INTERNATIONAL JOURNAL FOR INNOVATIVE RESEARCH IN MULTIDISCIPLINARY FIELD
ISSN(O): 2455-0620 [Impact Factor: 9.47]
Monthly, Peer-Reviewed, Refereed, Indexed Journal with IC Value : 86.87
Volume - 11, Issue - 09, September - 2025

DOIs:10.2015/IJIRMF/202509030 --:-- Research Paper / Article / Review

Design and Implementation of an ESP32-Based Bluetooth-Enabled Automatic Bell System with RTC Synchronization

[1] Dr. Kakade Sandeep , [2] Mr. Kotalwar Girish , [3] Miss. Kulkarni Saroja, [4] Mr. Kunal Ardad

[1]Assistant professor, Department of Electronics & Telecommunication Engineering, Vilasrao Deshmukh foundation Group of Institutions, Latur, India
[2]Final Year Student, Department of Electronics & Telecommunication Engineering, Vilasrao Deshmukh foundation Group of Institutions, Latur, India
[3] Final Year Student, Department of Electronics & Telecommunication Engineering , Vilasrao Deshmukh foundation Group of Institutions, Latur, India
[4] Final Year Student, Department of Electronics & Telecommunication Engineering , Vilasrao Deshmukh foundation Group of Institutions, Latur, India
Email - [1] kakadesandeep2000@gmail.com , [2] kotalwargirish7@gmail.com , [3] kulkarnisaroja0710@gmail.com , [4] kunalardad2003@gmail.com

Abstract: *Time management plays a vital role in maintaining discipline and efficiency in schools, colleges, and organizations. Conventional bell systems often rely on manual operation or mechanical timers, which are prone to human error, inconsistency, and limited flexibility. To address these challenges, this paper presents an ESP32-based automatic bell system controlled through a Bluetooth-enabled mobile application. The system integrates a Real-Time Clock (RTC) module for accurate timekeeping, a relay driver circuit to operate the bell, and a dot matrix display for real-time status updates. Users can easily configure or modify schedules using a dedicated mobile interface, eliminating the need for technical reprogramming or continuous Internet access. The proposed design was implemented and tested under real-time conditions, demonstrating precise synchronization, reliable wireless communication within a range of 8–10 meters, and consistent bell operation with negligible delay. Comparative evaluation with existing systems shows that the developed solution offers a cost-effective, user-friendly, and offline-capable alternative suitable for educational institutions and small-scale organizations. The system can be extended to larger setups with multi-bell control, centralized scheduling and optional IoT connectivity.*

Key Words: *Automatic bell system, ESP32 micro Controller, Bluetooth communication, Real-Time Clock (RTC), Wireless scheduling.*

1. INTRODUCTION:

Time management is one of the most critical aspects of educational and organizational environments. Institutions such as schools, colleges, industries, and offices rely heavily on scheduled events to maintain discipline, productivity, and efficiency. Traditionally, bells or alarms are operated manually or through mechanical timers, which often results in inconsistency, delays, and human error. Such systems lack the flexibility to adapt to sudden changes in schedules and require constant manual supervision, leading to inefficiencies in day-to-day operations. The rapid advancement of embedded systems and wireless communication technologies has enabled significant innovation in automating these processes. Microcontroller-based automatic bell systems have gained wide attention for their ability to execute accurate, reliable, and programmable schedules without continuous human intervention. Real-time clock (RTC) modules are frequently employed in such systems to maintain timing accuracy, even in the event of power failures. Several works have demonstrated the feasibility of using microcontrollers integrated with RTC modules to eliminate manual ringing while ensuring precise timing [1]–[3].

Wireless connectivity has further improved automation by allowing remote configuration and flexibility in setting or modifying schedules. For instance, many systems use Wi-Fi to synchronize time and provide remote control through web-based platforms [1]. Others integrate Bluetooth modules to enable local communication between a controller and a mobile device [3]. These wireless approaches reduce manual reprogramming, minimize errors, and allow schedule adjustments in real time. In addition, the use of microcontrollers such as Arduino and ESP32 has provided compact, cost-effective, and energy-efficient platforms for automation [2], [4].

Despite these advancements, challenges remain in existing designs. Many solutions are heavily dependent on WiFi or Internet connectivity for synchronization and remote access [1], [2]. This dependence creates issues in areas with unreliable networks, where the system may fail to function properly. Other systems, though effective, often require technical reprogramming whenever schedule changes are needed, limiting their usability for non-technical users [3]. Some implementations also increase hardware complexity and cost by adding display units, multiple communication modules, or Internet-based services, which are not always necessary for small or medium-sized institutions [4], [5].

There is a growing need for an automatic bell system that combines reliability, affordability, offline functionality, and user-friendly design. To address these gaps, the proposed work introduces an Automatic Bell System operated through a mobile application over Bluetooth, controlled by ESP32 microcontroller hardware**.** The system eliminates dependency on Wi-Fi or continuous Internet access, ensuring that core functionality remains intact even in low-connectivity environments.

The mobile app, provides an intuitive interface where schedules can be created, updated, or modified in real time. Once configured, the ESP32 executes the alarm schedule using its built-in Bluetooth communication and RTC synchronization, activating the bell at precise times without manual intervention.

The design objectives of this system are fivefold:

1. To achieve accurate and reliable bell ringing schedules using RTC support.
2. To allow real-time schedule modifications directly from a smartphone application.
3. To reduce reliance on Wi-Fi/Internet connectivity by focusing on Bluetooth communication.
4. To provide a cost-effective, low-complexity solution suitable for schools, colleges, and institutions with limited infrastructure.
5. To deliver a user-friendly mobile interface that can be easily operated by non-technical staff.

In addition to advancements in automation, developments in wireless communication technologies also play a crucial role in supporting systems like the proposed automatic bell. Research in antenna design for next-generation communication systems has shown how optimized wireless components improve reliability, data transfer, and connectivity for embedded devices [7]. The significance of this approach lies in its scalability and adaptability. While the system is designed primarily for educational environments, it can be extended to industries, healthcare centers, and office spaces where accurate scheduling and audible notifications are necessary. By combining ESP32 with Bluetooth and a dedicated mobile application, this system offers a balance of simplicity, affordability, and high functionality that addresses the shortcomings of earlier works.

2. LITERATURE REVIEW:

Automation of bell systems has been widely investigated across academic and industrial contexts to address the shortcomings of manual ringing. A notable contribution presented a microcontroller-based bell system developed for the College of Engineering, DMMMSU-MLUC, where an embedded controller was programmed to manage class schedules automatically. The design reduced the workload of faculty, ensured accurate ringing, and demonstrated how low-cost microcontroller solutions can improve institutional time management [8]. Similarly, earlier research explored a Lab VIEW-based approach in which both automatic and manual modes of operation were integrated for boarding and day schools. The system enabled remote control of bell operations, offering flexibility through software-driven interfaces, but required a computer platform for execution, making deployment more resource-intensive compared to hardware-only solutions [9]. Advancements in microcontrollers and communication technologies have further enhanced the reliability of automatic bell systems. A recent design implemented an I^2C-based real-time clock with ESP32 to achieve precise schedule control, while also incorporating mobile and web interfaces for dynamic updates. This design not only ensured accurate timekeeping but also addressed issues of drift and synchronization typically found in simpler systems [10]. In parallel, researchers have developed wireless bell control systems based on microcontrollers, leveraging RF transmission for triggering alarms at different nodes. Such systems demonstrated scalability and effective communication over moderate distances but were limited in terms of integration with modern mobile or IoT platforms [11]. The evolution of automation has also seen integration with IoT-enabled solutions. A recent study introduced a smart college clapper system that combined bell automation with faculty leave management. By embedding IoT features, the system allowed administrators to synchronize class schedules, manage notifications, and monitor faculty availability through a unified platform. This broadened the scope of traditional bell systems, transforming them into multi-functional institutional management tools [12].

Limitations

Although the reviewed studies highlight significant progress in automated bell systems, several challenges persist. Systems that rely on Lab VIEW or RF-based control [9], [11] require specialized hardware or software, raising costs and limiting deployment in resource-constrained environments. IoT-based designs [12], while innovative, often depend heavily on continuous Internet connectivity, which may not be reliable in all institutional settings. Some microcontroller-based approaches [8], [10] are cost-effective and accurate but provide limited user-friendly interfaces for non-technical operators, as scheduling often requires manual programming or technical expertise.

Summary

Overall, the literature reflects a steady progression from simple microcontroller-based systems to advanced IoT-enabled solutions. Early designs demonstrated accuracy and reduced human intervention, while more recent works integrated real-time clocks, wireless communication, and mobile/web interfaces for improved flexibility. However, there remains a need for solutions that combine offline reliability, cost-effectiveness, and intuitive mobile-based configuration without relying on continuous Internet. The proposed ESP32-based automatic bell system with Bluetooth-enabled mobile app control directly addresses these gaps by providing accuracy, real-time flexibility, and user accessibility in a low-cost, network-independent design.

3. RESEARCH METHODOLOGY:

The proposed automatic bell system was designed and implemented using a combination of microcontroller hardware, peripheral modules, and a mobile application interface. The methodology is divided into three stages: system architecture design, hardware development, software implementation.

System Architecture

The system architecture is centered on the ESP32 microcontroller, which functions as the processing unit for all operations. **A** Real-Time Clock (RTC) module provides continuous timekeeping to ensure accurate scheduling, while the dot matrix display is used to show real-time information such as the current time and active schedule. A **5V** regulated power supply was designed to provide stable voltage to the ESP32, RTC, display, and relay driver circuits. The system is connected to the user's mobile device via the ESP32's integrated Bluetooth module, enabling wireless communication for scheduling and control. The actual connection diagram (Figure 1) illustrates the integration of these components. The ESP32 communicates with the RTC through the I²C protocol to fetch real-time data. The dot matrix display is interfaced with the ESP32 using SPI for displaying current time and system status. The relay driver circuit is connected to the ESP32 GPIO pins, which in turn drives the bell mechanism. The regulated 5V supply powers all components, ensuring uninterrupted operation.

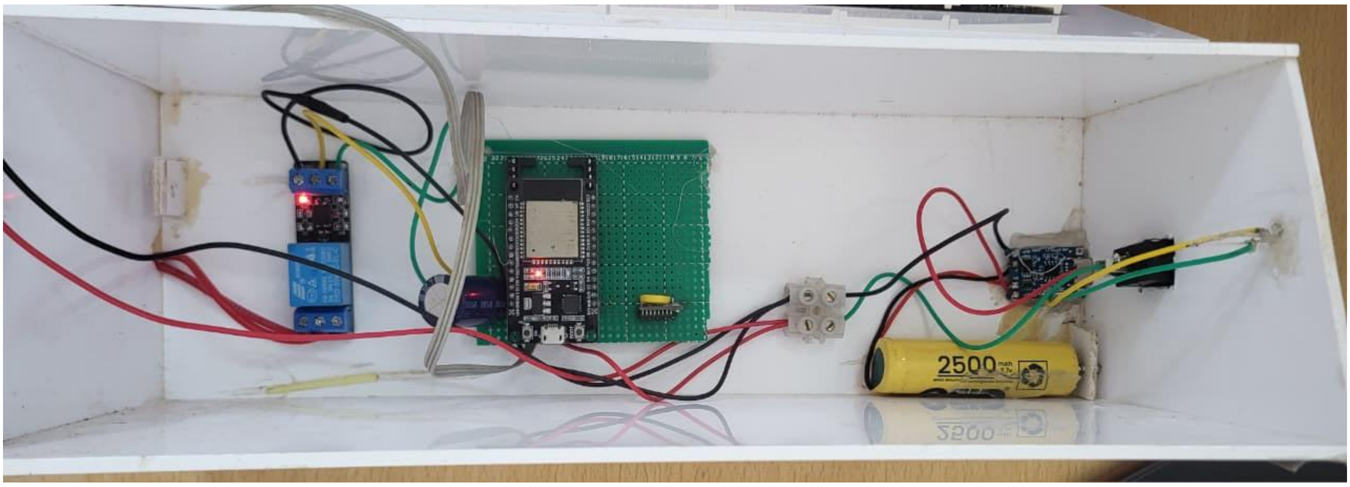

Figure 1. Circuit connection diagram of ESP32-based automatic bell system

Hardware Components

The hardware of the proposed automatic bell system primarily consists of the ESP32 microcontroller, which serves as the central processing unit with built-in Bluetooth functionality for wireless connectivity. An RTC module is integrated through the I²C protocol to maintain accurate and continuous timekeeping, even during power interruptions. A dot matrix display is interfaced with the ESP32 to provide real-time visual output, including clock information and schedule status. To ensure stable and uninterrupted operation of all modules, a regulated 5V power supply was designed using rectification, filtering, and voltage regulation stages. The bell mechanism is driven through a relay driver circuit, where the relay acts as an electronic switch controlled by the ESP32 to activate or deactivate the bell at scheduled intervals. Together, these hardware components form a compact, reliable, and cost-effective system suitable for institutional automation.

Software Implementation

The mobile application allows users to configure schedules and transmit them to the ESP32 over Bluetooth. The ESP32 firmware, developed in Arduino IDE, continuously monitors the RTC. When the current time matches the scheduled time stored in memory, the ESP32 triggers the relay driver to ring the bell. Concurrently, the dot matrix display updates the time and status.

Workflow

Figure 2 represents the step-by-step process followed by the design. The operation starts when the user configures or updates a schedule through the mobile application, which communicates the timing data to the ESP32 microcontroller over Bluetooth. The ESP32 stores this information in its memory and continuously synchronizes it with the Real-Time Clock (RTC) module to ensure precise timekeeping.

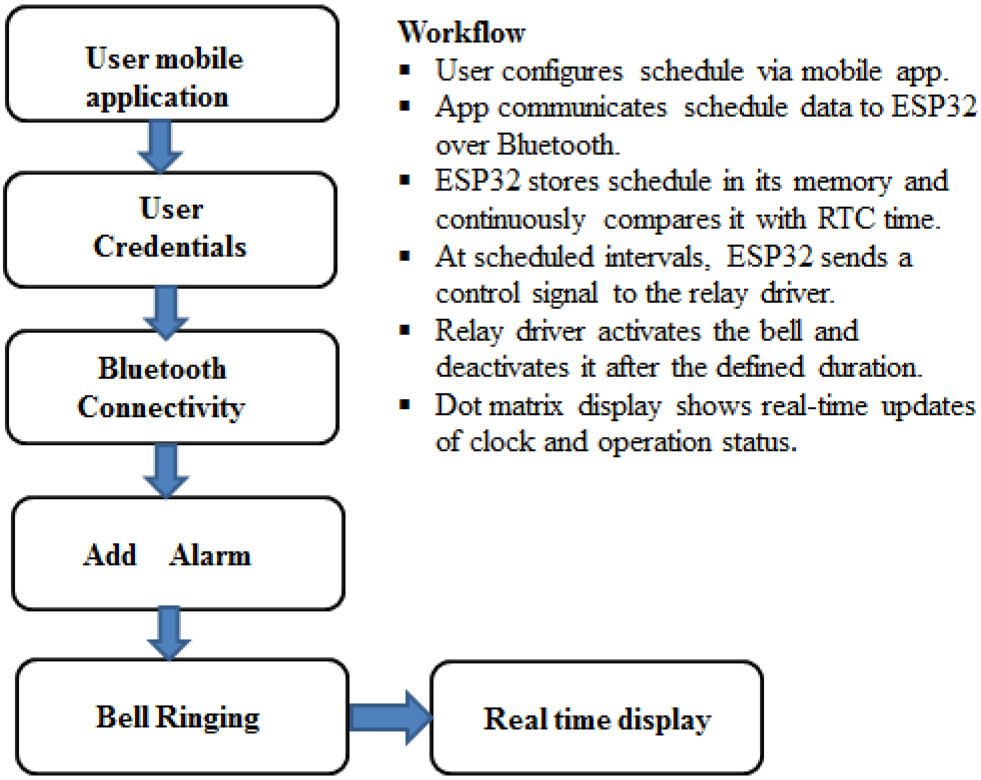

Figure 2. Workflow of ESP32-based automatic bell system with Bluetooth scheduling and RTC synchronization

At the scheduled interval, the ESP32 generates a control signal and sends it to the relay driver circuit, which activates the bell mechanism for a predefined duration before automatically switching it off. In parallel, the dot matrix display provides continuous real-time feedback of the current clock and system operation, allowing users to visually monitor the process. This workflow ensures seamless coordination between user input, microcontroller processing, and hardware actuation, making the system reliable, user-friendly, and efficient in real-world applications.

4. RESULTS:

The proposed ESP32-based automatic bell system was successfully designed, implemented, and tested in a real-time environment. The experimental setup consisted of the ESP32 controller, RTC module, dot matrix display, relay-driven bell, and a mobile application interface for schedule programming.

System Functionality

The system was evaluated under different operating conditions to verify its accuracy and usability. The mobile application provided two login modes – *User* and *Admin* – to ensure secure access (Figure 3). Once connected via Bluetooth, the administrator could configure the ringing schedule by entering the desired hours and minutes (Figure 4). The schedule was then transmitted wirelessly to the ESP32, which synchronized it with the RTC module.

Display Output

The dot matrix display provided continuous real-time information about system status. During initialization, the display scrolled text such as *"BELL SYSTEM"* to confirm successful startup (Figure 5). When the scheduled time was transmitted from the mobile application, the display updated with the corresponding clock value, for example *"TIME: 14:07"* (Figure 6). This visual confirmation eliminated ambiguity and allowed users to cross-check the stored schedule.

Bell Operation

At the programmed time, the ESP32 generated a control signal to activate the relay, which successfully triggered the connected 230V AC bell. The bell rang for the predefined duration and automatically turned off, ensuring energy efficiency and preventing continuous ringing. The ringing was synchronized precisely with the RTC time, with negligible delay (less than one second).

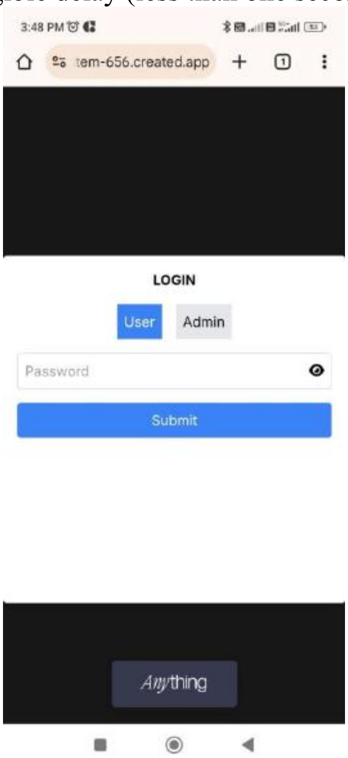

Figure 3: Login interface of mobile application for automatic bell system

Figure 4: Schedule configuration interface of mobile application connected to ESP32

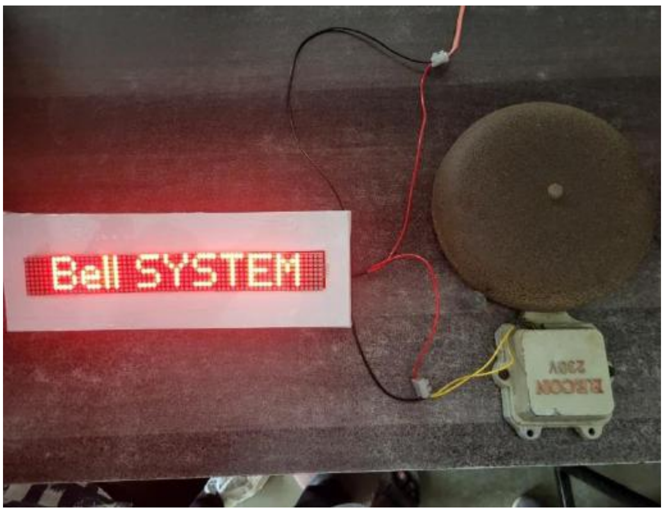

Figure 5: Dot matrix display showing system initialization message

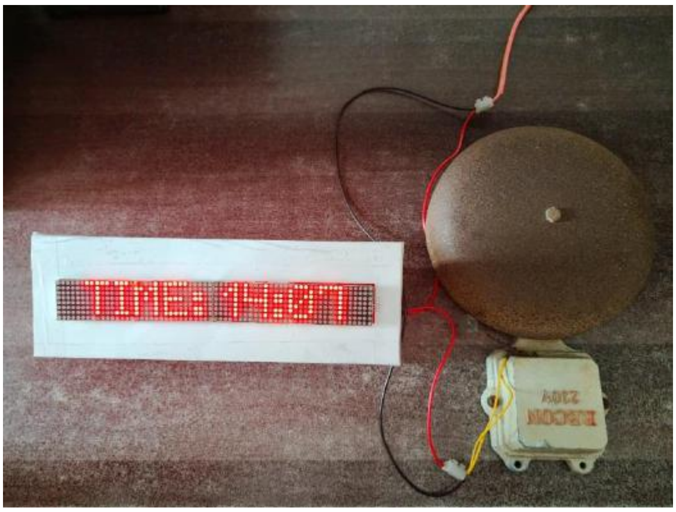

Figure 6: Dot matrix display showing real-time clock update with scheduled time

The system was thoroughly tested under multiple operating scenarios to validate its reliability and accuracy. During trials, the bell consistently rang at the exact scheduled time, with an observed delay of less than one second, demonstrating precise synchronization between the ESP32 and the RTC module. Bluetooth connectivity between the mobile application and the ESP32 remained stable within a range of 8–10 meters, which is adequate for typical classroom, laboratory, and office environments. The dot matrix display provided clear visibility of the initialization messages, system status, and real-time clock updates, even under ambient lighting conditions, allowing users to easily verify the programmed schedules. Importantly, the system functioned seamlessly without reliance on Wi-Fi or Internet services, making it highly suitable for institutions in semi-urban or rural areas where network infrastructure may be inconsistent. These evaluations confirm that the proposed automatic bell system achieves its design objectives of accuracy, reliability, user-friendliness, and offline operation.

5. DISCUSSION:

The experimental validation of the ESP32-based automatic bell system highlights several advantages over existing approaches. First, the integration of an RTC module ensures accurate and drift-free timekeeping, enabling precise ringing even during extended operation. The use of Bluetooth communication offers a practical alternative to Wi-Fi or Internet-based systems, eliminating reliance on network availability while still allowing wireless schedule configuration. This makes the system highly suitable for small- to medium-sized institutions, particularly in regions where Internet connectivity is either unreliable or unavailable. Another significant strength of the proposed design is its user-friendly mobile application interface. Unlike traditional microcontroller-based solutions that require manual programming or technical expertise, the developed app provides a simple platform for real-time schedule updates. This lowers the barrier for non-technical staff to operate the system, improving usability and adoption. The inclusion of a dot matrix display adds further value by providing immediate visual feedback regarding system status and scheduled timings, which minimizes the chances of human error. In comparison with existing systems, several clear benefits are observed. Computer- or LabVIEW-based designs, though functional, require additional hardware platforms, thereby increasing cost and complexity. IoT-based implementations extend flexibility but introduce heavy dependence on continuous Internet, which may not be feasible in all settings. RF-based wireless systems, while low-cost, often lack mobile app support and offer limited scalability. The proposed ESP32-based solution bridges these gaps by combining offline reliability, mobile app flexibility, and cost-effectiveness in a single design. Despite these strengths, some limitations remain. The current system is limited to single-bell control, which may not be sufficient for large campuses requiring multiple synchronized outputs. The Bluetooth communication range, though adequate for classroom-level operation, is inherently restricted compared to Wi-Fi or Internet-enabled systems. Additionally, schedule storage capacity is constrained by the onboard memory of the ESP32, which may limit the number of events that can be programmed simultaneously. Future enhancements could address these challenges. The integration of Wi-Fi or IoT features as optional modules would expand scalability while retaining offline functionality as a fallback. Cloud-based scheduling and multi-device synchronization could allow centralized control across larger institutions. Energy-efficient designs, such as incorporating solar-powered systems, could further reduce operating costs and enhance sustainability.
Overall, the discussion confirms that the proposed system provides a practical balance between simplicity, affordability, and functionality, making it a reliable solution for automating institutional time management.

6. CONCLUSION:

The proposed ESP32-based automatic bell system with Bluetooth-enabled mobile application control has been successfully designed, implemented, and tested. The system eliminates the need for manual intervention by accurately ringing the bell according to pre-programmed schedules synchronized with the RTC module. The integration of a user-friendly mobile interface allows real-time configuration without requiring technical expertise, while the use of Bluetooth communication ensures reliable operation without dependence on Internet connectivity. Experimental results confirmed that the system provides precise timing, stable wireless communication, and clear visual feedback through the dot matrix display. Compared with earlier solutions, the proposed design offers a cost-effective, low-complexity, and network-independent approach, making it especially suitable for schools, colleges, and institutions in areas with limited infrastructure. Looking ahead, the system can be further enhanced by integrating additional features such as multi-bell control, centralized scheduling for large campuses, and optional IoT connectivity for institutions with reliable networks. These improvements would increase scalability while retaining the core benefits of accuracy, affordability, and offline usability. Overall, the developed system demonstrates a practical and efficient solution for institutional time management through automation.

REFERENCES:

1. Mahmood, R. Z. (2022). High Accurate Automatic School Bell Controller Based On ESP-32 Wi-Fi. Technium, 4(1), 126-144.
2. Achari, v. T., kumar, s., & prashanth, k. (2024). Automatic college bell ringing system using rtc. Material science, 23(04).
3. Wohiduzzaman, K., Ullah, R., & Sharkar, A. (2024). Design and Development of a Low Cost Smart Electronic School Bell. *International Journal of Latest Technology in Engineering, Management & Applied Science*, *13*(10), 7-11.
4. Chaudhari, R. H., Rajgor, B. R., Modi, V. M., Patel, V., Chaudhari, D., & Gora, A. (2019). Design and Development of Microcontroller based Solar Powered College Bell. *Int. J. Curr. Microbiol. App. Sci*, *8*(4), 1326-1333.
5. Nalini, N. R., & Sharwanjana, S. K. V. (2020). IoT Based Wireless Automated Bell Ringing System In An.
6. Ayson, R. (2021). Microcontroller-based school bell system for the College of Engineering, DMMMSU-MLUC. *DMMMSU Research and Extension Journal*, *5*, 47-66.
7. Sandeep, K., Sharma, N., & Narawade, N. (2025). High-isolation dual-band slotted patch MIMO antenna for sub-6 GHz 5G applications. *International Journal of Advanced Technology and Engineering Exploration*, *12*(123), 301.
8. Ayson, R. (2021). Microcontroller-based school bell system for the College of Engineering, DMMMSU-MLUC. DMMMSU Research and Extension Journal, 5, 47-66.
9. Habiyaremye, J. (2013). Labview Based Remote Controlled Automatic & Manual Bell For Boarding and Day Schools. In *The International Journal Of Engineering And Science (IJES)* (Vol. 2, No. 7, pp. 96-102). Department of Electronics and Instrumentation, Bharathiar University.
10. Khan, M. H., Imran, M., & Ali, S. A. U. (2025). I2C Based Real Time-Clock College Bell Automation. *Indian Journal of Electronics and Communication Engineering (INDJECE)*, *2*(1), 01-07.
11. Zou, Z., Bie, Y., Zhao, Z., & Zhou, M. (2017, December). Design of wireless bell control system based on microcontroller. In *2017 IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)* (pp. 1403-1406). IEEE.
12. Ghosh, A., Ghosh, A. K., & Chowdhury, P. (2024, December). IoT-Enabled Smart College Clapper and Faculty Leave Management System. In *2024 IEEE Calcutta Conference (CALCON)* (pp. 1-5). IEEE.