**EchoJournal - Project Overview**

**Project Vision**

**Problem We're Solving: [Describe what gap in journaling/self-reflection you're addressing]**

**Target Users: [Who would benefit from this approach to journaling?]**

**What Makes EchoJournal Unique:**

* **AI-powered entry analysis and synthesis with multiple personality types**
* **Mode-based journaling experiences with specialized features:** 
  + **Mystical mode: Real-time astrological integration with planetary positions**
  + **Philosophical mode: Deep reflection prompts**
  + **(Other modes with unique characteristics)**
* **Multiple AI synthesis types (Reflect, Digest, Roast, Suggest)**
* **Premium tier system for advanced features**

**Current Features (Working)**

* **User Authentication: Custom signup/login with timezone handling**
* **Mode System: Different journaling modes with unique styling/themes**
* **Daily Entry Limits: 3 entries per day with real-time validation**
* **AI Integration:** 
  + **Groq API for entry synthesis and tag extraction**
  + **Smart pattern recognition in journal entries**
* **Dynamic UI: HTMX-powered interactions for smooth UX**
* **Entry Management: Filter by date, view same-day entries**

**Planned Features**

**Short Term (Next 2-4 weeks)**

* **Multiple AI synthesis types (Reflect, Digest, Roast, Suggest) to replace single synthesis**
* **Enhanced mode-specific styling and UI components**
* **Basic paywall structure implementation**

**Medium Term (1-3 months)**

* **Mystical mode astrological integration:** 
  + **Real-time planetary position calculations using Python astronomical libraries**
  + **AI cross-referencing of journal entries with astrological transits**
  + **User birth chart data collection and storage**
* **Premium feature gates and payment processing**
* **Advanced mode-specific functionalities**

**Long Term Vision**

* **Full astrological analysis system with personalized insights**
* **Additional specialized modes beyond mystical/philosophical**
* **Advanced AI personality customization per user**
* **Mobile app consideration**
* **Community features for premium users**

**Technical Architecture**

**Stack**

* **Backend: Django 5.2.4 + PostgreSQL**
* **Frontend: HTMX + Tailwind CSS**
* **AI: Groq API (compound-beta-mini model)**
* **Infrastructure: [Production plans?]**

**App Structure**

**echojournal/**

**├── journal/ # Core journaling functionality**

**├── userprofile/ # User preferences and settings**

**├── user\_management/ # Authentication and user admin**

**└── echojournal/ # Main project settings**

**Key Models**

* **JournalEntry: User entries with AI-generated tags and mode associations**
* **JournalMode: Different journaling experiences with premium flags**
* **UserProfile: User preferences, mode selections, and premium status**
* **Future: Birth chart data for astrological features**

**Development Approach**

* **Priority: Backend functionality → UI polish (2:1 ratio)**
* **Architecture: Scalable and modular without over-engineering**
* **Code Style: Clear, structured logic with explanations**
* **AI Integration: Thoughtful prompts with personality**

**Current Status & Next Steps**

**What's Working Well**

* **Mode switching system with session/preference fallback**
* **AI tag extraction and synthesis**
* **HTMX dynamic updates**
* **Clean URL structure and view organization**

**Known Issues/Technical Debt**

* **[List any bugs or cleanup needed]**
* **Mode handling functions could be unified**
* **[Other issues you've noticed]**

**Immediate Next Steps**

1. **Implement multiple synthesis types (Reflect, Digest, Roast, Suggest)**
2. **Set up basic premium feature framework**
3. **Research and test astronomical Python libraries for mystical mode**

**Notes & Decisions**

* **Daily entry limit (3/day) encourages quality over quantity**
* **Mode system allows for different journaling mindsets and specialized features**
* **AI provides multiple interaction types without being overwhelming**
* **Premium features considered from early development for sustainable monetization**
* **Astrology integration targets niche but engaged user base**
* **Backend-first development approach: 2 features → 1 UI polish cycle**