# Fastbreak Event Dashboard

**Challenge Description:**
Build a full-stack Sports Event Management application where users can create, view, and manage sports events with venue information.

**Technical Requirements:**

- **Framework:** Next.js 15+ (App Router)
- **Language:** TypeScript
- **Database:** Supabase
- **Styling:** Tailwind CSS
- **UI Components**: Shadcn
- **Deployment:** Vercel
- **Authentication:** Supabase Auth (Email and/or Google SSO)

**Core Requirements**

- **Authentication**
  - Sign up / Login with email & password or Google OAuth Sign-in
  - Protected routes (redirect to login if not authenticated)
  - Logout functionality
- **Dashboard**
  - When users login, they should be taken to a home page / dashboard that:
    - Display list of all sports events
    - Show key event details: name, date, venue, sport type
    - Navigate to create/edit event forms
    - Responsive grid/list layout
    - Search by name, filter by sport - should refetch from the database
- **Event Management**
  - User should be able to create events:
    - Event name
    - Sport type (e.g., Soccer, Basketball, Tennis)
    - Date & Time
    - Description
    - Venues (Plural)
  - User should be able to edit events
  - User should be able to delete events

**Additional Information**

- All database interactions MUST happen server-side
  - Server Actions
  - API Routes (Route Handlers)

    - Server Components
- NO direct Supabase client calls from client components
- Actions over API Routes. At Fastbreak, we’re pushing towards using only actions, instead of API routes.
    - Create generic helper(s) to ensure type safety in the application and consistent error handling.
- Use shadcn/ui components throughout
- Forms MUST use shadcn Form component with react-hook-form (https://ui.shadcn.com/docs/components/form)
- Consistent styling with Tailwind CSS
- Loading states and error handling
- Toast notifications for success/error states

**Submission**

- Deployed to Vercel (or other hosting platform) with a working public URL
- Code submitted via GitHub repository

**Time Expectation:** 2-3 hours