LCM — Life Credits Manna

A simple, community-verified credit system that turns everyday actions into trackable progress.

Table of Contents

What is LCM?

LCM (Life Credits Manna) is a lightweight, browser-based app that lets people log everyday tasks, get those tasks verified by their communities, and earn credits. It is designed to be child-friendly and community-first, with a clean dashboard, simple flows, and guardrails to discourage abuse.

LCM is 100% front-end (static files) connected directly to **Supabase** for storage, auth, and Row-Level Security (RLS). That means you can deploy it for free (Netlify Drop / GitHub Pages) and keep your data secured by database policies.

Core Concepts

- **Communities**: HOME, SCHOOL, CHURCH, WORK, TEAM (configurable). Each community can have verifiers.
- **Verifiers**: People in your community who confirm a task really happened. Some actions require a threshold of verifiers.
- **Tasks**: User-logged actions with a category, description, and selected verifiers.
- **Credits**:
- **Normal Credits**: Earned when tasks get verified (daily caps apply).
- **Bonus Credits**: Extra rewards for exceptional verification or effort (tracked separately; redeemable later).

- **Super Credits**: Milestone credit earned when a monthly target is met (up to 12 per year).
- **Policing Credits**: Earned for valid incident reporting that helps maintain integrity; redeemable later.
- **Policing & Redemption**: Community incident reporting, fair penalties, resolution logging, and a clear redemption path.

  Bonus logic (current design): - Verification-based bonus: tasks with **>10 verifiers** earn **0.1** per extra verifier (capped at **1** bonus credit per task). - Extra-effort bonus: more than **10 verified tasks** in a day → **0.1** bonus per extra task. - Bonus stored separately; **redemption** at **1 bonus = 0.5 normal credits** (configurable). - **Daily caps** are enforced by server rules so the system can't be gamed.

Key Features (MVP)

- **Tasks page**: Log a task, pick category, select verifiers, submit.
- **Verification flow**: Verifiers confirm/deny; threshold rules determine crediting.
- **Credits page**: See Normal, Bonus, Super, and Policing credits with clear separation and redemption controls.
- **Notifications**: Status updates when tasks are verified, bonus is awarded, incidents are resolved, etc.
- **Policing & Redemption**: Report misuse; single-incident grouping and fair penalties; first valid reporter earns a policing credit; redemption converts policing credits into normal credits at a defined rate.
- **Dashboard**: A simple, dark-theme layout (purple accents) with profile, communities, tables for daily/monthly/yearly stats.

Tech Stack

- **Frontend**: HTML, CSS, vanilla JavaScript (no build step required)
- **Backend (as-a-service)**: Supabase (Postgres + Auth + RLS)
- **Hosting**: Any static host (Netlify Drop, GitHub Pages, Cloudflare Pages) — all free tiers work

---

Architecture

```
Browser (static site: index.html, *.html, styles.css, *.js)
      ↓ (Supabase JS v2 via CDN, anon key only)
Supabase (Postgres + Auth + RLS + Functions/Triggers)
```

- **No servers to run.** The browser talks directly to Supabase using the public **anon key**. - **Security is in RLS policies**. Every read/write is permitted or denied by the database rules you set.

Getting Started (No-Code / Free)

1. **Clone or download** this repo.
2. **Add your Supabase credentials** in one of two ways:
3. **Inline (in HTML)**

```
<script src="https://cdn.jsdelivr.net/npm/@supabase/supabase-js@2"></
script>
<script>
  const SUPABASE_URL = "https://YOUR-PROJECT.supabase.co";
  const SUPABASE_ANON_KEY = "YOUR-ANON-KEY";
  window.supabaseClient = window.supabase.createClient(SUPABASE_URL,
SUPABASE_ANON_KEY);
</script>
```

4. **Or in** `supabaseClient.js` (make sure it loads **after** the CDN line above):

```
// supabaseClient.js
const SUPABASE_URL = "https://YOUR-PROJECT.supabase.co";
const SUPABASE_ANON_KEY = "YOUR-ANON-KEY";
window.supabaseClient = window.supabase.createClient(SUPABASE_URL,
SUPABASE_ANON_KEY);
```

5. **Open** `index.html` in your browser to preview locally.
6. **Deploy for free** (see next section) and add your live URL in Supabase settings.

If you see **createClient is not defined**, ensure the **CDN** `<script>` line comes **before** your app scripts and that you call `window.supabase.createClient(...)`.

Local Preview (Optional)

- Double-click `index.html` to open it locally, or
- Use VS Code's **Live Server** extension for auto-refresh.

Deployment (Free Hosting)

Option A — Netlify Drop (fastest)

1. Zip the project folder or keep it as a folder.
2. Go to **Netlify Drop** (web search it) and drag the folder/zip in.
3. You'll get a live `https://*.netlify.app` link immediately.

Option B — GitHub Pages (durable)

1. Create a public repo and upload the files with `index.html` **at repo root**.
2. **Settings → Pages → Build and deployment**: Source = **Deploy from a branch**; Branch = `main` (root).
3. Wait ~1 minute; you'll get `https://YOUR-USERNAME.github.io/REPO`.

   After deployment: add your live URL in **Supabase → Auth → URL Configuration** (and CORS allowlist, if present). Hard-refresh the site.

Supabase Configuration

- **Auth → URL Configuration**
- **Site URL**: your live URL (Netlify/GitHub Pages/Cloudflare)
- **Additional Redirect URLs**: add the same link and any other public pages that handle email/login links (if using auth)
- **Project Settings → API / CORS**
- If a **CORS allowlist** is shown, add your live URL(s)
- **RLS (Row-Level Security)**
- Keep RLS **enabled** on all tables. Policies define who can read/write what.

Security & Data

- Only the **anon key** is exposed in the frontend. This is expected.
- **NEVER** put your service role key in the frontend.
- RLS and DB triggers/constraints implement the rules: verification thresholds, credit caps, penalties, and redemption.

Roadmap

- ✅Task logging & verification threshold rules
- ✅Credits page (Normal, Bonus, Super, Policing) with redemption
- ✅Notifications (task/bonus/incident updates)
- ✅Policing & redemption flow (incident grouping, penalties, reset window, resolution logging)
- 🔆Invite flows for verifiers and community management UX
- 🔆Admin dashboards (moderation, audits, exports)
- 🔆Accessibility pass & mobile polish
- 🔆Multi-language support

Troubleshooting

- `createClient is not defined`: Load the Supabase v2 CDN **before** your scripts and use `window.supabase.createClient(...)`.
- **CORS errors**: Add your live URL in Supabase **Auth → URL Configuration** and any **CORS allowlist**. Then hard-refresh.
- **Blank page on GitHub Pages**: Ensure `index.html` is at the **repo root** (not nested in a folder).
- **404** `favicon.ico`: Add a small icon or ignore; it's harmless.
- **Mixed content**: All calls must be **HTTPS**.

License

MIT — use, modify, and share freely with attribution.

USER MANUAL — LCM (Life Credits Manna)

A quick guide for everyday users. No technical background required.

1) Getting Access

- Open the LCM link shared by your coordinator (it looks like `https://something.netlify.app` or `https://username.github.io/lcm`).
- If Sign-In is enabled, create an account or sign in. Otherwise, you may see the dashboard directly.

2) The Dashboard at a Glance

- **Top Bar**: "LCM DASHBOARD" centered at the top.
- **Profile**: Your name/email. (This section may have a grey strip across the page.)
- **My Communities**: HOME, SCHOOL, CHURCH, WORK, TEAM displayed in a row. Under each, see how many verifiers you have. You may be able to view the list, and, if allowed, add/remove verifiers.
- **What are you up to today?**: A simple box to type your task.
- **Action Buttons**: Choose a **Category**, pick **Verifiers**, and **Submit** the task.
- **Tables**:
- **Tasks Logged Today**
- **Special Credits** (Bonus/Policing)
- **Daily / Monthly / Yearly Stats**

3) Logging a Task (Everyday Flow)

1. Go to **"What are you up to today?"** and type your task (e.g., *"Made my bed"*, *"Helped mom with dishes"*).
2. Click **Category** and choose where it fits (HOME, SCHOOL, CHURCH, WORK, TEAM).

3. Click **Verifiers** and select who should confirm (or pick **All** if available).
4. Click **Submit**.
5. You'll see a confirmation message and the task will appear in **Tasks Logged Today**.

4) How Verification Works

- Your selected verifiers are notified and can confirm/deny.
- Some tasks require a **threshold** (e.g., at least X verifiers) before credits are granted.
- The **status** of each task updates as verifications come in (e.g., Pending → Verified).

5) Earning Credits

- **Normal Credits**: Earned when verified tasks meet the rules. Daily caps may apply.
- **Bonus Credits**: Extra rewards for
- Tasks with **more than 10 verifiers** (0.1 per extra verifier, capped at 1 per task)
- **More than 10 verified tasks** in a day (0.1 per extra task) Bonus credits are **separate** from normal credits and can be redeemed later.
- **Super Credits**: Earned when you hit a monthly target (e.g., 30 credits). Max **12 per year**.
- **Policing Credits**: Earned for valid incident reporting that helps keep the system fair.

Note: Daily caps and thresholds are set by the administrators and enforced by the system.

6) Redeeming Credits

- Open the **Credits** page.
- Use the **Redeem** actions to convert:
- **Bonus → Normal** at **1 bonus = 0.5 normal**
- **Policing → Normal** at **1 policing = 0.5 normal**
- Your total updates instantly after redemption.

7) Policing & Resolution (Integrity)

- If you notice misuse, you can **Report an Incident**.
- Reports about the **same incident** are grouped together; penalties apply **once per incident** to the reported user.
- Generally, the **first valid reporter** earns a policing credit (subsequent reporters may act as verifiers).
- Reported users can follow a **resolution path** (apologies, restorative actions). Partial credit restoration is possible.
- Penalty multipliers can **reset** after a clean period (e.g., 3 months with no incidents).

8) Notifications

- You will receive updates when tasks are verified, bonuses are awarded, incidents are resolved, and more.
- Use the notifications page to review recent activity.

9) Tips for Success

- Log tasks **daily** and vary them—LCM rewards healthy diversity and consistency.
- Choose verifiers who actually see your work. Honest verification means faster credits.
- Watch your **daily caps**. Spreading tasks across days can be more effective.

10) Privacy & Safety

- Only your community and authorized verifiers can see and verify your tasks.
- Incident reporting is for integrity, not punishment. Be fair and truthful.

11) Getting Help

- If something looks wrong, refresh the page.
- Still stuck? Share the exact message shown on the screen with your coordinator (or the project maintainer) so it can be fixed fast.

Enjoy using LCM!

If you're contributing development changes, please open a pull request with a clear description and screenshots of what changed.