Kylie Higashionna

# ICS 603 Project Description Document

What the application does:

This application allows users to create, store, and explore personal reflections. New reflections can be added through the **Add** tab, where each entry is automatically analyzed and classified with 1–3 key topics. All reflections appear in a structured table on the **View** tab, which can be filtered to show only the user's entries; clicking a reflection's title the user can open and read the full version of their past entry. The **Search** tab lets users find past reflections using semantic embeddings by entering keywords or phrases and optionally filtering by user. The **Recommendations** tab provides AI-generated, personalized recommendations based on a user's reflection history, using the context and prompt they provide to generate tailored insights.

The AI techniques it uses and how:

- **Topic Classification:** Uses PydanticAI with the GPT-4o-mini model to analyze reflection text and assign 1–3 relevant topics from existing categories or suggest new ones.
- **Text Embeddings & Semantic Search:** Generates high-dimensional vector embeddings using OpenAI's text-embedding-3-large model. Stores embeddings in ChromaDB vector database for efficient similarity search. Enables users to find reflections by meaning rather than exact keywords.
- **Recommendation Generation:** Uses Retrieval-Augmented Generation (RAG) to provide personalized recommendations. Retrieves the most relevant reflections based on the user's context query using vector-based semantic similarity. Generates personalized recommendations by grounding GPT's responses in the user's actual reflection history. The analyze_topics tool further enhances this by analyzing topic frequencies across reflections to identify dominant interests before making recommendations.

Key features and functionality:

**Reflection Creation and Reading**: Create and read past reflections.
**Automatic Topic Tagging:** AI-powered classification of reflections into meaningful categories.
**Semantic Search:** Natural language querying to discover past reflections by concept.
**Personalized AI Recommendations:** Context-aware suggestions generated based on your own reflection history and insights.
**Database Integration**: Uses SQLAlchemy with a Supabase Postgres backend for persistent storage.
**Embeddings Storage**: ChromaDB stores and queries embeddings for semantic search and recommendation tasks.