Here’s your **opening prompt** to paste into a brand-new Replit Agent chat:

Goal: Initialize a MOBILE-ONLY PWA scaffold for “SW Monthly Golf” with my exact preferences captured in a replit.md so future requests stay cheap and consistent.

Stack:

- Next.js (App Router) + TypeScript

- Tailwind + shadcn/ui (mobile-first)

- Prisma + SQLite (DB; just init in this step)

- Dexie (IndexedDB) for offline scoring queue (stub in this step)

- next-pwa (manifest + service worker)

Important build rules (apply now and record in replit.md):

- 18 holes only. All players on the same tee box.

- Input Handicap Index (HI) ONLY; Course Handicap (CH) is computed from Course Rating & Slope later.

- Net allowance defaults to 100% (organizer can adjust per event).

- Rounding: standard nearest (0.5 rounds up).

- Cap CH at 18 (enforced later).

- Games we will support in later steps: Gross Total, Net Total, Gross Skins (NO carry).

- Ties for Gross/Net resolved by USGA last 9/6/3/1 (implemented later).

- Groups are MANUAL ONLY with per-group tee times.

- Anyone with passcode can score; organizer can edit any score.

- Offline conflicts will be last-write-wins with an admin override screen (later).

- Public read-only share page (later). No exports.

Deliverables for THIS request (ONE checkpoint):

1) Project scaffold:

- Next.js App Router + TS configured

- Tailwind + shadcn/ui installed; create <MobileLayout/> with safe-area insets and bottom nav tabs:

Home, Players, Courses, Tournaments, Groups, Score, Leaderboards

- Basic mobile theme (brand green #16a34a) and dark mode: auto

2) PWA:

- next-pwa configured; web app manifest for “SW Monthly Golf” (name/short\_name/icons/theme\_color/background\_color)

- Service worker and offline fallback page; app is installable on iOS/Android

3) State & data stubs:

- Initialize Prisma + SQLite (no full schema yet)

- Set up Dexie with an “offline scoring queue” stub and a background sync placeholder (no API yet)

4) Auth/passcode stub:

- Add a simple “Join Tournament” passcode screen and store passcode in session/local state (no server validation yet)

- Place a small “You are organizer?” toggle in dev settings for later admin-only screens (stub)

5) DX & tests:

- Vitest + React Testing Library configured; include one example component test

- ESLint + Prettier strict

6) Docs:

- Create README with setup/run/test steps

- Create \*\*replit.md\*\* with the exact content below so the Agent follows it next time

replit.md content to write verbatim:

---

# Replit Agent Build Preferences — SW Monthly Golf (mobile PWA)

## Tech & Platform

- Next.js (App Router) + TypeScript

- Tailwind + shadcn/ui (mobile-first)

- Prisma + SQLite (server DB)

- Dexie (IndexedDB) for offline scoring; background sync on reconnect

- PWA via next-pwa (manifest + service worker)

## Functional Rules (to follow in subsequent tasks)

- Holes: 18 only. Single tee box for all.

- Roster & Courses: manually added; persist for reuse across events.

- Handicap: HI only. CH computed as: round(HI \* (Slope/113) + (Rating - Par)), then cap at 18.

- Net allowance: default 100%; adjustable per event.

- Rounding: nearest (0.5 up).

- Games: Gross Total, Net Total (Net = Gross − Playing CH), Gross Skins (lowest UNIQUE gross; NO carry). Show pushes like “Hole 4: push (2 at 3)”.

- Ties (Gross/Net): USGA last 9/6/3/1.

- Groups: Manual only with per-group tee times.

- Scoring: Anyone with passcode can score. Organizer may edit any score.

- Offline conflicts: last-write-wins with an admin review override (later).

- Sharing: Public read-only link; show HI/CH publicly.

## Non-Functional & DX

- Tests: Vitest + RTL. Prioritize scoring math, tiebreak, skins, and group/tee-time flows in later steps.

- Implement ONE cohesive feature per request and create EXACTLY ONE checkpoint.

- Keep diffs minimal. No unused deps. Mobile-only layouts.

- Do NOT enable Extended/High-Power/DI unless explicitly asked.

## Deliverable Expectations

- Update README with clear run/test steps.

- Include small seed placeholders to be expanded in later tasks.

---

Acceptance (must pass before creating the single checkpoint):

- `pnpm dev` runs without errors.

- PWA is installable; offline fallback reachable.

- Bottom tabs render with <MobileLayout/> and safe-area handling.

- Dexie queue + background sync placeholders compile (no network yet).

- Prisma & SQLite are initialized (no full schema yet).

- Lint and example tests pass.

- EXACTLY ONE checkpoint created. Do NOT enable Extended/High-Power/DI for this request.

If any assumption above is unclear, choose sensible defaults but still deliver the scaffold in this single checkpoint.