Design Documentation

1. Introduction

- **Problem:** Journaling is proven to improve mental health, yet many people struggle to not only build but maintain the habit. The most common pitfalls that make journaling feel like a chore instead of an impactful tool are:
    - Blank page anxiety ➳ the struggle of figuring out how to start an entry and/or the uncertainty of what to write about
    - Shallow logging ➳ entries that only consist of events rather than emotions and/or reflection
    - Lack of insight ➳ difficulty seeing progress or identifying patterns without spending extra time reading many old entries

- **Why it matters:** With anxiety and stress at an all time high worldwide, people need accessible outlets to support their mental health. Journaling has great potential to be a low cost widely available intervention, but its benefits are often lost because of the common pitfalls detailed above. Current journaling tools tend to focus on surface-level customization rather than encouraging reflection. Developing a more engaging, supportive journaling companion can help people of all ages document their days, build healthier habits, reduce stress, and deepen self-awareness.

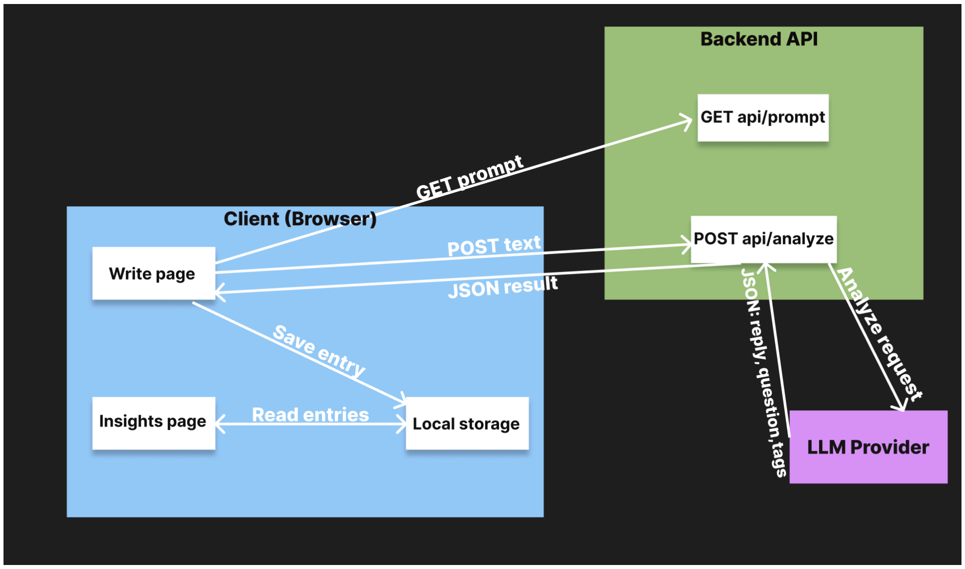
2. System Overview

- **Solution Summary:** A secure/private empathetic journaling companion where users can write an entry, the system will respond supportively, include a reflective question, automatically tag the entry topics, and illustrate simple insights.

- **How it works:**
    1. **Prompt** ➳ The system generates/provides a daily journaling prompt to help users struggling with blank page anxiety.
    2. **Write** ➳ The user writes and submits an entry using the journaling interface.
    3. **Analyze** ➳ The backend architecture utilizes an LLM to analyze the entry. It returns structured insights such as: empathetic response, reflective follow up question, sentiment score, key topics, and short summary.

4. **Storage** ➳ The entry and metadata/insights are saved in the users browser LocalStorage by default providing privacy to all users.
5. **Reflect** ➳ The app uses insights from LLM to quickly provide users with a supportive response and question to deepen reflection.
6. **Insights** ➳ The dashboard compiles entries into simple visuals to identify patterns or lack of in moods/emotions/themes.

- **System architecture:**



3. Design Outline

- **Core Features:**
  - **Guided Writing Prompts** ➳ Users receive a daily prompt from the backend to inspire journaling and reduce "blank page" anxiety.
  - **AI-Powered Insights** ➳ Entries are sent to the backend for analysis and the LLM provides summarized insights, suggested tags, and a follow up question.
  - **Local Storage of Entries** ➳ Journals are saved client-side, allowing offline access and ensuring user privacy while still enabling later review.
- **User Flow:**
  - **User opens app** / **visits site** ➳ Lands on the Write Page.
  - **Receives daily prompt** ➳ Client fetches a prompt from the backend API.
  - **Writes and saves journal entry** ➳ Entry is saved locally, then sent to the backend for analysis.

    - **AI processes entry** ➳ Backend forwards text to the LLM provider, receives insights/tags/questions, and returns them as JSON.
    - **User views insights** ➳ On the Insights Page, the app displays past entries with AI-generated insights and tags for reflection.

4. Tech Stack

- **Frontend:** React + TypeScript with Tailwind CSS used to build a responsive, minimal UI quickly.
- **Backend:** Node.js with Express (TypeScript) utilizing lightweight REST API handling two core endpoints (api/prompt, api/analyze) and managing communication with the LLM service.
- **Database/Storage:** LocalStorage, ensures user entries stay private on the client and data is not accessible by others. Also reduces setup overhead
- **Hosting:** Currently running locally.
- **Other Tools/Services:**
    - Ollama is used as the LLM provider since it runs models locally without token costs
    - OpenAI API — *not used* due to token limitations but possibility to be integrated
    - Zod (schema validation)
    - GitHub for version control

5. Future Enhancements

- **Customizable AI Companion:** Allow users to select the “personality” of their journaling companion (ex. blunt, cautious, optimistic, empathetic) for more personalized reflections.
- **Visual Personalization:** Offer journal background themes, custom fonts, and mood-based color schemes to make journaling feel unique to each user.
- **Various Media Inserts:** Support adding photos, emojis, or small audio snippets to capture a fuller picture of each day.
- **Structured Entry Management:** Provide a timeline or calendar view for past entries, with search, tags, and filters to easily revisit themes or moods.
- **Multi-Device Access:** Securely store entries in the cloud so users can journal seamlessly across devices while still maintaining privacy controls.

- **End-to-End Encryption:** Encrypt journal entries locally before any optional cloud sync to ensure only the user can access their content.
- **Export Controls:** Allow users to securely export/download their data in common formats (PDF, Markdown, JSON).

6. Conclusion

This project demonstrates how journaling can become more supportive, private, and engaging through the use of LLM-powered insights. By focusing on accessibility (local-first design), personalization (future customizable companions), and reflection (structured insights and visuals), the system reimagines digital journaling as an empathetic companion rather than a static notebook. The design balances technical feasibility with user impact, making it a solid start that can grow into a full impactful product!