Shahid Bahonar University of Kerman

Biomechanism and Bioenergy Research

Online ISSN: 2821-1855
Homepage: https://bbr.uk.ac.ir

Iranian Society of Agricultural Machinery Engineering and Mechanization

Design of a Soft Real-Time Web System for Monitoring Greenhouse Climate Using Arduino Platform and ESP32

Melika Yousefi sabet[1], Akbar Arabhosseini[1], Abolfazl Aghajanlou[1]

[1] Department of Agrotechnology, Faculty of Aburaihan, University of Tehran, Tehran, Iran

Corresponding author: Ahosseini@ut.ac.ir

ARTICLE INFO

Article type:

Research Article

Article history:

Received 21 February 2026

Received in revised form 14 March 2026

Accepted 01 May 2026

Available Online 30 June 2026

Keywords:

Arduino, ESP32, Greenhouse, IoT, Monitoring, Web Interface

ABSTRACT

Greenhouse crops are highly sensitive to fluctuations in environmental parameters; their survival and productivity depend heavily on maintaining optimal conditions. With the global expansion of greenhouse cultivation, including in Iran, intelligent climate monitoring systems have become essential. This project designs a smart system for monitoring greenhouse climates using Internet of Things (IoT) technology. Three key parameters—temperature, relative air humidity, and light intensity—are measured in real time and displayed on a web-based user interface. The microcontroller connects to the internet via Wi-Fi and transmits data collected by the DHT11 and GY-30 sensors to a web server using the HTTP protocol, enabling remote monitoring. The system utilizes cost-effective hardware, specifically the ESP32 board, and software tools like the Arduino IDE for data processing and communication. Key features include dynamic color changes of graphical gauges based on parameter status, automatic data updates every minute, a soft real-time display of the last update time, and visual alert mechanisms for critical conditions. Comparisons with related studies show that this system delivers reliable performance and is suitable for various greenhouse applications while maintaining simplicity and low cost.

Cite this article: Yousefi sabet, M., Arabhosseini, A., & Aghajanlou, A. (2026). Design of a Soft Real-Time Web System for Monitoring Greenhouse Climate Using Arduino Platform and ESP32. *Biomechanism and Bioenergy Research*, *5*(2), 31-45. https://doi.org/10.22103/bbr.2026.27194.1150

Publisher: Shahid Bahonar University of Kerman

DOI: https://doi.org/10.22103/bbr.2026.27194.1150

INTRODUCTION

The rapid evolution of communication technologies has revolutionized many fields, including agriculture. An IoT-based smart platform can monitor greenhouse climates to meet the growing demand for food driven by global population growth. Greenhouse monitoring and automation are crucial research topics in precision agriculture. Regulating environmental conditions is vital, as improper temperature and relative air humidity can cause large-scale crop damage.

Given the negative effects of global warming and climate change on food security, the role of technology in future agriculture is paramount. Within the IoT framework, these technologies mitigate climate risks by optimizing critical parameters and enabling intelligent data collection. These challenges can be partially addressed through IoT-driven greenhouse optimization. Over time, advancements have yielded sensors capable of recording environmental data accurately and promptly, facilitating frost prevention, remote crop monitoring, risk mitigation, and operational cost reduction (Sami et al., 2026).

The IoT in agriculture serves as a comprehensive platform for crop management, particularly in greenhouses, by integrating protected cultivation practices, environmental sensors, microcontrollers for data acquisition, servers for data processing, and cloud platforms for storage. This study aims to design and develop a web-based system for monitoring microclimatic conditions inside a greenhouse.

Motivation and Research Significance

There is limited evidence of technology adoption and precision agriculture in developing countries. Embracing emerging innovations in these regions can significantly improve productivity and address food security gaps between developed and developing nations.

Despite the widespread adoption of IoT systems in smart greenhouses, there remains an insufficient understanding of how these technologies can effectively optimize greenhouse environments. Most existing research has been localized to developed countries, where direct access to IoT systems and resources is readily available. The web-based system that accommodates the diverse needs of different plant species and provides a centralized, intuitive interface can significantly simplify greenhouse management and enhance the practicality of soft real-time supervision.

Research Objectives

A key goal is to minimize manual operations, enhance productivity, and optimize chemical usage. To improve yield and production quality, the aim of this study is to design and implement a system for remotely monitoring microclimatic conditions inside a greenhouse via a web-based application. The core idea is to facilitate the management of influential environmental factors within a controlled setting by designing an intuitive and accessible web interface for continuous, real-time observation.

Literature Review

In recent years, various intelligent systems have been developed for greenhouse climate monitoring and control. Massah et al. designed a system with measurement and control stations for air humidity, light intensity, and temperature, evaluating its impact on rose quality and energy consumption (Massah et al., 2023). Rishe et al. used neural networks to predict soil moisture based on sensors at multiple depths, which required high computational resources (Risheh et al., 2020). Alavian et al. proposed a fuzzy-controlled smart greenhouse for managing temperature, light intensity, soil moisture, and air humidity (Alaviyan et al., 2020).

Other researchers, such as Gaikwad et al. used Arduino and ZigBee to transmit time-controlled data to an Android application, while Hammami developed a web-based system using Arduino and Raspberry Pi to store and display real-time temperature and air humidity data (Gaikwad et al., 2016; Hammami, 2019). Tembhurne et al. presented a low-cost Arduino solution with DHT11, soil moisture, gas, and light intensity sensors, displaying results on an LCD panel (Tembhurne et al., 2022). Hong and Hsieh employed Bluetooth for wireless irrigation with a 100-meter range (Hong & Hsieh, 2016). Taru and Karwankar and Osman et al. introduced water quality monitoring systems using temperature, pH, turbidity, and EC sensors, along with alert mechanisms like LEDs and buzzers (Osman et al., 2018; Taru & Karwankar, 2017). Kumar et al. developed a solar-powered GSM-based monitoring system with Arduino for real-time data transmission to mobile devices (Kumar et al., 2022). Arathi et al. developed an IoT- and Arduino-based system utilizing four sensors (temperature, humidity, soil moisture, and light) to monitor and control greenhouse climate conditions (Arathi et al., 2023).

During the COVID-19 pandemic, social restrictions and workforce limitations created significant challenges for agricultural production and food security. Tejahjono et al. investigated the application of precision agriculture in a hydroponic greenhouse using design thinking methods and sensor-based monitoring to ensure effective process control. The study, conducted at the Puriponic greenhouse facility, proposed a hydroponic greenhouse monitoring and control system based on an Arduino microcontroller integrated with six sensors. A notable feature of this system was the actuator-driven response associated with each sensor, enabling automated management of greenhouse conditions (Tjahjono et al., 2023).

Pradeep et al. proposed an IoT-enabled remote greenhouse monitoring system to supervise both indoor and outdoor environmental conditions by integrating multiple sensors for temperature, relative humidity, and light intensity. The study demonstrated the potential of combining IoT technologies and intelligent algorithms for environmental supervision. (Pradeep et al., 2024). A recent study by Asane et al. introduced a Smart Greenhouse Monitoring System based on IoT technology, employing a Raspberry Pi as the main controller integrated with DHT11, soil moisture, and LDR sensors, along with actuators such as a water pump, fan, and bulb. The system automatically adjusted irrigation, temperature, and lighting using real-time data while transmitting sensor readings to the ThingSpeak cloud for remote monitoring (Asane et al., 2025).

In recent decades, intelligent systems have increasingly adopted short-range wireless technologies like Bluetooth and ZigBee. However, Wi-Fi (especially with internet access) offers broader coverage, making it more suitable for remote monitoring; hence, it was chosen for this study.

The novelty of the proposed system does not lie in the individual hardware components, such as the ESP32 or Wi-Fi communication, which have been adopted in previous studies. Rather, the contribution of this work is the integration of low-cost IoT hardware, a browser-based monitoring platform, dynamic graphical gauges, visual alert mechanisms, and automatic soft real-time updates into a unified system specifically designed for greenhouse climate monitoring. Unlike many existing solutions that rely on local displays, mobile applications, SMS notifications, or third-party cloud dashboards, the proposed platform provides direct access through a standard web browser without requiring dedicated software installation, configuration, or periodic updates, thereby improving accessibility, usability, and long-term maintainability for greenhouse operators.

MATERIALS AND METHODS

Experimental Site

The proposed system was implemented and tested in a hydroponic plant factory located at the University of Tehran, Campus of Aburaihan. The room was completely isolated, and only artificial lighting was used for cultivation (no natural light). The cultivated species was *Ocimum basilicum* var. Genovese (Italian basil), grown hydroponically in a vertical farming system with a total growing area of 200 m² (5 m wide by 40 m long). Figure 1 shows the hydroponic plant factory environment used for system evaluation.

Figure 1. The hydroponic plant factory was used as the experimental site for system implementation and testing

The lighting system consisted of specialized plant growth lights combining red (660nm) and blue (440 nm) spectrum LEDs. This spectral combination is optimized for photosynthesis and is ideal for basil growth. All climate parameters, including temperature, relative air humidity, and light intensity, were maintained as stably as possible throughout the day and night. This controlled environment ensured the reliability and reproducibility of data collection.

Monitored Parameters

In consultation with greenhouse experts and growers, the environmental variables with the most direct impact on plant performance were determined. Among the factors affecting greenhouse productivity, temperature, relative air humidity, and light intensity were selected for continuous monitoring.

Hardware Tools

DHT11 sensor

The DHT11 digital sensor was used to measure air temperature and relative humidity. The DHT11 was selected primarily because of its low cost, low power consumption, ease of integration, and adequate performance for greenhouse monitoring applications. Although more advanced sensors such as the DHT22 and SHT31 provide higher accuracy and wider measurement ranges, they also increase the overall system cost. Since the objective of this study was to develop a low-cost, accessible monitoring platform, the DHT11 was considered a suitable compromise between performance and affordability. Nevertheless, the measurement limitations of the DHT11 should be considered in applications requiring high-precision environmental control. The specifications of DHT11 are presented in Table 1.

Table 1. Specifications of the DHT11 Sensor

Temperature Measurement Range	**0 – 50 °C**
Temperature Accuracy	**±2 °C**
Air Humidity Measurement Range	**20% – 90% RH**
Air Humidity Accuracy	**±5% RH**
Operating Voltage	**3.3 – 5 V**
Current Consumption	**< 2.5 mA during measurement**
Communication Protocol	**Single-wire**

GY-30 Light intensity sensor

The GY-30 light intensity sensor (also known as BH1750) was used to measure ambient light intensity levels inside the greenhouse. The GY-30 is a highly accurate sensor widely used in environmental monitoring applications. The specifications of the GY-30 are given in Table 2.

Table 2. Specifications of the GY-30 Light Intensity Sensor

Light Intensity Range	**1 – 65,535 lux**
Operating Voltage	**3.3 – 5 V**

Microcontroller board

An ESP32 development board served as the main control unit for implementing commands and interfacing with sensors. The ESP32 is a powerful microcontroller with integrated Wi-Fi and Bluetooth (BLE/Smart Bluetooth) capabilities, making it highly suitable for IoT applications. Due to its low cost, the ESP32 board is accessible and practical for technicians with limited IT experience. It offers over 30 I/O pins, supports Analog-to-Digital Conversion (ADC) for reading sensor data, and PWM for controlling components such as lights or fans. It also supports low-power modes (such as Deep Sleep) for energy efficiency.

Software Tools

Arduino IDE[1]

The Arduino IDE is an open-source development environment used to program and upload code to Arduino-compatible boards such as the ESP32. It is lightweight, user-friendly, and compatible with Linux, Windows, and macOS.

XAMPP[2] Package

XAMPP is a free, open-source software package providing a complete web development environment. It includes the Apache web server, a MySQL database, and interpreters for PHP and Perl. In this project, XAMPP provided the local development environment (localhost). The designed web application utilizes a MySQL database for storing and managing sensor data, while Apache executes server-side code to transmit data to the web interface.

Notepad++

Notepad++ is a lightweight text editor used for writing and editing the source code of the web interface, offering features such as syntax highlighting and auto-completion.

Libraries

The libraries listed in Table 3 were incorporated into the Arduino software.

Table 3. Libraries used in the Arduino environment

ArduinoHttpClient	**Sends and receives data over HTTP/HTTPS protocols**
WiFi.h	Enables Wi-Fi connectivity for the ESP32 board
DHT.h	Reads and processes data from DHT series sensors (e.g., DHT11)
BH1750.h	Reads and processes light intensity data from the GY-30 sensor
WiFiUdp.h	Supports UDP communication used in combination with NTPClient
NTPClient.h	synchronizes system time using NTP servers for timestamps
Wire.h	Facilitates I2C communication between the microcontroller and peripherals

Programming Language

The primary programming language used in the Arduino IDE is a customized version of **C++**, which includes pre-built functions and classes to simplify interaction with hardware components, pins, sensors, and serial communication. C++ is one of the most powerful and widely used languages, especially in embedded systems such as microcontroller-based platforms.

Sensor Wiring

Sensor wiring establishes the electrical connections linking sensors to control units, data acquisition systems, or actuators (Massah et al., 2026). It involves the physical connection of a sensor's output signal, power supply, and ground. These connections enable the sensor to detect environmental parameters (like temperature, relative air humidity, light intensity, etc.) and transmit this information as electrical signals for processing.

DHT11 Sensor wiring

- Output type: Digital
- Wiring:

VCC → 3.3V pin of ESP32
GND → GND pin of ESP32
DATA → Digital GPIO pin of ESP32

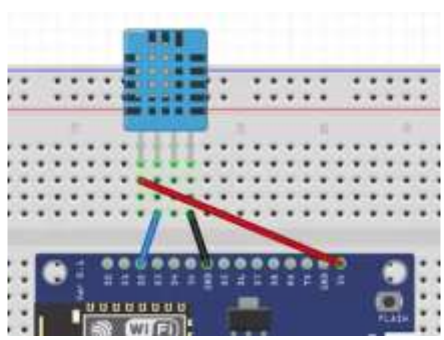

Figure 2. DHT11 Sensor wiring

GY-30 (BH1750) Sensor wiring

[1] Integrated Development Environment

[2] Cross-platform, Apache, MySQL, PHP & Perl

- Communication type: I²C (supports shared SDA and SCL lines)
- Wiring:

VCC → 3.3V pin of ESP32
GND → GND pin of ESP32
SDA → GPIO D12 (default SDA on ESP32)
SCL → GPIO D22 (default SCL on ESP32)

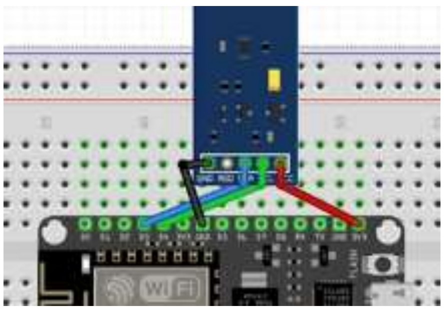

Figure 3. GY-30 (BH1750) Sensor wiring

Data Acquisition and Transmission to the Server

In IoT systems, data measured by sensors is transmitted wirelessly to a central server. Common wireless communication methods include Bluetooth, ZigBee, and Wi-Fi. While Bluetooth and ZigBee support short distances (10–100 meters), Wi-Fi offers broader coverage and eliminates the need for wired connections, especially when connected to the internet. Hence, Wi-Fi was used in this project for data transmission.

The proposed system utilizes a multi-node IoT architecture to ensure reliable data acquisition and transmission, as illustrated in the system workflow (Figure 4). Inside the greenhouse, environmental data collected by the DHT11 and GY-30 sensors are first processed by an ESP32 Client node. To optimize power consumption and reduce dependence on local Wi-Fi infrastructure within the cultivation area, the ESP32 Client transmits the sensor readings to an ESP32 Master node via the low-power ESP-NOW wireless communication protocol.

The ESP32 Master node acts as the central gateway. It receives the data packets from the client node, synchronizes the timestamp using the Network Time Protocol (NTP), and establishes a Wi-Fi connection to transmit the collected data to the central web server via HTTP. The Wi-Fi credentials (SSID and password) are pre-configured within the master node's software. The WiFi.h library handles the network connection, while HTTPClient.h manages the HTTP requests. Data is updated at one-minute intervals and displayed dynamically on the web interface.

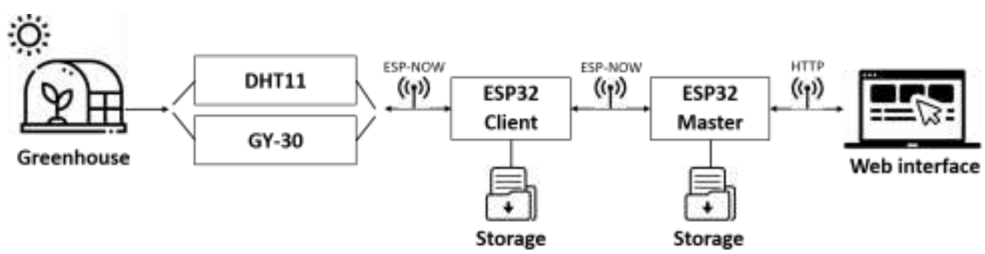

Figure 4. System Workflow Diagram

Data Storage

To store the collected sensor data, a MySQL relational database (part of XAMPP) was used. MySQL enables fast and efficient data storage, retrieval, and management. A custom database containing tables for temperature, air humidity, and light intensity was established (Figure 5).

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
☐	1	**id** 🔑	int(11)			No	*None*		AUTO_INCREMENT
☐	2	**datetime**	datetime			No	current_timestamp()		
☐	3	**ta**	float			No	*None*		
☐	4	**ha**	float			No	*None*		
☐	6	**li**	float			No	*None*		

Figure 5. The structure of the database table used for storing the collected data. 'datetime': timestamp, 'ta': air temperature, 'ha': air humidity, and 'li': light intensity.

Data collected by the sensors is transmitted from the ESP32 Master microcontroller to the server and stored in the MySQL database using PHP scripts. A graphical interface allows users to visually access the database tables and records.

System Design Process

User Interface

The web interface includes three graphical gauges visualizing temperature, air humidity, and light intensity, along with a day/night icon. The gauges were implemented using <svg> elements, and values are dynamically rendered via JavaScript. Each gauge has a circular design, with its background color shifting dynamically based on the measured values (Table 4).

Table 4. Indicator status based on parameter values

Parameter	**Unit**	**Parameter status**		
		Red (High)	Green (Normal)	**Yellow (Low)**
Temperature	°C	Above 28°C	18°C – 28°C	**Below 18°C**
Relative Air Humidity	% RH	Above 80%	60% – 80%	**Below 60%**
Light Intensity	lux	Above 1400	800 – 1400	**Below 800**

Database Connectivity

On the Back-End[3], the system uses PHP[4] to connect to the MySQL database and fetch the most recent sensor records. A script named "get_sensor_data.php" retrieves temperature, air humidity, and light intensity data based on the highest ID value in the database table.

Soft Real-Time Data Updates

The system utilizes the JavaScript Fetch API[5] to periodically request data from the server and update the gauge values in real time. The update interval was set to one minute.

Displaying Timestamp of Last Update

To inform users about data recency, the timestamp of the latest sensor data is extracted from the database and displayed at the top of the web interface.

RESULTS AND DISCUSSION

This system is built using the Arduino development platform and the ESP32 board, which are recognized for their low cost, modularity, and open-source nature. The final product is based on the ESP32 board, which receives data from environmental sensors for processing. The system monitors the cooling and heating needs of the greenhouse based on sensor readings. Since different crops require specific conditions, a web application has been implemented, enabling users to monitor the greenhouse environment through customized icons representing each parameter according to crop-specific needs.

[3] A combination of programming technologies that enhances the functionality of the website and improves user experience.

[4] Personal Home Page

[5] Application Programming Interface

Results

The performance of the developed monitoring system was evaluated over three days (July 1st, 2nd, and 3rd, 2025) inside the plant factory environment. Data for temperature, relative air humidity, and light intensity were recorded continuously at one-minute intervals over 24 hours each day. The following figures illustrate the daily variations of these parameters from 00:00 to 23:00.

As shown in Figure 6, light intensity sharply increased at around 5:00 AM on all three days, reaching a steady level between 1000 and 1200 lux. This increase corresponds with the scheduled activation of the artificial lighting system. Throughout the day, light intensity levels remained relatively constant with minor fluctuations. Short discontinuities observed in the graph are due to temporary power outages. These interruptions did not affect the overall system functionality, as the platform automatically resumed data collection once power was restored.

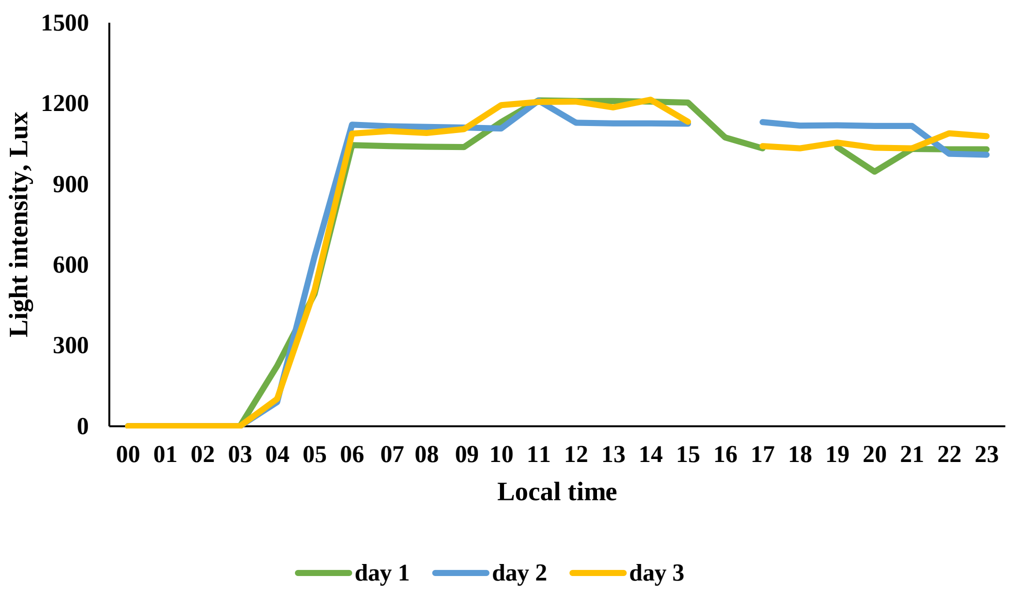

Figure 6. Light intensity during experiments

According to Figure 7, the temperature remained within a stable and desirable range throughout the three days. Between midnight and early morning (00:00 to ~06:00), the temperature stayed around 18 to 19°C. After the activation of artificial lighting, the temperature gradually increased and stabilized between 23 and 25°C. A temporary temperature spike was observed during a brief power outage, which led to a short pause in the ventilation system. Due to the enclosed environment of the plant factory, the temperature increased slightly but quickly returned to the desired range after power was restored and cooling resumed.

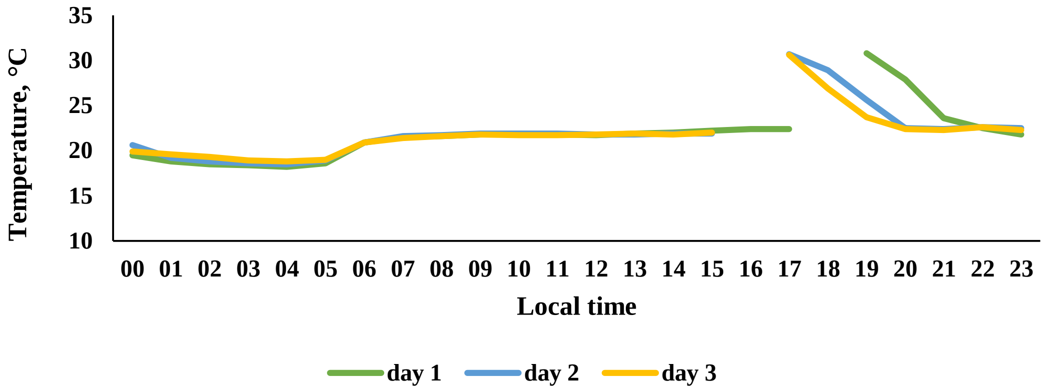

Figure 7. Air temperature during experiments

Relative air humidity was maintained within a stable range of 60% to 75%, which is considered ideal for basil growth under greenhouse conditions (Figure 8). During the night (00:00 to 06:00), air humidity levels slightly increased, then became more stable throughout the day following system activation. A sudden drop in air humidity occurred during the power outage, as the humidifier and ventilation systems temporarily stopped. Once power was restored, the humidity rapidly recovered. This behavior confirms the responsiveness of the monitoring system in recording critical environmental fluctuations in real time.

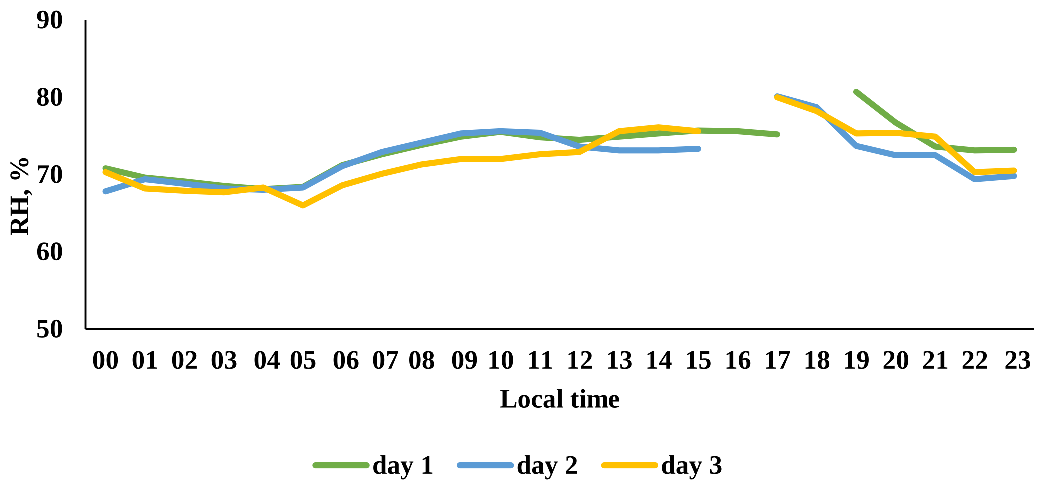

Figure 8. Relative air humidity during experiments

To provide a quantitative summary of the monitored environmental conditions, descriptive statistics, including mean, standard deviation, minimum, and maximum values, were calculated and presented in Table 5.

Table 5. Descriptive statistics of the monitored environmental parameters

Parameter	Mean	Standard Deviation	Min	Max
Temperature (°C)	21.9	2.8	18.2	30.8
Relative Air Humidity (% RH)	72.6	3.4	66	80.7
Light Intensity (lux)	833.8	463	0	1213.4

In addition to environmental measurements, the practical communication performance of the monitoring platform was evaluated and summarized in Table 6. Due to the one-minute refresh interval of the web interface, the practical response time was limited to a maximum of one minute under normal operating conditions. Sensor data were successfully transmitted, stored, and displayed throughout the experimental period. The average processing cycle time of the ESP32 node was approximately 259.5 ms. No communication-related data losses were observed during normal operation due to the implemented buffering mechanism. Additionally, no Wi-Fi interruptions were observed during normal operation. The only interruptions in data acquisition occurred during temporary power outages, when the entire system was disconnected from the power supply. Following power restoration, the monitoring platform automatically resumed data collection, transmission, and web-based visualization without manual intervention. These observations demonstrate the satisfactory reliability and responsiveness of the proposed soft real-time monitoring system under practical operating conditions.

Table 6. Practical communication performance of the monitoring system

Metric	Observation
Data update interval	1 min
Average processing cycle time	259.5 ms
Packet loss due to communication failure	Not observed
Wi-Fi interruptions during normal operation	Not observed
Recovery after power restoration	Automatic

To evaluate the visual performance of the system under real conditions, samples of the graphical status of the web-based user interface were captured and analyzed at 4:00, 19:00, and 21:00 on July 2, 2025. These timestamps represent normal operation, warning conditions, and post-crisis recovery.

At 4:00, the temperature and relative air humidity were within normal ranges, and their corresponding icons appeared in green (Figure 9). However, since the lighting system was turned off during the early morning hours, light intensity was below the acceptable threshold, displaying its icon in yellow (warning). This behavior aligns with the logic implemented in the system and confirms the functionality of the alert mechanism.

Figure 9. Graphical interface status at 04:00 on July 2, 2025, showing normal temperature and relative humidity conditions and a warning status for light intensity due to the scheduled shutdown of the artificial lighting system.

Following a temporary power outage and the restoration of electricity at 19:00, both temperature and air humidity temporarily exceeded their acceptable ranges, resulting in red warning icons. Meanwhile, light intensity returned to its normal range and was shown in green (Figure 10).

Figure 10. Graphical interface status at 19:00 on July 2, 2025, showing warning conditions for temperature and relative humidity immediately after power restoration.

Approximately two hours after power was restored, all environmental parameters returned to their normal ranges, and all interface icons shifted back to green. This demonstrates that the system is capable of accurately reflecting environmental changes and dynamically updating the monitoring interface in soft real-time under stable operating conditions (Figure 11).

Figure 11. Graphical interface status at 21:00 on July 2, 2025, showing recovery of all environmental parameters to their normal ranges.

To support the claim of low-cost implementation, an approximate cost analysis of the hardware components used is presented in Table 7. The total hardware cost was estimated at approximately 58,800,000 IRR, based on the market prices of the components, demonstrating that the platform can be implemented using widely available and affordable components. The use of the ESP32 microcontroller, with low-cost environmental sensors, contributed significantly to reducing the overall system cost while maintaining satisfactory monitoring performance.

Table 7. Approximate hardware and implementation costs of the proposed system

Component	Quantity	Unit Cost (IRR)	Total Cost (IRR)
ESP32	3	7,730,000	23,190,000
DHT11 sensor module	3	1,440,000	4,320,000
GY-30 sensor module	3	2,270,000	6,810,000
Power supply	3	5,580,000	16,740,000
Miscellaneous components	-	-	7,740,000
Total	-	-	58,800,000

Comparison with similar studies

Greenhouse monitoring systems have evolved from simple sensor-based data acquisition platforms toward integrated IoT solutions. Earlier studies, such as Gaikwad et al. and Hong and Hsieh, focused primarily on environmental monitoring and wireless communication using ZigBee and Bluetooth (Gaikwad et al., 2016; Hong & Hsieh, 2016). While these approaches successfully demonstrated remote data collection, their communication range and accessibility were limited compared to internet-based solutions.

More recent studies have adopted IoT architectures and cloud connectivity to improve monitoring capabilities. For example, Arathi et al., Tjahjono et al., Pradeep et al., and Asane et al. integrated multiple environmental sensors with Arduino- or Raspberry Pi-based platforms to provide real-time monitoring and, in some cases, automated control functions (Arathi et al., 2023; Asane et al., 2025; Pradeep et al., 2024; Tjahjono et al., 2023).

The proposed system shares common characteristics with these studies, including the use of low-cost microcontrollers, wireless communication, and continuous monitoring of environmental parameters. However, unlike many existing solutions that rely on local LCD displays, dedicated mobile applications, SMS notifications, or third-party cloud dashboards, the proposed system is designed around a lightweight browser-based platform. The graphical user interface provides dynamic gauges, color-coded visual alerts, and automatic soft real-time updates without requiring software installation or periodic client-side maintenance. This approach improves accessibility and simplifies long-term deployment for greenhouse operators. Furthermore, the use of low-cost components such as ESP32, DHT11, and LDR sensors reduces implementation complexity and overall system cost while maintaining satisfactory monitoring performance. Tables 8, 9, and 10 summarize these comparisons regarding objective and applications, technology, and implementation and costs.

Table 8. Comparison with similar studies in terms of objectives and applications

Source study	Main Objective	Strengths of the Proposed System
(Massah et al., 2023)	Climate control and automation in greenhouses	Monitoring of light intensity in addition to temperature and air humidity
(Taru & Karwankar, 2017)	Monitoring water quality using a web-based display	- Monitoring of light intensity in addition to temperature and air humidity - Web-based interface
(Osman et al., 2018)	Water quality monitoring system	Simultaneous monitoring of multiple parameters through a web environment
(Kumar et al., 2022)	Crop monitoring system in greenhouses	- Web-based UI and graphical gauges for monitoring - Usable on both mobile and desktop devices
(Tjahjono et al., 2023)	Hydroponic greenhouse monitoring and control using multiple sensors and actuators	- Browser-based monitoring platform - Dynamic graphical visualization - Automatic real-time updates

Table 9. Comparison with similar studies in terms of technology

Source Study	Main Objective	Strengths of the Proposed System
(Hong & Hsieh, 2016)	Bluetooth communication	- Communication via Wi-Fi - Web-based user interface
(Gaikwad et al., 2016)	ZigBee wireless module	- Communication via Wi-Fi
(Hammami, 2019)	Arduino-based control board	-Use of the ESP32 board with built-in Wi-Fi capability
(Tembhurne et al., 2022)	LCD display of parameter values	- Display of parameter values on the web interface
(Arathi et al., 2023)	Use of the Arduino microcontroller board	-Use of the ESP32 microcontroller board with Wi-Fi capability
(Asane et al., 2025)	Raspberry Pi as the main controller	- Use of ESP32 with built-in Wi-Fi instead of Raspberry Pi - Web-based interface

Table 10. Comparison with similar studies in terms of implementation and costs

Source study	Main Objective	Strengths of the Proposed System
(Risheh et al., 2020)	Smart irrigation using neural networks	- Simpler implementation - No need for complex algorithms
(Alaviyan et al., 2020)	Monitoring using fuzzy logic (temperature, light intensity, air humidity)	- Easier implementation - No need for high-end hardware
(Pradeep et al., 2024)	Multi-sensor IoT monitoring system with intelligent algorithms	- Simpler and lower-cost implementation - Reduced hardware and software complexity

CONCLUSIONS

Experiments conducted over three consecutive days in a controlled plant factory environment demonstrated that the proposed system effectively recorded environmental data and transmitted it to a web interface via HTTP. The system showed stable performance under normal conditions and critical times, such as power outages. During such disruptions, the parameters like temperature and relative air humidity temporarily deviated from their normal ranges. The system correctly issued visual alerts through the graphical interface, and once conditions returned to normal, the display was automatically updated. The system's simplicity, low cost, remote monitoring capability, automatic data updates, and intuitive graphical visualization are its most prominent advantages. Considering its performance in a pilot-scale setup, this system can be recommended as a practical solution for monitoring the climate of small to medium-sized greenhouses.

Although the system demonstrated satisfactory performance, the sensors used were not validated against calibrated reference instruments. Therefore, future studies should include quantitative sensor validation using standard measurement devices and performance indicators such as RMSE, MAE, and percentage error. Temporary interruptions in data acquisition occurred only during external power outages. Furthermore, future implementations may incorporate battery backup systems or uninterruptible power supplies (UPS) to prevent data gaps during power failures.

As a perspective for future development, the scalability of the proposed system can be extended to larger greenhouses. At a medium scale, painlessMesh networks based on ESP32 enable multiple sensing and control nodes to communicate across different points of the greenhouse without relying on a centralized access point. For longer distances or distributed clusters of greenhouses, LoRa technology offers a suitable alternative, as it provides low-power transmission of sensor data over several kilometers despite its lower data rate. Thus, the system can be adapted and deployed in larger and more diverse agricultural settings.

In addition, the platform can be extended to support active and intelligent greenhouse control by integrating actuators such as ventilation fans, irrigation pumps, humidifiers, and artificial lighting, allowing the ESP32 controller to automatically respond to sensor measurements and maintain environmental parameters within predefined thresholds using intelligent decision-making algorithms like fuzzy logic or machine learning models. For example, if the measured temperature exceeds or falls below predefined thresholds, the ESP32 controller can automatically activate cooling or heating devices to restore the desired environmental conditions. This would transform the current monitoring platform into an automated greenhouse management system.

REFERENCES

Alaviyan, Y., Aghaseyedabdollah, M., Sadafi, M., & Yazdizade, A. (2020). Design and manufacture of a smart greenhouse with supervisory control of environmental parameters using fuzzy inference controller. *2020 6th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS)*, (pp. 1-6). IEEE. (In Persian) https://doi.org/10.1109/ICSPIS51611.2020.9349619

Arathi, P., Krishnan, A., Krishna, D. Y., Gagana, Y., & Patel, G. (2023). Greenhouse monitoring and controlling using arduino. *International Journal of Engineering and Management Research*, *13*(3), 138–139.

Asane, S., Salve, S., Birajdar, S., Borkar, S., & Gawali, P. (2025). IoT-Driven Smart Greenhouse System for Real-Time Environmental Monitoring. *European Journal of Scientific Research and Reviews*, *2*(4), 220–220. https://doi.org/10.5455/EJSRR.20250430014602

Gaikwad, A., Ghatge, A., Kumar, H., & Mudliar, K. (2016). Monitoring of smart greenhouse. *International Research Journal of Engineering and Technology (IRJET)*, *3*, 573–575.

Hammami, A. (2019). Smart environment data monitoring. *2019 International Conference on Computer and Information Sciences (ICCIS)* (pp. 1-6). IEEE. https://doi.org/10.1109/ICCISci.2019.8716469

Hong, G.-Z., & Hsieh, C.-L. (2016). Application of integrated control strategy and bluetooth for irrigating romaine lettuce in greenhouse. *IFAC-PapersOnLine*, *49*(16), 381–386. https://doi.org/10.1016/j.ifacol.2016.10.070

Kumar, A., Singh, V., Kumar, S., Jaiswal, S. P., & Bhadoria, V. S. (2022). IoT enabled system to monitor and control greenhouse. *Materials Today: Proceedings*, *49*, 3137–3141. https://doi.org/10.1016/j.matpr.2020.11.040

Massah, J., Asefpour, K., Sami, S., & Jazayeri, I. (2026). Design, development, and performance evaluation of a GPS and machine vision-based navigation system for an intelligent field guard robot. *Iran Agricultural Research*, *45*(1), 113–126.

Massah, J., Tahmasebi, H., Roozban, M. R., & Azadegan, B. (2023). The effects of an automated system for climate control on the yield of greenhouse crops (case study on rose). *Journal of Crops Improvement*, *25*(1), 269–278. https://doi.org/10.22059/jci.2022.339739.2687

Osman, S. O., Mohamed, M. Z., Suliman, A. M., & Mohammed, A. A. (2018). Design and implementation of a low-cost real-time in-situ drinking water quality monitoring system using arduino. *2018 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)* (pp. 1-7). IEEE https://doi.org/10.1109/ICCCEEE.2018.8515886

Pradeep, M., Rinku, D. R., Swapna, P., Jyothi, V., Athiraja, A., & Prasannakumar, G. (2024). An IoT Based Greenhouse Remote Monitoring System for Automation of Supervision for Optimal Plant Growth. *2024 10th International Conference on Advanced Computing and Communication Systems (ICACCS)* (Vol. 1, pp. 797-802). IEEE. https://doi.org/10.1109/ICACCS60874.2024.10716941

Risheh, A., Jalili, A., & Nazerfard, E. (2020). Smart Irrigation IoT solution using transfer learning for neural networks. *2020 10th International Conference on Computer and Knowledge Engineering (ICCKE)* (pp. 342-349). IEEE. https://doi.org/10.1109/ICCKE50421.2020.9303612

Sami, S., Hajian, M., & Nazghelichi, T. (2026). A Time-Driven Approach Leveraging Universally Accessible Features for Photovoltaic Power Forecasting. *Iranica Journal of Energy & Environment*, *17*(2), 382–396. https://doi.org/10.5829/ijee.2026.17.02.13

Taru, Y. K., & Karwankar, A. (2017). Water monitoring system using arduino with labview. *2017 International Conference on Computing Methodologies and Communication (ICCMC)* (pp. 416-419). IEEE. https://doi.org/10.1109/ICCMC.2017.8282722

Tembhurne, V., Bhatkar, M., & Ikhe, Y. (2022). Automatic greenhouse environment monitoring and controlling system. *International Journal of Research in Engineering, Science and Management*, *5*(12), 30–33.

Tjahjono, B., Fauzia, Y., & Kurnia, R. (2023). Prototyping of precission farming hydroponic garden using arduino using design thinking method at puriponic greenhouse depok. *International Journal of Science, Technology & Management*, *4*(2), 305–310. https://doi.org/10.46729/ijstm.v4i2.774