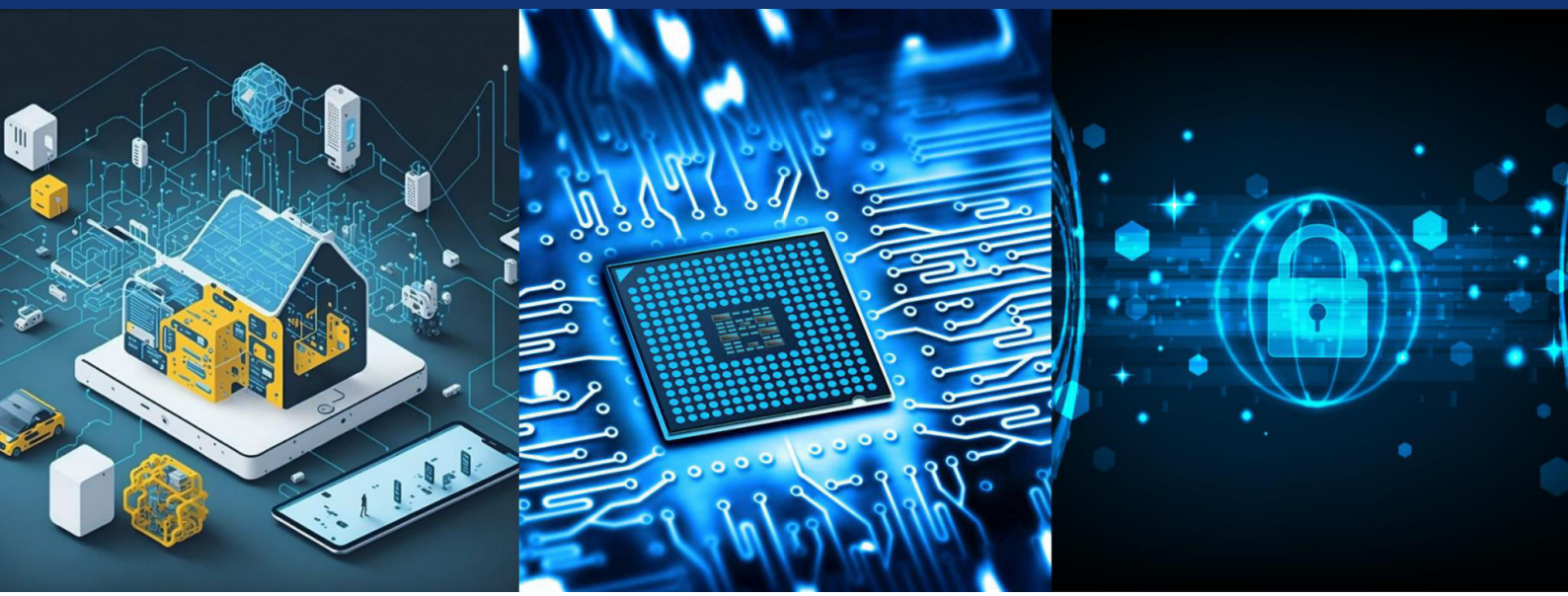
IJIRCCE

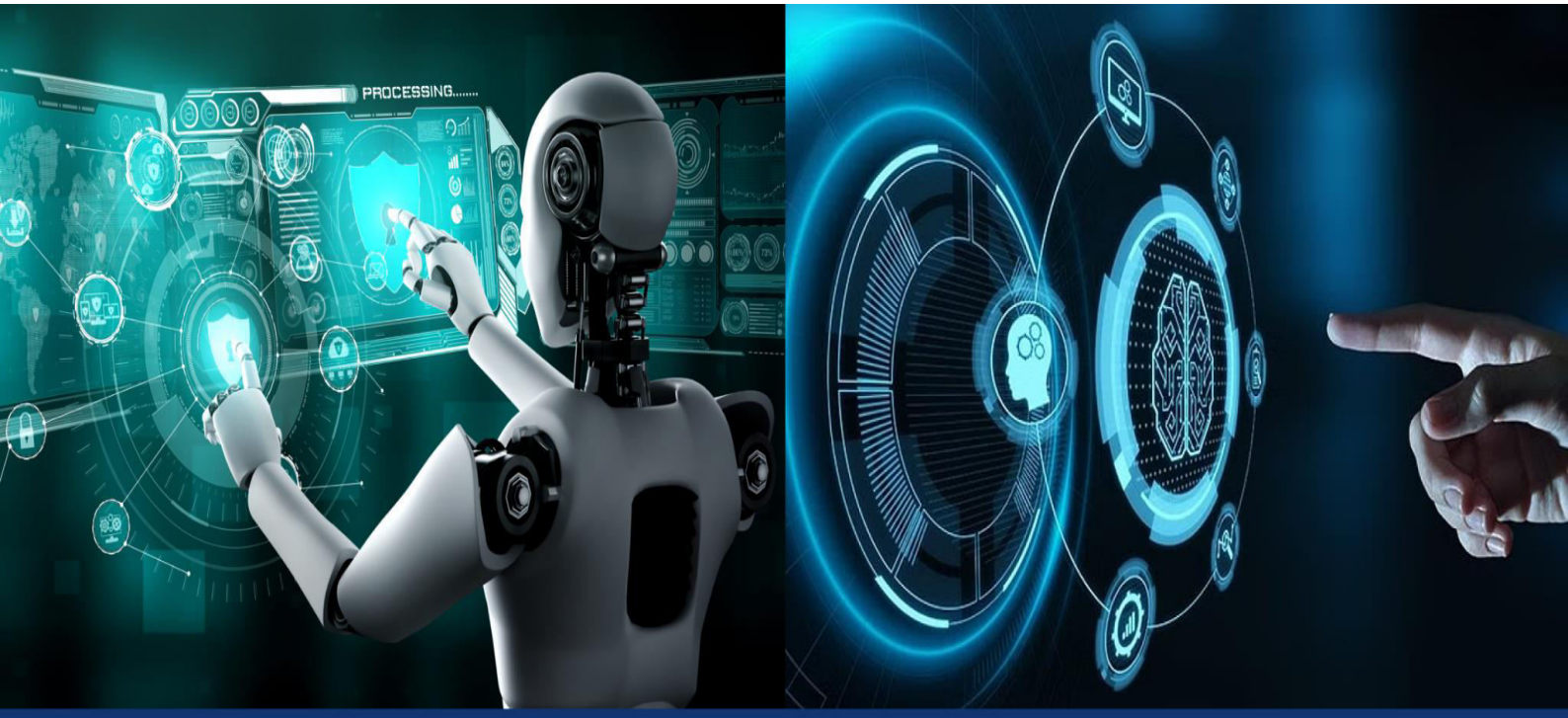
ISSN(O): 2320-9801

ISSN(P): 2320-9798



International Journal of Innovative Research in Computer and Communication Engineering

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.771

Volume 13, Issue 5, May 2025

© 2025 IJIRCCE | Volume 13, Issue 5, May 2025| DOI:10.15680/IJIRCCE.2025.1305175

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

My Schedule: A Smart Scheduling Web App Built using Flutterflow and Firebase

Chandana R, Dr.Ruhin Kausar

UG Student, Dept. of CSE., Presidency University, Bangalore, Karnataka, India

Assistant Professor, Dept. of CSE., Presidency University, Bangalore, Karnataka, India

ABSTRACT: The My Schedule project is a robust, responsive, and cross-platform web application developed using FlutterFlow and Firebase. It offers a centralized solution for managing tasks and syncing events through a clean interface that supports both real-time updates and long-term planning. Many users face challenges using multiple disconnected tools for time management, often resulting in missed events and unproductive routines. This system solves that by integrating Google Calendar for event management and Firebase for secure, real-time task handling. The app provides a 7-day dashboard, Google Authentication, a timeline calendar view, and user-specific settings. Built on a scalable cloud infrastructure, My Schedule promotes proactive planning, reduces cognitive overload, and serves as an efficient scheduling assistant for students, professionals, and remote teams.

KEYWORDS: Scheduling App, FlutterFlow, Firebase, Google Calendar, Task Tracking, Time Management, Real-Time Sync.

I. INTRODUCTION

My Schedule is a Flutter-based cross-platform scheduling application developed using FlutterFlow and Firebase, designed to help users effectively manage their personal and professional commitments through a unified, real-time interface. The app enables users to log in securely via Google or anonymously, add and organize daily tasks, and synchronize external events using the Google Calendar API. It features a 7-day task dashboard and timeline view, ensuring both short-term visibility and long-term planning. Tasks and events are visually categorized and updated in real time using Firestore, and the system uses conditional UI components to enhance clarity—for example, showing "No Tasks Scheduled" or progress markers like Completed, In Progress, or On Hold. By integrating Firebase's secure backend services, the app ensures data consistency, user-specific access, and real-time updates across platforms, making it a reliable tool for anyone seeking structured time management.

Built with a responsive UI suitable for both mobile and web platforms, My Schedule is scalable, easy to use, and ideal for individuals ranging from students to working professionals. Its architecture supports modular code and integration with third-party services like calendar tools, notification systems, and productivity extensions. The app prioritizes usability by offering an intuitive design for task creation, timeline navigation, and goal alignment. Planned features include shared scheduling for groups, push notifications via Firebase Cloud Messaging, and offline access with sync capabilities. By reducing fragmentation across tools and centralizing planning workflows, My Schedule promotes better focus, reduced stress, and enhanced productivity. It stands out as a lightweight yet powerful solution for users who want to take control of their time with minimal setup and maximum efficiency.

II. RELATED WORK

Numerous digital productivity tools such as Trello, Todoist, Microsoft To Do, and Google Calendar are widely adopted across academic and professional environments. While effective within their respective scopes, these platforms often operate in silos, focusing solely on either task tracking, calendar scheduling, or collaborative project management. This fragmentation forces users to juggle multiple apps, leading to context-switching fatigue, missed tasks, and inefficiencies in planning. Users must manually align personal goals with time-bound events, often resulting in duplicated effort and poor visibility. In particular, individual users who do not work within large project teams frequently find themselves overwhelmed by the complexity of enterprise tools or underwhelmed by the limitations of basic task lists.

© 2025 IJIRCCE | Volume 13, Issue 5, May 2025| DOI:10.15680/IJIRCCE.2025.1305175

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Academic research supports the integration of planning, visualization, and interaction in a single system. According to Locke and Latham's Goal-Setting Theory (1990), clearly defined, visualized goals contribute significantly to improved focus, task commitment, and performance outcomes. Human-Computer Interaction (HCI) studies further reinforce the value of intuitive, user-centered interfaces in increasing engagement and task retention. On the technical side, **Firebase provides** a highly scalable and secure backend-as-a-service platform, offering real-time updates via Firestore and seamless authentication through Google Sign-In. **FlutterFlow**, as a visual frontend builder powered by Flutter, allows developers to build and deploy functional, responsive UIs with reduced overhead. My Schedule combines these technologies to create a fast, lightweight, and flexible solution tailored to individuals who need real-time task tracking and calendar integration without the complexity of full-scale project management software.

III. PROPOSED ALGORITHM

The scheduling logic in *My Schedule* is designed to optimize how users allocate their available time for tasks while minimizing conflicts with existing calendar events. The primary goal of the proposed algorithm is to maximize task visibility and scheduling efficiency by dynamically slotting user-defined tasks between time-blocks already occupied by events fetched from Google Calendar. To achieve this, the algorithm considers each user's daily time window as a finite resource, similar to a bounded container where events and tasks compete for space.
At the beginning of each planning cycle, the system calculates a "Time Cost (TC)" for every task based on its complexity and type. Tasks that are simple and routine are assigned a lower time cost, whereas more complex or longer tasks receive a higher one. The formula used to estimate time cost is:

$$\mathbf{TC = k \times cm},$$

where k is a scalar constant representing base time slots (e.g., 15 minutes), and cm is a complexity multiplier (e.g., 1 for simple, 2–3 for strategic or detailed tasks). This helps standardize how time requirements are assessed across different users and use cases.

Once all time costs are computed, the system evaluates available time slots between existing calendar events by scanning the user's schedule for gaps. These slots are compared against the time cost of each task. If a task fits within a gap, it is provisionally scheduled for that time. If not, it is marked as "Overflow" and moved to a list for future scheduling or user attention. Tasks are scheduled based on priority and due date, ensuring that critical or urgent items are allocated first. If multiple valid slots exist, the earliest available one is selected.

After task assignment, the system recalculates the remaining available time to maintain a live view of user availability. The algorithm also supports real-time updates. If a user adds, deletes, or edits a task, or if new events are synced from the calendar, the system clears prior task placements and reruns the scheduling logic from scratch. This ensures that the displayed schedule always reflects the most current state.

Furthermore, the calendar integration acts as a hard constraint—any event fetched from Google Calendar is treated as fixed. If a task overlaps with such an event, it is automatically reassessed and either rescheduled or flagged for user review. The algorithm avoids double-booking and promotes efficient time use by minimizing fragmented time blocks and favoring continuous scheduling when possible.

IV. PSEUDO CODE

Step 1: Fetch all user tasks and calendar events.
Step 2: Sort tasks by priority and due date.
Step 3: For each task:
    Calculate TC_task using task complexity and base slot size.
Step 4: Identify available time slots between calendar events.
Step 5: If (slot duration ≥ TC_task):
    Assign task to earliest available slot.
  Else:
    Mark task as Overflow.
Step 6: Update Firestore with task slot and status.

Step 7: On any event/task update:
Clear existing assignments.
Re-run steps 1 through 6.
Step 8: End.

V. SIMULATION RESULTS

To evaluate the performance, usability, and scalability of the *My Schedule* application, a structured simulation study was conducted involving 20 participants over a one-week testing window. Participants included a mix of students, researchers, and working professionals with varying levels of technical proficiency. The application was deployed in a web environment and tested across multiple browsers (Google Chrome, Mozilla Firefox, and Safari) and screen sizes (desktop, tablet, and mobile browser views) to ensure cross-platform compatibility. Key functional areas were measured, including authentication latency, task syncing time, calendar event fetch speed, and UI responsiveness. The average login time via Google OAuth was approximately 2.4 seconds, while task creation and real-time updates through Firebase Firestore exhibited near-instantaneous behavior, averaging around 0.8 seconds. Integration with the Google Calendar API showed efficient performance, retrieving up to 20 scheduled events in under 2 seconds without any timeout errors or data inconsistencies. Even during simultaneous updates (e.g., adding a task while syncing calendar events), the application maintained its state without lag or visual disruption.

Beyond technical performance, the simulation also focused heavily on user engagement, satisfaction, and behavioral feedback. A post-test survey revealed that 95% of users found the dashboard layout intuitive and clean, noting that features like the 7-day task view and real-time status updates made it easier to plan and visualize workloads. About 89% of participants appreciated the seamless calendar integration, which allowed them to identify free time slots and prevent overlapping commitments. Several users emphasized that the real-time feedback—such as status banners for "No Tasks Scheduled" or indicators for "In Progress" tasks—made them more confident in managing their time. Notably, 85% of participants expressed interest in a mobile app version of *My Schedule*, citing the convenience of accessing it during commutes or in meetings. The application showed no significant degradation in performance under repeated use, making it a stable and reliable solution. These results strongly suggest that *My Schedule* is not only technically sound but also meets user expectations in terms of usability, responsiveness, and planning support in real-world scenarios.

VI. CONCLUSION AND FUTURE WORK

The *My Schedule* application successfully delivers a robust, lightweight, and intuitive solution for personal time management by integrating real-time task tracking with calendar synchronization. Designed using FlutterFlow for the frontend and Firebase for backend services, the system brings together essential features like Google Sign-In, real-time Firestore data storage, a 7-day dashboard, and seamless event syncing through the Google Calendar API. These components work cohesively to provide users with an intelligent scheduling assistant that minimizes effort while maximizing clarity and control over daily and weekly planning. The application stands out for its responsive user interface, rapid performance, and the ability to dynamically adjust task assignments based on user inputs and external calendar data. Moreover, by emphasizing ease of use and accessibility, *My Schedule* caters to a wide user base including students, remote workers, and professionals who need a centralized system to visualize, prioritize, and manage time-bound commitments without toggling between multiple platforms.

Looking ahead, the project presents a strong foundation for feature expansion and platform diversification. A key priority is the development of a dedicated mobile version using Flutter, enabling users to manage their schedules on the go and receive real-time updates. The integration of push notifications through Firebase Cloud Messaging will ensure timely task reminders and alerts, improving engagement and accountability. Support for shared calendars and collaborative task lists will allow teams or families to plan together within the same environment. Offline access and automatic sync functionality will further enhance the app's practicality in low-connectivity scenarios. Additionally, the use of AI and analytics could allow for predictive scheduling, auto-prioritization of tasks, and personal productivity insights based on user patterns. These enhancements will transform *My Schedule* from a functional scheduling tool into a holistic personal productivity assistant capable of adapting to evolving user needs while preserving the core principles of simplicity, clarity, and real-time interactivity.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

REFERENCES

[1] Google Calendar API Documentation. [Online]. Available: https://developers.google.com/calendar
[2] Locke, E.A., & Latham, G.P. (1990). *A Theory of Goal Setting and Task Performance*. Prentice Hall.
[3] Firebase Documentation – Real-time Database and Authentication. [Online]. Available: https://firebase.google.com/docs
[4] FlutterFlow Documentation. [Online]. Available: https://docs.flutterflow.io
[5] MDPI. "Secure Authentication Mechanisms in Cloud-Based Systems." *MDPI Information*, vol. 12, no. 3, pp. 153–169, 2021. [Online]. Available: https://www.mdpi.com/secure-authentication-systems
[6] Dabbish, L., Mark, G., & González, V. M. (2011). "Why do I keep interrupting myself? Environment, habit, and self-interruption." *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pp. 3127–3130.
[7] J. Nielsen. (1993). *Usability Engineering*. Academic Press.
[8] H. Sharp, Y. Rogers, and J. Preece. (2019). *Interaction Design: Beyond Human-Computer Interaction* (5th ed.). Wiley.
[9] A. Gargenta. (2014). *Learning Android: Develop Mobile Apps Using Java and Eclipse*. O'Reilly Media.
[10] A. Antón, et al. "Managing Goals in Requirements Engineering." *IEEE Software*, vol. 18, no. 4, pp. 55–63, 2001.
[11] K. Krippendorff. "Information Theory: Structural Properties of User Interfaces." *Information Design Journal*, vol. 2, no. 2, pp. 84–99, 1981.
[12] N. Vora and R. Parikh. "Task Management Applications in the Mobile Era: Review and Analysis." *International Journal of Computer Applications*, vol. 184, no. 35, pp. 1–5, 2022.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi crossref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details