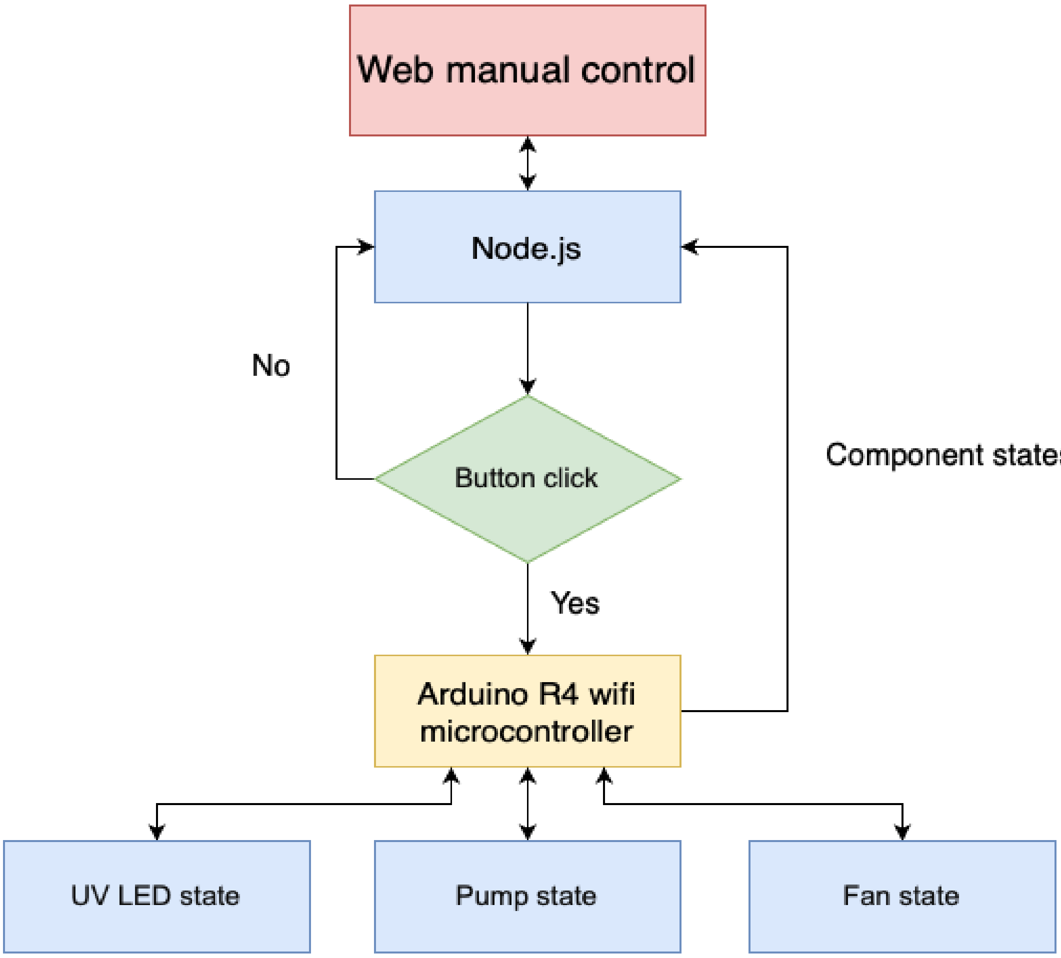


Figure 7. Web-Based Manual Control Block Diagram

This functionality is critical for specific user-driven cultivation strategies or emergency interventions.

Real-Time Monitoring and Data Visualization

Sensor data is wirelessly transmitted to a Node.js-based web server at regular intervals. Users can access:

- Live readings of soil moisture, temperature, humidity, and light intensity.
- Historical environmental data visualized through interactive graphs.

The web application enhances decision-making by enabling trend analysis, facilitating predictive maintenance, and allowing users to optimize farming practices based on data-driven insights.

Data Storage and Expandability

All environmental data streams are securely logged into a MySQL database. This enables:

- Historical data analysis for identifying long-term environmental patterns.
- Evaluation of system performance and plant growth conditions over time.
- The foundation for developing future smart features, such as machine learning algorithms for automated crop management.

The control framework is intentionally modular and scalable. Future expansions may include:

- Integration of CO/CO_2 sensors for monitoring indoor air quality and managing ventilation accordingly.
- Addition of water flow meters for precise irrigation volume control.
- Implementation of automated lighting schedules based on photoperiod algorithms.

Thus, the prototype not only meets immediate operational goals but is also strategically designed for long-term adaptability and intelligence enhancement.

4.2 Sensor Selection

Selecting appropriate sensors is critical to the success of any smart farming system. The sensors must be reliable, affordable, and suitable for indoor farming conditions. For the development of the indoor smart farm prototype, several key sensors were selected based on performance, cost-effectiveness, and ease of integration with the Arduino R4 Wi-Fi microcontroller.

The selected sensors are:

- Soil Moisture Sensor:
  - Purpose: To measure the volumetric water content of the soil and trigger irrigation when moisture drops below optimal levels.
  - Reason for Selection: Affordable, easy to calibrate, and provides simple analog readings suitable for threshold-based control of the water pump.

- DHT22 Temperature and Humidity Sensor:
  - Purpose: To monitor ambient temperature and humidity, ensuring the environment remains within the ideal range for plant growth.
  - Reason for Selection: Compared to DHT11, the DHT22 offers higher accuracy, a wider measurement range (−40°C to +80°C, 0–100% RH), and more stable long-term performance, which is crucial for managing ventilation and climate control.
- LDR (Light Dependent Resistor):
  - Purpose: To detect basic changes in ambient light intensity.
  - Reason for Selection: Extremely low-cost and simple to use; provides a basic measure to detect whether supplemental lighting might be necessary.
- BH1750 Digital Light Intensity Sensor (2):
  - Purpose: To accurately measure ambient light levels in lux, providing more precise information than the LDR.
  - Reason for Selection: High sensitivity, digital output, and easy I2C communication with the microcontroller make the BH1750 ideal for fine-tuning lighting strategies and supplementing UV LED usage.
- (Planned) CO or CO_2 Sensor:
  - Purpose: To monitor indoor air quality, ensuring a safe environment for plants and humans by detecting harmful gas levels or optimizing CO_2 concentration for photosynthesis.
  - Reason for Selection: Air quality sensors are important for future system upgrades to enhance plant growth and indoor environmental safety.

Each sensor is calibrated to ensure accurate readings and integrated into the control system through the Arduino R4 Wi-Fi.

Furthermore, all collected sensor data are stored in a MySQL database, enabling not only real-time monitoring but also historical data analysis for improving farming strategies and system optimization in the future.

By carefully selecting these sensors, the prototype ensures reliable monitoring of key environmental factors critical for successful indoor farming.

4.3 Component Selection

In addition to sensor selection, choosing the appropriate actuators, microcontroller, and supporting hardware components was essential to ensure the reliability, affordability, and functionality of the indoor smart farm prototype. Each component was carefully selected based on its compatibility, performance, cost-efficiency, and ease of integration.

The main components used are:

Arduino R4 Wi-Fi Microcontroller

- Purpose: Acts as the central processing unit of the system, reading sensor inputs, executing control logic, activating actuators, and communicating with the web server.
- Reason for Selection: The Arduino R4 Wi-Fi offers integrated wireless connectivity, sufficient processing power, and ease of programming, making it ideal for IoT-based indoor farming research where real-time monitoring and control are required.

Mini Water Pump with Relay Module

- Purpose: Automates irrigation by pumping water to the plants when soil moisture levels fall below a predefined threshold.
- Reason for Selection: The mini pump is compact, energy-efficient, and suitable for small-scale indoor setups. The relay module ensures safe and reliable switching of the pump without overloading the microcontroller.

To ensure that the selected pump could meet the irrigation needs, the required pumping power was calculated using the following formula:

Equation 1. Pumping Power equation

$$P = (Q * H * \rho * g)/\eta$$

Where:

P = pump power (Watts)

Q = flow rate (m^3/s)

H = pumping head height (meters)

ρ = density of water (1000 kg/m³)

g = gravitational acceleration (9.81 m/s²)

η = pump efficiency (assumed 70%)

Substituting example values (for a flow rate of 0.0005 m³/s and a head height of 1.5 meters):

$$P = (0.0005 * 1.5 * 1000 * 9.81)/0.7 \approx 10.5\,W$$

Thus, a mini pump rated around 10–12 W is sufficient for the prototype's irrigation needs.

UV LED Lighting

- Purpose: Provides supplemental light to support plant photosynthesis when natural or ambient light is insufficient.
- Reason for Selection: UV LEDs are compact, energy-efficient, and effective in promoting plant resilience, especially during Mongolia's long, dark winters when sunlight is scarce.

Ventilation Fan

- Purpose: Maintains optimal temperature and airflow by expelling hot air and bringing in fresh air whenever the indoor temperature exceeds acceptable limits.
- Reason for Selection: Small, low-power fans are ideal for small enclosed farming spaces to regulate temperature effectively.

To select an appropriate fan, the required airflow rate (Q) was estimated based on the volume of the enclosed growing space using the following formula:

Equation 2. Airflow Rate equation

$$Q = V * ACH$$

Where:

Q = required airflow (m³/h)

V = volume of the growing space (m³)

ACH = desired air changes per hour (typically 5–10 for small grow environments)

Assuming a growing volume of 0.5 m^3 and 7 air changes per hour:

Q = 0.5 *7 = 3.5 m/h

Thus, a fan with a minimum airflow capacity of 4–5 m^3/h was selected to ensure sufficient ventilation.

Relay Modules

- Purpose: Interfaces between the Arduino and higher-power devices like the mini water pump and ventilation fan, enabling safe switching operations.
- Reason for Selection: Relays are necessary to isolate the low-voltage Arduino logic circuits from the higher voltage/current demands of the actuators, protecting the microcontroller from electrical damage.

Power Supply Units

- Purpose: Provide regulated 5V and 12V power to the Arduino, sensors, and actuators.
- Reason for Selection: Proper voltage regulation ensures consistent system performance without overloading any individual component, especially critical for sensitive electronics like microcontrollers and sensors.

All components were selected with future expandability in mind. Additional modules, such as CO/CO_2 sensors, advanced lighting systems, or water flow meters, can be seamlessly integrated without major hardware redesign.

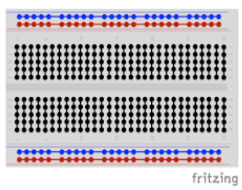
By carefully selecting low-cost yet highly functional components, the system remains accessible and affordable for Mongolian households, supporting the goal of promoting local indoor agriculture and enhancing food security.

4.4 Required Hardware List

The following table summarizes all the hardware components used in the development of the indoor smart farm prototype, along with their quantity and intended purpose:

Table 2 Required Hardware Table

No	Component	Quantity	Purpose	Picture
1	Arduino R4 Wi-Fi	1	Main controller for sensor reading and actuator control	
2	Soil Moisture Sensor	1	Measure soil water content for irrigation control	
3	DHT22 Temperature and Humidity Sensor	1	Monitor ambient temperature and humidity	
4	LDR (Light Dependent Resistor)	1	Detect basic ambient light levels	
5	BH1750 Light Intensity Sensor	1	Measure precise ambient light in lux	
6	Mini Water Pump	1	Automate watering based on soil moisture levels	
7	Relay Module (for Pump)	1	Control high-power switching for pump	

8	UV LED	1–2	Provide supplemental light for plant growth	
9	Ventilation Fan	1	Control temperature and air circulation	
10	Relay Module (for Fan)	1	Control high-power switching for fan	
11	Power Supply (5V/12V)	1	Provide stable power to Arduino, sensors, and actuators	
12	Connecting Wires	Several	Circuit connections between components	
13	Breadboard / PCB	1	Prototyping platform for hardware assembly	
14	Type-C Cable (for Arduino)	1	Upload programs and provide power during testing	

Future planned additions:

- CO or CO_2 Sensor: For air quality monitoring
- Water Flow Sensor: For more precise irrigation tracking

The complete hardware wiring and sensor-actuator integration are illustrated in Figure 9. This schematic visualizes the connections between the Arduino R4 Wi-Fi microcontroller, OLED display, DHT22 sensor, BH1750 light sensor, soil moisture sensor, relay module, and the mini water pump.

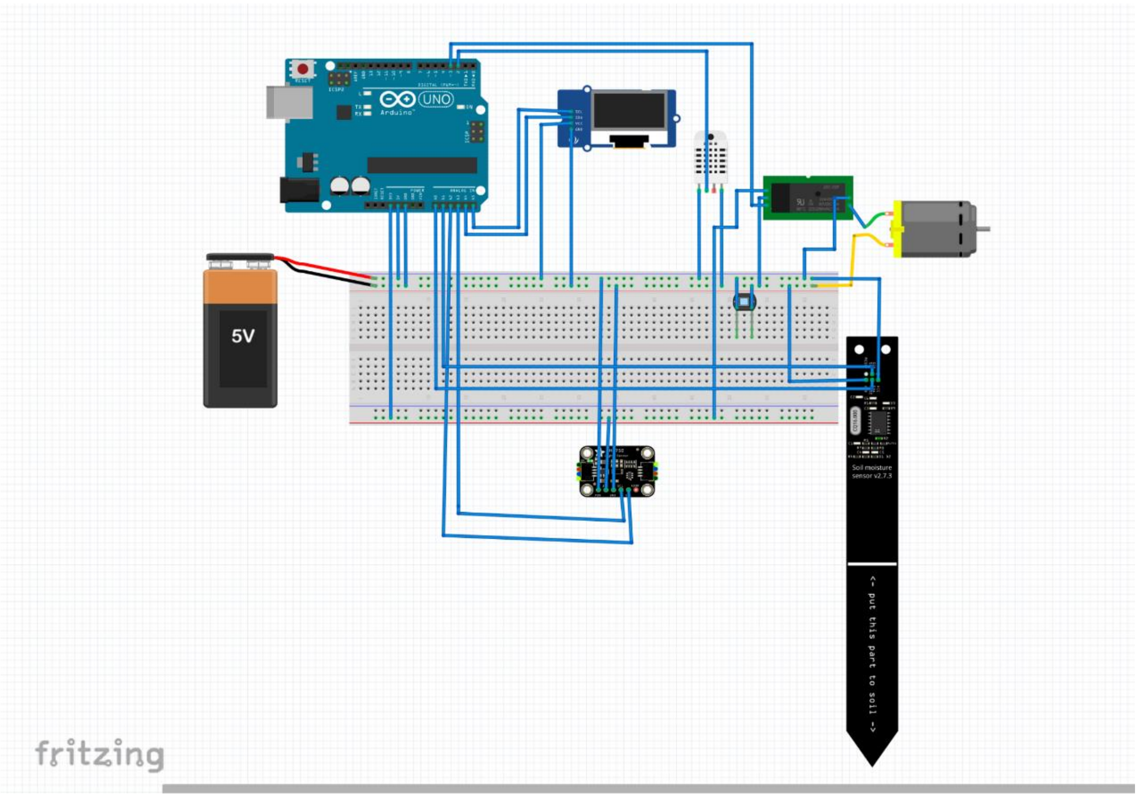


Figure 8 Schematic diagram of the smart farm system circuit, showing full wiring of sensors, actuators, and power connections using Fritzing.

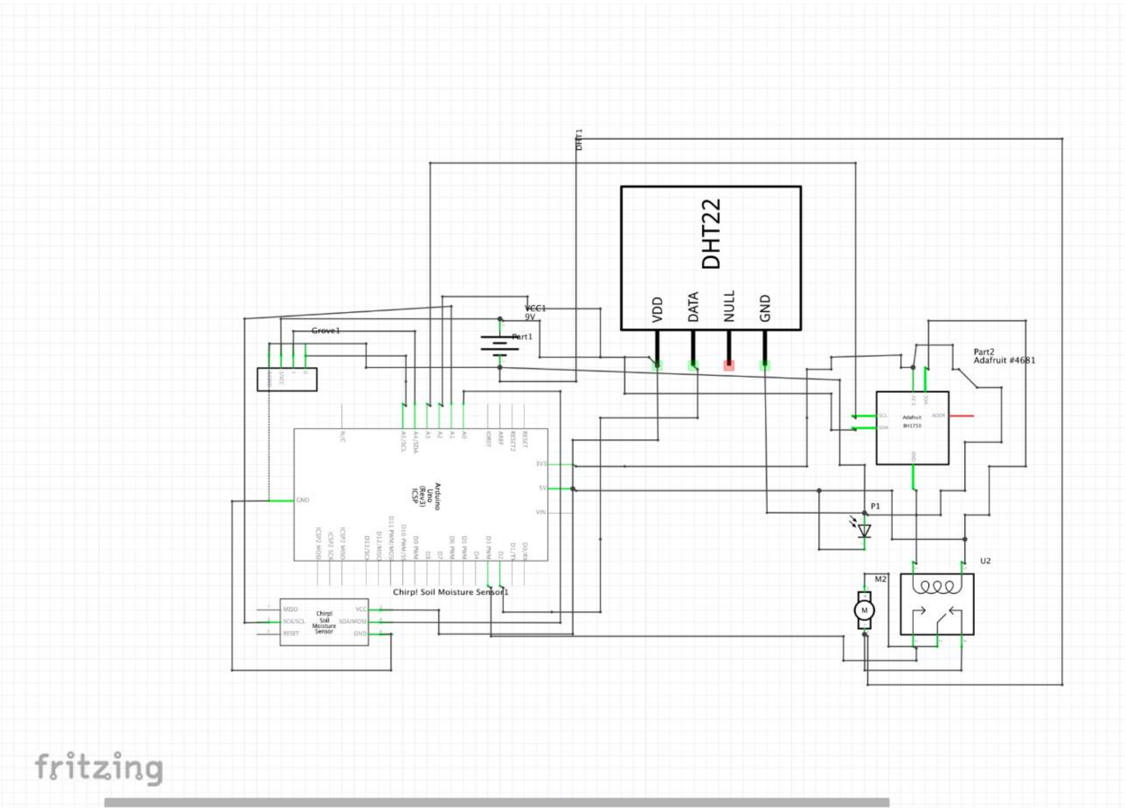


Figure 9 Electrical schematic of the smart farm circuit, displaying precise wiring and pin configuration between the Arduino R4 Wi-Fi, DHT22 sensor, soil moisture sensor, relay module, and water pump. Created using Fritzing software in schematic view mode.

4.5 Software Design

The software architecture of the indoor smart farm prototype was developed to enable seamless data acquisition, real-time control, secure data storage, and user interaction through a web-based interface. The design follows a layered approach, consisting of the embedded system (microcontroller firmware), backend server infrastructure, database management, and the frontend user dashboard.

4.5.1. Embedded Firmware (Arduino R4 Wi-Fi)

The Arduino R4 Wi-Fi microcontroller was programmed using the Arduino IDE, leveraging C++ for embedded systems. The firmware is responsible for:

- Sensor Data Acquisition: Periodically reading environmental parameters from the soil moisture sensor, DHT22 sensor, BH1750 light sensor, and LDR.
- Control Logic Execution: Comparing sensor readings against predefined thresholds and triggering actuator outputs (mini pump, ventilation fan, UV LED) accordingly.
- Data Transmission: Sending sensor data to the backend server over Wi-Fi using HTTP POST requests or WebSocket protocols for low-latency communication.

- OLED Display Updates: Locally displaying the latest sensor readings to enable immediate system status visualization.

The firmware incorporates interrupt handling, task scheduling, and basic fault tolerance to enhance system stability during long-term operation.

4.5.2. Backend Server (Node.js + Express Framework)

The backend server was implemented using Node.js and the Express framework, providing RESTful API endpoints for communication between the Arduino microcontroller and the frontend dashboard. Key backend functionalities include (10):

- Data Reception: Receiving incoming sensor readings from the Arduino at specified intervals.
- Command Routing: Sending control signals to actuators upon user commands initiated through the frontend.
- Session Management: Authenticating users and maintaining secure sessions for access control.
- Error Handling: Managing network disconnections and ensuring system robustness through retry mechanisms and status monitoring.

Node.js was chosen for its non-blocking asynchronous architecture, which ensures real-time performance and efficient handling of multiple concurrent connections.

4.5.3. Database Management (MySQL Server)

A **MySQL** relational database was deployed to persistently store sensor data, user profiles, and actuator control logs. The database schema was designed with normalization principles to ensure data integrity and scalability (4).

The database structure includes:

- **Sensor_Data** table: timestamp, device ID, soil moisture, temperature, humidity, and light intensity values.
- **User_Accounts** table: user authentication information (with encrypted passwords).
- **Control_Actions** table: logging manual interventions (e.g., pump ON/OFF, fan activation).

By archiving historical data, the system supports trend analysis, fault diagnosis, and future implementation of predictive models.

4.5.4. Frontend Web Application (EJS Templates + AJAX)

The user interface was developed using Embedded JavaScript (EJS) templates integrated with AJAX (Asynchronous JavaScript and XML) calls to facilitate dynamic updates without requiring full page reloads (10). The frontend features include:

- Real-Time Data Display: Live visualization of soil moisture, temperature, humidity, and light intensity.
- Graphical Trend Visualization: Historical environmental data is rendered into interactive line graphs to assist users in identifying patterns over time.
- Manual Device Control: Buttons for activating/deactivating the pump, fan, and UV LED.
- User Management: Login, registration, and account management functionalities.
- Responsive Design: The web dashboard is optimized for various device screens, enhancing accessibility for both desktop and mobile users.

The frontend interface, developed using EJS templates and styled with responsive CSS, provides users with real-time sensor monitoring and manual control. Figure 4.5 illustrates the live sensor data interface, which displays temperature, humidity, light status, soil moisture, and light intensity (lux). It also allows users to manually toggle the pump and UV lighting, and to switch between auto and manual operating modes.

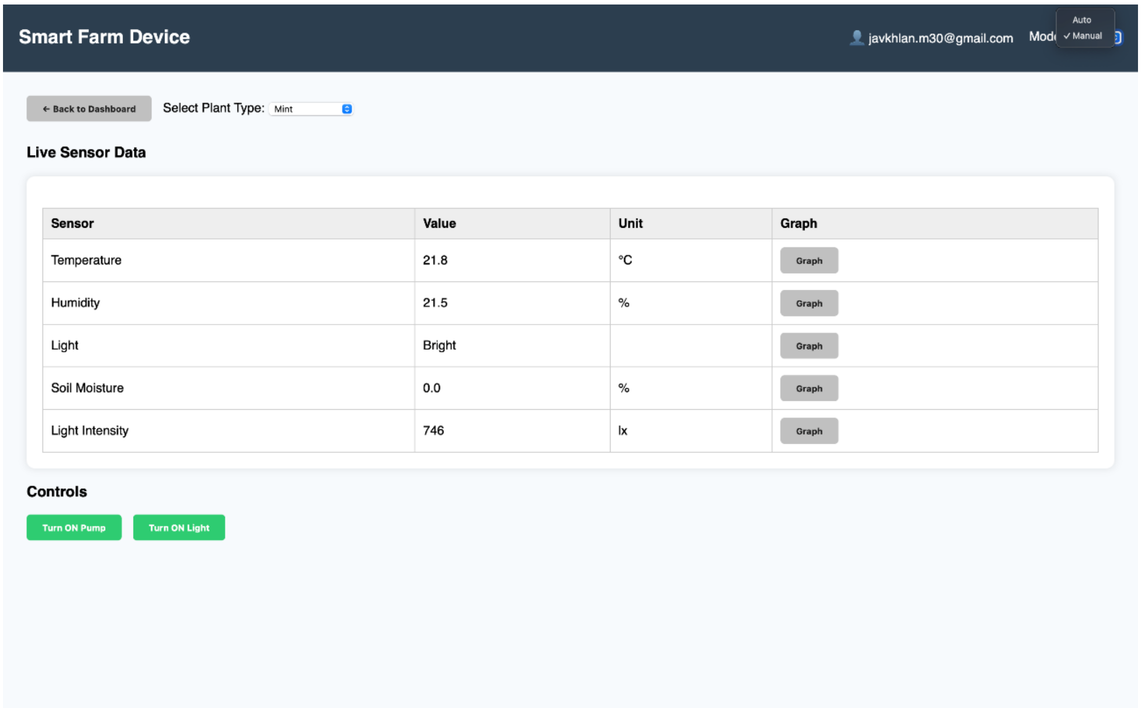


Figure 10 Screenshot of the Smart Farm Device interface showing live sensor data and manual controls for pump and lighting, with plant-specific threshold configuration.

The frontend communicates securely with the backend through authenticated API calls, maintaining data privacy and system integrity.

4.5.5. System Integration Overview

The overall communication flow is as follows:

- Arduino collects sensor data → sends to Node.js server.
- Server stores data into MySQL database → serves updated data to frontend.
- Frontend displays real-time updates → user can send manual control commands.
- Server relays commands back to Arduino → Arduino actuates devices accordingly.

This architecture ensures low-latency, robust, scalable, and user-friendly operation, making the smart farm system suitable for household implementation and future enhancements such as machine learning-driven automation.

5. IMPLEMENTATION

The implementation phase focused on the systematic assembly, integration, programming, and functional verification of the indoor smart farm prototype. Based on the design specifications outlined during the development phase, both hardware and software components were combined into a unified operational system. This chapter details the key steps undertaken for system assembly, deployment, testing, and troubleshooting.

5.1 Hardware Assembly

The hardware setup began with individual testing of all selected components to verify operational integrity and compatibility.

The following sequential steps were followed during the hardware assembly phase:

- The Arduino R4 Wi-Fi microcontroller was selected as the central processing unit and powered through a stabilized 5V regulated power supply.
- Environmental sensors, including the soil moisture sensor, DHT22 temperature and humidity sensor, LDR, and BH1750 light sensor, were connected to the corresponding analog and digital I/O pins of the Arduino.
- Actuators such as the mini water pump and ventilation fan were interfaced through relay modules to safely control higher current loads without exposing the microcontroller directly to switching currents.
- The UV LED lighting system was wired to a separate relay circuit to enable both manual and automatic activation based on ambient light levels.
- A 0.96-inch OLED display module was integrated via the I2C bus to provide immediate, local real-time visualization of environmental parameters such as soil moisture, temperature, humidity, and light intensity.
- Power management was carefully planned, ensuring that sensors and actuators operated within their designated voltage and current specifications. Separate voltage regulators and current protection mechanisms were employed where necessary.
- The initial system prototype was constructed using a breadboard to allow flexible reconfiguration during the testing stage.
- Upon successful validation, the circuit was transferred to a custom-designed PCB (Printed Circuit Board) to improve connection stability, minimize noise, and protect against mechanical failures.

- Comprehensive wiring diagrams were created and validated to ensure logical, efficient, and safe electrical connections. All critical pathways were tested for resistance and continuity to avoid short circuits or open faults.

The final prototype was assembled in a small planting container to simulate real-world growing conditions. The Arduino R4 Wi-Fi board, water pump, BH1750 sensor, OLED display, and relay modules were mounted on a foam-based board and integrated using a breadboard and jumper wires. Figures 9 and 10 illustrate the actual assembled model and the live sensor display on the OLED screen.

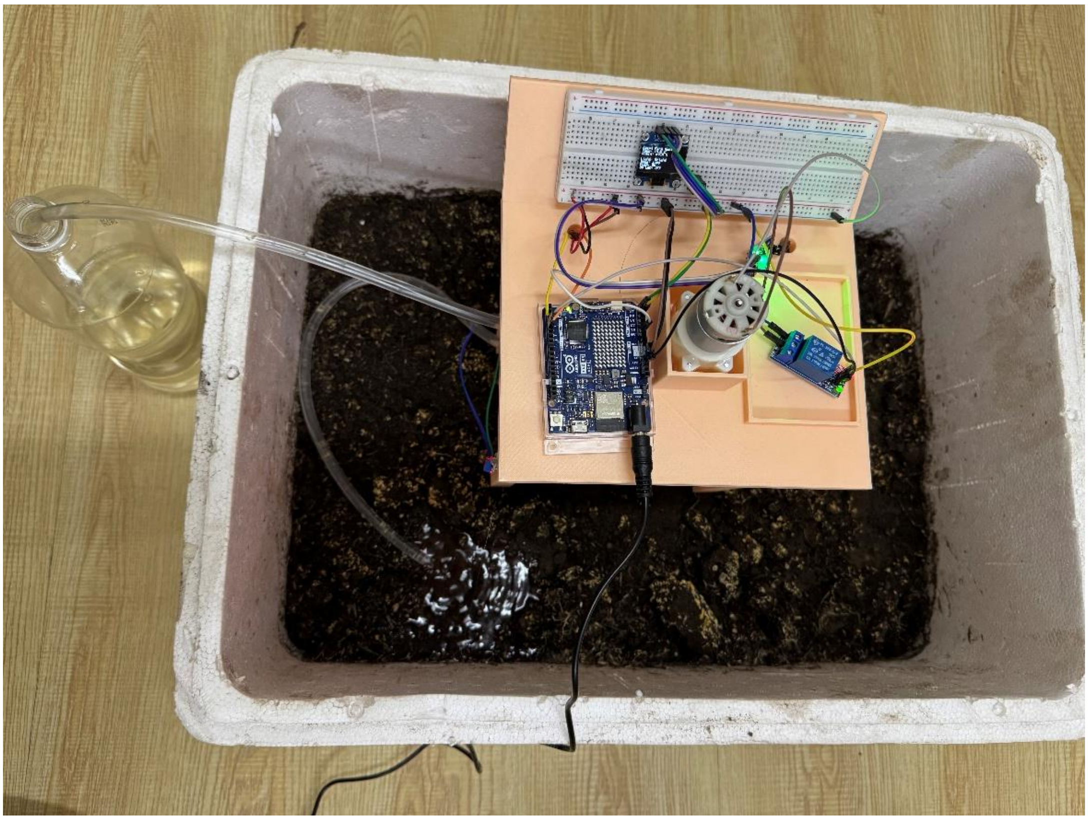


Figure 11 Photograph of the implemented smart farm prototype showing soil setup, Arduino board, relay modules, and water flow system.

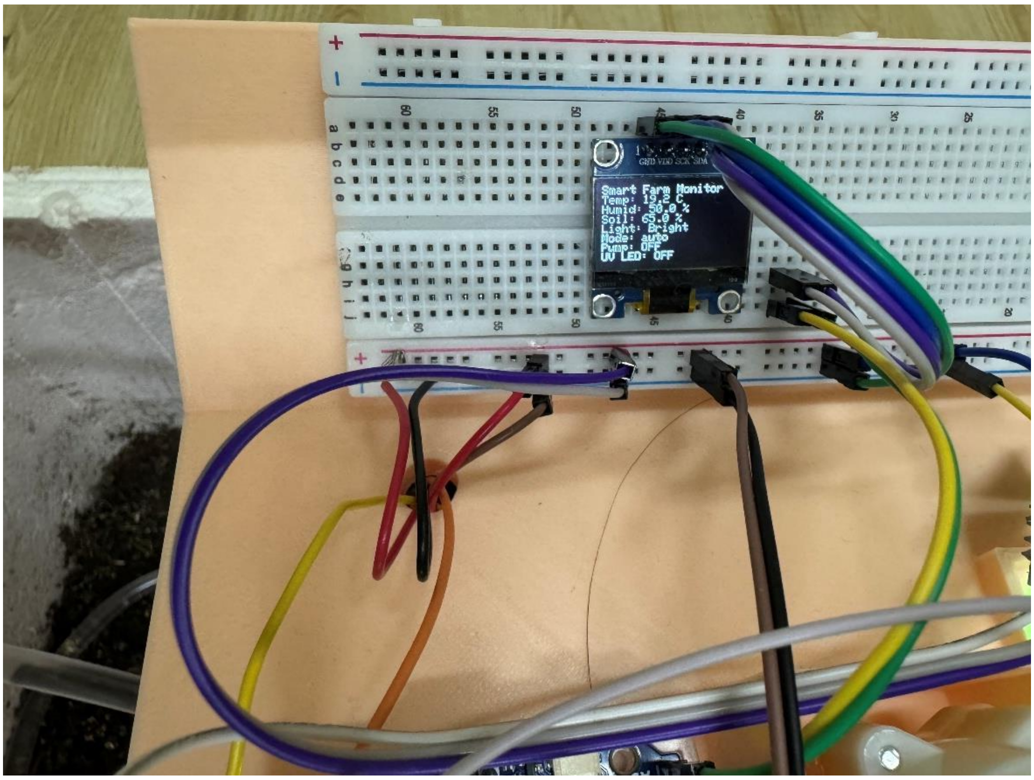


Figure 12 Close-up of the OLED display showing real-time readings of temperature, humidity, soil moisture, control mode, and component status

This structured approach ensured that the hardware subsystem maintained operational robustness and reliability under long-term working conditions.

5.2 Software Deployment

The software subsystem was developed to ensure seamless interaction between hardware sensors, actuators, and the user interface.

Microcontroller Firmware

- The firmware was developed and uploaded to the Arduino R4 Wi-Fi using the Arduino IDE.
- Main features programmed into the microcontroller include:
  - Sensor Data Acquisition: Periodic reading of sensor values at defined intervals.
  - Control Logic Execution: Comparison of real-time sensor readings against preset threshold values and activation/deactivation of actuators accordingly.
  - Data Transmission: Sending structured environmental data packets to the backend server over Wi-Fi using HTTP POST requests.

    - OLED Display Updates: Refreshing sensor values on the local OLED display for immediate visualization.
    - Manual Control Reception: Listening for user-generated manual control signals from the web server and executing corresponding actuator actions.

Backend Server (Node.js + MySQL)

- A Node.js server was deployed to facilitate communication between the Arduino device, MySQL database, and the frontend dashboard.
- RESTful APIs were developed to:
    - Receive sensor readings from the microcontroller.
    - Serve live sensor data to the frontend application.
    - Handle actuator control commands initiated by users.
- MySQL was configured to:
    - Create normalized tables for sensor data logging (timestamped), user management, and device action history.
    - Enable efficient querying for real-time and historical data visualization.

Frontend Web Application (EJS + AJAX)

- A responsive web dashboard was developed using EJS templates and styled with CSS for improved user experience across devices.
- AJAX was utilized to:
    - Continuously fetch live sensor data without requiring page reloads.
    - Dynamically update actuator status and historical data graphs.
- Interactive graphs were implemented to visualize trends in temperature, humidity, soil moisture, and light intensity over time, aiding users in better understanding environmental changes and plant responses.

The Arduino firmware and web application were tightly integrated to provide real-time two-way communication between hardware and user interface.

5.3 Testing and Troubleshooting

Following the integration phase, extensive testing and troubleshooting were conducted to validate system functionality under realistic operating conditions.

Key verification steps included:

- Sensor Accuracy Verification:
    - Repeated manual measurements were conducted to verify the accuracy of soil moisture, temperature, humidity, and light intensity readings.
    - Calibration adjustments were made where deviations exceeded acceptable tolerances.
- Actuator Response Testing:
    - Soil was intentionally dried to below-threshold moisture levels to verify the automatic activation of the water pump.
    - Temperature elevation was simulated to trigger automatic fan activation.
    - Light deprivation scenarios were created to test manual UV LED activation via the dashboard.
- Communication and Data Integrity Testing:
    - Sensor data packets were transmitted wirelessly to the server at regular intervals.
    - Successful insertion and retrieval of data from the MySQL database were confirmed.
    - Manual actuator commands from the frontend were verified for real-time execution by the Arduino.
- Web Dashboard Functionality Testing:
    - Real-time display of sensor readings was tested under fluctuating conditions.
    - Graphs were tested for proper rendering of historical data.
    - User authentication, device addition, and control permissions were validated.
- Network Stability Testing:
    - The system was operated over extended periods under different Wi-Fi signal conditions to ensure consistent communication without significant data loss or connection failures.
- Issue Resolution:
    - Minor timing conflicts between sensor reading loops and Wi-Fi transmissions were resolved by optimizing firmware scheduling.
    - Instability in relay activation under fluctuating supply voltages was mitigated through addition of filtering capacitors.

Upon completion of testing and minor refinements, the full system was confirmed to operate reliably, achieving its design objectives of providing an affordable, accessible, and modular smart farming solution for Mongolian households.

6. RESULT AND CONCLUSION

6.1 Results

The indoor smart farm prototype was successfully developed and implemented according to the design specifications. The system demonstrated stable and reliable performance during testing, achieving the core objectives outlined at the beginning of the research.

The key results are summarized as follows:

- Environmental Monitoring:
  - The soil moisture sensor accurately detected moisture levels and triggered automatic irrigation when required.
  - The DHT22 sensor consistently monitored ambient temperature and humidity, providing reliable data for environmental control.
  - The LDR and BH1750 sensors effectively measured ambient light intensity, enabling the system to detect low-light conditions.
  - All sensor readings were displayed in real-time on the OLED screen, allowing immediate local monitoring.
- Automated Control Functions:
  - The mini water pump was automatically activated based on soil moisture readings, ensuring optimal irrigation without human intervention.
  - The ventilation fan operated correctly by responding to temperature thresholds, maintaining a suitable environment for plant growth.
  - The UV LED lighting was successfully controlled manually through the web application to supplement lighting during low natural light conditions.
- Web-Based Monitoring and Control:
  - Real-time sensor data was transmitted via Wi-Fi to a Node.js server and stored in a MySQL database.
  - The web dashboard displayed live sensor values and allowed manual control of the pump, fan, and lighting.
  - Historical data was visualized through graphs, providing users with a clear understanding of environmental trends over time.
- System Stability:
  - The Arduino system maintained stable wireless communication with the server during prolonged operation.

- Database storage ensured that sensor data was preserved for analysis and future improvements.

The following figures illustrate the sensor data collected in real time, providing insight into the system’s accuracy and responsiveness under test conditions.

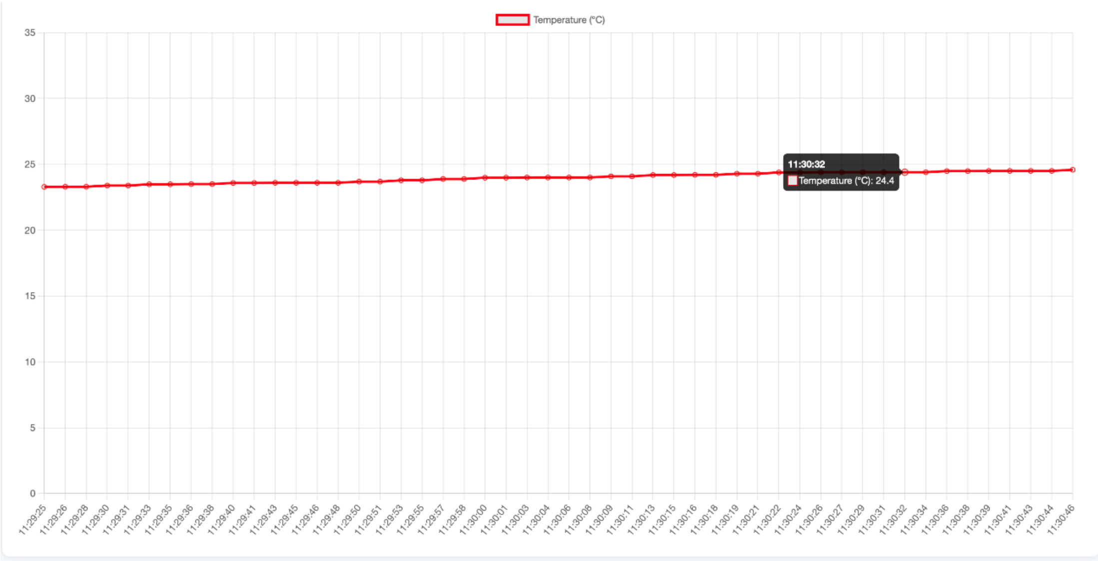


Figure 13 Temperature readings over time, remaining stable around 24°C, confirming consistent indoor environmental control.

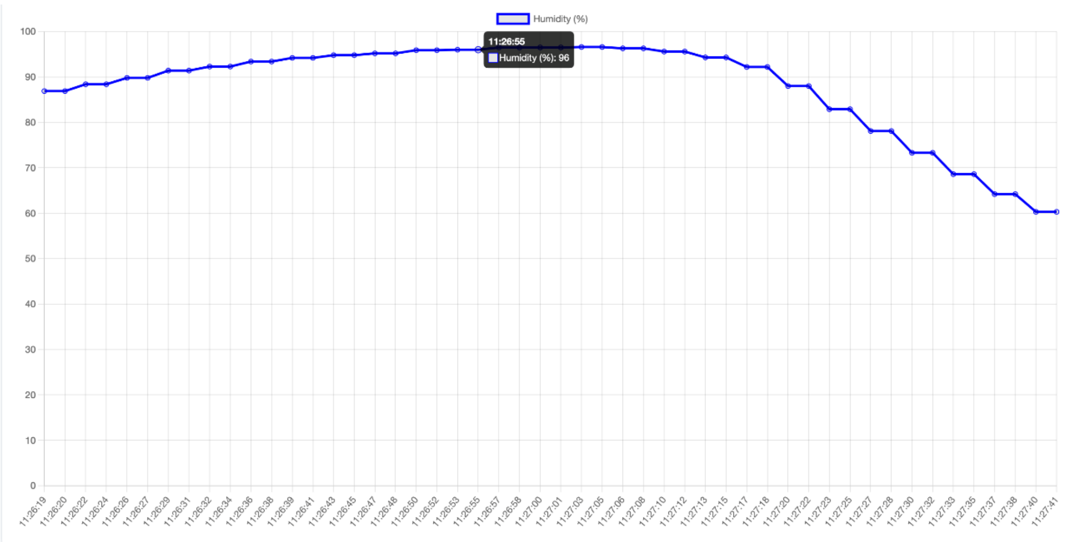


Figure 14 Humidity variation throughout the day, peaking at 96% and gradually dropping, demonstrating proper sensor sensitivity.

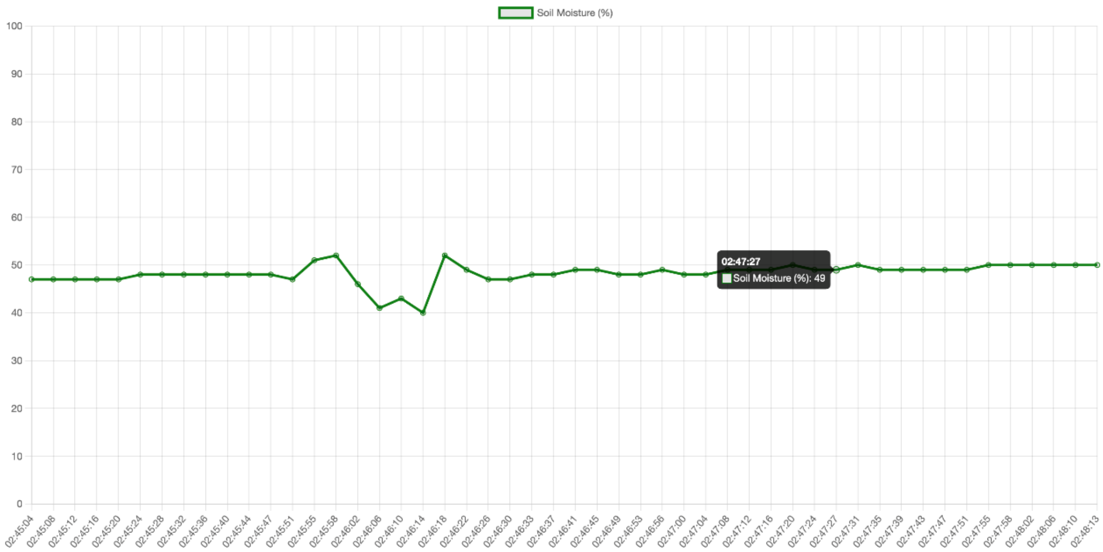


Figure 15 Soil moisture trends with slight fluctuations, showing pump activation in response to dry conditions and irrigation success.

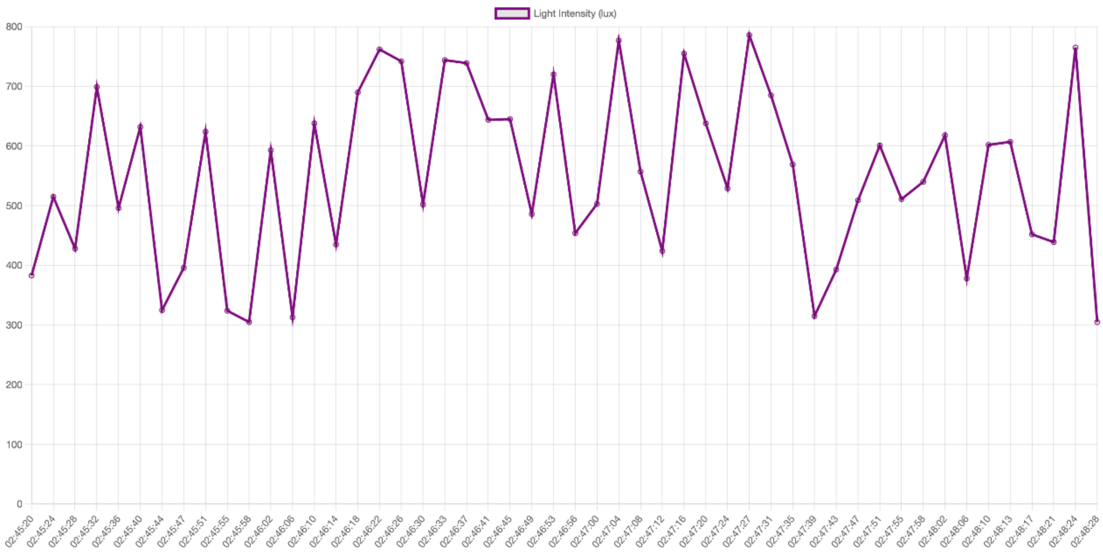


Figure 16 Light intensity data (lux), recorded via BH1750, capturing varying ambient light conditions.

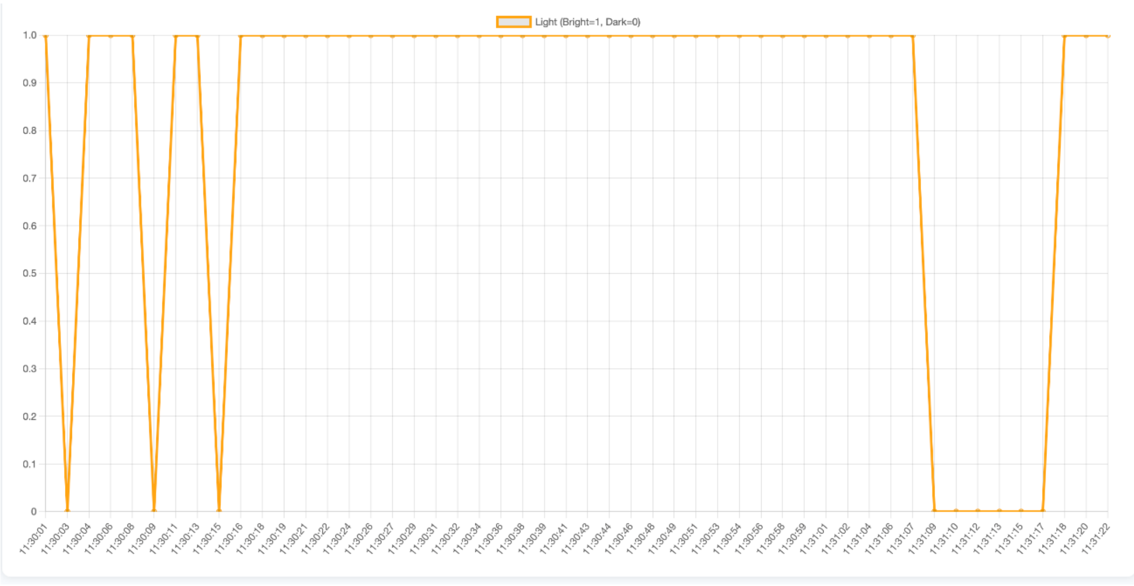


Figure 17 Binary light status using LDR sensor, with values representing Bright (1) and Dark (0), validating automatic detection for lighting control.

Overall, the prototype met the functional requirements and performed reliably under typical indoor environmental conditions.

6.2 Conclusion

This research successfully demonstrated the feasibility of creating an affordable, modular, and easy-to-use indoor smart farm system tailored for Mongolian households.

The system allows users to monitor and control essential environmental factors such as soil moisture, temperature, humidity, and light, both locally via an OLED display and remotely via a web-based platform.

The integration of automatic irrigation, ventilation, and lighting control provides significant convenience for users, reducing the need for constant manual supervision. Furthermore, by storing sensor data in a database and visualizing trends through a user-friendly interface, users can make better-informed decisions to optimize plant growth.

The prototype also highlights the potential for future expansions, such as:

- Integrating a CO/CO_2 sensor for air quality management.
- Adding water flow sensors for more precise irrigation control.
- Developing smart automation based on historical data trends.
- Incorporating mobile app support for easier remote management.

In conclusion, the design and implementation of this prototype provide a strong foundation for promoting indoor smart farming practices among Mongolian households, contributing to greater food security, self-sufficiency, and sustainable agriculture in the region.

7. DISCUSSION

The development and implementation of the indoor smart farm prototype provided valuable insights into the design challenges, practical limitations, and opportunities for future improvements in small-scale automated farming systems.

7.1 Challenges Encountered

Several challenges were faced during the research development:

- Sensor Calibration and Stability:
  Some sensors, particularly the soil moisture sensor, showed fluctuating readings depending on soil type and environmental noise. Repeated calibration and adjustment of thresholds were necessary to achieve reliable results.
- Wi-Fi Stability:
  Although the Arduino R4 Wi-Fi provided a convenient wireless solution, occasional network interruptions were observed, especially in areas with weak Wi-Fi signals. This caused delays in data transmission and occasional connection losses between the Arduino and the server.
- Power Supply Management:
  Ensuring that the mini pump, UV LED, fan, and Arduino could all be powered reliably from a shared source required careful planning. Voltage mismatches or insufficient current supply led to initial instability, which was resolved by adding separate regulated power supplies for actuators.
- Database Performance:
  Storing a large amount of sensor data over time increased the size of the MySQL database. Although manageable during the testing phase, long-term usage could lead to performance slowdowns if no data management (such as regular cleaning or archiving) is applied.
- OLED Display Integration:
  Updating the OLED display with new sensor readings while simultaneously managing server communications introduced timing challenges, requiring careful scheduling of tasks within the Arduino code.

7.2 Limitations of the Current Prototype

While the prototype met its main objectives, some limitations remain:

- Manual Lighting Control:
  Currently, UV lighting is activated manually through the web application. A fully automated light adjustment system based on real-time lux readings could improve the system's autonomy.
- Limited Air Quality Monitoring:
  Although future integration of CO or CO_2 sensors is planned, the current prototype does not monitor air quality, which limits the system's environmental control capabilities.
- Lack of Mobile Application Support:
  The system is currently accessible via a web browser. A dedicated mobile app could improve accessibility and offer real-time push notifications for alerts such as low moisture or high temperature.
- Limited Scalability for Larger Systems:
  The system is designed for small household farms. Scaling the system to manage multiple growing zones would require architectural adjustments, such as distributed sensor nodes and a more robust server infrastructure.

7.3 Future Work and Improvements

Based on the experience gained, several improvements are proposed for future versions of the smart farm system:

- Addition of Air Quality Sensors:
  Integrating CO and CO_2 sensors to monitor and automatically regulate indoor air quality.
- Full Automation of Lighting and Ventilation:
  Automating UV LED lighting based on BH1750 sensor readings and optimizing fan operation based on humidity as well as temperature.
- Mobile Application Development:
  Developing a cross-platform mobile application for better user accessibility and real-time alerts.
- Database Optimization:
  Implementing strategies for database management, such as periodic archiving of older sensor data to maintain system performance.

- Solar-Powered System:
  Investigating the use of small-scale solar panels and batteries to make the system more sustainable and independent of grid electricity.
- Machine Learning Integration:
  Analyzing historical data trends to develop predictive irrigation and lighting algorithms that optimize energy and water use automatically.

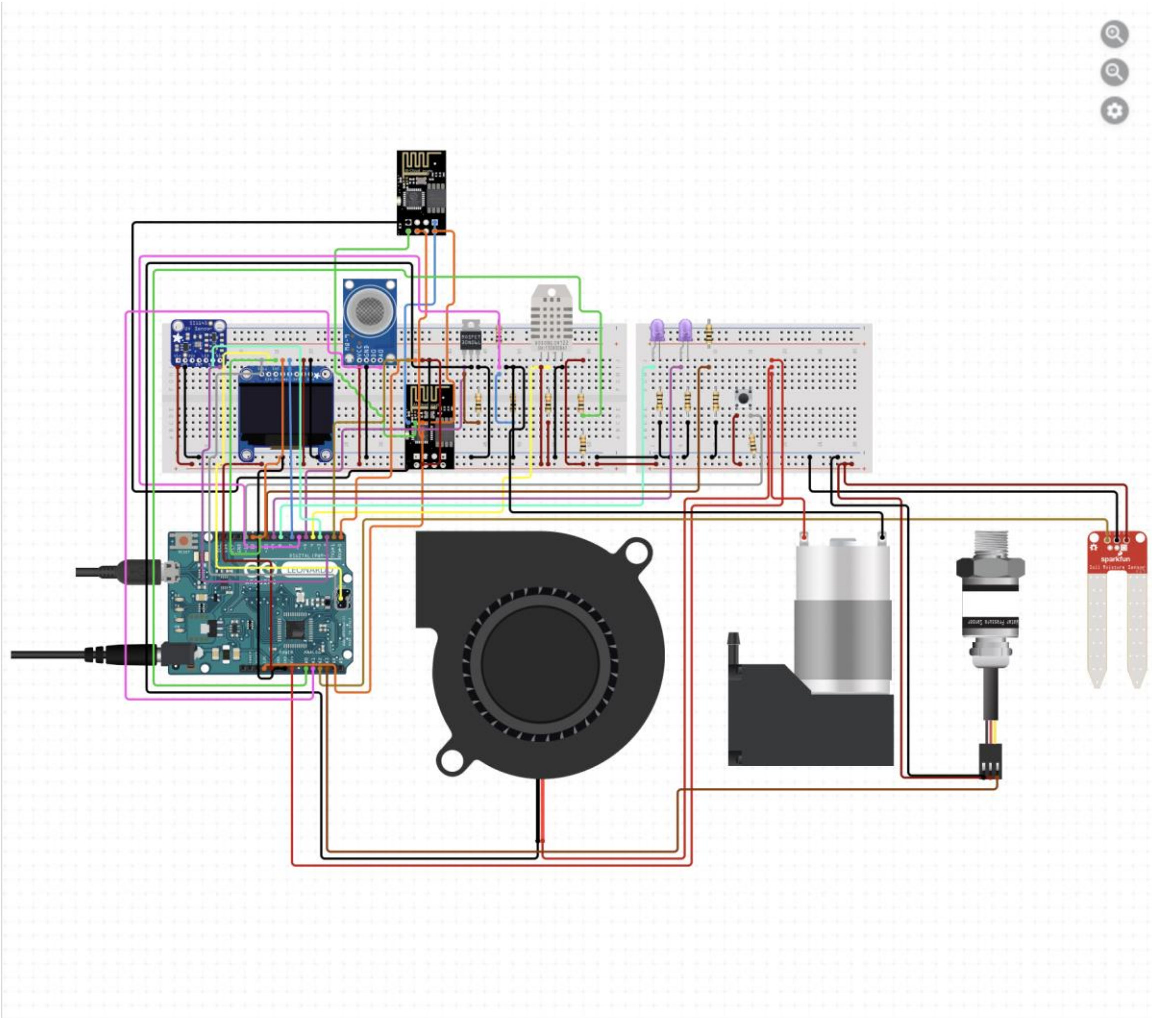
References

1. DHT22 Temperature and Humidity Sensor Datasheet. [Online].; 2023. Available from: https://www.adafruit.com/product/385.
2. BH1750 Ambient Light Sensor Datasheet. [Online].; 2023. Available from: https://www.adafruit.com/product/4681.
3. Arduino Documentation. [Online].; 2023. Available from: https://docs.arduino.cc.
4. MySQL Documentation. [Online].; 2023. Available from: https://dev.mysql.com/doc/.
5. Food and Agriculture Organization. [Online].; 2023. Available from: https://www.fao.org/in-action/scala/countries/mongolia/en.
6. Food and Agriculture Organization/ Production and food security. [Online].; 2000. Available from: https://www.fao.org/4/y3918e/y3918e04.htm.
7. T K. Plant Factory: Acedemic Press; 2015.
8. Sasireka Rajendran TD. Hydroponics. Exploring innovative sustainable technologies and applications across crop production, with Emphasis on potato mini-tuber cultivation. 2024; 10(5).
9. Sumit K. ResearchGate. [Online].; 2023. Available from: https://www.researchgate.net/publication/374616820_Hydroponics_and_Aeroponics_Advancement_in_Soilless_Cultivation.
10. Node.js Documentation. [Online].; 2023. Available from: https://nodejs.org.
11. Light Science Tech. [Online].; 2021. Available from: https://lightsciencetech.com/what-can-be-grown-in-a-vertical-farm/.
12. Wenyuan Wu LCRL. The role of light in regulating plant growth, development and sugar metabolism: a review. 2024; 15.
13. Chui Eng Wong ZWTLSHY. Seeing the lights for leafy greens in indoor vertical farming. 6th ed.: Science Direct; 2020.

APPENDICES

Appendix A – Schematic Diagram

- Full circuit diagram showing connection between Arduino R4 Wi-Fi, soil moisture sensor, DHT22, LDR, BH1750, OLED display, mini pump via relay, UV LED, fan, and power supply.
- Label each pin connection clearly.

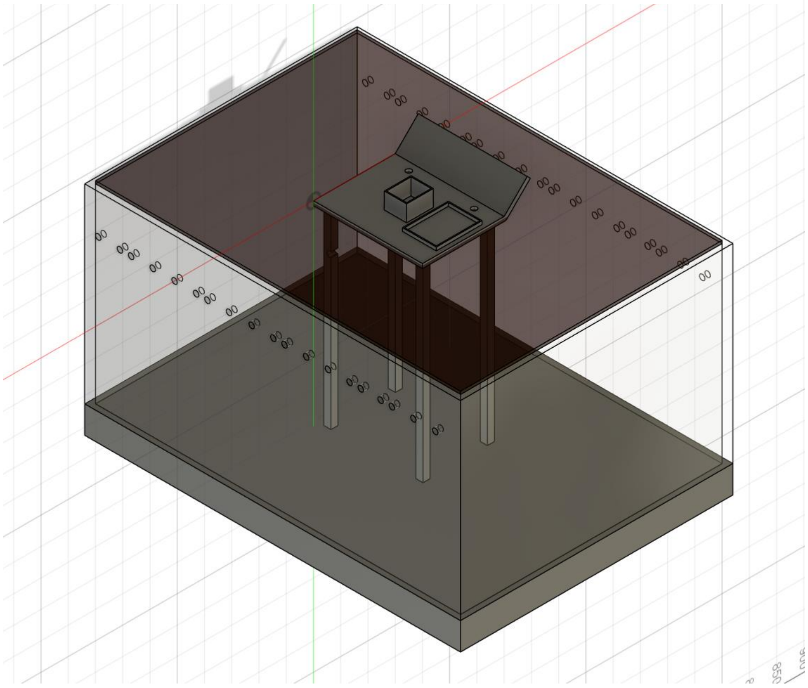


Appendix A. Future Work and Improvement Full Circuit Diagram

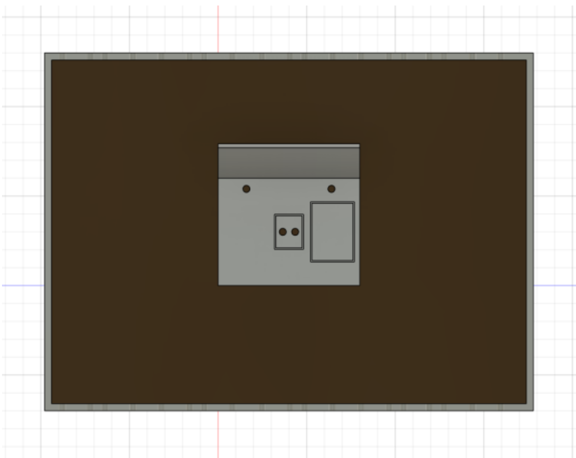
Appendix B – 3D Design Drawing

- 3D CAD model of your indoor smart farm base module.
- Show design views (top, side, isometric) if possible.
- Mention key dimensions or special features.

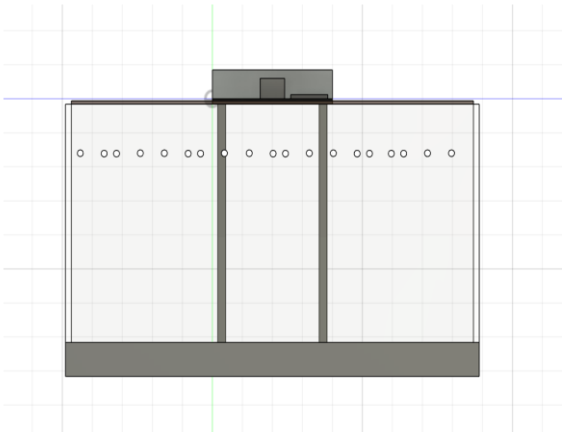
Appendix B.1. Isometric View of 3D Design



Appendix B.2. Top View of 3D Design



Appendix B.3. Side View of 3D Design



Appendix C – Arduino & Website Code

- Important Arduino sketches:
    - Main program setup, sensor reading, control logic, Wi-Fi communication.
    - OLED display updating code section.

Appendix C.1. Arduino Code Part 1

sketch_apr16a.ino

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <DHT.h>
#include <WiFiS3.h>
char ssid[] = "Javkhlan";
char pass[] = "123456789";
char server[] = "172.20.10.4";
int port = 3000;
WiFiClient client;
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);
#define DHTPIN 2
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);
#define SOIL_PIN A0
#define UV_PIN 6
#define RELAYPIN 3
#define LDR_PIN 4
String mode = "auto";
String pumpState = "OFF";
String uvState = "OFF";
float soilThreshold = 40;
float tempThreshold = 26;
float humidThreshold = 60;
void setup() {
  Serial.begin(9600);
  dht.begin();
  pinMode(RELAYPIN, OUTPUT);
  digitalWrite(RELAYPIN, HIGH);
  pinMode(LDR_PIN, INPUT);

  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {…
  }
  display.clearDisplay();
  display.setTextSize(0.8);
  display.setTextColor(SSD1306_WHITE);
```

Appendix C.2. Arduino Code Part 2

sketch_apr16a.ino

```
  display.setCursor(0, 0);
  display.println("Connecting WiFi...");
  display.display();
  while (WiFi.begin(ssid, pass) != WL_CONNECTED) {⋯
  }
  Serial.println("WiFi connected");
  display.clearDisplay();
  display.setCursor(0, 0);
  display.println("WiFi connected");
  display.display();
  delay(1000);
}
void loop() {
  float humidity = dht.readHumidity();
  float temperature = dht.readTemperature();
  int ldrValue = digitalRead(LDR_PIN);
  String lightStatus = (ldrValue == LOW) ? "Bright" : "Dark";
  int fakeLux = random(300, 800); // Simulate between 300 lux and 800 lux
  float luxThreshold = 500;  // default threshold
  int soilAnalog = analogRead(SOIL_PIN);
  float soil = map(soilAnalog, 1023, 0, 0, 100);

  if (mode == "auto") {
    if (soil < soilThreshold) {⋯
    } else {⋯
    }
  }
  pumpState = (digitalRead(RELAYPIN) == LOW) ? "ON" : "OFF";
  display.clearDisplay();
  display.setCursor(0, 0);
  display.println("Smart Farm Monitor");
  display.setCursor(0, 8);
  display.print("Temp: "); display.print(temperature, 1); display.println(" C");
  display.print("Humid: "); display.print(humidity, 1); display.println(" %");
  display.print("Soil: "); display.print(soil, 1); display.println(" %");
  display.print("Light: "); display.println(lightStatus);
  display.print("Mode: "); display.println(mode);
  display.print("Pump: "); display.println(pumpState);
```

Appendix C.3 Arduino Code Part 3

sketch_apr16a.ino

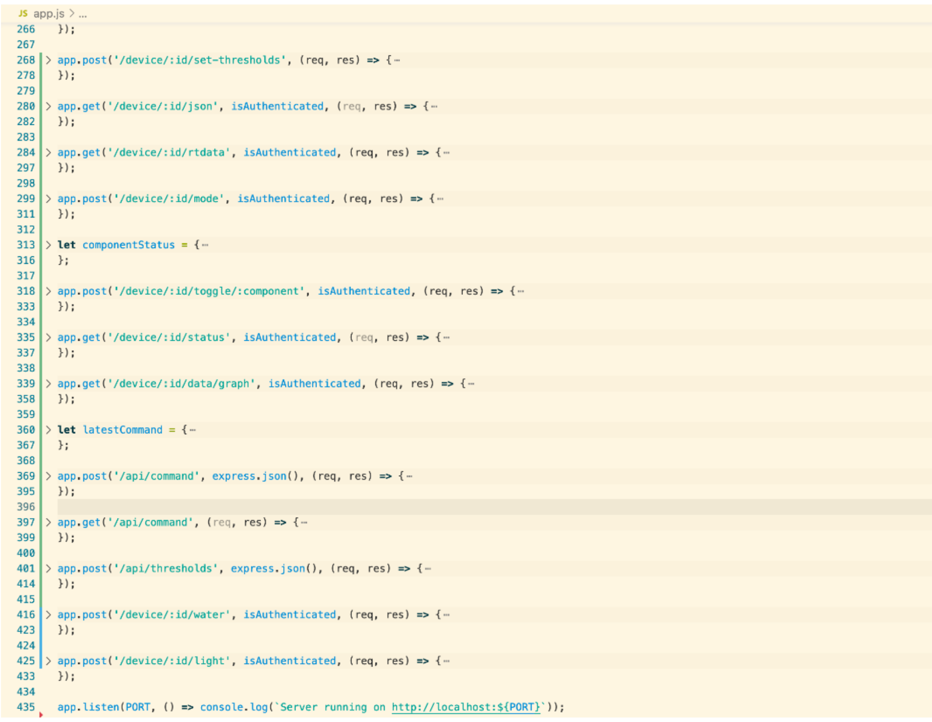
```
  display.print("Mode: "); display.println(mode);
  display.print("Pump: "); display.println(pumpState);
  display.print("UV LED: "); display.println(uvState);
  display.display();
  if (client.connect(server, port)) {⋯
  } else {⋯
  }
  client.stop();
  if (client.connect(server, port)) {
    client.println("GET /api/command HTTP/1.1");
    client.println("Host: " + String(server));
    client.println("Connection: close");
    client.println();
    String response = "";
    while (client.connected() || client.available()) {⋯
    }
    client.stop();
    int jsonStart = response.indexOf('{');
    if (jsonStart != -1) {
      String json = response.substring(jsonStart);
      Serial.println("📡 JSON from web: " + json);

      if (json.indexOf("\"mode\":\"manual\"") != -1) {⋯
      } else if (json.indexOf("\"mode\":\"auto\"") != -1) {⋯
      }
      if (json.indexOf("\"soilThreshold\"") != -1) {⋯
      }
      if (mode == "manual") {
        if (json.indexOf("\"pump\":\"on\"") != -1) {⋯
        } else if (json.indexOf("\"pump\":\"off\"") != -1) {⋯
        }
      }
      Serial.println("🧠 Mode: " + mode + " | 💧 Pump: " + pumpState);
    } else {⋯
    }
  }
  delay(3000);
}
```

- Node.js backend code: Server initialization, database communication routes, API for frontend.

Appendix C.4 Backend Code and Code Structure


```javascript
import express from 'express';…
const app = express();
const PORT = 3000;
app.set('view engine', 'ejs');
app.use(express.static('public')); // ensure product image is in public/images/
app.use(bodyParser.urlencoded({ extended: false }));
app.use(…
);
/*import { SerialPort } from 'serialport';…
function fetchChartData() {…
}

let latestData = { temp: '--', humid: '--', light: '--', soil: '--', lux:'--'};

app.post('/api/upload', (req, res) => {…
});

function isAuthenticated(req, res, next) {…
}

app.get('/', (req, res) => {…
});

app.get('/login', (req, res) => {…
});

app.post('/login', (req, res) => {…
});

app.get('/register', (req, res) => {…
});

app.post('/register', (req, res) => {…
});

app.get('/dashboard', isAuthenticated, (req, res) => {…
});

app.get('/logout', (req, res) => {…
});

app.post('/add-device', isAuthenticated, express.json(), (req, res) => {…
});

app.get('/device/:id', isAuthenticated, (req, res) => {…
```

Appendix C.5. Backend Code Part 2




```javascript
});

app.post('/device/:id/set-thresholds', (req, res) => {…
});

app.get('/device/:id/json', isAuthenticated, (req, res) => {…
});

app.get('/device/:id/rtdata', isAuthenticated, (req, res) => {…
});

app.post('/device/:id/mode', isAuthenticated, (req, res) => {…
});

let componentStatus = {…
};

app.post('/device/:id/toggle/:component', isAuthenticated, (req, res) => {…
});

app.get('/device/:id/status', isAuthenticated, (req, res) => {…
});

app.get('/device/:id/data/graph', isAuthenticated, (req, res) => {…
});

let latestCommand = {…
};

app.post('/api/command', express.json(), (req, res) => {…
});

app.get('/api/command', (req, res) => {…
});

app.post('/api/thresholds', express.json(), (req, res) => {…
});

app.post('/device/:id/water', isAuthenticated, (req, res) => {…
});

app.post('/device/:id/light', isAuthenticated, (req, res) => {…
});

app.listen(PORT, () => console.log(`Server running on http://localhost:${PORT}`));
```

- Frontend EJS code: Real-time data display, manual control page, graph generation logic.

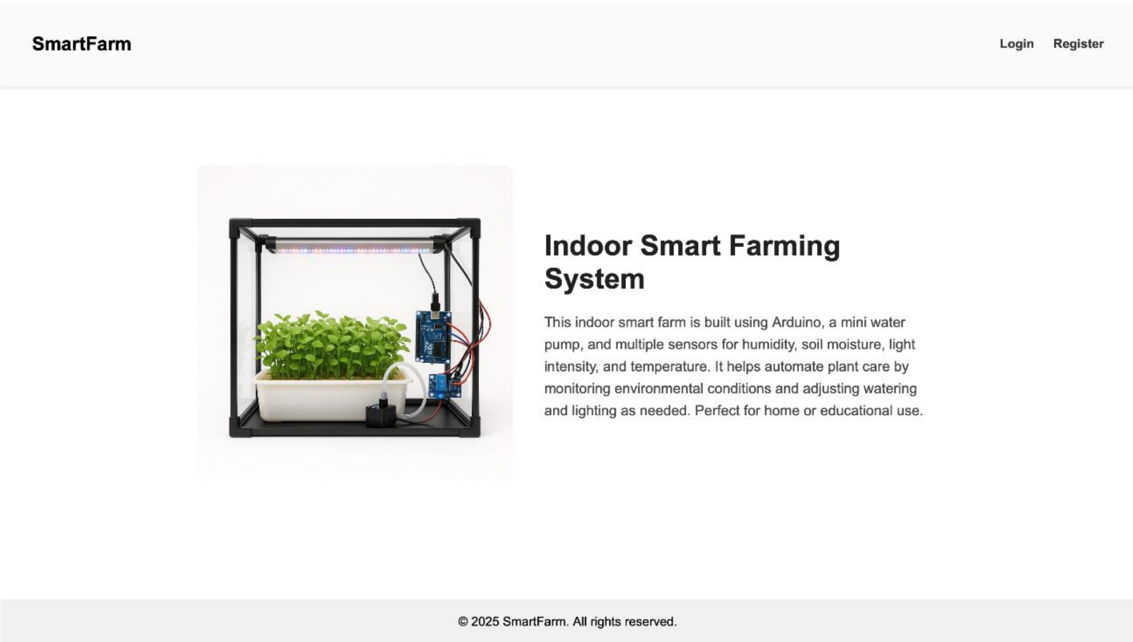
Appendix C.6. Frontend EJS Code

views > rtdata.ejs > html > body > header

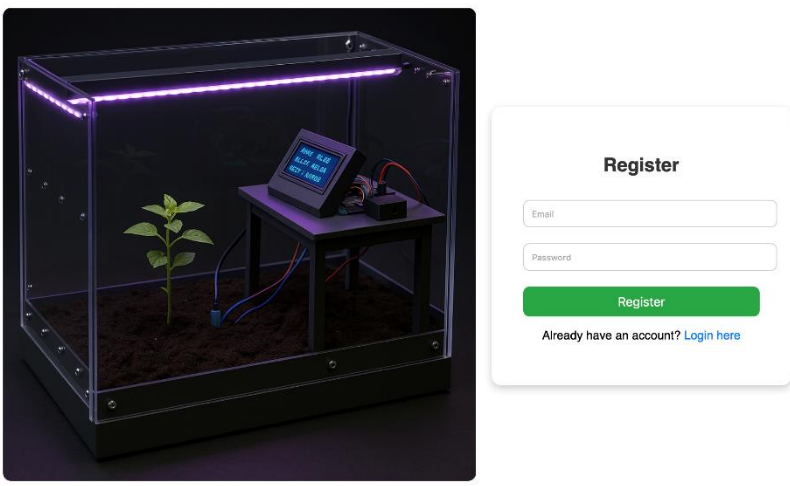
```
<!DOCTYPE html>
<html>
<head>⋯
</head>
<body>
  <header>⋯
  </header>

  <div class="container">⋯
  </div>

  <script>
    let chartTemp, chartHumid, chartLight, chartSoil, chartLux;

    function toggleChart(type) {⋯
    function applyPlantThresholds() {⋯

    function sendPumpCommand(state) {⋯

    function sendLightCommand(state) {⋯
    let pumpStatus = "<%= componentStatus && componentStatus.pump ? 'on' : 'off' %>";

    function updatePumpButton() {⋯

    function updateLightButton() {⋯

    function togglePump() {⋯

    let lightStatus = "<%= componentStatus && componentStatus.light ? 'on' : 'off' %>";

    function toggleLight() {⋯

    updateLightButton(); // initial load

    updatePumpButton(); // Initial label update on page load
    function fetchChartData() {⋯


    setInterval(fetchChartData, 3000);
    fetchChartData();

    setInterval(() => {⋯
  </script>
</body>
</html>
```

Appendix D – Website Preview

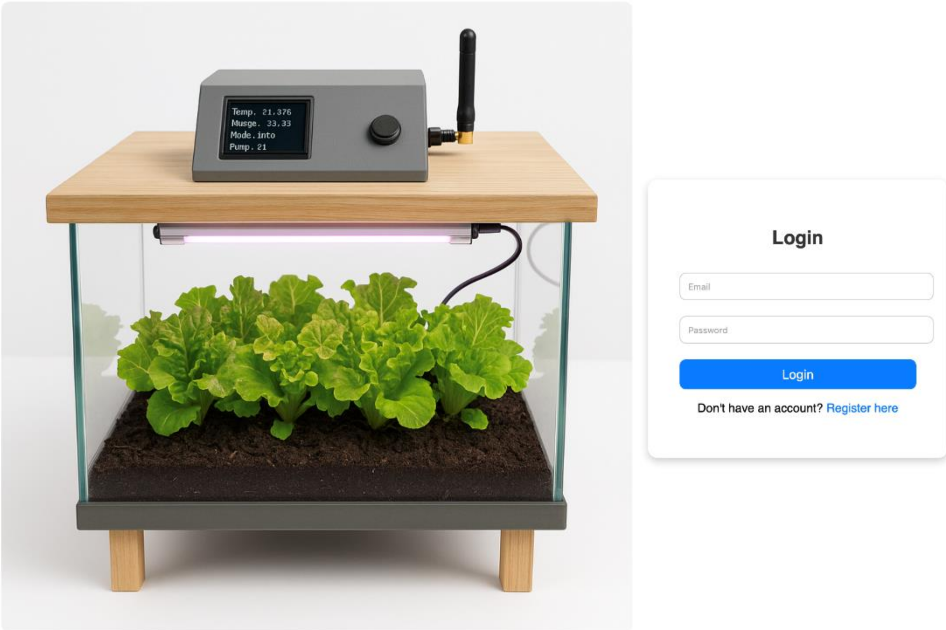
Appendix D.1 Homepage Preview



Appendix D.2 Register Page Preview



Appendix D.3 Login Page Preview



Appendix D.4 Dashboard & Add Device Preview