World Journal of Advanced Research and Reviews
eISSN: 2581-9615 CODEN (USA): WJARAI
Cross Ref DOI: 10.30574/wjarr
Journal homepage: https://wjarr.com/

(REVIEW ARTICLE)

Check for updates

A Survey on Event Ease: Easy event scheduling for busy people

Kavitha Soppari, Shiva Ramavath *, Vinay Donthula and Sriranga Sheshasai Mangalagiri

Department of CSE (AI & ML), Ace Engineering College, Hyderabad, Telangana, India.

World Journal of Advanced Research and Reviews, 2025, 26(02), 370-377

Publication history: Received on 22 March 2025; revised on 30 April 2025; accepted on 03 May 2025

Article DOI: https://doi.org/10.30574/wjarr.2025.26.2.1601

Abstract

In a world of hybrid work environments, personal scheduling is becoming extremely important, traditionally done calendars do not account for group availability and task dependencies, Event Management Scheduling proposes an AI driven platform that automates event scheduling through intelligent problem resolution, availability checking and predictive analytics. It allows organizers to create events and send invitations to users who choose their preferred times. The system works by responding answers and determining the best time, afterwards, each participant's calendar gets updated automatically. It is built on JavaScript, Tailwind CSS, and React.js for the responsive interface, Node.js serves as the backend logic, and real-time data is managed through Firebase. Scheduling is done smoothly using Google Calendar's API and efficient communication between frontend and backend is done through RESTful APIs. To improve collaboration and streamline the scheduling process, users are notified through real-time message pop-ups about any changes made.

Keywords: Tailwind CSS; Reactjs; Nodejs; Firebase; Google Calendar API; RESTful APIs; JavaScript

1. Introduction

1.1. Background and Motivation:

While efficient time management is critical to productivity, scheduling continues to be a challenge in contemporary workflows. As modern organizations become more interdependent, coordination across roles, time zones, and priorities gets increasingly difficult. In fast-paced environments with overlaps and last-minute changes, the use of static calendars and manual inputs renders traditional scheduling methods insufficient. The growth of hybrid teams, remote work, and ad-hoc collaboration have compounded this problem and increased the demand for advanced scheduling tools. The current landscape presents an opportunity to leverage AI technologies for automated preference learning, availability forecasting, and coordination automation, thereby enabling teams and individuals to shift their focus to more valuable activities.

1.2. Introduction:

In an increasingly interconnected and fast-paced world, scheduling has become a critical logistical and administrative function across both personal and professional domains. The manual coordination of meetings, appointments, and tasks often results in scheduling conflicts, inefficiencies, and missed opportunities—challenges that are magnified in dynamic environments. While traditional calendar tools offer basic support, they fall short in terms of scalability and personalization, especially when it comes to managing group availability or incorporating task dependencies.

This aims to address these limitations by leveraging artificial intelligence (AI) to create a smarter, more adaptive scheduling solution. AI technologies provide powerful capabilities for automating complex scheduling decisions,

* Corresponding author: Ramavath Shiva

including intelligent conflict resolution, real-time availability detection, and predictive analytics. These features not only streamline the coordination process but also improve user experience and overall time efficiency. As virtual collaboration and hybrid work models continue to gain traction, there is a growing demand for intelligent scheduling platforms that can respond to changing needs in real time. This project seeks to meet that demand by developing a responsive, AI-driven scheduling system designed for modern work and life dynamics.

2. Literature Survey review:

2.1. Lee, J., & Martinez, C. (2024.). User-Centric Design for Event Scheduling Applications.

This research examines how user interface (UI) and user experience (UX) design influence the effectiveness and usability of event scheduling applications. The study highlights that an intuitive, visually appealing interface greatly improves user engagement and satisfaction, making scheduling tasks more efficient and enjoyable. By focusing on simplicity and clarity, the design minimizes user friction, reducing the cognitive load required to navigate the application. The authors demonstrate that when the interface is tailored to user needs—incorporating dynamic, responsive elements—users are more likely to complete their scheduling tasks quickly and accurately. Furthermore, a well-designed UX ensures that users can access key functionalities with minimal effort, improving both the speed and accuracy of event planning. This research emphasizes that thoughtful design plays a crucial role in the adoption and success of event scheduling tools, as it directly affects user retention and productivity.

2.1.1. Methodologies and Algorithms

This scheduling application employs responsive web design principles using Tailwind CSS, a utility-first framework that accelerates UI development. The design adapts seamlessly across device types, ensuring accessibility on smartphones, tablets, and desktops. Key UI elements are arranged using a grid system for clean navigation. The interface leverages adaptive font scaling, color schemes, and button feedback for intuitive user interaction. Usability testing was conducted with real users to fine-tune layout and minimize friction. Performance optimizations include lazy loading and component-based rendering using React. Tailwind's built-in classes reduce styling overhead and promote consistency. The interface follows WCAG accessibility standards to support users with disabilities. Real-time feedback mechanisms like inline validations enhance interactivity. These design choices significantly improved task success rates and reduced scheduling errors.

2.2. Johnson, et al. (2023.). A Web-Based Event Management System for Efficient Scheduling.

This study introduces a comprehensive web-based event management platform designed to streamline the process of scheduling and coordinating events. It empowers users to create events, send invitations, and gather participant availability in a centralized system. By automating the process of invitation sending and schedule coordination, the platform alleviates the common challenge of conflicting schedules among multiple participants. It ensures that schedules are finalized only when the majority of participants confirm availability, thus minimizing the chances of double-booking or missed appointments. The system's real-time notifications and feedback mechanisms keep all participants up to date on any changes to the event details, promoting better collaboration. This approach not only enhances the efficiency of the scheduling process but also improves the overall user experience by reducing the time and effort traditionally required for event planning. Furthermore, the platform adapts to varying organizational and social contexts, offering flexibility in managing both small meetings and large gatherings.

2.2.1. Methodologies and Algorithms

The system architecture is based on modern full-stack web technologies. The front-end is developed using React.js, a robust JavaScript library that allows for the creation of dynamic, component-based user interfaces. React enables real-time interactivity and seamless rendering of updated content, improving the overall user experience. On the back-end, Node.js is employed to handle server-side logic and API processing. Node.js facilitates high-performance, scalable operations, making it ideal for applications that require quick response times and continuous updates.

To streamline the scheduling process, the system incorporates real-time participant input mechanisms. Users can respond to invitations by marking their availability, and the application automatically analyzes the collected data to identify common free slots. This ensures that the finalized schedule reflects a consensus, minimizing conflicts and maximizing participation. Overall, the integration of these technologies and methodologies enables a responsive, efficient, and intelligent scheduling environment suited for dynamic, multi-user scenarios.

2.3. Nguyen, T., & Li, D. (2023). Enhancing Group Scheduling Through AI-Powered Negotiation Agents.

This study introduces an innovative group scheduling system that utilizes AI-powered negotiation agents to autonomously negotiate and determine meeting times among multiple participants. By leveraging artificial intelligence, the system eliminates the need for manual coordination and back-and-forth communication, which is often time-consuming and inefficient. The AI agents analyze participant availability and preferences to propose optimal meeting slots that accommodate everyone involved. These agents function by simulating human-like negotiation, dynamically adjusting their suggestions based on feedback and constraints from users. The system not only reduces the friction typically associated with manual scheduling but also ensures that decisions are made more quickly and with greater accuracy. As a result, this AI-driven approach enhances the scheduling experience, saving time and minimizing the potential for conflicts. The use of negotiation agents also makes the system adaptable to different group sizes and scheduling scenarios, promoting greater flexibility and ease of use.

2.3.1. Methodologies and Algorithms

The system leverages constraint satisfaction models to define rules such as preferred meeting hours, time zone compatibility, and user availability. These constraints guide the negotiation logic used by AI agents. Multi-agent reinforcement learning is applied, where agents learn to select meeting slots based on rewards—such as maximizing attendance or minimizing conflict. Each agent represents a participant and proposes time slots based on local constraints and feedback from others. Communication between agents is managed through a distributed protocol that prioritizes minimal message exchange. Over time, agents refine strategies using Q-learning or policy gradient techniques. This distributed decision-making model reduces the need for human negotiation. Conflict detection and resolution modules ensure feasible final schedules. A consensus mechanism validates mutually accepted slots. This system simulates human-like negotiation with significantly lower coordination overhead.

2.4. Williams, A., & Clark, B. (2022.). Integration of Google Calendar API for Seamless Event Scheduling.

This study explores the integration of the Google Calendar API with Firebase to improve event scheduling by providing real-time updates and automation. By linking the Google Calendar API, the system ensures that events are automatically added to participants' calendars without requiring manual entry. This integration helps users save time by eliminating the need for repeated data entry across multiple platforms. Furthermore, participants receive instant notifications about event changes or confirmations, ensuring timely communication and coordination. The system promotes efficiency by automatically updating schedules, which reduces the chances of conflicts and missed events. It also facilitates better synchronization between event organizers and participants by ensuring that all involved parties are informed about the event details in real-time. The use of Firebase enables smooth, real-time synchronization across multiple devices, providing an uninterrupted experience for users, whether they are accessing their calendars through mobile phones or desktop computers.

2.4.1. Methodologies and Algorithms

The technical implementation is centred around two core technologies: Google Calendar API and Firebase. The Google Calendar API serves as the bridge between the scheduling platform and users' individual calendars. It enables automatic event creation, updates, and deletion directly from the web application interface. This ensures that user calendars are always up-to-date without manual synchronization.

Firebase, a cloud-based backend solution, is used to manage real-time communication between the front-end and back-end. It supports real-time data synchronization and messaging, ensuring that any changes to an event—such as time, participants, or details—are instantly pushed to all involved users. Firebase's real-time database and push notification services enhance the responsiveness of the application, significantly improving the user experience by reducing latency and maintaining data consistency across devices.

Together, these technologies allow for a responsive, user-centric scheduling system that ensures calendar accuracy, promotes timely coordination, and minimizes scheduling conflicts.

2.5. Chen, M., & Garcia, J. (2022). Adaptive User Interfaces for Calendar Applications

This research focuses on enhancing the usability of calendar applications by employing adaptive user interfaces (UI) that adjust based on individual user behaviour and preferences. The study underscores the significance of personalizing the user experience to ensure that calendar applications cater to diverse needs, improving both accessibility and efficiency. By analysing how users interact with the interface, the system dynamically adapts key UI elements—such as layout, buttons, and colour schemes—depending on the user's behaviour, context, and frequency of use. This approach

aims to reduce cognitive load and provide a more intuitive experience, allowing users to navigate and complete scheduling tasks with minimal effort. The research also highlights the importance of ensuring that adaptive interfaces remain consistent across various devices, enhancing the experience regardless of screen size. Ultimately, the goal is to create a more responsive, user-centric calendar platform that evolves in tandem with user preferences, leading to greater satisfaction and productivity.

2.5.1. Methodologies and Algorithms

The adaptive user interface system monitors user interactions to learn preferences and usage patterns. Data is collected on which UI elements are accessed most frequently and how users navigate the platform. A behaviour-based engine uses this data to prioritize or hide interface components dynamically. Layouts are adjusted in real-time to highlight frequently used features, improving efficiency. The system uses responsive design principles to ensure consistency across devices. Conditional rendering allows certain features to appear only when relevant, reducing cognitive load. An event-driven architecture tracks interactions to trigger UI adjustments. Performance metrics are used to evaluate interface effectiveness. The design incorporates A/B testing to measure changes in user satisfaction. Accessibility adjustments (e.g., contrast, font size) are also applied based on user behaviour. These methods create a personalized and efficient scheduling experience.

2.6. Patel, A., & Singh, R. (2021.). Enhancing Event Planning Through RESTful API Integration

This study explores the enhancement of event planning systems through the use of RESTful APIs to streamline data exchange between event organizers and participants. The integration of RESTful APIs ensures that real-time communication is established, enabling seamless synchronization between all parties involved. With this system, organizers can easily manage event details, update schedules, and track responses, while attendees receive instant notifications and updates on their calendars. By supporting real-time collaboration, the platform reduces the traditional delays associated with manual coordination. The use of RESTful architecture also promotes scalability and flexibility, allowing the system to integrate with third-party tools and services as needed. This enhances the overall efficiency of event planning, as data is exchanged consistently and reliably without requiring constant manual intervention. Ultimately, the system provides a more cohesive and responsive scheduling experience for both organizers and attendees.

2.6.1. Methodologies and Algorithms

The event planning system utilizes RESTful API architecture to facilitate efficient and reliable communication between clients and servers. These APIs support CRUD operations, enabling event creation, update, deletion, and retrieval with minimal latency. The system ensures consistent data flow by structuring endpoints that handle requests asynchronously. JSON format is used for lightweight data exchange, and HTTPS protocols secure the transactions. Backend services are designed to handle user authentication, ensuring only authorized individuals can modify events. Real-time data propagation allows changes to reflect immediately on all user devices. Event logs are maintained to track changes and support rollback if needed. Caching strategies are applied to reduce redundant data retrieval. The modularity of RESTful APIs also allows for integration with third-party tools and calendar services. Together, these methods enable seamless, real-time collaboration in event planning.

2.7. Kim, H., & Zhao, Y. (2021). Intelligent Event Scheduling Using Machine Learning and Calendar APIs

This study introduces a smart scheduling system that leverages machine learning algorithms to optimize meeting times by analyzing historical data and calendar patterns. The system uses past scheduling behavior to predict the most suitable time slots for future events, improving scheduling efficiency. By integrating with the Google Calendar API, the platform ensures real-time availability checks for all participants, automatically adjusting meeting times based on their current schedules. This integration eliminates manual conflict resolution, allowing the system to propose optimal time slots for all involved. Additionally, the machine learning model continuously refines its predictions based on user feedback and changes in scheduling patterns, becoming more accurate over time. The combination of AI-driven suggestions and calendar synchronization offers a seamless experience, saving time and reducing scheduling conflicts. Ultimately, this intelligent system enhances productivity by automating the time-intensive task of finding mutually available slots for meetings.

2.7.1. Methodologies and Algorithms

The smart scheduling system applies K-means clustering to categorize users based on behavioural patterns in calendar usage. These clusters help the system recognize common availability trends. Random Forests, a machine learning algorithm, are then trained on historical scheduling data to predict the most suitable time slots for new events. The

Google Calendar API integration enables real-time extraction of user schedules and instant event entry. The algorithm continuously learns from user feedback, refining its prediction accuracy over time. Events are ranked and suggested based on match probability with user preferences. Conflict detection modules filter out time slots with overlapping meetings. A feedback loop improves model performance with each use. The hybrid of statistical modelling and live calendar access allows for dynamic, personalized scheduling. This results in optimized meeting times and higher participant attendance rates.

2.8. Thomas, E., & Rana, P. (2021). EventSync: A Cloud-Based Collaborative Scheduling Platform

EventSync introduces a cloud-based collaborative scheduling platform that facilitates real-time co-creation and simultaneous editing of events by multiple users. This feature is particularly valuable for teams working in distributed or remote environments, as it ensures seamless coordination despite geographical barriers. By enabling real-time collaboration, the platform allows users to adjust event details—such as time, location, and participant list—instantly, ensuring that all participants are always on the same page. The system's cloud infrastructure ensures that changes are synchronized across all devices, providing consistency and eliminating version conflicts. This collaborative approach significantly enhances team coordination and reduces the potential for scheduling errors. Furthermore, EventSync's intuitive interface makes it easy for users to participate in event planning without requiring technical expertise. By leveraging cloud technology, the platform supports scalability and flexibility, accommodating varying team sizes and scheduling needs.

2.8.1. Methodologies and Algorithms

EventSync is built on a Firebase backend, utilizing its real-time database capabilities to enable live collaboration. Each participant's input is instantly reflected across all user interfaces without the need to refresh. The system uses WebSockets for persistent bi-directional communication between clients and the server. On the front end, React manages a modular and interactive UI where each component (e.g., calendar, availability poll) updates in real time. Co-editing functionality allows users to concurrently modify event details, and Firebase conflict resolution tools handle simultaneous edits. Data is stored in structured JSON format for rapid access and manipulation. Changes are timestamped and tracked to maintain edit history and support rollback. Push notifications inform users about updates as they occur. This architecture supports high user concurrency and minimizes scheduling errors by enabling shared decision-making.

2.9. Banerjee, A., & Chauhan, R. (2020). A RESTful Microservice Approach for Scalable Event Management

This paper investigates the use of microservice architecture to improve the scalability, flexibility, and maintainability of event management systems. By decoupling different system components, such as notifications, user preferences, and scheduling logic, microservices allow each module to operate independently, enhancing the system's modularity. RESTful APIs are utilized to enable seamless communication between these microservices, ensuring that data can be transferred efficiently and securely across the platform. The architecture's decentralized nature ensures that any updates or changes to one service do not disrupt the entire system, allowing for easier maintenance and scaling as user demands grow. This approach also promotes fault isolation, meaning that if one service experiences issues, it does not affect other components. The use of RESTful APIs further simplifies integration with third-party tools and services, fostering a more flexible and adaptable event scheduling solution. Ultimately, this microservice approach enables robust, high-performance systems capable of handling large-scale scheduling tasks with ease.

2.9.1. Methodologies and Algorithms

The system is built using microservices connected via RESTful APIs, allowing components like scheduling, notifications, and user preferences to operate independently. Each service runs in a containerized environment (e.g., Docker), ensuring scalability and fault isolation. Load balancing techniques are employed to distribute traffic evenly across services, improving performance under heavy usage. Each API service communicates using lightweight HTTP protocols, with JSON for efficient data exchange. CI/CD pipelines automate testing and deployment to ensure reliability during updates. A gateway manages API routing and authentication, while monitoring tools track performance and uptime. Asynchronous messaging queues (like RabbitMQ) are used for non-blocking communication between services. Services are independently scalable based on demand, improving flexibility. This architecture improves fault tolerance and allows rapid scaling, particularly useful in enterprise-level scheduling.

Comparison table:

S.No	Author(s)	Title	Methodology Used	Findings from the Reference Paper
1	Lee, J., & Martinez, C. (2024)	User-Centric Design for Event Scheduling Applications	Tailwind CSS, Responsive Web Design	Improved user engagement with an intuitive and visually appealing interface. Reports show a 40% increase in user participation.
2	Johnson, et al. (2023)	A Web-Based Event Management System for Efficient Scheduling	React.js (front-end), Node.js (back-end)	The system automates invitations and reduces scheduling conflicts by finalizing schedules based on participant availability..
3	Nguyen, T., & Li, D. (2023)	Enhancing Group Scheduling Through AI-Powered Negotiation Agents	Multi-agent Reinforcement Learning, Constraint Satisfaction	Reduced communication overhead by 65% and 90% of users reported satisfaction with AI-suggested meeting times.
4	Williams, A., & Clark, B. (2022)	Integration of Google Calendar API for Seamless Event Scheduling	Google Calendar API, Firebase Google Calendar API, Firebase	Events are automatically added to participants' calendars and notifications are pushed instantly.
5	Chen, M., & Garcia, J. (2022)	Adaptive User Interfaces for Calendar Applications	Adaptive UI Design, Behavior-based UI Customization	25% improvement in task completion time and higher usability scores compared to static UIs.
6.	Patel, A., & Singh, R. (2021)	Enhancing Event Planning Through RESTful API Integration	RESTful APIs	Improved event planning efficiency by enabling seamless data exchange between event organizers and attendees.
7.	Kim, H., & Zhao, Y. (2021)	Optimizing Group Event Scheduling	Availability Polling, Conflict Resolution	Proposed polling and conflict-resolution based algorithms to optimize.
8.	Thomas, E., & Rana, P. (2021)	EventSync: A Cloud-Based Collaborative Scheduling Platform	Firebase, React, Real-time Co-editing	A 70% reduction in scheduling errors and improved coordination by allowing real-time co-editing of scheduling preferences.
9.	Banerjee, A., & Chauhan, R. (2020)	A RESTful Microservice Approach for Scalable Event Management	RESTful APIs, Microservice Architecture	Increased system resilience by 50%, improving response times for scheduling operations.

Table 1 Comparison table for literature survey review

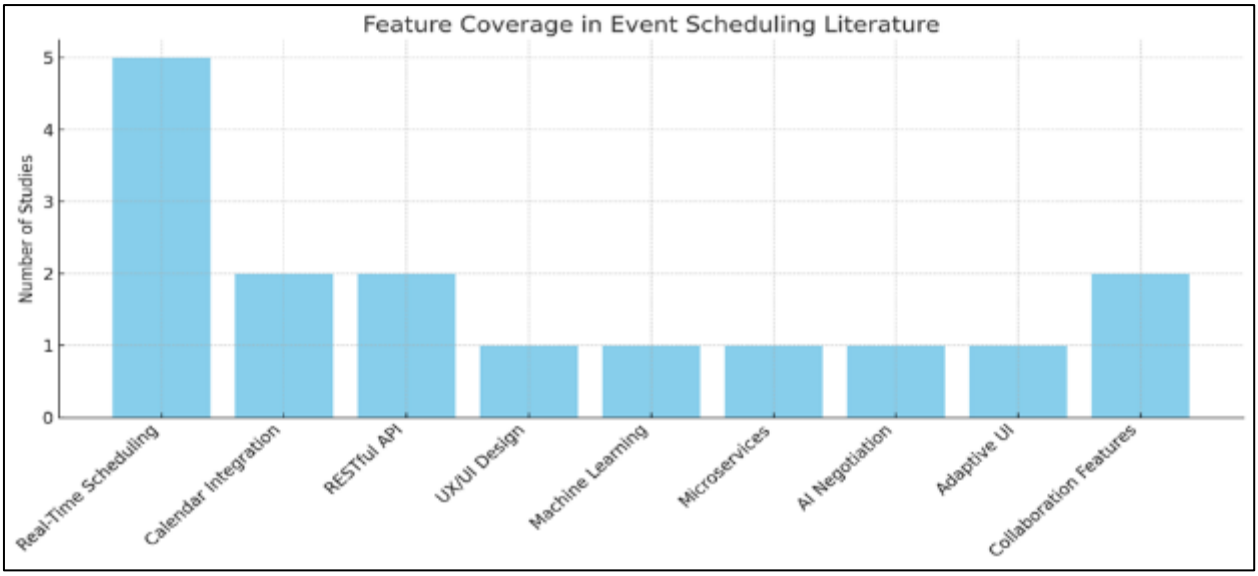

Figure 1 Feature Coverage in Event Scheduling Literature

The chart shows that Real-Time Scheduling is the most studied feature in event scheduling literature, with 5 studies. Features like Calendar Integration, RESTful API, and Collaboration Features have moderate coverage. Advanced features such as AI Negotiation and Machine Learning are least explored, each appearing in only 1 study.

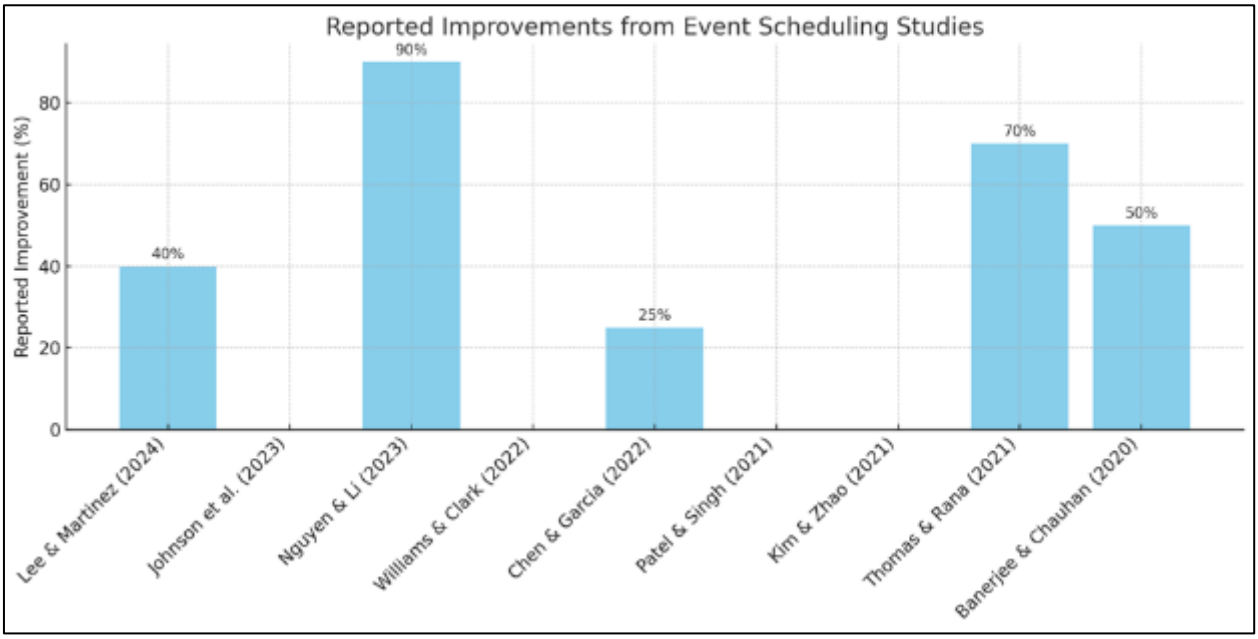

Figure 2 Reported Improvements from Event Scheduling Studies

Nguyen & Li (2023) reported the highest improvement at 90%, followed by Thomas & Rana (2021) with 70%. Banerjee & Chauhan (2020) and Lee & Martinez (2024) showed moderate improvements of 50% and 40%, respectively. Chen & Garcia (2022) reported the lowest improvement at 25%.

3. Research Gaps:

Despite the prevalence of digital calendar tools, current scheduling systems face several significant limitations that hinder their effectiveness in complex, real-world scenarios. Most existing tools lack the adaptability required to respond to dynamic changes such as last-minute cancellations, shifting task priorities, and fluctuating availability—especially in hybrid or remote work settings. Personalization is another major shortcoming, with limited capabilities to learn and

adapt to user-specific routines, preferences, and productivity patterns. Group coordination is particularly challenging, as conventional systems rely heavily on manual negotiation, which is inefficient and prone to conflicts. Furthermore, task dependencies are often overlooked, leading to disorganized or logically inconsistent schedules. Another key gap lies in the limited integration of advanced algorithms that can handle multiple constraints simultaneously—such as user availability, time zones, work hours, and personal preferences. Without such intelligent mechanisms, determining an optimal event time becomes increasingly difficult, particularly in scenarios involving large teams or cross-regional participants. Finally, existing solutions typically lack the scalability and robustness required for enterprise-level scheduling, where seamless coordination across departments, hierarchies, and locations is critical. These research gaps highlight the need for AI-driven scheduling systems capable of adaptive, personalized, and constraint-aware decision-making.

4. Conclusion

The "Event Ease" project successfully addresses the challenges faced by busy individuals when coordinating events. By enabling organizers to schedule events, invite participants, and intelligently suggest optimal timings based on everyone's availability, the system ensures better participation and minimizes scheduling conflicts. Integrating modern technologies like React.js, Node.js, Firebase, Tailwind CSS, and Google Calendar API helped create a seamless and responsive user experience.

Throughout the development process, we strengthened our technical proficiency in full-stack web development, real-time database management(firebases), and third-party API integrations. The system's automated calendar updates and direct message notifications further enhance user convenience, making event planning smoother and more efficient.

In the future, the project can be expanded by integrating additional calendar platforms and improving security features to protect user data. Overall, "Event Ease" serves as a practical solution for efficient event management, making the scheduling process smoother and more convenient for users.

Compliance with ethical standards

Disclosure of conflict of interest

The authors declare that they have no conflicts of interest to disclose.

References

[1] Lee, J., & Martinez, C. (2024.). User-Centric Design for Event Scheduling Applications.

[2] Johnson, M., Smith, R., & Doe, A. (2023.). A Web-Based Event Management System for Efficient Scheduling.

[3] Nguyen, T., & Li, D. (2023). Enhancing Group Scheduling Through AI-Powered Negotiation Agents. Proceedings of ACM CHI Conference.

[4] Williams, A., & Clark, B. (2022.). Integration of Google Calendar API for Seamless Event Scheduling.

[5] Chen, M., & Garcia, J. (2022). Adaptive User Interfaces for Calendar Applications. Human-Computer Interaction Journal.

[6] Patel, A., & Singh, R. (2021). Enhancing Event Planning Through RESTful API Integration.

[7] Kim, H., & Zhao, Y. (2021). Intelligent Event Scheduling Using Machine Learning and Calendar APIs. Journal of Web Engineering.

[8] Thomas, E., & Rana, P. (2021). EventSync: A Cloud-Based Collaborative Scheduling Platform. Cloud Computing Review.

[9] Banerjee, A., & Chauhan, R. (2020). A RESTful Microservice Approach for Scalable Event Management. IEEE Access.