Goal: Implement Prisma schema + admin CRUD for Players, Courses, Tournaments, Groups, and Entries, with HI→CH math per my rules. No scoring logic yet.

Read and follow replit.md. One cohesive feature → ONE checkpoint.

Models (Prisma):

- Player { id, name (req), email?, handicapIndex Float? } // roster saved for reuse

- Course { id, name (req), par Int (req), slope Int (req), rating Float (req) } // saved for reuse

- Tournament {

id, name (req), date (req), courseId (FK),

holes Int @default(18),

netAllowance Int @default(100), // adjustable per event

passcode String, // simple for now; server validation later

potAmount Int?,

participantsForSkins Int?,

skinsCarry Boolean @default(false) // MUST remain false (no carry)

}

- Group { id, tournamentId (FK), name (req), teeTime DateTime? } // manual groups

- Entry {

id, tournamentId (FK), playerId (FK),

courseHandicap Int, // CH after rounding + cap

playingCH Int, // after netAllowance + rounding

groupId String?

}

// NOTE: HoleScore will be added next step with scoring—do not add now.

Rules to implement:

1) Handicap entry mode = \*\*HI only\*\*. No manual CH entry anywhere.

2) When adding a Player to a Tournament (creating an Entry):

- Pull Course(par, slope, rating) from the Tournament’s Course.

- Compute CH = round( HI \* (slope/113) + (rating - par) ).

- Apply rounding = \*\*nearest (0.5 rounds up)\*\*.

- Cap CH at \*\*18\*\*.

- playingCH = round( CH \* netAllowance/100 ) using the same rounding rule.

- Persist CH and playingCH on Entry.

3) Tournament settings UI:

- Create /tournaments (list + create).

- Create/edit Tournament: choose Course (dropdown), set date/name, view/edit netAllowance (default 100), set passcode, optional pot fields.

- Show Entries table with Player name, HI, CH, playingCH. Add/remove Players from roster.

4) Roster & Course admin:

- /players: list/create/edit/delete Players (name, optional email, HI as number).

- /courses: list/create/edit/delete Courses (name, par, slope, rating).

- These are global (persist across events).

5) Groups admin (manual only):

- /tournaments/[id]/groups: create any number of Groups, set name and optional teeTime.

- Assign/unassign Entries to Groups with simple controls (drag-drop or dropdown). Persist immediately.

6) Passcode (stub):

- Ensure Tournament has a passcode field stored. Keep the existing join screen as-is; server validation will be added later.

UX constraints:

- Mobile-first pages (use the existing bottom tabs: Players, Courses, Tournaments, Groups).

- Keep diffs tight; reuse existing layout/components.

Tests (Vitest):

- HI→CH: cover typical cases + rounding at .5 and negative (rating < par) offsets.

- CH cap: values above 18 are clamped to 18.

- playingCH: respects netAllowance (100 and a different value like 85).

- Basic CRUD: creating Course, Player, Tournament, and adding Entry computes CH/playingCH as expected.

Acceptance (must pass before creating the single checkpoint):

- Prisma migrations succeed; seed minimal demo data: 6 Players with HI values, 2 Courses with realistic par/slope/rating.

- From the UI, I can create a Tournament, pick a Course, add Players → Entries appear with correct CH (capped) and playingCH.

- Manual Groups page exists with group create/rename, teeTime edit, and assigning Entries.

- Lint/tests pass. README updated with schema/migrate/seed instructions.

- EXACTLY ONE checkpoint. Do NOT enable Extended/High-Power/DI.