AI-Powered Chatbot for Efficient Navigation of Battery Research Literature

Student(s)

Hasan Fırat Yılmaz -29002
Ali Cenker Yakışır - 28831
Anıl Şen - 29556

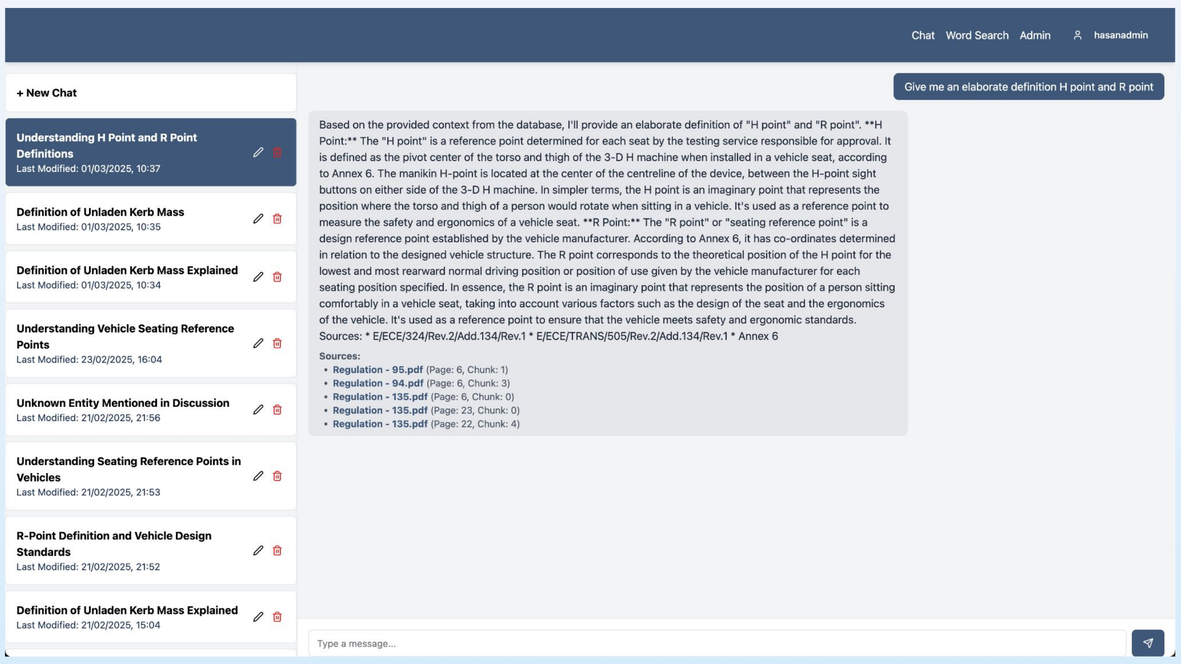
Faculty Member(s)

İnanç Arın

Company Advisor(s)

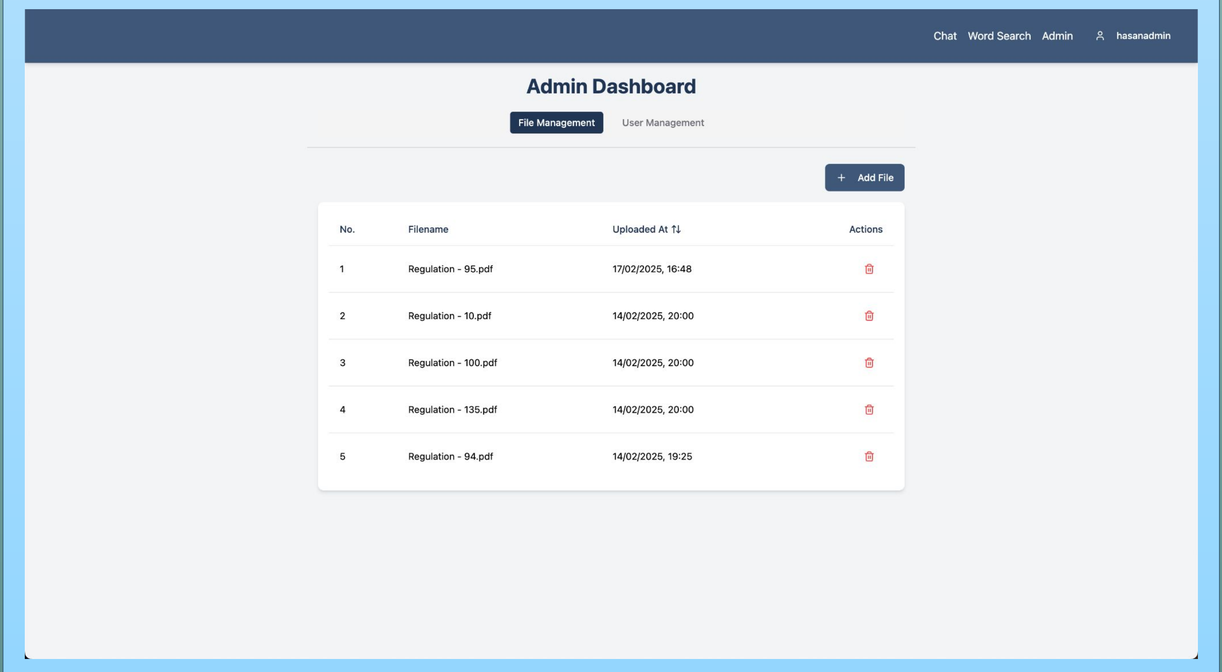
Burcu Oral
Selim Yılmaz
Çağla Odabaşı Özer

Sabancı Üniversitesi
FACULTY OF ENGINEERING AND NATURAL SCIENCES



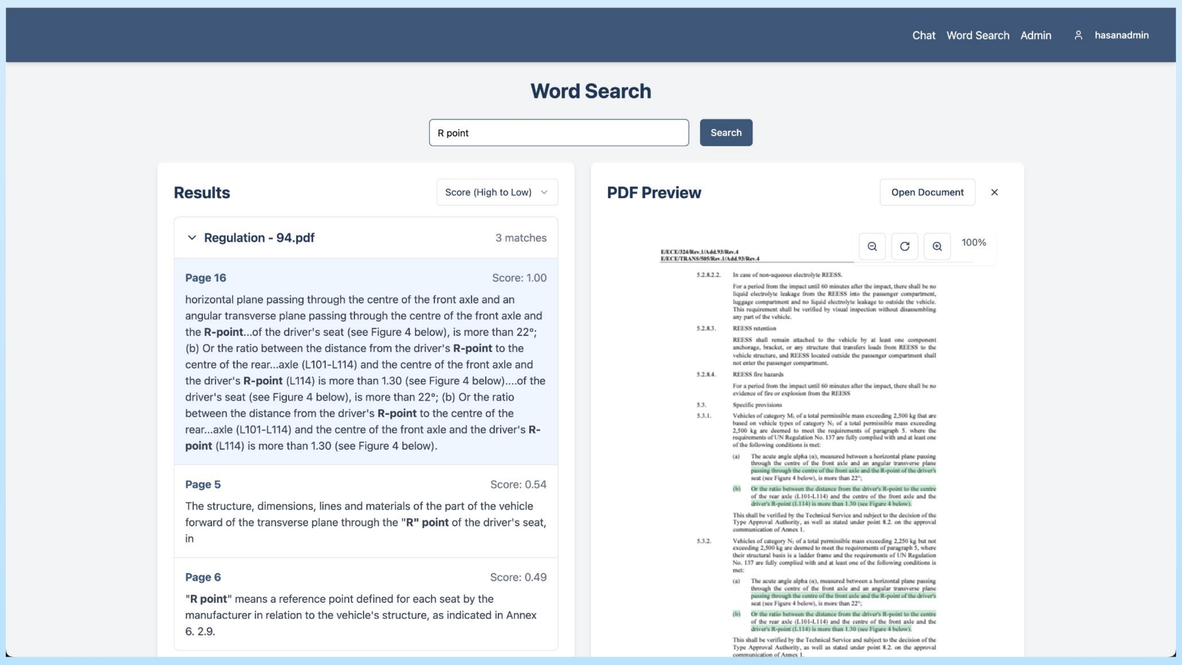
ABSTRACT

We developed a web-based platform to support engineers at Siro, especially homologation engineers, in navigating the complex vehicle regulations published by the United Nations Economic Commission for Europe (UNECE). These lengthy documents are often difficult to review manually, resulting in time-consuming compliance processes. A significant amount of engineering time is spent ensuring and cross-checking that the relevant regulations are properly followed. To address this, our system uses a Retrieval-Augmented Generation (RAG) approach powered by a Large Language Model, enabling users to ask natural language questions and receive accurate, context-based responses grounded in the uploaded regulations. Additionally, we implemented a keyword search mechanism that allows users to instantly locate specific terms across hundreds of documents spanning hundreds of pages, making it easy to identify relevant regulations quickly. With the help of our platform, we aim to improve the efficiency of the compliance process by enabling quick access to information through a chatbot interface and by allowing users to detect applicable regulations through targeted keyword searches.



PROJECT DETAILS

This project delivers an AI-powered chatbot, which is designed to assist engineers in searching over UNECE vehicle regulations through an efficient Retrieval-Augmented Generation (RAG) framework. Utilizing a LLaMA 3 large language model hosted locally via Ollama, the system retrieves contextually relevant information given the user input from a vector database using a hybrid bi-encoder and cross-encoder pipeline. Regulation documents are embedded with the nomic-embed-text model and indexed in a Chroma vector database for high-precision retrieval. Along with these, a keyword-based search system is built, which is powered by Elasticsearch. This system further enhances document exploration with support for fuzzy matching. The web application utilizes React framework for frontend and Django for backend, offering role-based access, chat functionality, PDF document highlighting, and user management. The application is containerized using Docker, whose modular architecture ensures scalable deployment and easy maintenance.



CONCLUSION

This project successfully demonstrates the practical application of a Retrieval-Augmented Generation (RAG) system for domain-specific question answering within the context of UNECE vehicle regulations. The hybrid retrieval approach—combining bi-encoder and cross-encoder techniques—proved highly effective in locating relevant document sections, enabling accurate and context-aware responses from the language model. Moreover, Elasticsearch-powered search provided keyword-based document navigation. The platform's full-stack web architecture includes secure user management, role-based access, and a responsive, user-friendly interface designed for both technical and non-technical users. With modular Docker deployment and an intuitive admin system, the solution is scalable, maintainable, and ready for real-world adoption.

REFERENCES

"Configuring Chroma Collections." Chroma Docs. docs.trychroma.com/docs/collections/configure. Accessed 28 Dec. 2024.

"Django REST Framework Simple JWT: Settings." Django REST Framework Simple JWT, https://django-rest-framework-simplejwt.readthedocs.io/en/stable/settings.html. Accessed 28 Dec. 2024.

"Django Documentation." Django, https://docs.djangoproject.com/en/5.1/topics/auth/passwords/#. Accessed 28 Dec. 2024.

"Elasticsearch Guide [7.17] | Elastic." Elastic, www.elastic.co/guide/en/elasticsearch/reference/7.17/index.html. Accessed 22 Mar. 2025.

"Llama 3.1:8b." Ollama, https://ollama.com/library/llama3.1:8b. Accessed 28 Dec. 2024.

"nomic-embed-text." *Ollama*, ollama.com/library/nomic-embed-text. Accessed 28 Dec. 2024.

"Okapi BM25." Wikipedia, 24 Feb. 2021, en.wikipedia.org/wiki/Okapi_BM25.

"Ollama." *Ollama*, ollama.com/search?c=embedding. Accessed 28 Dec. 2024.

"sentence-transformers/all-mpnet-base-v2." Hugging Face, huggingface.co/sentence-transformers/all-mpnet-base-v2. Accessed 28 Dec. 2024.

"Text embedding." Nomic Atlas Documentation, docs.nomic.ai/atlas/models/text-embedding. Accessed 28 Dec. 2024.

Thawal, Ankit. Elasticsearch Filters and BM25 Relevancy Scoring. 17 Mar. 2025, www.linkedin.com/pulse/elasticsearch-filters-bm25-relevancy-scoring-ankit-thawal-m15jf. Accessed 22 Mar. 2025.

Wikipedia Contributors. "Tf–Idf." Wikipedia, Wikimedia Foundation, 8 Sept. 2019, en.wikipedia.org/wiki/Tf%E2%80%93idf.