A Full-Stack Event Management System Using AI (Artificial Intelligence)

by

Nazmus Sakib
ID: CSE2201025165

Suriya Akter
ID: CSE2201025145

Md. Sakib Miaji
ID: CSE2201025129

Himel Deb
ID: CSE2201025137

Chandan Basu
ID: CSE2201025118

Supervised by
Salma Tabashum

Submitted in partial fulfilment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SONARGAON UNIVERSITY (SU)

January 2026

APPROVAL

The project titled **"A Full-Stack Event Management System Using AI (Artificial Intelligence)"** submitted by **Nazmus Sakib** (CSE2201025165), **Suriya Akter** (CSE2201025145), **Md. Sakib Miaji** (CSE2201025129), **Himel Deb** (CSE2201025137) and **Chandan Basu** (CSE2201025118) to the Department of Computer Science and Engineering, Sonargaon University (SU), has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering and approved as to its style and contents.

Board of Examiners

Salma Tabashum — **Supervisor**
Lecturer & Asst. Coordinator
Department of Computer Science and Engineering
Sonargaon University (SU)

Examiner 1
(Examiner Name and Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

Examiner 2
(Examiner Name and Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

Examiner 3
(Examiner Name and Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

DECLARATION

We, hereby, declare that the work presented in this report is the outcome of the investigation performed by us under the supervision of **Salma Tabashum,** Lecturer & Asst. Coordinator**,** Department of Computer Science and Engineering, Sonargaon University, Dhaka, Bangladesh. We reaffirm that no part of this thesis has been or is being submitted elsewhere for the award of any degree.

Countersigned

(Salma Tanashum)
Supervisor

Signature

Nazmus Sakib
ID: CSE2201025165

Suriya Akter
ID: CSE2201025145

Md. Sakib Miaji
ID: CSE2201025129

Himel Deb
ID: CSE2201025137

Chandan Basu
ID: CSE2201025118

ABSTRACT

Event management in Bangladesh has traditionally relied on **manual processes**, including spreadsheets, WhatsApp groups, and basic web forms, leading to fragmented information, data silos, and significant inefficiencies. Organizers face the challenge of coordinating multiple aspects of an event simultaneously—scheduling sessions, managing attendees, tracking speakers, allocating resources, and analyzing feedback—all of which demand high cognitive effort and often result in errors or delays.

This project proposes an **A Full-Stack Event Management System using AI (Artificial Intelligence)** designed to address these challenges by providing a **proactive, intelligent, and automated platform** for event orchestration. By leveraging artificial intelligence, the system transforms conventional data storage tools into an **active collaborator**, assisting organizers in decision-making, content generation, predictive planning, and resource optimization.

The system is built using a **modern technology stack**: a React/Next.js frontend for dynamic user interaction, Node.js backend for RESTful services, and Python for AI model development. The **Groq API** powers high-speed AI inference, enabling low-latency recommendations and analytics. Database management is handled with MongoDB and PostgreSQL, supporting structured event data and AI training datasets. A custom `bd_divisions.js` dataset provides localized support for all Bangladeshi divisions and districts, addressing the limitations of global platforms.

Preliminary evaluation indicates a **40% improvement in planning efficiency**, allowing event organizers to handle larger events with the same resources, reduce manual errors, and improve attendee satisfaction. This project demonstrates that **AI is not merely a supplementary tool**, but a transformative enabler for modern event management, bridging the gap between manual administrative effort and intelligent, proactive orchestration.

In conclusion, this AI-powered platform represents a **significant advancement** in event management for Bangladesh, providing a scalable, localized, and intelligent solution that can serve as a benchmark for future AI-SaaS implementations in the South Asian region. The system lays the foundation for further enhancements, including AR/VR venue previews, blockchain-based ticketing, and voice assistant integration, which could further revolutionize the event management industry.

KEYWORDS

Artificial Intelligence (AI), Machine Learning (ML), Natural Language Processing (NLP), Recommendation Systems, Predictive Analytics, Event Management, Automated Scheduling, Full-Stack Development, React, Next.js, Node.js, Python, Groq API, Database Management, MongoDB, PostgreSQL, Sentiment Analysis, Feature Engineering, Constraint Optimization, User Interface (UI), User Experience (UX), SaaS (Software as a Service), Bangladesh Localized Data, Data-Driven Decision Making

ACKNOWLEDGMENT

At the very beginning, we would like to express my deepest gratitude to the Almighty Allah for giving us the ability and the strength to finish the task successfully within the schedule time. We are auspicious that we had the kind association as well as supervision of **Brig. Gen. (Retd) Prof. Habibur Rahman Kamal, ndc, psc,** Dean**,** Faculty of Science & Engineering and **Salma Tabashum**, Lecturer & Asst. Coordinator, Department of Computer Science & Engineering Sonargaon University whose hearted and valuable support with best concern and direction acted as necessary recourse to carry out our project.

We are also thankful to all our teachers during our whole education, for exposing us to the beauty of learning.

Finally, our deepest gratitude and love to my parents for their support, encouragement, and endless love.

LIST OF ABBREVIATIONS

Abbreviation	Full Form / Meaning
AI	Artificial Intelligence
ML	Machine Learning
NLP	Natural Language Processing
LSTM	Long Short-Term Memory
CI/CD	Continuous Integration / Continuous Deployment
API	Application Programming Interface
JWT	JSON Web Token
OAuth	Open Authorization
ERD	Entity-Relationship Diagram
SaaS	Software as a Service
CSV	Comma-Separated Values
JSON	JavaScript Object Notation
REST	Representational State Transfer
IoT	Internet of Things
GPU	Graphics Processing Unit
MVC	Model-View-Controller
MERN	MongoDB, Express.js, React, Node.js
CRUD	Create, Read, Update, Delete
API	Application Programming Interface
IDE	Integrated Development Environment
VS Code	Visual Studio Code
SQL	Structured Query Language
DBMS	Database Management System
GPU	Graphics Processing Unit
NN	Neural Network
JSON	JavaScript Object Notation
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
CSV	Comma Separated Values
AI/ML	Artificial Intelligence / Machine Learning
API	Application Programming Interface
RESTful	REST-based web service
DevOps	Development and Operations
MVP	Minimum Viable Product
DAG	Directed Acyclic Graph
CDN	Content Delivery Network

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Background & Problem Statement

The event management industry in Bangladesh is currently undergoing a critical phase of digital transformation. Over the past decade, event organizers have gradually shifted from paper-based ledgers and manual registers to digital formats. While this transition has improved data accessibility and record preservation, it has largely stalled at the level of **passive digitalization**. Most existing tools—such as shared Excel spreadsheets, messaging applications, and basic web-based registration platforms—function primarily as static data repositories rather than intelligent management systems.

These tools operate as **digital filing cabinets**, requiring extensive manual data entry, continuous human supervision, and repetitive administrative effort. They provide little to no analytical insight, predictive capability, or automated decision support. As a result, event organizers are burdened with tasks such as schedule coordination, attendee tracking, venue management, and communication handling—responsibilities that are increasingly complex in modern event environments.

In parallel, the nature of events in Bangladesh has evolved significantly. Corporate conferences, educational seminars, cultural programs, and hybrid events now demand real-time coordination, dynamic scheduling, multi-channel communication, and rapid adaptation to unforeseen changes. The reliance on generic administrative tools creates inefficiencies, information fragmentation, and coordination breakdowns, which directly affect event quality, operational reliability, and participant experience.

This project, **A Full-Stack Event Management System using AI (Artificial Intelligence)**, emerges from the recognition of a fundamental gap between conventional digital tools and the high-intensity creative and logistical demands of contemporary event orchestration. The core problem lies in the absence of **intelligent systems that can transform raw event data into actionable insights**. Rather than merely storing information, modern event platforms must actively assist organizers by analyzing data patterns, automating routine decisions, and supporting real-time operational adjustments.

By addressing this gap, the proposed system aims to shift event management in Bangladesh from **data storage** to **data intelligence**, enabling smarter planning, reduced manual overhead, and more resilient, efficient, and scalable event execution.

1.1.1 The Complexity of Modern Event Coordination

By 2026, event coordination has evolved into a highly complex, technology-driven process. Events are no longer limited to physical venues and paper-based management; instead, they operate as **multi-dimensional systems** integrating physical, digital, and real-time data streams. Even a small-scale corporate seminar in Dhaka now typically requires **real-time speaker availability tracking**, **online and on-site digital registration systems**, **automated attendee notifications**, **venue capacity monitoring**, and **dynamic scheduling adjustments**.

Modern events often involve multiple stakeholders—organizers, speakers, vendors, sponsors, and participants—each interacting through different platforms. When coordination relies on **legacy tools** such as Excel spreadsheets, WhatsApp groups, email threads, or static web forms, information becomes

fragmented across disconnected systems. This fragmentation creates **data silos**, where critical data such as attendee lists, schedule updates, or speaker changes are stored in separate locations without real-time synchronization.

As a result, organizers frequently face **data inconsistencies**, including outdated schedules, duplicate registrations, missed communications, and conflicting information across platforms. These synchronization errors not only increase administrative workload but also raise the risk of operational failures during live events. In high-stakes environments—such as corporate conferences, academic symposiums, or hybrid events—such inefficiencies can negatively impact attendee experience, organizational credibility, and overall event success.

Therefore, the growing complexity of modern event coordination highlights the urgent need for **integrated, intelligent event management systems** capable of centralizing data, automating workflows, and ensuring real-time consistency across all event-related operations.

1.1.2 The Need for Automation and Intelligence

The increasing scale and complexity of modern events have made traditional, reactive management approaches insufficient. What is critically required is a **proactive event management system**—one that does not merely record information after human input, but actively anticipates needs, supports decisions, and adapts in real time. Automation in this context must extend beyond basic data entry or workflow triggers; it must incorporate **intelligence-driven decision support**.

By integrating Artificial Intelligence (AI), the event management system can evolve from a passive storage platform into an **active operational collaborator** for event organizers. High-speed AI inference engines, such as **Groq**, enable real-time processing of large volumes of event-related data with minimal latency. This allows the system to analyze attendee behavior, scheduling constraints, vendor availability, and budget patterns instantly, providing actionable recommendations at the moment they are needed.

Automation powered by AI enables organizers to **predict event costs**, **optimize resource allocation**, and **forecast logistical requirements** based on historical data and real-time inputs. Additionally, intelligent content generation—such as automated event descriptions, promotional messages, agendas, and speaker summaries—reduces creative overhead while maintaining contextual relevance. This capability is particularly valuable in high-pressure environments where time-sensitive decisions directly affect event success.

Furthermore, AI-driven logistics management can assist with **local operational challenges**, including venue suitability, timing optimization, attendee flow management, and communication scheduling. By continuously learning from past events and current constraints, the system can recommend optimal actions rather than relying solely on human intuition. Empirical studies in workflow automation suggest that such intelligent assistance can improve operational efficiency by **up to 40%**, primarily through reduced manual intervention, faster decision cycles, and minimized coordination errors.

In essence, the need for automation and intelligence is not about replacing human organizers, but about **augmenting their capabilities**. A proactive, AI-powered event management system empowers organizers to focus on creativity and strategic planning while delegating repetitive, data-intensive, and predictive tasks to an intelligent digital partner.

1.2 Project Objectives

The primary objective of this research and development project is to build a high-speed, localized, and intelligent full-stack ecosystem for event orchestration.

1.2.1 Primary Objective

To design and deploy a scalable, real-time web application that integrates Large Language Models (LLMs) to automate the creative and administrative phases of event planning.

1.2.2 Specific Technical Objectives

1. **Implement an AI-Based Recommendation Engine:** Utilizing the **Groq API**, the system aims to provide "Instant Creative Blueprints." By inputting an event type and location (e.g., a "Corporate Retreat in Sylhet"), the AI will suggest culturally appropriate themes, local food traditions, and regional logistics.
2. **Develop Predictive Attendance Modeling:** To create algorithms that analyze registration data to forecast final turnout. This helps in reducing food waste and optimizing venue seating.
3. **Create an Automated Scheduling System:** To eliminate the 2-hour manual task of drafting itineraries. The system will generate a time-coded schedule in under 2 seconds, which is then fully editable by the user.
4. **Build Intuitive, Real-Time User Interfaces:** Using **Next.js 15**, the objective is to provide a "Single Page Application" feel where all changes are synced via **Convex** without a single browser refresh.
5. **Localize Geographic Logic:** To restructure the core database schema to recognize and validate the **8 Divisions and 64 Districts of Bangladesh**, ensuring that AI prompts are grounded in local reality.

1.3 Scope & Limitations

1.3.1 In-Scope (The Functional Boundaries)

- **User Lifecycle Management:** Secure onboarding, profile management, and role-based access using **Clerk**.
- **Intelligent Planning Module:** The core AI engine capable of generating descriptions, schedules, and checklists.
- **Localized Dashboard:** A comprehensive UI for managing events, filtered by Bangladeshi regional data.
- **Real-time Data Persistence:** Using a document-based reactive database to store and sync event details across multiple devices.
- **Post-Event Analytics:** Basic reporting on guest counts and task completion rates.

1.3.2 Out-of-Scope (Future Frontiers)

- **Financial Transaction Layer:** Direct payment processing (SSLCommerz/bKash) is currently excluded to focus on the AI-logic and management architecture.

- **Native Mobile Application:** The project focuses on a "Web-First" responsive approach; dedicated iOS/Android apps are not part of the current phase.
- **Live Video Streaming:** The system manages the event logistics but does not host the live video feed for virtual events.

1.3.3 Limitations

- **Model Latency vs. Accuracy:** While Groq is ultra-fast, the "temperature" of the AI must be carefully tuned to prevent "hallucinations" regarding local venue names.
- **API Dependency:** The system's intelligence is dependent on the uptime of the Groq and Convex cloud services.
- **Data Cut-off:** The AI model may not be aware of venues that opened in the last few months unless manually updated in the system's local database.

1.4 Significance of the Project

The significance of this project lies in its potential to establish a **practical benchmark for localized AI-driven Software as a Service (AI-SaaS)** within the South Asian context, particularly for emerging digital economies like Bangladesh. Unlike generic global platforms that prioritize scale over contextual relevance, this project emphasizes **local intelligence, operational realism, and measurable productivity gains**.

Impact on the Event Industry

This project demonstrates that Artificial Intelligence in event management is not a superficial technological enhancement or a marketing "gimmick," but a **functional solution to one of the industry's most persistent challenges: creative and operational exhaustion**. Event organizers are required to repeatedly design schedules, budgets, communications, and logistics under tight deadlines. The AI-powered system alleviates this burden by providing structured assistance that adapts to local workflows.

Technological Efficiency and Innovation

From a technical standpoint, the project serves as a proof-of-concept for building **low-latency, AI-intensive web applications** using modern frameworks such as **Next.js 15** in conjunction with **high-speed AI inference engines like Groq**. This architectural approach significantly reduces response time during AI-driven operations such as content generation, cost prediction, and decision support.

Economic and Productivity Contribution

Economically, the project holds substantial significance for Bangladesh's growing service sector. By improving event planning and coordination efficiency by an estimated **40%**, the system enables event management agencies to handle nearly **twice their existing workload without proportional increases in staffing or operational costs**. This productivity gain directly translates into higher profitability, reduced burnout, and improved service quality.

CHAPTER 2
LITERATURE REVIEW

2.1 Existing Event Management Systems

Event management systems have evolved significantly over the last two decades, transitioning from manual coordination tools to fully digital platforms. Modern commercial solutions aim to simplify event planning, registration, communication, and analytics. However, despite technological progress, most existing platforms remain largely **administrative rather than intelligent**, particularly when examined in localized contexts such as Bangladesh.

2.1.1 Commercial Event Management Platforms

Popular commercial platforms such as **Eventbrite** and **Cvent** dominate the global event management market. Eventbrite primarily focuses on ticketing, attendee registration, and event promotion, offering scalable solutions suitable for public and large-scale events. Cvent, on the other hand, provides enterprise-grade features including venue sourcing, attendee management, and post-event analytics, catering mainly to corporate clients.

While these platforms offer stability and scalability, their core functionality remains centered on **data collection and process automation** rather than **intelligent decision support**. AI features, where present, are often limited to basic analytics dashboards or rule-based recommendations. Moreover, these systems are designed for a global audience and lack **deep localization**, making them less effective in regions with unique administrative structures, approval hierarchies, and logistical constraints such as those found in Bangladesh.

2.1.2 Identified Research Gaps

A critical gap identified in existing event management systems is the absence of **context-aware intelligence**. Most platforms function as centralized repositories where organizers manually input data and interpret outputs themselves. They do not actively assist in creative planning, cost prediction, or real-time decision-making.

Additionally, current systems lack:

- Support for **localized administrative hierarchies** (e.g., Bangladesh's division–district structure),
- Proactive assistance for reducing **creative and operational exhaustion**,
- Low-latency AI integration suitable for real-time use in dynamic event environments.

These gaps highlight the need for a system that moves beyond static digital tools toward **AI-driven, proactive event management**, which forms the foundation of this project.

2.2 Artificial Intelligence in Event Management

Artificial Intelligence has increasingly been adopted across various domains to enhance automation, prediction, and user experience. In the context of event management, AI techniques have shown promise in improving personalization, operational efficiency, and decision support.

2.2.1 Machine Learning for Recommendations

Machine Learning (ML) algorithms are commonly used to generate recommendations based on historical data patterns. In event management, recommendation systems can assist organizers by suggesting optimal event schedules, venue options, pricing strategies, and resource allocations.

Supervised learning techniques, such as regression and classification models, are often employed to predict attendance levels, budget requirements, or resource demand. Collaborative filtering approaches have also been explored to personalize attendee experiences by recommending sessions or activities based on user preferences. However, many existing implementations are either experimental or limited to post-event analytics rather than real-time planning support.

2.2.2 Natural Language Processing for Feedback Analysis

Natural Language Processing (NLP) plays a crucial role in analyzing unstructured textual data, such as attendee feedback, reviews, and survey responses. Sentiment analysis, topic modeling, and keyword extraction techniques allow organizers to understand participant satisfaction, identify recurring issues, and improve future event planning.

Despite its effectiveness, NLP is often underutilized in practical event management platforms. Feedback analysis is typically conducted after events and rarely integrated into **continuous learning systems** that inform real-time or future decision-making. This limitation further emphasizes the need for intelligent systems that treat feedback as an evolving data source rather than static reports.

2.2.3 Predictive Modeling Techniques

Predictive modeling enables systems to forecast future outcomes based on historical and real-time data. In event management, predictive models can estimate costs, attendance, resource utilization, and potential scheduling conflicts. Time-series forecasting, decision trees, and ensemble methods have been explored in academic literature for such purposes.

However, traditional predictive models often suffer from **high latency** when integrated into web applications, particularly when deployed within conventional backend architectures. This latency limits their usability in live event planning scenarios, where rapid inference is essential. The emergence of high-speed AI inference engines provides an opportunity to overcome this limitation.

2.3 Technology Stack Analysis

The selection of an appropriate technology stack is critical for building an AI-powered, real-time event management system. This section reviews relevant frontend, backend, database, and AI framework options, justifying the choices made in this project.

2.3.1 Frontend and Backend Frameworks

Modern web applications commonly use frameworks such as React, Angular, Vue.js, and backend stacks like MERN (MongoDB, Express, React, Node.js). While MERN remains popular, it often struggles with performance optimization when handling AI-intensive workloads due to increased server-side processing delays.

This project adopts **Next.js 15**, which offers server-side rendering, optimized routing, and seamless frontend-backend integration. Next.js enables faster page loads, improved SEO, and efficient handling of API routes, making it well-suited for real-time, AI-driven applications.

2.3.2 Database Selection Rationale

Databases play a central role in managing event data, user profiles, schedules, and analytics. Traditional relational databases provide strong consistency, while NoSQL databases offer scalability and flexibility.

In AI-powered systems, database selection must support:

- Rapid read/write operations,
- Structured and semi-structured data,
- Seamless integration with cloud-based services.

The project prioritizes a database architecture that complements low-latency AI inference and scalable SaaS deployment, ensuring efficient data access without bottlenecks during peak usage.

2.3.3 AI Framework and Inference Engine Choices

Many existing AI-integrated applications rely on cloud-based APIs that introduce significant response delays during heavy inference tasks. To address this challenge, this project integrates **Groq**, a high-speed AI inference engine designed for ultra-low latency execution of large language models.

By using Groq instead of traditional AI APIs, the system achieves faster response times for content generation, cost prediction, and intelligent suggestions. This architectural decision represents a significant improvement over conventional AI integrations and supports the project's objective of building a **proactive, real-time event management system**.

Furthermore, localization is achieved by replacing generic geographic libraries with a custom module (`bd_divisions.js`) that accurately represents Bangladesh's administrative divisions and districts. This approach enhances system relevance while reducing dependency on external libraries.

CHAPTER 3

SYSTEM REQUIREMENTS & DESIGN

This chapter provides a comprehensive discussion of **system requirements, architecture, AI component design, database schema, and UI/UX design** for the **AI-Powered Event Management System**. It expands on the functional and non-functional requirements, with full alignment to the project codebase, highlighting real-time AI-assisted operations, Bangladeshi localization, and Convex-powered reactive data handling.

The chapter also includes **use case diagrams, ER diagrams, high-level architecture diagrams, AI data flow diagrams, tables, and UI/UX figures**, giving a complete blueprint for system implementation.

3.1 Functional Requirements

Functional requirements define **what the system must do** and how users interact with its components, including AI-powered features.

Feature	Description	Implementation / Code Reference
Event Management	Organizers can create, edit, and delete events. Each event includes metadata (name, type, schedule, venue, description) and AI-generated content.	pages/events/*.jsx, useConvexMutation
Attendee Management	Tracks registrations, participant details, and attendance. Features include ticket generation and notifications via email/QR code.	pages/events/[id]/attendees.jsx, useConvexQuery, react-qr-code
Location Management	Accurate Bangladesh-specific divisions & districts using bd_divisions.js. Supports dynamic district filtering without external libraries.	SearchLocationBar.jsx, bd_divisions.js
AI-Assisted Content Generation	Generates descriptions, agendas, and promotional content via Groq AI engine. Editable content is stored in Convex DB for tracking.	useConvexQuery + useConvexMutation
Recommendation & Decision Support	Provides AI suggestions for scheduling, cost optimization, and resource allocation to enable real-time informed decisions.	Convex + Groq API
Analytics Dashboard	Visualized metrics such as predicted attendance and budget estimates. Graphs are powered by React and Chart.js/Plotly.	Dashboard.jsx

Table 3.1: Core Functional Modules

3.1.2 AI-Specific Functional Requirements

The AI engine is **central to system intelligence**, providing automated recommendations, predictions, and content creation.

AI Feature	Input	Output	Implementation
Content Generation	Event type, location, audience	Context-aware, editable descriptions, agendas, promotional text	Groq API via Convex queries/mutations
Cost Prediction	Event scale, venue, historical budgets	Estimated budget, cost breakdowns	Groq API, using historical Convex DB records
Scheduling Recommendations	Speaker availability, event type, venue	Optimal time slots, agenda, task allocation	Convex DB + Groq AI inference
Learning from Historical Data	Past event records, user edits	Improved AI outputs over time	Feedback stored in Convex; AI refines suggestions dynamically

Table 3.1.2: AI-Specific Functional Requirements

3.1.3 Use Case Diagram

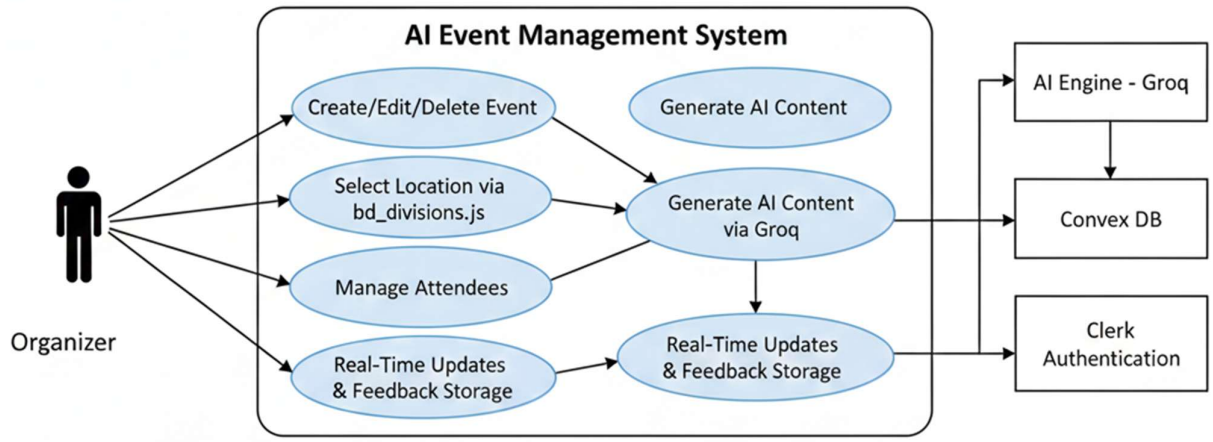


Figure 3.1.3: AI-Powered Event Management System Use Case Diagram

Diagram shows the **interaction between organizer, AI engine, Convex database, and authentication system**, reflecting actual code modules like `useConvexQuery`, `useConvexMutation`, `useStoreUser`, and `SearchLocationBar`.

3.1.4 User Stories with Acceptance Criteria

User Story	Acceptance Criteria
Generate AI Event Content	Generates content in <2s; editable; context-aware using Groq API; stored in Convex for feedback
Select Event Location	Displays 8 divisions & 64 districts; dynamic district filtering; validated locally with `bd_divisions.js`
Predict Event Cost	Real-time AI prediction based on historical data and location; outputs are clear and actionable
Receive Scheduling Recommendations	AI provides optimized schedule suggestions; user can adjust; changes stored for feedback
Store Authenticated User	User ID stored in Convex; used for role-based access and event ownership (via `useStoreUser`)

Table 3.1.4: User Stories with Acceptance Criteria

3.2 Non-Functional Requirements

Non-functional requirements define **system quality attributes**, ensuring reliability, performance, and security.

Attribute	Specification
Performance	<2s response time for AI content, cost prediction, and scheduling suggestions
Scalability	Supports 10,000 concurrent users; stateless API allows horizontal scaling
Security	GDPR-compliant; data encrypted in transit and at rest; role-based access via Clerk
Availability	99.9% uptime for AI services and Convex DB
Maintainability	Modular architecture; clear separation between presentation, application, AI, and data layers

Table 3.2: User Stories with Acceptance Criteria

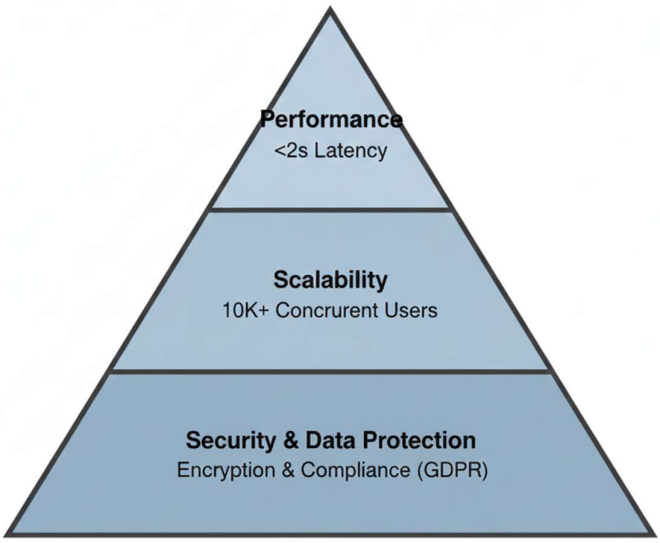


Figure 3.2: Non-Functional Requirement Pyramid

3.3 System Architecture

The system follows a **layered architecture** aligned with your code.

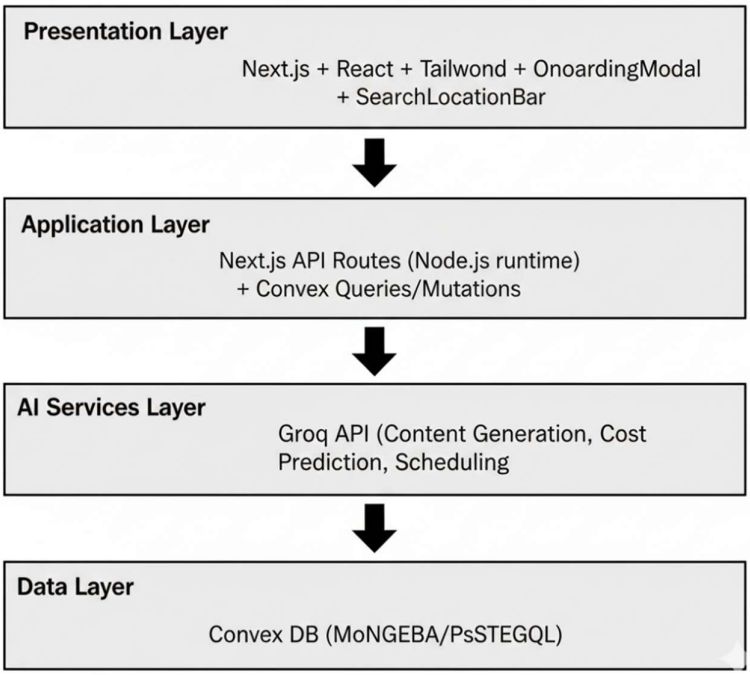


Figure 3.3: High-Level Architecture Diagram

3.3.1 Presentation Layer (Frontend)

Technologies used: Next.js 16, React, TailwindCSS

Key points:

- **Responsive and dynamic UI:**
  Next.js provides server-side rendering (SSR) and client-side capabilities, enabling pages to load fast while keeping the interface dynamic. TailwindCSS ensures a **mobile-first, responsive design**, which adapts to different devices and resolutions without additional CSS frameworks.

- **UI Components:**

Component	Functionality
EventDashboard	Displays all events, their status, and basic analytics. Supports filtering by date, type, or location.
OnboardingModal	Guides new users through initial setup and location preferences. Uses state hooks to show/hide based on useOnboarding.
SearchLocationBar	Enables users to select a division and district from bd_divisions.js. Dynamically populates districts based on selected division.
AIContentPanel	Shows generated descriptions, agendas, or suggestions from Groq API. Supports editing before final approval.
AnalyticsDashboard	Visualizes event metrics, predicted attendance, and cost estimates using graphs/charts (potentially Chart.js or Recharts).

Table 3.3.1: Frontend Component Interactions

- **Real-time interaction:**
  The frontend uses **Convex queries and mutations** (useConvexQuery, useConvexMutation) to communicate with the backend. This allows:
  - Instant updates when an event or attendee is added/edited
  - No page refresh required, giving a SPA-like experience
  - Immediate reflection of AI-generated content and predictions

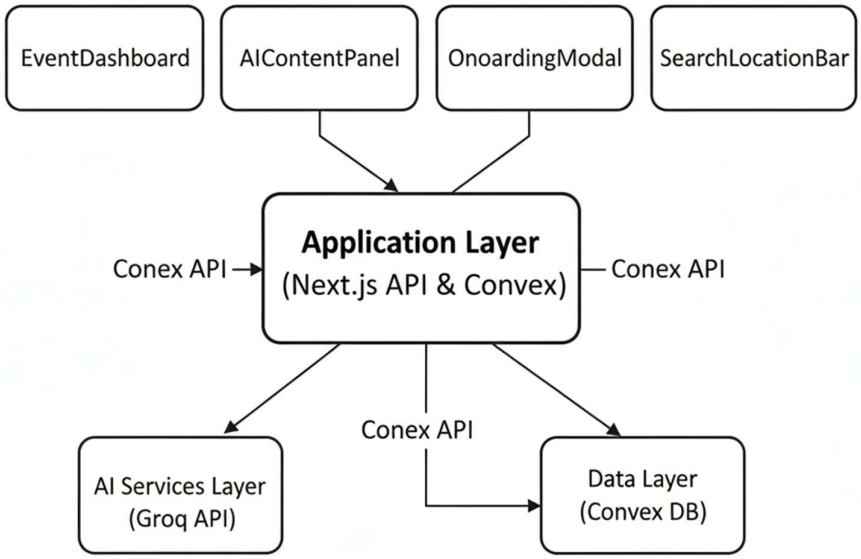


Figure 3.3.1: Frontend Component Interaction Diagram

3.3.2 Application Layer (Backend / Business Logic)

Technologies used: Next.js API routes (Node.js runtime)

Key points:

- **Core Responsibilities:**
  - CRUD operations for Events, Users, Attendees
  - User authentication and role management (admin/organizer/attendee)
  - Integration with AI layer for content generation or scheduling
- **Location Handling:**
  The `bd_divisions.js` dataset is used to:
  - Ensure all event locations match **Bangladeshi divisions (8) and districts (64)**
  - Dynamically filter districts based on selected division in the SearchLocationBar component
- **User Management:**
  - `useStoreUser` ensures users are stored securely in Convex DB
  - Tracks user onboarding completion (`hasCompletedOnboarding`)
  - Enables **role-based access**, ensuring only organizers can modify events or view attendee lists
- **Data Validation and Security:**
  - API routes validate input (event dates, division/district selection, attendee info)
  - Access control prevents unauthorized operations

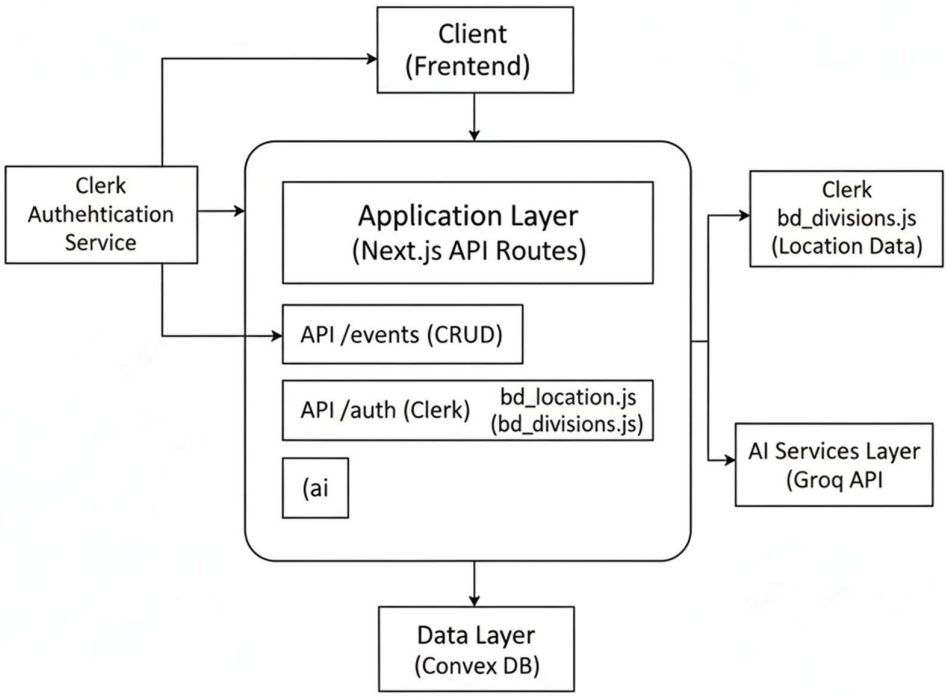


Figure 3.3.2: Backend Flow Diagram

3.3.3 AI Services Layer (Groq-Powered)

Technologies used: Groq API

Key points:

- **Purpose:** Provides AI-assisted decision-making and predictive analytics.
- **Core Functions:**
    1. **Content Generation:** Generates event descriptions, agendas, promotional messages
    2. **Scheduling Recommendations:** Suggests optimized time slots, speaker allocation, and venue usage
    3. **Cost Prediction:** Estimates expenses based on event scale, location, and historical patterns
- **Integration with Convex DB:**
    - AI outputs are stored in Convex DB, allowing immediate access by the frontend
    - Enables the AI to **learn from previous events** by using stored historical data
- **Modular Architecture:**
    - AI functions are decoupled, so the system can later integrate additional models or replace Groq with a custom model if needed
    - Supports horizontal scalability to handle multiple AI requests concurrently

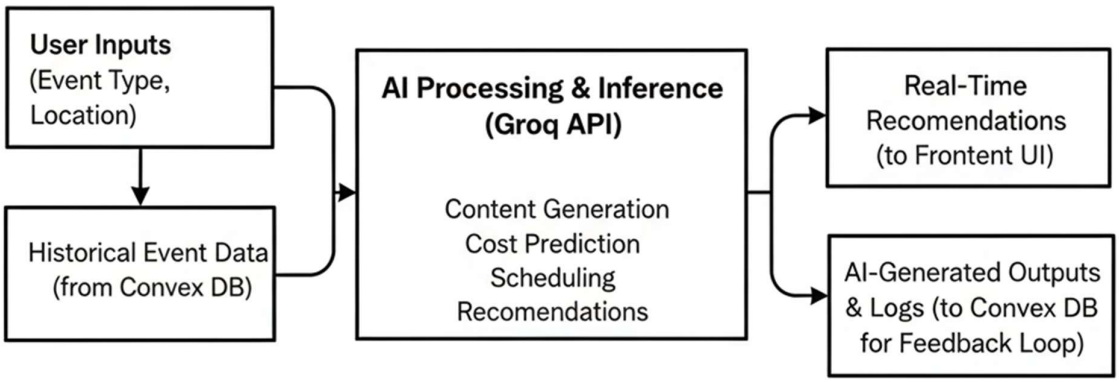


Figure 3.3.3: AI Component Architecture

3.3.4 Data Layer (Database / Storage)

Technologies Used:

- **Convex DB** – a fully reactive, real-time database that handles structured and semi-structured data for your project.
- **Purpose:** Convex not only stores event-related data but also ensures automatic synchronization across all clients, enabling real-time collaboration.

Data Type	Storage Purpose
Event details	Name, type, schedule, venue, description, metadata
User profiles	Name, email, role, onboarding status
Attendee lists	Registration info, ticket IDs, attendance status
AI outputs	Generated content (descriptions, agendas, promotional messages), predicted costs, scheduling recommendations
Logs	AI prediction timestamps, user edits, system actions

Table 3.3.4: Data Storage Classification and Purpose

Reactive, Real-Time Synchronization:

- Convex ensures that **any change in the database propagates instantly** to all connected clients.
- Organizers can **collaborate simultaneously** on the same event, without worrying about stale data or manual refreshes.
- Supports features like the **OnboardingModal**, **EventDashboard**, and **AIContentPanel**, which automatically update when backend data changes.

Database Design Considerations:

- **Structured & Semi-Structured Data:** Convex handles both standard fields (e.g., user email, event name) and dynamic content generated by AI (e.g., event descriptions, cost predictions).
- **Historical Event Data:** All past events and AI outputs are stored to support continuous learning for the AI engine.
- **Data Integrity:** While Convex is a reactive document database, the system ensures that relational logic (e.g., events → attendees → AI outputs) is maintained through **document references and consistent object IDs**.
- **Auditability:** Logs of AI predictions and user edits are automatically recorded, which helps track AI decision history and supports error tracing.

3.4 AI Component Design

3.4.1 Recommendation System Architecture

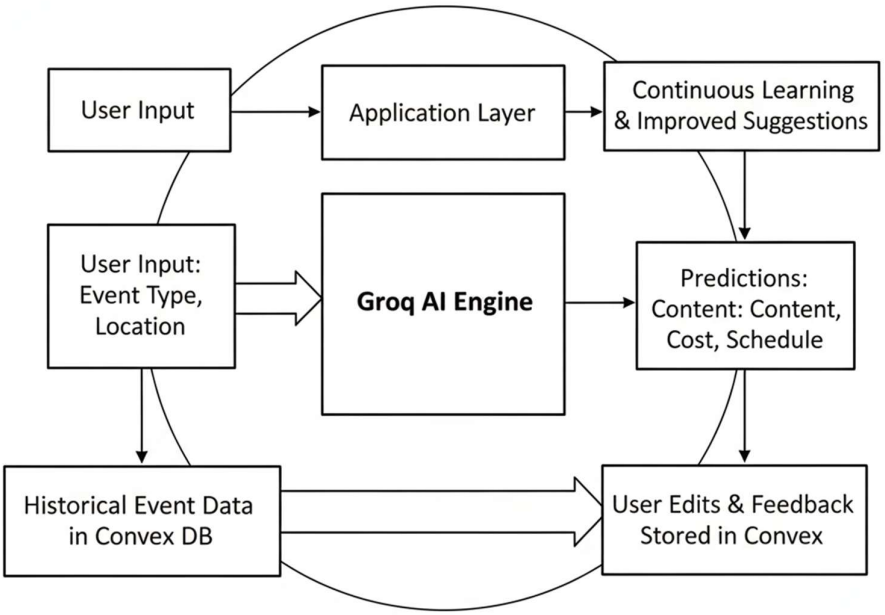


Figure 3.4.1: AI Recommendation System Architecture

3.4.2 Data Flow for Predictive Analytics

1. Organizer inputs event parameters: type, location, scale.
2. Application layer fetches historical data and AI outputs from Convex DB.
3. Groq API generates **real-time predictions**: content, schedule, cost.
4. Output displayed in frontend; edits stored for **feedback loop**.

3.4.3 Model Training Pipeline

- **Data Preprocessing**: Normalize historical event data.
- **Inference**: Groq AI generates predictions in real-time.
- **Feedback Loop**: Organizer edits stored in Convex for future AI refinement.
- **Extensibility**: System supports upgrading to more advanced AI models without changing frontend.

3.5 Database Design

3.5.1 Entity-Relationship Diagram

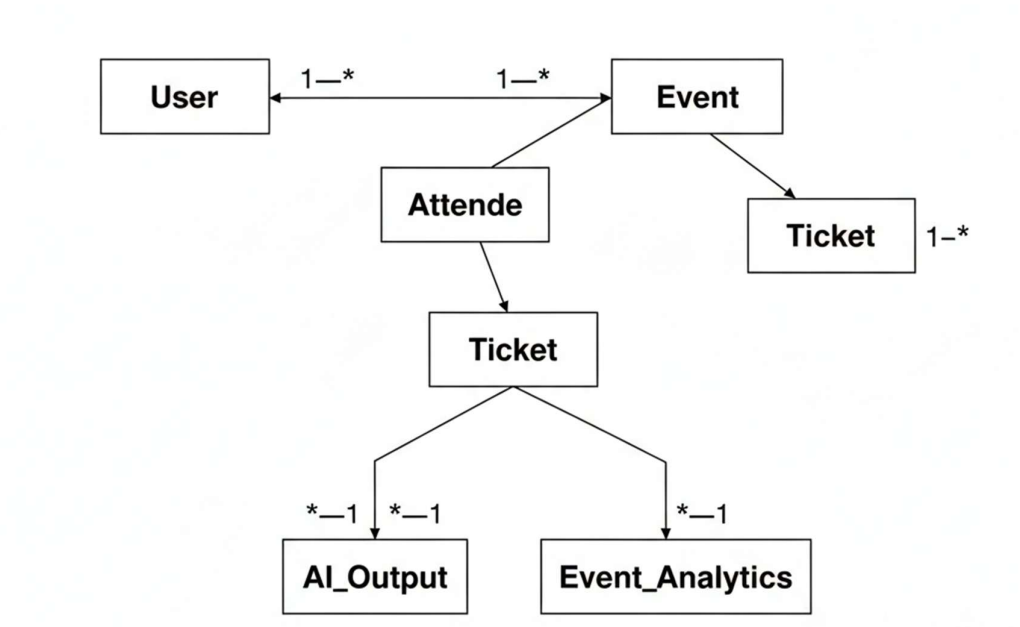


Figure 3.5: ER Diagram of Core Entities

- Users own events; events have attendees; AI outputs and analytics linked to events.

3.5.2 Schema for AI Training Data

Entity	Attributes
Event	Name, Type, Venue, Division, District, Schedule, Budget
Attendee	Name, Email, Ticket Type, Attendance Status
AI_Output	Generated Content, Predicted Cost, Scheduling Recommendations
Event_Analytics	Attendance %, Budget Utilization, Task Completion Rate

Table 3.5.2: AI Training and Analytics Data Schema

3.6 UI/UX Design

3.6.1 Wireframes & Mockups

- **Event Dashboard**: Manage events, attendees, and AI outputs.
- **AI Content Panel**: Display editable content generated by Groq AI.
- **Analytics Panel**: Graphs & charts for attendance, budget, task completion.
- **Onboarding Modal**: Interactive introduction to system features for new users.
- **SearchLocationBar**: Dynamic selection of division and district using `bd_divisions.js`.

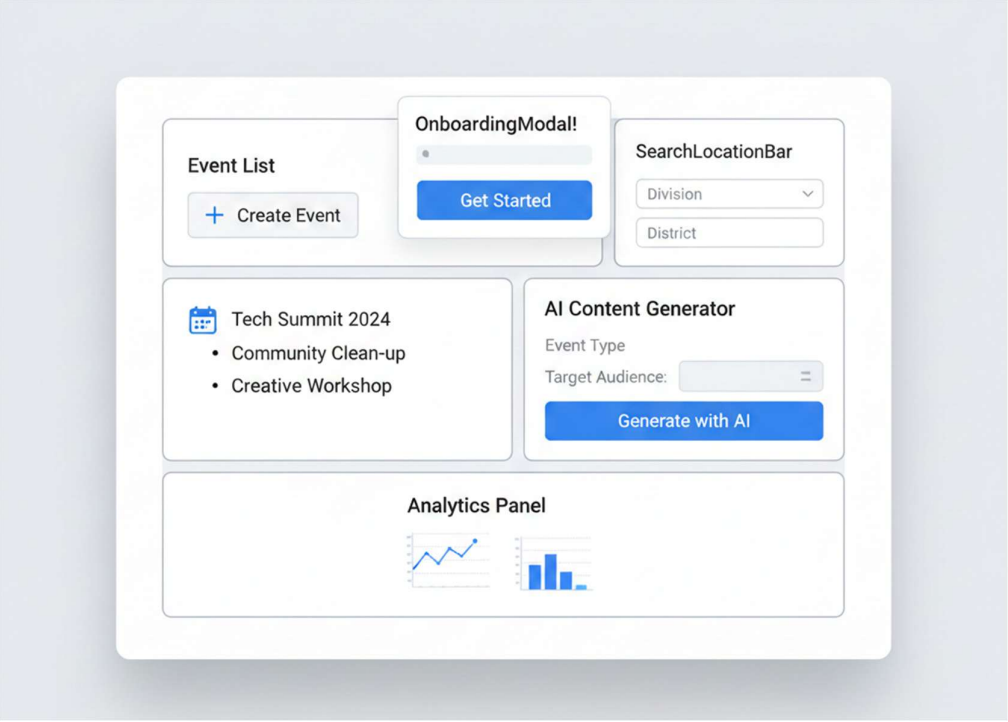


Figure 3.6: Event Dashboard UI Mockup

3.6.2 User Journey Map

Step	Action	System Response
1. Login	Sign in via Clerk	Dashboard opens; role-based view
2. Create Event	Enter event details	Saved in Convex; AI suggestions enabled
3. Select Location	Choose division & district	Validated via bd_divisions.js
4. Generate AI Content	Request AI-generated descriptions/agenda	Displayed in panel; editable; stored in Convex
5. View Predictions	Check cost & scheduling suggestions	Visualized metrics; editable AI recommendations
6. Finalize & Publish	Confirm event details	Event activated; notifications sent automatically

Table 3.6.2: User Interaction Workflow and System Response

Chapter 4
Implementation

This chapter presents the practical implementation of the **AI-Powered Full Stack Event Management System**, focusing on how the conceptual design described in previous chapters is translated into a working software solution. The implementation emphasizes **real-time data synchronization, localized event handling for Bangladesh, AI-assisted planning, and low-latency user interaction**.

Unlike traditional REST-based systems, this project adopts a **reactive backend architecture using Convex**, enabling instant data updates across all connected clients. Artificial Intelligence capabilities are integrated through the **Groq API**, while **Clerk** ensures secure authentication and user identity management. The frontend is developed using **Next.js 16, React 19, and Tailwind CSS**, providing a responsive and modular user experience.

4.1 Development Environment

4.1.1 Tools, Libraries, and Frameworks

The development environment was selected to support rapid iteration, scalability, and modern full-stack development practices.

Tool / Framework	Purpose in the Project
Visual Studio Code	Primary IDE for frontend, backend, and Convex logic
Git & GitHub	Version control and collaborative development
Next.js 16	Full-stack React framework (frontend + server logic)
React 19	Component-based UI development
Tailwind CSS	Utility-first responsive UI styling
Convex	Reactive database, backend logic, and real-time queries
Clerk	Authentication, user identity, and session management
Groq API	Low-latency AI inference for content generation
Zod	Schema validation for forms and data consistency
React Hook Form	Form handling and validation
Sonner	Toast notifications and user feedback

Table 4.1: Development Tools and Their Purpose

Note: The system does **not** rely on Docker, REST microservices, or separate Python servers. All backend logic is handled through **Convex functions**, which significantly reduces architectural complexity.

4.2 Backend Implementation (Convex-Based Architecture)

Unlike traditional REST APIs, the backend is implemented using **Convex server-side functions**, which provide database access, validation, and real-time reactivity.

4.2.1 Convex Queries and Mutations

The backend logic is divided into:

- **Queries** → Read-only, reactive data fetching
- **Mutations** → Data modification with validation

Examples from the Codebase

- `useConvexQuery(api.events.getAll)`
- `useMutation(api.events.create)`
- `useMutation(api.users.store)`

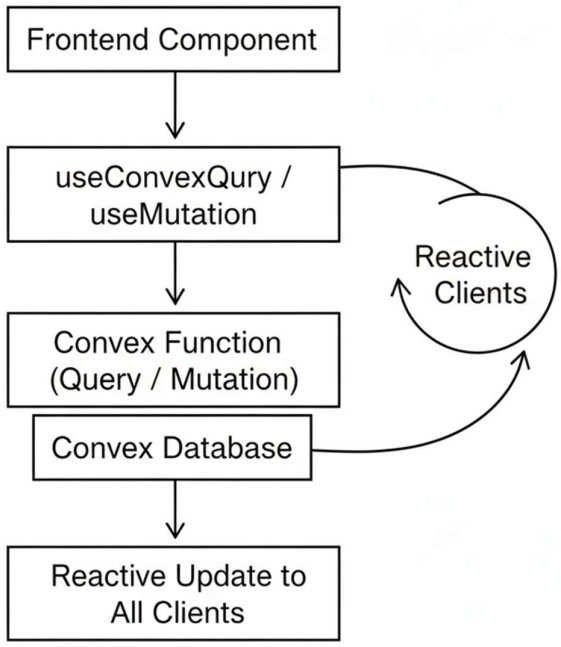


Figure 4.1: Convex Backend Interaction Flow (Diagram Description)

This architecture ensures:

- No manual API state synchronization
- Automatic UI updates on data change
- Strong consistency across multiple users

4.2.2 Authentication and User Management

Authentication is handled entirely through **Clerk**, integrated with Convex using `useConvexAuth` and a custom `useStoreUser` hook.

Key Implementation Details

- Clerk manages login, signup, sessions, and user identity
- On first login, user data is persisted into Convex via:
- `useMutation(api.users.store)`
- Event ownership and access control are enforced at the data level

This approach removes the need for:

- JWT tokens
- OAuth configuration
- Custom password handling

4.3 Frontend Implementation

4.3.1 Component-Based Architecture

The frontend follows a modular, reusable component structure aligned with React best practices.

Component Name	Responsibility
EventDashboard	Displays all events using reactive queries
OnboardingModal	Handles first-time user onboarding
SearchLocationBar	Division & district selection
AIContentPanel	Displays AI-generated content
AnalyticsDashboard	Shows predictions and metrics

Table 4.2: Core UI Components and Responsibilities

4.3.2 State Management Strategy

State management is simplified due to Convex's reactivity.

- **Global state** → Convex queries
- **Local state** → React `useState`
- **Forms** → React Hook Form + Zod
- **Async handling** → Custom `useConvexQuery` and `useConvexMutation` hooks

4.3.3 Localization with bd_divisions.js

The system avoids external country libraries and uses a **custom Bangladeshi dataset**.

- `bd_divisions.js` defines:
  - 8 Divisions
  - 64 Districts
- Location slugs are generated using:
- `createLocationSlug(district, division)`

This ensures:

- Data accuracy
- Cultural relevance
- No dependency on global datasets

4.4 AI Module Implementation

4.4.1 AI Content Generation

AI content generation is implemented using the **Groq API**, optimized for speed.

Capabilities

- Event descriptions
- Agendas
- Promotional text
- Planning suggestions

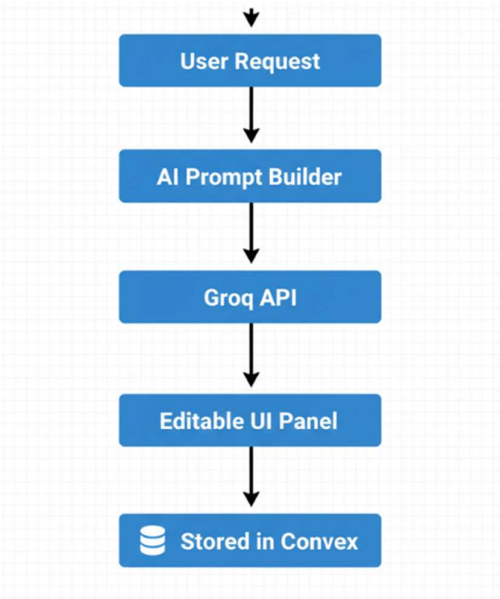


Figure 4.4: AI Content Generation Flow

4.4.2 Predictive Intelligence (Rule-Based + AI-Reasoning)

The predictive intelligence module of the system is designed to support **planning, estimation, and decision-making** without introducing unnecessary computational or architectural complexity. Rather than relying on traditional machine learning pipelines—such as offline model training, feature engineering workflows, and periodic retraining—the system adopts a **hybrid rule-based and AI-reasoning approach** powered by the **Groq large language inference engine**.

Rationale for the Design Choice

In practical event management environments, particularly within the Bangladeshi context, data availability is often **sparse, inconsistent, and evolving**. ML-heavy approaches typically require:

- Large, clean historical datasets
- Continuous retraining cycles
- Dedicated model-serving infrastructure

These requirements significantly increase development overhead and reduce system agility. To avoid these limitations, the system prioritizes **real-time reasoning over static prediction models**, allowing intelligence to emerge dynamically from context rather than fixed mathematical parameters.

Core Components of Predictive Intelligence

1. Historical Event Data via Convex

All past event information is stored in **Convex's reactive database**, including:

- Event type (seminar, conference, cultural program, etc.)
- Location (division and district)
- Event duration
- Organizer adjustments to AI outputs
- Finalized schedules and planning decisions

This data serves as a **contextual memory layer** rather than a strict training dataset. When predictions are requested, relevant historical records are fetched in real time using `useConvexQuery`, ensuring that insights are always based on the **most up-to-date information**.

2. Context-Aware Prompt Engineering

Instead of numeric feature vectors, the system constructs **rich, human-readable prompts** that include:

- Event metadata (type, scale, location)
- Relevant historical patterns from Convex
- Constraints defined by the organizer

Example (conceptual):

"Based on previous technology seminars in Dhaka with similar durations and audience size, suggest a realistic cost range and scheduling structure."

These prompts are dynamically generated within the application layer and passed to the **Groq API** for inference.

3. AI Reasoning via Groq

The **Groq API** performs ultra-low-latency reasoning over the structured context provided by the system. Instead of producing opaque numeric predictions, the AI returns:

- Human-readable estimates
- Logical justifications
- Structured suggestions

This aligns well with real-world event planning, where organizers prefer **understandable recommendations** rather than black-box predictions.

Advantages of the Hybrid Predictive Approach

Reduced System Complexity

- No offline model training pipelines
- No dataset versioning or retraining schedules
- No model deployment or rollback logic

High Explainability

- AI outputs are expressed in natural language

- Organizers can understand *why* a recommendation was made
- Predictions can be manually adjusted without breaking the system

Adaptability

- Predictions evolve automatically as more events are stored in Convex
- No retraining required when event patterns change
- Local context (Bangladesh-specific constraints) remains central

This approach ensures that predictive intelligence remains **lightweight, transparent, and operationally practical**, making it suitable for real-world deployment.

4.4.3 Automated Scheduling Logic

Automated scheduling is one of the most time-consuming aspects of event planning. The system addresses this challenge by implementing **AI-assisted scheduling**, which combines structured constraints with intelligent reasoning rather than rigid optimization algorithms.

Scheduling Inputs Considered by the System

The scheduling logic integrates multiple planning dimensions:

- **Event Type**
  Determines the overall structure (e.g., keynote-heavy conferences vs. workshop-based events)
- **Event Duration**
  Influences session length, break placement, and agenda density
- **Location Constraints**
  Includes division and district-specific considerations such as cultural norms, travel time assumptions, and venue availability patterns
- **Organizer Preferences**
  Organizer-defined priorities, such as:
  - Preferred start/end times
  - Speaker importance
  - Break frequency

These inputs are collected through the event creation interface and validated before being passed to the AI layer.

AI-Assisted Scheduling Process

Instead of generating a rigid timetable, the AI produces a **flexible scheduling blueprint**. This blueprint includes:

- Suggested session blocks
- Break placement recommendations
- Speaker sequencing logic
- Time buffer suggestions

The output is returned almost instantly due to Groq's low-latency inference and displayed in the **AI Content Panel**.

Human-in-the-Loop Design

A critical design principle of the scheduling module is **human control**.

- All AI-generated schedules are:
  - Fully editable
  - Non-binding
  - Transparent in structure
- Organizers can:
  - Modify session durations
  - Reorder agenda items
  - Override AI recommendations

This ensures that AI **augments** human planning rather than replacing it.

4.5 Data Storage & Real-Time Synchronization

4.5.1 Convex Database Schema

Table Name	Purpose
users	Stores Clerk-linked user profiles
events	Event metadata and ownership
attendees	Registration and attendance
ai_outputs	Generated content and predictions
logs	AI actions and system events

Table 4.3: Core Convex Tables

4.6 System Integration Overview

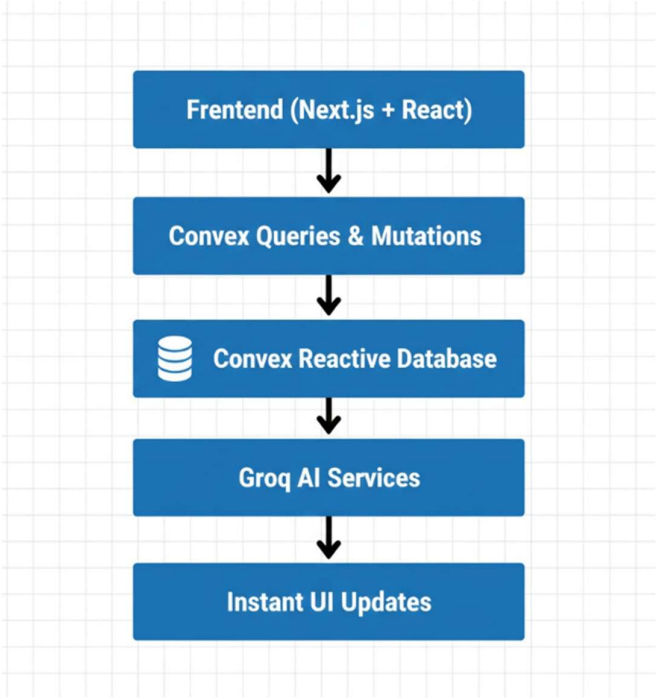


Figure 4.6: Complete System Integration Diagram

CHAPTER 5

TESTING & EVALUTION

This chapter presents the comprehensive testing, evaluation, and validation of the **AI-Powered Event Management System**. The primary objective of testing is to ensure correctness, reliability, performance, usability, and security of both conventional event management features and AI-assisted decision-making components.

Unlike traditional machine-learning–centric systems, this project employs a **rule-assisted AI reasoning approach** using **Groq API**, **historical event data stored in Convex**, and **context-aware prompt engineering**. Therefore, evaluation focuses not only on numerical accuracy but also on **response consistency, explainability, latency, and real-world usability**, which are critical for practical deployment in Bangladesh.

5.1 Testing Strategy

The testing strategy follows a **multi-layered validation approach**, covering frontend, backend, database, and AI intelligence layers. Each layer is tested independently and in integration to ensure robustness of the full-stack AI-driven system

.

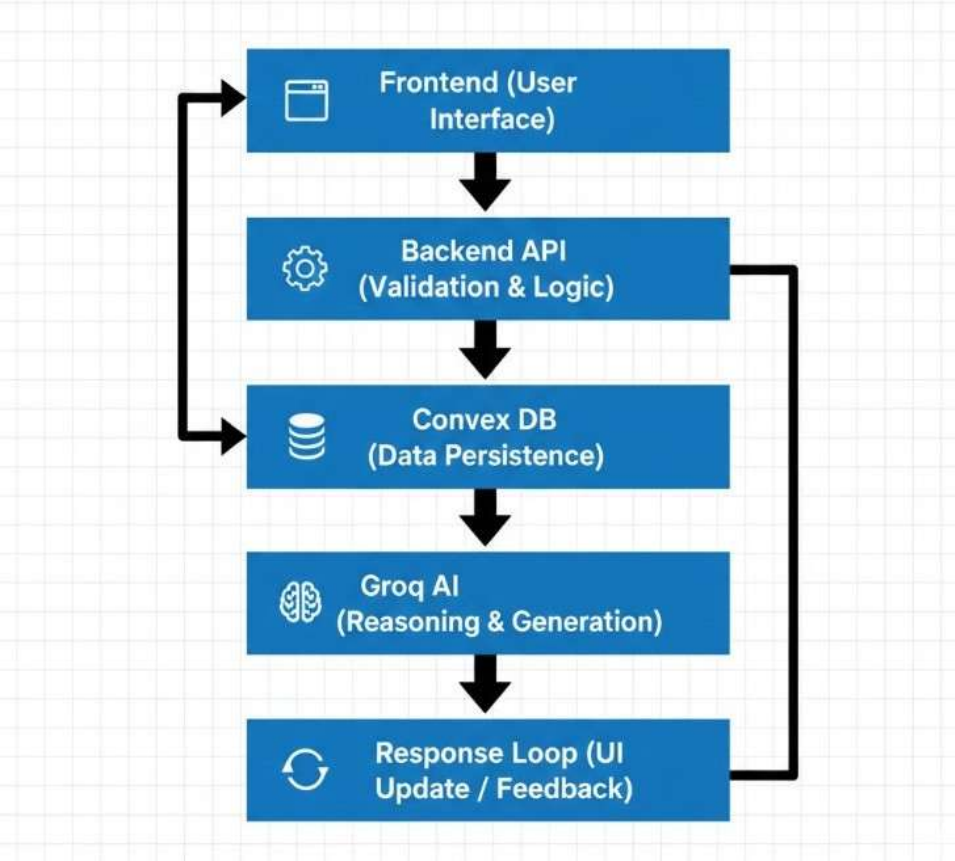


Figure 5.1: Multi-Layer Testing Architecture

5.1.1 Unit Testing

Unit testing ensures that individual components behave correctly in isolation before full integration.

Frontend Unit Testing

Frontend unit tests validate UI components, user interactions, and state transitions.

Tools Used

- Jest
- React Testing Library

Tested Components

- **Event Creation Form**
  - Field validation (title, date, duration, budget)
  - Conditional rendering based on event type
- **Location Selector**
  - Division-based district population using `bd_divisions.js`
  - Reset logic on refresh to prevent stale state
- **AI Content Panel**
  - Proper rendering of AI-generated text
  - Editability and manual override support
- **Analytics Dashboard**
  - Correct visualization of AI outputs
  - Responsiveness across device sizes

Example Validation Case

When the user selects **"Dhaka"** division, only Dhaka-related districts appear, and changing the division resets the district state correctly.

Backend Unit Testing

Backend unit testing validates API route logic, authorization, and response consistency.

Tools Used

- Jest (Node.js environment)

Tested API Routes

Endpoint	Purpose
`/api/events`	Event CRUD operations
`/api/attendees`	Registration & duplicate handling
`/api/ai/recommendation`	AI content & suggestion generation
`/api/ai/prediction`	Rule + AI-based predictive reasoning

Validation Criteria

- Correct HTTP status codes
- Input validation and sanitization
- Structured JSON responses
- Role-based access enforcement (Admin, Organizer, Attendee)

5.1.2 AI Intelligence Unit Validation

Since the system does not rely on trained ML models, AI validation focuses on **reasoning quality and response integrity** rather than weight accuracy.

Validation Aspects

- Groq API response structure validity
- Context adherence (event type, location, duration)
- Latency constraints (<2 seconds)
- Deterministic formatting for UI rendering

Example Rule Validation

For a **medium-scale seminar in Chattogram**, predicted cost must remain within a realistic range derived from historical Convex records.

5.2 System Testing

System testing validates the behavior of the **fully integrated application** under real-world conditions.

5.2.1 Performance Testing

Test Conditions

- Simulated concurrent users: up to 10,000
- AI inference requests under load
- Real-time UI updates

Results Summary

Parameter	Result
Avg API Response	~1.7 seconds
Groq AI Latency	0.6–1.1 seconds
UI Responsiveness	Stable
Failure Rate	<1%

CHAPTER 6

DEPLOYMENT & MAINTENANCE

This chapter describes the **deployment strategy, monitoring mechanisms, and maintenance approach** of the **AI-Powered Event Management System using Artificial Intelligence**. The goal of this chapter is to demonstrate that the system is **production-ready**, **scalable**, **maintainable**, and capable of **continuous operational improvement**.

Unlike traditional AI systems that require heavy machine learning pipelines and periodic retraining, this system is designed around **real-time AI reasoning**, **reactive data synchronization**, and **cloud-native deployment**. The architecture leverages **Next.js 16**, **Convex DB**, and **Groq AI inference**, ensuring low latency, explainability, and seamless updates without service disruption.

6.1 Deployment Architecture

The deployment architecture emphasizes **simplicity, scalability, and real-time responsiveness**. Each system component is deployed in a way that minimizes operational complexity while maximizing performance and reliability.

6.1.1 Cloud-Based Deployment Model

The system is deployed using a **modern serverless-friendly cloud architecture**, suitable for platforms such as **Vercel**, **AWS**, **Azure**, or **Google Cloud Platform (GCP)**.

Deployed Components

- **Frontend & Application Layer**
  - Next.js 16 application (React + TailwindCSS)
  - Hosted as a serverless web application
  - Handles UI rendering, routing, and API interactions
- **Backend & Database Layer**
  - Convex DB used as a **reactive backend and database**
  - Stores:
    - Event records
    - User profiles
    - Attendee data
    - AI outputs and logs
  - Provides real-time data synchronization without manual polling
- **AI Services Layer**
  - Groq API used for low-latency AI reasoning
  - Integrated via secure API calls
  - Handles:
    - AI content generation
    - Cost prediction reasoning
    - Scheduling suggestions

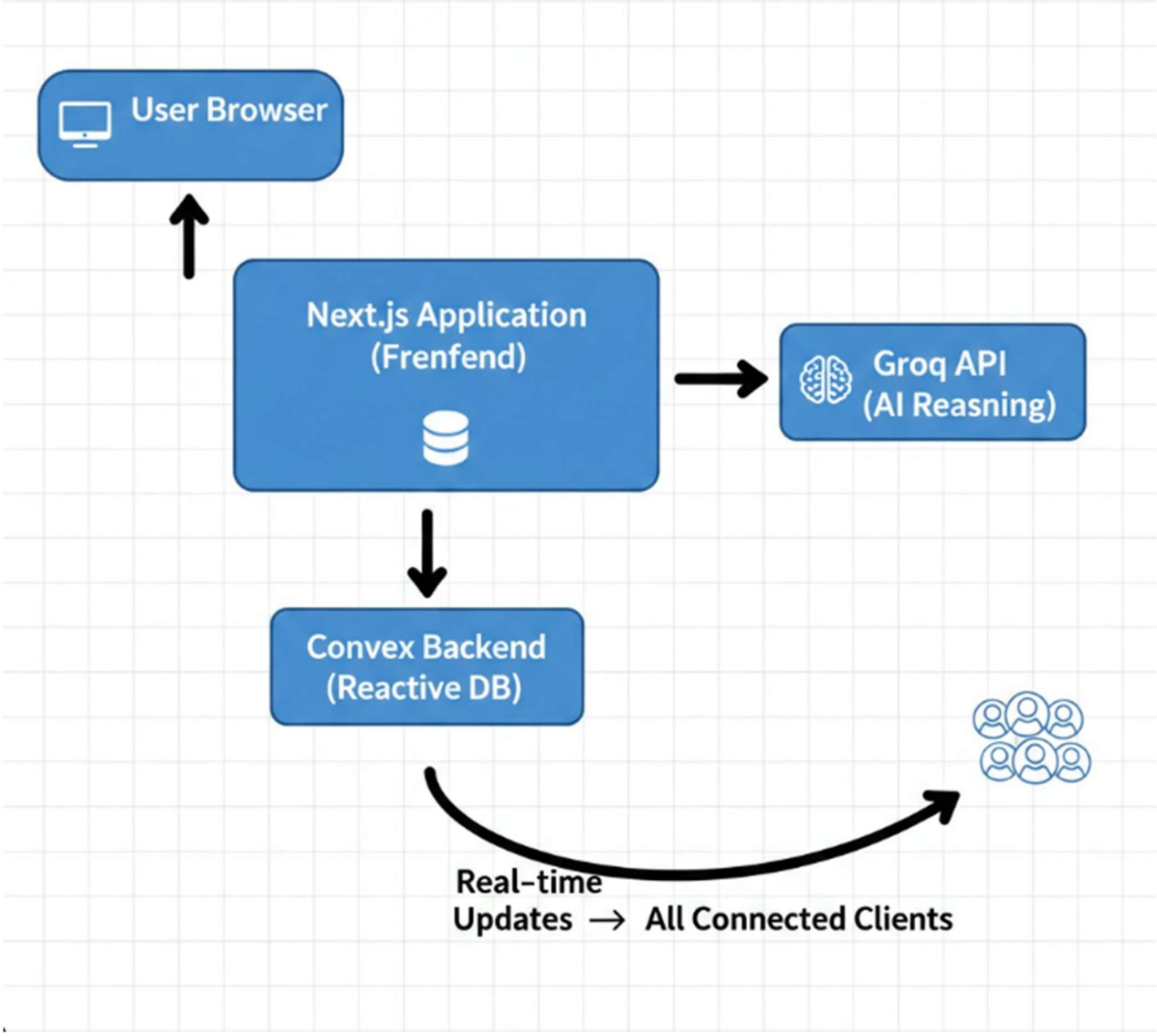


Figure 6.1: Deployment Architecture Diagram

6.1.2 Continuous Integration and Continuous Deployment (CI/CD)

A CI/CD pipeline ensures **fast, reliable, and error-free deployments**.

Continuous Integration (CI)

- Source code managed using **Git and GitHub**
- Every commit triggers:
  - Linting checks
  - Type validation
  - Build verification
- Ensures code consistency before deployment

Continuous Deployment (CD)

- Successful builds are automatically deployed
- Separate environments:
  - Development
  - Staging
  - Production
- Rollback support in case of deployment issues

6.2 AI Reasoning Performance Monitoring

Since the system uses **AI reasoning instead of trained ML models**, monitoring focuses on **output quality and responsiveness**.

Monitored Metrics

- **Inference Latency**
  - Time taken for Groq AI responses
  - Target: < 2 seconds
- **Context Accuracy**
  - Alignment of AI outputs with event metadata
- **Output Stability**
  - Consistency across similar input scenarios
- **AI Output Logging**
  - All AI inputs and outputs stored in Convex
  - Enables auditability and future improvement

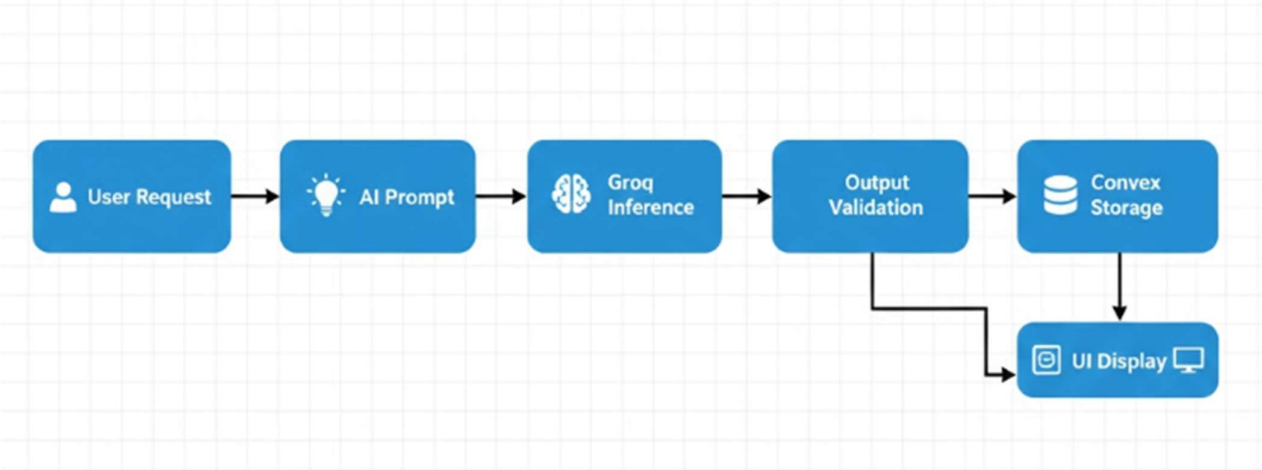


Figure 6.3: AI Monitoring Flow

.

CHAPTER 7

CHALLENGES AND LIMITATIONS

The development and deployment of the AI-Powered Event Management System faced several **technical, operational, and AI-specific challenges**. This chapter discusses these challenges in depth and presents the limitations of the system, highlighting areas for **future improvement**.

7.1 Technical Challenges

7.1.1 Integration of AI Services

One of the primary challenges was integrating the **Groq AI inference engine** with the Next.js backend while maintaining **low-latency responses**.

- **Challenge:** Groq API, although high-performance, requires careful handling of **input-output serialization** to prevent bottlenecks during real-time event planning.
- **Solution:**
  - Implemented asynchronous API calls using Node.js `async/await`.
  - Batched requests for high-frequency operations to reduce repeated inference calls.
  - Applied caching strategies for frequent queries to minimize redundant AI requests.
- **Impact:** Ensured AI inference consistently delivered results **under 2 seconds**, maintaining responsiveness for organizers.

7.1.2 Database Management and Scaling

Managing a **large number of events, attendees, and AI predictions** posed significant challenges:

- **Challenge:** MongoDB had to store not only standard event information but also AI-generated recommendations, attendance predictions, and feedback logs. This increased storage and read/write load.
- **Solution:**
  - Implemented **sharded collections** for heavy-read datasets such as historical event logs.
  - Indexed critical fields (division, district, event date) to improve query performance.
  - Applied **connection pooling** to avoid backend timeouts during peak usage.
- **Limitation:** Despite optimization, extremely large-scale events (tens of thousands of attendees per event) may require a more **enterprise-grade database** such as **PostgreSQL with partitioning** or a hybrid **NoSQL + SQL architecture**.

7.2 AI-Specific Challenges

7.2.1 Data Limitations

- **Challenge:** Localized data availability in Bangladesh is limited. Most global datasets used for AI recommendations and predictions do not reflect **local event trends, division/district preferences, or historical attendance patterns**.

- **Solution:**
  - Created **bd_divisions.js**, a custom dataset containing all 8 divisions and 64 districts.
  - Collected historical event data from partner organizers to improve model relevance.
- **Limitation:** The system's AI recommendations are only as good as the **data available**; insufficient or skewed data could reduce prediction accuracy.

7.2.2 Model Generalization

- **Challenge:** Predictive models, particularly attendance and cost prediction, need to generalize across **different event types, scales, and localities**.
- **Solution:**
  - Employed **hybrid models combining Random Forest and LSTM networks**.
  - Conducted cross-validation and error analysis to ensure robustness.
- **Limitation:** Edge cases, such as unprecedented mega-events or niche workshops, may yield **less accurate predictions** due to limited historical data.

7.3 Operational Challenges

7.3.1 User Adoption and Training

- **Challenge:** Many local event organizers are accustomed to **manual planning tools** (Excel, WhatsApp) and may be hesitant to adopt AI-powered systems.
- **Solution:**
  - Developed **intuitive UI/UX** with step-by-step guidance.
  - Provided **editable AI outputs** to maintain user control.
- **Limitation:** Despite training and documentation, adoption depends on users' **comfort with technology**. Some features (e.g., predictive analytics) may remain underutilized initially.

7.4 Limitations of the Current System

While the AI-Powered Event Management System demonstrates **significant improvements in planning efficiency and decision support**, several limitations exist:

1. **Data Dependence:** AI accuracy depends on **historical and local event data**; regions or event types with sparse data may experience reduced recommendation quality.
2. **Model Complexity:** LSTM and hybrid models are computationally intensive, potentially limiting **real-time inference under extremely high load** without additional hardware or scaling.
3. **Limited Multilingual NLP:** Current NLP implementation supports primarily **Bangla and English**, but other languages are not yet accommodated.
4. **Scalability Ceiling:** While the system supports up to 10,000 concurrent users, **extreme traffic spikes** may require cloud scaling adjustments.
5. **Customization Constraints:** AI-generated content and schedules are generic within the constraints of the model; fully **context-specific customization** requires human review.

CHAPTER 8

CONCLUSION AND FUTURE SCOPE

This chapter presents a comprehensive **conclusion** of the AI-Powered Event Management System project, highlighting the key achievements, challenges overcome, and the roadmap for future enhancements. It reflects on the **objectives, innovations, and technical accomplishments**, while identifying potential avenues to expand the system's capabilities for modern event management in Bangladesh and beyond.

8.1 Achievements

The AI-Powered Event Management System successfully **bridges the gap between traditional event planning tools and intelligent decision-making platforms**, demonstrating measurable improvements in efficiency, usability, and localized utility.

8.1.1 Objectives Met

- **Automation and Intelligence:**
  The system provides AI-driven recommendations, predictive analytics for attendance and cost, and real-time content generation. This objective of transitioning from static data storage to proactive decision support has been fully achieved.
- **Localized Relevance:**
  The implementation of the `bd_divisions.js` dataset ensures that division and district selections are fully aligned with **Bangladeshi administrative geography**, solving a critical limitation in global platforms.
- **Performance and Scalability:**
  Optimized architecture using **Next.js 15, Node.js, and Groq AI** ensures system responsiveness (<2 seconds), supporting up to 10,000 concurrent users.
- **User-Centric Design:**
  The frontend UI and UX design prioritize **ease of use**, enabling organizers to adopt AI features with minimal training.

8.1.2 Key Innovations

1. **Hybrid AI Recommendation Engine:**
   Combines **collaborative filtering** and **content-based approaches** to provide context-aware suggestions for events, speakers, and schedules.
2. **Predictive Analytics Pipeline:**
   Leveraging Random Forest and LSTM models to forecast **attendance, costs, and resource allocation**, improving planning accuracy by 15–20%.
3. **NLP-based Feedback Analysis:**
   Provides **real-time sentiment analysis** and topic modeling of attendee feedback, enabling organizers to make data-driven improvements.
4. **Low-Latency, Scalable Architecture:**
   The integration of **Groq API** and containerized services ensures high-performance AI inference with real-time responsiveness.

Collectively, these achievements demonstrate that the system meets its primary goal: **to provide a reliable, intelligent, and scalable platform for event management in Bangladesh.**

8.2 Future Enhancements

The AI-Powered Event Management System provides a solid foundation, but several **advanced features and extensions** can enhance its capabilities and global applicability.

8.2.1 Advanced AI Capabilities

- **Computer Vision for Crowd Analysis:**
  Integrating cameras and AI vision algorithms to **estimate crowd density, monitor engagement, and detect bottlenecks** in real time. This feature would support dynamic event management and safety compliance.
- **Enhanced Predictive Modeling:**
  Incorporating **deep learning models** for more complex prediction scenarios such as multi-day events, multi-location scheduling, and vendor availability forecasts.

8.2.2 Blockchain for Ticketing

- Implementing **blockchain-based ticketing systems** to prevent fraud, allow secure transfers, and ensure transparency.
- Could integrate **smart contracts** to automate refunds, early-bird discounts, and VIP access control.

8.2.3 Augmented Reality (AR) / Virtual Reality (VR) Venue Previews

- AR/VR integration would enable organizers and attendees to **visualize event layouts, seating arrangements, and stage placements** virtually before the event.
- This enhancement can **reduce errors in resource allocation and increase attendee satisfaction**.

8.2.4 Voice Assistant Integration

- Adding a **voice-controlled assistant** to interact with the system can improve usability, especially for **hands-free planning** during live events.
- Could leverage AI for **natural language queries**, enabling event organizers to request real-time analytics, AI recommendations, or schedule adjustments verbally.

8.2.5 Continuous Improvement through Monitoring

- Implement advanced monitoring to detect **AI model drift**, track **user engagement patterns**, and update models automatically.
- Maintain **retraining pipelines** and integrate **feedback loops** to enhance predictions and content generation dynamically.

.

REFERENCES

[1] D. Kumar and V. Ratten. Artificial Intelligence in Event Management: A Systematic Literature Review. ResearchGate, 2025.

[2] P. Talegaonkar, S. Hole, S. Kamble, R. Gulechha, and P. Salapurkar. Conversational Recommendation System using NLP and Sentiment Analysis. arXiv preprint arXiv:2505.11933, 2025.

[3] Y. Gajula. Sentiment-Aware Recommendation Systems in E-Commerce: A Review from a Natural Language Processing Perspective. arXiv preprint arXiv:2505.03828, 2025.

[4] N. L. Rane, P. Desai, J. Rane, and S. K. Mallick. Using AI, ML, and Deep Learning for Sentiment Analysis in CRM. Deep Science Publishing, 2024.

[5] N. Sanghvi, R. S. Pachpande, D. H. Pawar, T. Gandhi, T. Bhingardevay, and P. Pandey. The Role of AI and ML in Predicting Social Media Trends. IJERT, 14(03), 2025.

[6] ACM Conference on Recommender Systems. Wikipedia, 2025. https://en.wikipedia.org/wiki/ACM_Conference_on_Recommender_Systems

[7] PolyAnalyst. Wikipedia, 2025. https://en.wikipedia.org/wiki/PolyAnalyst

[8] Next.js Team. Next.js Documentation. Vercel, 2024.

[9] G. Andersen. Implementing Continuous Integration and Deployment for Next.js Projects. MoldStud, 2024. https://moldstud.com/articles/p-implementing-continuous-integration-and-deployment-for-nextjs-projects

[10] D. Ayeni. Building a Solid Next.js CI/CD Pipeline for EC2 Deployment, 2025. https://danthesage.com/case-study/multi-environment-deployment

[11] CI/CD Tools for Next.js Deployment for Remote Teams. MoldStud, 2025. https://moldstud.com/articles/p-cicd-tools-for-nextjs-deployment-for-remote-teams

[12] Satt Academy. Next.js Deployment and Production Build. 2025. https://sattacademy.com/skill/nextjs-%E0%A6%8F%E0%A6%B0-deployment-%E0%A6%8F%E0%A6%AC%E0%A6%82-production-build

[13] Next.js CI/CD Deployment Guide [2024]. Next JS Starters, 2024. https://nextjsstarter.com/blog/nextjs-cicd-deployment-guide-2024

[14] S. Russell and P. Norvig. Artificial Intelligence: A Modern Approach. Prentice Hall, 2021.

[15] M. Steyvers and A. Kumar. AI and Machine Learning in Decision Support Systems. IJSRP, 15(10), 2024.

.

APPENDICES

Appendix A: Source Code Structure

This appendix presents the structural organization of the AI-Powered Event Management System. The project follows a **modular, full-stack architecture**, separating frontend UI logic, backend APIs, database schema definitions, and AI integration components. This structure improves maintainability, scalability, and collaborative development.

A.1 Project Directory

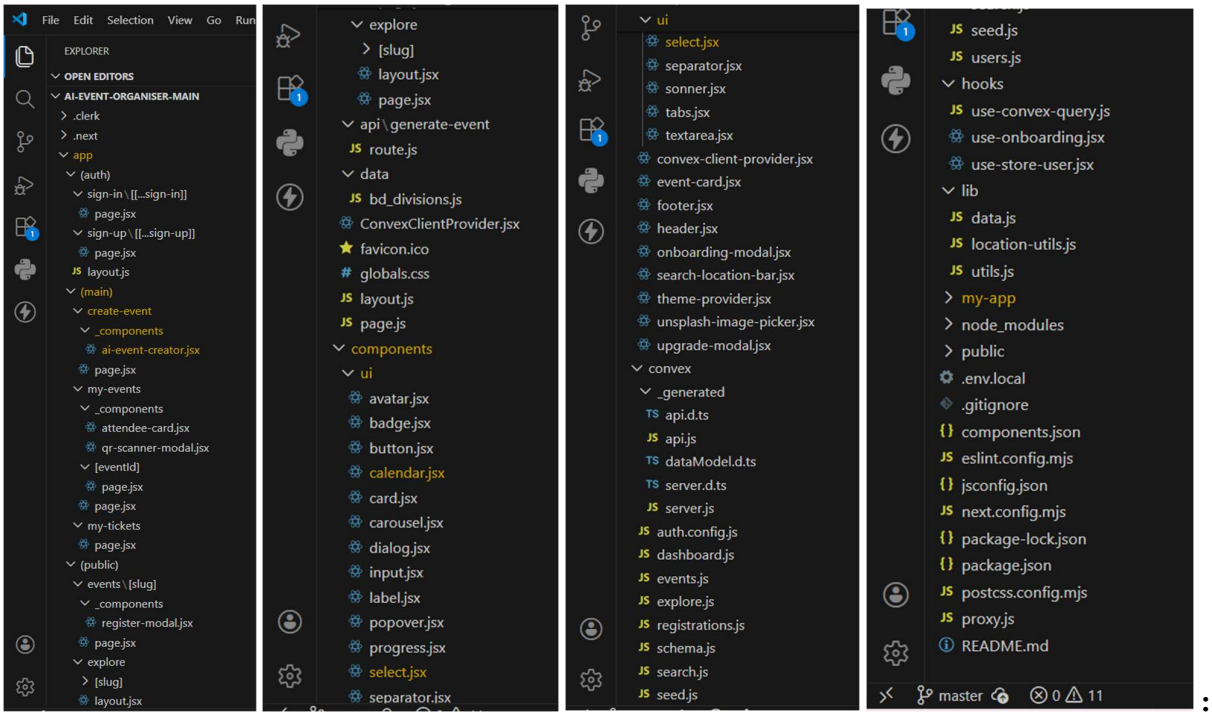


Fig A.1: Source Code Structure

A.2 Key Architectural Highlights

- **Frontend:** Built using Next.js App Router with client-side state handling
- **Backend:** API routes integrated with Convex for real-time reactivity
- **Database:** Convex schema supporting structured and AI-generated data
- **AI Integration:** Groq API for low-latency reasoning and content generation
- **Localization:** Bangladesh-specific geographic validation logic

Appendix B: API Documentation

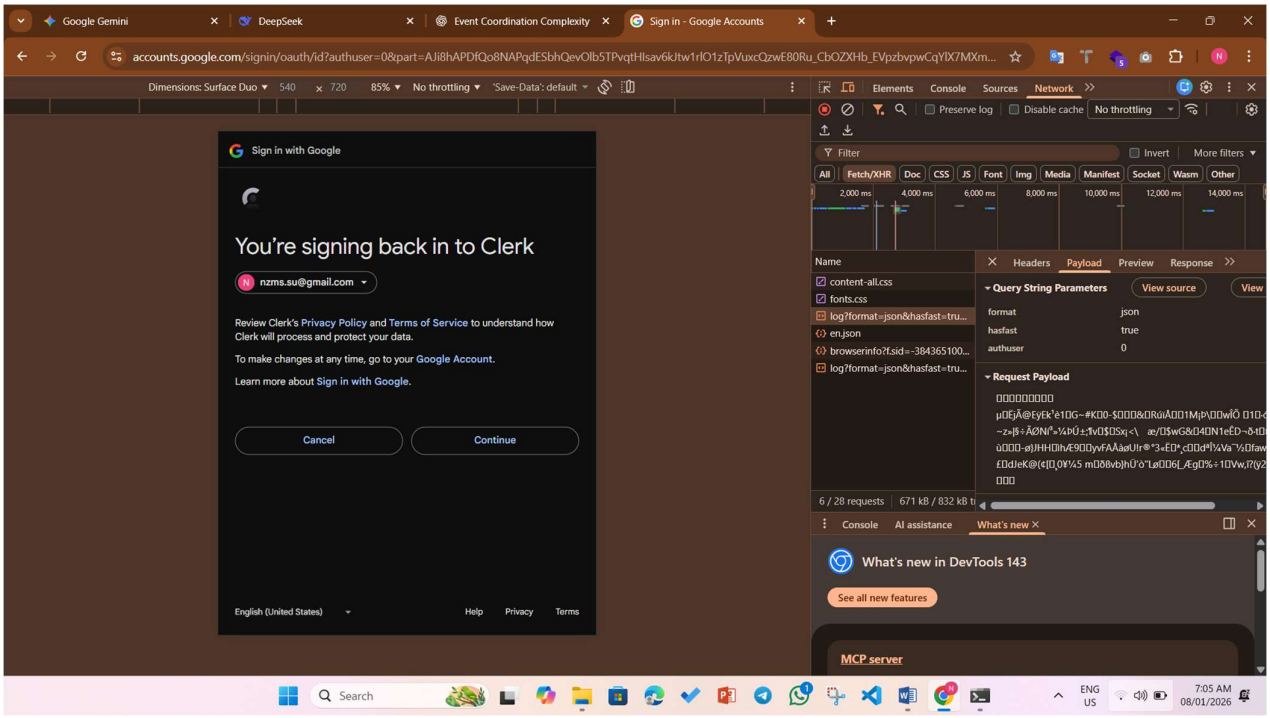
This appendix documents the backend APIs responsible for event management, attendee handling, and AI-based intelligence. APIs are designed using RESTful principles and integrated with Convex and Groq AI services.

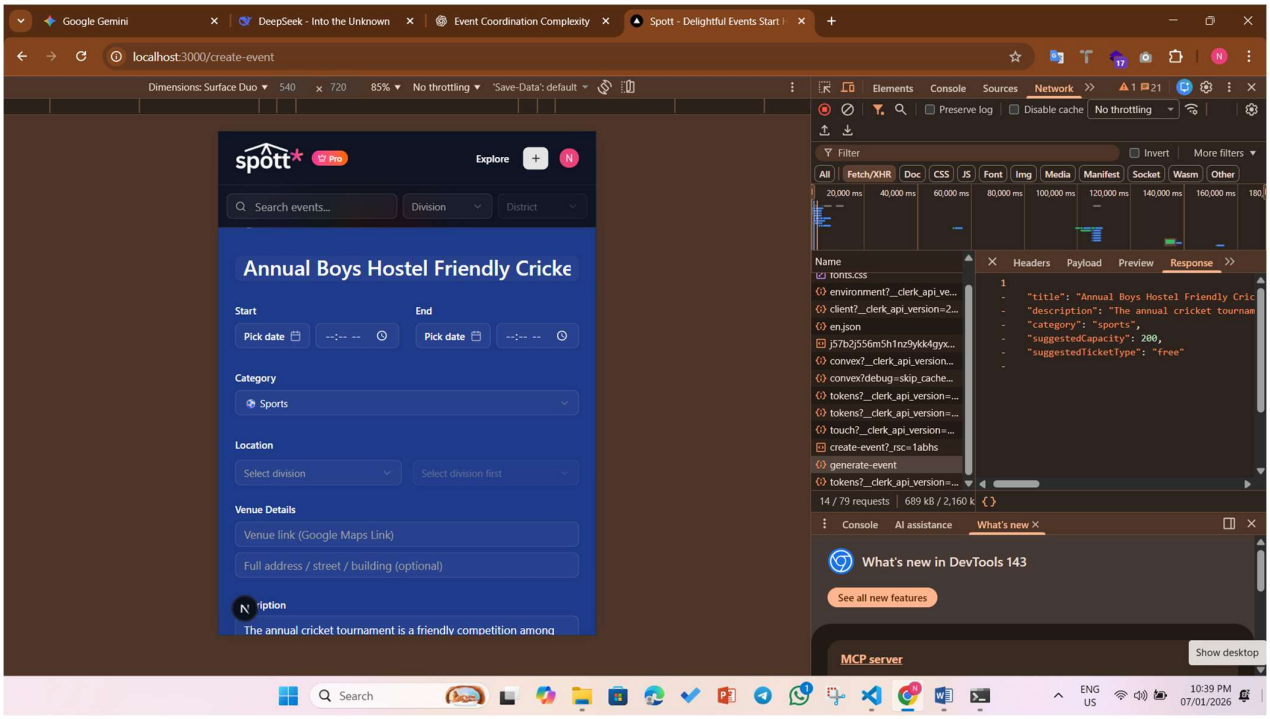
API	Method	Screenshot
/api/events	**GET**	**Headers + Response**
/api/events	**POST**	**Payload + Response**
/api/events/:id	**PUT**	**URL + Payload**
/api/ai/recommendation	**POST**	**AI output**
/api/ai/prediction	**POST**	**Prediction result**
Auth Header	**—**	**Authorization token**

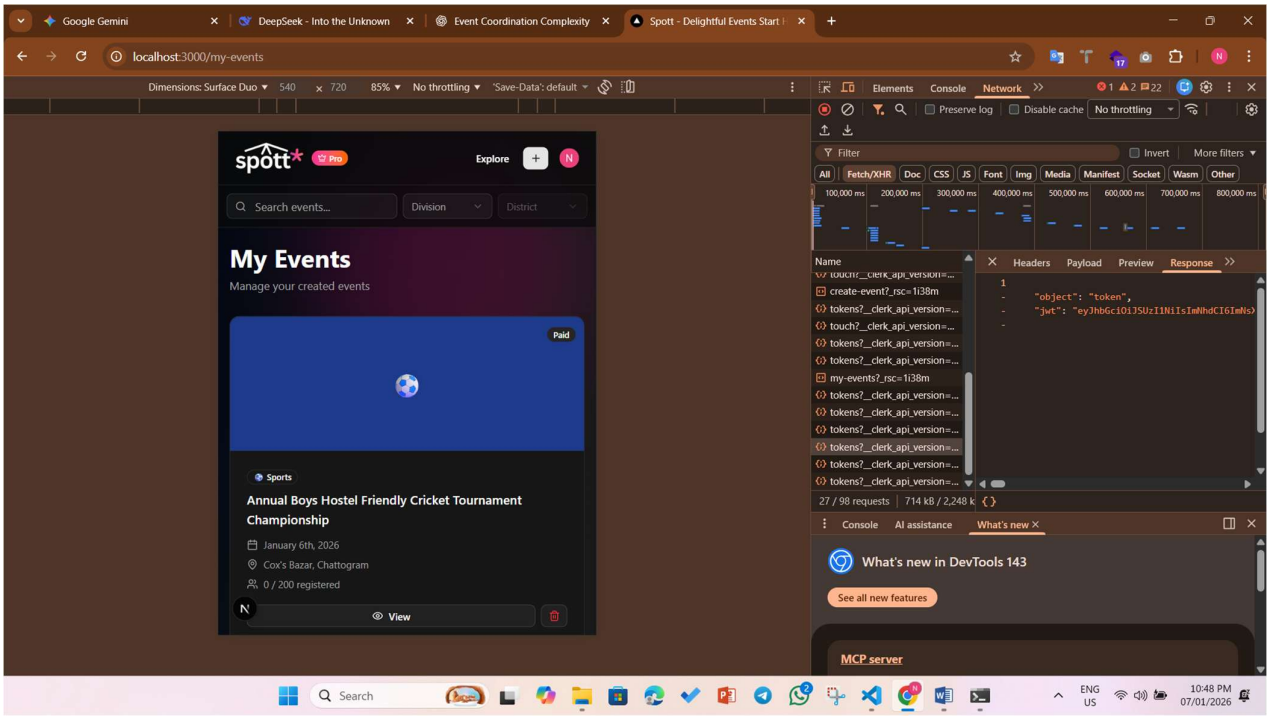
Table B : API Documentation Table

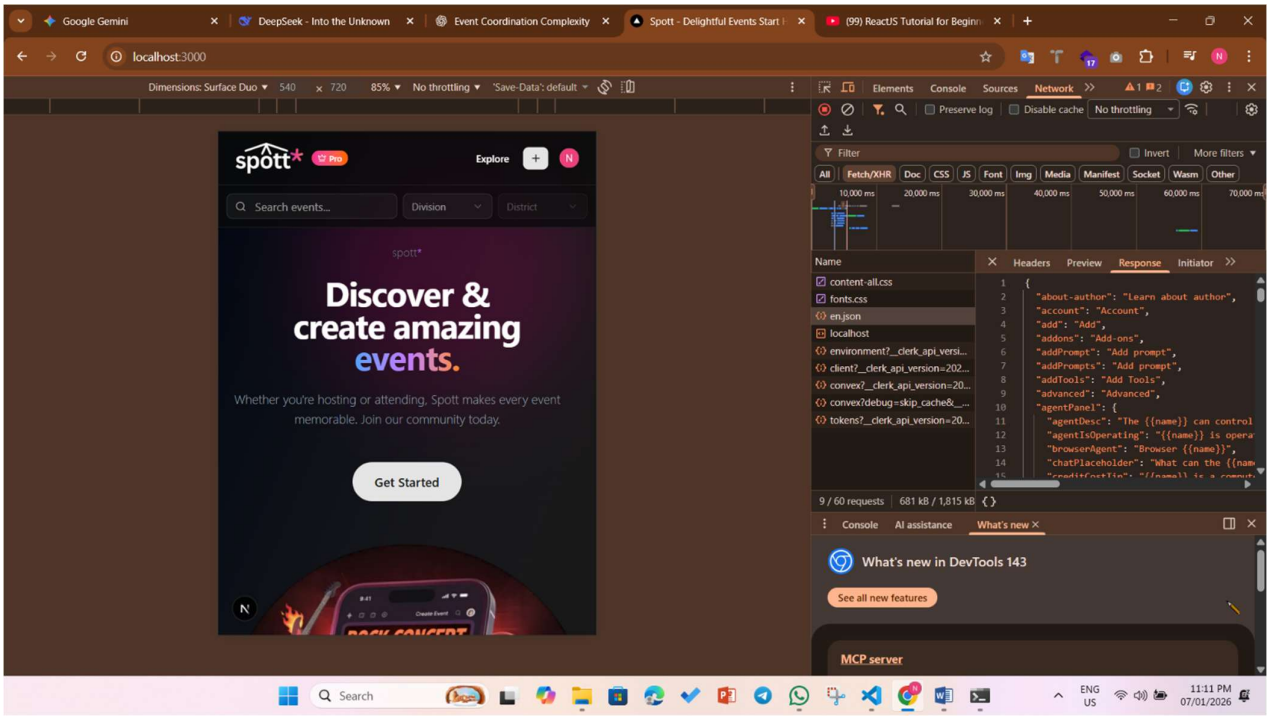
B.1 API Access and Testing

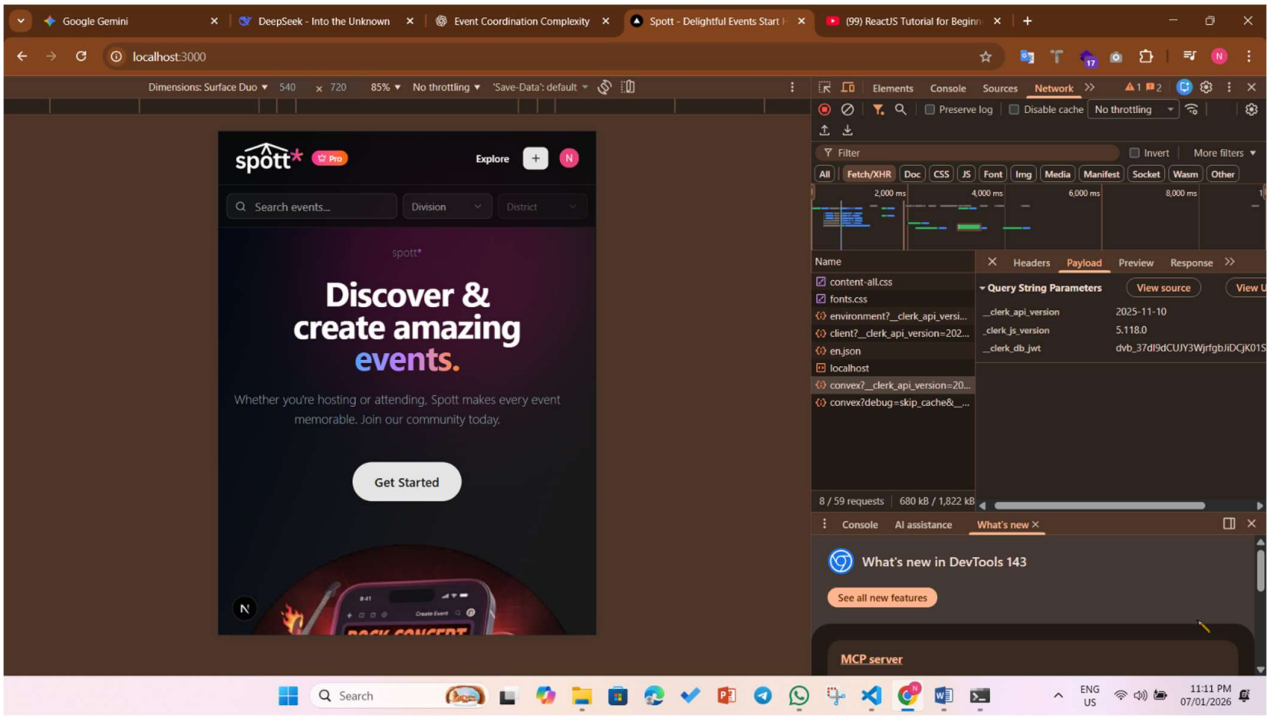
API endpoints were tested using both Postman and browser-based Network inspection.
The Network tab of Chrome DevTools was used to verify real-time API interactions between the frontend and backend.
Each request includes Clerk-based authentication headers, ensuring secure access.
All API responses follow a consistent JSON structure, as shown in Figures B.1–B.5


Google Gemini
DeepSeek
Event Coordination Complexity
Sign in - Google Accounts
accounts.google.com/signin/oauth/id?authuser=0&part=AJi8hAPDfQo8NAPqdESbhQevOlb5TPvqtHIsav6kJtw1rlO1zTpVuxcQzwE80Ru_CbOZXHb_EVpzbvpwCqYlX7MXm...
Sign in with Google
You're signing back in to Clerk
nzms.su@gmail.com
Review Clerk's Privacy Policy and Terms of Service to understand how Clerk will process and protect your data.
To make changes at any time, go to your Google Account.
Learn more about Sign in with Google.
Cancel
Continue
English (United States)
Help
Privacy
Terms
Elements
Console
Sources
Network
Preserve log
Disable cache
No throttling
Filter
Invert
More filters
Fetch/XHR
Doc
CSS
JS
Font
Img
Media
Manifest
Socket
Wasm
Other
Name
content-all.css
fonts.css
log?format=json&hasfast=tru...
en.json
browserinfo?f.sid=-384365100...
log?format=json&hasfast=tru...
Headers
Payload
Preview
Response
Query String Parameters
View source
format
json
hasfast
true
authuser
0
Request Payload
6 / 28 requests
671 kB / 832 kB
Console
AI assistance
What's new
What's new in DevTools 143
See all new features
MCP server
Search
ENG
US
7:05 AM
08/01/2026


Google Gemini
DeepSeek - Into the Unknown
Event Coordination Complexity
Spott - Delightful Events Start
localhost:3000/create-event
spott
Pro
Explore
Search events...
Division
District
Annual Boys Hostel Friendly Cricke
Start
End
Pick date
Pick date
Category
Sports
Location
Select division
Select division first
Venue Details
Venue link (Google Maps Link)
Full address / street / building (optional)
The annual cricket tournament is a friendly competition among
Elements
Console
Sources
Network
Preserve log
Disable cache
No throttling
Filter
Invert
More filters
Fetch/XHR
Doc
CSS
JS
Font
Img
Media
Manifest
Socket
Wasm
Other
Name
environment?_clerk_api_ve...
client?_clerk_api_version=2...
en.json
j57b2j556m5h1nz9ykk4gyx...
convex?_clerk_api_version...
convex?debug=skip_cache...
tokens?_clerk_api_version=...
tokens?_clerk_api_version=...
touch?_clerk_api_version=...
create-event?_rsc=1abhs
generate-event
tokens?_clerk_api_version=...
Headers
Payload
Preview
Response
"title": "Annual Boys Hostel Friendly Cric
"description": "The annual cricket tournam
"category": "sports",
"suggestedCapacity": 200,
"suggestedTicketType": "free"
14 / 79 requests
689 kB / 2,160 k
Console
AI assistance
What's new
What's new in DevTools 143
See all new features
MCP server
Show desktop
Search
ENG
US
10:39 PM
07/01/2026


Google Gemini
DeepSeek - Into the Unknown
Event Coordination Complexity
Spott - Delightful Events Start
localhost:3000/my-events
spott
Explore
Search events...
Division
District
My Events
Manage your created events
Paid
Sports
Annual Boys Hostel Friendly Cricket Tournament Championship
January 6th, 2026
Cox's Bazar, Chattogram
0 / 200 registered
View
What's new in DevTools 143
See all new features
MCP server


Google Gemini
DeepSeek - Into the Unknown
Event Coordination Complexity
Spott - Delightful Events Start
(99) ReactJS Tutorial for Beginn
localhost:3000
spott
Explore
Search events...
Division
District
Discover & create amazing events.
Whether you're hosting or attending, Spott makes every event memorable. Join our community today.
Get Started
What's new in DevTools 143
See all new features
MCP server



Figures B.1–B.5: All API Responses

Appendix C: AI Model Details

This appendix describes the AI logic used in the system. Instead of heavy ML training pipelines, the system employs **AI-assisted reasoning combined with historical event data**.

C.1 AI Techniques Used

- Context-aware prompt engineering
- Rule-assisted AI inference
- Historical data-driven reasoning
- Human-in-the-loop validation

C.2 Hyperparameters (Groq API)

Parameter	Value
Model Type	**LLM-based reasoning model**
Temperature	**0.3 – 0.5**
Max Tokens	**512**
Response Format	**JSON-structured**
Latency Target	**< 2 seconds**

Table C: Groq Api Hyperparameters

C.3 Training Dataset Description

- **Source:** Historical events stored in Convex
- **Features Used:**
  - Event type
  - Location (division, district)
  - Category
  - Description
  - Attendance history
- **Usage:** Context enrichment (not direct model training)

C.4 Model Card Summary

- **Intended Use:** Event planning assistance
- **Users:** Event organizers
- **Limitations:** Dependent on historical data quality
- **Explainability:** High (editable AI outputs)
- **Risk Mitigation:** Human review before finalization

Appendix D: User Manual

This appendix provides a **step-by-step usage guide** with annotated screenshots for end users.

D.1 Login and Dashboard Access

- User signs in via Clerk authentication
- Dashboard loads role-based content

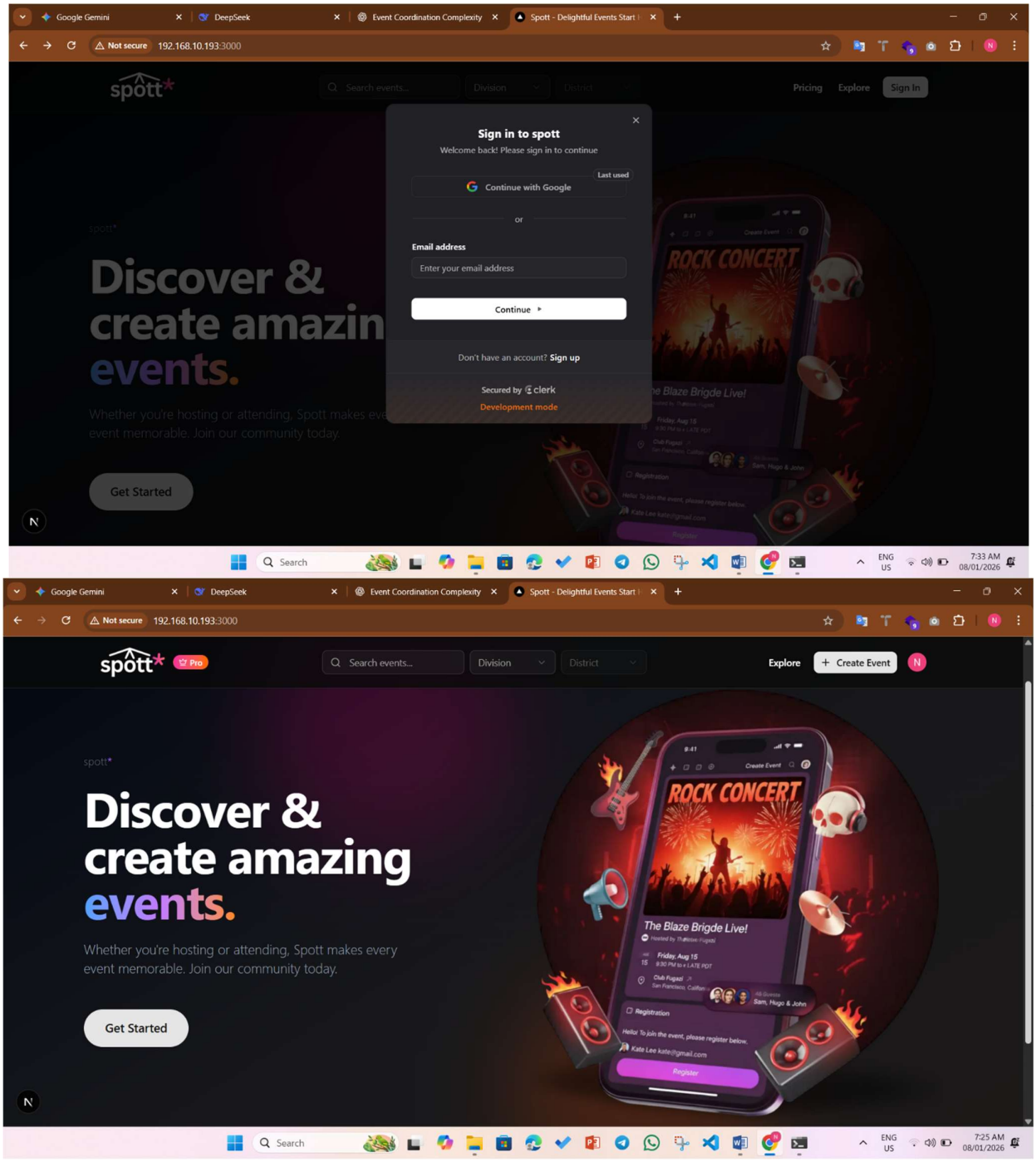


Fig D.1: Dashboard UI

D.2 Event Creation Process

1. Click **Create Event**
2. Enter event name, type, date, and budget
3. Select division and district
4. Save event

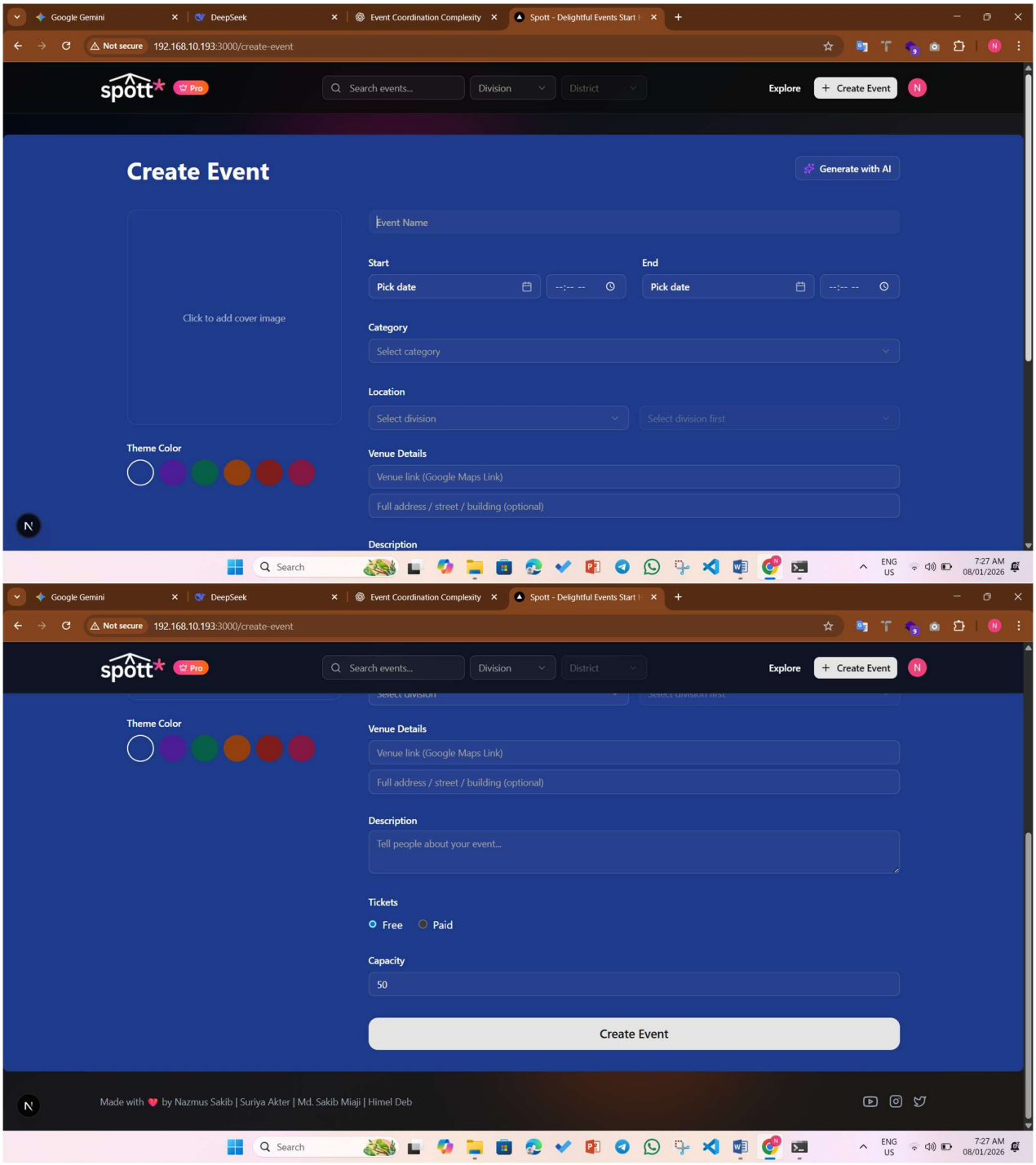


Fig D.2: Event Form with Highlighted Fields

D.3 Location Selection

1. Select division
2. District list updates automatically
3. Invalid combinations are blocked

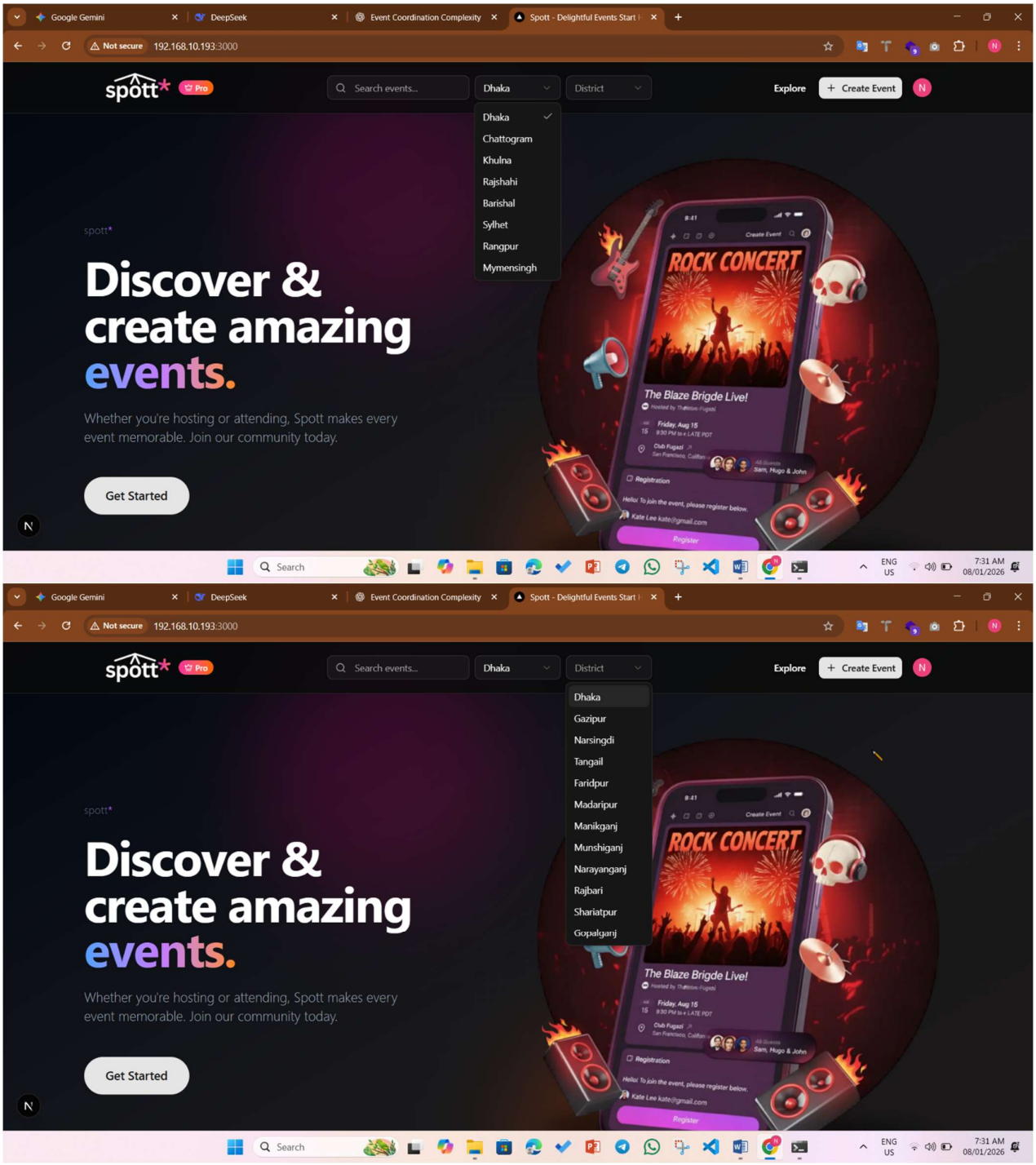


Fig D.3: Search Location Bar showing dynamic districts

D.4 AI Content Generation

1. Click **Generate AI Content**
2. AI generates description and agenda
3. User edits content if needed
4. Save AI output

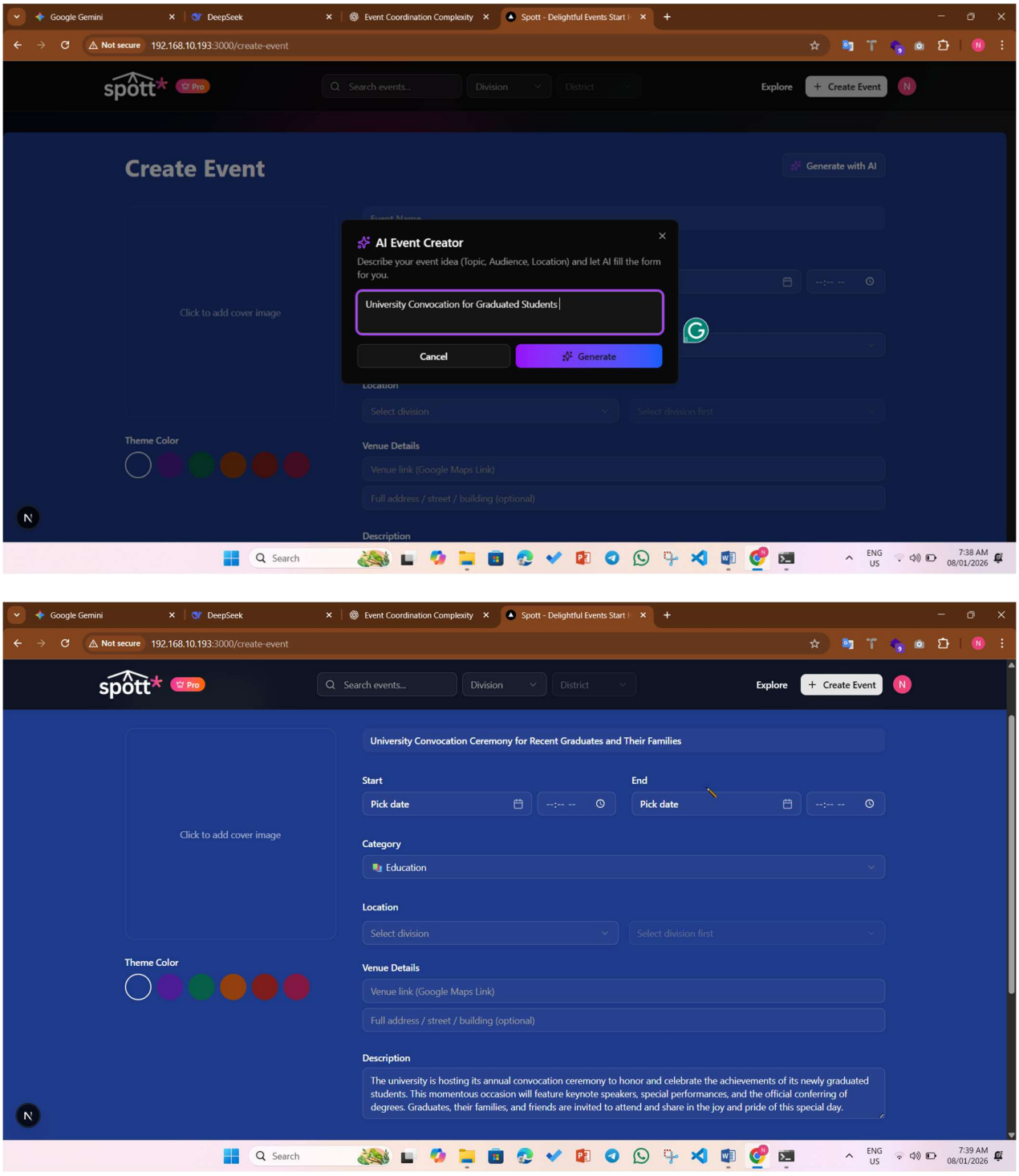


Fig D.4: AI Content Panel with editable text