Basic Project Plan

Phase 1: Foundation

1. Environment setup and basic project structure
2. Database schema design
3. Basic web scraping implementation
4. Document processing pipeline setup

Phase 2: Core Features

Task 1: Vector Database Integration
About:Setting up and integrating the vector database for embedding storage and retrieval.

Implementation Approach:
- Configure Qdrant
- Implement embedding generation
- Create storage procedures
- Set up retrieval mechanisms

Key Learning Needed
- Vector databases
- Embedding concepts
- Similarity search algorithms
- Performance optimization

Importance:
Critical for efficient similarity search and context retrieval in the RAG system.

Prioritization Rationale:
Vector database integration is essential for RAG functionality and needs to be implemented before the main RAG features.

Task 2: RAG System Implementation
About: Building the core RAG system with context retrieval and generation capabilities.

Implementation Approach:

- Integrate LLM API
- Implement context retrieval
- Set up prompt templates
- Create response generation pipeline

Key Learning Needed:

- LLM concepts
- Prompt engineering
- Context window management
- Token handling

Importance:

This is the core functionality that enables intelligent document interaction.

Prioritization Rationale:

RAG implementation is the main feature and needs sufficient time for testing and refinement.

Phase 3: Enhancement

Task 1: Frontend Development

About:Creating the user interface for document upload and interaction.

Implementation Approach:

- Set up frontend framework
- Create upload interface
- Implement chat UI
- Add progress indicators

Requirements:

- Next.js/Nuxt.js
- Real-time updates
- File upload handling
- User feedback mechanisms

Importance:

Essential for user interaction and system usability.

Prioritization Rationale:
Frontend development can proceed in parallel with backend refinements and needs significant testing time.

Phase 4: Finalization

Task 1: Testing and Optimization
About:Comprehensive testing and performance optimization.

Implementation Approach
- Write unit tests
- Perform integration testing
- Optimize performance
- Implement caching

Importance:
Critical for ensuring system reliability and performance.

Prioritization Rationale:
Testing and optimization need to be done after core features are implemented but before final deployment.