Module 2 Capstone Milestone: Project Proposal

Alexander Ricciardi

Colorado State University Global

CSC480: Capstone Computer Science

Dr. Shaher Daoud

June 22, 2025

Module 2 Capstone Milestone: Project Proposal

Title 30 of the Code of Federal Regulations (CFR), also referred to as the Mine Safety and Health Administration (MSHA) regulations, is a set of federal regulations that govern the United States mining industry. These regulations ensure that mine operations are safe and protect individuals. This document is a proposal for the development of an MSHA Regulatory Conversational Agent (MRCA) that will quickly and accurately access information about MSHA regulations.

The Problem

The sheer size and complexity of the CFR Title 30 make it very difficult for mine workers to find quick and accurate answers to specific questions about MSHA regulations, especially when on a mine site. Methods such as searching through large printed or digital volumes are time-consuming and very inefficient, and generic search engines (e.g., Google) and large language models (LLMs) often provide unreliable or fabricated information (Ricciardi, 2025). A possible solution to this problem is an MRCA.

The Solution

MRCA is an AI-powered web application that will provide a quick, reliable, and easy way to query MSHA regulations using natural language by combining an LLM with a GraphRAG hybrid search. The GraphRAG is "a powerful technique that enhances downstream task execution by retrieving additional information, such as knowledge, skills, and tools from external sources" (Han et al., 2025, p.1). The GraphRAG hybrid search, also called HybridRAG, is defined as a novel approach to the Retrieval Augmentation Generation (RAG) technique that combines GraphRAG and VectorRAG techniques to enhance question-answer systems for information extraction (Sarmah et al., 2024). HybridRAG extracts contextual relational data from

a knowledge graph using traversal search, and it extracts contextual semantic information for the vector store using semantic search. This hybrid approach to RAG significantly decreases LLM "hallucinations" with an accuracy of 92.7%, which is a significant improvement over baseline RAG (Vector RAG) with an accuracy of 52.2% (Xu et al., 2024). This combination of LLM and GraphRAG hybrid search is the core component of the MRCA, allowing it to provide users with a quick and easy way to access MSHA regulation information using natural language.

MRCA System Overview

MRCA will be designed to be modular, meaning that the application will be organized as a "logical grouping of related code" (Richards & Ford, 2020, p.40) distributed within a file structure and classes. The Title 30 CFR data will be downloaded from GovInfo.gov in PDF format, distributed in three volumes. Gemini 2.5 pro will be used to generate the knowledge graph and to chunk and vectorize the download Title 30 CFR data, and the resulting graph and vector store will be stored in a Neo4j Aura Database. OpenAI 4o will act as the conversational agent, handling the interaction with the user, embedding user prompts to perform semantic search. Then, based on the semantic search results, it will generate Cypher code that will be used to perform the traversal search. The application's front-end will be handled by a web chat system based on the framework Streamlit. The application's back end will be handled by FastAPI web framework for RESTful API. The project will be coded in Python 3.12.1 and deployed on Streamlit Community Cloud. The development process will use the iterative methodology Kanban, git/GitHub for version control, and Docker for environment containerization.

MRCA Timeline and Major Components

The project must be implemented within eight weeks. The table below illustrates the timeline and describes the project development phases and related components.

Table 1

Project Phases and Timeline

Timeline and Phases	Phase Description and Major Component
Phase 1 Backend Setup Weeks 1-2	This phase consists of setting up the environment and Docker; gathering the data, developing the `cfr_downloader.py` script to download the MSHA data; and developing the `build_graph.py` script to chunk, vectorize, and graph the downloaded data.
Phase 2 Agent & Tools Setup Weeks 3-4	This phase sets up the agent and its tools, such as `tools/vector.py` for vector search and `tools/cypher.py` for graph query. It will also implement `llm.py` for LLM connection and initialize the FastAPI implementation.
Phase 3 Agent & Tool Testing Weeks 5-6	This phase tests the agent and its tools. It implements `agent.py`, a ReAct agent for orchestrating tools, and `tools/general.py` for MSHA regulations domain restriction.
Phase 4 Testing, Optimization, & Deployment Week 7-8	This phase finalizes the Streamlit user interface, `bot.py`, performs end-to-end testing, optimizes, and deploys the app. It also cleans/retests the code, finishes testing the FastAPI integration, and finishes project documentation

Note: The table describes the project's phases, related major components, and timeline.

The development process shown in the Table seems to be strictly sequential; however, the process is iterative as each component is built and tested in iterative steps throughout the phases.

Conclusion

The MRCA web application is a solution for accessing quickly and easily accurate information about MSHA regulations. It will combine the functionality of LLM with HybridRA, which significantly reduces LMM hallucinations, making the combo a reliable method for information extraction. Although this RAG method has been shown to have an accuracy of 92.2%, MRCA should not be used as a replacement for the official Title 30 CFR documentation. Nonetheless, it is a valuable and practical tool for miners, contractors, and safety managers that provides quick and accurate access to MSHA regulations, helping them to stay informed, work safely, and comply with regulations.

References

Han, H., Wang, Y., Shomer, H., Guo, K., Ding, J., Lei, Y., Halappanavar, M., Rossi, R. A., Mukherjee, S., Tang, X., He, Q., Hua, Z., Long, B., Zhao, T., Shah, N., Javari, A., Xia, Y., & Tang, J. (2025, January 25). Retrieval-Augmented Generation With Graphs (GraphRAG). *arXiv*. https://arxiv.org/abs/2501.00309

Richards, M., & Ford, N. (2020). Chapter 4: Architecture Characteristics Defined. *Fundamentals of software architecture: An engineering approach (Softcover)* (pp. 39-56). O'Reilly Media. ISBN-13: 978-1-492-04345-4

Sarmah, B., Patel, S., Hall, B., Pasquali, S., Rao, R., & Mehta, D. (2024, August 9). HybridRAG: Integrating Knowledge Graphs and Vector Retrieval Augmented Generation for Efficient Information Extraction. *arXiv*. https://arxiv.org/abs/2408.04948

Ricciardi (2025, June 15). Module 1 Capstone Milestone: Topic Approval [Student Essay]. CSC480 Capstone Computer Science. CSU Global.

Xu, Z., Cruz, M. J., Guevara, M., Wang, T., Deshpande, M., Wang, X., & Li, Z. (2024, July 11). Retrieval-Augmented Generation with Knowledge Graphs for Customer Service Question Answering. *ACM Digital Library*, SIGIR '24: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, p.2905-2909. https://doi.org/10.1145/3626772.366137