

EventAI: An Intelligent Workspace for Automated Event Planning, Budget Analytics, and AI-Assisted Decision Support

Priyanshi Upadhyay *Dept. of CSE Parul University*

2403031059004@paruluniversity.ac.in

Yash Dave *Dept. of CSE Parul University*

2303031050697@paruluniversity.ac.in

Tirth Garala *Dept. of CSE Parul University*

2303031050643@paruluniversity.ac.in

Ayush Gujarati *Dept. of CSE Parul University*

2203031050212@paruluniversity.ac.in

Prof. Sanjay Pagare

Assistant Professor, Dept. of CSE Parul University

sanjay.pagare36802@paruluniversity.ac.in

Abstract—Event organisers routinely coordinate schedules, budgets, attendee records, and task lists across disconnected spreadsheets, messaging apps, and paper checklists. This fragmentation slows planning, hides budget overruns until it is too late to correct them, and leaves organisers without a single place to ask questions about their own event data. This paper presents EventAI, a unified, web-based event management workspace that combines event creation and calendar scheduling, budget and expense tracking, attendee lifecycle management, task coordination, and a conversational AI copilot inside one authenticated dash-board. The system is implemented using a React frontend and a Node.js/Express backend, with an integrated large-language-model service that answers natural-language questions about a user’s live event data and produces rule-derived planning insights (budget-utilisation warnings, attendance-engagement flags, and task-completion prompts). We describe the system’s modular architecture, the design rationale behind a single-workspace approach, and a functional evaluation of each module conducted through iterative development and manual testing. The results indicate that consolidating planning, financial tracking, and AI-assisted guidance into one workspace reduces the number of tools an organiser must reconcile and gives them a faster, clearer view of an event’s health at any point in the planning cycle.

Index Terms— *Event Management, Artificial Intelligence, Conversational Assistant, Web Application, Budget Analysis, Dashboard Design, React, Node.js*

I. INTRODUCTION

Planning an event, whether it is a corporate conference, a community fundraiser, or a private celebration, requires organisers to manage several types of information simultaneously. These include the event schedule and venue details, the budget and actual expenses, attendee registrations and confirmations, and the tasks that need to be assigned and completed within specific deadlines.

In many cases, this information is managed using separate tools. For example, the budget may be maintained in a spreadsheet, attendee details in another file, tasks through messaging applications, and the event schedule in a shared calendar. Since these tools are not connected, organisers often have to switch between multiple sources to obtain basic information. Even a simple question such as whether the event is currently over budget may require manually checking and comparing several documents.

This lack of integration leads to three recurring problems. First, budget visibility is delayed: overspending is often discovered only when a spreadsheet is manually reconciled, by which point corrective action is harder to take. Second, attendee and task status drifts out of sync across tools, so confirmations, RSVPs, and task completions are recorded inconsistently. Third, organisers have no lightweight way to ask exploratory questions about their own event data — how many attendees have confirmed, which tasks are overdue, whether Friday’s event is trending above its expected turnout — without manually querying several sources.

This paper presents EventAI, a browser-based event management workspace that consolidates event creation, a monthly calendar view, budget and expense tracking, attendee lifecycle management, task coordination, and a conversational AI assistant into a single authenticated dashboard. Rather than treating the AI assistant as a separate chatbot bolted onto an existing tool, EventAI grounds the assistant in the organiser’s own live data — its current events, budget figures, attendee counts, and task list — so that the answers it gives are specific to the workspace being viewed rather than generic event-planning advice. The primary contribution of this work is the design and implementation of an integrated pipeline that unifies planning, financial tracking, and AI-assisted guidance behind one interface, together with a derived-metrics engine that continuously recalculates budget utilisation, attendance rate, and task-completion percentage as the underlying data changes.

II. LITERATURE REVIEW

A. *Web-Based and Intelligent Event Management Platforms*

Several studies have examined web-based event management systems that centralise event creation, registration, and reporting. An AI-based event management web application allows users to search for and register for events from one place and recommends events to registered users based on their stated interests [2]. A related system built for locating and organising events across multiple formats — workshops, conferences, and festivals — addresses the difficulty of manually coordinating registration and ticketing when many similar events compete for an attendee’s attention [4], [13]. Krithika et al. propose a machine-learning-driven scheduling system that analyses participant behaviour to optimise event scheduling, reporting strong classification performance on their test data [1]. Separately, Li, Xu, and Li describe a smart event management system built on the Internet of Things and cloud computing, showing that combining sensor-level data collection with cloud-hosted processing can support more adaptive event operations [6]. A system built around generative AI similarly explores automatic reminders and resource suggestions to keep participants engaged with registered events [3].

Compared with these systems, which largely focus on discovery, registration, or scheduling optimisation in isolation, EventAI targets the operational side of running an event after it has been created — budget tracking, attendee status management, and task delegation — while still incorporating an AI layer, positioning it closer to an operations workspace than a discovery platform.

B. *Artificial Intelligence and Conversational Assistants in Planning Contexts*

Previous research has explored the growing role of artificial intelligence in the event and hospitality industry. Studies indicate that AI is being increasingly used in different stages of the event lifecycle, including planning, execution, and attendee engagement [5]. Outside the events domain specifically, research on conversational assistants for budget and financial-planning tasks is directly relevant to EventAI’s copilot design. Gupta et al. propose an evaluation framework for conversational AI

chatbots that scores systems on conversational intelligence, user experience, operational efficiency, and regulatory compliance, arguing that domain-specific evaluation matters more than generic chatbot benchmarks [7]. In the public-budget domain, Xu et al. present GRASP, a retrieval-augmented chatbot for municipal budget queries that substantially outperformed general-purpose language models on budget-specific questions by grounding its answers in the actual budget documents rather than relying purely on the model’s parametric knowledge [8]. This finding directly informs EventAI’s design choice to pass an organiser’s live event, budget, and attendee data into the assistant’s context rather than treating it as a stand-alone general-knowledge chatbot.

C. *Dashboard Design and Real-Time Analytics Visualisation*

Because EventAI surfaces budget, attendance, and task metrics through a live dashboard, its design also draws on dashboard UI/UX research. Recent research on dashboard design highlights the importance of clear presentation, effective visual hierarchy, and reduced cognitive load when developing real-time data dashboards. It suggests that key metrics should be positioned prominently so that users can identify them quickly, while additional details can be presented in a secondary layer [9]. EventAI follows this approach by displaying key indicators, such as upcoming events, total attendees, projected revenue, and task progress, at the top of the main workspace. More detailed information, including budget summaries, attendee records, and task lists, is organised within the sections below.

D *Full-Stack Web Architecture Choices*

Regarding the implementation stack, previous studies on the MongoDB–Express–React–Node (MERN) architecture have shown that using JavaScript throughout both the client and server sides can simplify development and provide a modular three-tier structure consisting of the frontend, API layer, and database [11], [12]. EventAI adopts a similar approach by using a React-based single-page application for the client and a Node.js/Express REST API on the server side. The persistence layer can also be modified according to the requirements of the deployment environment.

III. PROPOSED METHODOLOGY AND SYSTEM ARCHITECTURE

EventAI follows a client-server web application architecture. The client side is developed as a React single-page application that provides the Landing, Login, Register, and Dashboard views. On the server side, a Node.js/Express REST API manages user authentication, stores event, budget, attendee, and task-related data, and forwards requests to the AI copilot service. The overall system architecture is illustrated in Fig. 1..

A. *Event Lifecycle and Calendar Module*

To create an event, organisers enter details such as the event title, date and time, expected number of attendees, and estimated revenue. Once created, the event appears in the upcoming-events list as well as in a monthly calendar view. The calendar highlights the current date and displays the attendee count for days on which an event is scheduled. Users can navigate across different months and years and return to the current date when needed, allowing organisers to plan events several months in advance while maintaining an overview of their schedule..

B. *Budget and Expense Tracking Module*

Each workspace has a configurable total budget that serves as the reference point for tracking individual expenses. Each expense is recorded with its category, description, and amount. The system automatically calculates the total expenditure, remaining budget, and percentage of the budget used. The utilisation percentage is displayed through a progress indicator, allowing organisers to quickly assess how much of the allocated budget has been spent. Individual expenses can also be removed, while the remaining-budget value is visually highlighted when the total expenditure exceeds the allocated budget.

C. *Attendee Management Module*

Attendee records include the participant’s name, email address, and current status, which can be set to Pending, Confirmed, or Attended. The system automatically maintains the count of attendees in each status category and provides a search function for locating participants by name or email address. Organisers can update attendee status directly within the list as RSVPs are received or check-ins are completed. These updates are reflected in the summary dashboard, ensuring that attendance figures remain current without requiring a separate reconciliation process..

D. *Task and Timeline Management Module*

Tasks can be added by specifying a title, assigning a responsible person, and selecting a priority level of Low, Medium, or High. Each task moves through three stages: Pending, In Progress, and Completed. The system calculates the overall completion percentage based on the status of all tasks and presents it using a progress bar. This provides organisers with a quick overview of the event’s preparation status without having to review each task individually.

E. *AI Copilot Module*

The AI copilot is integrated into the dashboard through a built-in chat interface. Messages entered by users are sent to a backend endpoint, which forwards them to a large language model API along with relevant information from the organiser’s event, budget, attendee, and task records. This retrieval-grounded approach provides the model with application-specific context and has been shown to perform better than ungrounded chatbots in domain-specific applications [8]. The assistant can respond to questions related to budgets and attendance, create planning checklists based on the size of an event, and identify potential inconsistencies in the available data. For instance, it may highlight events scheduled within a short time of each other or identify a revenue value that differs significantly from similar events.

F. *Analytics and Rule-Based Insight Engine*

In addition to the conversational assistant, EventAI calculates several derived metrics each time the dashboard is loaded. These include the average event size, average projected revenue, attendance rate, and percentage of the budget used. These metrics are then processed by a lightweight rule-based insight generator, which produces brief recommendations in plain language. For example, the system can highlight cases where budget utilisation exceeds 80%, attendance confirmations fall below 70%, or task progress is slower than expected. Since these insights are generated locally without sending additional requests to the AI service, the dashboard remains responsive even when the AI copilot is not being used..

G. *Dashboard Visualisation Layer*

The presentation layer is designed according to dashboard design principles that place the most important information prominently at the top, while providing supporting details in the sections below [9]. The first row of the dashboard displays four key metrics: upcoming events, total attendees, projected revenue, and task progress. Below these metrics, a twelve-column grid is used to organise the main functional modules, including event creation, the AI copilot, calendar, budget management, attendee management, task tracking, and analytics. The layout positions related modules close to one another, making it easier for users to move between key metrics and the controls associated with them. This arrangement reduces unnecessary navigation and improves the overall usability of the dashboard..

IV. RESULTS AND ANALYSIS

EventAI was functionally implemented and evaluated throughout the development process rather than through a formal user study, considering the scope and objectives of the project. Each module was manually tested using representative sample data, including events, expenses, attendees, and tasks. These tests were used to verify that create, update, and delete operations correctly refreshed the associated derived metrics, such as budget utilisation, attendance rate, and task progress, EventAI was functionally implemented and evaluated throughout the development process rather

than through a formal user study, considering the scope and objectives of the project. Each module was manually tested using representative sample data, including events, expenses, attendees, and tasks. These tests were used to verify that create, update, and delete operations correctly refreshed the associated derived metrics, such as budget utilisation, attendance rate, and task progress, in real time. The tests also confirmed that changes were reflected on the dashboard without requiring a page refresh. Table I presents the functional status of each module observed during the evaluation. in real time. The tests also confirmed that changes were reflected on the dashboard without requiring a page refresh. Table I presents the functional status of each module observed during the evaluation.

TABLE I
MODULE-LEVEL FUNCTIONAL EVALUATION SUMMARY

System Module	Status
Event & Calendar	Stable
Budget & Expense	Stable
Attendee Management	Stable
Task & Timeline	Stable
AI Copilot	Functional
Analytics & Insights	Stable

- **Event & Calendar:** Newly created events are immediately reflected in both the event list and the monthly calendar view.
- **Budget & Expense:** The system recalculates the budget utilisation percentage and remaining budget whenever an expense is added or removed.
- **Attendee Management:** Changes to attendee status automatically update the summary counts and search results without requiring a page reload.
- **Task & Timeline:** The overall task completion percentage is updated whenever a task’s status is changed.
- **AI Copilot:** The assistant provides context-aware responses to budget, checklist, and attendance-related queries. The quality of these responses depends on the underlying large language model API.
- **Analytics & Insights:** Rule-based insights are updated immediately when the underlying data changes and remain available independently of the AI service.

Two observations follow from this evaluation. First, decoupling the rule-based insight engine from the AI copilot proved useful in practice: because the insight cards recompute directly from local state, the dashboard remains informative even if the AI service is slow or temporarily unavailable, while the copilot provides deeper, free-form answers when a network round trip is acceptable. Second, consolidating budget, attendee, and task data behind a single set of derived metrics — rather than computing them independently in each module — reduced the chance of the dashboard showing inconsistent figures between, for instance, the top-level attendee count and the attendee module’s own summary.

A limitation of this evaluation is that it reflects functional and usability testing during development rather than a controlled study with independent event organisers, so claims about time savings or planning-quality improvements should be treated as motivating hypotheses for future evaluation rather than established results.

V. CONCLUSION

This paper presented EventAI, a unified web-based workspace that brings event creation, calendar scheduling, budget and expense tracking, attendee management, task coordination, and an

AI-assisted conversational copilot into a single dashboard. The system’s core contribution is architectural: rather than adding AI as a separate feature, EventAI grounds its assistant and its rule-based insight engine in the organiser’s own live data, so that both the automated suggestions and the conversational answers a user receives are specific to the event actually being planned. Functional evaluation across the development cycle indicates that the modules interoperate correctly and that derived metrics remain consistent across the dashboard as underlying data changes. The approach demonstrates that consolidating planning, financial tracking, and AI guidance into one interface is both technically feasible with a standard React/Node.js stack and, based on the design principles it draws on, likely to reduce the coordination overhead that fragmented tool use currently imposes on event organisers.

4. “Intelligent Event Finder and Management System,” *IEEE Conference Publication*, IEEE Xplore, Doc. 9640719, 2021.
5. “The Transformative Role of Artificial Intelligence in the Event Management Industry,” ResearchGate, Dec. 2023.
6. K. Li, L. Xu, and S. Li, “A Smart Event Management System Based on Internet of Things and Cloud Computing,” *IEEE Access*, vol. 7, pp. 153584–153593, 2019.
7. S. Gupta, R. Ranjan, and S. N. Singh, “Comprehensive Framework for Evaluating Conversational AI Chatbots,” arXiv:2502.06105, 2025.
8. J. Xu, J. Wang, J. Leung, and J. Gu, “GRASP: Municipal Budget AI Chatbots for Enhancing Civic Engagement,” arXiv:2503.23299, 2025.
9. “Best Practices for Dashboard UI/UX to Enhance User Experience,” ResearchGate, May 2025.
10. “AI Event Management System,” ResearchGate, June 2026.
11. “Harnessing the MERN Stack for Scalable E-Commerce Website De-sign: A Full-Stack Approach with MongoDB, Node.js, Express.js, and React.js,” ResearchGate, Apr. 2025.
12. “Full Stack Web Application Using MERN,” *International Journal of Progressive Research in Engineering Management and Science*, vol. 2, no. 8, pp. 121–131, Aug. 2022.
13. “A Prototype for Efficient and Intelligent Event Management for Mobile Services,” *IEEE Conference Publication*, IEEE Xplore, Doc. 4446573.

VI. FUTURE WORK

- **Retrieval-grounded copilot responses:** extend the AI copilot with a retrieval-augmented generation layer over the organiser’s historical event records, following the approach shown to improve domain-specific factual accuracy in budget-chatbot research [8], so that answers can reference exact past figures rather than summarised context.
- **Predictive budget forecasting:** apply lightweight machine-learning models to historical expense and revenue data to forecast likely overspend before it occurs, rather than only flagging utilisation after the fact.
- **Calendar interoperability:** support two-way synchronisation with external calendar services so that event schedules created in EventAI are reflected in an organiser’s existing calendar tools.
- **Notification and reminder system:** add automated reminders for pending tasks, unconfirmed attendees, and approaching budget thresholds, reducing reliance on an organiser manually checking the dashboard.
- **Multi-organiser collaboration:** extend the single-workspace model to support role-based access for teams planning the same event together, with per-user activity logs.
- **Mobile companion application:** package the dashboard’s core monitoring views as a lightweight mobile client for on-the-go status checks during live events.

REFERENCES

1. D. R. Krithika *et al.*, “Smart Event Management System Using Machine Learning for Optimized Scheduling and Participant Behavior Tracking,” *IEEE Conference Publication*, IEEE Xplore, 2025.
2. “AI based Event Management Web Application,” *IEEE Conference Publication*, IEEE Xplore, Doc. 9850551, 2022.
3. “Event Management System Using Generative AI,” *IEEE Conference Publication*, IEEE Xplore, Doc. 10673057, 2024.