talenta PUBLISHER

Journal of Technomaterial Physics
Journal homepage: https://talenta.usu.ac.id/JoTP

Design and Analysis of an Integrated IoT Electronic Menu Based on The MERN (MongoDB, Express.js, React.js, and Node.js) Stack Website

Tulus Ikhsan Nasution and Syaipuddin Muda Pane
Department of Physics, Faculty of Mathematics and Natural Sciences, Universitas Sumatera Utara, Medan, North Sumatra 20155, Indonesia
*Corresponding Author: tulusikhsan@usu.ac.id

ARTICLE INFO

Article history:
Received 11 August 2025
Revised 07 September 2025
Accepted 16 October 2025
Available online 16 October 2025

E-ISSN: 2656-0755
P-ISSN: 2656-0747

How to cite:
T. I. Nasution, and S. M. Pane" Design and Analysis of an Integrated IoT Electronic Menu Based on the MERN (MongoDB, Express.js, React.js, and Node.js) Stack Website," Journal of Technomaterial Physics, vol. 07, no. 02, pp. 71-79, Aug. 2025, doi: 10.32734/jotp.v7i2.22434.

ABSTRACT

Restaurants often rely on traditional order-taking processes that are prone to delays, miscommunication, and human error. Earlier IoT-based electronic menus already introduced better order handling, but are limited by unidirectional communication and the scalability constraints of PHP + MySQL architectures. This research investigated the design and analysis of an MERN (MongoDB, Express.js, React.js, and Node.js) stack–based electronic menu, integrated with an Arduino Mega 2560 and ESP8266 module, to enable two-way communication between the client and server. The device was tested under varying Wi-Fi signal conditions, taking into account environmental factors such as distance, humidity, and obstacles. Results showed that Wi-Fi signal strength decreases linearly with distance and is attenuated by 6-7 dBm when passing through a concrete barrier, while high humidity (up to 95%) results in an additional 11-12 dBm reduction. Web application testing demonstrated the reliable data transmission of the MERN stack, which shows a striking difference compared to PHP + MySQL, outperforming it in scalability and processing speed. These results showed the advantages of MERN in supporting an interactive, scalable, and responsive electronic menu. Although measurements were based on averaged trials without complete statistical analysis, a promising approach was demonstrated in improving efficiency and reliability in restaurant environments.

Keywords: Internet of Things, Microcontroller, Arduino Mega 2560, MERN, Humidity, Wi-Fi.

ABSTRAK

Restoran sering mengandalkan proses pemesanan tradisional yang berisiko terhadap keterlambatan, miskomunikasi, dan kesalahan manusia. Menu elektronik berbasis IoT sebelumnya sudah memperkenalkan penanganan pesanan yang lebih baik tetapi terbatas oleh komunikasi satu arah dan kendala skalabilitas arsitektur PHP + MySQL. Penelitian ini mengusulkan desain dan analisis menu elektronik berbasis MERN (MongoDB, Express.js, React.js, dan Node.js) stack yang terintegrasi dengan Arduino Mega 2560 dan modul ESP8266 untuk memungkinkan komunikasi dua arah antara klien dan server. Perangkat diuji menggunakan kondisi sinyal Wi-Fi yang bervariasi di bawah faktor lingkungan, seperti jarak, kelembapan, dan penghalang. Hasil menunjukkan bahwa kekuatan sinyal Wi-Fi menurun secara linear seiring jarak dan tereduksi sekitar 6–7 dBm dengan penghalang beton, sementara kelembapan tinggi (hingga 95%) memberikan tambahan reduksi 11–12 dBm. Pengujian aplikasi web menunjukkan transmisi data MERN stack yang andal, yang memperlihatkan perbedaan mencolok dengan PHP + MySQL dan mengalahkannya dalam skalabilitas dan kecepatan pemrosesan. Hasil ini menunjukkan keunggulan MERN dalam mendukung menu elektronik yang interaktif, skalabel, dan responsif. Meskipun pengukuran didasarkan pada percobaan rata-rata tanpa analisis statistik penuh, pendekatan yang menjanjikan telah ditunjukkan dalam meningkatkan efisiensi dan keandalan di lingkungan restoran.

Kata kunci: Internet of Things, Mikrokontroler, Arduino Mega 2560, MERN, Kelembapan, Wi-Fi.

1. Introduction

Communication is a dynamic and practical method in every institution; in general, it plays a crucial role in all administrative processes and facilitates their performance, which is a common need for every individual in every Political, Social, cultural, Educational, and Other Organization [1]. Beyond this context, in restaurants, two-way communication plays a crucial role in processing orders. Customers communicate their orders to the waiter, who records them on a paper order form. This method is slow and relies on a large number of employees, particularly when many customers place orders simultaneously. Human error and miscommunication often occur when orders are handled in this manner [2]. This issue has led to the development of an IoT-based electronic menu. Most existing IoT and microcontroller-based systems adopt a unidirectional communication model, where information flows one-way, typically from device to server or display, with no return path for user interaction or feedback [9]. This design limits interactivity and responsiveness, making such systems unsuitable for applications that require two-way engagement, such as electronic menus or responsive displays. To manage the menu dynamically, we need a proper user interface (UI) in the server side. The commonly used technology to develop this particular UI is PHP. However, with the rise of web application complexity and performance expectations, the limitations in scalability and less flexible architecture become more apparent [5]. Varunkumar et al. [8] developed a PHP-MySQL-based smart dining system that enhanced order processing, but it lacked two-way interaction and scalability. Addressing these limitations, this study develops a MERN stack-based two-way communication electronic menu using an Arduino Mega 2560 microcontroller.

Two-way communication consists of a client and a server. Client-server is a term used in the world of information technology and communication, used for data exchange over the internet, which is then stored in a database [3]. The server side used MERN tech stack architecture (MongoDB, Express.js, React.js, Node.js) to manage and display data in the form of a user interface [7]. This electronic menu will operate within the scope of a Wi-Fi (Wireless Fidelity) network. The performance of this device will be influenced by the strength of the Wi-Fi signal, which can vary over time due to interventions from within the device itself or from external factors such as concrete barriers or humidity. The difference between the signal strength received before and the new signal strength received will be called the gain.

To determine the value of reinforcement received by the provided device. The calculation of gain value is formulated by equation (1) [4]:

$$G = (PR1 - PR2) + Ga \quad (1)$$

where, $PR1$ is gain value without obstacles (dBm), $PR2$ is gain value with obstacles (dBm), and Ga is gain received by the device.

Based on the above explanation, this study focuses on the design and performance analysis of an electronic menu system developed using the MERN (MongoDB, Express.js, React.js, and Node.js) stack architecture. The system is integrated with an Arduino Mega 2560 microcontroller and an ESP8266 Wi-Fi module, enabling seamless two-way communication between the client interface and the server.

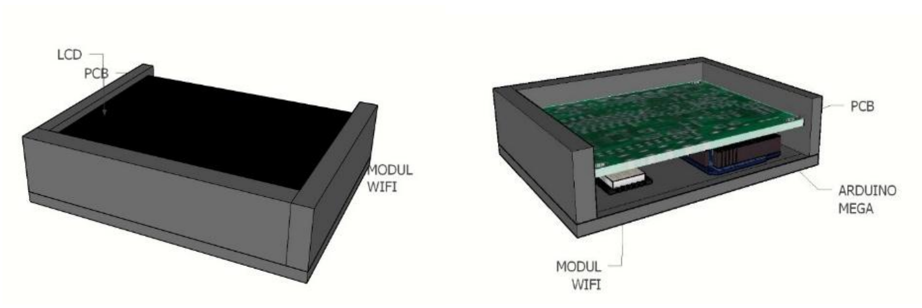
2. Server Development, Device Assembly, and Experimental Setup

2.1. Server Development

In the process of developing this electronic menu system, the first step is to create the backend, where all data processing will be carried out, including data addition, subtraction, and summation. The reason the backend of this device was developed initially is to simplify the development and testing process of the electronic menu. The framework used for the development of this backend itself is Express.js with MongoDB as its database. Furthermore, frontend development is carried out to create a UI (User Interface) that facilitates menu and order management. The framework used to develop this frontend is Next.js, with Tailwind CSS as its styling framework.

2.2. Device Assembly

This electronic menu is assembled using an Arduino Mega 2560 as its microcontroller and an ESP8266 module as the connection between this device and the internet, followed by the addition of DHT22 temperature and humidity sensors that allow this device to measure and record temperature and humidity in its surroundings, and finally, a 3.5-inch TFT (Thin Film Transistor) screen that functions as the input receiver from the user of this device. Figures 1(a) and 1(b) present the external design and internal design of the device, respectively.



(a) (b)

Figure 1. (a) External design of the device and (a) internal design of the device

Arduino Mega is an open-source electronic prototype platform based on flexible and easy-to-use hardware and software. It is aimed at artists, designers, hobbyists, and anyone interested in creating work environments or creative objects [8]. The ESP8266 is a chip featuring a 32-bit microprocessor with a RISC architecture. The microprocessor is a Tensilica Xtensa with RTOS support, operating at frequencies between 80 MHz and 160 MHz. The capability of the 3.5-inch TFT screen, which supports touch input, is the reason for using this component. To understand how the device's logic works, Figure 2 illustrates the device's logic flow.

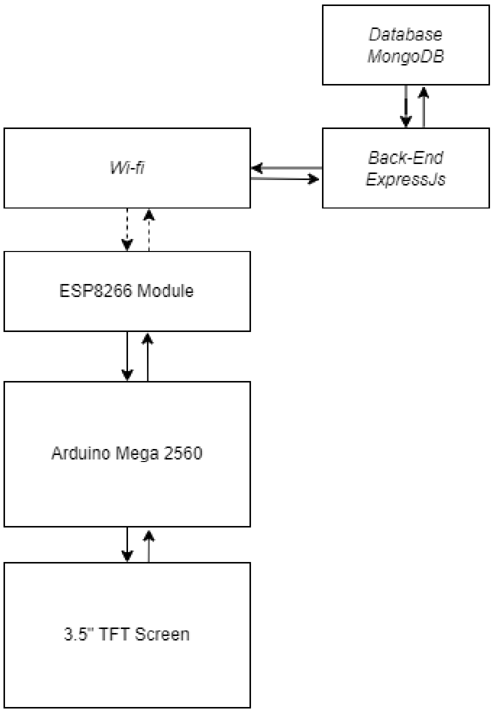


Figure 2. illustrating the bidirectional communication framework between the device and the server.

2.3. Experimental Setup

To evaluate system performance, the assembled electronic menu will be connected to a Wi-Fi network under varying signal strength conditions. Data will be collected on:

- Effect of environmental factors (e.g., humidity, barriers) on Wi-Fi connectivity.
- Accuracy and reliability of order transmission.
- Data processing speed between MERN & PHP + MySQL

The gain value of the Wi-Fi signal will be calculated according to Hanafi (2019) to analyze the impact of obstacles on system reliability.

3. Results and Discussions

All measurements presented in this section were obtained by averaging multiple repeated trials, each performed five times, to reduce random variation and increase reliability. However, further statistical analysis (e.g., variance, standard deviation, or confidence intervals) was not performed, which limits the ability to quantify variability.

3.1. Device Testing

The DHT22 sensor is used to record humidity and temperature around this device. This device will record temperature and humidity along with customer order data received and stored in the server through Express.js.

The testing conducted on this device involved signal strength measurements, which were influenced by several external factors, including concrete barriers and high humidity. For testing with concrete barriers, the distance will be varied while the humidity is constant. Tables 1, 2, and 3 present test results for conditions without barriers, with concrete barriers, and with humidity, respectively.

Table 1. Table of WiFi signal strength measurements on devices without barriers

No	Distance (m)	Signal Quality	RSSI (dBm)
1	5	Very Good	-56
2	10	Very Good	-59
3	15	Good	-65
4	20	Good	-68
5	25	Fairly Good	-70
6	30	Fairly Good	-72

Table 2. Table of WiFi signal strength measurements on devices with concrete barriers

No	**Distance (m)**	**Signal Quality**	**RSSI (dBm)**
1	5	Good	-61
2	10	Good	-64
3	15	Fairly Good	-71
4	20	Fairly Good	-73
5	25	Fairly Good	-75
6	30	Fairly Good	-78

The gain (G) between the transmitter and receiver was evaluated at various distances ranging from 5 to 30 meters using the empirical relation $G = (P_{R1} - P_{R2}) + 1$, where $P_{R1}$and P_{R2}represent the received power levels (in dBm) under two different conditions. At a distance of 5 meters, the received power values were -56 dBm and -61 dBm, yielding a gain of 6 dBm. Similarly, at 10 meters, the respective power readings of -59 dBm and -64 dBm also produced a gain of 6 dBm. For 15 meters, the power difference between -65 dBm and -71 dBm resulted in a gain of 7 dBm. At 20 and 25 meters, the calculated gains were both 6 dBm, corresponding to power readings of -68 dBm and -73 dBm, and -70 dBm and -75 dBm, respectively. Finally, at 30 meters, the gain increased slightly to 7 dBm, derived from -72 dBm and -78 dBm. These results indicate that the signal gain remained relatively stable, with minor fluctuations between 6 and 7 dBm across the tested distance range.

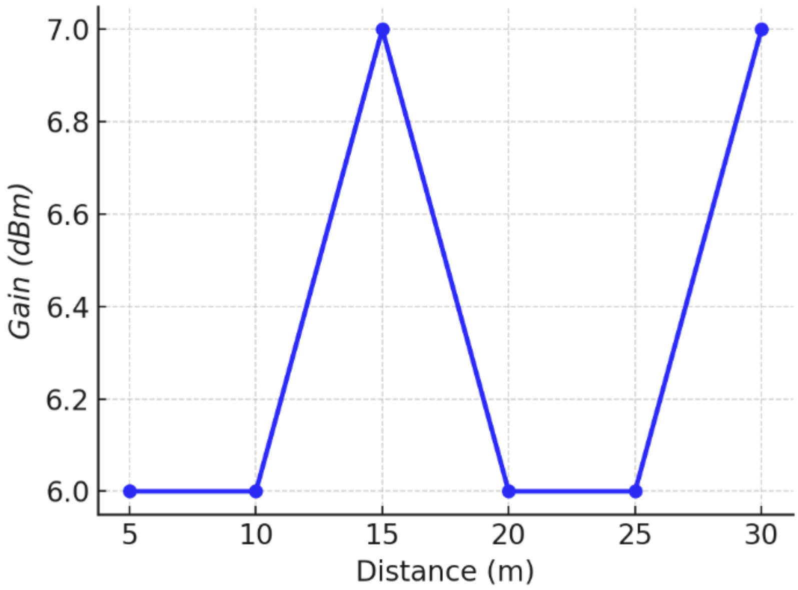


Figure 3. The graph illustrates the gain obtained from signal reception, comparing scenarios with and without concrete barriers.

The calculation results indicate that RSSI will weaken as the distance increases and more barriers are introduced. The obtained gain also fluctuates between 6 and 7 dB, proving that concrete barriers can provide a gain of approximately ±6 dBm.

Table 3. Table of WiFi signal measurements with humidity range 65% - 95%

No	Distance (m)	Humidity (%)	Signal Quality	RSSI (dBm)
1	3	65.40	Very Good	-50
2	3	68.10	Very Good	-51
3	3	74.30	Very Good	-50
4	3	76.70	Very Good	-52
5	3	80.60	Very Good	-54
6	3	83.60	Very Good	-56
7	3	85.80	Very Good	-57
8	3	93.70	Good	-63
9	3	94.50	Good	-62

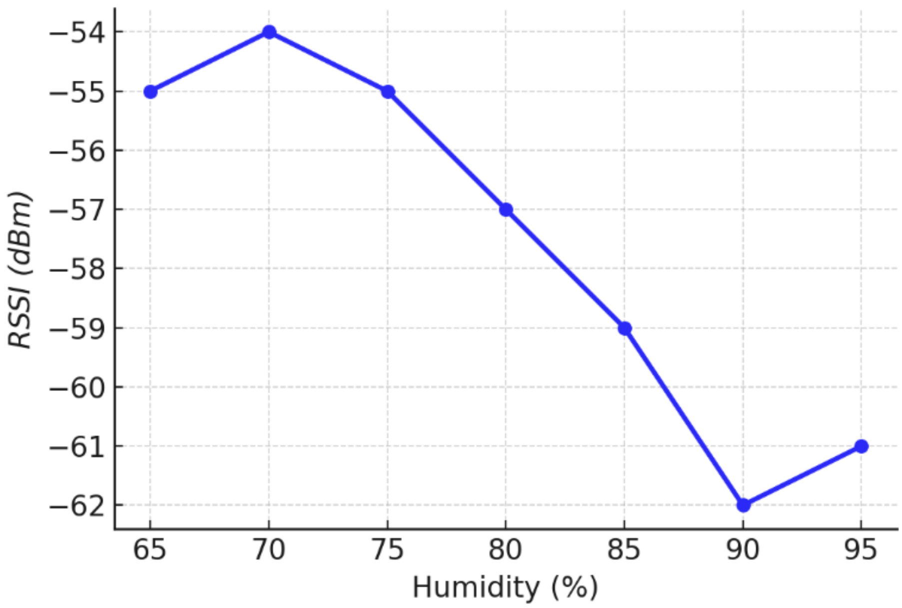


Fig 4. The graph illustrates the comparison of signal reception gain between scenarios with and without concrete barriers.

Tables 1 to 3 show a consistent reduction in signal strength as distance increases, whether in a no-barrier environment or with a barrier present. With no barrier, RSSI values decreased from -56 dBm at 5 m to -72 dBm at 30 m, while with a concrete barrier, the decrease was more pronounced (-61 dBm at 5 m to -78 dBm at 30 m). The calculated gain difference between the two scenarios ranged from 6 to 7 dBm, indicating that concrete shows measurable attenuation across all tested distances.

Humidity testing confirmed that environmental conditions have a significant impact on Wi-Fi performance. At 65% humidity, the RSSI was -50 dBm, at 94.5% humidity, it weakened to -62 dBm, reflecting a 12 dBm reduction. On average, at every ~5% increase in humidity above 75% the RSSI dropped approximately 1-2 dBm. These findings align with prior research demonstrating that the surrounding physical environment, especially humidity, affects the radio signal through absorption, scattering, and reflections [11].

3.2. Web Application Testing

3.2.1. Data Fetching from the server testing

This electronic menu used Express.js as a backend to process data. Testing was conducted using Hypertext Transfer Protocol (HTTP) with the GET method to fetch data from the server. Testing results are presented in Figure 5.

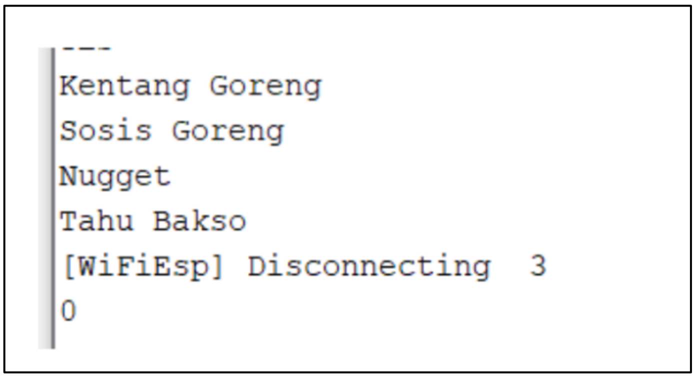


Figure 5. Data fetching result displayed on the serial monitor.

3.2.2. Data Posting from client testing

The data posting test used the same protocol as data fetching (HTTP), but employed the POST method. The request body was sent as data, which was then processed by the backend and presented to the UI.

```
[WiFiEsp] Connecting to 192.168.43.92
POSTING TO SERVER
```

Figure 6. Data posting result displayed on the serial monitor

3.2.3. Data processing and presentation testing

The data posted to the server was later processed and presented through the UI. This UI is used and maintained by the waiters to process customer orders. Waiters can also manage the available menu and prices through this UI. This UI also presents order history, including past orders that were paid or canceled (Figure 7).

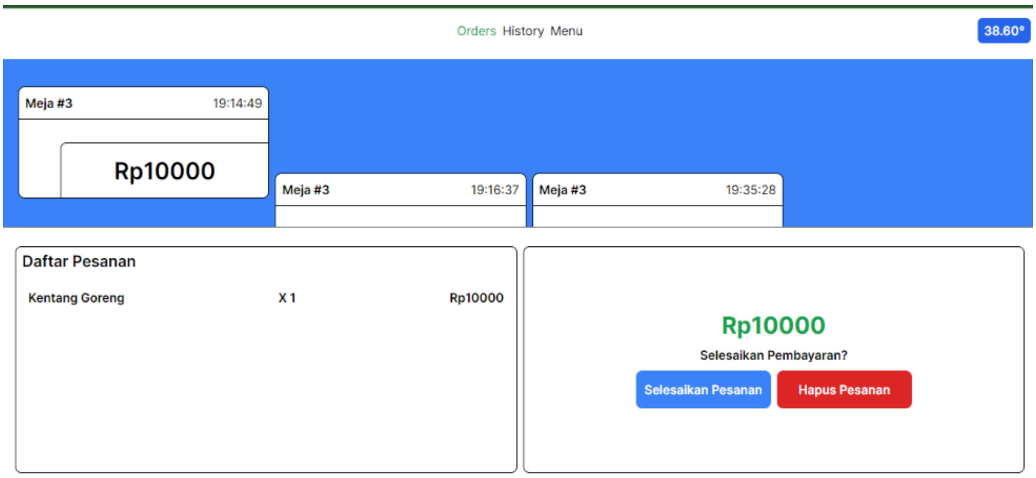


Figure 7. The UI of the MERN Stack-based Electronic Menu Server Website

The data processing speed is also tested by sending and displaying the string “1” 5,000 to 25,000 times using two different technologies: MERN and PHP + MySQL. The testing was conducted using a local server with no internet connection (Table 4).

Table 4. Processing speed comparison of MERN and PHP + MySQL

time (s)		Rows
MERN	PHP + MySQL	
1.68	16.87	5000
2.12	36.02	10000
2.71	59.23	15000
3.22	74.39	20000
3.83	98.76	25000

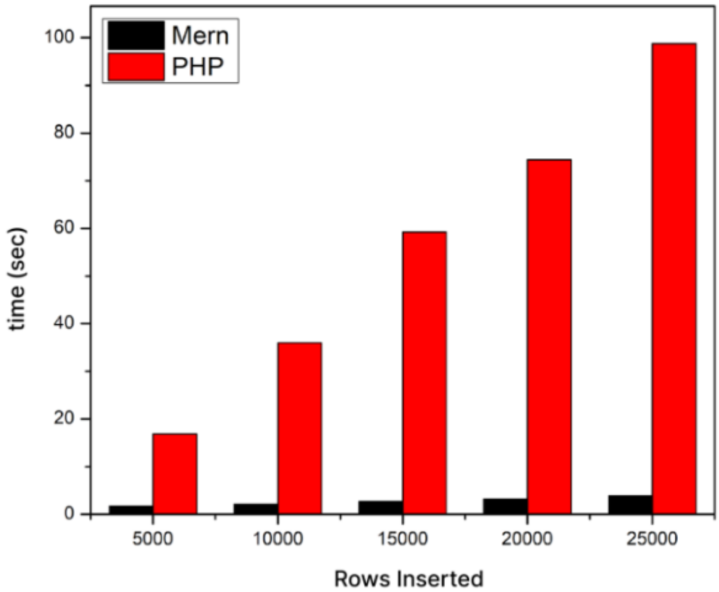


Figure 8. Graph presenting the Processing speed comparison of MERN and PHP + MySQL

The web application testing results (Figures 5 to 6 and Table 4) demonstrate the flow of two-way device communication, starting from sending data to the server, which the server processes and returns feedback to the device. The processing speed testing result, as presented in Figure 8, also demonstrates a striking difference between MERN and PHP+MySQL. The speed and scalability of MERN make it more reliable for handling a large number of orders simultaneously. This flow and processing speed solve the unidirectional communication problem, and the lack of speed also affects the scalability of the PHP + MySQL tech of the previous IoT innovative dining system by Varunkumar et al. [9].

4. Conclusion

This study successfully designed and implemented a MERN stack-based electronic menu system with two-way communication capabilities, addressing the limitation of prior IoT dining systems that relied on unidirectional and PHP+MySQL technologies. Testing results demonstrated that Wi-Fi signal strength decreases linearly with distance and is attenuated by environmental factors, such as concrete or humidity. Concrete introduces a measurable attenuation of approximately 6-7 dBm, while humidity contributes around an 11-12 dBm reduction at high levels. The server-side processing speed shows a striking difference between MERN and PHP+MySQL. MERN outperformed PHP+MySQL, proving to be more scalable and reliable for handling a large number of orders simultaneously. These findings indicate that the system not only enhances order management efficiency but also demonstrates a more interactive and responsive solution for order management applications. The measurements presented in this section were obtained by averaging of multiple repeated trials of 5 times to reduce random variation and increase reliability. However, further statistical analysis (e.g., variance, standard deviation, or confidence intervals) was not performed, which limits the ability to quantify variability. MERN stack-based electronic menu offers a promising approach to improving operational efficiency in restaurant environments, overcoming the scalability and performance constraints of earlier IoT smart dining systems.

References

[1] S. S. Abu Naser, M. J. Al Shobaki, and T. M. Ammar, "Impact of communication and information on the internal control environment in Palestinian universities," *International Journal of Hybrid Information Technology*, vol. 10, no. 11, pp. 41–60, 2017. https://doi.org/10.14257/ijhit.2016.10.11.05

[2] N. A. Pambudi and A. H. Hendrawan, "Digital information board using Raspberry Pi 3 Model B based on Raspbian," *Jurnal Mantik*, vol. 4, no. 4, 2021. https://iocscience.org/ejournal/index.php/mantik

[3] T. Budioko, "Sistem monitoring suhu jarak jauh berbasis Internet of Things menggunakan protokol MQTT," in *Proceeding Seminar Nasional Riset Teknologi Informasi (SRITI 2016)*, vol. 8, pp. 353–358, STMIK AKAKOM Yogyakarta, July 2016.

[4] M. Hanafi, F. Imansyah, and D. Suryadi, "Analisa simulasi pengaruh uji kuat sinyal Wi-Fi dari bahan–bahan obstacle," *Jurnal Teknik Elektro*, vol. 1, no. 1, 2019.
[5] A. S. Ingle, A. N. Ade, P. M. Chandane, D. B. Bhagat, and V. R. Dolase, "IoT based menu ordering system," *International Journal for Research in Applied Science and Engineering Technology (IJRASET)*, vol. 10, no. 11, pp. 1194–1199, 2022.
[6] G. O. Prajapati, O. G. Mahangare, A. R. Shaikh, R. S. Thite, M. Patil, and G. Narang, "Boosting scalability and performance: Transitioning from PHP to the dynamic MERN stack," *International Journal for Research in Applied Science and Engineering Technology (IJRASET)*, vol. 13, no. 2, pp. 812–819, 2025. https://doi.org/10.22214/ijraset.2025.66969
[7] S. Raju, S. Soundararajan, and V. Loganathan, *MERN Stack Web Application*, vol. 25, 2021. http://annalsofrscb.ro
[8] G. S. Sharath, N. Hiremath, and G. Manjunatha, "Design and analysis of gantry robot for pick and place mechanism with Arduino Mega 2560 microcontroller and processed using Python," *Materials Today: Proceedings*, vol. 45, pp. 377–384, 2020. https://doi.org/10.1016/j.matpr.2020.11.965
[9] B. Varunkumar, B. Jothish, M. Pavithran, and R. Rajesh, "Smart dining and smart food ordering system," *International Journal for Research in Applied Science and Engineering Technology (IJRASET)*, vol. 11, no. 5, pp. 1431–1435, 2023. https://doi.org/10.22214/ijraset.2023.55253
[10] D. Wang, Y. Yao, J. Wang, and H. Xu, "Unidirectional communication paradigms in IoT security: A survey," *Sensors*, vol. 24, no. 12, p. 3985, 2024. https://doi.org/10.3390/s24123985
[11] X. Zhang, L. Wang, X. Zhang, and Y. Wang, "WiHumidity: WiFi-based humidity measurement with commercial devices," in *Wireless Algorithms, Systems, and Applications (WASA 2017)*, K. Zheng, M. Li, and H. Jiang, Eds., *Lecture Notes in Computer Science*, vol. 10251, pp. 665–676. Springer, 2017. https://doi.org/10.1007/978-3-319-52015-5_55