

Podium Lab — Full Feature Inventory and Greenfield Rebuild & Test Guide

Version: 1.0 (generated from repository structure)

Stack reference: Expo 54, React Native + Web (react-native-web), Expo Router, Supabase, SWR, Zustand, Zod, Sentry, Reanimated, Capacitor, EAS Update, Jest. See package.json.

This document is **not** a request to clone the existing app. It records what exists today, labels **maturity** where obvious from code structure, and gives **phased build prompts** and **test prompts** for a faster, more reliable v2.

Part 1 — Purpose and principles

Goals for a greenfield v2

- **Perceived performance:** time-to-interactive, smooth navigation, no unnecessary request waterfalls, predictable loading states.
- **Reliability:** correct data for the signed-in role, RLS enforcement, empty and error states that help users.
- **Process:** ship **one vertical slice** at a time, run **verify + manual QA** before the next slice.

What this is not

- A pixel-perfect replica of v1.
- A mandate to build every v1 feature before the core value loop (assign work → member executes → progress visible) works end to end.

Maturity legend (used in the inventory)

Label	Meaning
Integrated	Clear data hooks, Supabase usage, and/or navigation to real screens; expect backend coupling.
Partial	Some wiring exists but gaps are likely (complex flows, edge cases, or inconsistent UX). Tester should assume verify.
UI shell	Mostly static or local mock data; building v2 may replace rather than “connect” this screen.

Known v1 example — UI shell: `app/(admin)/menu/index.tsx` uses in-file `menuCategories / menuItems` and `stock images`. Treat as design reference, not a data contract.

Nav vs routes — Partial risk: The coach sidebar in `components/AdminSidebar.tsx` lists items such as `Segments`, `Onboarding`, `Tags`, `Profile Fields`, and `Activity` under **Members** that do **not** have matching route files under `app/(admin)/members/` in this repo (only `access-groups`, `audience`, `bulk-import`, `bulk-logs`, `invite-links`, `index`, `parent-invites`, `[userId]` exist). Those links are **likely broken or legacy** until confirmed in runtime.

Tabs group — Partial risk: `app/(tabs)/_layout.tsx` registers `calendar`, `teams`, and `settings` tab screens, but the repository under `app/(tabs)/` only contains `index`, `library`, and nested `library` routes. Expect missing tab screens or redirect stubs in v1 unless added elsewhere.

Part 2 — Cross-cutting v2 standards (apply from day one)

Data and security

- **RLS on by default** for all member-facing tables; ship SQL policy tests in CI (see npm run test:rls in package.json as a v1 pattern).
- **Idempotent** client fetches: stable keys, avoid duplicate in-flight requests for the same resource.
- **Role routing**: server truth (JWT + RLS) must match what the client shows; never hide UI only on the client for security.

Client performance (web and native)

- **Route-level code splitting** on web for heavy admin areas.
- **Parallelize** independent reads; avoid “await A then B” when A and B do not depend on each other.
- **Images**: explicit dimensions, caching, avoid enormous remote assets on first paint.
- **Lists**: flat list patterns, stable keys, pagination or windowing for long feeds.

UX quality

- **Loading**: skeletons or inline spinners; avoid blank screens.
- **Optimistic updates** only when rollback is clear (messaging) or when conflicts are rare.
- **Error boundary** at app shell; recoverable errors inline with retry.

Testing pyramid (grow over time)

1. **RLS / SQL** — critical for multi-tenant trust.
2. **Unit tests** — pure functions (e.g. nutrition math, date helpers).
3. **Narrow E2E** — the core loop and billing touches only as they mature.

Part 3 — Product inventory by persona and domain

Priority tags

Tag	Suggested meaning
P0	Core loop and retention; v2 should not launch without.
P1	Major differentiators; next after P0.
P2	Valuable but can follow or be feature-flagged.

Platform and shell (P0)

Feature	Surfaces / paths	Maturity	Notes
Root layout, fonts, splash	app/_layout.tsx	Integrated	Auth timeout, Sentry, env inspection.
Theming	contexts/ThemeContext.tsx	Integrated	
Team context / switcher	contexts/TeamContext.tsx	Integrated	Drives most admin data.
Auth gating (shells)	hooks/useAppShellAuthGate, AdminLayout, AthleteLayout, ParentLayout	Integrated	Feature flags, app

Whitelabel	lib/whitelabel.ts	Integrated name, colors.
Public legal	app/terms-of-service.tsx, app/privacy-policy.tsx, app/terms.tsx	Partial
Not found	app/+not-found.tsx	Integrated
Error boundary	components/ErrorBoundary.tsx	Integrated
Team landing / join	app/team/[slug].tsx, app/join/[code].tsx, app/join/parent/[code].tsx	Partial
Tabs alternate shell	app/(tabs)/ (home, library, ...)	Partial See Part 1 nav note.

Authentication and onboarding (P0)

Feature	Surfaces / paths	Maturity	Notes
Sign in / sign up	app/(auth)/sign-in.tsx, sign-up.tsx, podium-access.tsx	Integrated	
Email verify	app/(auth)/verify-email.tsx	Integrated	
Reset password	app/(auth)/reset-password.tsx	Integrated	
Onboarding flow	app/(onboarding)/ — index, role-select, organization-setup, team-setup, plan-selection, features, feature-config	Partial	
Coach			
Admin home / dashboard	app/(admin)/index.tsx → dashboard, app/(admin)/dashboard.tsx	Integrated	
Admin menu (cafe-style)	app/(admin)/menu/index.tsx	UI shell	Mock data.
Account / branding / coach profile	app/(admin)/account-settings.tsx, branding.tsx, coach-profile.tsx	Partial	
Athlete (client)			
Client home	app/(client)/index.tsx	Integrated	
Profile	app/(client)/profile/index.tsx	Integrated	
Settings	app/(client)/settings/index.tsx	Integrated	
Health	app/(client)/health.tsx	Partial	
Privacy	app/(client)/privacy/index.tsx	Partial	
Parent			
Parent home	app/(parent)/index.tsx	Partial	
Parent profile / settings	app/(parent)/profile/index.tsx, app/(parent)/settings/index.tsx	Partial	

Library and training content (P0 / P1)

Feature	Surfaces / paths	Maturity	Notes
Library hub (admin)	app/(admin)/library/index.tsx, LibraryContent.tsx	Integrated	
Exercises	app/(admin)/library/exercise/index.tsx; tabs: (tabs)/library/exercise	Partial	
Workouts (templates)	app/(admin)/library/workouts/index.tsx	Partial	
Programs / systems	app/(admin)/library/programs/index.tsx, programs/[id].tsx	Partial	“Systems” in UI.

Video vault	app/(admin)/library/videos/index.tsx	Partial	
Prehab	app/(admin)/library/prehab/index.tsx	Partial	
Skills	app/(admin)/library/skills/index.tsx	Partial	
Metrics	app/(admin)/library/metrics/index.tsx	Partial	
Quizzes	app/(admin)/library/quizzes/index.tsx	Partial	
Courses	app/(admin)/library/courses/index.tsx, courses/[id].tsx	Partial	
Nutrition library			
Recipes	app/(admin)/library/recipes/index.tsx	Partial	
Ingredients	app/(admin)/library/ingredients/index.tsx	Partial	
Meal plans	app/(admin)/library/meal-plans/index.tsx	Partial	
Athlete consumption			
Workouts	app/(client)/workouts/index.tsx, workouts/[id].tsx	P0, Partial	
Workout preferences	app/(client)/workout-preferences/index.tsx	Partial	
Tabs library mirrors	(tabs)/library/*	Partial	Simpler/mirror UIs.

Athlete: habits, goals, nutrition (P1)

Feature	Path	Maturity
Habits	app/(client)/habits/index.tsx, habits/new.tsx	Partial
Goals	app/(client)/goals/index.tsx	Partial
Nutrition	app/(client)/nutrition/index.tsx, log.tsx, targets.tsx	Partial

Scheduling and calendar (P0 / P1)

Feature	Path	Maturity Notes
Admin calendar	app/(admin)/calendar/index.tsx, events.tsx, upcoming.tsx, [eventId].tsx, division/[divisionId].tsx	Partial
Client calendar	app/(client)/calendar/index.tsx, calendar/[eventId].tsx	Partial
Parent calendar	app/(parent)/calendar/index.tsx	Partial

Communication (P1)

Feature	Path	Maturity
Inbox (admin)	app/(admin)/inbox/index.tsx, compose, [conversationId].tsx	Partial
Inbox (client)	app/(client)/inbox/...	Partial
Inbox (parent)	app/(parent)/inbox/index.tsx	Partial
Bulletin (admin/athlete/parent)	app/(admin)/bulletin-board/, app/(client)/bulletin-board/, app/(parent)/bulletin-board/	Partial
Forums	Admin, client, parent forum indexes + [topicId]; homework-hub, getting-started	Partial

Journaling and forms (P1)

Feature	Path	Maturity	
Admin journal	app/(admin)/journal/index.tsx	Partial	Forms and submissions.
Client journals / logs	app/(client)/journals/index.tsx	Partial	

Gamification and engagement (P1 / P2)

Feature	Path	Maturity
Gamification hub (admin)	app/(admin)/gamification/index.tsx	Partial
Badges, groups, competitions	badges/, groups/, competitions/index.tsx	Partial
Leaderboards	gamification/[leaderboardId]/ (admin + client)	Partial
Client gamification, achievements	app/(client)/gamification/, achievements/, competitions/	Partial

Members, teams, and organizations (P0 / P1)

Feature	Path	Maturity	Notes
Members list / detail	app/(admin)/members/index.tsx, members/[userId]/index.tsx	Partial	
Audience	members/audience/index.tsx	Partial	
Access groups	members/access-groups/index.tsx	Partial	
Invite links	members/invite-links/index.tsx	Partial	
Parent invites	members/parent-invites.tsx	Partial	
Bulk import / bulk logs	members/bulk-import/, members/bulk-logs/	Partial	
Teammates	app/(admin)/teammates/index.tsx	Partial	
Teams	app/(admin)/teams/index.tsx, new.tsx, teams/[teamId]/* (settings, members, events, media, news, performance, analytics, divisions)	Partial	Rich surface.
Organizations	app/(admin)/organizations/index.tsx, create.tsx	Partial	
Invite flow	app/(admin)/invite/index.tsx, invite/create.tsx	Partial	
Admin console approvals	app/(admin)/admin-console/approvals.tsx	Partial	
Sidebar-listed member routes (missing files)	e.g. /members/segments, onboarding, tags, profile-fields, activity-logs	Unknown / likely broken	No matching app files; verify in app or remove in v2.

Tasks and assignments (P1)

Feature	Path	Maturity
Tasks	app/(admin)/tasks/index.tsx	Partial

Marketplace, billing, and payments (P2, high risk)

Feature	Path	Maturity
Marketplace	app/(admin)/marketplace/index.tsx, browse/, create/, add-product.tsx	Partial
Billing	app/(admin)/billing/index.tsx	Partial
Payments	app/(admin)/payments/index.tsx	Partial
Affiliates	app/(admin)/affiliates/index.tsx	Partial
Seller onboarding	app/(admin)/settings/seller-onboarding.tsx	Partial

Integrations, automation, and analytics (P2)

Feature	Path	Maturity
Automation	app/(admin)/automation/index.tsx	Partial
Analytics	app/(admin)/analytics/index.tsx, AnalyticsContent.tsx	Partial
API keys	app/(admin)/settings/api-keys/index.tsx	Partial
Webhooks	app/(admin)/settings/webhooks/index.tsx	Partial
Email templates	app/(admin)/settings/email-templates/, domains/	Partial
Embed widgets	app/(admin)/settings/embed-widgets/index.tsx	Partial
Notifications (admin)	app/(admin)/notifications/index.tsx	Partial
Team / account settings	app/(admin)/settings/index.tsx, SettingsContent.tsx, team-manager.tsx	Partial

Site admin (P2 / internal)

Feature	Path	Maturity
Site admin entry and dashboard	app/(site-admin)/index.tsx, app/(site-admin)/dashboard.tsx	Partial Small stack.

Client email and messaging preferences (P1 / P2)

Feature	Path	Maturity
Email preferences	app/(client)/email-preferences/index.tsx	Partial
Message settings	app/(client)/message-settings/index.tsx	Partial

Part 4 — Phased build order (A–L) with prompts and manual QA

For each phase below:

1. **Build prompt** — what to implement in v2, in the right order inside the phase.
2. **Verify prompt** — give this to a reviewer or AI after code is up.
3. **Manual QA** — exact steps a human can run.
4. **Definition of done and gate** to the next phase.

Phase A — Foundation (P0)

Build prompt (copy to your AI or ticket):

Create a new Expo Router app (React Native + Web) with TypeScript strict mode, Zod-validated environment variables, and a single design token file (colors, spacing, type scale). Add Supabase auth

(email magic link or password per your product decision). After login, load the user's profile and team memberships. Implement role-based entry: coach (admin shell), athlete (client shell), parent (parent shell) using guarded layouts—no data-heavy screens yet. Add an error boundary, loading gate, and empty-state placeholder for “Select a team” on coach. Whitelabel read from env. Do not add marketplace, webhooks, or gamification. Performance: ensure root layout does not block interactivity for more than one serial network round-trip where avoidable; document parallelized calls.

Verify prompt:

Review the Phase A implementation. List: (1) any route reachable without auth that should not be, (2) any role that can read another team's data due to client-only checks, (3) places where RLS is assumed but not documented, (4) performance issues such as auth blocking fonts or team list blocking the shell. Suggest three concrete tests to add to CI or manual checklist.

Manual QA

1. Open the app in a private window; confirm unauthenticated users cannot see coach dashboard data.
2. Sign in as a user with one team; confirm team context loads and switching teams (if applicable) changes visible team id in UI.
3. Sign in as athlete and parent (test accounts); confirm each lands in the correct shell, not the coach shell.
4. Throttle network to “Slow 3G” in devtools (web) or network link conditioner (iOS); confirm app shows loading then content or a clear error—not a white screen.
5. **Negative:** Open an authenticated URL, then sign out in another tab; confirm next action does not show stale private data (or that session is invalidated correctly).

Definition of done: Auth, profile, team membership, and three shells with guards are production-runnable; env and error handling are explicit.

Do not start Phase B until manual QA 1–5 pass and **Verify** has no P0 security items open.

Phase B — First vertical slice: assign workout → complete → signal success (P0)

Build prompt:

Implement the minimal data model: exercises (or movement catalog entries), workout templates, and member assignments. Coach can assign a template to a member; athlete can open the assignment, mark sets/reps or “complete” as a single state machine, and the coach sees completion on a minimal dashboard or list. RLS: coaches only their team; athletes only their rows. Add indexes and policies in SQL migrations; run policy tests. Client: SWR (or your chosen data layer) with stable keys, no N+1 list explosion on first load. Web: lazy-load the admin library subtree.

Verify prompt:

Trace every CRUD path for assignment and completion. Can athlete A mark athlete B's assignment? Can a former member still update rows? List race conditions (double-tap complete, offline). Suggest RLS and UI fixes.

Manual QA

1. Coach: create two exercises and one workout template; assign to athlete.
2. Athlete: open assignment, complete; confirm state persists on refresh.
3. Coach: see completion status and timestamp (or “completed” flag).
4. **Negative:** As athlete, attempt to open another user's assignment URL (or ID); expect deny or 404 from API.

5. **Empty:** New team with no exercises—UI explains how to add first content.

Definition of done: End-to-end core value loop is reliable under RLS.

Do not start Phase C until this loop is stable for two members on one team.

Phase C — Library expansion: programs, video, prehab, skills, metrics, quizzes, courses (P1)

Build prompt:

*Add program/system grouping (workout sequences over time) after templates are stable. Then add optional media (video) with lazy load and storage policies. Add prehab, skills, metrics, quizzes, and courses in **dependency order**—only wire a new type when the previous pattern (list → detail → admin CRUD → athlete read) is tested. Reuse one list/detail pattern. Cap video autoplay; measure scroll performance on web.*

Verify prompt:

For each content type, document who can create, update, and delete, and which role can consume. Check storage path conventions and that public URLs are not over-exposed. Flag duplicate patterns to consolidate.

Manual QA (per type, repeat pattern)

1. Coach: create, edit, delete one item; confirm list updates.
2. Athlete: open consumer view; confirm read-only and correct team scoping.
3. **Negative:** wrong-team ID in URL.
4. **Performance:** open library index with 100+ items (seed data); scroll stays smooth; images do not jank the main thread.

Gate: All P1 content types you need for launch are covered by the same RLS and UX pattern; no one-off security exceptions.

Phase D — Calendar and events (P0 / P1)

Build prompt:

Integrate team calendar: events, optional RSVP, division-level views if required. Timezone: store UTC, display in user/team zone. Subscriptions/ICS or internal-only first—scope explicitly. Minimize refetch on navigation.

Verify prompt:

List timezone bugs (DST), duplicate events, and permission gaps for parent vs athlete vs coach.

Manual QA

1. Create event as coach; appears for athlete in calendar.
2. Move event; confirm all roles see new time.
3. Parent: see child-relevant events only.
4. **Negative:** user without access cannot open eventId from deep link.

Gate: Calendar correct for at least one timezone and one recurrence pattern if you support recurrence.

Phase E — Comms: inbox, bulletin, forums (P1)

Build prompt:

Implement real-time or polling inbox with conversation membership enforced by RLS. Bulletin: posts visible to the right audience segments. Forums: topic ACLs, moderation hooks if needed. Prefer optimistic send only with rollback. Rate-limit message creation server-side if possible.

Verify prompt:

Can a user add themselves to a conversation? Can a deleted user's messages leak? List XSS or rich-text risks (if any).

Manual QA

1. Send message coach → athlete; reply; read state updates.
2. Post bulletin; confirm intended audience only.
3. **Negative:** try conversationId of another user.
4. **Performance:** open inbox with long history; avoid loading all messages at once (pagination).

Gate: No cross-tenant message leakage in spot checks and RLS tests.

Phase F — Journals and forms (P1)**Build prompt:**

Define form schema (JSON or tables), assign to members, collect submissions, coach review. Export optional. Validate inputs server-side. Mobile-friendly forms (large tap targets, keyboard types).

Verify prompt:

Check file uploads if any: virus scanning / size / MIME, storage policies.

Manual QA: submit as athlete; review as coach; filter by type; test required-field validation; **negative:** another member's submission id.

Gate: Coach cannot see other teams' submissions; athlete cannot edit after lock if that is a rule.

Phase G — Nutrition: recipes, ingredients, meal plans, logging (P1)**Build prompt:**

Build nutrition on top of stable library patterns. ingredients → recipes (composition) → meal plans → assignment/logging. Reuse lib/nutrition.ts-style pure functions with unit tests. Large datasets: search and filter server-side.

Verify prompt:

Unit consistency (g vs oz), rounding, allergen display. Data integrity: recipe totals match ingredients.

Manual QA: create recipe; log meal; see totals; **negative** invalid serving size; performance with search on many items.

Gate: Nutrition numbers match test vectors in unit tests.

Phase H — Gamification (P1 / P2)**Build prompt:**

Connect points, badges, leaderboards, and competitions to **verifiable** events (workouts completed, journal entries)—avoid manual score editing except admin audit. Recalculate jobs idempotent. Feature-flag public

leaderboards.

Verify prompt:

Exploits: repeat POST to gain points, backdated events, team hopping for ranks.

Manual QA: complete workout; see points; leaderboard order; open competition with end date. **Negative:** non-member on leaderboard.

Gate: Scores match a fixed scenario in automated tests or seed script.

Phase I — Tasks / assignments (P1)

Build prompt:

Generalize assignments beyond workouts if the product needs homework or checklists. Unified “task” model or clear separation; due dates, reminders (optional), completion audit.

Verify prompt:

Orphaned tasks when member removed; permissions on bulk operations.

Manual QA: assign task, complete, report; test overdue display; **negative** cross-team.

Gate: Clear distinction from workout assignments if both exist; no duplicate UX confusion.

Phase J — Marketplace, billing, payments, affiliates (P2, high risk)

Build prompt:

Use Stripe (or your processor) with webhooks, idempotent checkout.session.completed, customer portal, test mode. Marketplace listings separate from core training; feature flags. Log every money-moving event. PCI: never log card numbers. Seller onboarding gating for payouts.

Verify prompt:

Duplicate webhook delivery, race on subscription status, failed partial refunds, affiliate attribution edge cases.

Manual QA: test mode: subscribe, cancel, fail card, success; webhook logs; **negative** tampered client price.

Gate: Finance review + staging soak before production traffic.

Phase K — Integrations: webhooks, public API, embeds, email templates, automation, analytics (P2)

Build prompt:

Outgoing webhooks with signed payloads and retry/backoff. Public API: keys scoped to team, rate limits, OpenAPI spec. Email templates with variable linting. Automation rules: explicit triggers, dry-run, guardrails. Analytics: event naming convention, PII policy.

Verify prompt:

Webhook secret leakage, API key in URL, embed XSS, SSRF in webhook targets if user-configurable.

Manual QA: trigger webhook to request bin; call API with valid/invalid key; **negative** read another team with wrong key.

Gate: Security review of integration surfaces; docs for every external system.

Phase L — Parent portal polish, site admin, native/Capacitor, OTA (P2)

Build prompt:

Parent: limited to linked athlete(s)—explicit relationship model. Site admin: internal-only routes, strong role check. Capacitor: only after web is stable. EAS Update: channel strategy (prod/preview), rollback playbooks.

Verify prompt:

Parent cannot escalate to coach; site admin cannot be spoofed by client role flags.

Manual QA: parent sees only correct bulletin/calendar; site admin on separate allowlist. Native: push registration, background refresh expectations.

Gate: App store review readiness checklist complete.

Part 5 — Reusable template (for any new slice)

Build prompt — include: *goal; roles; data tables + RLS intent; API shape; UI scope; non-goals; performance budget (TTI, list scroll); accessibility (labels for primary actions).*

Verify prompt — *threat model (cross-tenant, IDOR, auth bypass), edge cases, empty states, error states, offline behavior.*

Manual QA — *happy path; two negative paths; one performance check; one accessibility check (e.g. VoiceOver focus order for primary flow).*

Appendix A — app/ route file index (v1)

Sorted list of route-level screens under app/ (Excluding _layout only files). Use for diffing v2; some routes may be unreachable from current nav.

- (admin)/account-settings.tsx
- (admin)/admin-console/approvals.tsx
- (admin)/analytics/AnalyticsContent.tsx (imported)
- (admin)/analytics/index.tsx
- (admin)/automation/index.tsx
- (admin)/affiliates/index.tsx
- (admin)/billing/index.tsx
- (admin)/branding.tsx
- (admin)/bulletin-board/index.tsx
- (admin)/calendar/[eventId].tsx
- (admin)/calendar/division/[divisionId].tsx
- (admin)/calendar/events.tsx
- (admin)/calendar/index.tsx
- (admin)/calendar/upcoming.tsx
- (admin)/coach-profile.tsx
- (admin)/dashboard.tsx
- (admin)/forums/[topicId]/index.tsx
- (admin)/forums/getting-started.tsx
- (admin)/forums/homework-hub.tsx
- (admin)/forums/index.tsx

- (admin)/gamification/[leaderboardId]/index.tsx
- (admin)/gamification/badges/index.tsx
- (admin)/gamification/competitions/index.tsx
- (admin)/gamification/groups/index.tsx
- (admin)/gamification/index.tsx
- (admin)/inbox/[conversationId].tsx
- (admin)/inbox/compose.tsx
- (admin)/inbox/index.tsx
- (admin)/index.tsx (redirect to dashboard)
- (admin)/invite/create.tsx
- (admin)/invite/index.tsx
- (admin)/journal/index.tsx
- (admin)/library/courses/[id].tsx
- (admin)/library/courses/index.tsx
- (admin)/library/exercise/index.tsx
- (admin)/library/ingredients/index.tsx
- (admin)/library/index.tsx
- (admin)/library/LibraryContent.tsx
- (admin)/library/meal-plans/index.tsx
- (admin)/library/metrics/index.tsx
- (admin)/library/prehab/index.tsx
- (admin)/library/programs/[id].tsx
- (admin)/library/programs/index.tsx
- (admin)/library/quizzes/index.tsx
- (admin)/library/recipes/index.tsx
- (admin)/library/skills/index.tsx
- (admin)/library/videos/index.tsx
- (admin)/library/workouts/index.tsx
- (admin)/marketplace/add-product.tsx
- (admin)/marketplace/browse/index.tsx
- (admin)/marketplace/create/index.tsx
- (admin)/marketplace/index.tsx
- (admin)/menu/index.tsx
- (admin)/members/[userId]/index.tsx
- (admin)/members/access-groups/index.tsx
- (admin)/members/audience/index.tsx
- (admin)/members/bulk-import/index.tsx
- (admin)/members/bulk-logs/index.tsx
- (admin)/members/index.tsx
- (admin)/members/invite-links/index.tsx
- (admin)/members/parent-invites.tsx
- (admin)/notifications/index.tsx
- (admin)/organizations/create.tsx
- (admin)/organizations/index.tsx
- (admin)/payments/index.tsx
- (admin)/settings/api-keys/index.tsx
- (admin)/settings/email-templates/[type].tsx
- (admin)/settings/email-templates/domains/add.tsx
- (admin)/settings/email-templates/domains/[id].tsx
- (admin)/settings/email-templates/index.tsx
- (admin)/settings/embed-widgets/index.tsx
- (admin)/settings/index.tsx

- (admin)/SettingsContent.tsx (imported)
- (admin)/settings/seller-onboarding.tsx
- (admin)/settings/team-manager.tsx
- (admin)/settings/webhooks/index.tsx
- (admin)/tasks/index.tsx
- (admin)/teammates/index.tsx
- (admin)/teams/index.tsx
- (admin)/teams/new.tsx
- (admin)/teams/[teamId]/analytics.tsx
- (admin)/teams/[teamId]/divisions.tsx
- (admin)/teams/[teamId]/events.tsx
- (admin)/teams/[teamId]/index.tsx
- (admin)/teams/[teamId]/media.tsx
- (admin)/teams/[teamId]/members.tsx
- (admin)/teams/[teamId]/news.tsx
- (admin)/teams/[teamId]/performance.tsx
- (admin)/teams/[teamId]/settings.tsx
- (auth)/podium-access.tsx
- (auth)/reset-password.tsx
- (auth)/sign-in.tsx
- (auth)/sign-up.tsx
- (auth)/verify-email.tsx
- (client)/achievements/index.tsx
- (client)/bulletin-board/index.tsx
- (client)/calendar/[eventId].tsx
- (client)/calendar/index.tsx
- (client)/competitions/index.tsx
- (client)/email-preferences/index.tsx
- (client)/forums/[topicId].tsx
- (client)/forums/getting-started.tsx
- (client)/forums/index.tsx
- (client)/gamification/[leaderboardId]/index.tsx
- (client)/gamification/index.tsx
- (client)/goals/index.tsx
- (client)/habits/index.tsx
- (client)/habits/new.tsx
- (client)/health.tsx
- (client)/inbox/[conversationId].tsx
- (client)/inbox/compose.tsx
- (client)/inbox/index.tsx
- (client)/index.tsx
- (client)/journals/index.tsx
- (client)/message-settings/index.tsx
- (client)/nutrition/index.tsx
- (client)/nutrition/log.tsx
- (client)/nutrition/targets.tsx
- (client)/privacy/index.tsx
- (client)/profile/index.tsx
- (client)/settings/index.tsx
- (client)/workout-preferences/index.tsx
- (client)/workouts/[id].tsx
- (client)/workouts/index.tsx

- +not-found.tsx
 - join/[code].tsx, join/parent/[code].tsx
 - (onboarding)/feature-config.tsx, features.tsx, index.tsx, organization-setup.tsx, plan-selection.tsx, role-select.tsx, team-setup.tsx
 - privacy-policy.tsx
 - team/[slug].tsx, terms-of-service.tsx, terms.tsx
 - (parent)/bulletin-board/index.tsx, calendar/index.tsx, forums/[topicId].tsx, forums/index.tsx, inbox/index.tsx, index.tsx, profile/index.tsx, settings/index.tsx
 - (site-admin)/dashboard.tsx, index.tsx
 - (tabs)/index.tsx, (tabs)/library/... (exercise, index, meal-plans, metrics, prehab, programs, skills, workouts)
-

End of guide.