faculty of Industrial and Energy Technology
Information Technology Program

AgriStation Project

Graduation Project
Spring Semester
Academic Year 2024/2025

By:

AGRIVOLT SQUAD

Submitted By:

	Hardware Team	
1	عبد الرحمن محمد السيد	2320335
2	أدهم احمد حسنين احمد	2320083
3	حسن إبراهيم محمد	2320195
4	محمد صابر احمد	2320559
5	محمد ناجي احمد	2320516
6	شهد أحسن عبد الله	2320288
7	علا عبد السلام احمد	2320384
8	فاطمة محمد السيد	2320436

Table 1 Hardware Team

	Software Team	
1	محمد عبد اللطيف إبراهيم	2320527
2	يوليوس جرجس نجيب	2320775
3	مريم عدنان مساح	2320602
4	ايريني رأفت نعيم	2320130
5	شروق احمد ابراهيم	2320276
6	اسراء عطية عبد المجيد	2320096
7	اسراء مسعد حميدة	2320100
8	الشيماء احمد حسن	2320115
9	هدي فتحي محمد	2320779

Table 2 Software Team

Under Supervision of: [Dr. Eman Shawky]

Under Supervision of: [Eng. Esraa Mohsen]

ABSTRACT:

The Smart AgriStation is a modern agricultural project that aims to transform traditional farming by combining smart technologies, renewable energy, and real time automation. The system integrates a smart greenhouse, a renewable energy station powered by solar panels, and a sensor-based irrigation system, all controlled through a centralized web based interface. At the core of the system is the ESP32 microcontroller, which collects real-time environmental data using sensors such as the DHT11 (for temperature and humidity), soil moisture sensors, LDR (light sensor), and ultrasonic sensor for water level detection. Based on the sensor readings, the microcontroller makes automated decisions such as activating irrigation when soil moisture drops below a threshold or turning on the fan when temperature exceeds the optimal range. The system is supported by renewable energy, using solar panels to power operations. Data is transmitted and stored using Firebase Realtime Database,

Figure 1 Abstract

allowing seamless communication between the hardware and the user interface. Figure 1 Abstract A responsive web interface that allows users to monitor live sensor data, control components remotely, and receive real-time system alerts. The interface provides practical and user-friendly experience, making the system accessible even to non-technical users. This project demonstrates the effective use of IoT (Internet of Things), embedded systems, and clean energy to build a self-sustaining smart agriculture station. By automating key processes, it enhances productivity, minimizes resource waste, and promotes sustainable farming practices. It reflects the future of agriculture smart, data-driven, and eco-friendly.

ACKNOWLEDGMENT:

First and foremost, we thank **Allah** for granting us the strength, patience, and determination to accomplish this project.

We also extend our deepest gratitude to the **administration of our university** for providing us with the opportunity to bring this project to life. This chance to explore our ideas, experiment, and grow through practical experience has been invaluable.

We would like to give special thanks to our respected professors:

- **Dr. Osama El-Nahas**
- **Dr. Eman Shawky**
- **Eng. Esraa Mohsen**

for their exceptional support, guidance, and dedication. Their expertise, encouragement, and generosity with their time and resources were instrumental in transforming our vision into reality. Without their guidance, this project would not have been possible.

We also extend our heartfelt appreciation to the **President of the University, Prof. Dr. Mohamed Morsi Al-Gohary**, who has been like a father to us throughout our academic journey. His unwavering support and belief in our potential have inspired and motivated us every step of the way.

Finally, we would like to express our deepest gratitude to everyone who contributed to the success of this project—our families, friends, and colleagues—for their endless support, encouragement, and patience throughout this journey. This achievement is as much theirs as it is ours.

LIST OF CONTENTS

Table 3 Content

LIST OF ABBREVIATIONS

Abbreviation	Full Name
%RH	Relative Humidity Percentage
°C	Degrees Celsius
AC	Alternating Current
ADC	Analog to Digital Converter
BH1750	BH1750 Digital Ambient Light Sensor
BMS	Battery Management System
CCS811	Carbon Dioxide and Total Volatile Organic Compound Sensor
Chart.js	Chart JavaScript Library (open-source charting library)
CO_2	Carbon Dioxide
DC	Direct Current
DHT11	Digital Humidity and Temperature Sensor (Model DHT11)
DHT22	Digital Humidity and Temperature Sensor (Model AM2302)
ESP32	Espressif Systems Platform 32-bit
GND	Ground
Gerber	Gerber Files (standard file format used for PCB manufacturing)
GPIO	General Purpose Input/Output
I^2C (or I2C)	Inter-Integrated Circuit (2-wire serial communication protocol)
JLCPCB	JLC Printed Circuit Boards (PCB manufacturing service)
LDR	Light Dependent Resistor
LED	Light Emitting Diode
MHz	Megahertz
MPPT	Maximum Power Point Tracking
NiCd	Nickel-Cadmium Battery
NiMH	Nickel-Metal Hydride Battery
PCB	Printed Circuit Board
PCBWay	PCBWay (PCB manufacturing service, brand name)
ppb	Parts Per Billion
Proteus	PCB Design and Simulation Software
PV	Photovoltaic
PWM	Pulse Width Modulation
RF	Radio Frequency
SCL	Serial Clock (I2C Communication Line)
SDA	Serial Data (I2C Communication Line)

Table 4 Abbreviations

LIST OF TABLES

LIST OF FIGURES

Chapter 1: Green House

The greenhouse is the heart of the Smart AgriStation a controlled environment designed to provide optimal conditions for plant growth regardless of external weather changes. In this chapter, we dive into how our greenhouse combines smart technology with traditional farming principles to create a space where temperature, humidity, and lighting are monitored and adjusted automatically. By using sensors and automation, we can simulate the best possible climate for crops to thrive, while also conserving energy and water. This makes farming not only smarter but also more sustainable and future ready.

in recent years, greenhouses have become essential for improving crop production, especially in areas with unpredictable weather or limited natural resources. In the Smart AgriStation, the greenhouse isn't just a physical structure—it's an intelligent environment where technology meets nature to create ideal growing conditions for plants.

Purpose of the Greenhouse:

The greenhouse in the Smart AgriStation serves as a **controlled agricultural environment** that enables consistent and efficient crop growth regardless of changing weather conditions. Traditional farming often suffers from unpredictable factors like heat waves, heavy rain, or droughts — but the greenhouse solves this by creating a stable, monitored microclimate. With the help of sensors and an ESP32 microcontroller, the system automatically manages temperature, humidity, soil moisture, lighting, and air quality. For example, when the soil dries out, the irrigation system turns on automatically, and when it gets too hot, ventilation fans activate. These real-time adjustments allow plants to grow in ideal conditions, improving both health and yield.

The greenhouse also supports **sustainable farming**, using **solar energy** to power its components and a **drip irrigation system** to conserve water. This makes it not just a space for growing crops, but a model for eco-friendly, smart agriculture that can be applied in areas with limited resources or extreme climates.

How the Greenhouse Works (Automation):

Using sensor readings, the ESP32 is programmed to automatically respond to environmental changes. Here's how:

- If the temperature rises above a set limit, the ventilation fan turns on.
- If the soil is dry, the system activates the irrigation pump to water the plants.

- If the water tank is low, the system can notify the user through the website.

This creates a fully automated system that reacts in real-time to maintain perfect conditions for plant growth.

Powering the Greenhouse:

To support sustainability, the greenhouse and its components are powered using **solar panels** installed on the AgriStation. This reduces energy costs and aligns with the goal of using **renewable energy sources** in smart farming.

1.1. Steps of Building the Greenhouse:

1. **First step of building:** We started by preparing the frame, cutting and assembling the metal pipes According to the greenhouse dimensions. During this step, we made sure structure was stable and properly aligned.

2. **Fixing the frame to the ground:** After cutting and preparing the pipes, we fixed Them firmly into the ground using cement to ensure stability and durability, Especially in case of wind or changing weather.

3. **Preparing the outer cover (Plastic Sheet):** We used 13.5 meters of glass-textured plastic Sheet to cover the greenhouse, as it provides good insulation against water and UV rays. It was stretched over the structure after installing the top triangular roof section.

4. **Soil Preparation and Planting:** After completing the structure and covering, we Prepared the soil inside the greenhouse for planting. The area was cleaned and leveled, and organic compost along with phosphate fertilizer was added to improve soil fertility. Planting rows were created, and we planted eggplant and pepper seedlings, as these crops thrive in warm and humid environments typically maintained inside greenhouse.

5. **Irrigation System Installation:** We installed the water pump and drip irrigation pipes along the planting rows to ensure that water reaches the roots directly. The irrigation system was connected to the soil moisture sensor, enabling automatic watering based on real-time soil moisture levels.

6. **Ventilation and Lighting:** A motorized window was installed to allow automatic ventilation when the internal temperature exceeds a set limit. This system is controlled by a temperature sensor. Additionally, the greenhouse was oriented to maximize natural sunlight exposure throughout the day, ensuring optimal lighting for plant growth.

7. **System Calibration and Testing:** After installation, we tested all components to ensure they were functioning correctly. We calibrated sensors like soil moisture and DHT22, checked the water flow from the irrigation system, and monitored automatic responses like fan activation and pump control. This testing phase helped fine-tune performance and eliminate bugs before full deployment.

8. **Dashboard Integration:** We connected the ESP32 to our web dashboard to display real-time sensor data (like temperature, humidity, and soil moisture). This allowed remote monitoring and control from any device. It made the greenhouse smarter, more interactive, and user-friendly.

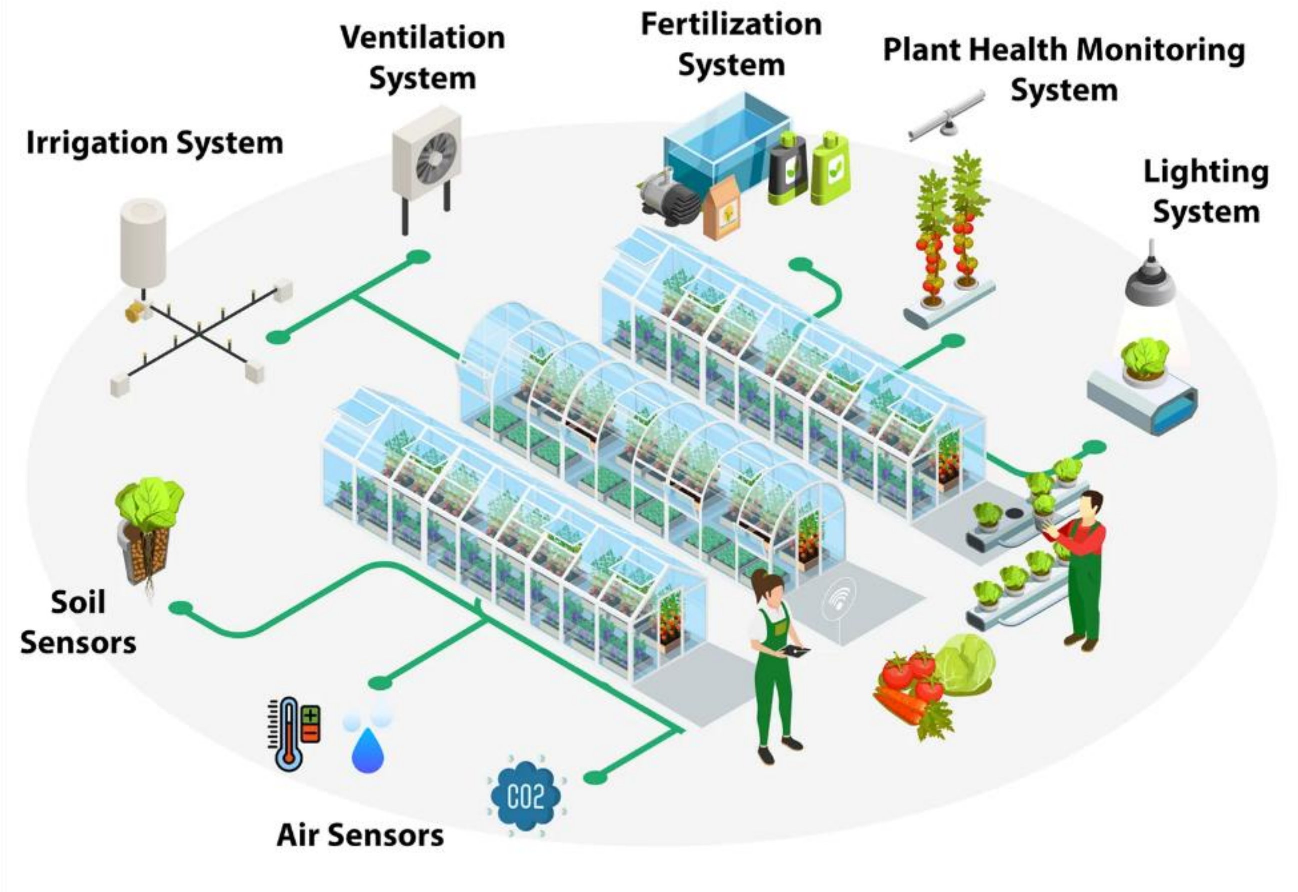

Figure 2 Steps of Building the Greenhouse

1.2. Components Used in the Greenhouse:

To build and operate the smart greenhouse, we used the following components:

- ESP32 microcontroller (2 pieces)
- DHT22 Sensors (2 pieces)
- BH1750 Light Sensors (2 pieces)
- CCS811 CO_2 Sensors (2 pieces)
- Soil Moisture Sensors with Relay (2 sets)
- Water Pump
- Electric Valve
- DC Motor
- Motor Driver Module
- 4-Channel Relay Module
- Step-down Voltage Regulator
- Solar Panel
- Plastic Sheets (13.5 meters)
- Metal Pipes (5 Pieces of iron. pipes)
- Iron. Mesh + Cement (1.5 bag)
- Compost & Phosphate Fertilizer
- Water Tank
- Drip lrrigation Hoses
- Cooling Unit
- Lock & Control Board
- Rechargeable Batteries

Servo Motor (Full Metal Gear)
MG995 TowerPro 180 Degree

Chapter 2: Sensors

In any smart agricultural system, sensors are the eyes and ears of the entire operation. They collect real-time data from the environment, turning physical conditions—like soil moisture, temperature, humidity, and light intensity—into digital information that our system can understand and act on. In the Smart AgriStation, sensors play a vital role in monitoring and controlling farm activities with high accuracy and efficiency.
By using sensors, we reduce the need for guesswork and manual observation. Instead, decisions are made based on accurate, up-to-date information. This helps us optimize irrigation, manage energy use, and create the ideal environment for plant growth inside the greenhouse. It's not just about automation—it's about making agriculture smarter, more sustainable, and responsive to changing conditions.
In this chapter, we'll explore the key sensors used in the Smart AgriStation, how each one works, where it's placed, and how the data it collects is used to control systems through embedded and IoT technologies.

2.1. DHT22 (Temperature and Humidity) Sensor:

A digital sensor that measures air temperature and relative humidity with high accuracy, providing data as a direct digital signal.
- How does it work?
It has a built-in temperature and humidity sensor and uses a simple digital communication protocol with the microcontroller.
- ESP32 Connection:
- The DATA pin is connected to GPIO12.
- Requires a 3.3V power supply and ground.

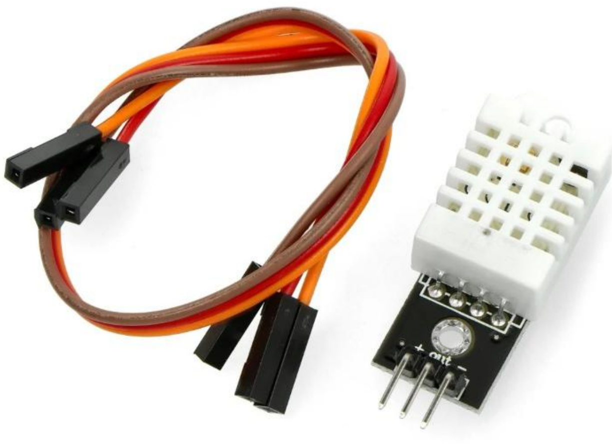

Figure 3 DHT22 sensor

- Use in the project:
Measuring the surrounding environment to assess the atmospheric conditions around plants inside a greenhouse.

Parameters of the temperature and humidity sensor DHT22

- Sensor AM2302 (DHT22)
- Supply voltage: from 3.3 V to 5.5 V

- Average current consumption: 0.2 mA
- Does not require additional external components
- 3 x female-female wires – included
- Module dimensions: 44 x 16 x 10 mm
- **Temperature**
    - Measurement range: -40 to +80°C
    - Resolution: 8-bit (0.1 °C)
    - Accuracy: ± 0.5 °C
    - Response time: on average – 2 seconds
- Humidity:
    - Measuring range: 0 – 99.9 %RH
    - Resolution: 8-bit (±0,1 %RH)
    - Accuracy: ±2 %RH*
    - Response time: on average – 2 seconds

2.2. Soil Moisture Sensor with Relay

An analog sensor that measures soil moisture using the soil resistance between two electrodes. Resistance decreases with increasing humidity.

- How does it work?

It outputs an electrical voltage inversely proportional to the soil moisture content.

- ESP32 Connection:
- Analog output connected to GPIO34 (ADC input)
- Relay connected to GPIO14 to control the on/off devices such as a water pump based on soil conditions

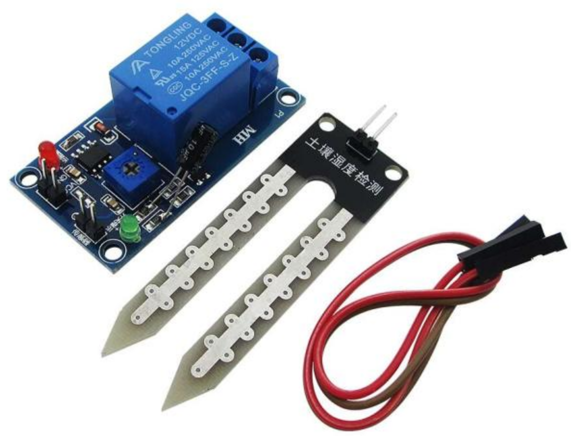

Figure 4 Soil Moisture Sensor

- Project Use:

Monitoring soil moisture and automatically turning on irrigation systems when they dry out.

Electrical Parameters

Supply voltage: 12VDC input current: greater than 100mA

Load: 250V 10A AC or 30V 10A DC (less than this range of current can be used)

3.3. CCS811 Sensor (CO_2)

A smart sensor for measuring air quality, measuring the concentration of carbon dioxide (CO2) and TVOC (volatile organic compounds).

- How does it work?

Uses I2C gas sensing technology to read air pollution levels.

- ESP32 Connection:
- SDA → GPIO21
- SCL → GPIO22
- Operates on 3.3V
- Project Use:

Monitoring air quality inside a greenhouse to ensure a healthy environment for plants.

Figure 5 CCS811 Sensor

Electrical Parameters

- Operating Voltage: 3.3V~5.5V
- Warm-up Time: <15s
- IIC Address: 0x5A (in default) / 0x5B
- Operating Temperature: -40°C~85°C
- Operating Humidity: 0ppb~1100ppb
- Dimension: 16*20mm / 0.63*0.79 "

2.4. BH1750 Light Intensity Sensor

A digital light sensor that measures ambient light intensity in lux via the I2C protocol.

- How does it work?

It converts light intensity into a digital signal that can be read via I2C.

- Connecting the ESP32:
- SDA → GPIO21
- SCL → GPIO22
- VCC → 3.3V
- Use in the project:

Figure 6 BH1750 Light Intensity Sensor

Measuring light level to adjust greenhouse lighting or set light exposure times.

Specifications

- Supply voltage: from 3.3 V to 5 V
- Able to detect ambient brightness and light intensity
- Adjustable sensitivity (via blue digital potentiometer adjustment)
- Operating voltage 3.3V-5V
- With fixed bolt hole for easy installation
- Power indicator (Red) and the digital switch output indicator (Green)
- Dimensions (Without Pins): 32 x 14 mm

2.5. Relay 4 Channel

A relay is an electronically controlled electrical switch that turns on or off electrical devices such as a pump or valve.

- How does it work?

When an electrical signal is given to the relay from the ESP32, it turns on or off the electrical circuit. - ESP32 Connection:

- Pump relay connected to GPIO27
- Valve relay connected to GPIO26
- Project Use:

Turn the pump and valve on and off to automatically control plant watering.

Figure 7 Relay to control pump

Specifications:

Table 5 Relay to control pump and

Specification	Details
Relay Input Voltage:	12V DC (Direct Current)
Relay Output Voltage:	220V AC (Alternating Current)
Relay Output Current:	10A max at 220V AC
Frequency:	433 MHz RF
Number of Channels:	2 Channels (controls two devices independently)
Control Range:	50-100 meters
Control Modes:	Momentary Mode (active while button is pressed)

Toggle Mode::	(on/off with each press)
Remote Control Power:	Battery-operated 12V or 23A battery
Relay Module Dimensions:	Varies by manufacturer (compact size for easy installation)
Working Temperature:	-20°C to 70°C
Security Code:	Learning Code Function (for secure pairing)
Power Consumption:	Low power consumption (operates on 12V DC)
Installation Requirements:	Requires 12V DC power supply, suitable for 220V AC devices

2.6. DC Motor with Driver (Motor Driver + DC Motor)

A DC motor used to control mechanical motion, with a driver to regulate operation, speed, and direction.
- How does it work?
It uses PWM signals to control motor speed and a digital signal to change the direction of rotation.
- ESP32 Connection:
- PWM speed connected to GPIO25
- Direction of rotation connected to GPIO33
- Project Use:
Move mechanical parts in the greenhouse, such as opening/closing ventilation windows or operating the irrigation system.

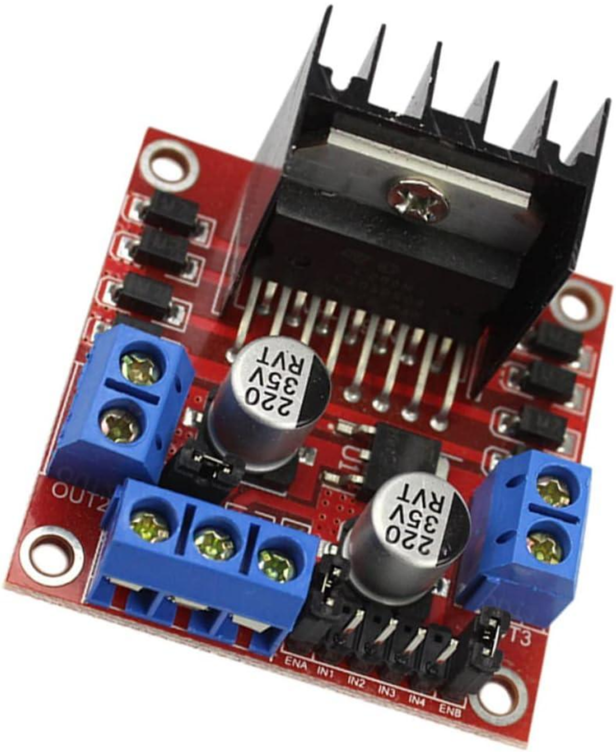

Figure 8 DC Motor with Driver

Chapter 3: ESP Programming

The brain behind our smart system lies in the programming of the ESP microcontroller. This chapter explains how the ESP module is programmed to receive sensor data, process it, and control outputs such as irrigation pumps, lights, or fans. We'll walk through the logic, code structure, and communication protocols used—especially how ESP interacts with both hardware and cloud systems to enable real-time decisions and remote control. With efficient coding and lightweight architecture, ESP programming is what turns our Smart AgriStation from a concept into a fully functional system.

Figure 9 ESP32

Figure 10 ESP32 Pinout

3.1 ESP Simulation:

This chapter presents the electronic simulation of the Smart Greenhouse system, created using EasyEDA. The schematic simulates how all electronic components — including sensors, relays, motors, and the ESP32 microcontroller — are connected and powered to enable smart automation within the greenhouse. This simulation was used as a prototype before assembling the real hardware, helping us validate our design, check wiring accuracy, and avoid short circuits or misconfigurations.

The central controller used is the **ESP32-WROOM-32D**, chosen for its powerful processing capabilities, built-in Wi-Fi, and rich GPIO options.

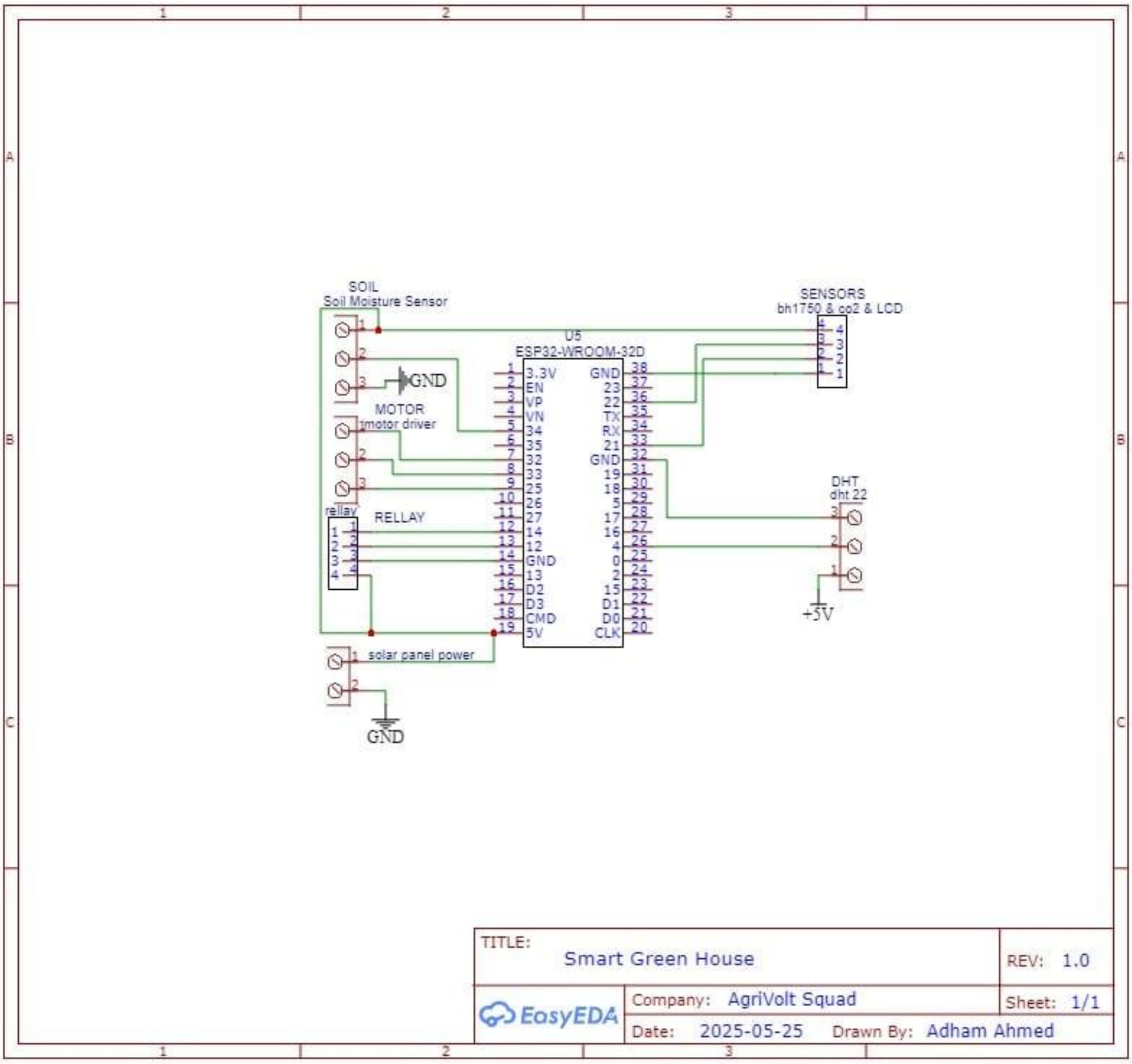

Figure 11 ESP Simulation

Purpose of the Simulation

- All sensors and actuators are correctly connected to the microcontroller.
- The pin configuration matches the software code logic.
- Power distribution is safe and efficient.
- The system can operate as expected before building the physical circuit.

Pin Mapping Summary:

Component	Sensor/Unit Pin	ESP32 Pin	Notes
DHT22 (Temp & Humidity)	DATA	GPIO 4	Requires a 10kΩ pull-up resistor between DATA and 3.3V
	VCC	3.3V	
	GND	GND	
BH1750 (Light Sensor)	SDA	GPIO 21	Shared with I2C bus (Wire library)
	SCL	GPIO 22	
	VCC	3.3V or 5V	Supports both
	GND	GND	
CCS811 (Air Quality)	SDA	GPIO 21	Shares I2C bus with BH1750
	SCL	GPIO 22	
	VCC	3.3V	**Never connect to 5V** — it's sensitive
	GND	GND	
Soil Moisture Sensor	AOUT (Analog Output)	GPIO 34 (ADC1)	Avoid using GPIO 36 or 39 with PWM
	VCC	3.3V or 5V	Depends on sensor version
	GND	GND	
Relay Module (4-Ch)	IN1	GPIO 16	Controls water pump
	IN2	GPIO 17	Controls water valve
	VCC	5V	Make sure it's enough to activate the relay
	GND	GND	Common ground with ESP
Water Pump	Connected via Relay Output	-	Powered through relay, not directly from ESP32
Valve	Connected via Relay Output	-	Same as the pump
L298N Motor Driver	IN1	GPIO 18	Motor direction control
	IN2	GPIO 19	Reverse direction control
	ENA (PWM Motor Enable)	GPIO 5	Supports PWM
	VCC	12V External Supply	For motor power
	GND	Shared GND	Must connect to ESP32 ground as well

DC Motor	OUT1 / OUT2	From L298N	Not connected directly to ESP32
LCD 20x4 (I2C)	SDA	GPIO 21	Same I2C line as BH1750 & CCS811
	SCL	GPIO 22	
	VCC	5V	Needs 5V for full brightness
	GND	GND	

Table 6 ESP Pin Mapping Summary

3.2ESP32 Programming – Main Code:

This section contains the full source code uploaded to the ESP32 microcontroller to manage sensor readings, environmental data display, and automatic control of irrigation and ventilation systems. The code uses various libraries for sensor integration, LCD display, and motor control. It's written in Arduino C++ and tested on the ESP32 board.

Libraries Used:

- DHT.h: for temperature and humidity sensor
- BH1750.h: for light intensity sensor
- Adafruit_CCS811.h: for CO2 & TVOC sensor
- LiquidCrystal_I2C.h: to display data on LCD screen
- Wire.h: for I2C communication with BH1750 & LCD

Key Code Sections & Explanations:

- **Sensor & Component Initialization:**

```
#define DHTPIN 4
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);
```

This defines the DHT22 sensor, which reads temperature and humidity. It's connected to GPIO pin 4 on the ESP32.

```
BH1750 lightMeter;
Adafruit_CCS811 ccs;
```

Initializes the **BH1750** light sensor (for brightness level in lux) and the **CCS811** sensor (for CO_2 and TVOC air quality readings).

```
LiquidCrystal_I2C lcd(0x27, 20, 4);
```

Initializes a 20x4 I2C LCD screen at address 0x27 to display sensor data (like temp, humidity, light, etc.).

```
#define SOIL_PIN 34
```

Specifies the analog pin (GPIO 34) used to read **soil moisture** data. Low value = dry soil

---

- **Relay & Motor Configuration:**

```
#define RELAY_PUMP 16
#define RELAY_VALVE 17
```

These define GPIO pins used to control the **water pump** and **valve** via relay modules.

```
#define MOTOR_IN1 18
#define MOTOR_IN2 19
#define MOTOR_EN 5
```

Pins connected to a **DC motor driver (L298N)**. Used for ventilation fan or auto window. The MOTOR_EN pin controls speed via PWM.

- **Sensor Readings:**

```
float temperature = dht.readTemperature();
float humidity = dht.readHumidity();
```

Reads the current temperature and humidity from the DHT22 sensor.

```
uint16_t lightLevel = lightMeter.readLightLevel();
```

Reads the brightness level from the BH1750 light sensor in lux.

```
if (ccs.available()) {
  if (!ccs.readData()) {
    co2 = ccs.geteCO2();
    tvoc = ccs.getTVOC();
```

Reads **CO_2 (eCO2)** and **TVOC** (Total Volatile Organic Compounds) levels from the CCS811 sensor when data is available.

- **Displaying Data on LCD:**

```
lcd.setCursor(0, 0);
lcd.print("T:");
lcd.print(temperature);
lcd.print("C H:");
lcd.print(humidity);
```

Displays temperature and humidity on the first row of the LCD screen. Other rows show light level, CO2, TVOC, and soil moisture.

- **Automatic Irrigation Logic:**

```
if (soil < 2000) {
  digitalWrite(RELAY_PUMP, LOW);   // Turn ON
  digitalWrite(RELAY_VALVE, LOW);  // Turn ON
} else {
  digitalWrite(RELAY_PUMP, HIGH);  // Turn OFF
  digitalWrite(RELAY_VALVE, HIGH); // Turn OFF
}
```

If the soil is dry (value under 2000), the system automatically turns on the water pump and valve using relays. Once soil is moist, it turns them off. This is the core of your smart irrigation system

---

- **Motor Control (Ventilation/Fan):**

```
rotateMotorForward();
delay(500);
stopMotor();
```

The motor runs forward for 0.5 seconds, then stops. This is a simple control pattern, and it can be used for ventilation flaps or a rotating fan.

```
void rotateMotorForward() {
  digitalWrite(MOTOR_IN1, HIGH);
  digitalWrite(MOTOR_IN2, LOW);
  analogWrite(MOTOR_EN, 200);
}
```

Makes the DC motor rotates forward at medium speed (PWM 200). You can change direction or speed using the other two functions rotateMotorBackward() and stopMotor().

Chapter 4: User Interface

The User Interface (UI) of the Smart AgriStation project was designed to bridge the gap between users and the intelligent greenhouse system through a clean, responsive, and intuitive web application. At its core, the website acts as the main control hub, allowing users to monitor real-time data such as temperature, humidity, and soil moisture, while also enabling direct control of greenhouse systems like irrigation, lighting, and ventilation. These features are powered by a seamless integration with Firebase, which manages both the Realtime Database and user authentication, and communicates efficiently with the ESP32 microcontroller that collects data from sensors and executes control commands. This chapter explores the full structure of the interface from the layout and design of the pages to how Firebase connects everything in real-time, making the user experience smooth, practical, and reliable for managing a smart agricultural environment.

AgriStation is an intelligent farming ecosystem website designed to monitor and control smart greenhouse components. It integrates:

- Real-time monitoring of environmental data (temperature, humidity, light, CO_2, soil moisture, nutrient levels, pH).
- Control features for irrigation, ventilation, lighting, and nutrient dispensing.
- Firebase Realtime Database integration with ESP (e.g., ESP32/ESP8266).
- Responsive frontend interface using HTML, CSS, and JavaScript.

4.1 Front-End Structure:

This Part outlines the complete frontend development of the AgriStation website, which is part of a smart farming ecosystem integrating renewable energy, smart irrigation, and real-time monitoring and control.

Why We Built a Website for Smart AgriStation:

In our Smart AgriStation project, we chose to build a **website** as the central interface for users to interact with the smart greenhouse system. The website plays a crucial role in delivering a seamless, user-friendly experience and connecting all parts of the system in real time.

Here's why we built the website:

1. **Real-Time Monitoring Made Simple**
   The website displays live data from sensors such as temperature, humidity, soil moisture, light, and CO_2 levels. This allows users to monitor the environmental conditions of the greenhouse instantly — from anywhere, on any device.
2. **Remote Control Panel Access**
   Through the website, users can control essential systems like irrigation, ventilation fans, grow lights, and nutrient dispensers. These are connected via Firebase to the ESP microcontrollers, enabling real-time toggling of devices with just a click.
3. **User-Friendly Interface**
   The website provides a clean and organized interface that is easy to use for both technical and non-technical users. Features like profile management, dashboards, data charts, and control buttons make it intuitive and accessible.
4. **Data Visualization with Charts**
   We used chart libraries to display weekly trends (e.g., temperature and humidity over time), helping users make informed decisions about their plants based on visual insights.
5. **Centralized Management System**
   The website acts as a central hub where all data — user profiles, sensor values, and control states — are organized and managed in one place. This increases efficiency and reduces the need for physical supervision of the greenhouse.
6. **Responsive and Cross-Platform**
   Since it's web-based, our system works on **mobile, tablet, and desktop** without needing any app installation. This makes it ideal for remote farmers, greenhouse operators, and students.
7. **Prepared for Future Scalability**
   As we add more sensors, features, or even expand to multiple greenhouses, the website can scale easily without needing hardware changes.

Key Features

1. **Login & Sign-up Pages:**
   - **Login Page:**
     1. Form fields: Email, Password

2. Button: Login
3. Link to Sign-Up

– **Sign-Up Page:**

1. Form fields: Name, Email, Password, Confirm Password
2. Button: Create Account
3. Link to Login

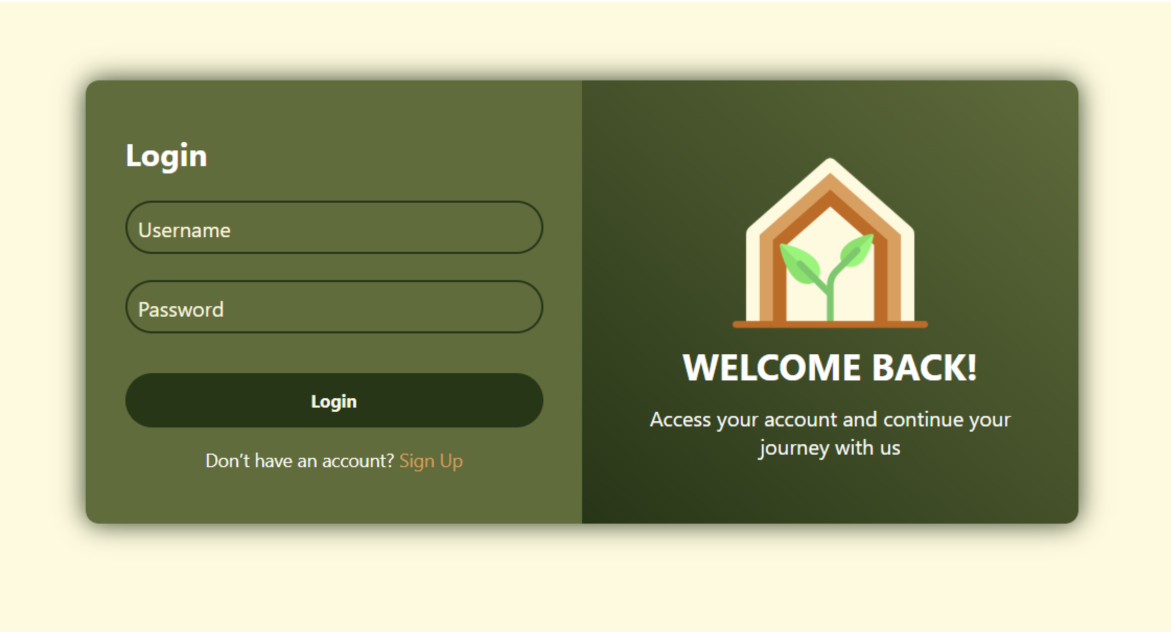

Figure 12 Login Page

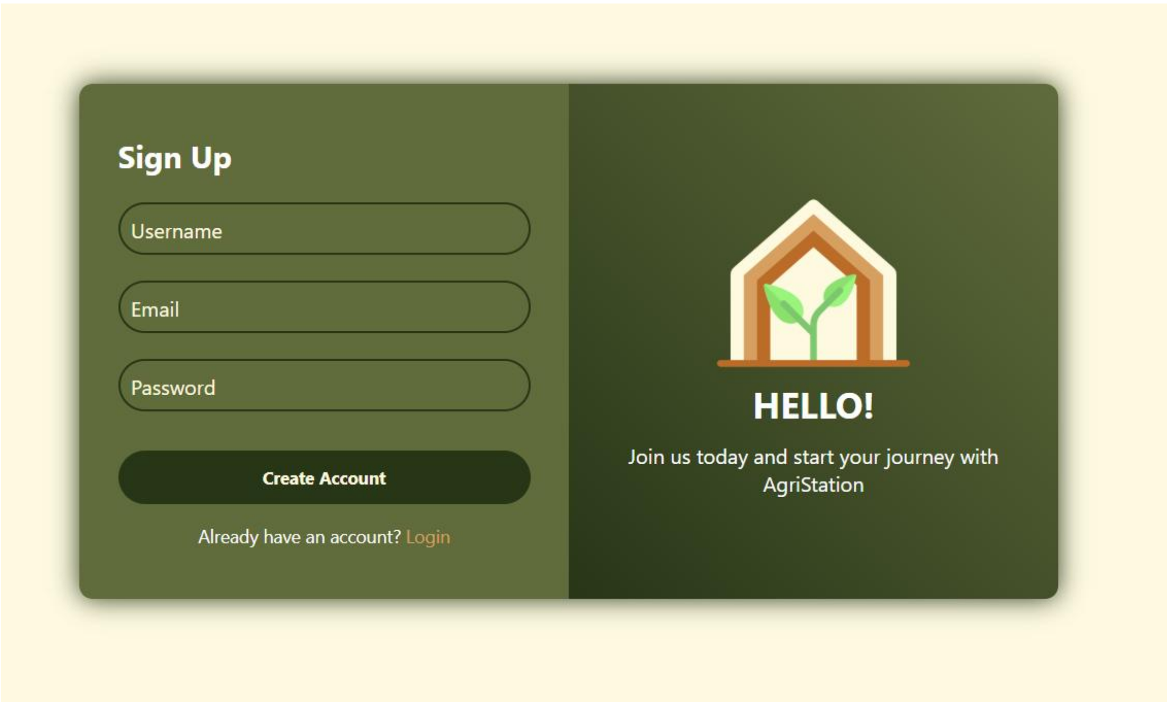

Figure 13 Sign up Page

2. **Home Page:**
   It displays the idea of the project as well as a view of the team and the background of our efforts
   - Header & Footer
   - Hero section with project intro.
   - About the project.
   - Embedded video of the Greenhouse journey
   - Team members.

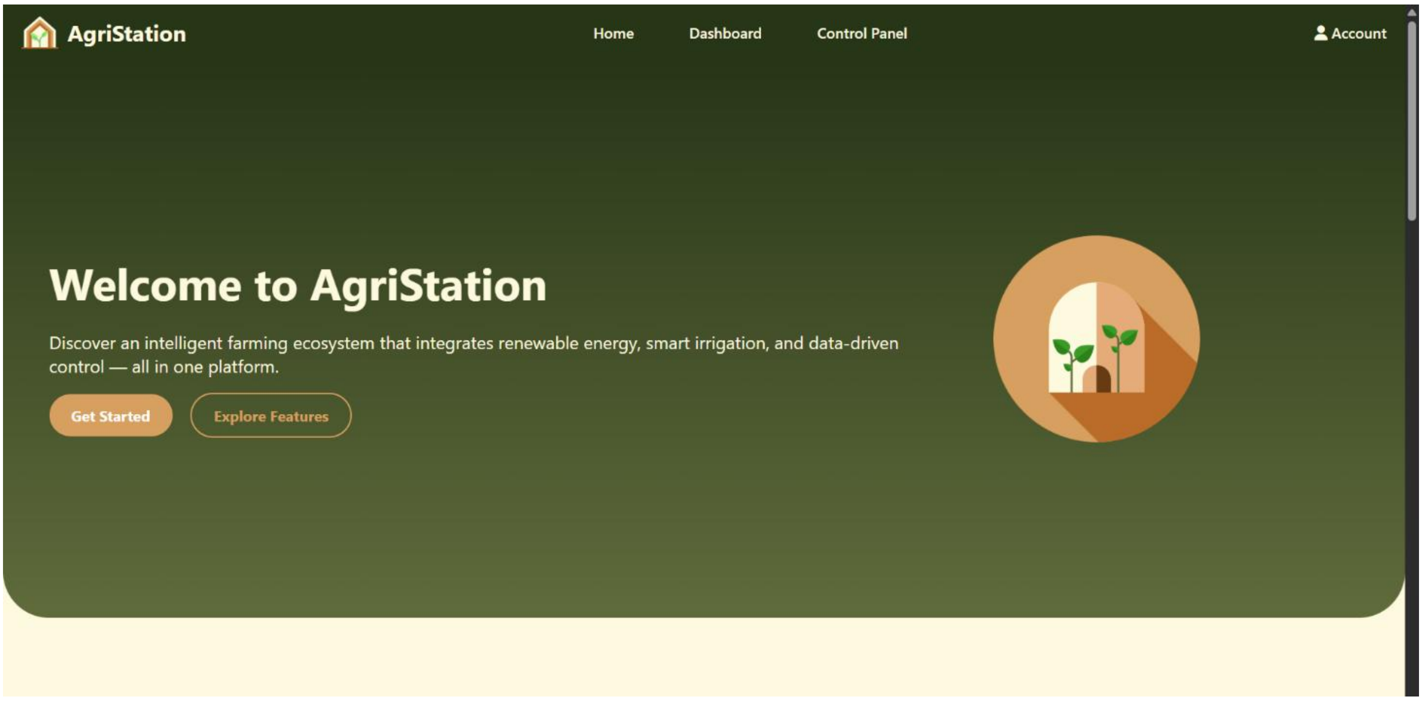

Figure 14 Home Page

3. **Live Dashboard:**
   It displays real-time readings from sensors placed in the greenhouse and field. These include:
   - **Sensor Cards Section:**
     Cards displaying real-time data:
     1. Temperature
     2. Humidity
     3. Soil Moisture
     4. Light Intensity
     5. CO2 Level
     6. Nutrient Level

- **Charts Section (Chart.js):**
  Temperature Over the Week
  Humidity Over the Week
- **Plant Status Section:**
  Crop cards (e.g., pepper, eggplant), each showing image, name, and inf

These readings are fetched from the ESP microcontroller, which sends data from physical sensors to the Firebase.

Figure 15 Dashboard Page

4. **Control Panel**
   This section allows the user to:
   - **Control Items:**
     Each has icon, name, status, and a toggle switch:
     1. Ventilation Fan
     2. Irrigation System
     3. Grow Lights
     4. Nutrient Dispenser

- **Apply Settings Button** to send data to Firebase/ESP.
- (Below this) you can add live status indicators or logs.

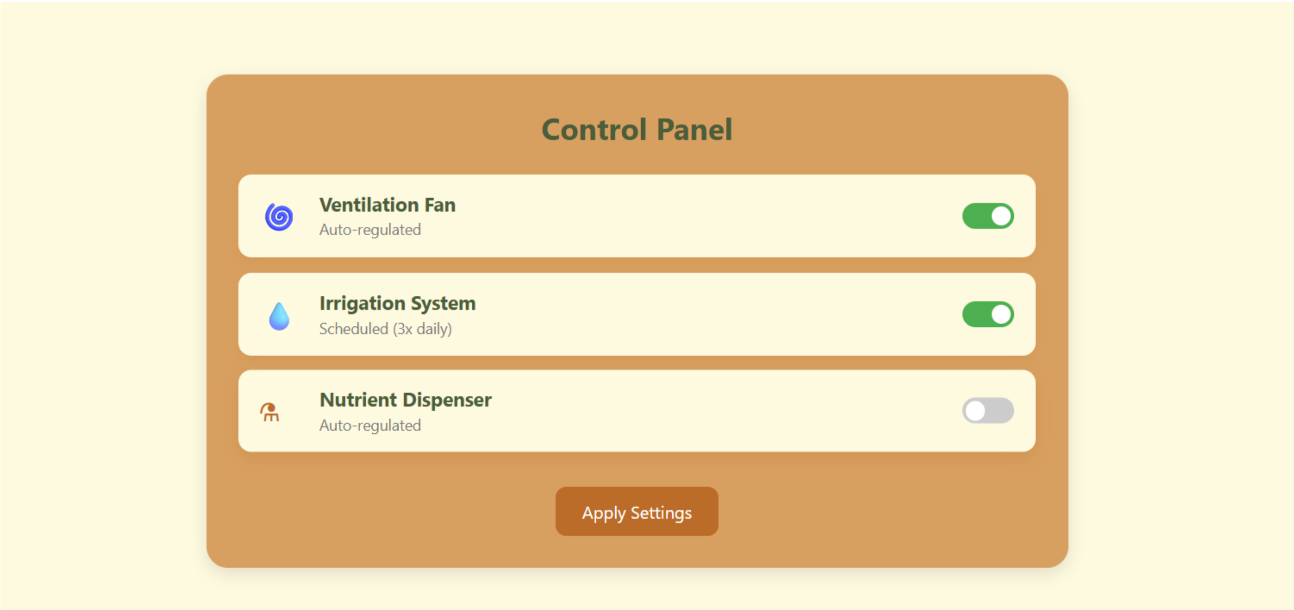

Figure 16 Control Panel

---

5. **Profile Page:**
   - Profile Card:
   - Image, name, email, role
   - Edit Profile button
   - Log out button

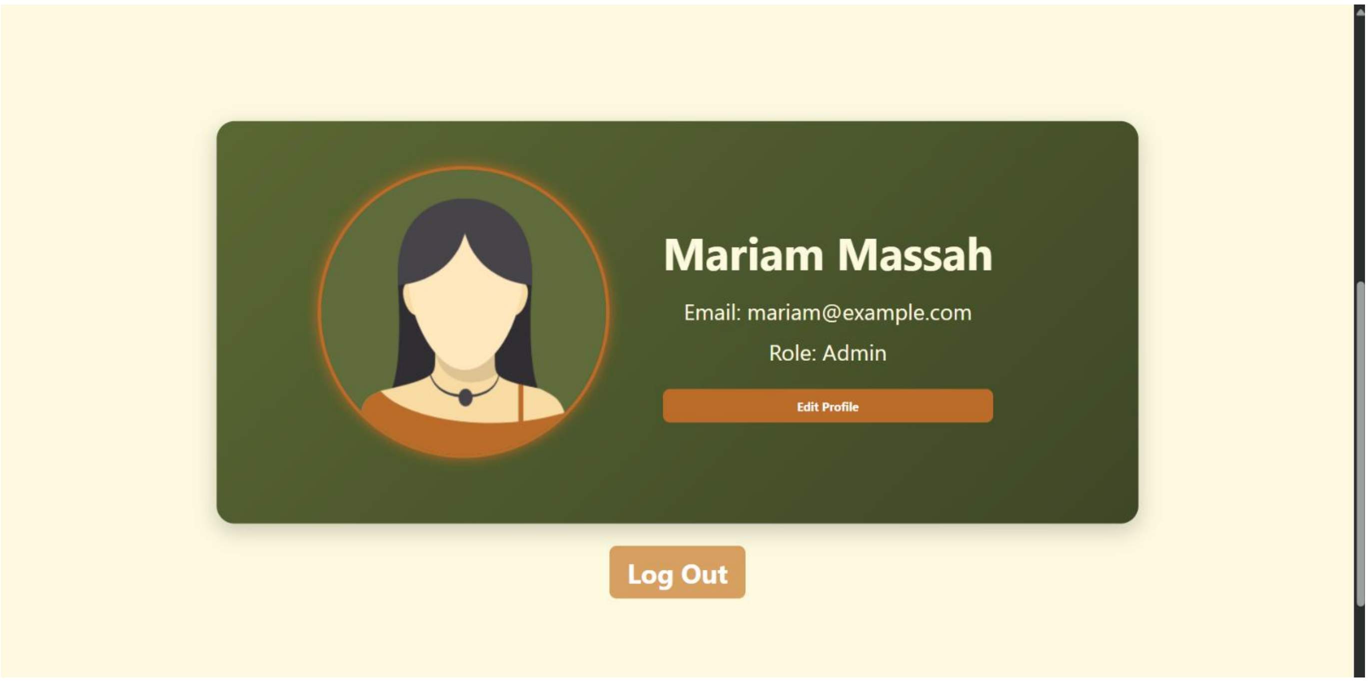

Figure 17 Profile page

---

6. **User-Friendly Design:**

   – **Color Theme:** Earthy greens and the highlight color #bb6c28
   – **Typography:** Clean, readable fonts
   – **Icons:** Font Awesome or inline emojis
   – **Responsive Design:** Use CSS Flexbox and Grid
   – **Hover Effects:** Cards and buttons with smooth transitions
   – **Reusable Components:** Nav, Footer, Cards, Charts

This setup ensures that your project is clean, scalable, and ready to be integrated with Firebase and ESP for a full smart farming experience.

4.2 Firebase Structure:

This Part explains how to connect your AgriStation website to Firebase Realtime Database and ESP8266/ESP32 to send and receive live sensor data and control signals. This enables real-time monitoring and automation through the dashboard and control panel.

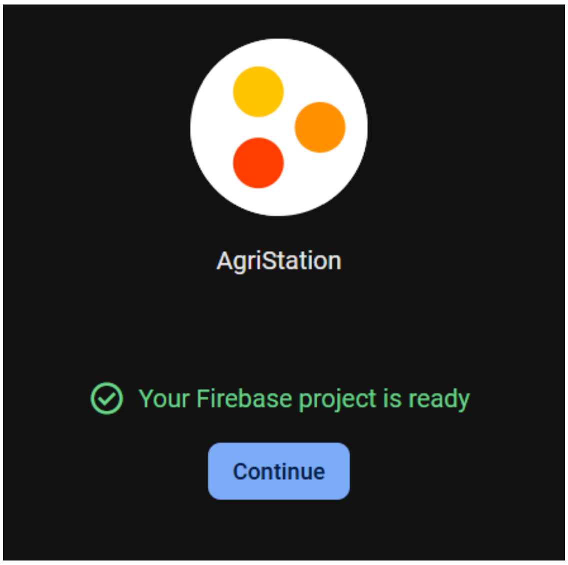

Figure 18 Firebase

Why We Chose Firebase:
Firebase, developed by Google, is a comprehensive platform that provides backend services to help developers build web and mobile applications faster and more efficiently. For our Smart AgriStation project, Firebase was the perfect fit for several reasons:

1. **Real-Time Data Synchronization**:
   Our smart greenhouse sensors (like temperature, humidity, light, etc.) send data in real time via ESP microcontrollers. Firebase Realtime Database allows us to instantly reflect those readings on our web dashboard without manual refreshments.
2. **Ease of Integration with ESP**:
   Firebase provides official Arduino and ESP libraries which made it very easy to

connect our ESP32/ESP8266 microcontrollers directly to the database and send/receive data.

3. **Hosting for the Website**:
   Firebase Hosting offers a fast, secure, and free way to deploy our frontend website. It supports HTTPS, which adds a layer of security to our application — especially when working with sensitive data or controls.
4. **Authentication for User Profiles**:
   Firebase Authentication simplifies login/signup flows. It supports email/password authentication out of the box, allowing us to manage user sessions securely (useful for different greenhouse users, admins, etc.).
5. **Scalability and Performance**:
   Firebase can easily scale to support larger databases, more users, and faster operations, making it suitable even as our project expands.

Advantages of Firebase in Our Smart Farming Context

- **Real-time monitoring** of environmental conditions.
- **Remote control** of greenhouse systems via the web interface.
- **Centralized data management** across different devices and users.
- **Fast deployment** and minimal setup time.
- **Zero backend code** required — saving development time and effort.

Firebase Services We Used:

Firebase Service	Purpose in Our Project
Realtime Database	To store and sync sensor values like temperature and humidity
Firebase Hosting	To host the frontend dashboard and profile pages
Firebase Authentication	To manage login and user profiles

Table 7 Firebase Services

How Firebase Interacts with ESP and Frontend:

- The ESP32 collects sensor data and uploads it to Firebase every few seconds.
- The frontend dashboard listens to data changes and updates the UI in real time using JavaScript.

- The control panel switches (like turning on irrigation or lights) are also synced to Firebase, and the ESP reads these values to trigger actuators.

How Firebase Works in AgriStation

1. Reading Sensor Data

The ESP reads values from sensors (e.g., DHT11 for temperature/humidity, soil moisture sensor) and sends them to the Firebase database using simple HTTP or Firebase libraries

2. Sending Control Signals

When a user clicks a control button on the web dashboard (e.g., to turn on the irrigation pump), the value is updated in Firebase.

The ESP listens to this value, and when it sees the change, it activates the corresponding relay or actuator.

3. Real-time Display on Website

The frontend continuously listens to Firebase for updates and changes the UI accordingly:

This gives the user live feedback on what's happening in the greenhouse without needing to refresh the page.

4. **Security and Rules**

Firebase allows the use of **database rules** to protect data and control read/write access.

For prototypes, open rules are acceptable, but for real deployments, you should implement authentication and more specific security rules to protect the system from unauthorized access.

Chapter 5: Power Generation Station

The **Power Station** is a key component of the Smart AgriStation project. It is responsible for supplying clean, renewable, and reliable energy to all systems within the smart farm — including the greenhouse, sensors, irrigation, and control units. By utilizing **solar energy**, and optionally **wind energy**, the power station ensures energy independence, reduced carbon footprint, and long-term sustainability.

In many farming environments, especially rural or off-grid areas, access to reliable electricity can be limited or costly. Our solution addresses this by using **renewable energy sources** to power the entire system autonomously, making smart agriculture possible without depending on the national power grid.

Power Station19 Figure

Project Concept:

The Smart Solar Energy System works by converting sunlight into electricity using photovoltaic (PV) panels. This electricity is then regulated and either stored in batteries or directly used to power connected devices. The system is managed through a microcontroller (such as Arduino or ESP32), which collects real-time data from sensors and uploads it to **Firebase Realtime Database**. A web dashboard is used to monitor and control the system remotely.

5.1. How the System Works:

1. **Power Generation:**
    - Solar panels generate DC electricity from sunlight during the day.
2. **Power Regulation:**
    - The solar charge controller ensures safe delivery of energy to the battery.

    - Excess voltage is either stored or regulated.
3. **Data Monitoring:**
    - Sensors collect environmental and electrical data.
    - The microcontroller reads and sends this data to Firebase.
4. **User Interaction:**
    - A real-time web dashboard displays live readings (voltage, current, temperature).
    - Users can monitor the system remotely and trigger manual controls (e.g., fans, tracking motors, irrigation systems).
5. **Energy Storage and Usage:**
    - Electricity is stored in batteries and can be used during the night or in low sunlight.
    - Inverters convert DC to AC if AC-powered appliances are used.

Why It's Smart:

- Uses **real-time sensor feedback** to optimize energy production.
- Integrates with cloud databases like **Firebase** for instant data visibility.
- Enables **remote control and automation** via a web interface.
- Employs **solar tracking** to follow the sun and maximize efficiency.

Benefits:

- **Eco-friendly:** No harmful emissions.
- **Cost-effective:** Reduces electricity bills over time.
- **Flexible:** Can be used in farms, homes, greenhouses, or remote areas.
- **Expandable:** More panels, sensors, and features can be added later

---

5.2 Power Station Components:

Core Solar Energy Components:

1. **Solar Panels** (Photovoltaic cells)
    - Types: Monocrystalline, Polycrystalline, Thin-film, Bifacial
    - Function: Convert sunlight into DC electricity.
2. **Solar Charge Controller**
    - Types: PWM (used in this project), MPPT

  - Regulates power from solar panels to batteries.
3. **Batteries**
   - Types: Lithium-ion, Lead-acid, NiCd, NiMH, Flow Batteries
   - Store solar energy for later use.
4. **Battery Management System (BMS 4S)**
   - Ensures safe battery operation by managing charging/discharging cycles.
5. **DC/DC Step-Down Converter (Buck Converter)**
   - Lowers voltage from solar panels to suitable levels for devices and charging.
6. **DC Motor (Direct Current Motor)**
   - Used in solar tracking or mechanical movements powered by solar energy.

Sensors & Monitoring:

7. **DHT11 Sensor**
   - Measures temperature and humidity around the panels.
8. **LDR (Light Dependent Resistor)**
   - Detects light intensity, used in solar tracking systems.

Control & Actuation:

9. **Servo Motors** (Continuous Rotation)
   - Adjust solar panel orientation for optimal sunlight tracking.
10. **Arduino Uno**
    - Microcontroller that reads sensors and controls other components.

Electronics & Supporting Hardware:

11. **LEDs** (Light Emitting Diodes)
    - Visual indicators or used in solar-powered lighting.
12. **Breadboard**
    - For prototyping and testing connections without soldering.
13. **Jumper Wires**
    - Electrical wires for interconnecting components on a breadboard or modules.
14. **Resistors**
    - To manage current flow and protect components.
15. **Battery Holder**
    - Physical holder for organizing and connecting batteries safely

Chapter 6: PCB Design and Implementation

The Printed Circuit Board (PCB) is one of the most essential components in any electronic project. It serves as a platform for organizing and connecting electronic components in a clean, efficient, and reliable manner. In this project, a custom-designed PCB was used to integrate sensors, control units, and actuators to ensure optimal performance and long-term stability.

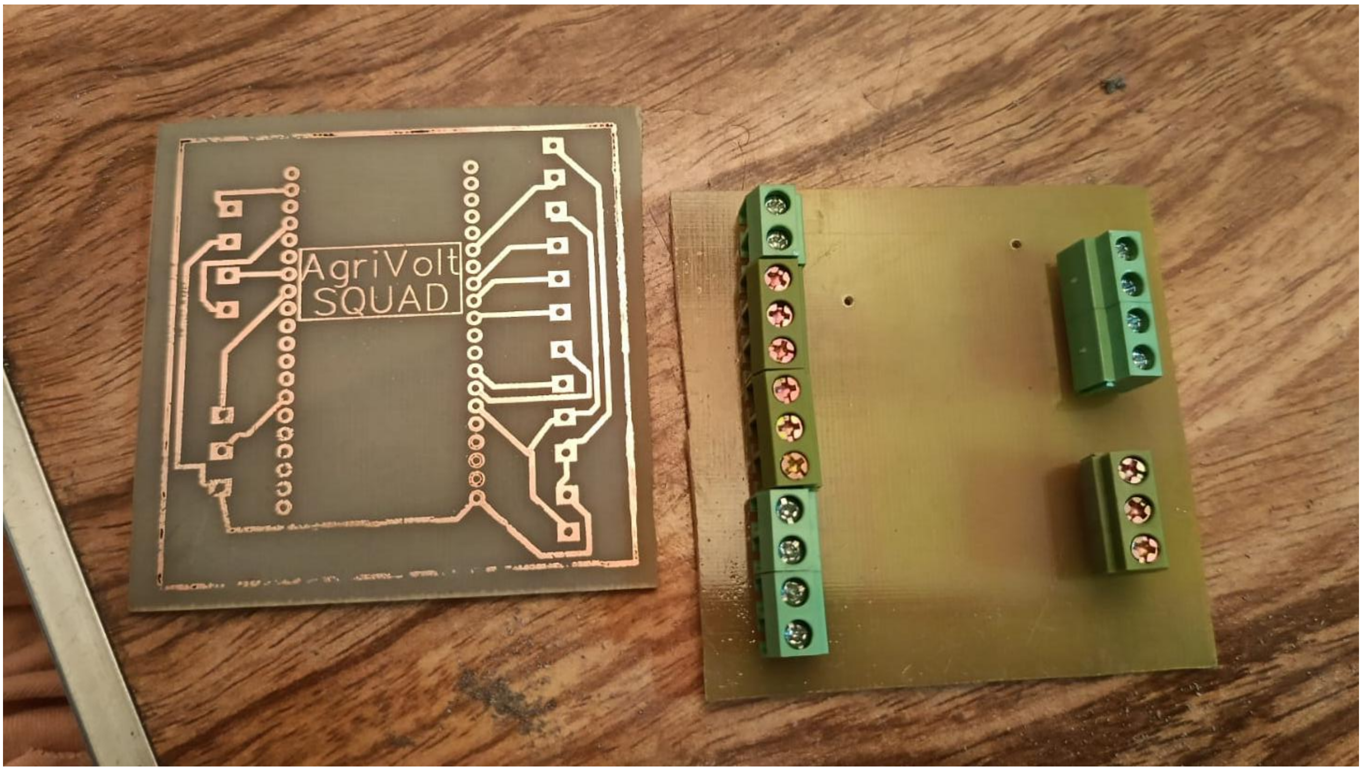

Figure 20 PCB

Importance of Using a PCB in the Project:

The decision to implement a PCB in this project was based on several critical advantages:

- Minimizing Wiring Errors: Reduces the chances of faulty or loose connections.
- Enhancing System Reliability: Especially important for continuous operation in environmental or industrial settings.
- Better Component Organization and Cooling: PCB layout allows for proper component placement and heat dissipation.
- Ease of Maintenance and Future Upgrades: A well-designed PCB makes troubleshooting and expanding the system much easier.

6.1 PCB Design Stages:

The PCB was designed according to the specific requirements of the project's components. The process involved the following steps:

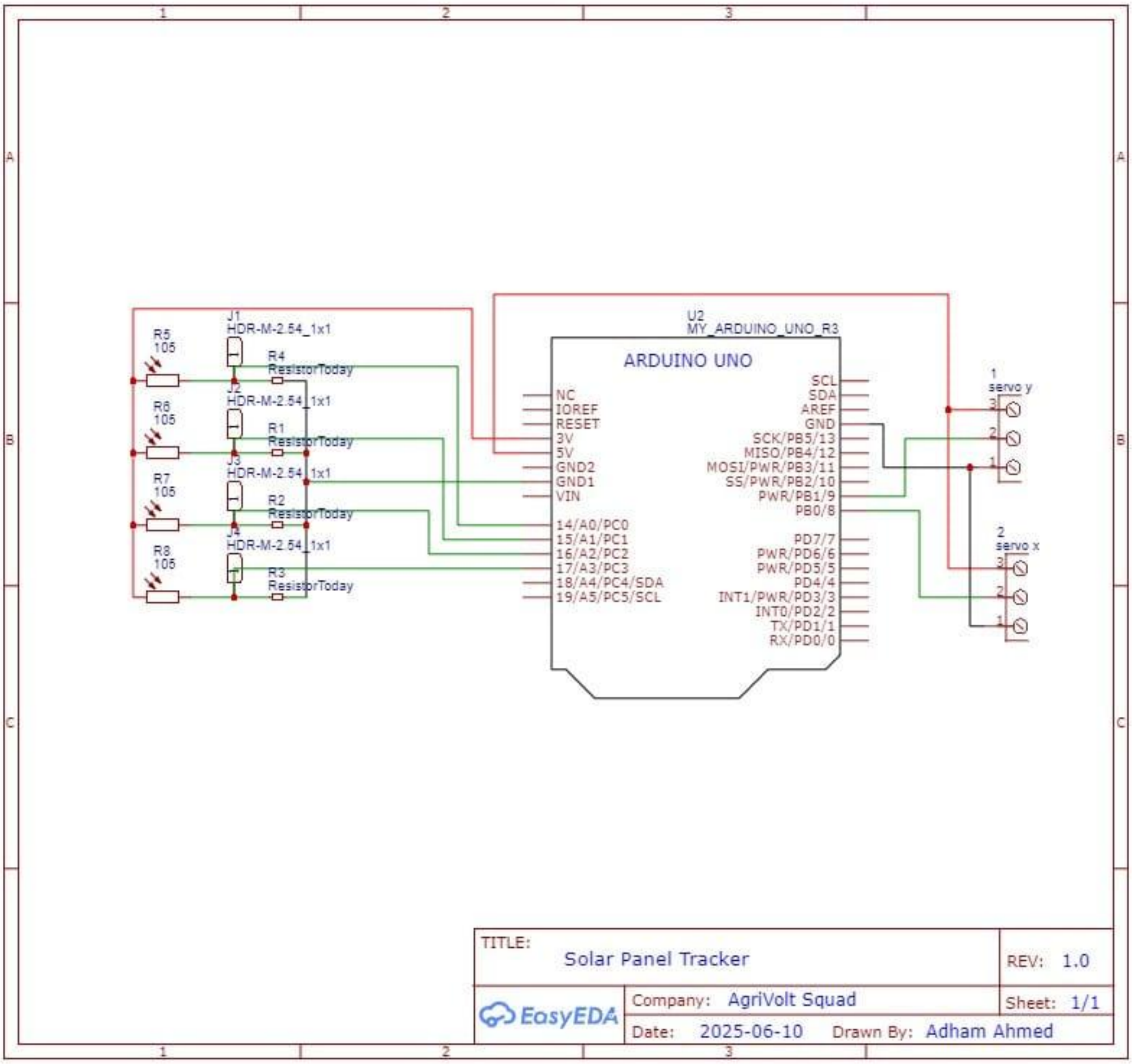

Figure 21 PCB Simulation

1. Schematic Design: Using software such as Proteus or EasyEDA, the schematic diagram was created to define all components (ESP32, DHT22, BH1750, CCS811, relay, water pump) and their connections.

2. Converting Schematic to PCB Layout: After validating the schematic, components were placed and arranged with consideration to:

3. Minimizing the length of electrical traces.

4. Isolating high-voltage paths (e.g., relays) from low-voltage sensors.

5. Creating a solid ground plane for signal stability.

6. Layer Selection: A single-layer PCB was used to reduce manufacturing costs, although a double-layer design remains an option for future enhancements.

7. Routing: Traces were routed manually or with auto-routing tools to ensure electrical efficiency and maintain safe space between paths.

8. Generating Gerber Files: Final design files (Gerber format) were exported for PCB manufacturing.

6.2 PCB Fabrication and Assembly:

1. The Gerber files were submitted to a PCB manufacturing service (such as JLCPCB or PCBWay). After fabrication, the following steps were completed:

2. Component Soldering: Components were soldered onto the PCB manually using a soldering station.

3. Connection Testing: A multimeter was used to verify continuity and check for short circuits.

4. Full System Testing: The assembled PCB was integrated into the main system and tested under actual operating conditions.

6.3 Challenges and Solutions:

During the PCB development process, several challenges were encountered:

- Limited Board Space: Solved by optimizing component placement and rerouting.

- Signal Interference Among Sensors: Addressed by adding filtering capacitors and separating ground paths.

- Manual Soldering Issues: Solved by using detachable sockets and headers for ease of maintenance.

Results and Evaluation:

- The use of a PCB in the system provided multiple benefits:
- Significantly reduced error rates and wiring issues.
- Improved system performance and operating stability.
- Simplified future modifications and component replacement.

PCB design is a fundamental aspect of any successful electronic system. In this project, the custom PCB played a vital role in achieving a clean, organized, and highly reliable setup. Its implementation significantly contributed to the professional quality and technical robustness of the final product.

Chapter 7: *Components And Cost*

	Component	Price
1	ESP32 microcontroller*2	700
2	BH1750 Light Sensors*2	360
3	DHT22 Sensors*2	420
4	CCS811 CO_2 Sensors2	800
5	Soil Moisture Sensors with Relay*2	230
6	Water Pump	185
8	DC Motor	180
9	Motor Driver Module	80
10	4-Channel Relay Module	125
11	Step-down Voltage Regulator	110
12	Solar Cell Panel	800
13	LCD Screen	280
14	PCB Board Fiberglass	95
15	Terminal Block Pins	68
16	Character LCD 20*4	255
17	Pressure Plastic Water Soleoid Valve	220
18	General Purpose Relay*2	22
19	Itoc*2	90
20	Jumper Wires *80	60
21	Metal Pipes (6 Pieces of iron pipes)	1260
22	Plastic Sheets (15 meters)	225
23	Cement& white glue	225
24	cotton rope*2	180
25	nails	95
26	Other Components	920
	Total	**7985**

Table 8 Components & Prices

Conclusion & Future Work:

The **AgriStation** project represents a promising step toward the future of smart agriculture, combining modern technology with environmental sustainability to provide practical and efficient solutions to the challenges of traditional farming. By integrating the Internet of Things (IoT), renewable energy, and remote automation, the system offers a fully intelligent and automated agricultural environment.

The smart greenhouse in AgriStation not only monitors environmental conditions but also responds to them in real time, reducing the need for manual intervention and maximizing the efficient use of resources like water and energy. Through its interactive web interface, users can remotely control systems such as irrigation, lighting, and ventilation, while also accessing live data about the environment.

This project demonstrates how technology can be used to improve the quality of farming, minimize waste, and increase productivity in a smart and sustainable way. AgriStation lays a strong foundation for developing even more advanced agricultural systems in the future — systems that can support food security and address the challenges of climate change.

Future Work:

While AgriStation in its current form presents a strong and functional prototype, there are several future directions that could greatly enhance its capabilities and real-world applicability:

- **Mobile App Development:** Creating a cross-platform mobile app (using Flutter or similar frameworks) would allow users to monitor and control their greenhouse from anywhere, making the system even more accessible and convenient.
- **Advanced Analytics Dashboard:** Adding visual data insights, trend graphs, and automated reports would help users better understand their system's performance and make informed decisions.
- **Multi-Greenhouse Management:** Scaling the platform to handle multiple greenhouses or farm sections from a single dashboard would be ideal for larger operations.
- **Offline Support with Local Storage:** To make the system more reliable in rural or low-connectivity areas, a local control system could be added that syncs with Firebase only when internet is available.

Each of these additions would not only strengthen AgriStation’s core but also push it closer to being a fully-deployable smart farming solution — one that could make a real difference in addressing food security, climate challenges, and resource scarcity.

Reference:

1. **ESP32 Technical Reference Manual**
   Espressif Systems. (2023). *ESP32-WROOM-32D Datasheet*.
   mailto:https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf
2. **Arduino Documentation**
   Arduino. (2024). *Official Arduino Language Reference*.
   mailto:https://www.arduino.cc/reference/en/
3. **DHT22 Sensor Guide**
   Adafruit. (2023). *DHT22 Temperature and Humidity Sensor Guide*.
   mailto:https://learn.adafruit.com/dht
4. **BH1750 Light Sensor**
   Random Nerd Tutorials. (2023). *Guide for BH1750 Light Sensor with Arduino and ESP32*.
   mailto:https://randomnerdtutorials.com/esp32-bh1750-ambient-light-arduino/
5. **CCS811 Air Quality Sensor**
   Adafruit. (2023). *Using the CCS811 Gas Sensor with Arduino or ESP32*.
   mailto:https://learn.adafruit.com/adafruit-ccs811-air-quality-sensor
6. **LiquidCrystal_I2C Library**
   Arduino Libraries. (2023). *Liquid Crystal I2C Library Documentation*.
   https://github.com/johnrickman/LiquidCrystal_I2C
7. **Firebase Realtime Database**
   Google Firebase. (2024). *Firebase Realtime Database Documentation*.
   mailto:https://firebase.google.com/docs/database
8. **IoT Smart Greenhouse Projects**
   Circuit Digest. (2023). *IoT Based Smart Greenhouse Monitoring System Using ESP32 and Firebase*.
   mailto:https://circuitdigest.com/microcontroller-projects/iot-based-smart-greenhouse-monitoring-system-using-esp32
9. **Drip Irrigation in Smart Agriculture**
   ResearchGate. (2021). *Automation in Drip Irrigation System Using IoT*.
   https://www.researchgate.net/publication/351048070_Automation_in_Drip_Irrigation_System_using_IoT

الملخص العربي:

يهدف مشروعنا إلى تصميم وتنفيذ نظام ذكي يعتمد على **الطاقة الشمسية** لتوفير الكهرباء وإدارة الطاقة داخل البيوت الزراعية الذكية. يعتمد النظام على **ألواح شمسية** تقوم بتحويل ضوء الشمس إلى طاقة كهربائية تُخزّن في بطاريات مناسبة لاستخدامها عند الحاجة، مما يضمن استمرارية التشغيل دون الاعتماد على مصادر الكهرباء التقليدية.

يعتمد المشروع بشكل أساسي على متحكم **ESP32** الذي يتلقى بيانات دقيقة من الحساسات البيئية مثل **DHT11** لقياس درجة الحرارة والرطوبة، و **LDR** لقياس شدة الإضاءة. يتم إرسال هذه البيانات إلى **قاعدة بيانات Firebase** ، لعرضها بشكل لحظي على لوحة تحكم تفاعلية يمكن من خلالها المراقبة والتحكم.

يشمل النظام أيضًا استخدام **محركات Servo** لتحريك الألواح الشمسية تلقائيًا نحو اتجاه الشمس لزيادة الكفاءة في إنتاج الطاقة. بالإضافة إلى ذلك، يمكن تشغيل أو إيقاف الأجهزة المختلفة مثل التهوية والري من خلال لوحة التحكم عن بُعد.

يساهم هذا المشروع في **تعزيز مفهوم الاستدامة البيئية** وتقليل الاعتماد على مصادر الطاقة التقليدية، كما يعمل على **تحسين كفاءة استخدام الطاقة في البيئات الزراعية**، مما يجعله حلًا عمليًا ومتكاملًا للمزارعين والمناطق الريفية.

مفهوم المشروع:

يقوم المشروع على بناء محطة زراعية ذكية تتكون من صوبة زراعية متطورة يتم التحكم بها تلقائيًا بواسطة متحكم ESP32، مدعومة بطاقة شمسية، وتحتوي على نظام مراقبة متكامل لظروف البيئة الداخلية. يتم جمع البيانات من خلال مجموعة من الحساسات المتصلة بوحدة التحكم، والتي تقيس العناصر الحيوية لنمو النبات مثل:

- درجة الحرارة
- الرطوبة النسبية
- شدة الإضاءة
- جودة الهواء (CO_2)
- رطوبة التربة

بناءً على هذه البيانات، يتخذ النظام قرارات لحظية مثل تشغيل التهوية، تفعيل نظام الري، أو إرسال إشعارات إلى المستخدم، ما يجعل البيئة داخل الصوبة مستقرة ومثالية لنمو المحاصيل مثل الباذنجان والفلفل.

أهداف المشروع:

- بناء بيئة زراعية ذكية تستجيب تلقائيًا للتغيرات البيئية.
- تحسين إنتاجية المحاصيل من خلال توفير أفضل الظروف الممكنة للنمو.
- تقليل الفاقد في المياه والطاقة باستخدام الري بالتنقيط والطاقة الشمسية.
- تمكين التحكم والمراقبة عن بُعد من خلال واجهة ويب متصلة بقاعدة بيانات.
- تقديم نموذج عملي يمكن تطبيقه وتطويره في المناطق الزراعية الفقيرة أو الصحراوية.

التقنيات والأدوات المستخدمة:

- ESP32 Microcontroller: للتحكم في النظام وتجميع بيانات الحساسات.
- Firebase Realtime Database: لتخزين البيانات وعرضها عبر الإنترنت.
- واجهة ويب تفاعلية :تتيح للمستخدم رؤية البيانات والتحكم في الري والتهوية من أي مكان.

- حساسات متعددة:
  - DHT22لقياس الحرارة والرطوبة
  - BH1750لقياس شدة الإضاءة
  - CCS811لقياس جودة الهواء
  - Soil Moisture Sensorلقياس رطوبة التربة
- نظام ري ذكي بالتنقيط، مرتبط بحساس رطوبة التربة ومضخة ماء.
- مروحة تهوية وصمام تحكم للتحكم في حرارة ورطوبة الجو.
- وحدة طاقة شمسية تتضمن ألواح وبطارية لتشغيل النظام بالكامل.
- شاشة LCD 20x4 لعرض البيانات بشكل لحظي.
- محاكاة برمجية باستخدام Proteus لتجربة النظام افتراضيًا قبل تنفيذه فعليًا.

أهمية المشروع:

تكمن أهمية Smart AgriStation في تقديم حل ذكي ومستدام للتحديات الزراعية الحالية، خاصة في الدول التي تعاني من مناخ قاسي أو نقص في الموارد. فهو يقلل الاعتماد على العامل البشري، ويضمن زراعة ناجحة باستخدام أقل كمية ممكنة من المياه والطاقة. كما أنه يمثل خطوة فعلية نحو التحول الرقمي في الزراعة، ويمكن تطويره مستقبلًا ليخدم مزارع أكبر، أو لدمجه مع أنظمة ذكاء اصطناعي لتحسين اتخاذ القرار الزراعي.

المشروع لا يمثل فقط نموذجًا هندسيًا ناجحًا، بل أيضًا رؤية مستقبلية لزراعة أكثر كفاءة، واعتمادًا على البيانات، وأقل تأثيرًا على البيئة.