Date	15 April 2025
Team ID	SWTID1743599215
Project Name	EventEase: Event Management

1. PROJECT DESIGN

1.1 Problem Solution Fit

Event planning, once a straightforward process, has become increasingly complex with the advent of digital platforms and hybrid formats. The primary concern today is not just hosting an event, but ensuring smooth communication, participant engagement, automated logistics, and integrated data handling. Organizers are burdened with managing multiple platforms—each serving one function such as ticketing, emails, or attendee management—leading to fragmentation and inefficiency.

Event attendees, on the other hand, seek intuitive, fast, and transparent interactions. The delay in confirmations, manual errors in registrations, and scattered communication often lead to a poor user experience.

In this environment, a consolidated event management system is not just a convenience but a necessity. The absence of such solutions highlights a critical gap in the market—a gap InVITe aims to fill.

Our core hypothesis rests on delivering an all-in-one solution that addresses pain points across the event lifecycle, from creation to conclusion. By integrating ticketing, registration, email communication, and role-based access within a single, easy-to-use platform, InVITe achieves strong problem-solution alignment.

1.2 Proposed Solution

The proposed solution, InVITe, is a cloud-based web application that offers a seamless experience for event organizers and attendees. The solution is structured around a triad of core user types—Users, Admins, and Developers—each provided with customized interfaces and functionalities.

The User Interface (UI) offers a dashboard for discovering and registering for events. It ensures fast navigation, easy registration, and instant feedback through real-time notifications and email alerts.

The Admin Interface includes tools for creating new events, viewing existing registrations, managing ticketing, and sending event updates. It features a comprehensive analytics module (under

development) that will allow organizers to gain insight into engagement trends.

Developers are granted access to configuration and system-level controls including admin creation and role management. While this role is currently open for system configuration, future versions will restrict this to super-admins or product managers for enhanced security.

Features in detail include:

• Event Creation Wizard: Enables quick event setup with fields for title, date, description, tickets, and images.

• Ticket Booking System: Fully integrated with Stripe, supporting secure payments and real-time transaction validation.

• Email Notification Pipeline: Automatic dispatch of transactional and promotional emails using NodeMailer.

• Attendee Monitoring: Real-time dashboard displaying attendee count, time stamps for check-ins, and drop-off metrics.

• Admin Management System: Secure login, JWT-based session control, and multi-admin functionality to allow collaboration.

This comprehensive stack ensures not only a superior experience for all stakeholders but also establishes a strong foundation for future enhancements like QR code scanning, live chat modules, and AI-based recommendation systems for events.

1.3 Solution Architecture

The architecture of InVITe is designed for scalability, maintainability, and modularity. It employs a microservice-oriented design that separates the frontend, backend, and database layers, each communicating through secure API endpoints.

• Frontend: Built using Next.js, the frontend provides dynamic rendering and optimized loading. Tailwind CSS ensures a responsive, elegant UI adaptable across devices.

• Backend: Node.js with Express.js serves as the backend framework. The backend manages routing, authentication, and data processing. APIs are secured using JWT tokens and input validations to prevent unauthorized access.

• Database: MongoDB (hosted on MongoDB Atlas) stores event data, user credentials, registration details, and payment records. Indexing and optimized queries ensure fast data retrieval.

• Payment Gateway: Stripe integration supports real-time, secure financial transactions. The system uses public/private keys to manage transactions, and webhooks to handle callbacks and status updates.

• Email System: NodeMailer is used with Gmail SMTP to send confirmation emails and updates. Emails are templated for branding consistency and usability.

• Environment Configuration: All sensitive information such as API keys and secrets are stored in .env files. These are excluded from version control to maintain security.

Future upgrades to this architecture may include Redis for caching, RabbitMQ for message queuing, and Docker-based containerization for CI/CD pipelines.