

Scope

This document covers only the frontend implementation. Backend, database, and infrastructure will be handled by the other team.

Overview

- Built with React + Vite + TypeScript
- White + green consistent theme
- Smooth UI/UX with Framer Motion animations
- All views connected to backend API endpoints
- Modular, maintainable components

Key Features Implemented

- Events & Coaches:
 - Display cards with soft shadows, gradients, rounded corners
 - Hover animations + smooth transitions
 - Filtering + pagination (category/date/price)
- Profile:
 - Editable profile with validation
 - Consistent card-based layout
- Bookings:
 - "My Bookings" page connected to backend `/api/bookings/my-bookings`
 - Cancel booking action updates UI instantly
- Admin:
 - Add/Delete Coaches & Events (with image upload)
 - Toggle event "Sold Out" status
- Visual Polish:
 - Unified typography & spacing
 - Hover effects & card scaling
 - White/green palette

Frontend Env Variables

Must be set in ``.env`` (or ``.env.production`` for deployment):

`VITE_SUPABASE_URL=your-supabase-url`

`VITE_SUPABASE_ANON_KEY=your-supabase-anon-key`

Running Locally

`npm install`

`npm run dev`

`npm run build`

Code Structure

`src/`

`components/` # Reusable UI components (cards, modals, animations)

`pages/` # Page-level components (Events, Coaches, Profile, Bookings)

`services/` # API calls + Supabase integration

styles/ # Global styles + Tailwind config
App.tsx # Routing and layout

Integration Notes for Backend Team

- All data fetching is done via /api/... endpoints
- Image uploads expect backend storage handling
- Bookings/events/coaches rely on backend validation & DB structure
- Admin role check is backend-driven
- Supabase is only accessed via backend — frontend does not directly query DB

Final Note

Frontend is production-ready — only needs the backend endpoints and Supabase env values set before deployment.