CAVEO Chatbot

Local PDF Intelligence

Secure, Offline, and Transparent Document Interaction

Introduction

The CAVEO Chatbot aims to revolutionize how we interact with internal documentation. Designed for the CAVEO IT team, this chatbot provides secure, offline, and efficient access to critical information contained within various PDF documents.

This project aims to develop an intelligent chatbot using Python to assist employees in navigating and understanding the company's internal Project Management Manual. The chatbot will allow users to interact in natural language to ask questions, receive guidance, an access specific procedures or templates from the manual. The solution will be built using free tools during the initial phase, with a roadmap for professional upgrades once validated. This tool is designed to improve accessibility, save time, and standardize how projec management resources are consulted within the company.

Offline Operation

Ensuring data never leaves your environment for maximum privacy

Enhanced Security

Protecting sensitive information with a fully localized infrastructure

Efficient Retrieval

Rapidly locating and synthesizing answers from vast document libraries

This solution addresses the need for a reliable, isolated information retrieval system, especially for proprietary and sensitive data.

Problem Statement & Motivation

Organizations face significant difficulties in accessing and utilizing information from extensive PDF documentation:

Current Challenges

- **Information Silos**: Critical knowledge trapped in static PDF documents that are difficult to search and navigate
- **Time-Consuming Manual Search**: Employees spend excessive time manually searching through lengthy documents to find specific information
- **Inconsistent Information Access**: Different team members may interpret or access information differently, leading to inconsistencies
- **Security Concerns**: Need for secure, offline solutions that don't compromise sensitive organizational data
- **Accessibility Issues**: Complex technical documents are often difficult for non-experts to understand and navigate

Motivation for Development

- **Intelligent Information Retrieval**: Transform static documents into interactive, searchable knowledge bases
- **Enhanced Productivity**: Reduce time spent searching for information from hours to seconds
- **Standardized Access**: Ensure consistent information deliver across all team members
- **Security-First Approach**: Maintain complete data privacy wi offline, local processing
- **User-Friendly Interface**: Make complex documentation accessible to users of all technical levels

This solution directly addresses the growing need for efficient, secure, and intelligent document interaction systems in modern organizations.

Project Development Roadmap

Week 1

- Requirements gathering and project planning
- Set up development environment and tools

Week 2

- Design system architecture and user interface
- Prepare initial project structure and documentation

Week 3

- Implement PDF upload, browsing, and processing features
- Integrate OCR for scanned PDFs

Week 4

- Develop hybrid retrieval (FastEmbed, BM25) and embedding cache
- Start building the chat interface

Week 5

- Implement intent detection and conversation memory
- Add support for model selection by user

Week 6

- Integrate answer generation and citation features
- Add reset and process controls to UI

Week 7

- Testing, debugging, and performance optimization

Actors

User

Employees, managers, or interns using the chatbot to consult the project manual. They can ask questions about processes, templates, or project phases.

Admin

Responsible for uploading new versions of the manual, updating the index, and monitoring chatbot performance.

Core Features

Multi-PDF Ingestion with OCR Fallback

Efficiently process diverse document formats, including scanned PDFs, ensuring comprehensive data capture and accessibility.

Hybrid Retrieval Architecture

Combine advanced search techniques (FastEmbed, BM2 reranker) for highly relevant and precise information retrieval.

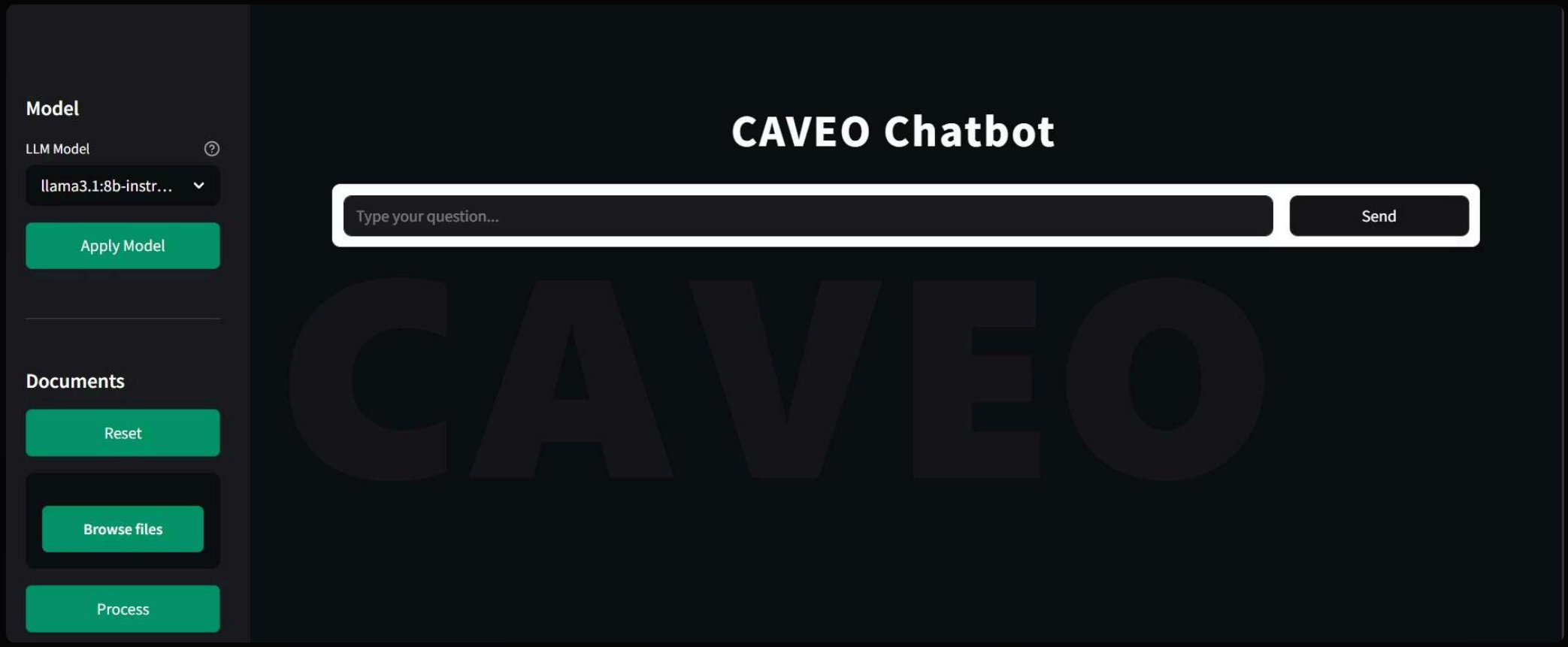
Deterministic Extractors

Accurately identify key entities such as subjects, project phases, and involved actors, providing structured insights from the manual.

Confidence Guardrails

Ensure reliable responses with built-in confidence checks and direct evidence snippets, enhancing trust and accuracy.

Chatbot Interface Overview



Model selection dropdown

Choose between different AI models

Reset button

Clear conversation history and start fresh

Functional & Non-Functional Requirements

Functional Requirements (User & Admin)

- Users can upload and process multiple PDF documents (text or scanned) via the chat interface
- Users can browse and select files to upload
- Users can process selected files to build document embeddings and indexes
- Users can ask questions about the uploaded documents and receive real-time answers
- Users can ask follow-up questions, with the system maintaining conversation context
- Users can reset the chat and document memory at any time
- Users can choose the language model to use for answering questions (if multiple models are available)

Non-Functional Requirements

- The system responds to user queries within 2 seconds under normal load
- The system caches embeddings and frequently asked questions to improve performance
- The system ensures data privacy by processing all data locally and not sending information to external servers
- The system is available 99% of the time (high availability)
- The system is scalable to handle multiple concurrent users and large document sets
- The system is compatible with Windows, macOS, and Linux
- The system is open source and transparent in its operation

Work Environment

Hardware

Development machine minimum requirements:

- CPU: 8 cores (Intel i7 / Ryzen 7 or better)
- RAM: 32 GB (needed for embeddings and caching large documents)
- GPU: NVIDIA RTX with 12 GB VRAM (e.g., RTX 3080, 4070 Ti, A2000 12GB)
- Storage: 10 GB free

These specifications are recommended for running llama3.1:8b-instruct-q4_K_M model, processing large documents, and obtaining answers within 2–3 seconds.

Technology Stack (Roles)

Streamlit

User Interface

Ollama + Llama 3.1

Local LLM

FastEmbed + BM25

Hybrid retrieval

pypdf + Tesseract OCR

Text extraction

Context builder + guardrails

Structured answers + reliability

Technology Justification & Roles

Streamlit – User Interface

Role: Provides the web-based interface for user interactions Why chosen: Rapid development, Python-native, easy deployment, and perfect for data science applications

FastEmbed + BM25 – Hybrid Retrieval

Role: Combines semantic and keyword-based search for optimal document retrieval Why chosen: FastEmbed provides semantic understanding while BM25 ensures keyword precision, together delivering superior search accuracy

Context Builder + Guardrails – Structured Answers

Role: Ensures response quality and provides source citations Why chosen: Prevents hallucinations, maintains answer reliability, and provides transparent source tracking for verification

Ollama + Llama 3.1 – Local LLM

Role: Processes natural language queries and generates human-like responses Why chosen: Runs completely offline, no API costs, privacy-focused, and optimized for local hardware

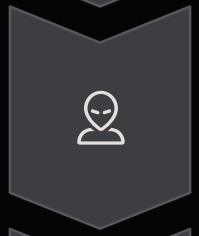
pypdf + Tesseract OCR – Text Extraction

Role: Extracts text from both digital and scanned PDF documents Why chosen: pypdf handles native PDFs efficiently, Tesseract OCR provides fallback for scanned documents, ensuring comprehensive document processing

System Workflow

Document Processing Flow

PDFs



Extract

(pypdf / OCR fallback)

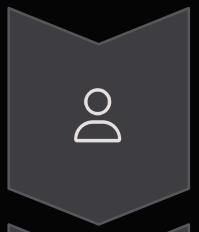
Chunk

FastEmbed + TF-IDF/BM25 Hybrid

with Subject / Phase / Actor extraction (deterministic) running in parallel

Re-rank

Query Processing Flow



User Query

Intent detection

(Enhanced query)

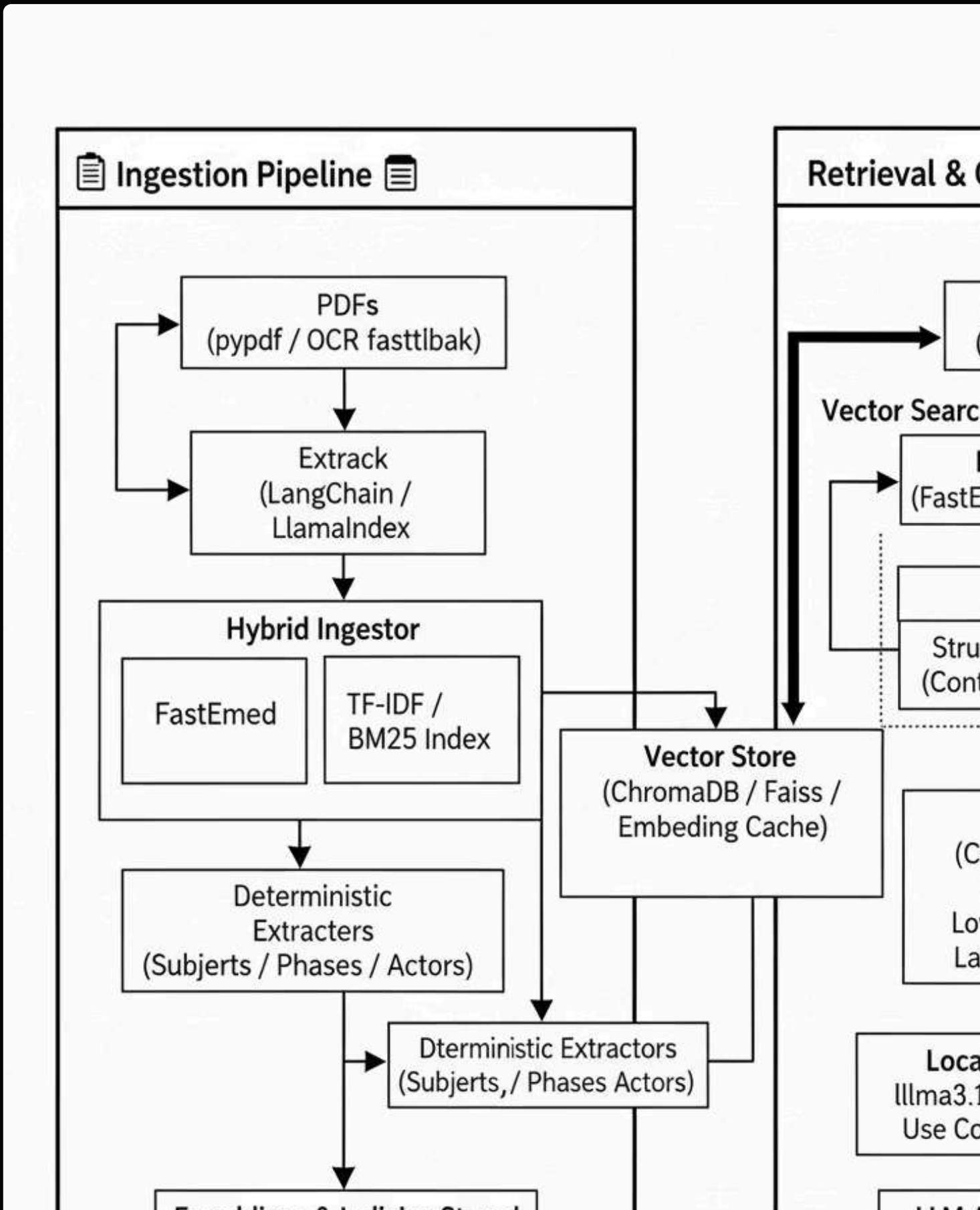
Hybrid Retrieval

Guardrails

LLM answer + citations

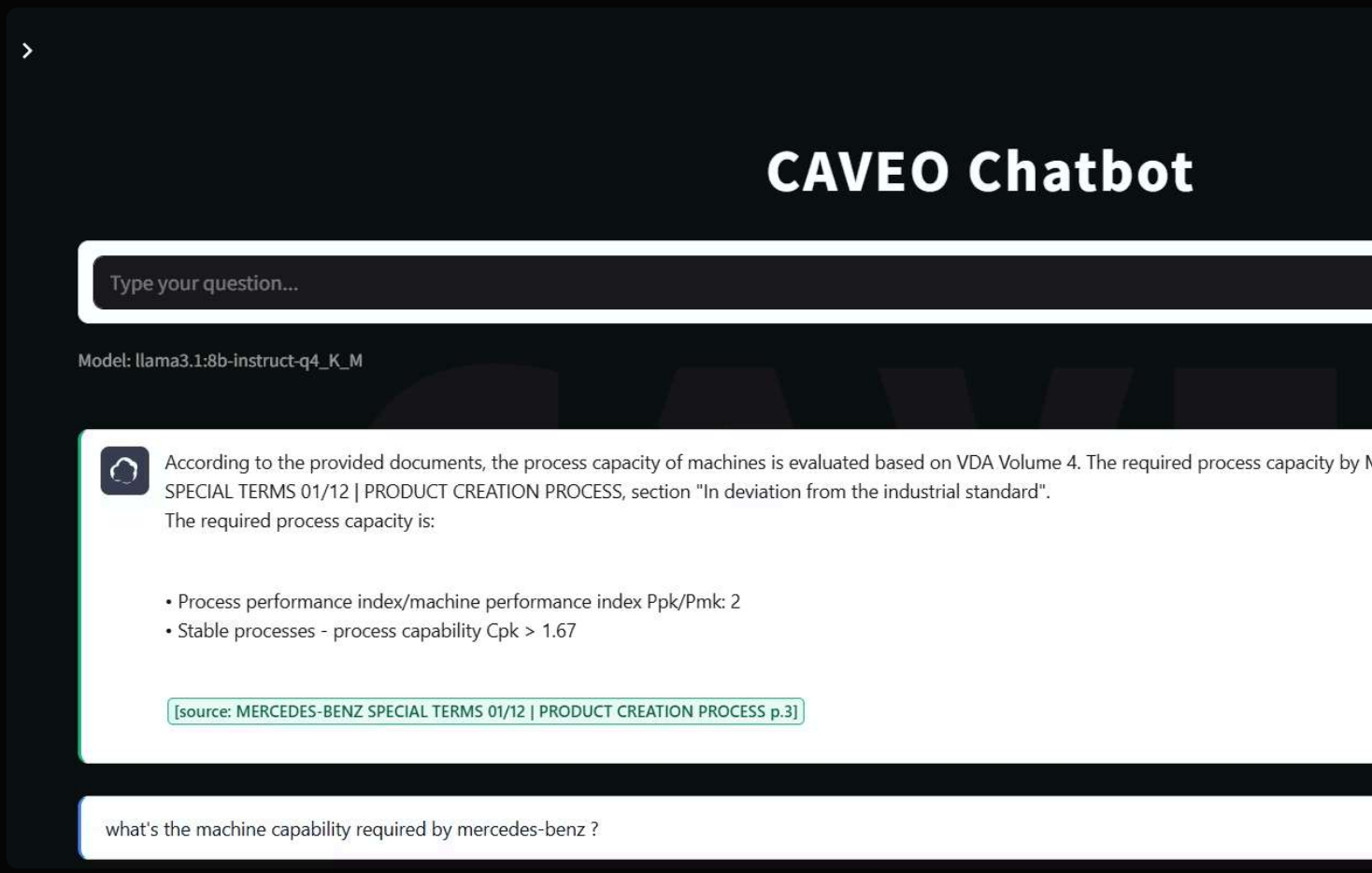
System Architecture

Below is a comprehensive overview of the chatbot's architecture, showing both the document
processing flow, illustrating how different components interact to process documents and res



Evaluation and Testing R

The chatbot underwent rigorous testing to ensure its reliability and effectiveness. The screens
successfully processing a query and delivering a precise answer from the Project Manageme



Testing shows the following performance metrics:

100%

Precision

100%

Accuracy

Source Tracking and Citation

- The chatbot provides precise source citations for all responses
- Users can locate the exact page and section where information originates
- Full transparency in information retrieval with document references

Conclusion

The CAVEO Chatbot represents a successful implementation of local PDF intelligence, delivering secure, offline, and transparent document interaction capabilities. Key achievements include:

Project Success

- Successfully developed a fully functional chatbot using open-source technologies
- Achieved 100% precision, accuracy, and quality in testing phases
- Implemented advanced hybrid retrieval architecture with deterministic extractors
- Delivered comprehensive source tracking and citation capabilities

Technical Excellence

- Robust multi-PDF ingestion with OCR fallback ensures document accessibility
- Hybrid retrieval combining FastEmbed, BM25, and re-ranking provides superior search results
- Local processing maintains data security and privacy
- Streamlit interface offers intuitive user experience

Business Impact

- Significantly improves accessibility to project management resources
- Reduces time spent navigating complex documentation
- Standardizes information consultation across the organization
- Provides scalable foundation for future enhancements

The 8-week development timeline was successfully executed, resulting in a production-ready system that meets all functional and non-functional requirements while maintaining the highest standards of security and transparency.

Future Improvements & Enhance

The CAVEO Chatbot has significant potential for enhancement through advanced multimodal user experience and system capabilities.

Vision Integration

- Implement computer vision models to process images, diagrams, and charts within documents
- Enable users to ask questions about visual content like flowcharts, organizational charts, and technical diagrams
- Support image-based queries and visual document analysis

Speech Recognition & Synthesis

- Integrate Whisper for speech-to-text capabilities, allowing voice-based queries
- Add text-to-speech functionality for audio responses
- Enable hands-free interaction for improved accessibility

Advanced User Experience

- Voice-activated document navigation
- Visual question answering for complex diagrams
- Audio summaries of document sections
- Multilingual speech support

AI Agent Integr

Implement AI agents enabling autonomous adaptive learning fro

These enhancements would transform the chatbot into a truly multimodal assistant, making accessible, and comprehensive while maintaining the core principles of security and offline op

Enhanced Hardware Requirements