iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** III **Month of publication:** March 2026

DOI: https://doi.org/10.22214/ijraset.2026.78538

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue III Mar 2026- Available at www.ijraset.com

A Smart IoT-Enabled Task Scheduling Framework with Integrated Wall Clock for Real-Time Alerts & Efficient Time Management

Minu Sunitha Mary S[1], Nivethitha. E[2], Monisha M[3], Jesica J[4], Nargese Banu. S[5]
Department of CSE, School of Engineering, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore 108

Abstract: ***Effective time management is essential in daily life, yet existing task scheduling systems often fail to provide reliable real-time reminders. This paper proposes an IoT-based Personal Task Scheduler integrated with a Wall Clock, which synchronizes tasks from a mobile application to a digital clock using wireless communication. The system delivers timely alerts through buzzer sounds, LED indicators, and mobile notifications, ensuring improved task adherence. By combining digital scheduling with physical notification mechanisms, the proposed system enhances productivity and reduces the chances of missed tasks. Future enhancements include AI-based intelligent scheduling for adaptive and efficient task management.***
Keywords: ***IoT, Task Scheduler, Smart Clock, Real-Time Notification, ESP32, Mobile Application, Time Management***

I. INTRODUCTION

Time management plays a crucial role in enhancing productivity and maintaining a well-organized daily routine. In today's fast-paced environment, individuals often find it challenging to complete tasks on time due to distractions, increasing workload, and limitations in existing reminder systems. Traditional task management approaches, such as handwritten notes, basic alarms, or mobile applications, largely depend on continuous user attention and interaction. As a result, important reminders are often missed, especially when users are occupied with other activities or overwhelmed by a large number of smartphone notifications.

With the rapid advancement of technology, task scheduling systems have evolved into digital platforms that offer automated reminders and alerts. However, most of these systems are still limited to software-based notifications, which can be easily ignored or unintentionally dismissed. The growing dependence on smartphones has introduced issues such as notification fatigue, where users tend to overlook important alerts due to the excessive volume of notifications from multiple applications. This limitation emphasizes the need for a more reliable and efficient task reminder system that extends beyond traditional digital notifications. The emergence of the Internet of Things (IoT) has enabled the seamless integration of physical devices with digital systems, creating opportunities for more intelligent and interactive solutions. IoT-based systems support real-time communication between devices, making it possible to develop applications that provide both digital and physical feedback. By incorporating hardware components such as sensors, microcontrollers, and display modules, these systems can enhance user experience and significantly improve the reliability of notifications. In this context, the proposed system presents a Personal Task Scheduler integrated with a Wall Clock, combining mobile-based task management with physical alert mechanisms. The system enables users to add and manage tasks through a mobile application, which are then synchronized with a smart clock using wireless technologies such as Wi-Fi or Bluetooth. This integration ensures that tasks are continuously updated and monitored in real time. When the scheduled time arrives, the system generates multiple types of alerts, including buzzer sounds, LED signals, and mobile notifications. This multi-modal notification approach makes sure that users receive reminders even when they are not actively using their smartphones. The addition of physical alerts improves visibility and reduces the chances of missing tasks, making the system more reliable than traditional applications. The proposed system is built using components such as ESP32 for communication, Real-Time Clock (RTC) modules for accurate time tracking, and Android-based mobile interfaces for user interaction. These components work together to deliver a smooth and efficient task management experience. The system architecture ensures reliability, scalability, and ease of use, making it suitable for both personal and professional environments. Moreover, this research aims to bridge the gap between conventional task scheduling methods and modern IoT-based solutions. By integrating both hardware and software components, the system offers a more complete and user-friendly approach to managing time. It not only helps users stay on track with their tasks but also improves productivity by ensuring activities are completed on time.

©IJRASET: All Rights are Reserved

In addition, the proposed system creates opportunities for future improvements such as AI-based smart scheduling, personalized task suggestions, and adaptive notification systems based on user behavior. These enhancements can further increase the efficiency and usability of the system, making it even more effective in smart living environments.

Overall, the IoT-based Personal Task Scheduler integrated with a Wall Clock provides an innovative solution to address the limitations of existing task management systems by offering reliable, real-time, and context-aware notifications.

II. LITERATURE SURVEY

Task scheduling and time management systems have been widely researched in recent years, with a strong focus on improving productivity, automation, and real-time task handling. Different approaches have been developed, ranging from basic digital to-do list applications to advanced IoT-based and intelligent scheduling systems. However, many of the existing solutions still face challenges in terms of real-time response, physical interaction, and user engagement.

Fellmann (2020) introduced an intelligent personal task and time management system designed to improve productivity through structured task planning and monitoring. The study highlighted the role of smart to-do lists, reminders, and tracking features in enhancing daily task completion. Although the system helped users stay organized, it mainly depended on software-based notifications and did not include integration with physical devices. Because of this, reminders were less effective, especially when users were not actively using their devices.

Zhou et al. (2020) investigated the use of Deep Reinforcement Learning (DRL) for task scheduling in Space-Air-Ground Integrated Networks (SAGIN). Their work aimed at improving scheduling efficiency and minimizing delays in complex network systems. While the approach showed strong results in computational scheduling and resource management, it was mainly developed for large-scale network environments and is not directly suitable for personal task management. In addition, the system did not focus on user-centered notifications or real-time physical alerts.

Wu et al. (2020) proposed a data age-aware scheduling method for Industrial Internet of Things (IIoT) systems. The study focused on energy efficiency and timely task execution in industrial setups. By considering the freshness of data, the system was able to improve scheduling decisions and overall performance. However, this approach was mainly intended for industrial automation and did not include user-level interaction or multi-mode notification features.

Bommisetty and Venkatesh (2021) developed a policy gradient-based scheduling method for Time Slotted Channel Hopping (TSCH) networks in IoT environments. Their research enhanced adaptability and efficiency in changing network conditions by optimizing communication schedules. Even though it was effective at the network level, the system was not designed for personal task management and lacked practical implementation for real-time user notifications.

In addition to these studies, most current task scheduling systems are available as mobile or web applications that rely heavily on user interaction. These systems usually provide notifications through pop-ups or alerts on smartphones. However, due to the large number of applications and notifications, users often experience notification fatigue, which leads to missed reminders and reduced effectiveness. Moreover, these systems do not include physical alert methods such as sound or visual indicators outside the device, which limits their usefulness in real-life situations. Recent developments in IoT technologies have made it possible to build smart systems that combine hardware and software for better performance. IoT-based scheduling systems can support real-time communication, automation, and improved reliability. However, many existing IoT solutions are designed for industrial or large-scale applications and often involve complex designs, higher costs, and maintenance difficulties, making them less practical for everyday personal use. Therefore, the proposed system addresses the gap identified in the existing research by providing a simple, cost-effective, and user-focused solution for real-time task management. It connects traditional software-based schedulers with modern IoT systems by including physical alert mechanisms and real-time synchronization, ultimately improving task completion and overall productivity.

III. SYSTEM MODULES AND ARCHITECTURE

The Personal Task Scheduler Integrated with Wall Clock is designed as a modular system made up of six interconnected components. Each module works together to convert user-defined tasks into real-time alerts and intelligent scheduling support, ensuring effective time management and improved productivity.

A. User Interaction and Task Input Module

The User Interaction Module serves as the main entry point of the system, allowing users to create and manage tasks in a simple and organized way. A major design focus of this module is usability, ensuring that even non-technical users can use the system easily.

Through a mobile application or graphical user interface, users can enter task details such as task name, execution time, and optional priority levels. The interface is designed to be clean and minimal, reducing complexity while maintaining functionality. Input validation features are included to prevent incorrect or incomplete entries, thereby maintaining data accuracy.

In addition to task creation, the module offers real-time visibility of scheduled activities. Users can view pending, completed, and upcoming tasks in an organized format. This continuous feedback helps users stay aware and manage their time more effectively.

By focusing on simplicity and responsiveness, this module ensures strong user engagement and acts as the foundation for all other system operations.

B. Data Storage and Management Module

The Data Management Module handles the storage, retrieval, and organization of all task-related data. It uses lightweight database solutions such as SQLite or structured local storage to ensure quick access and efficient performance.

Each task is stored as a structured record containing multiple attributes, including:

- Task identifier
- Task description
- Scheduled timestamp
- Completion status
- Notification configuration

The module supports efficient CRUD (Create, Read, Update, Delete) operations, allowing users to manage tasks dynamically. Data indexing techniques are applied to improve retrieval speed, especially when handling multiple tasks across different time periods.

An important feature of this module is its persistence. Even if the application is closed, the device is restarted, or there is a temporary disconnection, the stored data remains safe and accessible. This ensures continuity and prevents data loss.

Additionally, the modular design allows future integration with cloud storage systems, enabling synchronization across multiple devices and improving scalability.

C. Scheduler and Time Monitoring Module

At the core of the system is the Scheduler and Time Monitoring Module, which acts as the time management engine. Its main role is to continuously track system time and match it with scheduled task events.

The module uses system-level timers, background services, or scheduling frameworks (such as Alarm Manager in Android) to ensure uninterrupted operation. Even when the application is not actively used, the scheduler continues running in the background, maintaining real-time tracking.

The scheduling process works by regularly comparing the current system time with stored task timestamps. When a match is found, the module triggers the corresponding notification. This real-time comparison ensures accurate execution without delays.

The module is also designed to handle multiple tasks efficiently. It prioritizes tasks based on time and manages overlapping schedules without conflict.

This continuous monitoring transforms the system into an active, time-aware assistant rather than just a passive reminder tool.

D. Notification and Alert Module

The Notification Module acts as the execution layer, converting scheduled tasks into alerts that users can notice easily. Traditional task managers often rely only on mobile notifications, which can be ignored. This system overcomes that limitation by using multiple alert methods.

- Upon activation, the system generates:
- Sound alerts using buzzers or alarm tones
- Visual signals through LEDs or screen indicators
- Mobile push notifications

This multi-layered alert system ensures that reminders are clear and effective, greatly reducing missed tasks.

The module also allows customization, enabling users to choose their preferred alert types. For example, users can select silent alerts, vibration, or high-priority alarms depending on their needs.

From a technical point of view, the module is optimized for fast response. Notifications are delivered immediately when triggered, ensuring real-time feedback.

By combining both digital and physical alerts, this module improves reminder effectiveness and user response.

E. Time Synchronization and World Clock Module

Accurate time tracking is essential for any scheduling system. The Time Synchronization Module ensures that all system operations are based on correct and consistent time.

The system integrates both:

• Device system clock for local tracking

• World Time API for real-time synchronization

This dual approach allows the system to automatically adjust to time zone changes, daylight saving variations, and incorrect device time settings. If users travel or operate in different regions, the system updates schedules automatically, avoiding manual changes.

Regular synchronization ensures that the system stays aligned with standard time sources, improving reliability. This module plays a key role in maintaining time accuracy throughout the system.

F. Settings and Customization Module

The Settings Module provides flexibility, allowing users to personalize the system according to their preferences.

Users can configure options such as:

• Alarm tones and volume levels

• Notification types (visual, audio, or both)

• Task repetition (daily, weekly, etc.)

• Interface preferences

These settings are stored using lightweight storage methods such as Shared Preferences or local configuration files, ensuring they are retained across sessions.

Customization improves user satisfaction by adapting the system to individual needs. For instance, students may prefer frequent reminders, while professionals may choose fewer but more important alerts.

This module ensures the system remains flexible and suitable for different types of users and environments.

G. System Integration and Workflow

The system operates as a well-connected yet modular workflow, where each component contributes to a smooth and continuous process. The interaction between modules ensures efficient data flow, quick response, and minimal delay in task execution.

Initially, the User Interaction Module collects task inputs, including scheduling details and user preferences. These inputs are passed to the Data Management Module, where they are structured, validated, and stored securely.

The Scheduler and Time Monitoring Module continuously compares the system time with stored task data using background processes, ensuring real-time monitoring even when the app is inactive. To maintain accuracy, the Time Synchronization Module aligns the system time with standard references.

When a scheduled time is reached, the Notification Module is triggered, generating alerts through sound, visual signals, and mobile notifications. The behavior of these alerts is controlled by user settings configured in the Settings Module.

This coordinated workflow turns the system into an active assistant that manages tasks automatically. Unlike traditional systems that rely on user attention, this approach ensures timely reminders and better task completion.

H. Architectural Significance

The architecture of the system represents a shift from traditional software-only task managers to a hybrid model that combines IoT-based physical interaction with intelligent scheduling. This approach improves reminder effectiveness by extending notifications beyond screens into the physical environment.

A key advantage of this design is its modular structure, which allows clear separation of functions while maintaining smooth communication between components. This simplifies development, debugging, and future maintenance.

From a scalability point of view, the system is designed to support future features such as AI-based scheduling, user behavior analysis, and adaptive reminders. These additions can be implemented without major structural changes.

The system is also highly extendable, allowing integration with smart home devices and other IoT systems, making it suitable for advanced intelligent environments.

Overall, the architecture achieves a strong balance between reliability, flexibility, and user-focused design, making it a powerful solution for real-time task management in both standalone and connected systems.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

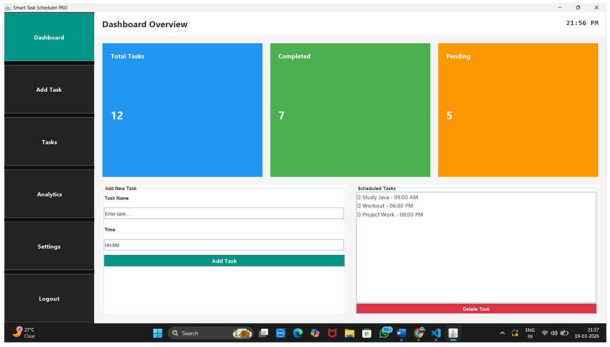


Fig.1.Dashboard

I. Complete Design Flow: Personal Task Scheduler Integrated with Wall Clock

The complete process flow of the Personal Task Scheduling System is represented through a structured block diagram. The system is designed with multiple interconnected modules that work together to provide real-time synchronization, intelligent scheduling, and context-aware reminders.

1) Data Acquisition Module

This module collects all necessary input data from users and external sources.

a) Task Input and User Interaction: Users can create, edit, and manage tasks using a mobile or web-based application. They can define details such as task name, priority, deadline, recurrence, and reminder settings.

b) Device Integration (Wall Clock Sync): The application connects to a smart wall clock (or enhanced traditional clock) using wireless technologies like Bluetooth or Wi-Fi. This allows real-time synchronization between the application and the physical device.

c) External Context Data Integration: The system can optionally include external inputs such as calendar events, weather updates, and system time. These help improve contextual awareness and enable smarter scheduling.

2) Input Data Processing

This stage prepares raw input data before applying scheduling logic.

a) Task Data Processing: Task-related information like deadlines, priorities, and recurrence rules are validated and structured to ensure consistency and completeness.

b) Time and Context Data Processing: System time, date, and contextual inputs (such as user activity or environment) are processed to align tasks with real-world conditions, ensuring accurate scheduling.

3) Task Data Formation

After processing, all task-related and contextual information is combined into a unified dataset called Task Data.
This acts as the core input for scheduling and decision-making, enabling intelligent and adaptive task handling.

4) Data Reprocessing Module

This step ensures that the data is clean and usable for scheduling.

a) Handling Missing or Incomplete Data: Incomplete entries (like missing deadlines or priorities) are handled using default values or user prompts, ensuring all tasks are schedulable.

b) Data Cleaning: Duplicate tasks, invalid entries, and inconsistent formats are removed or corrected, improving reliability and avoiding redundant alerts.

5) Conversion to Structured Form

The cleaned Task Data is transformed into a structured format suitable for processing by scheduling algorithms.
This includes:

• Time normalization (standard formats)
• Priority encoding (high, medium, low)
• Recurrence pattern structuring

• Context tagging (work, personal, urgent)
This structured format ensures efficient and accurate scheduling.

6) Scheduling Engine Module
This is the core logic layer of the system.
a) Task Scheduling and Optimization: Tasks are arranged based on priority, deadlines, and user preferences. The system allocates time slots and resolves conflicts between overlapping tasks.
b) Context-Aware Adjustment: The scheduler adapts dynamically to changes such as delays, inactivity, or updated priorities. Tasks may be rescheduled automatically to maintain efficiency.
c) Smart Reminder Generation: The system calculates the best times to send reminders based on task importance and deadlines. It supports multiple reminder levels like early alerts, on-time alerts, and overdue notifications.

7) Synchronization Module
This module ensures that scheduled data is reflected in the wall clock.
a) Real-Time Sync with Wall Clock: Scheduled tasks are sent to the wall clock device, which updates its internal system to display upcoming tasks and reminders.
b) Continuous Update Mechanism: Any changes made in the application (such as task updates or deletions) are instantly synchronized with the wall clock to maintain consistency.

8) Notification and Alert Module
This module delivers reminders to the user through multiple channels.
a) Visual Notifications (Wall Clock Display): The wall clock shows alerts using LEDs, digital text, or indicator lights.
b) Audio Alerts: Sound alarms or buzzers are triggered for important tasks.
c) App Notifications: The mobile application also sends push notifications to ensure accessibility and backup alerts.

9) Output Generation
The system produces the following outputs:
• Real-time task reminders
• Visual and audio alerts via the wall clock
• Updated task schedules
• Task completion status and tracking

10) Feedback and Learning Module (Optional Enhancement)
This module enhances the system by learning from user behavior.
a) User Interaction Feedback: The system monitors how users respond to reminders (completed, snoozed, or ignored).
b) Adaptive Scheduling: Based on user habits, the system adjusts reminder timings and task priorities to provide a more personalized experience..

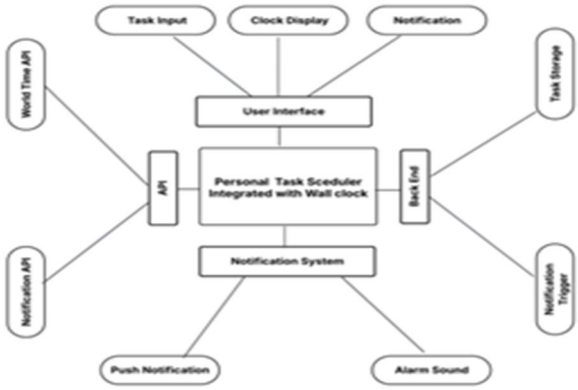


Fig.2. Block Diagram

IV. CONCLUSION

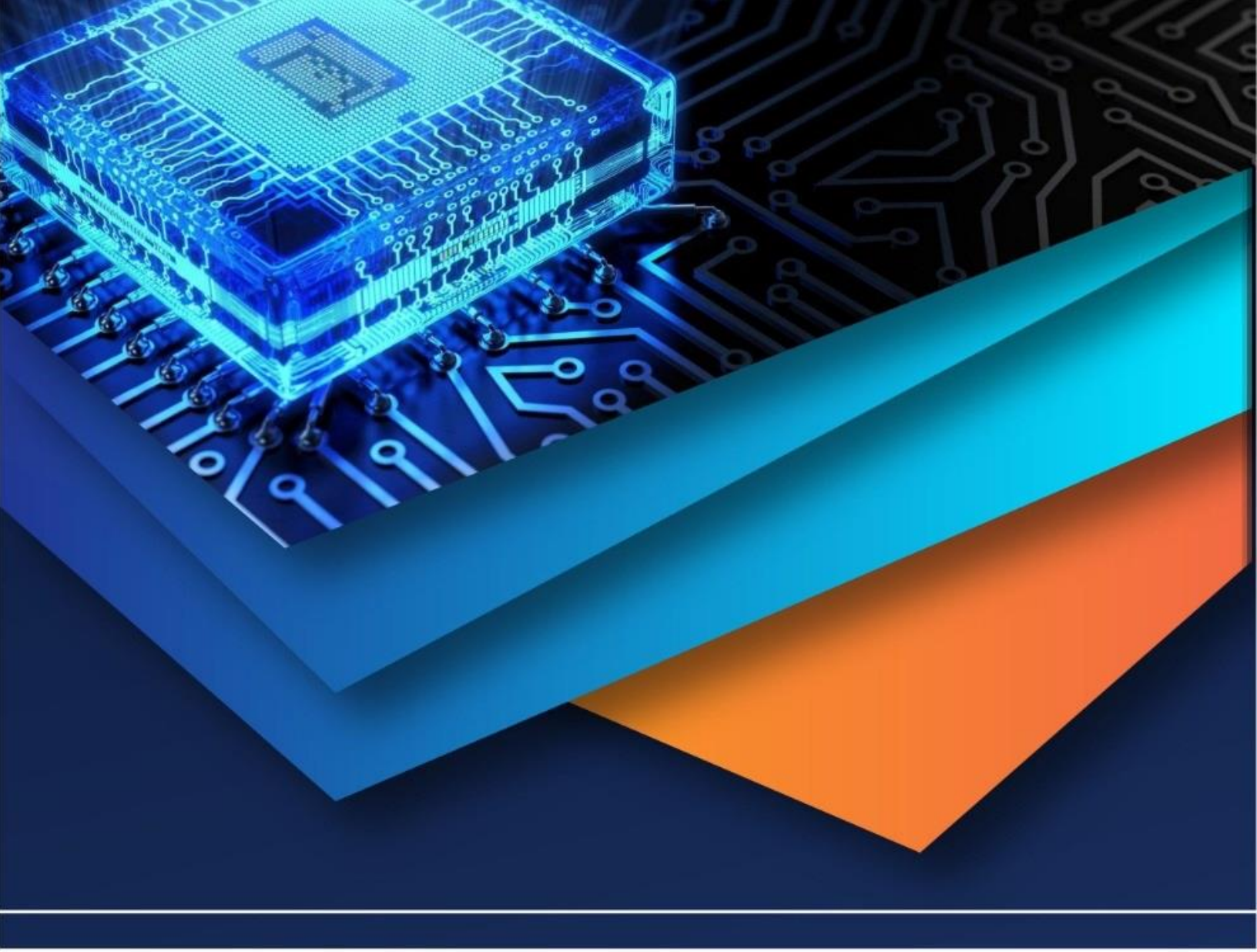
The integrated personal task scheduler and wall clock framework serves as an effective solution for smarter, real-time time management. By combining structured task scheduling with context-aware synchronization, the system ensures that reminders are both timely and reliable while adapting to user needs. Unlike standalone scheduling applications, this hybrid approach improves usability by extending digital task management into a physical, always-visible interface, helping users stay on track and reducing missed commitments. Beyond basic scheduling, the real strength of the system lies in providing clear and practical reminders through visual and audio signals directly from the wall clock. With intelligent prioritization and real-time synchronization, users are encouraged to manage their time more effectively and respond quickly to upcoming tasks. This user-focused design supports better productivity, minimizes procrastination, and helps in planning daily activities more efficiently.

With future improvements such as learning user behavior, adaptive scheduling, and enhanced context awareness, the system has the potential to evolve into a highly personalized assistant. Overall, the proposed framework offers a practical and scalable solution for modern smart environments, successfully connecting digital planning tools with everyday physical spaces.

REFERENCES

[1] Arakawa, R., Yakura, H., & Goto, M. (2023). CatAlyst: Domain-Extensible Intervention for Preventing Task Procrastination Using Large Generative Models. arXiv.
[2] Abbas, A., & Lee, S. W. (2024). PITCH: Productivity and Mental Well-being Coaching through Daily Conversational Interaction. arXiv.
[3] Lalwani, H., & Salam, H. (2025). Supporting Productivity Skill Development in College Students through Social Robot Coaching: A Proof-of-Concept. arXiv.
[4] Xu, Z., Gong, Y., Zhou, Y., Bao, Q., & Qian, W. (2024). Enhancing Task Scheduling with Deep Learning and Reinforcement Learning in Cloud Computing. arXiv.
[5] International Journal of Innovative Science and Research Technology. (2025). Task Scheduling Mobile Application for Student Productivity Improvement.
[6] International Journal of Scientific Research & Engineering Trends. (2024). AI-based Anti-Distraction and Task Management System.
[7] MoldStud Research Team. (2024). Organize Your Life: How Task Management Apps Improve Productivity.
[8] Producti Inc. (2025). AI-Powered Time Management and Scheduling Systems.
[9] Google. (2024). Google Calendar Help Center. Retrieved from https://support.google.com/calendar
[10] Microsoft. (2024). Microsoft To Do Documentation. Retrieved from https://support.microsoft.com
[11] Mozilla Developer Network (MDN). (2024). JavaScript Guide. Retrieved from https://developer.mozilla.org
[12] Fielding, R. T. (2000). Architectural Styles and the Design of Network-based Software Architectures.
[13] Tilkov, S., & Vinoski, S. (2010). Node.js: Using JavaScript to Build High-Performance Network Applications. IEEE Internet Computing.
[14] Facebook (Meta). (2024). React Documentation. Retrieved from https://react.dev
[15] Forbes Technology Council. (2024). Top Productivity and Task Management Applications.
[16] Journal of Biomedical Research & Environmental Sciences. (2026). Productivity Apps for Academics in USA and India.
[17] TempusTact. (2024). Top Time Management Applications for Productivity Enhancement.
[18] Nørlem, H. L., & Azari, M. W. (2024). Tiimo: AI-based Task Planning Application for Neurodivergent Users.
[19] Eighty Percent Solutions. (2025). Freedom App: Distraction Blocking for Productivity.
[20] Kelly, J. (2025). Best Productivity Apps and Task Managers. Forbes.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)