Enhancing Security and Customization in IoT-Based Systems with Applications to Broiler Room Management

S. Sabo [1,*]
A. M. Umaru [2]
L. A. Yusuf [3]

Received: 10/04/2025
Revised: 12/07/2025
Accepted: 13/07/2025

[1] Department of Computer Science, National Open University of Nigeria, Nigeria
[2] Department of Computer, Yusuf Maitama Sule University, Nigeria. Kano, Nigeria
[3] Directorate of Information and Communication Technology, National Open University of Nigeria. , Nigeria
*Corresponding Author's Email: 1440msusabo@gmail.com

Enhancing Security and Customization in IoT-Based Systems with Applications to Broiler Room Management

Shamsu Sabo
Department of Computer Science,
National Open University of Nigeria
Nigeria
1440msusabo@gmail.com

Abubakar Muhammad Umaru
Department of Computer
Yusuf Maitama Sule University
Nigeria.
Kano, Nigeria
Amumaru@yumsuk.edu.ng

Lawan Abdullahi Yusuf
Directorate of Information and Communication Technology,
National Open University of Nigeria.
Nigeria
lyusuf@noun.edu.ng

Abstract— Although IoT-based solutions are widely employed in various fields, they frequently rely on traditional email notifications or third-party platforms like ThingSpeak, Blynk, AWS IoT, Microsoft Azure IoT, and Google Cloud IoT for monitoring and control. These strategies do, however, have significant disadvantages, such as restricted adaptability and customization, security flaws, and privacy issues with data. In order to overcome these constraints, this work presents an improved software solution that is incorporated into an ESP32 IP address-based interface, with particular applications for managing broiler rooms. By combining IP address-based access with secure login authentication using a username and password, the suggested method makes sure that only authorized users can watch over and manage the broiler room. Significant gains in security, privacy, flexibility, and user-specific customization are shown by the experimental findings. The system is easy to install and use. It offers a reliable substitute for IoT-based systems by doing away with the need for third-party platforms. Because of its flexible nature, users can customize features to meet particular requirements, getting around the strict limitations of third-party frameworks. This system provides secure remote access with increased mobility, in contrast to email-based alternatives that lack real-time involvement. This approach adds login authentication and customization styles to the current program by integrating hardware and parts of the software from our prior system [1]. The suggested approach performs better than the prior system and other IoT-based systems that rely on third-party platforms in terms of security and customization.

Keywords— Security in IoT Systems, IP Address-Based Interface, Secure Login Authentication, ESP32 Microcontroller, Third-Party Dependencies, Poultry Farm Management.

I. Introduction

The poultry farming business has seen a substantial transformation because of the incorporation of Internet of Things (IoT) technology into broiler room control systems, which allow for precise, real-time management of crucial environmental elements like temperature, humidity, and lighting. These developments are essential for maintaining the health of chickens, raising output, and increasing operational effectiveness. Notwithstanding these advantages, a large number of IoT-based systems currently in use rely on third-party platforms such as ThingSpeak, Blynk, AWS, Microsoft Azure, and Google Cloud, which present significant security, privacy, and customization issues. For example, ThingSpeak has been connected to data transfer flaws that could allow unwanted access to private data [2]. Blynk, as reported, has also been associated with security vulnerabilities. Specifically, a specially crafted font file embedded in a PDF can trigger an out-of-bounds read vulnerability in Adobe Acrobat Reader, potentially exposing sensitive data [3]. While updating to the latest version of Blynk may address some security issues, users should exercise caution, as both older and even some of the most recent versions have been reported to have vulnerabilities. These platforms impose strict frameworks that might restrict their ability to adapt to certain operating requirements.

Furthermore, current solutions, such as email-based notification systems, are less effective in dynamic contexts because they lack real-time control and customization options, even though they are helpful for remote alerts.

On the other hand, using the ESP32 microcontroller to create an IP address-based web interface offers a more straightforward method. In contrast to the more intricate configuration needed for platforms, this approach uses HTML, CSS, and JavaScript code to produce an intuitive user interface that can be accessed through a browser and essentially operates as a stand-alone system. Programming languages such as Python, C++ with MicroPython, or Arduino IDE are used to build functionality and control on the ESP32. By putting all data under the user's control, this simplified method not only makes deployment easier but also eliminates the need for third-party platforms, improving security and privacy.

II. Literature Review

The incorporation of IoT technology into broiler room management systems continues to face a number of obstacles, especially with regard to security, privacy, adaptability, and system customization. This section examines pertinent research, highlighting the benefits, drawbacks, and potential areas for development of the email notification, ThingSpeak, Blynk, AWS, Microsoft Azure, and Google Cloud platforms.

In IoT applications, ThingSpeak has been extensively used for data collecting, analysis, and visualization. One such use was demonstrated by [4], who used IoT sensors to monitor chicken weight and chick cycles in real time, enhancing the quality of manufacturing. The platform's intrinsic limits in terms of flexibility for customized system design, data privacy, and potential security vulnerabilities were noted, despite the fact that ThingSpeak offered an efficient solution for data display. ThingSpeak was used in an IoT-based system for weather monitoring in a related study [5], enabling

remote display of environmental data. The study did draw attention to the platform's incapacity to accommodate the intricate adaptations needed for a variety of application scenarios.

Additionally, research conducted by [6] and [7], who integrated ThingSpeak for environmental monitoring in poultry farms, showed that although the platform enabled efficient data visualization, its limited customization capabilities and potential security flaws—like unauthorized access to private information—may have prevented it from reaching its full potential in handling increasingly complicated IoT systems.

Blynk's automated features and user-friendly interface have made it a popular option for IoT-based poultry farming systems. [8] Suggested a smart poultry farm system that uses Blynk to automate important tasks, including environmental management, egg collection, and feeding and water supply. Blynk's real-time monitoring capabilities and ease of integration were credited with the system's success. But like ThingSpeak, the study pointed out that Blynk's possible security flaws and limited customization options would make it less suitable for specific applications. Similar to ThingSpeak, which could expose the system to illegal access and data breaches. Blynk was integrated into an autonomous poultry cage system in a different study [9], simplifying sensor-based environmental factor control. Although the study recognized Blynk's advantages in system automation, it also highlighted the platform's security and customization shortcomings, which could make it more difficult to use in more intricate IoT configurations.

According to studies like [10] and [11], users' capacity to efficiently monitor and manage email-based systems is severely limited by their lack of flexibility. The system's overall operating efficiency and responsiveness are decreased because users cannot remotely control system parameters (temperature, humidity, and ventilation) via emails.

The security and privacy issues of cloud-based IoT systems, such as AWS, Microsoft Azure, and Google Cloud, are assessed in [12]. The authors point out that although these platforms have strong integration and scaling features, they are vulnerable to issues like data interception in transit and illegal access to stored data. The study also notes that these systems' inflexible foundation frequently restricts user-specific customization, which presents difficulties for applications that need customized setups. These problems highlight the necessity of ongoing improvements in security procedures and more adaptable platform architectures in order to successfully meet user-centric needs.

IoT-based systems have advanced thanks to platforms like ThingSpeak, Blynk, AWS, Microsoft Azure, and Google Cloud, but there are still a number of typical issues, especially with regard to security, privacy, and system customization. Security flaws are introduced by reliance on third-party systems where sensitive data may be compromised by illegal access or cyberattacks. Furthermore, these platforms have few modification possibilities, which makes it difficult to modify the system to meet the particular requirements of various applications, like managing a broiler room.

This study suggests a browser-based approach that improves security, privacy, and flexibility in order to overcome these drawbacks. Only authorized users can monitor or operate the system by using an IP address and login credentials to access the system interface. This ensures strong access control and guards against unwanted manipulation. The local area network is used for demonstration purposes; for a bigger system, a virtual private network might be used in its place to enable access from any location on the globe. In this configuration, the user receives data from a central server, and the monitoring and control interface is only accessible by those who have the correct IP address. This approach reduces the possibility of security lapses and outside meddling by avoiding third-party platforms.

In addition, the suggested solution offers more customization options than cloud-based platforms, enabling users to modify features like automation and monitoring criteria to suit their needs. It also addresses the limits and delays of email-based solutions by enabling real-time monitoring and control using a browser interface. This method is a safer and more adaptable way to handle broiler rooms since it guarantees quick reactions and enhanced flexibility.

III. METHODOLOGY

We adopted the approach of [1], which accessed and controlled the system remotely using a web interface. Login authentication was introduced to the current web interface with the intention of enhancing system security and adaptability, which is how the proposed system differs from the current one. In order to use the broiler room system's control and monitoring functions, unauthorized users must now authenticate using legitimate login credentials (password and username).

Figure 1 illustrates the main components of the proposed system, including the web interface, inputs, control, and output units.

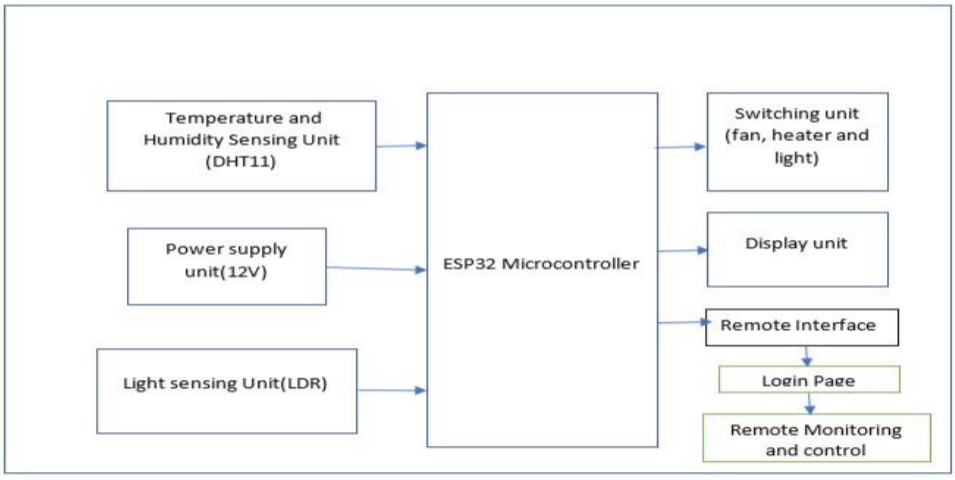

Figure1: Block Diagram of the Proposed System

A. Hardware Implementation

The hardware components, which include the ESP32 development board, sensors (temperature, humidity, and light), actuators (fan, heating system, and LEDs), and the power supply unit, are all the same as in the previous design. The ESP32 functions as the central microcontroller, gathering data from sensors and controlling the actuators according to predetermined thresholds. The system uses a 12V battery with an LM7805 regulator for stable power distribution to the components. The ESP32's GPIO pins are used to interface with sensors and actuators, and its Wi-Fi capabilities for communication ensure smooth hardware and software integration. Figure 2's circuit diagram from the earlier article [1] applies to the current system as well.

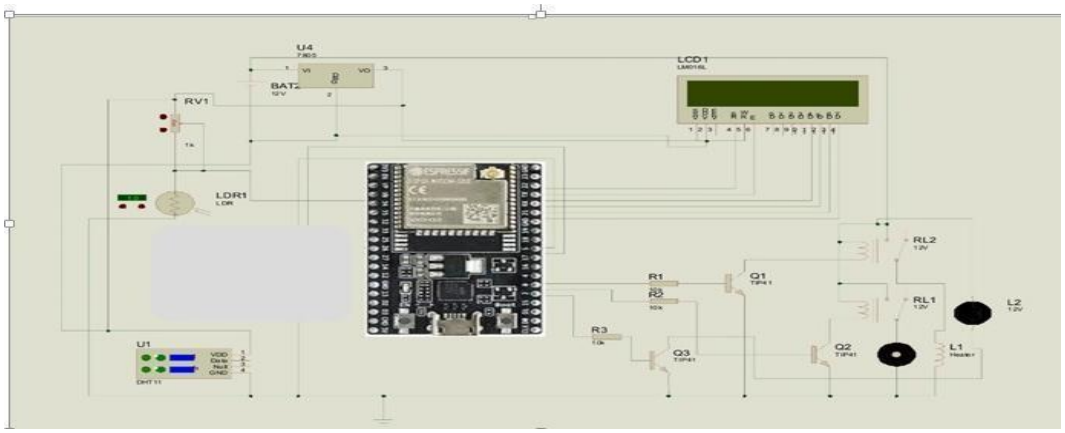

Figure 2: proposed system's circuit diagram.

B. Software Implementation with Authentication

The main software improvement in this study is the addition of login authentication for the web-based interface, which can be accessed by entering the ESP32's IP address into a web browser. In the previous version of the system [1], users could access and control broiler room parameters directly without the need for authentication, which presented a potential security risk because unauthorized users could change the system's management and monitoring features. A login page was incorporated into the system to guarantee that only authorized individuals could access the crucial control features. Every time a user wants to access the system, they must enter their username and password on this page. After a successful identification process, users can monitor and adjust important components like temperature, humidity, light, and other vital aspects. A crucial part of the authentication solution is demonstrated by the code sample that follows: const char* http_username = "shamsu557"; const char* http_password = "@Shamsu1440";

```
bool isAuthenticated(AsyncWebServerRequest *request) {
if (!request->authenticate(http_username, http_password)) {
request->requestAuthentication();     return false;
  }
 return true;
}
```

The isAuthenticated() function in this code determines whether the user-supplied credentials correspond to the http_username and http_password that have been predefined. The system asks the user to enter their credentials again if authentication is unsuccessful. Users can access real-time data and engage with the system's functionality on the monitoring and control page after properly authenticating. Users can keep an eye on variables including temperature, humidity, light levels, and device statuses (fan, heater, and light) using the same monitoring page as the prior system. Additionally, the page allows users to manually turn on and off the fan, heater, and light.

In order to further safeguard the system, a secondary password is used to restrict access to manual control of the devices (fan, heater, and light), guaranteeing that only authorized users can alter the devices' states:

```
bool isControlAuthenticated(AsyncWebServerRequest *request) {  if (request->hasParam("control_password")) {
  AsyncWebParameter* p = request->getParam("control_password");  if (p->value().equals(http_password)) {  return true;
  }  }  return false;
}
```

In this instance, before permitting modifications to the device states, the isControlAuthenticated() method verifies that the control password is correct.

The ESP32's IP address is used by the system to enable remote access, providing a straightforward and safe way to communicate with the broiler room control system. An extra degree of protection is provided by the login page, which keeps sensitive control features safe from unwanted access.

The interface of the login page, which users must navigate before they can access the system, is seen in Figure 3 below. Users can control the broiler room environment after successfully authenticating, as shown in Figure 4, which takes them to the monitoring and control page.

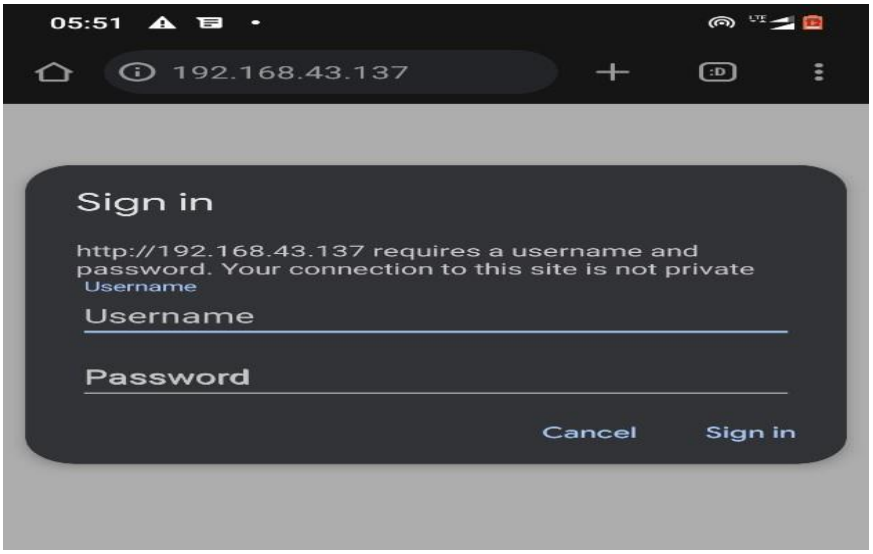

Figure 3: Web Interface Login Authentication Screen

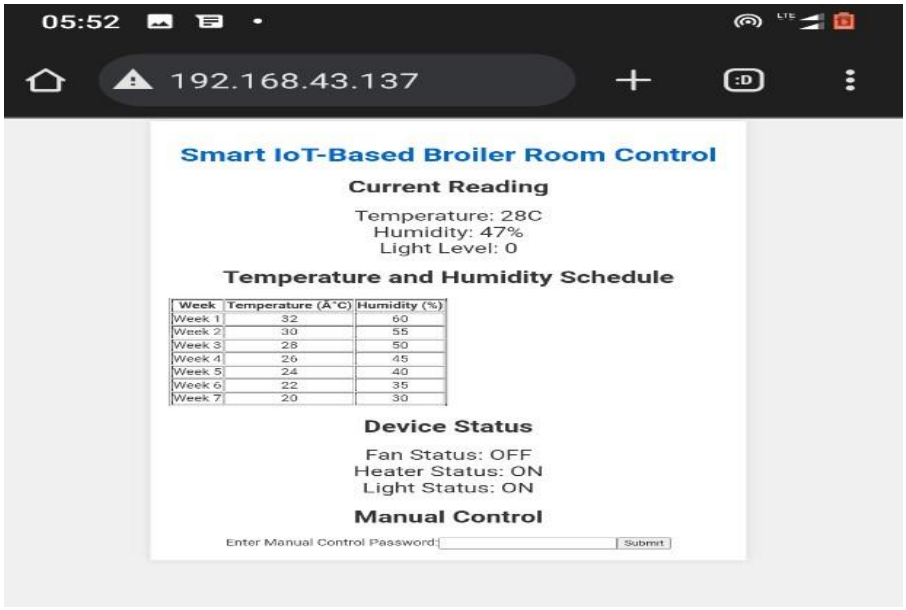

Figure 4: Remote monitoring and control Page

C. Complete System Demonstration

A thorough overview of the fully set up and functional system is given in Figure 5. This picture shows the device's physical configuration as well as the web interface that shows its status and current data readings. It is the ideal example of how smooth operation and integration are made possible by the Internet of Things-based broiler room controller system with an IP address-based interface.

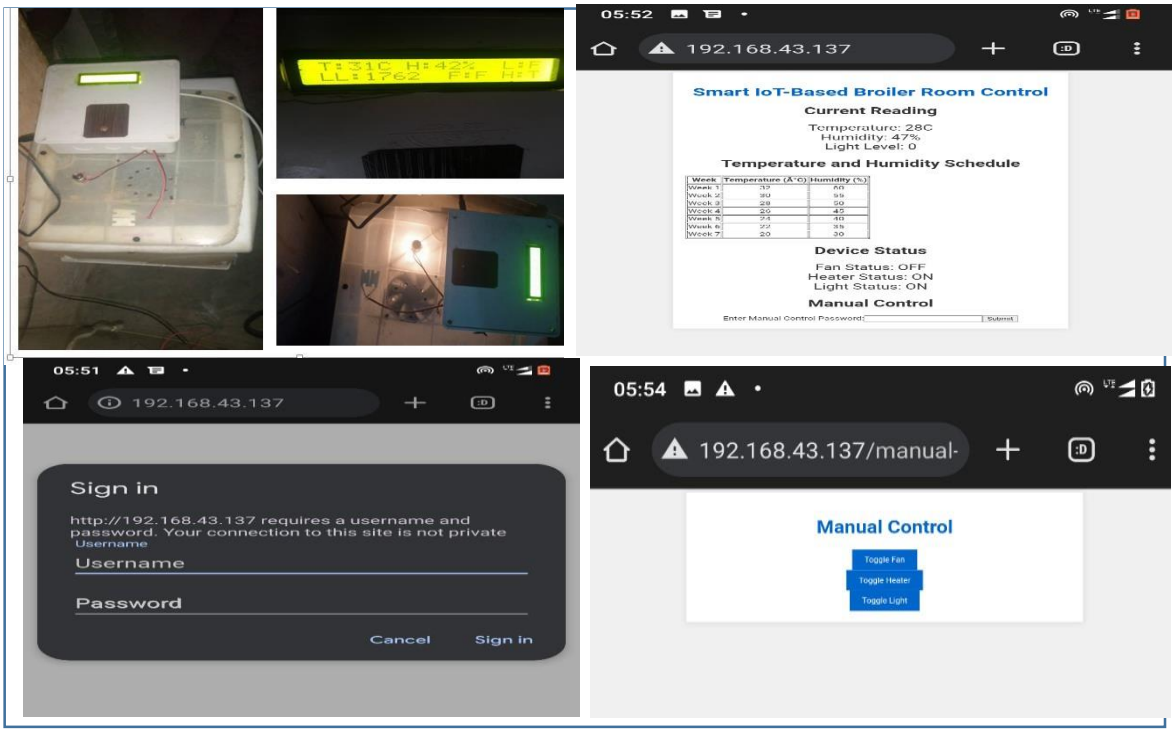

Figure 5: Fully Developed Systems with the Interface

The system's architecture incorporates advanced security measures and customizable features to improve its functionality. In order to ensure effective broiler management, its adaptable physical design and safe IP-based web interface put user comfort and data security first.

IV. RESULTS AND DISCUSSION

This study highlights the improved security, adaptability, and customization potential of an Internet of Things (IoT)-based broiler room management system. It enhances user-centric control while addressing the critical limitations of earlier prototypes and third-party platforms.

A. A. System Security and Flexibility

By adding login authentication for access via the IP address, the security of the Internet of Things-based broiler room management system has been greatly improved. Because the prior version [1] permitted unfettered access within the local network, it had security vulnerabilities.

The new system lowers the risk of unwanted access by requiring login credentials, which guarantee that only authorized users may engage with its control and monitoring capabilities.

B. Flexibility and Customization

With the wide range of customization options offered by the upgraded system, users can adapt the functionality and interface to meet their unique needs. This system offers complete control over its operations and design, which makes it flexible enough to accommodate a variety of IoT applications, in contrast to third-party platforms that restrict design and feature modifications. Additionally, by doing away with the need for outside services, direct IP address access simplifies setup and increases system flexibility.

C. Security Vulnerabilities in Third-Party Platforms

Although third-party IoT solutions are widely used, they pose security risks. There have been reports of security flaws in ThingSpeak that put users in danger and stop it from offering the customization needed for particular applications. Just as unpatched or outdated Blynk versions can jeopardize security, so too can AWS, Microsoft Azure, Google Cloud, and the most recent version. Because third-party platforms typically entail exchanging user data with external parties, which raises confidentiality problems, data privacy is another important issue. However, the IP address-based strategy described in this study reduces these risks by strengthening privacy protection, lowering reliance on external services, and restricting access to data to those having the necessary IP address, login credentials, and network.

D. Comparison with the Previously Developed System

IP address-based access was also used by the previous system [1], which provided flexibility and ease of use but lacked security because login authentication was not required. By adding login authentication, the upgraded version strengthens this base and fixes the shortcomings of the old system while maintaining its benefits for quick deployment and

customization. The improved system offers more control over functionality and security than third-party platforms, offering an adaptable, scalable, and safe option for IoT-based broiler room management.

E. Testing and Evaluation

The robustness and efficacy of the login authentication functionality were assessed through extensive testing. Only authorized users were able to access the system, as demonstrated by a variety of test scenarios, including both successful and unsuccessful login attempts. The system's capacity to handle several connections at once without sacrificing stability or reaction time was confirmed by performance tests. The outcomes show that the system is effective, safe, and suitable for administration based on the Internet of Things.

The improved prototype was tested under a variety of real-world conditions during a seven-week deployment in a nearby broiler room. Security, dependability, and responsiveness were the main testing priorities.

Table 1: Security and Performance Evaluation of the IoT System

Evaluation Metric	Description	Result	Notes
Valid Login Success Rate	Percentage of successful logins	100%	20 users over LAN; 200 valid login attempts tested
Invalid Login Rejection Rate	Incorrect credentials blocked	100%	10 users; 100 invalid attempts blocked; no unauthorized access observed
Command Execution Delay	Time from input to device action	~210 ms	Based on timing measurements
System Uptime	Duration of uninterrupted operation	7 weeks (100%)	No crashes; stable under manual reset
Remote Access Range (LAN)	Wi-Fi communication range	10–20 meters	Dependent on signal strength
Communication Protocol	Interface transport protocol	HTTP	Local only; not encrypted
Credential Storage Method	Storage of login information	Hardcoded (firmware)	Planned migration to hashed storage

F. Key Innovations And Deployment Strategy

The local area network (LAN)-first deployment architecture of this solution makes it distinct since it incorporates login authentication without depending on outside services, allowing for localized, private, and secure management over IoT operations. It uses embedded modular user interfaces made with HTML, CSS, and JavaScript, which gives it more flexibility and customization than the majority of IoT frameworks.

The technology offers simple deployment even in contexts with limited resources and guarantees total data privacy by avoiding any external data sharing.

Additionally, the system is built to scale in the future using tools like Render and AlwaysData, which will make it easier to:

a. HTTPS encryption for secure access
b. Storage of credentials supported by Bcrypt hashing
c. Access to the internet worldwide while maintaining security

When necessary, this two-phase strategy provides a smooth transition to cloud-enabled expansion while ensuring safe local operation during first deployment.

Table 2: Comparative Feature Overview

Feature	ThingSpeak	Blynk	AWS IoT	Proposed System (Prototype)	Proposed System (Future Deployment)
Requires Internet	Yes	Yes	Yes	No (LAN-based)	Yes (via Render/AlwaysData)
Custom UI/UX	Limited	Limited	Moderate	Full control (HTML/CSS)	Full control (HTML/CSS)
Security (Default)	Moderate	Moderate	High (paid)	High (Local Auth)	High (Bcrypt + HTTPS)
Real-time Control	Limited	Yes	Yes	Yes	Yes
Cost	Varies	Varies	High	Low / Free (Open source)	Low (free-tier hosting)
Offline Functionality	No	No	No	No	No
Deployment Complexity	Medium	Low	High	Low (plug-and-play)	Moderate (DB setup + hosting)

V. CONCLUSION

By improving security and adaptability, this study tackles important problems with IoT-based systems, especially in the operation of broiler rooms. In contrast to third-party platforms such as ThingSpeak, Blynk, AWS, Microsoft Azure, and Google Cloud, which offer restricted customization choices and security flaws, the suggested method guarantees safe access via login authentication.

Additionally, it permits a great deal of flexibility, allowing users to modify the system to suit their own requirements without depending on third-party platforms. The lack of customization and remote control that comes with email alerts is also addressed by this method, which offers a safer, more adaptable, and more user-friendly way to oversee broiler room operations while putting data privacy and system effectiveness first.

The proposed system differs from third-party platforms such as Google Cloud, Microsoft Azure, AWS, Blynk, and ThingSpeak in that it eliminates cloud dependency, enables LAN-based secure access, supports direct interface modification, improves data privacy and control, and can be advantageous for low-resource or rural areas.

Additionally, it fixes security flaws in the previous version [1] and provides a path to internet-based deployment that is suitable for the future by utilizing encryption and password hashing. With a focus on both functionality and user autonomy, this system offers an innovative, useful, and user-centric alternative for actual IoT installations.

REFERENCES

[1] A. M. Umaru, S. Sabo, J. A. Ibrahim, and A. T. Sulaiman, "Design and Implementation of Smart IoT-Based Broiler Room Controller," *Journal of Science Technology and Education*, vol. 12, no. 2, 2024. [Online]. Available: www.atbuftejoste.com.ng

[2] D.-A. Andrioaia, "Cyber Security Analysis of IoT Devices Transmitting Data in the ThingSpeak Platform Cloud," *Journal of Engineering Studies and Research*, vol. 28, no. 3, pp. 29–33, Oct. 2022, doi: 10.29081/jesr.v28i3.003.

[3] Talos Intelligence, "Vulnerability Report TALOS-2024-2070," [Online]. Available:

[4] https://talosintelligence.com/vulnerability_reports/TALOS-2024-2070. [Accessed: 03Jan-2025].

[5] R. Sasirekha, R. Kaviya, G. Saranya, A. Mohamed, and U. Iroda, "Smart Poultry House Monitoring System Using IoT," in *E3S Web of Conferences*, EDP Sciences, Jul. 2023, doi: 10.1051/e3sconf/202339904055.

[6] Girija C., "Internet of Things (IoT) Based Weather Monitoring System," [Online]. Available: www.ijert.org

[7] A. Susanto, A. R. Ardi Agung, M. Ibrahim, T. D. Sugiarto, A. Yuswanto, and B.

[8] Wibowo, "Design of a Temperature and Humidity Monitoring System in Broiler Farms Using Internet of Things-Based ThingSpeak," *Jurnal Komputer dan Elektro Sains*, vol. 1, no. 1, pp. 9–13, Jun. 2023, doi: 10.58291/komets.v1i1.92.

[9] R. Patil and M. Anusha, "IoT Based Smart Poultry Farm," *International Research Journal of Engineering and Technology (IRE Journals)*, vol. 4, no. 11, pp. 104–107, 2020. [Online].

[10] Available:https://www.irejournals.com/formatedpaper/1702461.pdf. [Accessed: 03-Jan-2025].

[11] S. Swetha, S. N. Ilakkiya, R. Nevetha, S. Sarathy, and R. Deepa, "Automatic Room Temperature and Monitoring System Using Arduino," *International Journal of Innovative Research in Science, Engineering and Technology*, vol. 8, no. 4, pp. 4647– 4651, Apr. 2019, doi: 10.15680/IJIRSET.2019.0804128.

[12] R. A. Butt, T. Rehman, and M. A. Qureshi, "A Smart IoT-Enabled Cage for the Farming of Ground Birds †," *Engineering Proceedings*, vol. 46, no. 1, 2023, doi: 10.3390/engproc2023046026.

[13] V. M. Dilpak, A. S. Devade, V. S. Mare, S. M. Shinde, and A. D. Jori, "IoT Based Smart Humidity and Temperature Monitoring System Additional by using WiFi Module," *International Journal of Advance Research and Innovative Ideas in Education*, vol. 8, no. 3, pp. 451–455, May-Jun.2022. [Online]. Available:

[14] https://ijariie.com/AdminUploadPdf/IoT_Based_Smart_Humidity_and_Temperature_Monitoring_System_Additional_by_using_WiFi_Module_ijariie16564.pdf. [Accessed: Feb. 17, 2024].

[15] M. Z. Islam, M. A. Based, M. M. Rahman, and M. Rahman, "IoT-Based Temperature and Humidity Real-Time Monitoring and Reporting System for COVID-19 Pandemic Period," *International Journal of Scientific Research and Engineering Development*, vol. 4, doi: 10.13140/RG.2.2.16906.80321.

[16] N. Singh, R. Buyya, and H. Kim, "Securing Cloud-Based Internet of Things: Challenges and Mitigations," *Sensors*, vol. 25, no. 1, p. 79, Dec. 2024, doi: 10.3390/s25010079.