LLM Usage Appendix (CSE3063 Term Project)

Group: 13 **Project:** MiniRAG Chatbot (Iteration 2)

1. Overview of Usage

In Iteration 2, we utilized Large Language Models (specifically Google Gemini) primarily as a 'Refactoring Partner' and 'Test Engineer.' The core focus was transitioning from a purely heuristic system to a Dual-Mode Architecture (Simple vs. Cosine) integrated with Vector Embeddings. The LLM played a key role in integrating sentence-transformers for semantic search and implementing complex scoring logic, such as Scope Penalty and Definition Priority. Furthermore, it assisted in generating robust unit tests that mocked heavy libraries to achieve the >80% coverage requirement. No generated code was accepted blindly; all logic was carefully validated and adapted.

2. Key Prompts and Adaptations

Scenario A: Enhancing Precision in Retrieval (Logic Refactoring)

- **Prompt:** "Fix impl.py so that staff office info (e.g., Mustafa Ağaoğlu) isn't cut off by long credential lists. Also, ensure queries for course codes (CSE3063) return the definition, not prerequisites."
- **LLM Output:** Suggested increasing the line scan range in VectorAnswerAgent and adding a hardcoded "Silver Bullet" score boost for lines starting with the exact course code.
- **Our Validation & Edit:** The initial scan range fix caused profile merging (combining two professors). We added a logic check to stop scanning if a new title (e.g., "Prof.") is detected. We also verified the "Silver Bullet" regex against edge cases like "CSE3063.1".

Scenario B: Achieving Code Coverage (Testing)

- **Prompt:** "Generate unit tests (≥6 tests) covering Intent, Stopwords, Reranking, and Citations. Tests must mock embedding vectors."
- **LLM Output:** Provided a unittest suite (test_rag.py) mocking get_embedding.
- **Our Validation & Edit:** We added specific test cases to cover the CosineReranker class, ensuring >80% coverage.

3. Code Correctness & Design Quality

Logic Verification: The "Header Lookback" algorithm (used to find regulation headers for bullet points) was manually stepped through in the debugger to ensure it correctly handles nested lists.