

Reflect AI

Design Choices, Architecture, and Technical Stack

Overview

Reflect AI is a privacy focused, AI assisted journaling application designed to help users reflect consistently and meaningfully. The system emphasizes simplicity, user trust, and optional AI support while keeping personal data local by default.

The application is intentionally minimal, hackathon ready, and designed with a clear growth path toward more advanced features without compromising user privacy.

System Architecture

Reflect AI follows a simple client server architecture optimized for rapid development and strong privacy guarantees.

Frontend (Client)

- Runs entirely in the browser
- Manages journaling UI, calendar navigation, streak tracking, and insights
- Stores all journal entries locally on the user device

Backend (Server)

- Lightweight Flask based REST API
- Handles only AI related requests such as text rewriting and reflection prompt generation
- Does not store or persist journal data

AI Layer

- Large Language Model used for text rewriting and reflective prompt generation
- Invoked only when the user explicitly requests AI assistance

External Services

- Weather API used for optional contextual information within journal entries

Data Flow

1. The user writes or edits a journal entry locally in the browser
2. Journal data remains on the device by default
3. When AI assistance is requested, only the selected text is sent to the backend
4. The backend forwards the request to the AI layer
5. The AI response is returned to the client and applied within the UI

Design Choices

Local First Storage

Journal entries are stored locally using a JSON based structure.

Reasoning

- Protects sensitive personal data
- Builds strong user trust
- Simplifies MVP deployment and infrastructure

Tradeoff

- No automatic multi device synchronization in the MVP

Lightweight Full Stack

The application uses a Flask backend paired with a vanilla JavaScript frontend.

Reasoning

- Faster development during a hackathon
- Minimal dependencies and tooling
- Easier debugging and deployment

Assistive AI Design

AI is used to support reflection rather than generate content autonomously.

Reasoning

- Preserves the user's authentic voice
- Reduces hallucination and overreach risk
- Ensures predictable and safe outputs

Gentle Habit Building

The system encourages consistency through streaks and positive reinforcement instead of aggressive gamification.

Reasoning

- Encourages regular reflection without pressure
- Aligns naturally with journaling behavior

Technical Stack

Frontend

- HTML5
- CSS3
- Vanilla JavaScript
- Chart.js

Backend

- Python
- Flask

AI and NLP

- Large Language Model for rewriting and reflection prompts
- Lightweight sentiment analysis for mood trend visualization

Storage

- Local JSON storage
- Manual import and export support

External APIs

- Weather API

Privacy and Security

- No authentication required in the MVP
- Journal content stays local unless AI is explicitly triggered
- AI calls are always user initiated
- Full data ownership through import and export functionality

Future Enhancements

- Encrypted local storage
- Optional cloud synchronization with authentication
- Mobile first and responsive experience
- Deeper AI driven insights and summaries
- Offline or fully on device AI support

Summary

Reflect AI prioritizes privacy, simplicity, and meaningful AI assistance. The architecture and design choices emphasize user trust and clarity, making the application ideal for hackathons while maintaining a strong foundation for future expansion.