A PROJECT REPORT

ON

"AUTOMATIC HYDROPONICS SYSTEM USING IOT"

Submitted To

Savitribai Phule Pune University

By

PRATIK R. WAGHMODE (B190653018)

For The Partial Fulfillment Of The Requirements For Project Stage II For Final Year Of Engineering (E&Tc).

(Accredited By NAAC 'A+' Grade)

Department of Electronics and Telecommunication

Trinity College of Engineering and Research

Kondhwa-Saswad Road, BopdevGhat, Pune, Maharashtra (India).

2023-2024

Trinity College of Engineering and Research, Pune

CERTIFICATE

This is to certify that, the project report titled

"AUTOMATIC HYDROPONICS SYSTEM USING IOT"

is a work carried out

By

PRATIK R. WAGHMODE (B190653018)

Under the supervision/guidance of

Prof. Mrs. Pranjali Deshmukh

During the academic sessions of year 2023 to 2024 and it is here by approved for the partial fulfillment of the mandatory requirement of Savitribai Phule Pune University, for Project, under the faculty of Final year of Engineering (E&TC), of Savitribai Phule Pune University.

Prof. Mrs. Pranjali Deshmukh	**Dr. S. M. Handore**	**Dr.A.B.Auti**
Project Guide	**HOD (E&TC)**	**principle**

Date:

Place: Pune. **External**

ACKNOWLEDGEMENT

We take a great pleasure in submitting the seminar report on “Automatic Hydroponics System Using IoT” for partial fulfillment of project examination of final year of engineering (E&TC). We could not have achieved this endeavor without support of many people in this organization and we would like to thank them.

Firstly, we are sincerely thankful to all professors for their guidance for our project. Without their help, it would have been a tough job for us. We really thank them for their relevant help and providing the necessary guidance.

We are highly encouraged by our project guide Prof.Mrs. Pranjali Deshmukh, who has devoted her time as and when we visited to her with some problems.

We are also proud to thank Dr. S. M. Handore, HOD, Department of Electronics & Telecommunication and our Principal Dr. A.B.Auti for permitting us to submit this project and for their moral support.

Last but not the least we would like to thank many other people in the department, without their help we could not have attained this hard success. Also, we thank all those who were involved directly and indirectly.

Name of candidates: -

PRATIK R. WAGHMODE

ABSTARCT

Hydroponics is a way of growing plants in water with nutrients instead of soil. This method allows for faster growth and higher yields compared to traditional farming. It's particularly useful in areas with poor soil quality. This study focuses on using the Internet of Things (IoT) to automate monitoring of hydroponic systems. By constantly checking factors like temperature, humidity, and nutrient levels, IoT systems can help reduce workload for farmers and optimize plant growth. These smart systems have been successful in growing fruits, lettuce, cucumbers, and tomatoes. IoT technology involves interconnected devices that can collect and share data without needing constant human intervention. These devices can connect to the internet, be controlled remotely, and come equipped with various sensors and tools. It reduces the amount of work they need to do. It helps plants grow better by keeping everything in the perfect balance. It can even be used to grow fruits and bigger plants like lettuce, tomatoes, and cucumbers. So, with the help of technology, farming can be more convenient and productive!

KEYWORDS: *IoT, Hydroponics, sensors, Esp32, GUI, Automation.*

CHAPTER NO: 1
INTRODUCTION

1.1 INTRODUCTION:

Hydroponics is a technique of using water and fertilizer solutions as the growing medium, increasing productivity through monitoring of environmental conditions compared to traditional agricultural methods. In a hydroponic system, plants are kept in tubs, and their roots float in nutrient-rich liquid, allowing them to develop rapidly and become a mass. Hydroponics, which translates to “water work”, comes from the Greek terms hydro, which means “water”, and ponos, which means “labor” .This type of irrigation is frequently used in areas where the soil is not fertile enough to support crop production .

The problem that this study attempts to solve is how to reduce farmers’ efforts in checking the elements that the plant needs without the requirement for farmer intervention. The hydroponics monitoring systems using Internet of Things (IoT) can be employed to reduce losses, optimize efficiency, increase productivity, and lessen the time and effort required. Monitoring the humidity, temperature, salinity, pH, oxygen, and nutritional levels of plants is an important aspect of farming. Smart systems are employed for their convenience, precision, and efficiency brought about by technological developments .This technique has been used to grow fruits properly and has produced good results for larger plants, particularly lettuce, cucumbers, and tomatoes .The Internet of Things (IoT) is a network of connected computers with mechanical and smart machinery, objects, living creatures, and people.

These devices have identifiers and the ability to communicate information with one another and computers without the need for direct human or computer involvement. They can link and interact with others through the web, be automatically evaluated and controlled, and are packed with tools, online networks, and other equipment like sensorsing.

1.2 MOTIVATION:

The motivation for developing an automatic hydroponics system with a Graphical User Interface (GUI) encompasses several key aspects that address the challenges and opportunities within modern agriculture, technology integration, and user experience enhancement. Here are the primary motivations:

1. Increased Efficiency and Productivity

Optimal Resource Utilization: Automating nutrient delivery, pH balancing, and environmental controls ensures plants receive the exact amounts needed for optimal growth, reducing waste and increasing yield.

Consistent Growth Conditions: Automation provides consistent and precise control over the growing environment, leading to healthier plants and faster growth cycles.

2. Accessibility and User Friendliness
Simplified Operation: A GUI makes hydroponics accessible to users of all skill levels, removing the complexity of manual adjustments and monitoring.
Real-Time Data Visualization: Users can easily view and understand the status of their system at a glance, making it easier to manage and troubleshoot.

3. Time and Labor Savings
Reduced Manual Labor: Automating routine tasks such as nutrient mixing, pH adjustments, and environmental monitoring frees up time for growers to focus on other important activities.

4. Scalability and Flexibility
Adaptability: A modular and scalable system can grow with the user's needs, whether they are hobbyists or commercial growers.
Customizability: Users can tailor the system settings to specific plants or growing conditions, providing flexibility for diverse crops and research applications.

1.3 OBJECTIVES:

Creating an automatic hydroponics system using a Graphical User Interface (GUI) involves a blend of automation technology, user-friendly design, and efficient plant management. The objectives of such a system can be outlined as follows:

1. Automation and Control
Nutrient Solution Management: Automatically mix and dispense the correct nutrient solutions to the plants based on their growth stage and needs.
pH and EC Monitoring: Continuously monitor and adjust the pH (potential hydrogen) and EC (electrical conductivity) levels of the nutrient solution to ensure optimal plant growth conditions.
Water Level Management: Automatically refill the water reservoir to maintain consistent water levels and prevent the system from drying out.

2. Environmental Monitoring and Control
Temperature and Humidity Control: Monitor and regulate the temperature and humidity of the growing environment to maintain optimal conditions for plant growth.
Light Control: Automate the lighting schedule using grow lights to mimic natural daylight cycles, ensuring plants receive adequate light for photosynthesis.

3. User Interface (GUI)
Ease of Use: Provide a user-friendly interface that allows users to easily monitor and control the hydroponics system without needing extensive technical knowledge.

Real-Time Monitoring: Display real-time data on water levels, pH, EC, temperature, humidity, and light intensity on the GUI.
Alerts and Notifications: Send alerts and notifications to the user in case of any abnormalities or if maintenance is required (e.g., low water levels, pH out of range).
Historical Data and Analytics: Store historical data and provide analytics to help users track plant growth, identify trends, and optimize growing conditions.

4. Customization and Flexibility
Preset and Custom Growth Programs: Offer preset growth programs for different plant types and allow users to create and save custom programs.
Remote Access: Allow users to access and control the system remotely via the internet, enabling them to monitor and adjust settings from anywhere.

5. Maintenance and Troubleshooting
System Diagnostics: Include diagnostic tools within the GUI to help users identify and troubleshoot issues with the system.
Maintenance Reminders: Provide reminders and instructions for routine maintenance tasks such as cleaning components, replacing filters, and checking nutrient levels.

1.4 Technical And Commercial Feasibility Of Project

Technical Feasibility

1. System Components

- **Sensors:** For monitoring parameters such as pH, temperature, humidity, light, and nutrient levels.
- **Actuators:** For controlling water pumps, nutrient dispensers, grow lights, and ventilation systems.
- **Microcontroller/Computer:** To process sensor data and control actuators. Options include Arduino, Raspberry Pi, or other microcontrollers.
- **Communication Modules:** For remote monitoring and control, e.g., Wi-Fi, Bluetooth, or Zigbee.

2. Software Development

- **Sensor Integration:** Code to read data from various sensors.
- **Control Algorithms:** Logic to maintain optimal growing conditions by controlling actuators.
- **Data Logging:** Storing historical data for analysis and optimization.
- **GUI Development:** A user-friendly interface to monitor and control the system. Could be web-based or a desktop application.

3. Automation

- **Real-time Monitoring:** Continuous monitoring and real-time adjustments.
- **Alerts and Notifications:** SMS or email alerts for critical conditions.

- **Machine Learning:** Potential for implementing predictive maintenance and growth optimization using data analytics.

Commercial Feasibility

1. Market Research

- **Demand Analysis:** Assessing the market demand for automated hydroponics systems.
- **Target Audience:** Identifying potential customers such as urban farmers, hobbyists, and commercial growers.
-

4. Distribution Channels

- **Online Sales:** Through e-commerce platforms and a dedicated website.
- **Retail Partnerships:** Collaboration with gardening stores and hydroponic equipment suppliers.

1.5 PROJECT PLANNING:

Time Span	Working
15/6/2023 to 30/6/2023	Group member selection and started searching various domains for project
1/7/2023 to 31/7/2023	Finalize project topic and studied various IEEE papers and made abstract for 1st review.
1/8/2023 to 31/8/2023	We made a first/ primary level project presentation on automatic Hydrophonics System Using IoT
1/9/2023 to 30/9/2023	Started Practical work on Automatic Hydrophonics system and Understanding Algorithms of Programming
1/10/2023 to 31/10/2023	Worked on Hydrophonics Database Collection
1/11/2023 to 30/11/2023	Completed programming for key features
1/12/2023 to 31/12/2023	Project Stage 1 Completed

1/1/2024 to 31/1/2024	Worked on Frontend and Other Web Paages
1/2/2024 to 30/2/2024	Worked on Backend and Linking Of Database
1/3/2024 to 31/3/2024	Collected data on hydrophonics system on gui
1/4/2024 to 30/4/2024	Completed with Project testing and implementation

CHAPTER NO: 2
LITERATURE REVIEW

2.1 LITERATURE REVIEW:

1. C. Treftz, ST. Omaye, Hydroponics: potential for augmenting sustainable food production in non-arable regions, Nutr. Food Sci. 46 (5) (2016) 672–684 Hydroponics system uses nutrient-rich water as the growing medium for plants. The use of hydroponic systems in agricultural technology has grown significantly. It has the potential to partially replace conventional soil-based growth methods in global food production.

2. Z. Zuriati, M. Apriyani, A.R. Supriyatna, Design and implementation automation system for hydroponic vegetable cultivation, in: Proceedings of the International Conference on Agriculture and Applied Science, 2021.One benefit of hydroponic growing systems is the ability to control environmental factors to maximize production in vertical gardens to constrained spaces. Other benefits include reducing water waste through recirculation, growing crops in controlled environments (such as monitoring nutrition, plant pests, and other aspects necessary for optimal growth of plants), and the ability to control circumstances to increase the output of vertical gardens in limited spaces.

3. K. Kularbphettong, U. Ampant, N. Kongrodj, An automated hydroponics system based on mobile application, Int. J. Inf. Educ. Technol. 9 (8) (2019) 548–552. Kularbphyttong et al. developed a mechanism for controlling plant growth. This system can regulate essential environmental elements that affect a plant's growth, such as temperature, humidity, and water. The application system automatically blends the chosen solution to determine the correct amount, collects data on the quantity of solution mixed during planting, and may be used to assess the cost of producing vegetables and determine the profitability of each produce to help with growth decisions.

4. S. Huo, J. Liu, M. Addy, P. Chen, D. Necas, P. Cheng, K. Li, H. Chai, Y. Liu, R. Ruan, The influence of microalgae on vegetable production and nutrient removal in greenhouse hydroponics, J. Clean. Prod. 243 (2020) 118563.Huo et al. explored the influence of microalgae on vegetable growth and evaluated the nutrient removal in the greenhouse for three kinds of vegetables produced hydroponically in greenhouses using nitrate-rich synthetic wastewater. The results indicated that most vegetable types produced more as a response to the use of microalgae as a technique of sustainable production system.

5. JCV. Puno, JJI. Haban, JD. Alejandrino, AA. Bandala, EP. Dadios, Design of a nutrient film technique hydroponics system with fuzzy logic control, in: Proceedings of the IEEE Region 10 Conference (Tencon), IEEE, 2020, pp. 403–408.Puno et al. developed a hydroponics system that combines fuzzy logic control and nutrient film technology.

Several crops can be grown within a limited space using this technique. The main parameters for crop survival were monitored, and fuzzy logic was used to drive the pumps for the tanks holding fresh water and nutrient concentrates. The drains of the mixing tanks were also monitored using data from sensors that detected electrical conductivity, pH, and water levels in the tank.

6. C. Ramos, L. Nóbrega, K. Baras, L. Gomes, Experimental NFT hydroponics system with lower energy consumption, in: Proceedings of the 5th Experiment International Conference (exp. at’19), IEEE, 2019, pp. 102–106. Ramos et al. developed a hydroponics system with a nutrient film technique to investigate and create algorithms that will enable effective water recirculation, resulting in electricity savings of about 40% compared to conventional systems.

CHAPTER NO: 3
STUDY OF DIFFERENT SYSTEMS

3.1 STUDY OF DIFFERENT SYSTEMS:

Here are descriptions of different systems available in automatic hydrophonics using IoT:

1.Bluetooth based automatic hydroponics system –
Bluetooth is a short-ranged wireless technology which is generally used to establish communication between several different devices for transferring of media or instructions. It uses radio waves having short wavelengths that cannot cover up large distances (maximum 100m) It can be used to connect devices. The system allows multiple device controllers to be connected to the host controller. In some ideal conditions bluetooth has the highest up to some 100m range. Comparatively bluetooth communication usually consumes higher power, so the batteries of devices need to be frequently recharged or replaced. Bluetooth technology should only be used when there is quick short lived communication with a very small concern of security.

2.Wi-fi based automatic hydroponics system –
Wireless-Fidelity which is popularly known as Wi-Fi uses radio waves for the transmission of data. It provides high-speed internet and network connections. It is a wireless medium for communicating to different locations in the house and connecting different devices. It can be used in variety of specification which varies with the purpose. Equipment can be placed anywhere. No unnecessary cords are required in your home. There is no need for additional Ethernet output and it also provides a wide range and is more efficient. Wi-Fi is a popular choice among people.

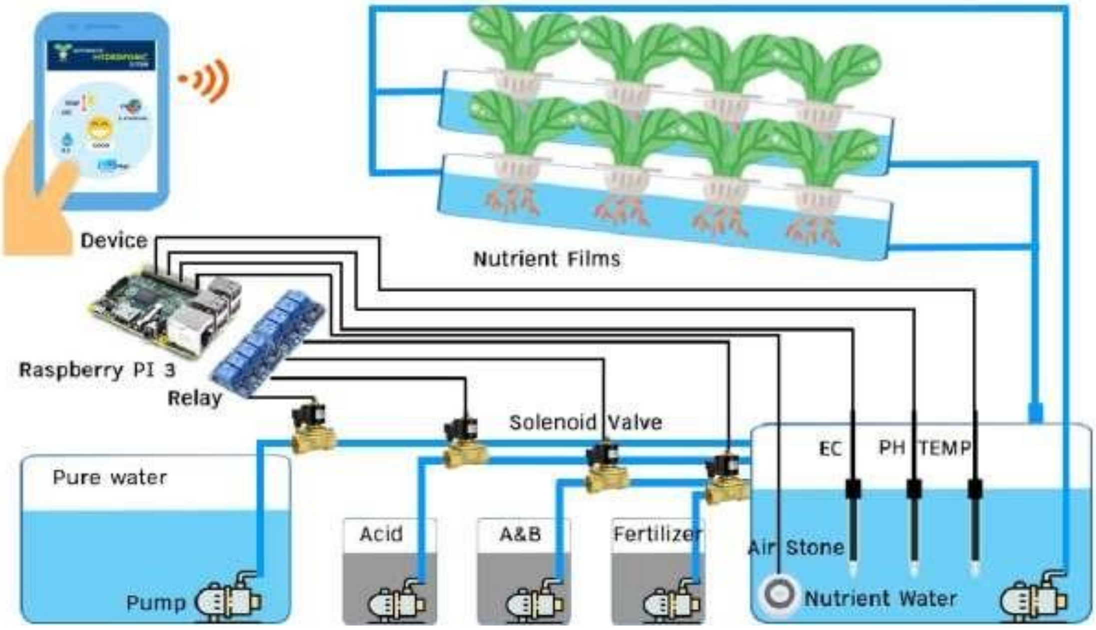
3. Mobile based automatic hydroponics –
Mobile based smart home light system are striking to companies because of the fame of mobile phones and GSM. This system detects the illegitimate invasions at home and allows only legitimate users to alter the passkey for the gate and control lights in the house. The illegitimate invasions into the home are identified by monitoring the state of the home door which is done using sensors. In this system an android application made to run on the user's mobile phone. Legitimate users can log in to the application using their username and password and remotely control along with some of the functions from the list of available user actions. The application will send the required notification to the user.

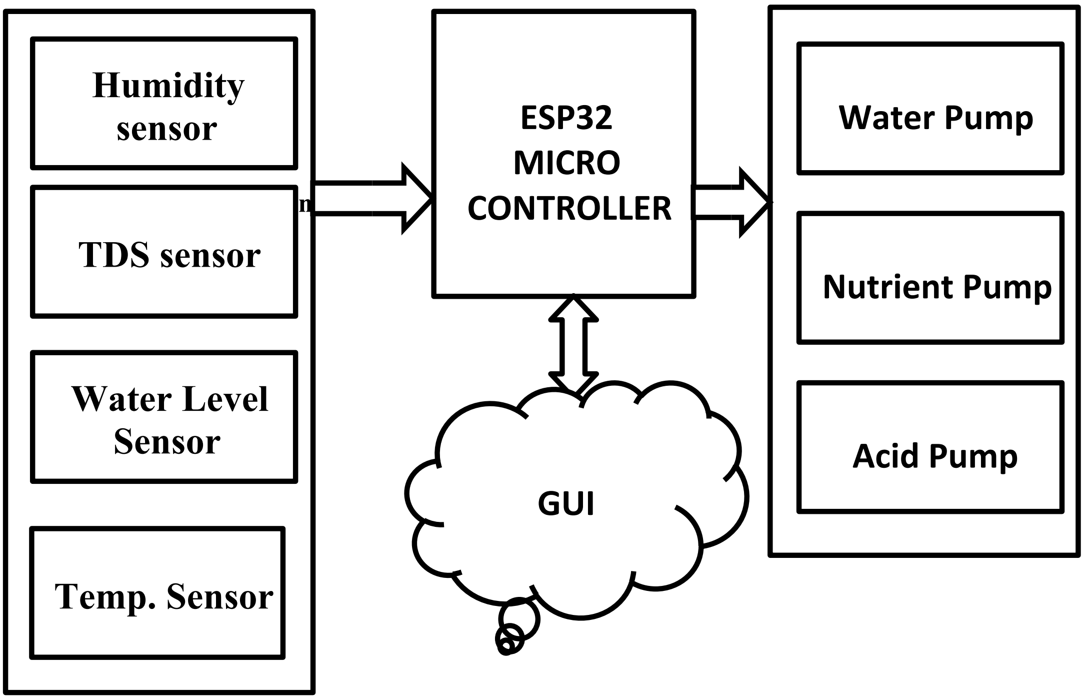
CHAPTER 4

BLOCK DIAGRAM

GENERAL PROJECT DESCRIPTION

4.1 DESCRIPTION OF BLOCK DIAGRAM



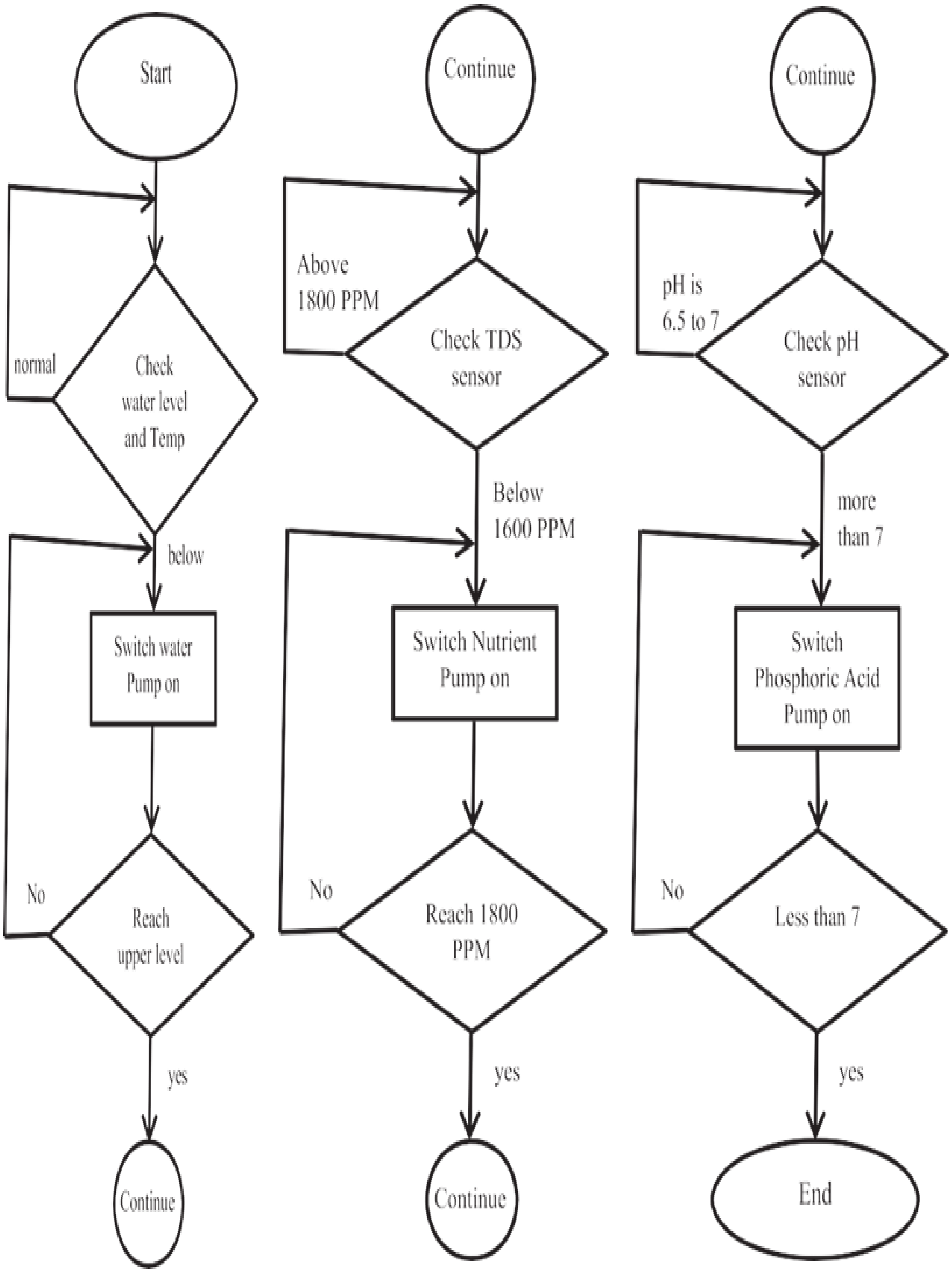


A simplified explanation of how an automatic hydroponics system with a GUI works:

1. **Sensors Collect Data:**
    - Sensors measure key parameters like pH, temperature, humidity, light, and nutrient levels in the water.
2. **Data Sent to Controller:**
    - The sensor data is sent to a microcontroller or computer (e.g., Arduino, Raspberry Pi).
3. **GUI Displays Data:**
    - The data is displayed on a graphical user interface (GUI) that you can view on a computer, tablet, or smartphone.
4. **Monitor Conditions:**
    - The GUI shows real-time readings, allowing you to monitor the growing conditions of your plants.
5. **Automated Adjustments:**
    - Based on the sensor data, the system automatically adjusts the environment by controlling water pumps, nutrient dispensers, grow lights, and ventilation.
6. **User Control via GUI:**
    - Through the GUI, you can manually adjust settings, override automatic controls, and set parameters for automation.
7. **Alerts and Notifications:**
    - The system can send alerts (e.g., via SMS or email) if any parameter goes out of the desired range, so you can take action if needed.
8. **Data Logging:**
    - The system logs historical data, which you can view and analyze through the GUI to optimize plant growth over time.
9. **Maintenance Reminders:**
    - The GUI can provide reminders for system maintenance, such as cleaning filters or refilling nutrient solutions

CHAPTER 5

FLOW CHART


Start
Check water level and Temp
normal
below
Switch water Pump on
Reach upper level
No
yes
Continue
Continue
Check TDS sensor
Above 1800 PPM
Below 1600 PPM
Switch Nutrient Pump on
Reach 1800 PPM
No
yes
Continue
Continue
Check pH sensor
pH is 6.5 to 7
more than 7
Switch Phosphoric Acid Pump on
Less than 7
No
yes
End

CHAPTER 6

SOFTWARE DESCRIPTION

1.1 FRONTEND TECHNOLOGIES

- **HTML (Hypertext Markup Language):** HTML is the standard markup language for creating web pages and web applications. It provides the structure for web content by using a system of tags and attributes. Most existing systems heavily rely on HTML for structuring their frontend content.
- **JavaScript:** JavaScript is a versatile programming language commonly used for creating interactive effects within web browsers. It enables developers to manipulate webpage content, respond to user actions, and dynamically update the page without requiring a full page reload. JavaScript frameworks like React, Vue.js, and Angular are widely used to streamline frontend development.
- **CSS (Cascading Style Sheets):** CSS is used for styling HTML elements, defining layout structures, and enhancing the visual presentation of web pages. CSS frameworks like Bootstrap, Foundation, and Materialize CSS offer pre-designed styles and components to expedite frontend development and ensure consistency across different browsers and devices.

  Evaluation of User Interfaces:

  The responsiveness and intuitiveness of user interfaces (UIs) in existing systems can vary depending on factors such as design principles, frontend technologies used, and the level of user experience (UX) considerations. Here are some criteria to evaluate UIs:
- **Responsiveness**: A responsive UI adjusts seamlessly to different screen sizes and orientations, providing an optimal viewing and interaction experience across devices ranging from desktops to smartphones. This is typically achieved through CSS media queries and flexible layout techniques.
- **Intuitiveness:** An intuitive UI is easy to understand and navigate without requiring extensive user guidance or training. It involves clear visual hierarchy, consistent design patterns, and intuitive user interactions. Techniques such as user testing, feedback collection, and iterative design are often employed to enhance UI intuitiveness.

2. REAL-TIME UPDATES:

1. **Sensors**:
   - **pH Sensor**: Measures the acidity or alkalinity of the nutrient solution.
   - **EC (Electrical Conductivity) Sensor**: Measures the nutrient concentration.
   - **Temperature Sensor**: Monitors the temperature of the nutrient solution and the growing environment.
   - **Humidity Sensor**: Monitors the relative humidity of the growing environment.
   - **Water Level Sensor**: Ensures the water levels are adequate.
   - **Light Sensor**: Monitors the intensity of light the plants receive.
2. **Actuators**:
   - **Pumps**: For circulating nutrient solution and water.
   - **Lights**: For providing artificial light.
   - **Fans**: For ventilation and temperature control.
   - **Heaters/Coolers**: For maintaining the ideal temperature.
3. **Microcontroller/Computer**:
   - Examples include Arduino, Raspberry Pi, or any other microcontroller/computer that can interface with sensors and actuators.
4. **Communication Modules**:
   - WiFi or Bluetooth modules for wireless communication.
   - GSM module for cellular communication if internet is not available.
5. **Software**:
   - Firmware for the microcontroller to handle sensor readings and actuator controls.
   - Web server or cloud service for real-time monitoring and updates.
   - Mobile or web app for user interface.

3. USER INTERFACE DESIGN

. User interface (UI) design is a critical aspect of creating digital products that are both usable and accessible. Usability refers to how easily and efficiently users can interact with a system, while accessibility ensures that the system is usable by individuals with diverse abilities and disabilities.

3.1 Usability

Explore the user interface design principles applied into existing systems to ensure a positive user experience.Responsive Design:Description: Ensuring that the UI adapts seamlessly to different screen sizes and devices.

Example: A website that looks and functions well on both desktop computers and mobile devices.

3.2 Accessibility

Making an automatic hydroponics system accessible involves designing it so that it is easy to use and understand by a wide range of users, including those with disabilities. Here are some ways to enhance accessibility:

1. User-Friendly Interface

- **Simplified Dashboard**: Use a clean and intuitive user interface for the dashboard, ensuring that key information is easily visible.
- **Responsive Design**: Ensure the dashboard is responsive and works well on various devices, including smartphones, tablets, and desktops.

2. Voice Control and Feedback

- **Voice Commands**: Integrate voice control using virtual assistants like Amazon Alexa, Google Assistant, or Siri. Users can give commands such as "Check pH level" or "Turn on the lights".

- **Text-to-Speech**: Implement text-to-speech for alerts and notifications to assist users with visual impairments.

3. Mobile App Accessibility

- **Screen Reader Compatibility**: Ensure the app is compatible with screen readers like VoiceOver (iOS) and TalkBack (Android).
- **Adjustable Text Size and Color Contrast**: Provide options to adjust text size and color contrast for better readability.
- **Gesture Controls**: Support gesture controls for users with mobility impairments.

4. Physical Accessibility

- **Ergonomic Design**: Ensure that the physical setup of the hydroponics system is at a comfortable height and reach for users, including those using wheelchairs.
- **Easy-to-Operate Components**: Use components that are easy to handle and operate, such as lever-operated valves and large, easy-to-press buttons.

5. Remote Monitoring and Control

- **Cloud-Based Monitoring**: Allow users to monitor and control the system remotely through a cloud-based platform. This helps users who may have difficulty physically accessing the system.
- **Real-Time Notifications**: Provide real-time notifications via email, SMS, or app alerts for important updates like low water levels or nutrient deficiencies.

6. Educational Resources

- **Clear Instructions**: Provide clear, step-by-step instructions and tutorials on how to use the system. Use simple language and avoid technical jargon.
- **Video Tutorials**: Create video tutorials with captions and descriptive audio to cater to different learning preferences.
- **User Manuals**: Offer accessible user manuals in various formats, including braille, large print, and digital formats that support screen readers.

7. Support and Community

- **Customer Support**: Provide robust customer support with multiple contact options such as phone, email, and live chat.

- **Community Forums**: Establish online forums or social media groups where users can share tips, ask questions, and support each other.

CHAPTER 7

CODING AND IMPLEMENTATIONS

7.1 IMPLEMENTATION:

Step 1: Define the Project Requirements

- Objective: get inputs from various sensors in real-time and show on website
- Hardware: water level sensor, temperature sensor, humidity sensor, PH level sensor, TDS sensor, pumps, battery, relay, etc.
- Software: graphical user interface, HTML,CSS,Javascript.
- Dataset: various inputs from sensors

Step 2: Setup the Environment

- Setup the hydrophonics model using various sensors and models

Step 3: Data Collection and Preprocessing

- Collect data of sensors from various inputs.

Step 4: Integrate the Model with Real-Time Detection

- Setup a sensors and integrate it with the model for real-time detection:

Step 6: Develop a Web Interface

- Create a GUI to serve the detection results

Step 7: Testing and Deployment

- Test the system in various data.
- Deploy the system on the edge devices and set up the server for processing.

Step 8: Monitoring and Maintenance

- Monitor the performance of the system and update the model with new data as required.
- Implement logging to track detections and system performance.

7.2CODING

1. OVERALL CODE

```
#include <WiFi.h>
#include <WebServer.h>
#include <DHT.h>
#define TdsSensorPin 39
#define VREF 3.3        // Analog reference voltage(Volt) of the ADC
#define SCOUNT  30      // Number of sample points for median filtering
#define ARDUINO_RX 22   // ESP32 TX connected to ESP32 RX (GPIO22)
#define ARDUINO_TX 23   // ESP32 RX connected to ESP32 TX (GPIO23)
#define BAUD_RATE 115200

const char* ssid = "A K";
const char* password = "1234567890";

WebServer server(80);

// Define motor control pins
const int MOTOR1_PIN = 12;
const int MOTOR2_PIN = 13;

bool motor1State = false;
bool motor2State = false;

// Define DHT sensor pin
const int DHT_PIN = 2;
#define DHTTYPE DHT11
DHT dht(DHT_PIN, DHTTYPE);
```

```
float temperature = 0.0;
int waterLevelPercentage = 0;
float humidity = 0.0;
float lastTdsValue = 0.0; // Global variable to store the last TDS value

const int WATER_LEVEL_PIN = 34;

void setup() {
  Serial.begin(115200);
  Serial.println("Starting ESP32 Motor Control...");

  connectToWiFi();

  pinMode(MOTOR1_PIN, OUTPUT);
  digitalWrite(MOTOR1_PIN, LOW);

  pinMode(MOTOR2_PIN, OUTPUT);
  digitalWrite(MOTOR2_PIN, LOW);

  dht.begin();

  server.on("/", HTTP_GET, handleRoot);
  server.on("/toggleMotor1", HTTP_GET, toggleMotor1);
  server.on("/toggleMotor2", HTTP_GET, toggleMotor2);

  server.begin();
```

```
  Serial.println("HTTP server started.");


  Serial.begin(BAUD_RATE);    // Start the built-in Serial for debugging

  Serial2.begin(BAUD_RATE, SERIAL_8N1, ARDUINO_RX, ARDUINO_TX);
// Start Serial2 for communication with the other ESP32

}


void loop() {

  server.handleClient();

  readDHTSensor();

  delay(2000); // This delay is already present for pacing the loop execution


  if (Serial2.available() > 0) { // Check if data is available to read from Serial2

    String esp32Data = Serial2.readStringUntil('\n'); // Read data from Serial2
until newline character

    if (esp32Data.length() > 0) { // Check if valid data was received

      Serial.print("Received from ESP32: ");

      Serial.println(esp32Data);


      // Parse the received data to get the TDS value

      float tdsValue = parseTdsValue(esp32Data);

      if (!isnan(tdsValue)) {

        lastTdsValue = tdsValue; // Update the global TDS value

        Serial.print("TDS Value: ");

        Serial.print(tdsValue);

        Serial.println(" ppm");

      }
```

```
    }
  }

  delay(3000); // Correctly placed delay for pacing sensor readings and processing
}

float parseTdsValue(String data) {
  // Example parsing logic to extract TDS value from received data
  // Assuming data format is "TDS: value"
  int index = data.indexOf(':');
  if (index != -1) {
    String valueStr = data.substring(index + 1);  // Get substring after ':'
    valueStr.trim();  // Remove leading/trailing whitespace
    return valueStr.toFloat();  // Convert string to float
  }
  return NAN;  // Return NaN if parsing fails
}

void connectToWiFi() {
  Serial.println("Connecting to WiFi...");
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
```

```
  }
  Serial.println("WiFi connected.");
  Serial.print("IP address: ");
  Serial.println(WiFi.localIP());
}

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

  if (isnan(temperature) || isnan(humidity)) {
    Serial.println("Failed to read from DHT sensor!");
    temperature = 0.0;
    humidity = 0.0;
  } else {
    Serial.print("Temperature: ");
    Serial.print(temperature);
    Serial.print(" °C | Humidity: ");
    Serial.print(humidity);
    Serial.println("%");
  }
}

void handleRoot() {
  String html = "<!DOCTYPE html><html><head><title>ESP32 Sensor
Dashboard</title>";
  html += "<style>";
```

```
html += "body { font-family: Arial, sans-serif; margin: 0; padding: 20px; }";

html += "h1 { text-align: center; }";

html += ".container { max-width: 600px; margin: 0 auto; }";

html += ".motor { display: flex; justify-content: space-between; align-items: center; margin-bottom: 20px; }";

html += ".motor button { padding: 10px 20px; font-size: 16px; border: none; color: white; cursor: pointer; }";

html += ".motor button.on { background-color: green; }";

html += ".motor button.off { background-color: red; }";

html += ".sensor { margin-top: 30px; }";

html += ".progress-bar { height: 30px; margin-bottom: 10px; }";

html += ".temp-bar { width: " + String(map(temperature, 0, 50, 0, 100)) + "%; background-color: " + getTempColor(temperature) + "; }";

html += ".humid-bar { width: " + String(map(humidity, 0, 100, 0, 100)) + "%; background-color: " + getHumidColor(humidity) + "; }";

html += ".water-bar { width: " + String(readWaterLevel()) + "%; background-color: " + getWaterColor(readWaterLevel()) + "; }";

html += ".tds-bar { width: " + String(map(lastTdsValue, 0, 2000, 0, 100)) + "%; background-color: " + getTdsColor(lastTdsValue) + "; }";

html += "</style>";

html += "</head><body>";


html += "<div class='container'>";

html += "<h1>ESP32 Sensor Dashboard</h1>";


html += "<div class='motor' id='motor1'>";

html += "<h2>Motor 1</h2>";

html += "<p>Status: <span id='motor1Status'>" + String(motor1State ? "On" : "Off") + "</span></p>";
```

```
  html += "<button id='toggleMotor1' onclick='toggleMotor(1)'>" +
String(motor1State ? "Turn Off" : "Turn On") + "</button>";

  html += "</div>";


  html += "<div class='motor' id='motor2'>";

  html += "<h2>Motor 2</h2>";

  html += "<p>Status: <span id='motor2Status'>" + String(motor2State ? "On"
: "Off") + "</span></p>";

  html += "<button id='toggleMotor2' onclick='toggleMotor(2)'>" +
String(motor2State ? "Turn Off" : "Turn On") + "</button>";

  html += "</div>";


  html += "<div class='sensor'>";

  html += "<h2>DHT11 Sensor</h2>";

  html += "<p>Temperature: " + String(temperature) + " °C</p>";

  html += "<div class='progress-bar temp-bar'></div>";

  html += "<p>Humidity: " + String(humidity) + "%</p>";

  html += "<div class='progress-bar humid-bar'></div>";

  html += "</div>";


  html += "<div class='sensor'>";

  html += "<h2>Water Level</h2>";

  html += "<p>Percentage: " + String(readWaterLevel()) + " %</p>";

  html += "<div class='progress-bar water-bar'></div>";

  html += "</div>";


  html += "<div class='sensor'>";

  html += "<h2>TDS Sensor</h2>";
```

```
html += "<p>TDS Value: " + String(lastTdsValue) + " ppm</p>";
html += "<div class='progress-bar tds-bar'></div>";
html += "</div>";


html += "</div>";


html += "<script>";
html += "function toggleMotor(motorNum) {";
html += "  var xhr = new XMLHttpRequest();";
html += "  xhr.open('GET', '/toggleMotor' + motorNum, true);";
html += "  xhr.onload = function() {";
html += "    if (xhr.status == 200) {";
html += "      updateMotorState(motorNum);";
html += "    }";
html += "  };";
html += "  xhr.send();";
html += "}";
html += "function updateMotorState(motorNum) {";
html += "  var motorStatus = document.getElementById('motor' + motorNum
+ 'Status');";
html += "  var toggleButton = document.getElementById('toggleMotor' +
motorNum);";
html += "  if (motorStatus.innerHTML === 'On') {";
html += "    motorStatus.innerHTML = 'Off';";
html += "    toggleButton.innerHTML = 'Turn On';";
html += "    toggleButton.className = 'off';";
html += "  } else {";
```

```
  html += "    motorStatus.innerHTML = 'On';";
  html += "    toggleButton.innerHTML = 'Turn Off';";
  html += "    toggleButton.className = 'on';";
  html += "  }";
  html += "}";
  html += "</script>";

  html += "</body></html>";

  server.send(200, "text/html", html);
}

void toggleMotor1() {
  motor1State = !motor1State;
  digitalWrite(MOTOR1_PIN, motor1State ? HIGH : LOW);
  Serial.println("Motor 1 " + String(motor1State ? "On" : "Off"));
  server.sendHeader("Location", "/");
  server.send(303);
}

void toggleMotor2() {
  motor2State = !motor2State;
  digitalWrite(MOTOR2_PIN, motor2State ? HIGH : LOW);
  Serial.println("Motor 2 " + String(motor2State ? "On" : "Off"));
  server.sendHeader("Location", "/");
  server.send(303);
```

```
}

String getTempColor(float temp) {
  if (temp < 20) {
    return "#007bff"; // Blue
  } else if (temp >= 20 && temp < 30) {
    return "#28a745"; // Green
  } else {
    return "#dc3545"; // Red
  }
}

String getHumidColor(float humidity) {
  if (humidity < 40) {
    return "#dc3545"; // Red
  } else if (humidity >= 40 && humidity < 70) {
    return "#28a745"; // Green
  } else {
    return "#007bff"; // Blue
  }
}

int readWaterLevel() {
  int val = analogRead(WATER_LEVEL_PIN);  // Read the analog value from
sensor
  Serial.print("Raw Water Level Value: ");
  Serial.println(val);
```

```
  waterLevelPercentage = map(val, 0, 4095, 0, 100); // Assuming 12-bit ADC resolution

  // Temporary debugging to see what's happening

  Serial.print("Mapped Water Level Percentage: ");

  Serial.println(waterLevelPercentage);


  return waterLevelPercentage;

}



String getWaterColor(int waterLevel) {

  if (waterLevel < 20) {

    return "#dc3545"; // Red

  } else if (waterLevel >= 20 && waterLevel < 70) {

    return "#ffc107"; // Yellow

  } else {

    return "#28a745"; // Green

  }

}


String getTdsColor(float tdsValue) {

  if (tdsValue < 500) {

    return "#28a745"; // Green

  } else if (tdsValue >= 500 && tdsValue < 1500) {

    return "#ffc107"; // Yellow

  } else {

    return "#dc3545"; // Red
```

```
  }
}
```

2.TDS Value Code

```
#define TdsSensorPin 39
#define VREF 3.3         // Analog reference voltage(Volt) of the ADC
#define SCOUNT  30       // Number of sample points for median filtering

void setup() {
  Serial.begin(115200);  // Initialize serial communication
}

void loop() {
  float tdsValue = readTds();

  // Send TDS value over serial to ESP32
  Serial.print("TDS: ");
  Serial.println(tdsValue);

  delay(3000); // Delay for 3 seconds before the next reading
}

float readTds() {
  int analogBuffer[SCOUNT];

  // Read multiple samples from the TDS sensor
```

```
  for (int i = 0; i < SCOUNT; i++) {
    analogBuffer[i] = analogRead(TdsSensorPin);
    delay(10);
  }

  // Sort the readings in ascending order
  for (int i = 0; i < SCOUNT - 1; i++) {
    for (int j = 0; j < SCOUNT - i - 1; j++) {
      if (analogBuffer[j] > analogBuffer[j + 1]) {
        int temp = analogBuffer[j];
        analogBuffer[j] = analogBuffer[j + 1];
        analogBuffer[j + 1] = temp;
      }
    }
  }

  // Get the median value
  int medianValue = analogBuffer[SCOUNT / 2];

  // Convert analog reading to voltage (0-3.3V)
  float voltage = (float)medianValue * VREF / 4095.0;

  // Calculate TDS value (example formula, adjust based on your sensor's
calibration)
  float tdsValue = voltage * 1000; // Example calibration factor

  return tdsValue;
```

}

CHAPTER 8

DATABASE AND RESULT

1. Enhanced Monitoring and Control

The use of a GUI significantly improves the ability to monitor and control the hydroponics system. Users can view real-time data on key parameters such as pH levels, nutrient concentration, temperature, and humidity. This real-time monitoring allows for immediate adjustments, ensuring that the plants are always in optimal growing conditions. The GUI simplifies complex data into easy-to-understand visuals like graphs and charts, making it accessible even for users with minimal technical knowledge.

2. Increased Efficiency and Productivity

Automatic hydroponics systems equipped with a GUI lead to increased efficiency and productivity. Automated systems manage the nutrient delivery, watering schedules, and environmental controls without the need for constant manual intervention. The GUI facilitates this automation by providing a centralized platform to set and adjust these parameters. As a result, plants receive consistent care, which often leads to faster growth rates and higher yields compared to traditional methods.

3. User-Friendly Experience

A well-designed GUI makes the hydroponics system user-friendly, appealing to both beginners and experienced gardeners. The interface typically includes intuitive controls and straightforward navigation, allowing users to easily access and modify system settings. This ease of use encourages more people to adopt hydroponic gardening, as it lowers the barrier to entry and reduces the intimidation factor associated with managing a sophisticated growing system.

4. Remote Access and Convenience

One of the significant advantages of integrating a GUI is the ability to access the hydroponics system remotely. Many modern GUIs are web-based or have associated mobile applications, enabling users to monitor and control their system from anywhere with an internet connection. This remote access adds a layer of convenience, particularly for users who travel frequently or manage multiple growing sites. Alerts and notifications about critical conditions can be sent directly to users' devices, ensuring timely intervention when necessary.

5. Data Logging and Analysis

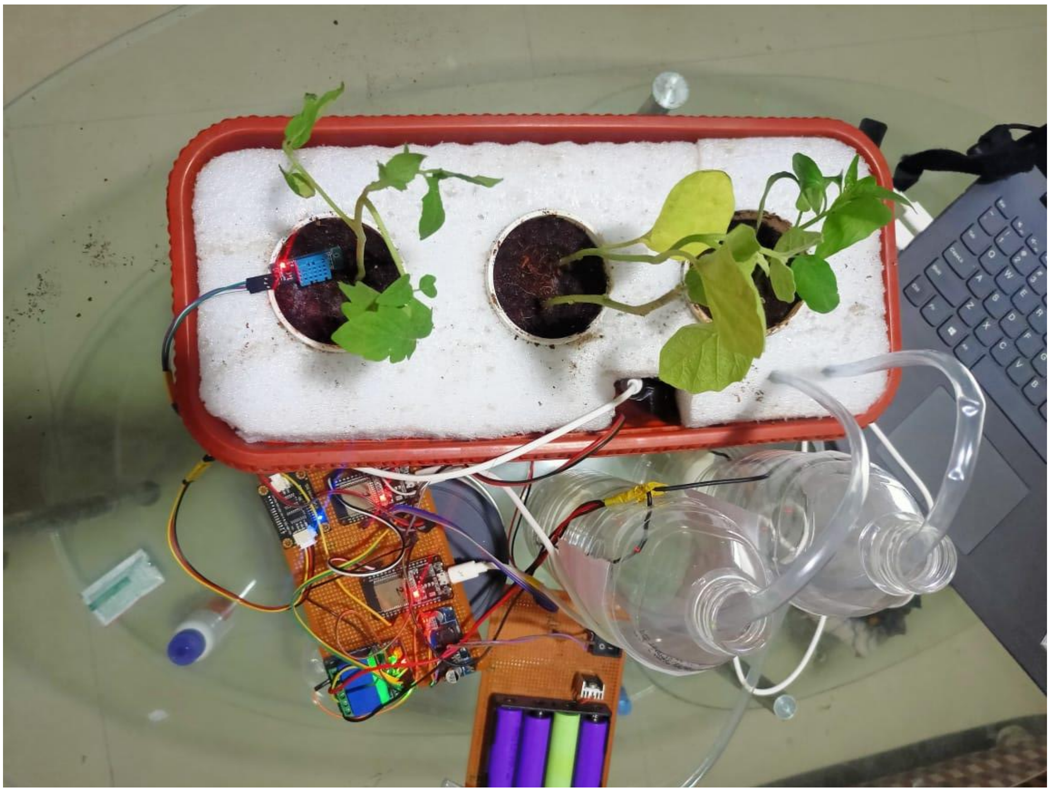
The GUI often includes features for data logging and analysis, which are crucial for optimizing plant growth. Historical data can be stored and analyzed to identify trends, assess the effectiveness of different growing strategies, and make informed decisions. This data-driven approach helps in fine-tuning the system for better performance and can lead to improved plant health and productivity over time.

6. Customization and Scalability

Automatic hydroponics systems with GUIs offer a high degree of customization and scalability. Users can tailor the system settings to suit specific plant types or growth stages, ensuring that each plant receives the care it needs. Additionally, as users gain experience and possibly expand their operations, the system can be scaled up or down with relative ease. The GUI simplifies the process of adding new components or adjusting existing ones, making the system adaptable to changing needs.

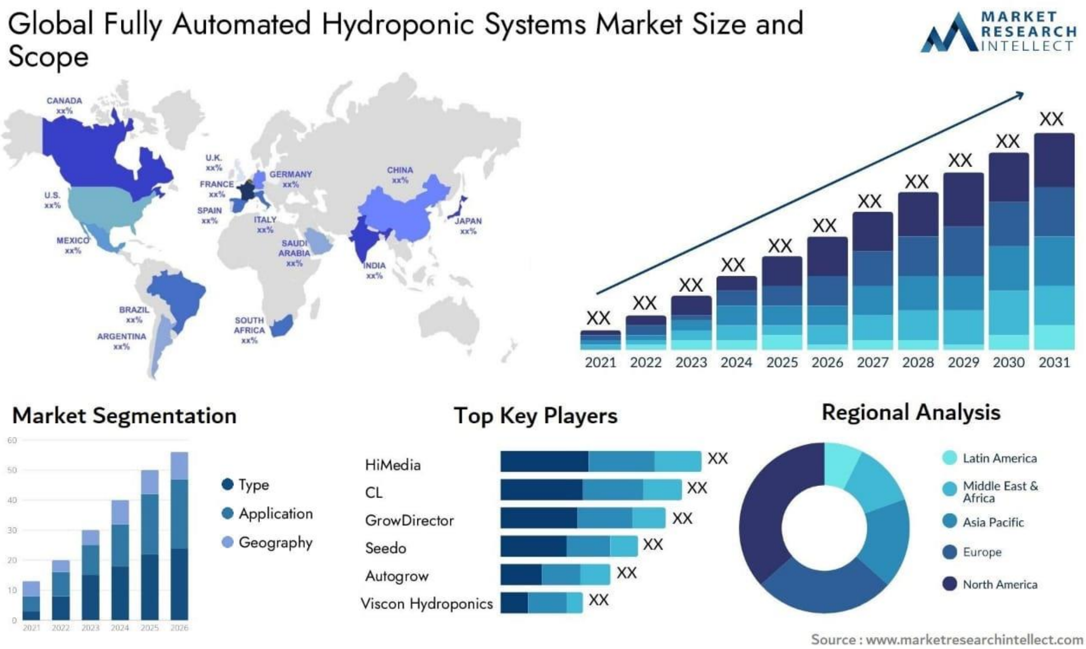
7. Environmental and Resource Management

The integration of a GUI in hydroponics systems also contributes to better environmental and resource management. Automated controls ensure efficient use of water and nutrients, minimizing waste. The ability to monitor and adjust environmental conditions helps in reducing energy consumption, especially in systems that use artificial lighting and climate control. This sustainable approach not only benefits the environment but also reduces operational costs over time.



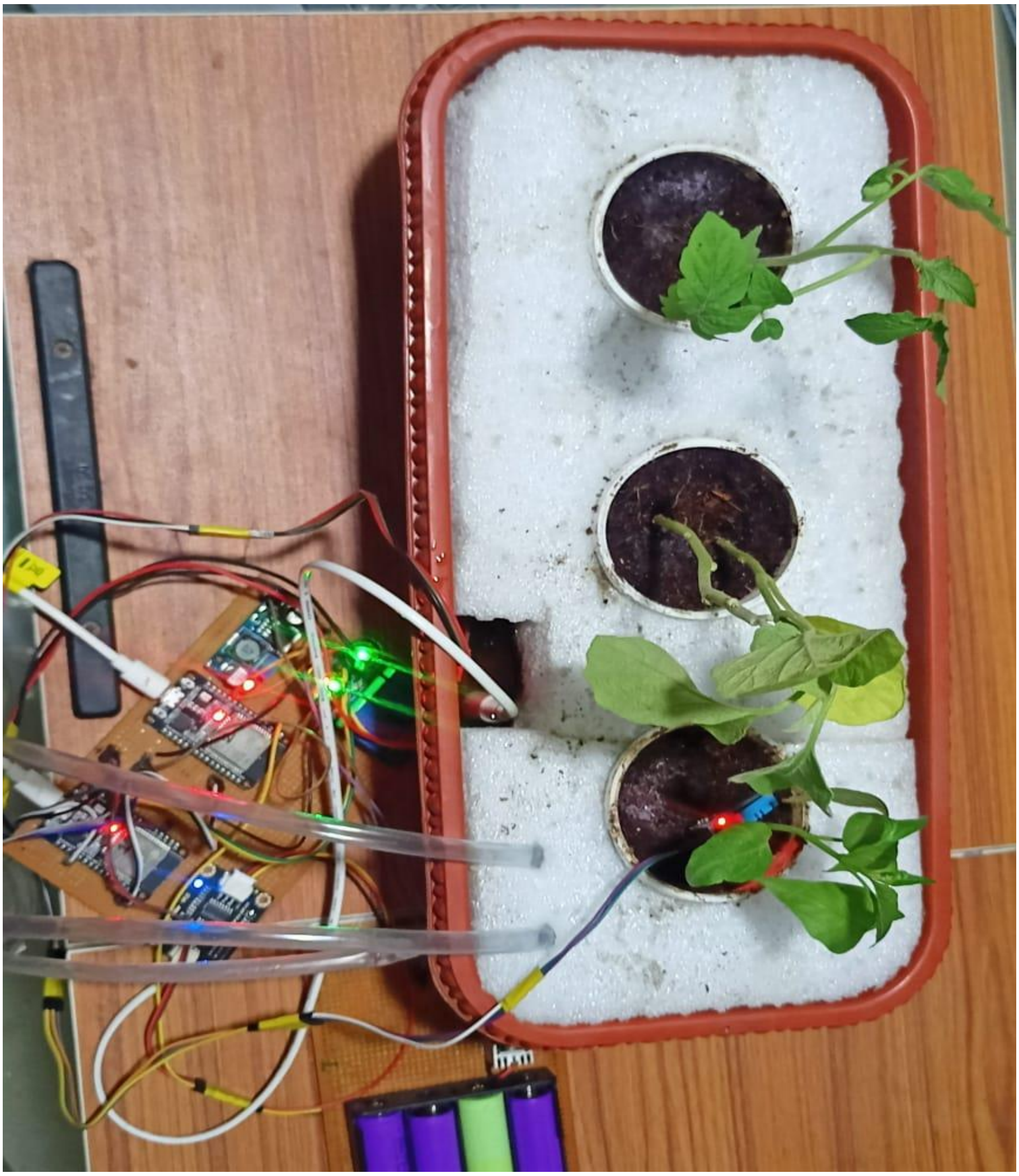
Statistics:

Nutrient (chemical symbol)	Approximate content of plant (% dry weight)	Roles in plant	Source of nutrient available to plant
Carbon (C), hydrogen (H), oxygen (O)	90+%	Components of organic compounds	Carbon dioxide (CO_2) and water (H_2O)
Nitrogen (N)	2 to 4%	Component of amino acids, proteins, coenzymes, nucleic acids	Nitrate (NO_3^-) and ammonium (NH_4^+)
Sulfur (S)	0.50%	Component of sulfur amino acids, proteins, coenzyme A	Sulfate (SO_4^-)
Phosphorus (P)	0.40%	ATP, NADP intermediates of metabolism, membrane phospholipids, nucleic acids	Dihydrogen phosphate ($H_2PO_4^-$), Hydrogen phosphate (HPO_4^{2-})
Potassium (K)	2.00%	Enzyme activation, turgor, osmotic regulation	Potassium (K^+)
Calcium (Ca)	1.50%	Enzyme activation, signal transduction, cell structure	Calcium (Ca^{2+})
Magnesium (Mg)	0.40%	Enzyme activation, component of chlorophyll	Magnesium (Mg^{2+})
Manganese (Mn)	0.02%	Enzyme activation, essential for water splitting	Manganese (Mn^{2+})
Iron (Fe)	0.02%	Redox changes, photosynthesis, respiration	Iron (Fe^{2+})
Molybdenum (Mo)	0.00%	Redox changes, nitrate reduction	Molybdate ($MoO4^{2-}$)
Copper (Cu)	0.00%	Redox changes, photosynthesis, respiration	Copper (Cu^{2+})
Zinc (Zn)	0.00%	Enzyme cofactor-activator	Zinc (Zn^{2+})
Boron (Bo)	0.01%	Membrane activity, cell division	Borate (BO^{3-})
Chlorine (Cl)	0.1 to 2.0%	Charge balance, water splitting	Chlorine (Cl^-)
Nickel (Ni)	0.000005 to 0.0005%	Component of some enzymes, biological nitrogen fixation, nitrogen metabolism	Nickel (Ni^{2+})

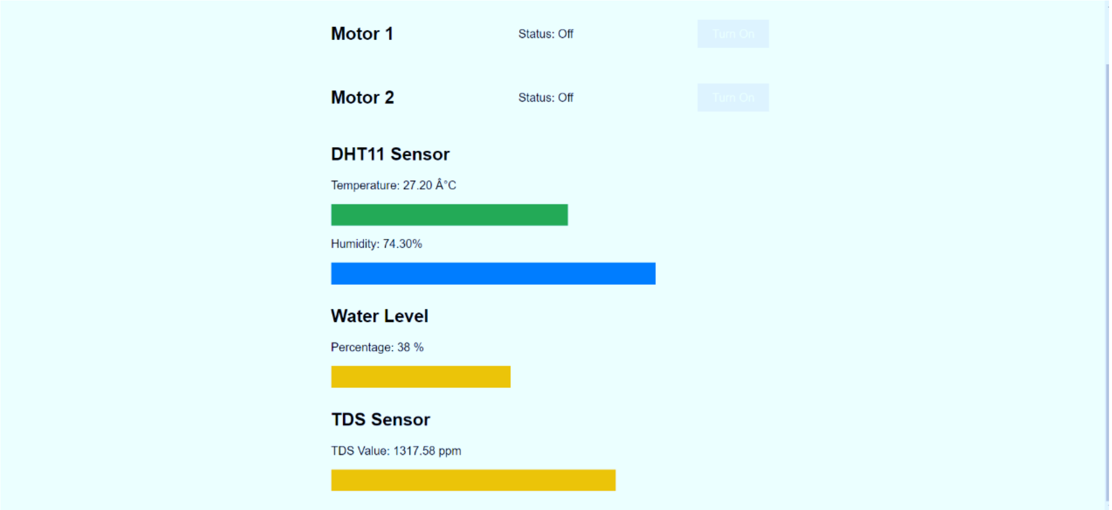

Global Fully Automated Hydroponic Systems Market Size and Scope
MARKET RESEARCH INTELLECT
CANADA xx%
U.S. xx%
MEXICO xx%
BRAZIL xx%
ARGENTINA xx%
U.K. xx%
FRANCE xx%
SPAIN xx%
GERMANY xx%
ITALY xx%
SAUDI ARABIA xx%
SOUTH AFRICA xx%
CHINA xx%
INDIA xx%
JAPAN xx%
2021 2022 2023 2024 2025 2026 2027 2028 2029 2030 2031
Market Segmentation
Type
Application
Geography
Top Key Players
HiMedia
CL
GrowDirector
Seedo
Autogrow
Viscon Hydroponics
Regional Analysis
Latin America
Middle East & Africa
Asia Pacific
Europe
North America
Source : www.marketresearchintellect.com

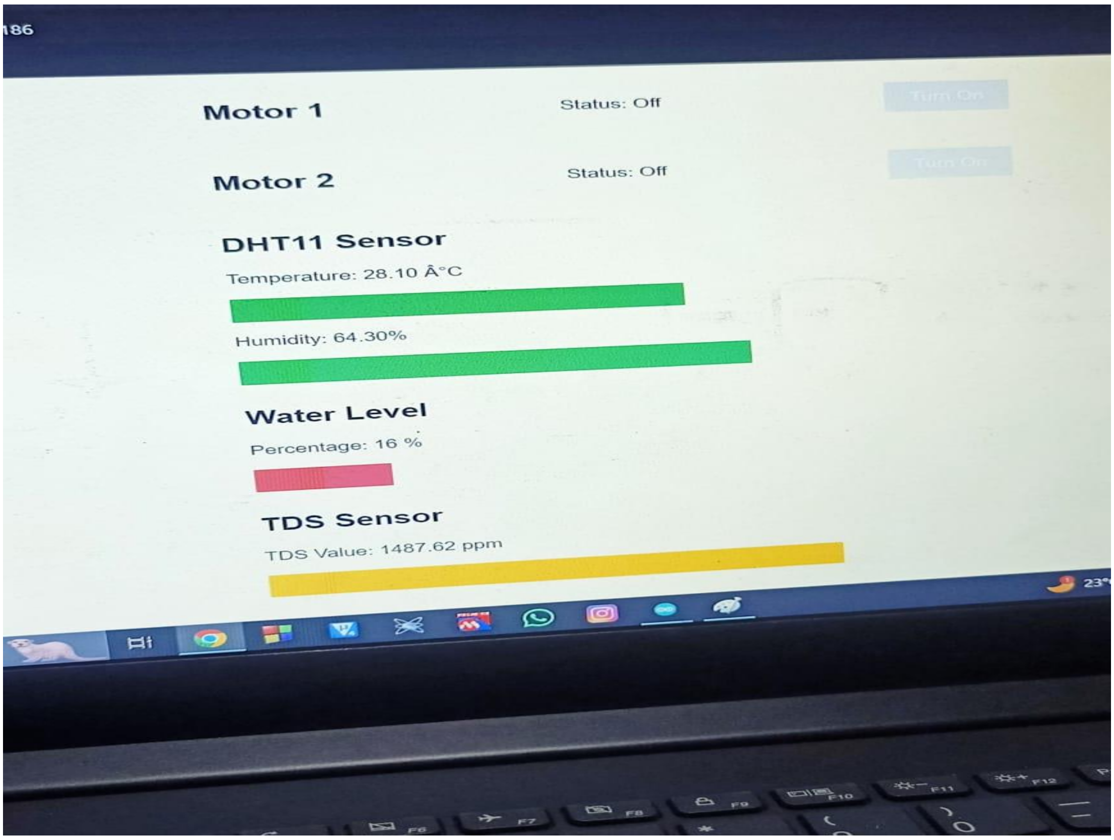
FINAL OUTPUT:

1)For Input data:



2)Result:


Motor 1
Status: Off
Motor 2
Status: Off
DHT11 Sensor
Temperature: 27.20 Â°C
Humidity: 74.30%
Water Level
Percentage: 38 %
TDS Sensor
TDS Value: 1317.58 ppm


Motor 1
Status: Off
Motor 2
Status: Off
DHT11 Sensor
Temperature: 28.10 Â°C
Humidity: 64.30%
Water Level
Percentage: 16 %
TDS Sensor
TDS Value: 1487.62 ppm

CHAPTER 9

ADVANTAGES AND DISADVANTAGES

9.1 ADVANTAGES:

1. Water Efficiency

- Reduced Water Usage: Hydroponics systems use up to 90% less water compared to traditional soil-based gardening. Water is recirculated and reused, minimizing wastage.

2. Space Efficiency

- Vertical Farming: Plants can be grown in stacked layers, making better use of vertical space and allowing more plants to be grown in smaller areas.

3. Faster Growth Rates

- Optimal Conditions: Plants in hydroponic systems often grow faster because they receive optimal nutrients, water, and oxygen directly to their roots.

4. Higher Yields

- Controlled Environment: By precisely controlling the growing environment (light, temperature, humidity, pH, and nutrients), hydroponic systems can produce higher yields compared to traditional methods.

5. Pest and Disease Control

- Soil-Free: Eliminating soil reduces the risk of soil-borne pests and diseases. This can lead to healthier plants and reduced need for pesticides.

6. Year-Round Production

- Indoor Growing: Hydroponic systems can be set up indoors or in greenhouses, allowing for year-round cultivation regardless of external weather conditions.

7. Automation and Ease of Use

- Automated Systems: Automatic hydroponics systems can handle watering, nutrient delivery, and environmental control, reducing the need for constant manual intervention.

- Remote Monitoring: Modern systems often include remote monitoring and control via smartphones or computers, making it easy to manage even when you're not physically present.

8. Nutrient Efficiency

- Precise Nutrient Delivery: Nutrients are delivered directly to the plant roots in precise amounts, ensuring optimal growth and reducing waste.

9.2 DISADVANTAGES

1. High Initial Costs

- Setup Costs: The initial investment for setting up a hydroponic system can be high, including the cost of equipment, sensors, and automation technology.

2. Technical Knowledge Required

- Complexity: Operating an automatic hydroponics system requires a certain level of technical knowledge to manage the equipment and troubleshoot issues.
- Maintenance: Regular maintenance of sensors, pumps, and other components is essential to ensure the system runs smoothly.

3. Dependence on Power

- Electricity Dependency: Hydroponic systems rely on electricity to power pumps, lights, and other automated components. Power outages can disrupt the system and harm the plants.

4. Risk of System Failures

- Component Failures: Failures in key components like pumps, sensors, or automation controls can have a significant impact on plant health if not quickly addressed.
- Monitoring: Continuous monitoring is required to catch and resolve issues promptly.

5. Waterborne Diseases

- Disease Spread: If a waterborne disease or pathogen enters the system, it can quickly spread to all plants since they share the same water source.

6. Nutrient Management

- Nutrient Imbalance: Maintaining the correct balance of nutrients is crucial and can be challenging. An imbalance can lead to poor plant health or growth.
- Regular Adjustments: Nutrient solutions need to be regularly monitored and adjusted.

7. Limited Crop Variety

- Crop Suitability: Not all crops are suitable for hydroponic growing. Root vegetables like potatoes and carrots are difficult to grow in a hydroponic setup.

CHAPTER 10

CONCLUSION

Implementing an automatic hydroponics system with a graphical user interface (GUI) offers a modern, efficient, and user-friendly approach to soilless plant cultivation. The integration of automation and GUI technology simplifies the complex processes involved in hydroponics, making it accessible to a wider range of users. A well-designed GUI provides an intuitive platform for monitoring and controlling various aspects of the system, such as nutrient levels, pH, temperature, and humidity. This ease of use is particularly beneficial for beginners who may lack technical expertise, as it reduces the learning curve and allows for more precise management of the growing environment.

One of the primary advantages of using an automatic hydroponics system with a GUI is the ability to achieve optimal plant growth conditions with minimal manual intervention. Automated sensors and actuators continuously monitor and adjust the system parameters, ensuring that plants receive the right amount of nutrients and water. The GUI serves as a central hub where users can view real-time data, receive alerts, and make adjustments as needed. This not only enhances the efficiency and productivity of the system but also provides peace of mind to users, knowing that their plants are being well cared for, even when they are not physically present.

However, there are challenges associated with implementing such a system. The initial setup cost can be high, and there is a need for ongoing maintenance and troubleshooting to ensure that all components function correctly. Additionally, users must have some level of technical knowledge to effectively operate and maintain the system, despite the user-friendly nature of the GUI. Moreover, reliance on electricity and internet connectivity can pose risks in the event of power outages or connectivity issues, potentially disrupting the automated processes and harming plant health.

Despite these challenges, the benefits of an automatic hydroponics system with a GUI are significant. It offers a sustainable and efficient method of plant cultivation that can be utilized in various settings, from small home gardens to large commercial operations. The ability to control and monitor the system remotely adds a layer of convenience and flexibility, making it an attractive option for busy individuals and businesses looking to optimize their agricultural practices. With the right planning and implementation, an automatic hydroponics system with a GUI can revolutionize the way we grow plants, leading to higher yields, better resource management, and a more sustainable future for agriculture.

CHAPTER 11

REFERENCES

[1] L. Cifuentes-Torres, LG. Mendoza-Espinosa, G. Correa-Reyes, L.W. Daesslé, Hydroponics with wastewater: a review of trends and opportunities, Water Environ. J. 35 (1) (2021) 166–180.

[2] N. Sharma, S. Acharya, K. Kumar, N. Singh, O.P. Chaurasia, Hydroponics as an advanced technique for vegetable production: an overview, J. Soil Water Conserv. 17 (4) (2018) 364–371.

[3] D.J. Singh, J. Davidson, Introduction to Hydroponics-Growing Your Plants Without Any Soil, Mendon Cottage Books, 2016.

[4] A. Abu Sneineh, WA. Salah, Design and implementation of an automatically aligned solar tracking system, Int. J. Power Electr. Drive Syst. 10 (4) (2019) 2055.

[5] A. Abu Sneineh, WA. Salah, Palestine automotive license identity recognition for intelligent parking system, J. Eng. Sci. Technol. 12 (5) (2017) 1216–1226.

[6] DS. Domingues, HW. Takahashi, C.A.P. Camara, SL. Nixdorf, Automated system developed to control pH and concentration of nutrient solution evaluated in hydroponic lettuce production, Comput. Electron. Agric. 84 (2012) 53–61.

[7] A. Ullah, S. Aktar, N. Sutar, R. Kabir, A. Hossain, Cost effective smart hydroponic monitoring and controlling system using IoT, Intell. Control Autom. 10 (04) (2019) 142.

[8] F. Kalantari, O.M. Tahir, A.M. Lahijani, S. Kalantari, A review of vertical farming technology: a guide for implementation of building integrated agriculture in cities, Adv. Eng. Forum 24 (2017) 76–91.

[9] R. Pandey, V. Jain, K.P. Singh, Hydroponics Agriculture: Its status, Scope and Limitations, 20, Division of Plant Physiology, Indian Agricultural Research Institute, New Delhi, 2009.

[10] C. Treftz, ST. Omaye, Hydroponics: potential for augmenting sustainable food production in non-arable regions, Nutr. Food Sci. 46 (5) (2016) 672–684.

[11] Z. Zuriati, M. Apriyani, A.R. Supriyatna, Design and implementation automation system for hydroponic vegetable cultivation, in: Proceedings of the International Conference on Agriculture and Applied Science, 2021.

[12] K. Kularbphettong, U. Ampant, N. Kongrodj, An automated hydroponics system based on mobile application, Int. J. Inf. Educ. Technol. 9 (8) (2019) 548–552.

[13] S. Huo, J. Liu, M. Addy, P. Chen, D. Necas, P. Cheng, K. Li, H. Chai, Y. Liu, R. Ruan, The influence of microalgae on vegetable production and nutrient removal in greenhouse hydroponics, J. Clean. Prod. 243 (2020) 118563.

[14] JCV. Puno, JJI. Haban, JD. Alejandrino, AA. Bandala, EP. Dadios, Design of a nutrient film technique hydroponics system with fuzzy logic control, in: Proceedings of the IEEE Region 10 Conference (Tencon), IEEE, 2020, pp. 403–408.

[15] C. Ramos, L. Nóbrega, K. Baras, L. Gomes, Experimental NFT hydroponics system with lower energy consumption, in: Proceedings of the 5th Experiment International Conference (exp. at’19), IEEE, 2019, pp. 102–106.

[16] Datasheet, Gravity: analog TDS sensor /meter for arduino SKU: SEN0244. (20/6/2022). https://media.digikey.com.

[17] N.D. Susanti, D. Sagita, I.F. Apriyanto, C.E.W. Anggara, D.A. Darmajana, A. Rahayuningtyas, Design and implementation of water quality monitoring

[18] N. Agnihotri, “Arduino-based water temperature monitor using DS18B20 1wire temperature sensor,” (20/11/2022). https://www.engineersgarage.com/ arduinobased-water-temperature-monitor-using-ds18b20-1-wire-temperature-sensor/

[19] A. Al-Yemni, S. Al-Balam, S. Al-Kulib, Q.A. Al-Haija, An Arduino based smart faucet design, Int. J. Adv. Comput. Technol. 7 (5) (2018) 2752–2754.

[20] H.N. Ang, M.W. Lim, W.S. Chua, Design of a water quality monitoring system utilizing IoT platform for hydroponics application, in: Proceedings of the AIP Conference, 2610, AIP Publishing LLC, 2022.

[21] H. Singh, B. Dunn, M. Payton, Hydroponic pH modifiers affect plant growth and nutrient content in leafy greens, J. Hortic. Res. 27 (1) (2019) 31–36.

[22] P. Karuppusamy, A sensor based IoT monitoring system for electrical devices using Blynk framework, J. Electr. Inform. 2 (3) (2020) 182–187.

[23] S. Pramono, A. Nuruddin, M.H. Ibrahim, Design of a hydroponic monitoring system with deep flow technique (DFT), in: Proceedings of the AIP Conference, 2217, AIP Publishing, 2020.