Project Report

# Author

Name: Laksh Krishna Sharma

Email: asklakshsharma@gmail.com

I am a passionate computer science student currently pursuing a dual degree: a Bachelor of Science in Data Science from IIT Madras and a Bachelor of Technology in Computer Science from Maharaja Agrasen Institute of Technology. My academic and professional interests lie in the areas of intelligent systems, machine learning, and quantitative finance. I am an active contributor to open-source communities and have hands-on experience with tools and frameworks such as LangChain, Flask, FastAPI, Docker, and various machine learning and deep learning libraries.

# Project Overview

This project presents an interactive AI-powered chatbot that assists users in conducting research across large sets of documents. The system is capable of:

- Extracting relevant information from diverse document types.

- Performing intelligent similarity searches.

- Identifying recurring themes across documents.

- Generating comprehensive, well-cited answers to user queries.

It is particularly well-suited for use cases such as legal research, academic literature review, and knowledge discovery in large text corpora.

# System Architecture

The backend is modular and structured into several components:

-Document Processor: Handles file uploads, content extraction (from PDFs, DOCX, images, etc.), and text chunking for analysis.

- Vector Store: Stores semantic embeddings of document chunks for fast and efficient similarity-based retrieval.

- Query Processor: Manages semantic search, invokes the LLM (hosted via Groq), and synthesizes theme-based responses.

- Metadata Management: Basic information such as document name, paths, and processed file references are maintained for user reference.

Workflow Overview:

1. Users upload one or more documents.

2. Text is extracted, cleaned, and divided into manageable chunks.

3. Each chunk is embedded and stored in the vector database.

4. A user query is semantically matched with relevant document chunks.

5. The system identifies key information, recurring themes, and provides a comprehensive response with citations.

# Technologies Utilized

|  |  |
| --- | --- |
| Technology | Purpose |
| FastAPI | Backend framework for building high-performance RESTful APIs. |
| LangChain | Facilitates large language model orchestration and query resolution. |
| pypdf | Enables extraction of text from PDF documents. |
| Chroma (Vector DB) | Stores and retrieves document chunks based on vector similarity. |
| Docker | Ensures containerized deployment and environment consistency. |
|  |  |

# API Endpoints

-POST /api/documents/upload

Allows multiple document uploads. Extracts and chunks content, stores embeddings, and returns metadata.

-GET /api/documents

Fetches a list of all uploaded documents and their metadata.

-POST /api/query

Processes a user’s query and returns relevant information with citation, identified themes across documents, and a synthesized answer.

# Demo Video

[Video Link](https://drive.google.com/file/d/1DeYpGnvD4Q0Gh0tFz11o14s01qIdc6cD/view?usp=drive_link)

# Conclusion

This project demonstrates how large language models can be effectively combined with vector databases to create intelligent and insightful research assistants. The chatbot offers an efficient and user-friendly way to explore large volumes of documents, find patterns, and extract meaningful insights, all while maintaining citation and context. With potential applications in academia, law, journalism, and research, the system lays the groundwork for more advanced AI-assisted information retrieval systems.