Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

AI-Powered Event Management with Smart Quotation and Automated GST Invoicing

Ishaan Abhijeet Hingway, Vedant Nilesh Chavan, Pratik Eknath Ravan, Varada Vitthal Pawar, Simoni Surendra Sawant

Department of Computer Science & Engineering, D. Y. Patil College of Engineering & Technology,

Kasaba Bawada, Kolhapur, Maharashtra, India

Academic Year 2026–27 | Group G12 | Guide: Prof. M. A. Pardesi

Abstract - Event management workflows often depend on manual quotation preparation, inventory checking, booking coordination, payment tracking, and invoice generation. This paper presents an AI-powered event management system that combines smart quotation generation with deterministic business rules for booking, payment, GST calculation, and invoicing. The system uses a Next.js frontend, NestJS backend, PostgreSQL hosted on Supabase, Prisma ORM, Google Gemini 2.5 Flash, and Razorpay Sandbox. Customers provide event type, area, and selected requirements; the backend retrieves authoritative catalog data and sends structured context to the AI module. Gemini returns a validated quotation containing suggested items, pricing, GST information, a budget alternative, and a plain-language explanation. A rule-based fallback engine is used when the AI service encounters an API error, timeout, rate limit, or schema-validation failure. Financial calculations remain server-side and use decimal arithmetic. Implementation verification includes 73 passing backend unit tests across 10 suites, 10 AI quotation scenarios, four fallback conditions, ten concurrent invoice-number requests with zero duplicates, a successful frontend production build across nine routes, and an end-to-end booking, sandbox-payment, GST-invoice, and PDF-download workflow. The results demonstrate the feasibility of combining generative AI with deterministic validation and transaction logic for a practical event-management application.

Keywords - automated invoicing, event management, generative AI, GST, smart quotation, vendor management

I. INTRODUCTION

Event-service businesses commonly coordinate venue requirements, decoration, catering, equipment, quotations, bookings, payments, and tax invoices through disconnected manual processes. Such workflows can make quotation preparation slow, make inventory information difficult to synchronize, and increase the possibility of pricing or billing inconsistencies. The project addressed in this work proposes a centralized web-based event management platform that combines customer event configuration, catalog management, booking approval, AI-assisted quotation generation, payment processing, GST calculation, invoicing, and notifications.

Recent research has explored generative AI for event-management applications, machine learning for event scheduling and participant analysis, AI-assisted security and authentication, and large-language-model-based decision support [1]–[5]. These studies establish the usefulness of intelligent methods in event-related workflows, but the present project focuses on an integrated operational workflow in which AI recommendations are connected to authoritative catalog data and followed by deterministic booking and financial processing.

The main contribution of the proposed system is therefore not the use of a language model alone, but the integration of generative quotation assistance with controlled application logic. AI output is constrained by structured input and schema validation, while catalog prices, booking states, GST calculations, invoice numbering, and payment verification remain under server-side control.

II. LITERATURE SURVEY

Thummala et al. presented an event management system using generative AI to support event-related content, participation, reminders, and user engagement [1]. The work demonstrates the applicability of generative AI to event-management interfaces, but its emphasis is on user engagement and event assistance rather than a complete quotation-to-invoice business workflow.

Krithika and Subhasri proposed a smart event-management system using machine learning for optimized scheduling and participant behavior tracking [2]. Their evaluation reports classification-oriented performance measures, showing the value of machine learning for event scheduling and behavioral analysis. The proposed system in this paper instead concentrates on operational purchasing and vendor-service workflows.

Jensina et al. investigated AI-based authentication techniques for an event organization system and described a centralized platform integrating cloud, AI, security, and event operations [3]. This motivates the importance of centralized and secure access control, which is incorporated here through JWT authentication, role-based authorization, validation, and server-side protection.

Wang and Wu studied ChatGPT as a multi-criteria decision maker for supplier evaluation and compared model-based evaluations with expert assessments [4]. This work supports the use of language models for recommendation and decision-support tasks, while the present implementation adds deterministic validation because generated text must not directly control financial transactions.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

Smirnov, Levashova, and Ponomarev examined LLM-assisted decision support using contextual problem modeling, ontology-driven knowledge representation, and computational decision components [5]. The work highlights the need for domain knowledge, user preferences, alternative generation, and explainable decision support. The proposed event-management system follows a related principle by supplying catalog context to the AI module and returning both a quotation and a plain-language explanation.

The literature indicates a gap between intelligent event assistance and an end-to-end operational system that links AI-assisted recommendations with inventory, booking state management, payment verification, GST computation, and invoice generation. The proposed system is designed around this integration.

III. PROBLEM STATEMENT

Event-service operations require customers and staff to repeatedly exchange event requirements, item selections, quantities, prices, availability information, booking decisions, payment status, and tax documents. When these activities are handled manually or through disconnected tools, quotation turnaround time and operational consistency can suffer. The problem is to develop a centralized event-management system that can generate context-aware quotations while preserving deterministic and auditable control over catalog pricing, booking transitions, payments, GST calculations, invoice numbering, and invoice generation.

IV. PROPOSED METHODOLOGY

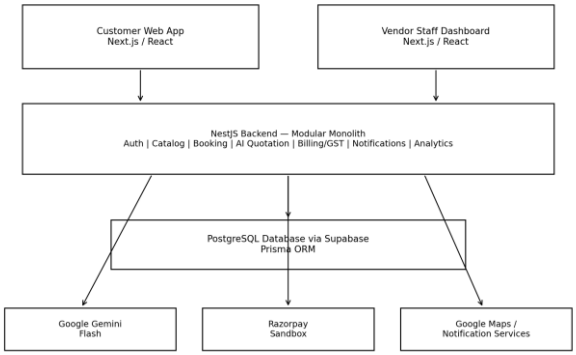


Fig. 1. Overall system architecture of the proposed event-management platform.

A. Event Configuration and Catalog Retrieval

A customer selects an occasion, event area, and optional catalog items or packages. The backend retrieves current catalog records from the database rather than trusting client-provided prices. This keeps quotation inputs aligned with the authoritative inventory and pricing data.

B. AI-Based Smart Quotation

The quotation endpoint accepts event requirements and retrieves relevant catalog information on the server. The backend constructs a structured prompt containing event context and catalog values and sends it to the Gemini 2.5 Flash model. The requested response is structured JSON containing a suggested package, itemized pricing, GST-related information, an alternative budget option, and a plain-language explanation. The response is parsed and validated against a Zod schema before it reaches the application layer. Consequently, unstructured or malformed model output cannot directly drive booking or billing logic. The frontend presents the quotation through an AI Smart Quotation Assistant and allows selected items to be applied to the booking.

C. Rule-Based Fallback

Because external AI services can fail or become unavailable, the system implements a deterministic fallback quotation engine. Four failure conditions are explicitly handled: API error, response timeout, rate limit, and schema-validation failure. The fallback uses catalog data and deterministic rules to compute quantities, subtotal, GST, and a budget alternative. Its response shape is standardized so that the user interface can display either AI-generated or fallback results consistently.

D. GST and Invoice Processing

Financial processing is performed on the server. For each invoice line i, the line amount is calculated as $q_i p_i$, where q_i is quantity and p_i is unit price. The subtotal is the sum of all line amounts. GST is then aggregated by applicable tax rate using decimal arithmetic, and the grand total is the subtotal plus total GST.

Invoice numbers use an atomic sequence in the form INV-YYYY-00000. The implementation uses decimal monetary types rather than floating-point arithmetic and verifies invoice generation under concurrent requests to reduce race-condition risk.

E. Booking and Payment Workflow

Bookings follow controlled status transitions. Staff can review, approve, modify, or reject a customer request, while invalid transitions are rejected by the booking state machine. After approval, the customer can use the Razorpay Sandbox flow. Payment verification updates the booking to confirmed and triggers GST invoice generation. The customer can then retrieve invoice details and download the generated PDF.

F. Security and Reliability

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

The implementation uses JWT-based authentication, role-based guards, bcrypt password hashing, input validation, sanitization, rate limiting on sensitive endpoints, CORS restrictions, and server-side secrets. Sensitive keys such as the Gemini API key and payment secrets are kept on the backend. Client-submitted totals are not treated as authoritative financial values.

V. IMPLEMENTATION

The implementation uses Next.js 14+ with React and Tailwind CSS for the frontend and NestJS with Node.js and TypeScript for the backend. PostgreSQL is hosted through Supabase, with Prisma used for database access and migrations. Authentication is implemented with NestJS-native JWT and Passport, with bcrypt for password hashing. The AI layer uses the Google Gemini API and the tested payment integration uses Razorpay Sandbox. PDFKit is used for server-side GST invoice PDF generation.

Layer	Technology / Role
Frontend	Next.js / React, Tailwind CSS
Backend	NestJS, Node.js, TypeScript
Database	PostgreSQL via Supabase
ORM	Prisma
Authentication	JWT, Passport, bcrypt
AI	Google Gemini 2.5 Flash
Payment	Razorpay Sandbox
Invoice PDF	PDFKit

TABLE I. IMPLEMENTED TECHNOLOGY STACK

The database contains users/staff, catalog items, packages, bookings, booking items, invoices, invoice lines, payments, GST rates, and notifications. The live cross-role verification used a Supabase-hosted PostgreSQL instance and confirmed synchronization between catalog, booking, payment, and invoice operations.

VI. RESULTS AND DISCUSSION

The implemented system was evaluated using unit tests, controlled failure tests, concurrency checks, build verification, and an end-to-end functional workflow. The results below describe the current implementation state and are intended as functional verification rather than a large-scale performance benchmark.

Evaluation Item	Observed Result	Interpretation
Backend unit tests	73 passed / 10 suites	Core services and guards passed
AI quotation scenarios	10 scenarios passed	Structured quotation output verified
AI fallback conditions	4 conditions passed	Graceful fallback verified
Concurrent invoice requests	10 requests / 0 duplicates	Atomic numbering verified
Frontend build	9 routes / 0 TS-lint errors	Production build successful
Frontend secret audit	0 sensitive API-key matches	Secrets remained server-side

TABLE II. FUNCTIONAL AND QUALITY VERIFICATION RESULTS

Ten distinct AI quotation scenarios were tested across event types and areas, and all four planned fallback conditions were explicitly exercised. The fallback engine returned non-error quotation responses with the expected source designation. The security audit found no Gemini API key references in the frontend codebase.

GST processing was tested for single-rate and mixed-rate cases. The implementation uses Decimal arithmetic and aggregates GST by rate slab. Ten concurrent invoice-number requests produced zero duplicate numbers, supporting the use of atomic numbering under concurrent generation.

A complete cross-role workflow was also verified using live Supabase persistence: customer registration, staff login, catalog retrieval, booking creation, staff review and approval, customer status retrieval, Razorpay Sandbox order creation, payment verification, GST invoice generation, and PDF download. One verified invoice had a subtotal of ₹145,500.00, GST of ₹26,190.00, and a grand total of ₹171,690.00.

The results indicate that the system can combine flexible AI assistance with deterministic application rules. The strongest reliability feature is the separation between recommendation and transaction logic: the language model proposes a structured quotation, while the server remains responsible for authoritative prices, GST, booking state, payment verification, and invoice numbering. However, the present evaluation does not establish user-perceived quotation quality, production-scale throughput, or long-term pricing accuracy. Those require larger datasets, user studies, and production deployment.

VII. CONCLUSION AND FUTURE SCOPE

This paper presented an AI-powered event management platform integrating smart quotation generation with catalog management, booking workflows, payment verification, GST computation, automated invoice generation, and notifications. The implementation demonstrates a practical architecture in which generative AI is constrained by server-fetched data, structured output validation, and deterministic fallback logic. Testing confirmed 73 passing backend unit tests, ten AI quotation scenarios, four fallback conditions, zero duplicate invoice numbers across ten concurrent requests, a successful frontend production build, and an end-to-end booking-to-invoice workflow.

Future work can extend the system toward multi-vendor marketplace capabilities, vendor ranking, historical-data-based demand and pricing prediction, optimization-based quotation selection, richer retrieval-augmented domain knowledge, production payment and communication

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

integrations, and formal user studies measuring quotation usefulness and decision quality. Deployment through the planned hosting and CI/CD pipeline also remains a future operational step.

ACKNOWLEDGMENT

The authors express their gratitude to D. Y. Patil College of Engineering & Technology, Department of Computer Science & Engineering, for providing the academic environment and project facilities. The authors sincerely thank Prof. M. A. Pardesi for guidance, technical direction, and continuous support during the development and evaluation of the project. The team also acknowledges the project stakeholders and sponsor-side inputs that helped define the event-management workflow.

REFERENCES

[1] S. Thummala, S. Thammishetti, S. Varkol, A. Thirunahari, and V. V. S. L. Kanthey, “Event Management System Using Generative AI,” in Proc. 7th Int. Conf. Circuit Power and Computing Technologies (ICCPCT), Kollam, India, 2024, pp. 624–628, doi: 10.1109/ICCPCT61902.2024.10673057.

[2] D. R. Krithika and V. M. Subhasri, “Smart Event Management System Using Machine Learning for Optimized Scheduling and Participant Behavior Tracking,” in Proc. Int. Conf. Computational Robotics, Testing and Engineering Evaluation (ICCRTEE), Virudhunagar, India, 2025, doi: 10.1109/ICCRTEE64519.2025.11052976.

[3] S. Giftlin Jensina, B. Shamreen Ahamed, V. Amutha Mary, M. S. Godwin Premi, D. Susitra, and K. Dhanalakshmi, “Authentication Techniques on Event Management System (EMS) using Artificial Intelligence,” in Proc. 7th Int. Conf. Intelligent Sustainable Systems (ICISS), 2025, doi: 10.1109/ICISS63372.2025.11076271.

[4] X. Wang and X. Wu, “Can ChatGPT Serve as a Multi-Criteria Decision Maker? A Novel Approach to Supplier Evaluation,” in Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP), Seoul, Republic of Korea, 2024, pp. 10281–10285, doi: 10.1109/ICASSP48485.2024.10447204.

[5] A. V. Smirnov, T. Levashova, and A. Ponomarev, “Decision Support Based on Large Language Models: Ontology-Based Architecture and Generalized Scenario,” in Proc. 39th Conf. Open Innovations Association (FRUCT), Helsinki, Finland, 2026, doi: 10.23919/FRUCT70069.2026.11506563.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)