

Project: Humans Square — a World Mini App

Tagline: A bot-proof global square for verified humans to talk, learn, and build.

Why World-native: Uses **Distribution (World App)** + **Proof-of-Human (World ID)** + **Wallet rails** (future “Room Rain”).

MVP focus: One lively, always-on, human-only chat with light structure and upvotes; plus a toggleable **Work Mode** feed for short job/help offers. Points only (no real token transfers) to stay compliant for the hackathon.

0) Bootstrap & Guardrails

- **Scaffold** using World’s official template:
 - Use the command recommended in the docs: *Next 15 app router + TypeScript + MiniKit React*.
 - Include **@worldcoin/minikit-js** + **@worldcoin/minikit-react**.
 - **Files & config to produce:**
 - **.env.example** with: **NEXT_PUBLIC_MINIKIT_APP_ID**, **NEXT_PUBLIC_APP_URL**, any realtime URL.
 - A **README** that explains local dev, env vars, and how to open inside World App (so `MiniKit.isInstalled()` returns true).
 - **Design rules:** Follow World **Design Guidelines** defaults:
 - 24px base spacing grid, clear empty states, loading/skeleton states, safe-area padding, big tap targets.
 - Light, fast, legible. No custom fonts needed.
 - **Review safety:** No NSFW, no crypto payouts, no promos via notifications. Functional notifications only (if any). Handle low bandwidth gracefully.
-

1) Information Architecture (pages & routes)

1. **Landing (/)**
 - Hero: “Humans Square”
 - Button: **Enter Global Square**
 - Read-only preview of the live feed (latest 10 messages) even if not verified.
 - CTA to verify with World ID when user tries to post/like/DM.
2. **Global Square (/room/global)**
 - **Header:** “Global Square” + Today’s topic (server-configurable; default: “What are you building today?”).
 - **Message list:**
 1. Chronological, auto-updating; new messages appear live.

2. Each message shows **pseudonymous handle**, timestamp, and ★ **upvote** count.
 - **Composer:**
 1. Disabled until **World ID** verification succeeds.
 2. Soft character limit (e.g., 240 chars). Show counter.
 - **Actions:** ★ upvote (1 per user per message), **Report** (abuse), **Mute** (hide a user locally).
 - **Rate limits:** Per verified human: up to **5 messages/min, 60/hour, 200/day** (tuneable constants).
 - **Presence chip:** “N humans online” (rounded; don’t reveal exact counts if that’s simpler).
 3. **Work Mode (/room/work)**
 - Toggle on the Global Square: **Work Mode**.
 - A separate feed for **short job offers / requests / help** (strictly SFW, no adult content, no financial promises).
 - Each post has **Category** (Help Wanted, Advice, Collab), **Link (optional)**, and **Location** (Global / Country / City).
 - Rate-limit link posts: at most **2 links per 10 minutes** and **4 per hour** per human.
 - Switch back to Global with one tap.
 4. **Capsule Onboarding (/capsule)**
 - A 3-step, skippable intro:
 1. “You’re entering a real-human space.”
 2. “Be kind. No spam. No NSFW. No scams.”
 3. “Post as a human: verify once with World ID.”
 - End with **Verify & Continue** (triggers MiniKit World ID flow).
 - Store a flag so we don’t show again.
 5. **Profile Lightbox (no page)**
 - Tap a handle to open a lightweight modal: first-seen date, total posts, stars received.
 - Button: **Request Connect** → asks consent to open a future DM (stub only; no actual chat required in MVP).
-

2) Data Model (simple, hackathon-friendly)

Do NOT store PII. Use the **per-app poll-scoped nullifier** (or app-scoped nullifier) as the core identity key.

- **Human**
 - **human_id** (hashed nullifier)
 - **joined_at, capsule_seen** (bool), **mute_list** (array of human_ids)
- **Message**
 - **message_id, room** (“global” or “work”)

- `author_human_id`, `text`, `created_at`
 - `stars_count` (int), `reports_count` (int)
 - `category`, `link`, `geo_scope` (work mode only)
- **Star**
 - `message_id`, `human_id`, `created_at`
- **RateLimit**
 - rolling counters per `human_id` per room
- **Theme**
 - `date`, `topic_text` (today's "prompt of the day")

Storage: For the hack, you can start with an in-memory store plus a simple persistent KV (e.g., a Replit DB or SQLite). Clearly separate a thin **DataAccess** layer so we can swap to Supabase/Neon later.

3) Realtime

- Implement realtime **message streaming** so posts and stars appear live (WebSocket or SSE).
 - Fall back to periodic polling if sockets fail (e.g., 2s interval) so the demo never stalls.
-

4) World ID & MiniKit Integration (must-have)

- Gated actions (post/star/report/connect) require **World ID verification** via MiniKit.
 - Use `MiniKit.isInstalled()` to detect World App environment; show **Open in World App** hint otherwise.
 - Save only the **nullifier hash** (or a salted hash) as `human_id`. Never store the user's wallet or PII.
 - If verification fails, show a clean retry path and an explanatory empty state.
-

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

Prepare a "**Room Rain**" placeholder for later (not active in MVP).

- A `PayoutsService` interface with methods like `previewSplit(activeHumans)` and `distributePoints()`.
- For now, award **points** (not tokens) to recently active, upvoted participants in a demo command.

- Add a **transparent ledger page** (</ledger>) that lists recent “point rains” (demo data only).

Note: **Permit2** will be required for real ERC-20 transfers later. Create a comment/todo in the README explaining that the MVP intentionally ships **without** any on-chain payout.

6) Safety & Moderation

- **Content filter:** client-side bad-word list (light), server-side sanity checks (length, URL whitelist).
 - **Report flow:** any message can be reported; if reports exceed a threshold or admin flag is set, hide from default view.
 - **Mutes:** local user can mute another human; the UI hides their content.
 - **Audit log** (server console is okay for the hack) for posts/flags/mutes.
-

7) Rate Limits & Abuse Controls

- Enforce per-human (nullifier-based) limits:
 - **Messages:** 5/min, 60/hr, 200/day.
 - **Links (Work Mode):** max 2 per 10 minutes & 4 per hour.
 - **Stars:** 1 per message; reasonable per-minute cap to avoid spam (e.g., 20/min).
 - On violations, show friendly toasts and grey out the composer for a short cooldown.
-

8) UX copy & empty states (use these strings)

- Landing headline: “**A global square for verified humans.**”
 - Subtext: “**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.**”
-

9) Accessibility & Performance

- Prefer semantic HTML, readable contrast, large tap targets.
 - Skeleton loaders (not spinners) for lists.
 - Optimistic UI for posts and stars; reconcile on server ack.
 - Measure TTFMP under low network; keep bundle lean.
-

10) Telemetry (lightweight)

- Console metric counters (or a tiny endpoint) for: `verify_success`, `message_sent`, `star_given`, `report_submitted`.
 - A `/health` endpoint for uptime checks.
-

11) Feature Flags & Env

- `NEXT_PUBLIC_MINIKIT_APP_ID` (required)
 - `NEXT_PUBLIC_APP_URL` (for deep links)
 - `RATE_LIMITS_*` constants (optional)
 - Document how to run **inside World App** for the demo (important).
-

12) Definition of Done (acceptance checks)

- Opens in World App; `MiniKit.isInstalled()` returns true.
 - **Read-only preview** works without verification.
 - **Verify** → **post** → **see it live** works end-to-end.
 - **Upvote** works; can't double-star the same message.
 - **Rate limits** trigger friendly cooldowns.
 - **Work Mode** shows categorized posts; link limits enforced.
 - **Report** and **Mute** flows work.
 - Handles poor network (skeletons, retry).
 - No PII stored; only hashed nullifier for `human_id`.
 - All UI uses design spacing/states consistent with World guidelines.
-

13) Stretch (time-boxed, optional)

- **Topic of the Day**: server-controlled string that rotates daily.
- **Local language hint**: detect device language and surface a “join {Language} channel” suggestion (non-blocking).

- **Invite deep link:** “Bring a friend to Global Square” (plain URL share for now).

Please implement the above, produce a clear README with run instructions, and keep the UI minimal, fast, and World-native. No code samples are needed from me—build directly from this spec.