

Campus Event Management System

Overview

This is a full-stack campus event management system built with React and Express. The application enables colleges to manage campus events, student registrations, attendance tracking, and feedback collection. It provides a comprehensive dashboard for event organizers to create events, track participation, and generate reports on event performance and student engagement.

User Preferences

Preferred communication style: Simple, everyday language.

System Architecture

Frontend Architecture

Framework: React with TypeScript Routing: Wouter for client-side routing State Management: TanStack Query for server state management UI Components: Radix UI components with shadcn/ui design system Styling: Tailwind CSS with CSS variables for theming Form Handling: React Hook Form with Zod validation Build Tool: Vite for development and bundling

Backend Architecture

Runtime: Node.js with Express.js REST API Language: TypeScript with ES modules Database ORM: Drizzle ORM for type-safe database operations Validation: Zod schemas shared between client and server API Design: RESTful endpoints with proper HTTP status codes and error handling

Database Design

Database: PostgreSQL with Neon serverless hosting Schema: Relational design with the following entities: - Colleges: Institution management - Events: Event details with college association - Students: Student profiles linked to colleges - Registrations: Event registration tracking - Attendance: Attendance marking for registered students - Feedback: Event feedback with ratings and comments Relationships: Proper foreign key constraints with cascade deletes

Key Features

Dashboard: Real-time statistics and analytics Event Management: Create, update, and delete events Student Management: Register and manage student profiles Registration System: Students can register for events with capacity limits Attendance Tracking: Mark and track student attendance Feedback Collection: Collect ratings and comments for events Reporting: Generate participation and engagement reports

Development Environment

Package Manager: npm with lockfile versioning Development Server: Vite dev server with HMR Database Migrations: Drizzle Kit for schema management Code Quality: TypeScript strict mode with path aliases

External Dependencies

Core Framework Dependencies - @tanstack/react-query: Server state management and caching - drizzle-orm: Type-safe ORM for database operations - @neondatabase/serverless: Neon PostgreSQL serverless driver - wouter: Lightweight React router - zod: Runtime type validation and schema definition

UI and Styling

- @radix-ui/*: Headless UI component primitives - tailwindcss: Utility-first CSS framework - class-variance-authority: Component variant management - clsx: Conditional className utility

Development Tools

- vite: Build tool and dev server - typescript: Type checking and compilation - @replit/vite-plugin-runtime-error-modal: Development error handling - drizzle-kit: Database schema management and migrations

Database and Storage

- Neon Database: Serverless PostgreSQL hosting - connect-pg-simple: PostgreSQL session store (configured but not actively used)

Form and Validation

- react-hook-form: Form state management - @hookform/resolvers: Form validation resolvers - drizzle-zod: Generate Zod schemas from Drizzle tables