AI SYNC DAY Starter Kit Readme

> **The Official Starter Repo for the TechSapiens Vibe Coding Contest (Batumi, Dec 7)**

🏆 The Challenge: "The Serendipity Engine"

The Problem:
Communities are currently just lists of names in a Telegram chat or a Luma guest list. How do you know that the person standing 5 meters away from you has the exact healthcare product experience you need? You don't. We have the people, but we lack the graph.

The Task:
Build a "Community OS" (Operating System) prototype that turns this passive guest list into an active, searchable, high-value network.

Feature Ideas (Pick one or remix):

1. **The Agentic Search:** A RAG chatbot that answers *"Who here knows Rust and likes hiking?"*
2. **The Social Anxiety Traffic Light:** A real-time status app (Green = "Pitch me", Red = "Deep Work").
3. **The "Coffee Break" Roulette** An algorithm that pairs people during breaks based on *non-obvious* connections (e.g., "You both hate React" or "You both lived in Berlin").

💡 **Any product manager can come up with an infinite number of additional features. Your chances of winning the hackathon will be higher if you manage to come up with and implement an additional feature idea for a community tool that isn't on this list.**

The Constraint:
You need to finish the project by 12:00 Sunday. Doing this by hand is impossible. You must use AI Agents (Cursor, Windsurf, Replit, Lovable) to architect, code, and debug this fast enough to demo it on Sunday.

⚡ The "Zero-Config" Stack

We recommend this stack because it lets you focus on **logic**, not infrastructure. Do not waste time configuring Webpack.

- **Agentic AI IDE:** Cursor AI or Google Antigravity
- **Frontend:** Next.js 14 (App Router) + Tailwind CSS + shadcn/ui
- **Backend:** Supabase (Postgres, Auth, Realtime)
- **AI Memory:** `pgvector` is pre-enabled for RAG (Retrieval-Augmented Generation)
- **Deployment:** Vercel

We **recommend** this list for those who haven't tried vibe coding yet. The key is to ensure that the majority of the product code is generated, not hand-coded. If you're used to your own AI setup, go with it, but if you're new to the topic, use the one recommended above or check out the expanded lists of popular vibe coding AI tools:

- Zapier Blog Post
- Curated by us and generated by Gemini 3 😅 .

🚀 Quick Start (The Speed Run)

▲ Deploy

Step 1: The Instant Backend
Click the **"Deploy with Vercel"** button above. It will:

1. Clone this repo to your GitHub.
2. Spin up a free Supabase project for you.
3. Deploy the live site.

Step 2: The "Vibe Check"
Clone your new repo locally.

```
git clone [https://github.com/YOUR_USERNAME/techsapiens-vibe-starter.git]
(https://github.com/YOUR_USERNAME/techsapiens-vibe-starter.git)
cd techsapiens-vibe-starter
npm install
npm run dev
```

Step 3: The Data Injection

We have pre-loaded src/data/participants.json with a dataset of attendees (anonymized/sanitized).

- **Your First Task:** Ask your AI agent (Cursor/Windsurf/Replit) to: *"Write a script to seed the Supabase 'users' table with the data from participants.json."*

📜 The Rules of Engagement (How to Win)

This is a **Vibe Coding** contest. We don't care how fast you type. We care how well you **orchestrate**.

1. AI-First Architecture — 25%

If you are manually writing SQL queries, you are doing it wrong. Use your Agent (Cursor, Claude, Gemini) to generate the schema, the API routes, and the frontend components.

2. Features Completion — 25%

You should implement at least 1 of our pre-selected feature ideas. The implementation of any suitable feature that was not on the list will be a bonus for the jury's decision.

3. Mandatory Documentation (`VIBE_LOG.md`) — 40%

You must maintain a log in docs/VIBE_LOG.md. We want to see your prompts.

- *Bad:* "We built a search bar."
- *Good:* "We pasted `participants.json` into Claude context, asked it to design a vector schema, hit an error with RLS policies, and fixed it by prompting..."

💼 If you don't have a technical background, focus more on your flow, process organization, pipeline, and brainstorming logging. How did you find the necessary information for technical implementation? Don't be shy about sharing mistakes — the audience will learn how to avoid them. Basically, anything that can clearly demonstrate the effectiveness of your approach is appropriate.

The documentation's value lies in showing *how* you leveraged the agent to overcome technical hurdles, aligning with the focus on process organization and pipeline.

4. Scalability — 10%

Hard-coding "Nikita" into the homepage is a fail. Build a system that could work for *any* event. Scalability includes not only avoiding hard-coding but also how well the system uses the tech stack for future growth.

📝 License & "Slop" Policy

- **No Slop:** If your app looks generic and hallucinates features that don't work, you lose points.
- **Open Source:** The starter kit is MIT Licensed. What you build is yours, but we encourage open-sourcing it for the community.

📦 Submission method

You need to publish the result of your work as a public project on GitHub, Vercel deployed site, and send the links in this form.

Happy Vibe Coding. 🧹