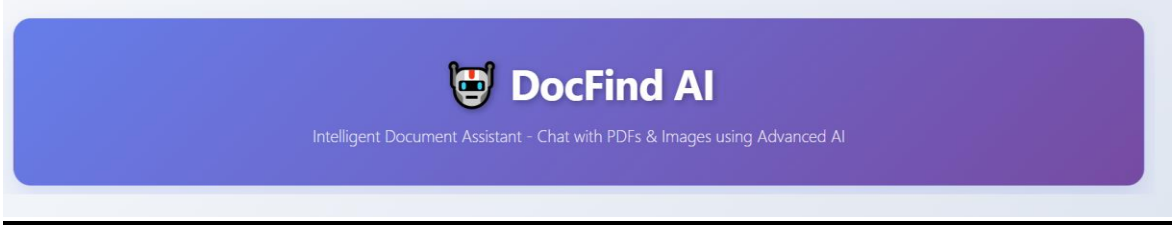


Project Report: DocFind AI – Intelligent Document Assistant

1. Project Title

DocFind AI: Natural Language Query System for Documents Using LLMs and Vector Search

---

2. Objective

To develop a web-based intelligent assistant capable of understanding and interacting with PDF and image-based documents (JPG, PNG) through natural language queries. The application extracts contextual insights, provides document citations, and performs thematic analysis using state-of-the-art AI models.

---

3. Key Features

- Support for multi-document upload (PDF, JPG, PNG)
- AI-powered conversational interface using LLaMA-3 70B
- Context-aware document question answering
- Accurate citation extraction (document name, page, paragraph, extract)
- Thematic summarization of user queries
- Real-time interaction with stream-based LLM response generation
- Professionally styled responsive UI with a modern design aesthetic
- Deployment-ready with Gradio on Hugging Face Spaces

---

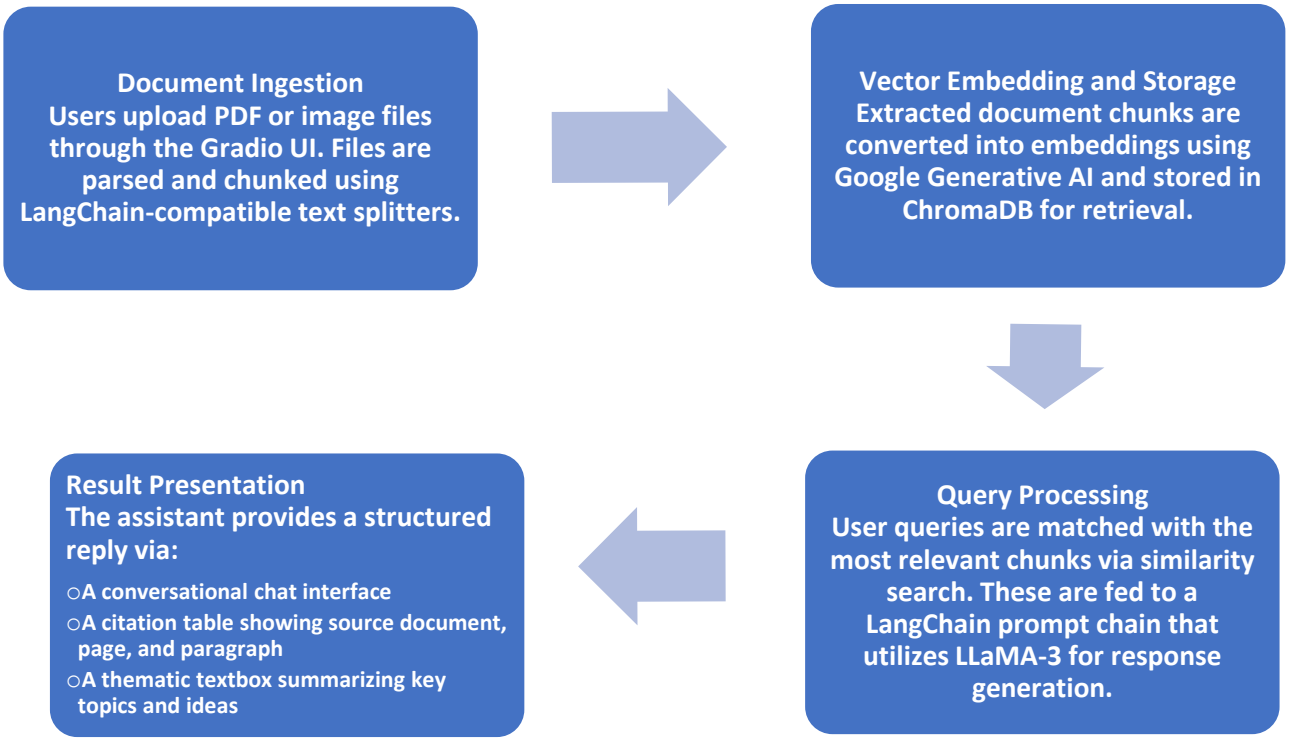
4. Technology Stack

Layer	Technology Used
Frontend UI	Gradio Blocks with Custom CSS
Backend Logic	Python
LLM Integration	LLaMA-3-70B via LangChain
Embeddings	Google Generative AI Embeddings

Layer	Technology Used
Vector Store	ChromaDB
OCR Utility	PyMuPDF, PIL (planned extension)
Deployment	Hugging Face Spaces

---

5. Architecture Overview



---

6. Backend Modules

Module	Description
upload_handler	Handles file parsing, text extraction, chunking, embedding, and vector storage
chat_fn	Accepts user query and chatbot history, performs vector retrieval and invokes the LLM for response

---

7. Model Configuration

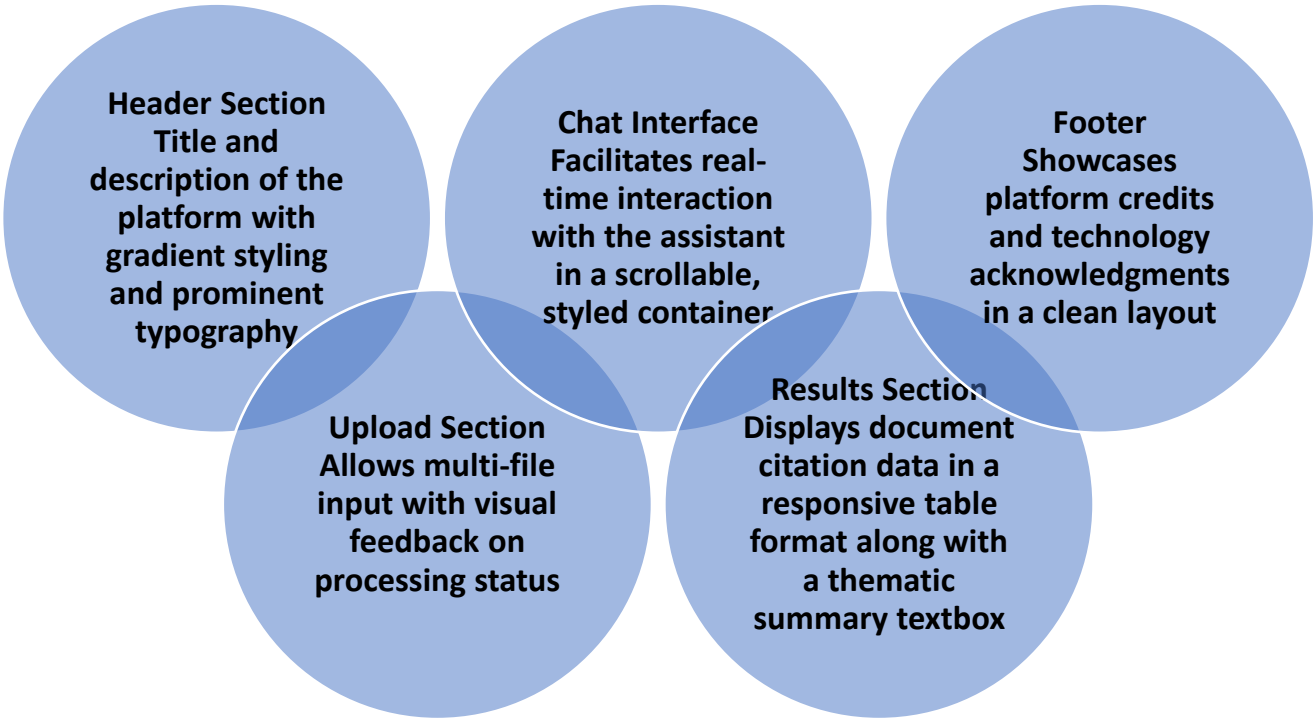
- **Language Model**: LLaMA-3-70B integrated via LangChain Retrieval QA chain
- **Prompt Template:** Context-aware query + retrieved chunk injection
- **Embedding Engine:** HuggingFaceEmbeddings

- **Vector Similarity Search:** FAISS via ChromaDB

---

8. User Interface Design

The application interface is divided into distinct sections:



Custom CSS was used to ensure a modern, minimalistic, and accessible interface with support for responsive layouts and smooth transitions.

---

9. Deployment

- Platform: Hugging Face Spaces
- Interface Framework: Gradio
- Deployment Entry Point: app.py
- Launch Configuration: Auto-launch with debugging enabled, ready for public access

---

10. Future Enhancements

- Multilingual OCR for scanned documents (including Hindi and regional languages)
- Document summarization and section-wise breakdown
- Multi-turn memory-based chat for prolonged context handling

- Downloadable reports with chat and citation logs
- Integration with enterprise knowledge bases and search APIs

---

11. Author & Acknowledgments

Author: Ritwik Chandra
Role: Full-Stack Developer, AI Integrator
Affiliation: AIML Lead, Google Developer Student Clubs | Former IBM Data Analytics Intern

Acknowledgments:
This project utilizes open-source technologies including LangChain, ChromaDB, Gradio, and models from Hugging Face. It is inspired by advancements in Retrieval-Augmented Generation (RAG) systems.