### 5.1 Ask Feature

**Feature Goal**: Provide users with a dynamic and engaging way to explore philosophical questions by interacting with AI personas representing different philosophical tones.

#### System Architecture

**Frontend Components**:

* AskScreen.tsx: Main screen for the Ask feature
* QuestionInput.tsx: Text input for questions
* ToneSelector.tsx: Component for selecting philosophical tones
  + Displays 10 defined philosophical tones (Socratic, Oracular, Epic, etc.)
  + Loads tone definitions from configuration file
  + Provides visual representation of each tone
  + Allows previewing tone characteristics
  + Remembers user's preferred tones
* ResponseCard.tsx: Display for AI responses
* ConceptBadge.tsx: Clickable concept tags
* SaveButton.tsx: Allows saving responses to journal
* ShareButton.tsx: Enables sharing insights
* ContemplativeOrb.tsx: Rating component for AI responses

**Backend Components**:

* ai\_router.py: Intelligent model selection (see Section 9)
* ai\_service.py: AI API integration
* concept\_service.py: Concept extraction and linking
* tone\_service.py for philosophical tone management

**Database Models**:

* ai\_interactions: Stores questions, responses, and metadata

CREATE TABLE ai\_interactions (

id UUID PRIMARY KEY DEFAULT uuid\_generate\_v4(),

user\_id UUID NOT NULL REFERENCES users(id),

question TEXT NOT NULL,

response TEXT NOT NULL,

tone VARCHAR(50),

model\_used VARCHAR(50) NOT NULL,

concepts JSONB,

created\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),

is\_saved BOOLEAN DEFAULT FALSE,

is\_expanded BOOLEAN DEFAULT FALSE,

metadata JSONB

);

* concepts: Shared with Explore feature (see Section 5.3)

#### API Endpoints

* POST /api/v1/ai/ask: Submit a question and get a response
  + Request:

{

"question": "What is the meaning of life?",

"tone": "socratic",

"interaction\_id": null

}

* + Response:

{

"interaction\_id": "550e8400-e29b-41d4-a716-446655440000",

"question": "What is the meaning of life?",

"response": "...",

"tone": "socratic",

"model\_used": "claude",

"concepts": [

{

"id": "123",

"name": "meaning",

"description": "...",

"relevance\_score": 0.95

},

...

],

"created\_at": "2025-05-28T15:30:00Z"

}

* POST /api/v1/ai/interactions/{id}/save: Save an interaction
  + Request: Empty body
  + Response:

{

"success": true,

"interaction\_id": "550e8400-e29b-41d4-a716-446655440000",

"xp\_earned": 2

}

* GET /api/v1/ai/interactions: Get user's interaction history
  + Query Parameters:
    - limit: Number of interactions to return (default: 20)
    - offset: Pagination offset (default: 0)
    - saved\_only: Return only saved interactions (default: false)
  + Response:

{

"interactions": [

{

"interaction\_id": "550e8400-e29b-41d4-a716-446655440000",

"question": "What is the meaning of life?",

"response\_preview": "...",

"tone": "socratic",

"created\_at": "2025-05-28T15:30:00Z",

"is\_saved": true

},

...

],

"total": 42,

"limit": 20,

"offset": 0

}

#### User Experience Flow

1. User enters a philosophical question in the QuestionInput component
2. User selects a philosophical tone using the ToneSelector component (or uses default)
3. System sends request to backend via POST /api/v1/ai/ask
4. AI Router selects optimal model (Grok/Claude/Gemini/ChatGPT) based on question content, tone, and user context
5. Selected AI generates response with appropriate philosophical tone
6. Response is displayed in the ResponseCard component with concept badges
7. User can:
   * Save the interaction (earning XP)
   * Expand the insight (get a more detailed response)
   * Seek clarity (ask a follow-up question)
   * Share the interaction
   * Explore related concepts by tapping concept badges

#### XP Integration

* XP awarded for asking questions (+5 XP)
* XP awarded for using different tones (+3 XP per new tone)
* XP awarded for saving insights (+2 XP)
* Badges for question milestones (10, 50, 100 questions)
* Streaks for daily questioning

#### Context-Specific Tone Prompt Workflow

The Ask feature implements the Context-Specific Tone Prompt Workflow (see Section 9.3) to ensure high-quality, consistent philosophical responses across all AI models. Each tone (Socratic, Analytical, Existential, etc.) has specific prompt templates optimized for each AI model.

#### Implementation Considerations

* **Caching Strategy**: Frequently asked philosophical questions will be cached in Redis with a TTL of 24 hours to improve response times and reduce AI API costs.
* **Offline Support**: Recent interactions will be stored locally for offline viewing, but new questions require an internet connection.
* **Concept Extraction**: Performed asynchronously via Celery tasks to avoid blocking the response.
* **Rate Limiting**: Free tier users limited to 10 questions per day, Premium to 50, Pro unlimited.
* **Error Handling**: Graceful fallback to alternative models if primary model fails or times out.
* **Philosophical Tone Implementation**:
  + Configuration-based tone management using `/src/config/philosopher\_tones.json`
  + ToneService integration for consistent tone application
  + Tone-specific prompt templates for each AI model
  + Tone selection persistence in user preferences
  + Analytics tracking for tone usage and effectiveness
  + Redis caching of tone configurations for performance

# 5.1.1 Seek Clarity Feature

**Feature Goal**: Transform the existing "Seek Clarity" button from a simple follow-up question mechanism into a comprehensive philosophical engagement hub with five distinct pathways that create deeper, more meaningful interactions with philosophical content.

## System Architecture

### Frontend Components

* SeekClarityPanel.tsx: Main component that expands from button to action panel
* PathwayCard.tsx: Reusable component for each pathway option
* Pathway-specific components:
  + PracticalExampleModal.tsx: Displays real-world case studies
  + PracticalRephrasingModal.tsx: Shows context-specific applications
  + WisdomJournalIntegration.tsx: Creates structured challenges
  + WisdomSharingModal.tsx: Provides curated sharing options
  + LearningPathModal.tsx: Presents concept-based learning paths

### Backend Components

* seek\_clarity\_service.py: Core service for managing Seek Clarity interactions
* seek\_clarity\_tasks.py: Celery tasks for asynchronous processing
* API endpoints in seek\_clarity\_routes.py

### Database Models

* seek\_clarity\_interactions: Core table tracking all pathway interactions

CREATE TABLE seek\_clarity\_interactions (

id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

user\_id UUID NOT NULL REFERENCES users(id),

response\_id UUID NOT NULL REFERENCES ai\_responses(id),

pathway\_type VARCHAR(50) NOT NULL,

interaction\_data JSONB,

created\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),

completed\_at TIMESTAMP WITH TIME ZONE,

xp\_earned INTEGER DEFAULT 0

);

* practical\_examples: Stores AI-generated examples

CREATE TABLE practical\_examples (

id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

interaction\_id UUID NOT NULL REFERENCES seek\_clarity\_interactions(id),

title VARCHAR(255) NOT NULL,

content TEXT NOT NULL,

source\_url VARCHAR(255),

source\_name VARCHAR(255),

concept\_ids JSONB NOT NULL,

created\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW()

);

* journal\_challenges: Links challenges to journal entries

CREATE TABLE journal\_challenges (

id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

interaction\_id UUID NOT NULL REFERENCES seek\_clarity\_interactions(id),

journal\_entry\_id UUID REFERENCES journal\_entries(id),

challenge\_type VARCHAR(50) NOT NULL,

challenge\_content TEXT NOT NULL,

due\_date TIMESTAMP WITH TIME ZONE,

is\_completed BOOLEAN DEFAULT FALSE,

completion\_date TIMESTAMP WITH TIME ZONE,

concept\_ids JSONB NOT NULL,

created\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW()

);

* learning\_path\_progress: Tracks progress through concept learning paths

CREATE TABLE learning\_path\_progress (

id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

user\_id UUID NOT NULL REFERENCES users(id),

interaction\_id UUID NOT NULL REFERENCES seek\_clarity\_interactions(id),

concept\_id UUID NOT NULL REFERENCES concepts(id),

progress\_percentage INTEGER DEFAULT 0,

is\_completed BOOLEAN DEFAULT FALSE,

created\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),

updated\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW()

);

## API Endpoints

* POST /api/v1/seek-clarity/practical-examples
  + Request:

{

"response\_id": "550e8400-e29b-41d4-a716-446655440000",

"concept\_ids": ["123e4567-e89b-12d3-a456-426614174000", "123e4567-e89b-12d3-a456-426614174001"]

}

* + Response:

{

"task\_id": "750e8400-e29b-41d4-a716-446655440000",

"status": "processing",

"estimated\_time": "15-30 seconds"

}

* GET /api/v1/seek-clarity/practical-examples/{task\_id}
  + Response:

{

"status": "completed",

"examples": [

{

"title": "The Trolley Problem in Modern Healthcare",

"content": "...",

"lesson": "...",

"source\_name": "Journal of Medical Ethics",

"source\_url": "https://example.com/article"

},

{

"title": "Corporate Responsibility and Utilitarianism",

"content": "...",

"lesson": "...",

"source\_name": "Harvard Business Review",

"source\_url": "https://example.com/article2"

}

],

"xp\_earned": 10

}

* POST /api/v1/seek-clarity/practical-rephrasing
  + Request:

{

"response\_id": "550e8400-e29b-41d4-a716-446655440000",

"context": "workplace",

"original\_question": "What is the nature of justice?",

"response\_content": "..."

}

* + Response:

{

"task\_id": "750e8400-e29b-41d4-a716-446655440001",

"status": "processing",

"estimated\_time": "10-20 seconds"

}

* POST /api/v1/seek-clarity/journal-challenge
  + Request:

{

"response\_id": "550e8400-e29b-41d4-a716-446655440000",

"challenge\_type": "weekly\_challenge",

"concept\_ids": ["123e4567-e89b-12d3-a456-426614174000"],

"response\_content": "...",

"custom\_notes": "My personal challenge to practice mindfulness..."

}

* + Response:

{

"challenge\_id": "850e8400-e29b-41d4-a716-446655440000",

"journal\_entry\_id": "950e8400-e29b-41d4-a716-446655440000",

"challenge\_type": "weekly\_challenge",

"xp\_earned": 15,

"reminder\_scheduled": true,

"reminder\_time": "3 days"

}

* POST /api/v1/seek-clarity/wisdom-sharing
  + Request:

{

"response\_id": "550e8400-e29b-41d4-a716-446655440000",

"sharing\_type": "quote",

"selected\_text": "The unexamined life is not worth living.",

"concept\_ids": ["123e4567-e89b-12d3-a456-426614174000"],

"custom\_note": "This Socratic wisdom reminds me to reflect daily."

}

* + Response:

{

"share\_id": "850e8400-e29b-41d4-a716-446655440001",

"share\_image\_url": "https://example.com/share/850e8400.jpg",

"xp\_earned": 8

}

* POST /api/v1/seek-clarity/learning-path
  + Request:

{

"response\_id": "550e8400-e29b-41d4-a716-446655440000",

"concept\_ids": ["123e4567-e89b-12d3-a456-426614174000", "123e4567-e89b-12d3-a456-426614174001"]

}

* + Response:

{

"learning\_path\_id": "850e8400-e29b-41d4-a716-446655440002",

"concepts": [

{

"id": "123e4567-e89b-12d3-a456-426614174000",

"name": "Justice",

"description": "...",

"resources": [...]

},

{

"id": "123e4567-e89b-12d3-a456-426614174001",

"name": "Equality",

"description": "...",

"resources": [...]

}

],

"xp\_earned": 10

}

## User Experience Flow

### Stage 1: Initial Engagement (0-5 seconds)

1. User reads AI response with concept badges
2. Seek Clarity button pulses subtly, indicating available actions
3. Hover state reveals preview text: "Turn this insight into action"

### Stage 2: Pathway Selection (5-15 seconds)

1. Click opens elegant overlay with five pathway cards
2. Each pathway shows:
   * Clear action-oriented title
   * One-line benefit statement
   * Relevant icon and concept tags
   * Estimated time commitment

### Stage 3: Deep Engagement (15 seconds - 10 minutes)

1. User selects pathway based on their immediate need
2. Experience adapts to chosen pathway with appropriate interface
3. Cross-pathway integration allows seamless transitions

### Stage 4: Closure & Continuation (Final 30 seconds)

1. Action completion triggers appropriate XP rewards
2. Related pathway suggestions appear
3. Return to main conversation with enhanced context

## The Five Pathways

### Pathway 1: "See It in Action" (Practical Example Modal)

* **User Journey**: Click reveals real-world case studies tied to the response's philosophical concepts
* **Content Structure**: Story-driven examples with "Learn More" links to authoritative sources
* **Technical Implementation**:
  + Asynchronous generation via Celery tasks
  + AI Router selects optimal model (preferring Claude and Grok)
  + Examples stored in practical\_examples table
  + XP awarded on viewing examples

### Pathway 2: "Make It Practical" (Practical Rephrasing)

* **User Journey**: AI rephrases the philosophical response for real-world application
* **Interactive Element**: Users can specify their context (workplace, relationships, community)
* **Technical Implementation**:
  + Context-specific rephrasing via AI Router
  + Pragmatic tone selection
  + Follow-up system for deeper questioning
  + XP awarded for each context explored

### Pathway 3: "Take Action" (Wisdom Journal Integration)

* **User Journey**: Transform insights into structured, time-bound challenges
* **Content Types**:
  + Micro-Steps: 5-minute philosophical practices
  + Weekly Challenges: Meaningful actions tied to concepts
  + Reflection Prompts: Questions for deeper contemplation
* **Technical Implementation**:
  + Integration with Journal feature
  + Challenge templates based on type
  + Notification reminders via Celery scheduled tasks
  + Progress tracking and completion rewards

### Pathway 4: "Share Wisdom" (Enhanced Sharing)

* **User Journey**: Curated sharing options with philosophical framing
* **Content Varieties**:
  + Quote cards with personal insights
  + Challenge announcements
  + Wisdom milestones
* **Technical Implementation**:
  + Image generation for shareable cards
  + Social media integration
  + Community hashtag strategy
  + XP rewards for sharing

### Pathway 5: "Explore Deeper" (Gamified Learning Paths)

* **User Journey**: Concept trails that connect related philosophical ideas
* **Progression System**: Points and badges tied to meaningful engagement
* **Technical Implementation**:
  + Concept relationship mapping
  + Progressive learning resources
  + Progress tracking in learning\_path\_progress table
  + XP rewards for milestones and completion

## Celery Integration

The Seek Clarity feature leverages Celery for asynchronous processing with specialized queues:

# Queue structure

seek\_clarity\_exchange = Exchange('seek\_clarity', type='direct')

task\_queues = (

Queue('seek\_clarity\_ai', seek\_clarity\_exchange, routing\_key='seek\_clarity.ai',

queue\_arguments={'x-max-priority': 7}),

Queue('seek\_clarity\_journal', seek\_clarity\_exchange, routing\_key='seek\_clarity.journal',

queue\_arguments={'x-max-priority': 5}),

Queue('seek\_clarity\_analytics', seek\_clarity\_exchange, routing\_key='seek\_clarity.analytics',

queue\_arguments={'x-max-priority': 3}),

)

# Task routing

task\_routes = {

'tasks.seek\_clarity\_tasks.generate\_practical\_examples': {

'queue': 'seek\_clarity\_ai',

'routing\_key': 'seek\_clarity.ai'

},

'tasks.seek\_clarity\_tasks.generate\_practical\_rephrasing': {

'queue': 'seek\_clarity\_ai',

'routing\_key': 'seek\_clarity.ai'

},

'tasks.seek\_clarity\_tasks.create\_journal\_challenge': {

'queue': 'seek\_clarity\_journal',

'routing\_key': 'seek\_clarity.journal'

},

'tasks.seek\_clarity\_tasks.track\_pathway\_analytics': {

'queue': 'seek\_clarity\_analytics',

'routing\_key': 'seek\_clarity.analytics'

},

}

Key tasks include:

* generate\_practical\_examples: Creates real-world examples using AI Router
* generate\_practical\_rephrasing: Rephrases philosophical content for practical contexts
* create\_journal\_challenge: Generates structured challenges in the Journal
* send\_pathway\_completion\_notification: Sends reminders and updates
* track\_pathway\_analytics: Analyzes pathway usage patterns

## Notification Integration

The Seek Clarity feature integrates with the notification system for:

1. **Pathway Completion Notifications**:
   * Practical examples ready for viewing
   * Practical rephrasing completed
   * Journal challenge reminders
   * Learning path progress updates
2. **Achievement Notifications**:
   * XP earned from pathway interactions
   * Badges unlocked through pathway usage
   * Challenge completion celebrations
3. **Engagement Notifications**:
   * Reminders for incomplete challenges
   * Suggestions for new pathways based on usage patterns
   * Community engagement opportunities

Notifications are delivered through:

* Push notifications (for critical updates)
* In-app notifications (for all interactions)
* Email digests (for weekly summaries)

## Journal Integration

The "Take Action" pathway deeply integrates with the Journal feature:

1. **Challenge Templates**:
   * Micro-Step (5-minute practices): Daily template with progress tracking
   * Weekly Challenge: 7-day structure with daily reflection sections
   * Reflection Prompt: Guided philosophical contemplation
2. **Progress Tracking**:
   * Challenge status in Journal entries
   * Due dates and reminders
   * Completion celebrations
3. **Concept Tagging**:
   * Automatic tagging with relevant philosophical concepts
   * Concept mastery progress through challenge completion

## AI Router Integration

The Seek Clarity feature leverages the AI Router for intelligent content generation:

1. **Pathway-Specific Model Selection**:
   * Practical Examples: Prefers Claude and Grok for educational content
   * Practical Rephrasing: Prefers Grok and Gemini for pragmatic applications
   * Journal Challenges: Prefers Claude and ChatGPT for contemplative prompts
   * Learning Paths: Prefers Claude for structured educational content
2. **Tone Selection**:
   * Educational tone for practical examples
   * Pragmatic tone for practical rephrasing
   * Contemplative tone for journal challenges
   * Reverential tone for wisdom sharing
   * Socratic tone for learning paths
3. **Context-Aware Prompting**:
   * Includes user's selected context in prompts
   * Incorporates concept information for relevance
   * Adapts to user's philosophical interests

## XP Integration

The Seek Clarity feature awards XP for meaningful engagement:

| **Action** | **XP Value** |
| --- | --- |
| Opening Seek Clarity panel | 2 XP |
| Viewing practical examples | 10 XP |
| Generating practical rephrasing | 12 XP |
| Creating journal challenge | 15 XP |
| Completing journal challenge | 25 XP |
| Sharing wisdom | 8 XP |
| Starting learning path | 10 XP |
| Learning path progress milestone | 5 XP |
| Completing learning path | 30 XP |

Special badges include:

* **Clarity Explorer**: Used all five Seek Clarity pathways (50 XP)
* **Practical Philosopher**: Completed 10 practical rephrasing interactions (75 XP)
* **Challenge Master**: Completed 5 weekly philosophical challenges (100 XP)
* **Wisdom Sharer**: Shared philosophical insights 20 times (60 XP)
* **Learning Path Pioneer**: Completed 3 philosophical learning paths (125 XP)

## Analytics Integration

The Seek Clarity feature tracks detailed analytics:

1. **Engagement Metrics**:
   * Pathway selection distribution
   * Time spent in each pathway
   * Completion rates by pathway
   * Return engagement patterns
2. **User Value Metrics**:
   * XP earned through Seek Clarity
   * Challenge completion rates
   * Learning path progression
   * Sharing engagement
3. **Content Performance**:
   * Most engaging practical examples
   * Most effective rephrasing contexts
   * Most popular challenge types
   * Most traversed learning paths

## Implementation Considerations

1. **Progressive Disclosure**:
   * Initial simple interface with depth on demand
   * Complexity increases with user engagement
   * Advanced features revealed through usage
2. **Performance Optimization**:
   * Caching of AI-generated content
   * Preloading of pathway options
   * Efficient database queries for related concepts
3. **Accessibility**:
   * Full keyboard navigation support
   * Screen reader compatibility
   * Color contrast compliance
   * Touch-friendly mobile interactions
4. **Error Handling**:
   * Graceful fallbacks for AI generation failures
   * Offline support for previously loaded content
   * Clear error messaging and recovery options

## Implementation Phases

The implementation will follow these phases:

### Phase 1: Foundation (Weeks 1-3)

* Redesign Seek Clarity button interface
* Implement basic pathway navigation
* Create database schema and models
* Set up API endpoints

### Phase 2: Core Features (Weeks 4-8)

* Build all five pathway experiences
* Integrate with existing Wisdom Journal
* Implement XP and notification integration
* Set up Celery tasks for AI processing

### Phase 3: Intelligence (Weeks 9-12)

* Add contextual content matching with AI Router
* Implement cross-pathway integrations
* Enhance sharing capabilities
* Add analytics tracking

### Phase 4: Polish (Weeks 13-16)

* Accessibility optimization
* Performance optimization
* Community features enhancement
* Final testing and deployment