**HYDROPONICS SYSTEM**

**A Major Project-Report**

**Submitted in Partial fulfillment for the award of**

**Bachelor of Technology in CSE-IoT**

Submitted to

**RAJIV GANDHI PROUDYOGIKI VISHWAVIDYALAYA**

**BHOPAL (M.P)**



**MAJOR PROJECT-I REPORT**

Submitted by

**Sneha[0103IS221200]**

**Varun Singh[0103IS221220]**



**Department of CSE-IOT**

**Lakshmi Narain College of Technology, Bhopal (M.P.)**

**Session 2025-26**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **TOPICS** | **PAGES** |
| **1.** | **Problem Domain** | 5-7 |
| **2.** | **Literature Survey** | 8-10 |
| **3.** | **Major objective & scope of project** | 11-12 |
| **4.** | **Problem Analysis and requirement**  **specification** | 13-14 |
| **5.** | **Case Study** | 15 |
| **6.** | **Detailed Design(Modeling and ERD/DFD)** | 16-19 |
| **7.** | **Hardware/Software platform environment** | 20-22 |
| **8.** | **Snapshots of Input & Output** | 23-26 |
| **9.** | **Coding** | 27-37 |
| **10.** | **Project limitation and Future scope** | 38-39 |
| **11.** | **References** | 40 |
| **Total Pages : 40 Pages** | | |

## CHAPTER 1

**PROBLEM DOMAIN**

Agriculture is a critical sector of the global economy, and irrigation plays a key role in ensuring sustainable crop production. Traditional irrigation practices often lead to water wastage and fail to consider real-time environmental and soil conditions.

The Smart Irrigation System aims to address these inefficiencies by integrating Internet of Things (IoT) technologies.

The system uses components like the ESP8266 microcontroller, soil moisture sensor, DHT11 temperature and humidity sensor, and the Blynk app to monitor and control irrigation remotely. Agriculture is one of the oldest industries and remains a critical component of global food security. However, with population growth and environmental challenges, sustainable farming practices have become a necessity.

Inefficient water usage in agriculture leads to over-irrigation, waterlogging, or drought conditions, which reduce productivity and harm the environment.

### 1.1 Environmental Monitoring & Control

An automated hydroponics system relies on precise environmental control to ensure optimal plant growth. Advanced climate regulation systems, including temperature and humidity sensors, constantly monitor and adjust conditions to maintain a stable environment. Some setups also integrate CO₂ enrichment, which enhances the photosynthesis process, leading to faster and healthier plant growth. To prevent overheating and improve airflow, automated ventilation systems play a crucial role in ensuring that air circulates properly within the growing space.

Water and nutrient management are equally vital in hydroponics. Automated systems monitor pH levels and electrical conductivity (EC) to maintain ideal nutrient concentrations. By utilizing AI-driven nutrient dosing mechanisms, plants receive the exact amount of essential minerals they need. Water recycling systems further enhance efficiency by filtering and reusing water, minimizing waste and reducing overall consumption. These features make hydroponic farming not only sustainable but also highly productive.

Pest and disease detection is another major aspect of hydroponic automation. AI-powered cameras and sensors can identify early signs of infestations, helping farmers take preventive actions before damage occurs. Automated spraying systems apply organic pesticides only when necessary, reducing excessive use of chemicals. Additionally, UV sterilization techniques are employed to eliminate harmful bacteria and fungi, ensuring that plants remain healthy and disease-free.

#### 1.2 Automation & Robotics

The foundation of an automated hydroponics system lies in the integration of smart technology. IoT-based sensors constantly collect real-time data, allowing for informed decision-making. Wireless connectivity enables remote access, meaning farmers can monitor and adjust settings through mobile applications. Cloud-based data storage preserves historical plant health records, which can be analyzed to improve future yields and optimize growing conditions.

Automation extends to irrigation and nutrient delivery. Drip irrigation systems precisely control the amount of water and nutrients each plant receives, minimizing waste. AI-driven algorithms dynamically adjust irrigation schedules based on factors such as weather conditions and plant growth stages. Furthermore, self-cleaning mechanisms prevent clogs and algae formation in water reservoirs, ensuring uninterrupted system operation.

Robotic assistance is becoming increasingly common in hydroponics. AI-powered robotic harvesters identify ripe produce and pick it efficiently, reducing labor costs and improving consistency. Automated seed planting ensures uniform distribution, enhancing the efficiency of crop growth. AI-driven predictive models analyze plant development patterns to estimate growth rates and determine the best harvesting times, thereby optimizing yield potential.

##### 1.3 Resource Optimization

Resource conservation is one of the primary advantages of automated hydroponic systems. Closed-loop hydroponic setups recycle water efficiently, reducing waste and conserving valuable resources. Smart filtration systems remove contaminants from water, ensuring a clean and healthy environment for plants. Some farms incorporate rainwater harvesting, supplementing their water supply and further enhancing sustainability.

Energy efficiency is another critical aspect. Solar-powered hydroponic systems reduce dependence on traditional electricity sources, making the operation more environmentally friendly. LED grow lights are used instead of conventional lighting, as they consume significantly less power while providing optimal light wavelengths for plant growth. AI-based power management systems continuously analyze energy consumption and adjust power usage to minimize costs and environmental impact.

Space utilization is another area where hydroponics excels. Vertical farming techniques allow growers to maximize yield in limited areas by stacking plants in multiple layers. Automated plant spacing systems ensure that each plant receives sufficient light and air circulation, leading to improved productivity. Multi-crop adaptability enables hydroponic systems to accommodate a variety of plant species, providing flexibility and supporting diverse agricultural needs.

##### 1.4 System Scalability & Maintenance

Scalability and maintenance are essential for ensuring that automated hydroponic systems remain efficient and reliable. AI-based diagnostics monitor pumps, sensors, and nutrient delivery systems, identifying potential failures before they become critical issues. Automated alerts notify farmers of system malfunctions, allowing for quick intervention and troubleshooting. Some advanced setups even feature self-healing mechanisms that adjust parameters to prevent breakdowns, significantly reducing downtime.

Remote monitoring and control features allow farmers to oversee their hydroponic system from anywhere in the world. Cloud-based dashboards display real-time insights, giving users a detailed view of environmental conditions, plant health, and resource consumption. Mobile app integration makes it easy for growers to adjust system settings, analyze historical data, and receive automated reports, enabling data-driven decision-making and improving overall productivity.

Scalability is a crucial consideration for expanding hydroponic operations. Modular system designs make it easy to expand and customize setups according to specific farming needs. Adjustable growth parameters allow growers to tailor conditions for different crops, ensuring maximum yield. AI-based yield optimization enhances productivity by continuously analyzing plant health and adapting farming strategies to improve output.

Automated hydroponic systems represent the future of agriculture, offering **precision, efficiency, and sustainability**.

## CHAPTER 2

**LITERATURE SURVEY**

**Literature Survey on Automated Hydroponics Systems**

### 2.1 Introduction

Hydroponics is a **soilless farming technique** that relies on nutrient-rich water solutions to grow plants efficiently. With the rise of **automation and smart technologies**, hydroponics has evolved into a highly optimized system that integrates **IoT, AI, and robotics** to enhance productivity, reduce labor, and improve sustainability. Several studies have explored the effectiveness of automated hydroponics systems in **resource management, environmental control, and scalability**.

The application of technology in agriculture is not a new concept, but the rapid advancement of IoT has brought significant innovations. Traditional irrigation systems, such as flood or sprinkler irrigation, often result in overwatering or underwatering, leading to resource wastage and reduced crop yield.

### 2.2 Smart Hydroponics Farming & IoT Integration

A study titled "A Survey of Smart Hydroponic Systems" discusses the integration of **IoT and AI-based technologies** in hydroponic farming. The research highlights how **temperature, humidity, and water quality** are crucial factors influencing plant growth. AI-driven automation ensures that plants receive the necessary nutrients while optimizing environmental conditions. The study also explores **image processing techniques** for plant disease detection and yield prediction.

Another research paper, "Automated Hydroponics System", focuses on **deep water culture and nutrient film techniques** to automate plant growth. The study employs **linear regression models** to control **electrical conductivity (EC) and pH levels**, ensuring optimal nutrient absorption. The system uses **microcontrollers** to adjust nutrient concentrations dynamically, reducing human intervention.

* Studies demonstrate that IoT-based systems can reduce water usage by 30-50%.
* Integration with weather forecasts and AI models improves decision-making.

### 2.3 AI-Based Optimization & Resource Management

The paper "A Survey on Hydroponic Methods of Smart Farming" examines the role of **AI and embedded controllers** in hydroponic automation. The study emphasizes **pesticide reduction**, stating that hydroponics eliminates nearly **80% of pest attacks** compared to traditional farming. The research also discusses **cloud-based data analysis**, which allows farmers to access global advancements and optimize their farming techniques.

Additionally, AI-driven **nutrient management** plays a crucial role in hydroponics. The study "Automated Hydroponics System" highlights how **electrical conductivity (EC) monitoring** ensures plants receive balanced nutrients. AI algorithms analyze **total dissolved salts, pH levels, and nutrient concentration ratios**, adjusting them dynamically to improve plant health.

### 2.4 Scalability & Future Prospects

Scalability is a key challenge in hydroponics. The study "A Survey of Smart Hydroponic Systems" explores **modular system designs** that allow farmers to expand their setups efficiently. AI-based **yield prediction models** help optimize crop cycles, ensuring maximum productivity. The research also discusses **urban hydroponics**, where small-scale automated systems enable farming in limited spaces.

Future advancements in **hydroponic automation** include **robotic harvesting, AI-driven climate control, and blockchain-based supply chain management**. Researchers are exploring **machine learning algorithms** to predict plant growth patterns and optimize nutrient delivery.

### 2.5 Advanced Hydroponic Techniques

Recent studies have explored **advanced hydroponic techniques** that integrate **machine learning, deep learning, and cloud computing**. AI-powered **image recognition systems** can detect plant diseases early, reducing losses and improving yield. Some research focuses on **automated irrigation systems** that adjust water flow based on real-time environmental conditions.

**Existing Systems:**

* Manual Irrigation Systems: Labor-intensive, inconsistent, and time-consuming. TimerBased Systems: Limited by fixed schedules, which may not align with soil needs.
* Sensor-Based Smart Systems: Use soil moisture sensors and weather data but often lack remote access and real-time data monitoring.

A study on **vertical hydroponics** highlights how **multi-layered farming** can maximize space utilization. AI-driven **light optimization** ensures that plants receive adequate illumination, improving photosynthesis efficiency. Additionally, **automated climate control** systems regulate temperature and humidity dynamically, creating an ideal growing environment.

**6. Conclusion**

Automated hydroponics systems represent a **sustainable, efficient, and scalable** approach to modern agriculture. Research studies emphasize the importance of **IoT, AI, and robotics** in optimizing plant growth, reducing resource consumption, and improving scalability. As technology advances, hydroponics is expected to play a crucial role in **urban farming and global food security**.

## CHAPTER 3

**MAJOR OBJECTIVE & SCOPE OF PROJECT**

### 3.1 Major Objectives

Automated hydroponics systems aim to revolutionize agriculture by integrating **technologydriven solutions** for efficient, sustainable, and high-yield farming. The key objectives include:

1. **Optimized Resource Utilization:**

Hydroponics significantly reduces **water consumption** compared to traditional farming. Automated systems ensure **precise nutrient delivery**, minimizing waste and maximizing efficiency.

1. **Enhanced Crop Yield and Quality:**

AI-driven monitoring and automation help maintain **ideal growing conditions**, leading to **higher yields** and **better-quality produce**. Controlled environments eliminate external threats like pests and unpredictable weather.

1. **Reduction in Labor and Operational Costs:**

Automation reduces the need for manual labor by handling **irrigation, nutrient management, and climate control**. This lowers operational costs and makes farming more accessible.

1. **Sustainability and Environmental Benefits:**

Hydroponics eliminates soil degradation and reduces **chemical runoff** into the environment. Automated systems optimize **energy consumption**, making farming more eco-friendly.

1. **Scalability and Adaptability:**

Modular hydroponic systems allow **easy expansion** for commercial farming. AI-based adaptability ensures **multi-crop compatibility**, making hydroponics viable for various plant species.

**3.2 Goals :**

* Goal 1: To automate irrigation based on real-time soil moisture and environmental data.
* Goal 2: To provide a reliable communication system for remote monitoring.
* Goal 3: To enhance water usage efficiency, conserving resources and increasing crop yield.

### 3.3 Future Scope

The future of automated hydroponics is promising, with advancements in **AI, IoT, and robotics** shaping the next generation of farming. Key areas of development include:

1. **AI-Driven Predictive Farming**

Machine learning models will analyze **historical data** to predict **optimal growth conditions**, improving yield forecasting and reducing resource wastage.

1. **Integration with Smart Cities and Urban Farming**

Hydroponics will play a crucial role in **urban agriculture**, enabling **vertical farming** and **rooftop gardens** to maximize space utilization in cities.

1. **Blockchain-Based Supply Chain Management**

Blockchain technology will enhance **traceability and transparency** in food production, ensuring **quality control** and reducing fraud in agricultural supply chains

1. **Robotic Harvesting and Automation**

AI-powered **robotic arms** will automate harvesting, reducing labor dependency and improving efficiency in large-scale hydroponic farms.

1. **IoT-Based Remote Monitoring and Control**

Farmers will be able to **monitor and control** hydroponic systems remotely via **cloudbased dashboards**, improving accessibility and decision-making.

1. **Integration with Renewable Energy Sources**

Future hydroponic systems will incorporate **solar and wind energy**, making farming **self-sufficient** and reducing reliance on traditional power sources.

1. **Space Agriculture and Hydroponics in Extreme Environments**

Hydroponics is being explored for **space missions** and **desert farming**, proving its potential for growing food in **challenging environments**.

Automated hydroponics is set to **transform agriculture**, making it **more efficient, sustainable, and accessible**.

## CHAPTER 4

**PROBLEM ANALYSIS AND REQUIREMENT**

**SPECIFICATION**

### 4.1Problem Analysis

Automated hydroponics systems address several challenges in modern agriculture, particularly in **resource efficiency, environmental control, and scalability**. The key problems include:

1. **Water and Nutrient Management**

Traditional farming methods often lead to **water wastage and inefficient nutrient absorption**. Hydroponics requires precise control over **pH levels, electrical conductivity (EC), and nutrient concentration** to ensure optimal plant growth.

1. **Environmental Control**

Maintaining **temperature, humidity, and CO₂ levels** is crucial for plant health. Automated hydroponics systems must integrate **climate control mechanisms** to regulate these factors dynamically.

1. **Pest and Disease Prevention**

Soil-based farming is prone to **pest infestations and diseases**. Hydroponics eliminates soil-related issues but requires **automated monitoring systems** to detect and prevent plant diseases.

1. **Energy Consumption**

Hydroponic farms rely on **artificial lighting, pumps, and climate control systems**, leading to high energy consumption. Efficient **power management and renewable energy integration** are necessary for sustainability.

1. **Scalability and Cost Efficiency**

Expanding hydroponic farms requires **modular system designs** and **cost-effective automation solutions**. AI-driven **yield prediction models** help optimize crop cycles and improve productivity.

#### 4.2 Requirements Specification

To address these challenges, an automated hydroponics system must meet the following requirements:

1. **Hardware Requirements** o **Sensors**: pH, EC, temperature, humidity, CO₂, and light sensors.
   * **Actuators**: Water pumps, nutrient dosing systems, ventilation fans, and LED grow lights.
   * **Microcontrollers**: Arduino, Raspberry Pi, or other embedded systems for automation.
   * **Power Supply**: Solar panels or energy-efficient power management systems.

1. **Software Requirements** o **AI-Based Monitoring**: Machine learning algorithms for **predictive analytics** and **automated adjustments**.
   * **IoT Integration**: Cloud-based dashboards for **remote monitoring and control**. o **Automated Alerts**: Notifications for **system failures, nutrient imbalances, and environmental fluctuations**.
   * **Data Logging**: Historical tracking of **plant growth, resource consumption, and environmental conditions**.

1. **Functional Requirements** o **Automated Irrigation**: Dynamic water flow control based on **real-time sensor data**.
   * **Nutrient Delivery System**: AI-driven **nutrient mixing and dosing**. o **Climate Control**: Automated **temperature and humidity regulation**.
   * **Pest Detection**: AI-powered **image recognition for disease prevention**.

1. **Non-Functional Requirements** o **Scalability**: Modular design for **easy expansion**.
   * **Energy Efficiency**: Optimized **power consumption** for sustainability.
   * **User-Friendly Interface**: Mobile app and web-based dashboard for **easy system management**.
   * **Security**: Data encryption and **secure access controls** for remote monitoring.

Automated hydroponics systems are **transforming agriculture** by integrating **AI, IoT, and robotics** for **efficient, sustainable, and high-yield farming**.

## CHAPTER 5

**CASE STUDY**

Hydroponic farming is one of the most promising innovations in agriculture, especially where traditional soil-based farming faces challenges like water shortage, unpredictable weather, and poor soil quality. To make hydroponic systems more efficient and less dependent on manual efforts, an IoT-based smart system was developed. This system uses an ESP8266 microcontroller to monitor real-time conditions such as temperature, humidity, soil moisture, and human movement through various sensors. The collected data is instantly updated to the Firebase Realtime Database, allowing users to view live environmental information from any location.

A web application developed using the MERN stack (MongoDB, Express, React, Node.js) enables farmers to access and control their hydroponic systems remotely. The user-friendly dashboard shows live sensor readings and provides a control button for the irrigation motor, allowing automated watering when the soil moisture drops below a safe level. Additionally, the system collects user feedback and queries through a contact form, storing them securely in the MongoDB database.

By combining IoT hardware with full-stack web technologies, this hydroponic smart system achieves greater water efficiency, saves manual labor, and ensures healthy plant growth even when farmers are not physically present. It demonstrates how smart farming solutions can bridge the gap between technology and agriculture, helping to create sustainable farming practices for the future. This project stands as a model for the integration of automation, IoT, and agriculture, showcasing a scalable solution that can be expanded for commercial farming use with further advancements.

**Key Highlights:**

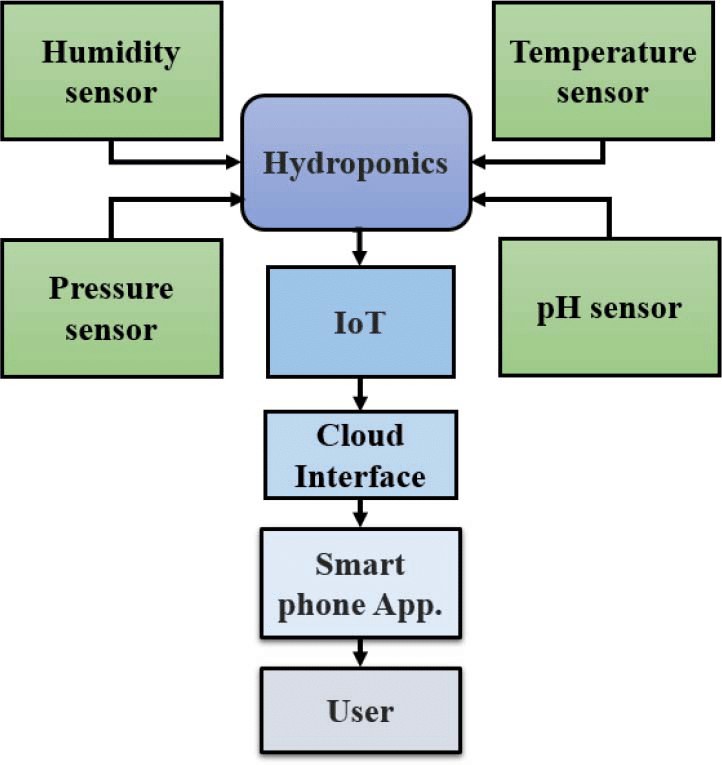
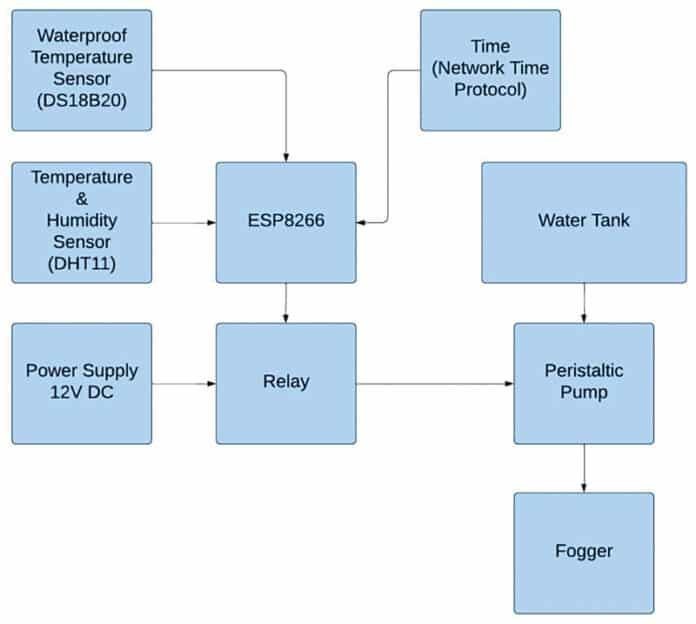
* Real-time sensor monitoring through DHT11, Soil Moisture Sensor, and PIR sensor.
* ESP8266 microcontroller for Wi-Fi enabled data transmission.
* Firebase Realtime Database for storing and updating sensor readings.
* React.js web dashboard for live data visualization and control.
* MongoDB Atlas database for securely handling user queries.
* Automated irrigation based on soil moisture detection.

## CHAPTER 6

**Detailed Design(Modeling and ERD/DFD)**

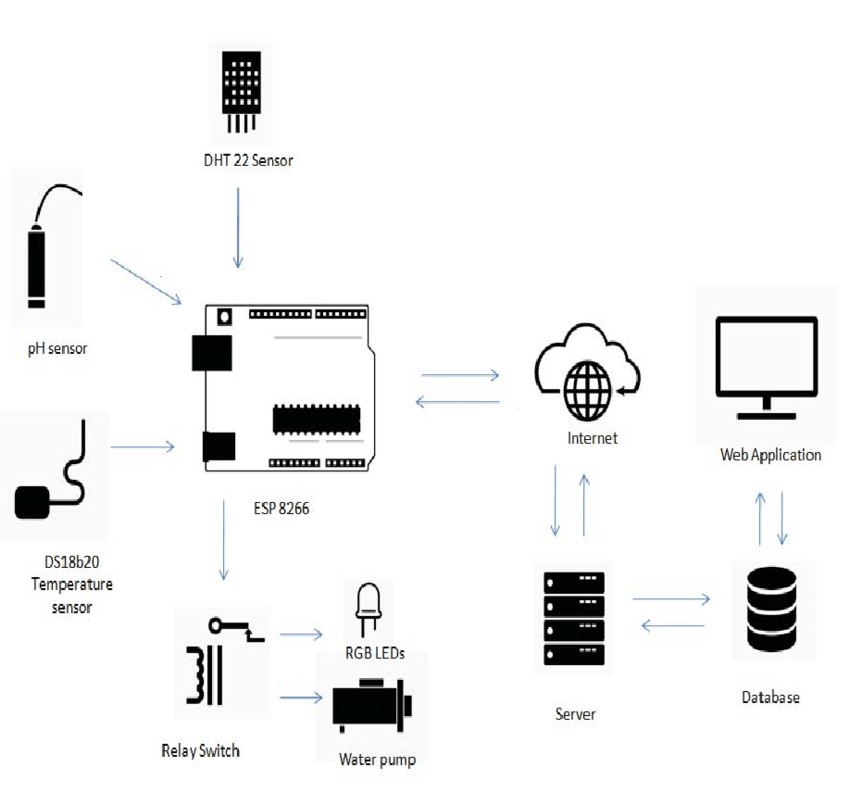
A little DFD is essentially a high-level overview of how data moves within a system. For an automated hydroponics system, this diagram illustrates the main components (or nodes) involved in data acquisition, processing, and control, along with the key information exchanges between them.

### 6.1 SYSTEM ARCHITECTURE



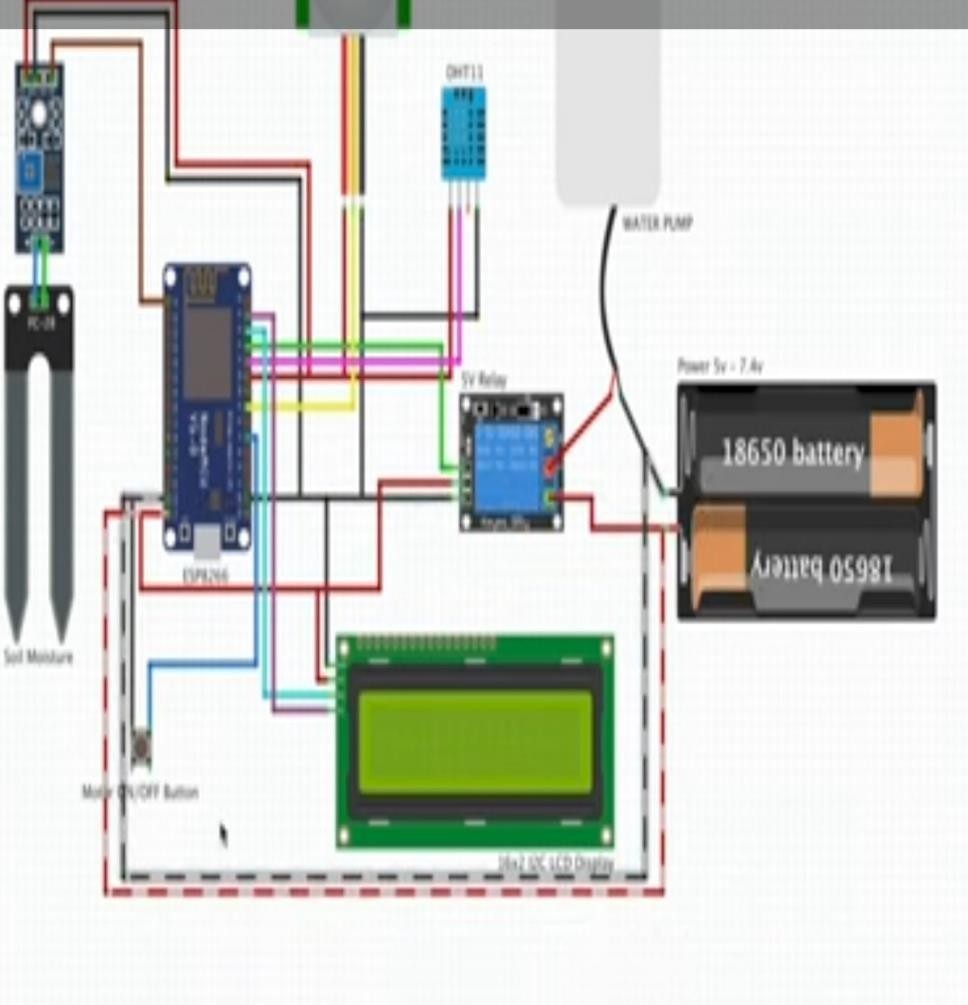
### .2 SYSTEM ARCHITECTURE

The automated hydroponics system is designed to continuously monitor and adjust environmental conditions to optimize plant growth. The architecture is divided into three major layers: the Hardware Layer (data acquisition and actuation), the Middleware/Communication Layer (data transmission and processing), and the Software Layer (data storage, processing, and user interaction). This modular design facilitates scalability, remote management, and seamless integration of various technological components.



### .3 Circuit Diagram

Provide a detailed schematic of all hardware connections.



**Include a detailed schematic showing connections between sensors, ESP8266, relay, and power supply.**

### .3 Flow Chart: -

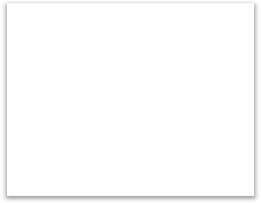
Data

Collection

sensors

)

(

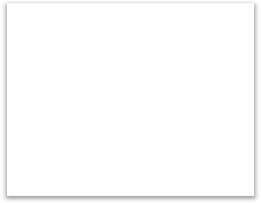


Data Processing

IoT

(

)



Decision Making

(

Weather API

)



User

Interaction

## CHAPTER 7

**HARDWARE/SOFTWARE PLATFORM**

**ENVIRONMENT**

### 7.1 Hardware Platform Environment

The foundation of the automated hydroponics system is its hardware, which comprises a suite of sensors and actuators designed to monitor key environmental parameters and control irrigation. The main components include:

* **ESP8266 WiFi Module:**

Serving as the central microcontroller, the ESP8266 is responsible for aggregating sensor inputs and transmitting data wirelessly. Its built-in WiFi capabilities make remote data transmission and system control possible, even in environments that may not have direct access to wired networks.

* **DHT11 Temperature & Humidity Sensor:**

This sensor monitors ambient temperature and humidity levels. By providing continuous feedback on the growing environment, it ensures that conditions remain optimal for plant growth, enabling the system to trigger climate control responses when necessary.

* **Soil Moisture Sensor:**

Although utilized in a hydroponic setting, sensors like the soil moisture sensor (or its aquatic equivalent) help determine the water (or nutrient solution) content within the growing medium. The sensor detects when the moisture level is below the set threshold, prompting the irrigation routine to deliver the precise amount of water required.

* **PIR Motion Sensor:**

Integrated as a secondary security and monitoring measure, the PIR sensor detects unexpected human or animal presence around the system. This function is especially beneficial in ensuring that the controlled environment is not tampered with or disturbed.

* **Relay Module & Water Pump Motor:**

The relay module permits the microcontroller to interface with high-power devices safely. It controls the water pump motor that is responsible for circulating the nutrient solution. When environmental data indicates that irrigation is necessary, the system engages the relay to activate the pump, ensuring that plants receive timely water delivery.

This hardware ensemble works reliably even in physically challenging environments—whether positioned in an indoor laboratory setting or on an outdoor rooftop farm—providing robust, realtime, and automated control over the system's operation.

### 7.2 Software Platform Environment

The software environment in the hydroponics system is designed to facilitate robust data processing, real-time user interaction, and seamless scalability. The software stack comprises multiple layers:

* **Frontend:**

Built with **HTML, CSS, JavaScript, and React.js**, the frontend provides a dynamic and responsive dashboard that displays real-time sensor data, historical trends, and system statuses. This interactive user interface allows end-users to configure system parameters such as moisture thresholds and irrigation schedules and includes features for manual override, live notifications, and alert systems.

* **Backend:**

Developed using **Node.js** and **Express**, the backend serves as the central processing engine. It:

* + Receives and parses incoming data from the ESP8266.
  + Executes decision-making algorithms (for example, determining when the water pump should be activated based on soil moisture levels). o Exposes RESTful APIs to allow secure communication with both the frontend and the hardware components.

* **Databases:**

The system utilizes two distinct data storage solutions:

* + **MongoDB:** Handles the storage of historically structured sensor data, system logs, and configuration parameters, enabling trend analysis and performance monitoring.
  + **Firebase Realtime Database:** Facilitates real-time data synchronization, ensuring that the frontend dashboard reflects the most current state of the system without delay.

This layered software environment supports both batch processing (via MongoDB for historical analysis) and immediate, real-time operations (via Firebase), ensuring that the system is responsive and capable of adapting to rapid environmental changes.

### 7.3 Integration and Communication Infrastructure

A well-coordinated integration between the hardware and software components is critical to the success of the automated hydroponics system. The following elements are key to this integration:

* **Data Transmission and Protocols:**

The ESP8266 transmits sensor data in structured JSON format via WiFi, using a secure

HTTP protocol to interact with the backend. RESTful APIs developed in Node.js and Express handle these inbound data streams, parse the information, and execute corresponding control actions.

* **Real-Time Synchronization:**

The Firebase Realtime Database plays an essential role by pushing live data updates to the frontend. This bidirectional communication ensures that any change—whether it's a sensor reading or a system override command—is immediately reflected across the platform, thereby maintaining system integrity.

* **Feedback Loops:**

A control feedback mechanism enables the backend to send commands (such as activating the relay for the water pump) back to the hardware. This closed-loop ensures that the system adapts continuously to the environmental conditions reported by the sensors.

This integrated architecture bridges the physical hardware with the cloud-based software environment, ensuring a seamless and robust operation that can be managed remotely and in real-time.

### 7.4 Operating Environment and Conditions

The automated hydroponics system is designed to function effectively under a variety of environmental conditions:

* **Physical Environment:**

The hardware is built to withstand fluctuations in temperature and humidity while operating reliably within the confines of both controlled indoor environments and variable outdoor conditions. Power supply considerations (such as integration with solar panels or energy-efficient power management systems) ensure continuous operation.

* **Network Environment:**

The use of the ESP8266 and wireless WiFi networks enables connectivity even in locations where wired infrastructure is unavailable. However, considerations for network security and reliability are embedded from the ground up, with data encryption and secure communication protocols in place.

* **User and Development Environment:**

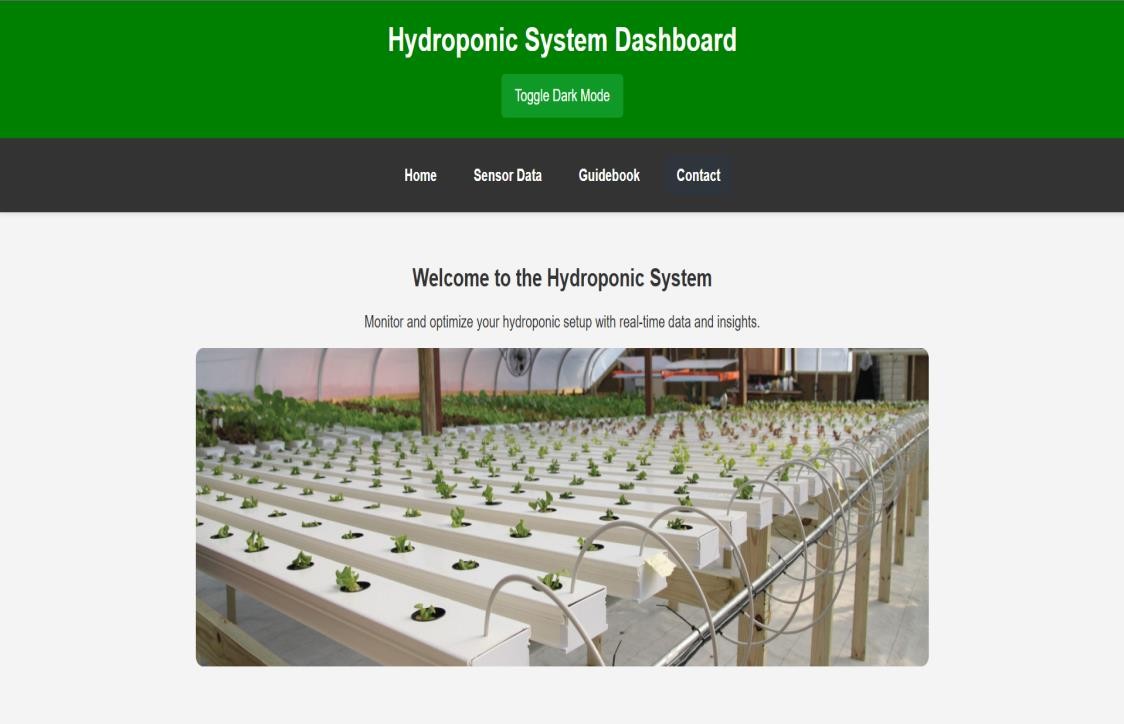
Developers working on the frontend and backend platforms are equipped with modern development tools (like Visual Studio Code, Git, and containerization systems for

deployment) that support agile development and rapid scaling.

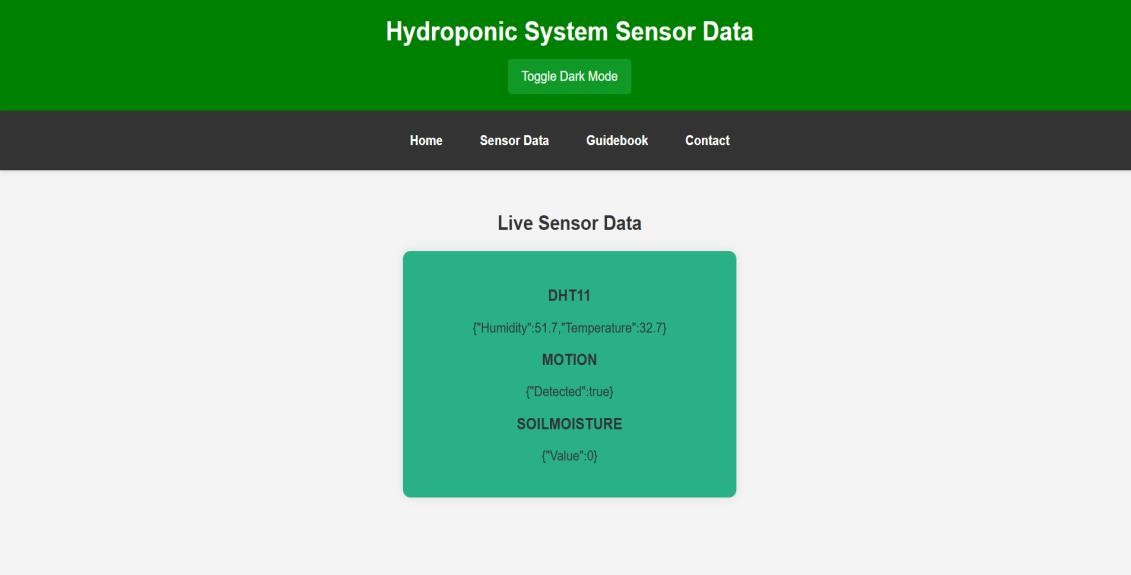
## CHAPTER 8

**SNAPSHOTS OF INPUT & OUTPUT**

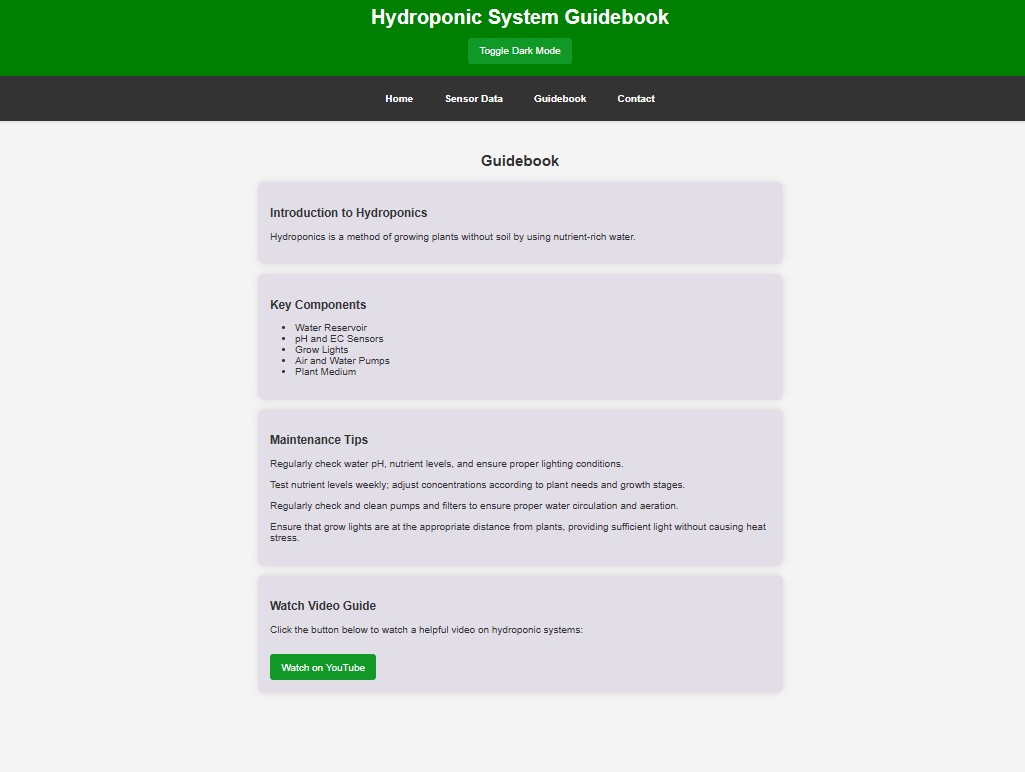
1. HOMEPAGE



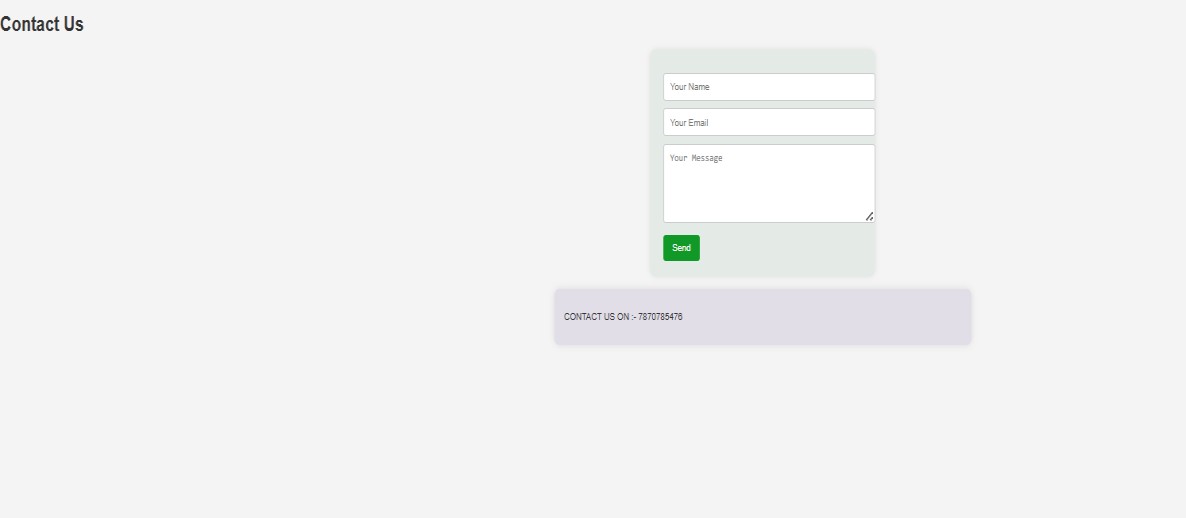
1. SENSOR DATA PAGE



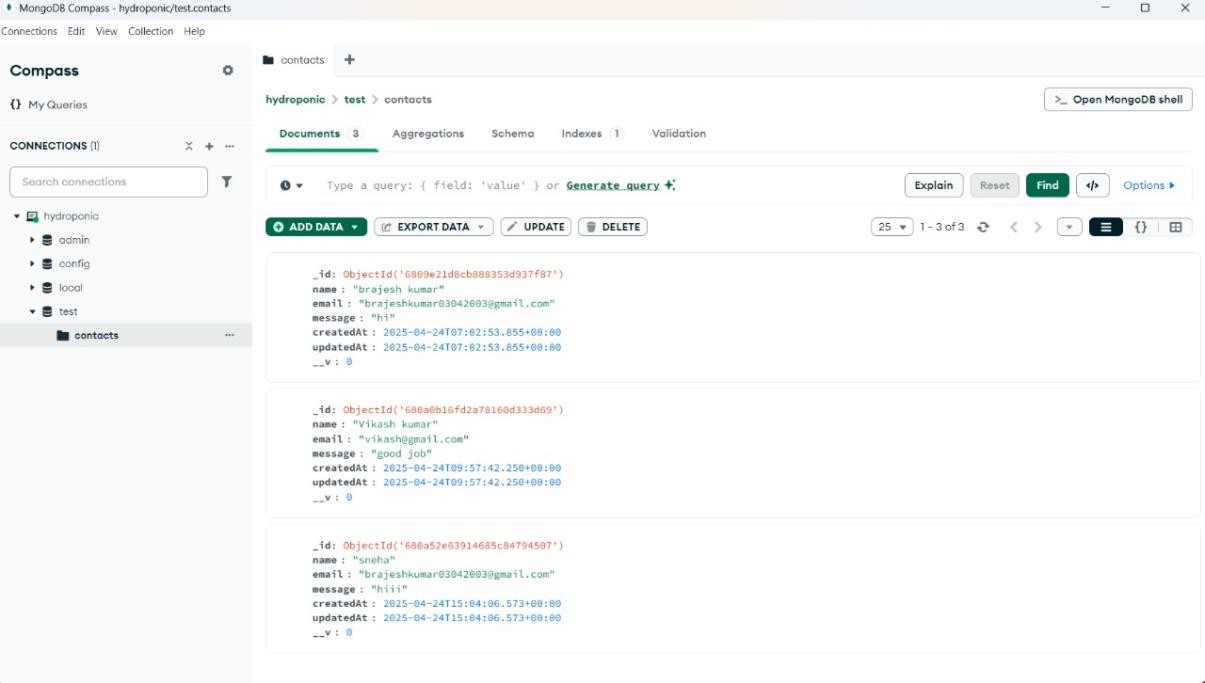
1. GUIDEBOOK



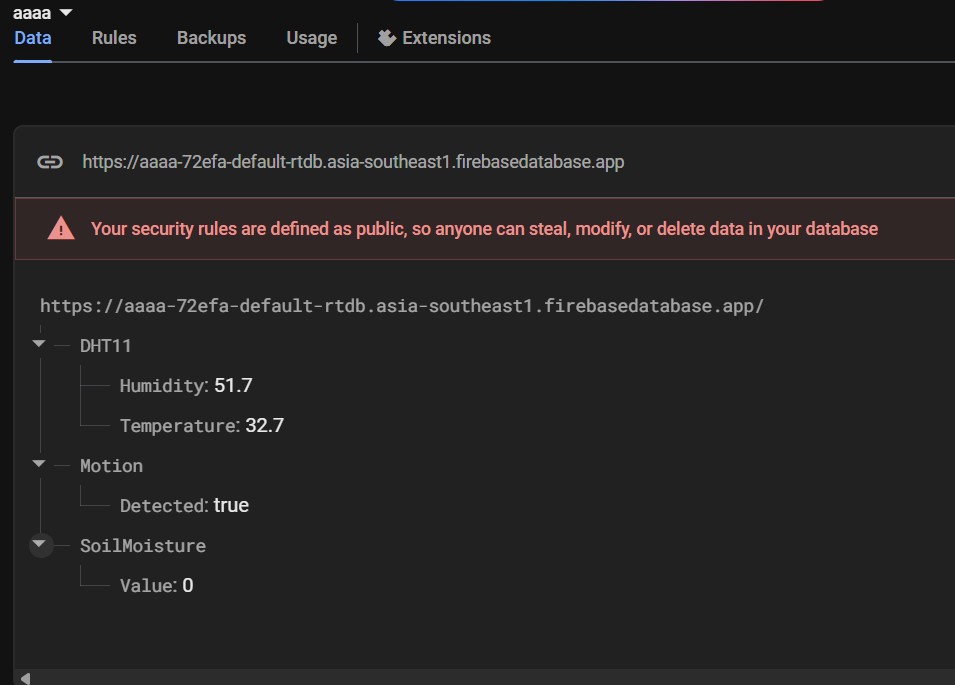
1. CONTACT PAGE



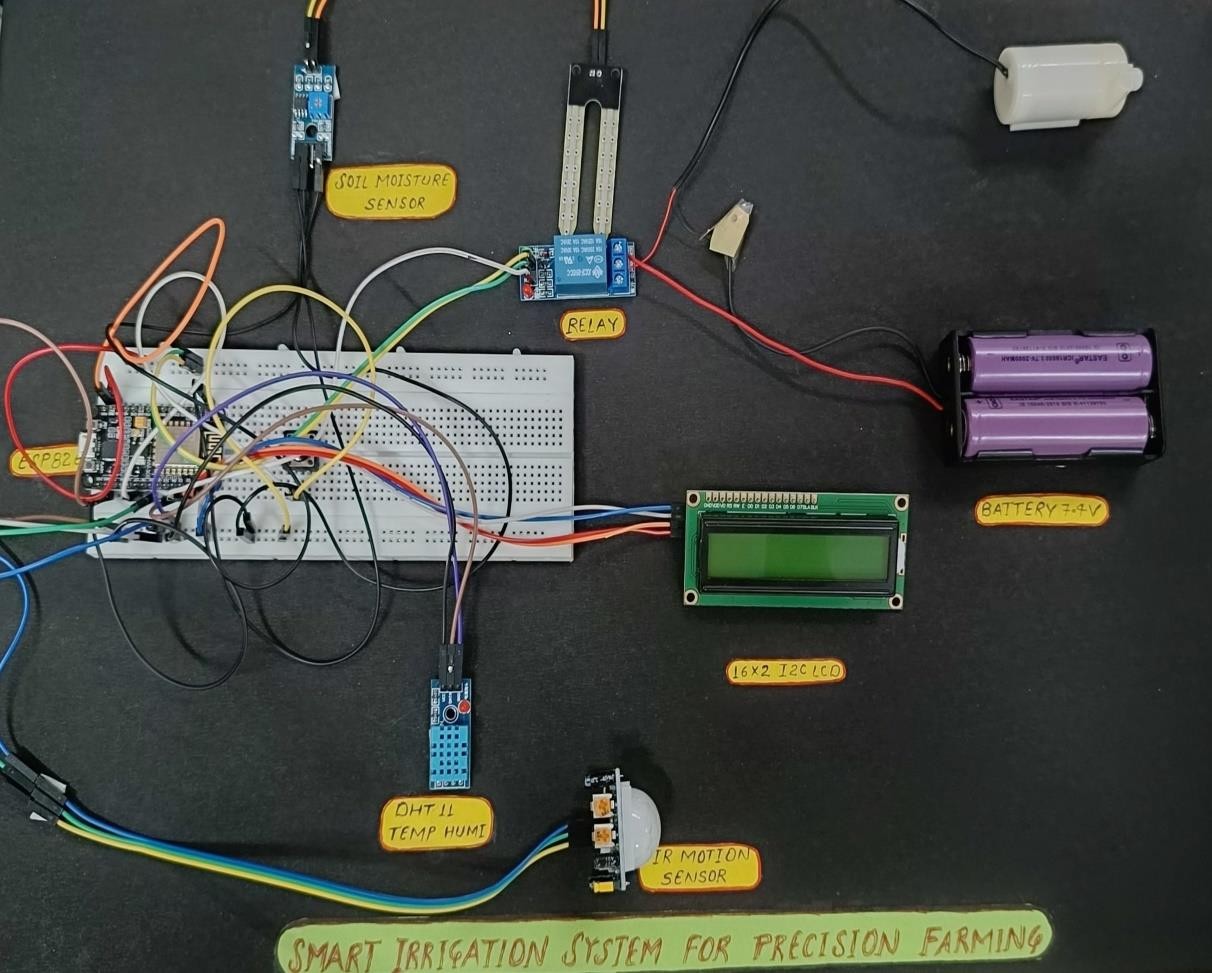
1. MONGODB



6.FIREBASE



### Project setup;-



## CHAPTER 9

**CODE**

### 9.1 CONTACT.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Contact - Hydroponic System</title>

<link rel="stylesheet" href="styles.css">

</head>

<body>

<h1>Contact Us</h1>

<form id="contact-form">

<input type="text" id="name" placeholder="Your Name" required>

<input type="email" id="email" placeholder="Your Email" required>

<textarea id="message" placeholder="Your Message" required></textarea> <button type="submit">Send</button>

</form>

<article>

<p>CONTACT US ON :- 7870785476</p>

</article>

<p id="status-msg"></p>

<script>

document.getElementById("contact-form").addEventListener("submit", async function (e) { e.preventDefault();

const data = {

name: document.getElementById("name").value, email: document.getElementById("email").value, message: document.getElementById("message").value

};

try {

const response = await fetch("http://localhost:5000/api/contact", { method: "POST", headers: { "Content-Type": "application/json" }, body: JSON.stringify(data)

});

const result = await response.json(); document.getElementById("status-msg").textContent = result.message || "Message sent!"; document.getElementById("contact-form").reset();

} catch (err) {

document.getElementById("status-msg").textContent = "Submission failed!";

}

});

</script>

</body>

</html>

### 9.2 GUIDEBOOK.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Guidebook - Hydroponic System</title>

<link rel="stylesheet" href="styles.css">

</head>

<body>

<header>

<h1>Hydroponic System Guidebook</h1>

<button id="toggle-dark-mode">Toggle Dark Mode</button>

</header>

<nav>

<a href="index.html">Home</a>

<a href="sensor-data.html">Sensor Data</a>

<a href="guidebook.html">Guidebook</a>

<a href="contact.html">Contact</a>

</nav>

<main>

<section id="guidebook">

<h2>Guidebook</h2>

<article>

<h3>Introduction to Hydroponics</h3>

<p>Hydroponics is a method of growing plants without soil by using nutrient-rich water.</p>

</article>

<article>

<h3>Key Components</h3>

<ul>

<li>Water Reservoir</li>

<li>pH and EC Sensors</li>

<li>Grow Lights</li>

<li>Air and Water Pumps</li>

<li>Plant Medium</li>

</ul>

</article>

<article>

<h3>Maintenance Tips</h3>

<p>Regularly check water pH, nutrient levels, and ensure proper lighting conditions.</p> <p>Test nutrient levels weekly; adjust concentrations according to plant needs and growth stages. </p>

<p>Regularly check and clean pumps and filters to ensure proper water circulation and aeration.</p>

<p>Ensure that grow lights are at the appropriate distance from plants, providing sufficient light without causing heat stress.</p>

</article>

<article>

<h3>Watch Video Guide</h3>

<p>Click the button below to watch a helpful video on hydroponic systems:</p>

<button onclick="window.open('https://youtu.be/evX94tEQC1Q', '\_blank')"> Watch on YouTube

</button>

</article>

</section>

</main>

<script src="script.js"></script>

</body>

</html>

### 9.3 INDEX.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Hydroponic System</title>

<link rel="stylesheet" href="styles.css">

<script defer src="script.js"></script>

</head>

<body>

<header>

<h1>Hydroponic System Dashboard</h1>

<button id="toggle-dark-mode">Toggle Dark Mode</button>

</header>

<nav>

<a href="index.html">Home</a>

<a href="sensor-data.html">Sensor Data</a>

<a href="guidebook.html">Guidebook</a>

<a href="contact.html">Contact</a>

</nav>

<main>

<section id="home">

<h2>Welcome to the Hydroponic System</h2>

<p>Monitor and optimize your hydroponic setup with real-time data and insights.</p> <img src="https://www.growspan.com/wp-content/uploads/2023/05/CS-Mantua-Gardens06e1685106121666.jpg" alt="Hydroponic System" style="width:200%; max-width:1000px; borderradius:10px;">

</section>

</main>

</body>

</html>

### 9.4 Script.js

document.addEventListener("DOMContentLoaded", function () { const dataContainer = document.getElementById("data-container"); const darkModeToggle = document.getElementById("toggle-dark-mode"); const API\_URL = "https://aaaa-72efa-default-rtdb.asia-southeast1.firebasedatabase.app/.json";

async function fetchData() {

try {

const response = await fetch(API\_URL);

if (!response.ok) throw new Error("Network response was not ok");

const data = await response.json(); displayData(data); } catch (error) {

console.error("Fetch error:", error); dataContainer.innerHTML = "<p class='error-message'>Error fetching data.</p>";

}

}

function displayData(data) {

dataContainer.innerHTML = "";

if (data) {

for (const key in data) { const itemDiv = document.createElement("div"); itemDiv.className = "sensor-card"; itemDiv.innerHTML = `

<h3>${key.replace(/\_/g, ' ').toUpperCase()}</h3>

<p>${JSON.stringify(data[key])}</p>

`;

dataContainer.appendChild(itemDiv);

}

} else {

dataContainer.innerHTML = "<p class='no-data'>No data found.</p>";

}

}

if (darkModeToggle) { darkModeToggle.addEventListener("click", function () {

document.body.classList.toggle("dark-mode");

});

}

if (window.location.pathname.includes("sensor-data.html")) { fetchData(); setInterval(fetchData, 5000); // Fetch data every 5 seconds

}

});

### 9.5 Sensordata.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Sensor Data - Hydroponic System</title>

<link rel="stylesheet" href="styles.css">

<script defer src="script.js"></script>

</head>

<body>

<header>

<h1>Hydroponic System Sensor Data</h1>

<button id="toggle-dark-mode">Toggle Dark Mode</button>

</header>

<nav>

<a href="index.html">Home</a>

<a href="sensor-data.html">Sensor Data</a>

<a href="guidebook.html">Guidebook</a>

<a href="contact.html">Contact</a>

</nav>

<main>

<section id="sensor-data">

<h2>Live Sensor Data</h2>

<div id="data-container">Fetching data...</div>

</section>

</main>

</body>

</html>

### 9.6 Server,js

const express = require('express'); const mongoose = require('mongoose');

const cors = require('cors');

const app = express();

const PORT = 5000;

app.use(cors());

app.use(express.json());

// MongoDB connection (local) mongoose.connect('mongodb://0.0.0.0/test', { useNewUrlParser: true, useUnifiedTopology: true,

})

.then(() => console.log("MongoDB connected"))

.catch((err) => console.log("Mongo error:", err));

// Mongoose schema

const contactSchema = new mongoose.Schema({ name: String, email: String, message: String,

}, { timestamps: true });

const Contact = mongoose.model('Contact', contactSchema);

// API route

app.post('/api/contact', async (req, res) => { try {

const newContact = new Contact(req.body); await newContact.save();

res.json({ message: 'Message received!' });

} catch (error) {

res.status(500).json({ message: 'Failed to save contact', error });

}

});

// Test route

app.get("/", (req, res) => {

res.send("Hydroponic backend is running.");

});

app.listen(PORT, () => { console.log(`Server is running on port ${PORT}`);

});

### 9.7 Style.css

/\* General Styles \*/ body { font-family: Arial, sans-serif; margin: 0; padding: 0; background-color: #f4f4f4; color: #333; transition: background 0.3s, color 0.3s;

}

header { background: green; color: white; padding: 20px; text-align: center;

box-shadow: 0px 2px 8px rgba(0, 0, 0, 0.2);

}

header h1 { margin: 0; font-size: 2em;

}

/\* Navigation \*/ nav { display: flex; justify-content: center; background: #333; padding: 12px; flex-wrap: wrap;

box-shadow: 0px 2px 5px rgba(0, 0, 0, 0.1);

}

nav a { color: white; margin: 5px 10px; text-decoration: none; padding: 10px 15px; font-weight: bold; border-radius: 5px; transition: background 0.3s ease;

}

nav a:hover { background: #0056b3;

}

/\* Main Content \*/ main { padding: 30px 20px; text-align: center;

}

/\* Sensor Data Container \*/ #data-container { background: rgb(41, 176, 135); padding: 25px; border-radius: 10px; box-shadow: 0px 0px 12px rgba(0, 0, 0, 0.1); max-width: 400px; margin: 20px auto

}

/\* Guidebook Articles \*/ article { background: rgb(225, 222, 232); padding: 20px; margin: 15px auto; max-width: 800px; border-radius: 10px; box-shadow: 0px 0px 12px rgba(0, 0, 0, 0.1);

text-align: left;

}

/\* Contact Form \*/ form { background: rgb(228, 234, 229); padding: 25px; border-radius: 10px; box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1); max-width: 400px; margin: 20px auto;

}

input, textarea { width: 100%; padding: 12px; margin-top: 12px; border: 1px solid #ccc; border-radius: 5px; font-size: 16px;

}

textarea { resize: vertical; min-height: 100px;

}

button { background: #119928; color: white;

border: none; padding: 12px 18px; margin-top: 15px; border-radius: 5px; cursor: pointer; font-size: 16px; transition: background 0.3s;

}

button:hover { background: #0056b3;

}

#status-msg { margin-top: 15px; font-weight: bold;

}

/\* Dark Mode \*/ .dark-mode { background-color: #333; color: white;

}

.dark-mode header, .dark-mode nav { background: #222;

}

.dark-mode form,

.dark-mode #data-container, .dark-mode article { background: #444; color: white; box-shadow: 0px 0px 10px rgba(255, 255, 255, 0.1);

}

.dark-mode input, .dark-mode textarea { background: #555; color: white; border: 1px solid #888;

}

### 9.8 Arduino Ide

#include <Wire.h>

#include <Adafruit\_SSD1306.h>

#include <ESP8266WiFi.h>

#include <FirebaseESP8266.h>

#include <DHT.h>

// WiFi Credentials

#define WIFI\_SSID "VarunSingh1airfibre5g"

#define WIFI\_PASSWORD "8463024454"

// Firebase Credentials

#define FIREBASE\_HOST "https://aaaa-72efa-default-rtdb.asia-southeast1.firebasedatabase.app/"

#define FIREBASE\_AUTH "fhNY1y2C06Krzliv9j2DnqRHKZNceOkzLbTGicHk"

// OLED Display

#define SCREEN\_WIDTH 128

#define SCREEN\_HEIGHT 64

Adafruit\_SSD1306 display(SCREEN\_WIDTH, SCREEN\_HEIGHT, &Wire, -1);

// DHT11 Sensor

#define DHTPIN D6

#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

// Soil Moisture Sensor

#define SOIL\_MOISTURE\_PIN A0

// PIR Motion Sensor

#define PIR\_SENSOR\_PIN D5

// Buzzer

#define BUZZER D1

// Firebase objects

FirebaseData firebaseData;

FirebaseAuth auth;

FirebaseConfig config;

void setup() {

Serial.begin(57600);

// Connect to Wi-Fi

WiFi.begin(WIFI\_SSID, WIFI\_PASSWORD);

Serial.print("Connecting to Wi-Fi");

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");

}

Serial.println("\nConnected!");

// Set up Firebase

config.host = FIREBASE\_HOST; config.signer.tokens.legacy\_token = FIREBASE\_AUTH;

Firebase.begin(&config, &auth);

Firebase.reconnectWiFi(true);

// Initialize DHT11 Sensor

dht.begin();

// Initialize Buzzer

pinMode(BUZZER, OUTPUT);

digitalWrite(BUZZER, LOW);

// Initialize PIR Motion Sensor

pinMode(PIR\_SENSOR\_PIN, INPUT);

// Initialize Soil Moisture Sensor

pinMode(SOIL\_MOISTURE\_PIN, INPUT);

// Initialize I2C for OLED (SDA on D2, SCL on D3)

Wire.begin(D2, D3);

if (!display.begin(SSD1306\_SWITCHCAPVCC, 0x3C)) { Serial.println("SSD1306 allocation failed"); for (;;);

}

display.clearDisplay();

}

void playBuzzer(int frequency, int duration) { tone(BUZZER, frequency, duration);

delay(duration);

noTone(BUZZER);

}

void loop() { float temperature = dht.readTemperature(); float humidity = dht.readHumidity();

int soilMoistureValue = analogRead(SOIL\_MOISTURE\_PIN); int soilMoisturePercent = map(soilMoistureValue, 1023, 0, 0, 100);

bool motionDetected = digitalRead(PIR\_SENSOR\_PIN);

if (isnan(temperature) || isnan(humidity)) { Serial.println("Failed to read from DHT sensor!"); return;

}

Serial.print("Temperature: ");

Serial.print(temperature);

Serial.print("°C Humidity: ");

Serial.print(humidity);

Serial.print("% Soil Moisture: ");

Serial.print(soilMoisturePercent);

Serial.print("% Motion: ");

Serial.println(motionDetected ? "Detected" : "Not Detected");

// Send Data to Firebase

Firebase.setFloat(firebaseData, "/DHT11/Temperature", temperature);

Firebase.setFloat(firebaseData, "/DHT11/Humidity", humidity);

Firebase.setInt(firebaseData, "/SoilMoisture/Value", soilMoisturePercent);

Firebase.setBool(firebaseData, "/Motion/Detected", motionDetected);

// Motion Detected - Buzzer ON for 2 sec at 6000Hz if (motionDetected) {

Serial.println(" Motion Detected! Activating Buzzer for 2 seconds."); playBuzzer(6000, 2000);

}

// Buzzer Warning for High Temp or Low Moisture if (temperature > 30 || soilMoisturePercent < 30) {

Serial.println(" Warning! High Temperature or Low Soil Moisture!"); playBuzzer(5000, 300);

}

// Update OLED Display display.clearDisplay(); display.setTextSize(1); display.setTextColor(WHITE); display.setCursor(0, 0); display.println("Sensor Data:"); display.print("Temp: "); display.print(temperature); display.println(" C"); display.print("Humidity: "); display.print(humidity); display.println(" %"); display.print("Moisture: "); display.print(soilMoisturePercent); display.println(" %"); display.print("Motion: "); display.println(motionDetected ? "Yes" : "No"); display.display();

delay(5000); // Update every 5 seconds }

**CHAPTER 10**

**Project limitation and Future scope**

### 10.1 Project Limitations

While the implemented automated hydroponics system marks a significant step toward precision agriculture, several limitations have been identified that may affect performance, scalability, and overall usability:

1. **Hardware Constraints:**

The system utilizes components such as the ESP8266 WiFi module, DHT11 sensor, soil moisture sensor, PIR sensor, relay module, and water pump. Although these components offer a cost-effective solution for small-scale implementations, they come with inherent drawbacks. For instance, the DHT11 sensor is not as precise or durable as more advanced alternatives, and the moisture sensor may require frequent calibration or replacement over time. Additionally, the ESP8266, while excellent for basic data collection and transmission, has limited processing power and memory, which may hamper the system’s ability to handle more complex decision-making or larger sensor networks.

1. **Software and Data Processing Limitations:**

The backend, built on Node.js and Express, effectively manages real-time control and data logging. However, the reliance on RESTful APIs and intermittent connectivity can lead to occasional latency or data synchronization issues—especially when integrating MongoDB for batch historical data storage with Firebase Realtime Database for live updates. In instances where network connectivity is weak or unstable, delayed sensor readings or control signals may result in less-than-optimal actuation of the system (for instance, delayed water pump activation).

1. **Scalability Challenges:**

Currently, the system is prototyped for a limited number of sensors and a controlled environmental setup. Scaling up from a small-scale prototype to a commercial or industrial hydroponics installation requires additional consideration. The architecture may encounter bottlenecks if more sensors, diverse actuator types, or higher volumes of real-time data are introduced. Moreover, ensuring robust performance in a multi-node environment (such as vertical farms or large greenhouses) would demand upgrades in both hardware (using more robust microcontrollers) and software (improved data management and analysis algorithms)

1. **Environmental and Operational Factors:**

The system’s performance is also subject to external factors such as power supply stability, extreme environmental conditions, and network reliability. In outdoor or variable ambient conditions, hardware components might degrade faster or provide inconsistent readings. Additionally, the reliance on WiFi connectivity means that areas with poor network infrastructure could face challenges in remote monitoring and control.

1. **Security and Data Privacy Concerns:**

As with any IoT-based system, ensuring end-to-end security is critical. While the current implementation employs standard encryption protocols for data transmission, the architecture would need to be fortified against emerging cyber threats—especially when expanding the system or integrating it with broader smartcity or industrial applications.

### 10.2 Future Scope

Despite these limitations, the automated hydroponics system presents numerous opportunities for advancement and adaptation. Future work could address current constraints while expanding its capabilities:

1. **Enhanced Sensor Technologies:**

Future iterations could integrate higher-precision sensors, such as digital temperature sensors with wider operating ranges, advanced pH and nutrient concentration monitors, and robust atmospheric sensors. This would not only improve data accuracy but also allow the system to adapt better to diverse and challenging farming environments.

1. **Improved Processing and Edge Computing:**

Shifting more processing power to the edge by using advanced microcontrollers or IoT gateways (such as Raspberry Pi or similar devices) can facilitate local data processing. This may reduce dependency on constant cloud connectivity and enable more sophisticated real-time control algorithms, such as adaptive feedback loops based on machine learning models.

1. **Scalability for Commercial and Vertical Farming:**

As hydroponic farming moves toward commercial scale and urban agriculture, the system architecture can be extended by adopting modular, scalable designs. Integration with Industrial IoT (IIoT) platforms can support larger sensor networks, enhance data aggregation, and provide comprehensive decision support systems. Furthermore, cloud-based machine learning models could be employed to predict plant growth trends, manage nutrient cycles, and automate resource allocation across larger installations.

1. **Integration with Renewable Energy and Sustainability Practices:**

Future designs could incorporate renewable energy sources—such as solar, wind, or even bio-energy—to make the system more self-sustaining, thereby reducing operational costs and environmental impact. This includes the development of power management algorithms that optimize energy consumption based on real-time usage patterns

1. **Advanced Security Measures:**

To address data privacy and cybersecurity challenges, stronger security protocols (including advanced encryption, multi-factor authentication, and possibly blockchain-based data integrity solutions) can be implemented. This ensures that as the system scales and becomes a part of larger networks, the integrity of data and control processes remains uncompromised.

1. **Enhanced User Interface and Analytics:**

The frontend can evolve from a basic monitoring dashboard to a comprehensive control and analytics platform. Integrating real-time analytics, predictive maintenance alerts, and user-customizable dashboards will provide farmers and system administrators with deeper insights into system performance and crop health. Mobile app integrations, voice control, and virtual assistants could further enrich user experience.

1. **Interdisciplinary and Integrated Farming Solutions:**

In the long run, the technology behind automated hydroponics can be integrated with other agricultural systems. This might include interfacing with sensor networks in traditional farms or coupling with market data for supply chain optimization. Such integration could lead to a more resilient and interconnected food production ecosystem.

In summary, while the current version of the automated hydroponics system may face challenges in hardware accuracy, scalability, and data security, its future scope is promising. By incorporating advanced technologies, enhancing data processing capabilities, and expanding systems integration, the platform is well-poised to transform modern agriculture, particularly in urban and resource-constrained environments.

## CHAPTER 11

**REFERENCES**

1. **ESP8266 WiFi Module Datasheet**

Espressif Systems. Retrieved from [https://www.espressif.com/sites/default/files/documentation/0aesp8266ex\_datasheet\_en.pdf.](https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf)

1. **DHT11 Temperature & Humidity Sensor Datasheet**

Aosong Electronics Co., Ltd. Retrieved from [https://cdn-shop.adafruit.com/datasheets/DHT11.pdf.](https://cdn-shop.adafruit.com/datasheets/DHT11.pdf)

1. **Soil Moisture Sensor Guides and Specifications**

Various online resources provide specifications and best practices for soil moisture sensors used in DIY projects.

1. **Firebase Realtime Database Documentation**

Google Firebase. Retrieved from [https://firebase.google.com/docs/database.](https://firebase.google.com/docs/database)

1. **MongoDB Documentation**

MongoDB Inc. Retrieved from [https://docs.mongodb.com.](https://docs.mongodb.com/)

1. **Node.js and Express Documentation**

Node.js: <https://nodejs.org/en/docs/>

Express.js: [https://expressjs.com/en/starter/installing.html.](https://expressjs.com/en/starter/installing.html)

1. **React Documentation**

Facebook. Retrieved from [https://reactjs.org/docs/getting-started.html.](https://reactjs.org/docs/getting-started.html)

1. **InTechOpen – Automation and Robotics in Hydroponics**

Sample research articles and chapters available at InTechOpen provide insights into the integration of robotics and automation within hydroponics environments. For example, see: [InTechOpen Hydroponics Article.](https://www.intechopen.com/chapters/70662)

1. **Additional Research Papers on Automated Hydroponics** o *A Survey on Smart Hydroponics Farming* o *Automated Hydroponics System: IoT and AI Integration* o *A Survey on Hydroponic Methods of Smart Farming*

(Note: These titles represent sample references; actual source details should be used if they are available.)