Introduction

This document is created for an AI-based Document Search and Knowledge Retrieval project. Its purpose is to test document loading, chunking, vector embeddings, semantic search, and conversational question answering.

Problem Statement

Organizations store large volumes of documents in PDF and text formats. Searching manually for information is time-consuming and inefficient. There is a need for an intelligent system that allows users to ask natural language questions and receive accurate answers from documents.

Objectives

The main objectives of this project are:

1. Upload and process documents
2. Split documents into meaningful chunks
3. Generate embeddings for semantic search
4. Retrieve relevant content using vector databases
5. Provide conversational answers using an LLM

System Architecture

The system consists of a Streamlit-based user interface, a backend built using LangChain, a vector database for similarity search, and a large language model for generating answers. The user interacts only with the UI, while all processing happens in the backend.

Technologies Used

The project uses Python as the programming language. Streamlit is used for the frontend interface. LangChain is used for document loading and retrieval. ChromaDB is used as the vector store. HuggingFace sentence transformers are used for embeddings. Google Gemini is used as the language model.

Use Cases

This system can be used in education for searching academic notes, in companies for querying policy documents, and in research for fast information retrieval from large reports.

Advantages

The system reduces manual effort, saves time, supports natural language queries, and improves information accessibility.

Limitations

The quality of answers depends on the quality of documents. Highly repetitive documents may produce similar answers. The system requires internet access for the language model.

Future Enhancements

Future improvements may include multi-document comparison, answer citations with page numbers, support for images and tables, and deployment on cloud platforms.

Conclusion

This document demonstrates how an AI-powered document copilot can simplify knowledge retrieval. The project showcases practical usage of modern NLP and vector search technologies in real-world applications.