

ConXion App Summary

WHAT IT IS

ConXion is a Next.js + Supabase MVP for trusted networking in the dance community. It helps members and trip travelers discover each other, connect with context, and message after acceptance.

WHO IT IS FOR

Primary persona: social dancers and dance ecosystem roles (organizers, teachers, DJs, artists, promoters, studio owners) who want local and travel connections.

WHAT IT DOES

- Email magic-link sign-in and callback session exchange via Supabase Auth.
- Discover members from profiles_feed and travellers from trips, with filters for location, roles, styles, languages, and interests.
- Send contextual connection requests with reason selection in member and trip flows.
- Manage connection requests (incoming/outgoing, accept, decline, cancel, block, report).
- Open message threads for accepted connections and send messages via `rpc("send_message")`.
- Maintain personal profile data (avatar upload, roles, skills, languages, interests, links) and trips.
- Provide admin tools for verification badges and profile field moderation behind an `admins-table` check.

HOW IT WORKS (REPO-EVIDENCED ARCHITECTURE)

- UI: Next.js App Router pages in `app/*`, React client components, TypeScript, and Tailwind.
- Data/Auth: browser Supabase client in `lib/supabase/client.ts` for auth and direct table reads/writes.
- APIs: `app/api/connect/route.ts` inserts connection rows server-side; `app/api/unsplash/route.ts` proxies/falls back for hero images.
- Storage: avatars and trip heroes are served from Supabase Storage URLs, with Unsplash fallback for city hero images.
- Local persistence: onboarding draft state is stored in `localStorage` via `lib/onboardingDraft.ts`.
- Data flow: Auth -> `/auth/callback` session exchange -> route pages query and update tables -> messaging sends via RPC and reloads thread data.
- Not found in repo: SQL migration/schema files for tables and RLS policies; formal backend service outside Next.js routes.

HOW TO RUN (MINIMAL)

1. Install dependencies: `npm install`
2. Create `.env.local` with:
 `NEXT_PUBLIC_SUPABASE_URL`
 `NEXT_PUBLIC_SUPABASE_ANON_KEY`
 `SUPABASE_SERVICE_ROLE_KEY` (server-only)
3. Start dev server: `npm run dev`
4. Open `http://localhost:3000`