Build exactly to this spec (Humans Square — World Mini App)
Use the attached PDF "Project: Humans Square — a World Mini App" as the single source of truth. Build a minimal, fast, World-native MVP that opens inside World App via MiniKit.

0) Scaffold & dependencies

- Create a **Next.js 15 (App Router) + TypeScript** project using World's mini-app template.
- Install **@worldcoin/minikit-js** and **@worldcoin/minikit-react** and wrap the app in `MiniKitProvider`.
- Add `.env.example` with:
    - `NEXT_PUBLIC_MINIKIT_APP_ID=` (placeholder)
    - `NEXT_PUBLIC_APP_URL=` (http URL Replit will host)
    - `REALTIME_URL=` (WS/SSE endpoint you create)
- Add a **README**: local dev, env vars, how to open inside World App (so `MiniKit.isInstalled()` returns true).

1) Information architecture (routes)

- `/` **Landing**
    - Hero "Humans Square."
    - Button: **Enter Global Square**
    - **Read-only preview**: latest 10 messages from Global (works pre-verify).
    - Attempting to post/like opens **World ID verify** gate.
- `/room/global` **Global Square**
    - Header shows **Today's topic** (server string; default "What are you building today?").
    - Live **chronological feed** with new messages appended in real time.
    - Message cells: pseudonymous handle, timestamp, ⭐ count.
    - Composer (240 char soft limit, counter). Disabled until verified.
    - Actions per message: ⭐ **upvote** (1 per human), **Report**, **Mute** (local hide).
    - Presence chip: "N humans online" (rounded).
- `/room/work` **Work Mode** (toggle from Global)
    - Separate feed for short **Help Wanted / Advice / Collab** posts.
    - Fields: **Category, Link (optional), Location (Global/Country/City)**.
    - **Link rate-limit**: ≤ 2 per 10 min and ≤ 4 per hour per human.
    - One-tap back to Global.
- `/capsule` **Capsule onboarding** (3 skippable steps)
    - "You're entering a real-human space."
    - "Be kind. No spam. No NSFW. No scams."
    - "Post as a human: verify once with World ID."
    - Ends on **Verify & Continue** (runs MiniKit flow). Persist a "seen" flag.

- **Profile lightbox (modal)**
    - First-seen date, total posts, stars received.
    - **Request Connect** button (just records consent + stub; no DMs in MVP).

2) Data model (no PII; use hashed nullifier)

Use **nullifier hash** (or salted hash) from World ID as `human_id`. Minimal DAL so we can swap storage later.

- Human: `human_id`, `joined_at`, `capsule_seen`, `mute_list[]`
- Message: `message_id`, `room`(global|work), `author_human_id`, `text`, `created_at`, `stars_count`, `reports_count`, (work-only: `category`, `link`, `geo_scope`)
- Star: `message_id`, `human_id`, `created_at`
- RateLimit: rolling counters per `human_id` per room/action
- Theme: `date`, `topic_text`

Storage: Start with Replit KV/SQLite. Keep a thin `DataAccess` layer.

3) Realtime

- Live updates over **WebSocket or SSE** (your choice).
- **Fallback to polling** (e.g., 2s) if the socket drops so demo never stalls.

4) World ID + MiniKit (must-have)

- Gate **post / star / report / connect** behind MiniKit World ID verification.
- Detect World App using `MiniKit.isInstalled()` and show **Open in World App** hint otherwise.
- Never store wallet or PII; only the hashed nullifier.

5) Wallet rail (stub only; no real transfers)

- Implement a `PayoutsService` interface + **points** (not tokens) demo called **"Room Rain"** that splits a small **points** pool among recently active, upvoted humans; show a transparent `/ledger` list of "point rains."
- Add README note: "Real ERC-20 transfers would later use Permit2 and sponsored gas; **not** included in MVP."

6) Safety, moderation, and limits

- **Content filter**: simple bad-word list client-side + server sanity checks (length, URL whitelist).
- **Report** flow: threshold hides from default view; keep a simple admin flag.

- **Mute** list per user (local hide).
- **Rate limits (tunable constants)** per verified human:
    - Messages: **5/min, 60/hr, 200/day**
    - Stars: 1 per message (also cap ~20/min)
    - Work-mode links: **≤2/10min, ≤4/hr**
- Friendly cooldown toasts; grey out composer during cooldown.

7) Copy & empty states (use exactly)

- Landing: “A global square for verified humans.”
  Sub: “Talk. Learn. Build. One human, one voice.”
- Composer placeholder: “Say something useful, kind, or curious…”
- Verify gate: “Posting is human-only. Verify once with World ID.”
- Empty Global: “Be the first to say hi today 👋”
- Empty Work: “No requests yet. Ask for help or offer your skills.”
- Report confirm: “Thanks. We’ll review and keep this space healthy.”
- Cooldown: “You’re sending messages fast. Take a breath—back in {N} sec.”

8) Design requirements

- Follow **World Design Guidelines** defaults: **24px spacing grid**, safe-area padding, skeleton loaders, large tap targets, readable contrast; keep it **light and fast**.
- No custom fonts needed.

9) Telemetry & health

- Simple counters (console or tiny endpoint): `verify_success`, `message_sent`, `star_given`, `report_submitted`.
- `/health` endpoint for uptime.

10) Definition of Done (acceptance checks)

- Opens inside World App; `MiniKit.isInstalled()` is true.
- Landing shows read-only preview (no verify needed).
- **Verify** → **post** → **live update** works E2E.
- ⭐ upvote works; can’t double-star same message.
- Rate limits & link limits enforced with friendly cooldowns.
- Report & Mute work.
- Work Mode shows categorized posts.
- Handles poor network (skeletons + retry).
- Stores **only hashed nullifier**; **no PII**.
- UI spacing/states consistent with World guidelines.
- README explains local run + “open in World App” steps.

11) Non-goals / constraints for MVP

- No DMs, no notifications beyond functional, **no NSFW or adult services**, no on-chain payouts, no KYC.

Please implement the above end-to-end, keep code clean and commented, and produce a short screencap/GIF in the README showing: open → verify → post → star → report → Work Mode post with link → cooldown trigger.

“Follow the written spec; do not add features outside scope.”