

Hackathon Case Study: **AI-Powered Journaling Companion**

Introduction

For this hackathon, I chose to make an AI-Powered Journaling Companion. The app facilitates journaling habits by generating prompts based on previous entries and tracking patterns across entries. I've always advocated for mental health, so this project resonated with me and I believe that AI is an effective tool to encourage people to keep a journal.

Functionality

Users can generate prompts at random, from user input, or from a previous entry, giving them inspiration to write their entries. Each entry is automatically analyzed for emotions and encoded as vectors, making searching through past entries more intuitive. Semantic search allows users to easily filter and retrieve previous entries based on content or emotional context. Full CRUD operations are available, enabling users to create, read, update, and delete entries with changes being persisted to the backend database. I also implemented input validation to prevent prompt injection or other adversarial inputs.

Trade offs

Implementing a local LLM helps keep data private and ensures fast responses, but it also limits the model size. SQLite is simple and lightweight, which made it ideal for a hackathon prototype, though it may not scale well for extremely large datasets.

Tech Stack

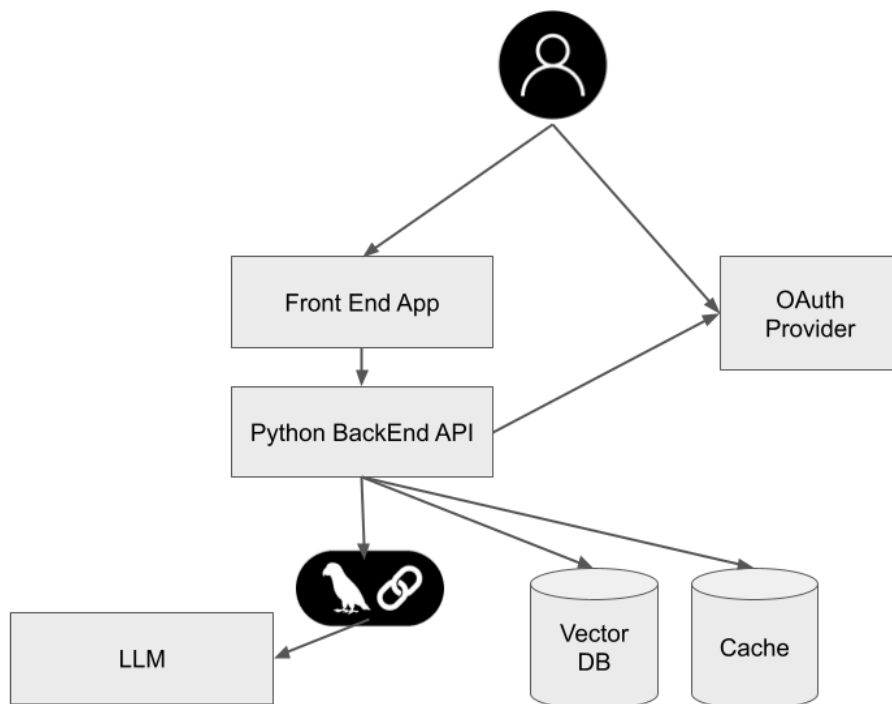
The backend is built with Python and SQLite, with `sqlite-vec` enabling semantic search on journal entries. I chose FastAPI for its simplicity and speed in serving backend requests. The AI text generation runs on a local LLM (Ollama, Phi4-mini), which preserves privacy while providing high-quality text. Emotion classification is handled by the `bhadresh-savani/distilbert-base-uncased-emotion` model, automatically detecting the user's emotions. Text is encoded for semantic search using SentenceTransformer (all-MiniLM-L6-v2) embeddings. On the frontend, the app currently uses HTML, CSS, and JavaScript, with plans to transition to React for a polished web experience and React Native for cross-platform mobile support.

Key Learnings

I learned that user experience is critical, as even small delays or inconsistent interfaces can disrupt engagement. Working with generative AI also highlighted the risks of biased or inappropriate outputs if safeguards are not in place. Security is equally important, as users could manipulate prompts to bypass system instructions or access sensitive information.

Ideas for the Future

In the future, I plan to track mood trends over time using visual graphs and summarize key emotional patterns from past entries. Conducting bias audits across different demographics and use cases will also be important. Additional features could include templates for 1, 3, and 5-minute entries, voice-to-text journaling, alerts for recurring themes like stress or social anxiety, and guided prompts tailored to specific intents such as stress relief or self-reflection. Users could select their mood before and after journaling using emojis or words, receive motivational quotes or affirmations based on their emotional state, get gentle reminders if they haven't journaled in a while, and see summaries of entries that highlight emotional patterns or triggers.



This diagram displays my original design that I modified to fit the scope of this hackathon.