GymBag Coding Agent Prompt Pack

Copy/paste any of the prompts below into your coding agent. Everything here mirrors the earlier playbook so you can run it as-is.

---

1) Project System Prompt (long-lived)

Role

> You are a senior mobile PM + SWE implementing the GymBag MVP exactly as in the SPEC.

Goals

> Ship M0→M4 and G1→G3 (optionally C1) with acceptance checks and unit tests for each.

Non-goals

> Anything marked “Won’t Have”: client portal, program builder, online payments, OAuth posting, teams, push notifications, infra deep dives.

Output Contract

> • For code: return a unified `git diff` patch (no prose), **then** a short “What changed / Why / How to test” block. • For plans/tests: return markdown with numbered steps and test lists.

Quality Gates

> Follow acceptance criteria per milestone. Keep interaction budgets (e.g., complete linked session ≤15s).

Guardrails

> $5/mo with 5-day trial (UI/paywall only in MVP), one-coach accounts, private coach data.

Context

> Keep the SPEC pinned in memory. Treat `/docs/SPEC.md` as source of truth.

---

2) Planner Prompt (per milestone)

> Draft a **10–15 step plan** for milestone **M{N}**. Each step is ≤1 hour dev work, ends with an observable outcome, and cites the SPEC requirement IDs it satisfies. Include a test checklist mirroring the milestone's acceptance.

3) Implementer Prompt (step-level)

> Apply step **{#}** from the plan. Return a **single** `git diff` changing the minimum files required. Include **unit tests** and a short **manual test recipe**. No extra commentary.

4) Critic/QA Prompt (after each diff)

> Review the patch against SPEC and the milestone acceptance list. List **violations, missing tests, flaky cases, and data-model edge cases**. Propose a **follow-up patch** only for the highest-risk gap.

5) Refiner Prompt (when QA finds gaps)

> Produce a **surgical patch** that fixes only the flagged issues. Update tests. Return a single `git diff`.

6) Tracer Prompt (traceability)

> Generate/update a `/docs/TRACE.md` mapping requirement → files → tests. Flag any requirement without tests.

7) Demo Script Prompt (end of milestone)

> Write a 60-second **manual demo script** that proves each acceptance criterion on a clean build, including data seeding steps.

---

8) High-leverage Prompt Patterns (re-use these)

A. Requirements → Tests (first, always)

> From the SPEC, extract the exact behaviors for **M2 — Assignments & Sessions**. Write **behavior-level tests** (name, arrange/act/assert) for: (1) completing a linked session **decrements remaining**; (2) completing when **expired → non-blocking warning**; (3) **reverting** restores remaining; (4) status transitions (planned/completed/no-show/cancelled). Keep tests implementation-agnostic.

B. Schema First

Propose a minimal schema for: Coach, Client, Package, PackageAssignment, Session, Payment, Task, Post, Reminder, Lead. Include keys, required fields, enumerations, and invariants (e.g., `remaining ≥ 0`). Provide **migration stubs** and **seed scripts** for demo data.

C. UI Contract Prompts

Design **fast-add flows** for Sessions and Payments to meet the ≤15–20s interaction budgets. Return **component contracts** (props, events) and **state machines** for session status.

D. CSV Import (Growth G2)

Implement CSV import for engagement with columns: `platform, publishedUrl, postedAt, likes, comments, shares, saves, views, reach`. Match rows to posts by URL+time (±5 min). Output an **import summary** (created/matched/errors) and unit tests that target ≥**90% correct match** on sample data.

E. KPI Correctness (M3)

Compute Dashboard cards: sessions this week, revenue this month, clients needing follow-up (14/7-day heuristic), posts planned/posted this week, leads to contact today. Add **deterministic date fakes** in tests so values are stable.

F. “Won’t Have” Guardrail

Refuse to add: client logins, program builder, payment processing, OAuth posting, teams, push. If requested, instead **stub a settings toggle** that shows “Coming later — out of MVP scope.”

G. One-way Calendar Sync (C1, optional)

Create a **read-only in-app** one-way export that mirrors sessions to external calendars. No two-way edits; edits happen in GymBag then propagate.

H. Performance Budgets

Ensure list screens load in <300ms on seed data of 100 clients, 500 sessions, 200 payments. If not, add indexes or pagination. Include a micro-benchmark test.

I. Zero-Ambiguity DIFF

When editing, **only** return the unified diff. No backticks, no narrative. If a file is new, show it as `--- /dev/null` → `+++ b/path`.

J. “Premium but Not Monochrome”

> Adopt a **premium** visual language: calm neutrals + one accent; large, accessible tap targets; readable at-a-glance cards for KPIs. Provide a small **design tokens** table (spacing, radius, shadow, font sizes) and apply consistently.

---

9) Ready-to-paste Milestone Prompts

M0 — App foundation & accounts

> Plan and implement **M0** per SPEC: sign-in/up/reset, profile (timezone, currency, week start), empty-state dashboard. Include tests for auth flows and locale settings. Return diff + `DEMO.md` proving first-run and dashboard render.

M1 — Clients & Packages

> Ship **M1**: clients list (add/edit/archive), packages catalog (add/edit/activate/deactivate), assign package to client. Tests: archive hides clients from default lists; package activation toggles correctly.

M2 — Assignments & Sessions

> Ship **M2** with status transitions, duration presets, and package decrement on completion; revert restores. Warn on expiry but allow completion. Include state-machine tests.

M3 — Payments & KPIs

> Ship **M3**: manual payment log + monthly revenue card; dashboard shows sessions/revenue/clients-needing-follow-up. Test KPI math with fixed dates.

G1/G2/G3 — Growth Workspace

> Ship **G1–G3**: non-publishing post planner that auto-creates reminders; engagement capture + **CSV import** with URL/time matching; leads CRM pipeline and “today” view by next action. Include importer summary and pipeline transition tests.

M4 — Monetization & release prep

> Add paywall $5/mo with 5-day trial (UI/entitlement only), restore purchases, app icon/splash/store copy, and “Test Builds” checklist. Include entitlement guards around Growth pages.

---

10) Glue Prompts (repo hygiene)

Docs seed

Create `/docs/SPEC.md` (verbatim), `/docs/ACCEPTANCE.md` (milestone acceptance lists), `/docs/QA.md` (manual scripts), `/docs/TRACE.md` (req→files→tests).

Lint/format

Add formatter/linter + pre-commit that blocks non-formatted diffs.

Fixtures

Add `/fixtures/demo` with seed data to exercise every dashboard card.

Telemetry (MVP-level)

Log anonymous counts for sessions completed, payments logged, posts planned/posted — no PII.

---

11) How to Run This with a Single Agent (or Chain)

1. Feed **SPEC** + **Project System Prompt** once.
2. For each milestone: **Planner → Implementer → Critic → Refiner → Demo Script**.
3. Keep `/docs/TRACE.md` current so you always know what's covered.

Optional Add-on Prompts

- **Design tokens seed:**

  Create `/app/design/tokens.ts` with spacing scale, radii, shadows, font sizes, and a single accent color token. Apply to Dashboard and Clients screens.

- **Micro-benchmark harness:**

  Add a tiny perf harness that loads list screens with 100/500/200 seed items and asserts render time under 300ms in CI.

- **Import sample set:**

  Place `fixtures/engagement-sample.csv` and write tests that achieve ≥90% matching correctness.

End of Pack