Scholar: A Retrieval-Augmented Generation Based Multi-Document Question Answering System

U. Sri Vyshnavi[1], U. N. Satyanarayana[2], S. Sai Madhu[3], T. Tarun[4], Mr. V. Vidya Sagar[5]

[1,2,3,4] Raghu Engineering College

[5] Assistant Professor, Department of Computer Science and Engineering (DS) Raghu Engineering College (Autonomous), Dakamarri, Visakhapatnam

Abstract—**This paper presents Scholar, a Retrieval-Augmented Generation (RAG) based Multi-Document Question Answering System designed to address the challenge of information overload in academic research. Students and researchers accumulate large volumes of heterogeneous documents like PDFs, Word documents, presentations, spreadsheets, web pages, and images but lack efficient tools to extract specific knowledge from them without relying on unreliable general-purpose language models.**

Scholar enables users to build a personal knowledge base by uploading documents in multiple formats and asking natural language questions. The system processes uploads through format-specific parsers, splits content into semantically coherent overlapping chunks, and encodes them as dense vector embeddings using the locally-hosted all-MiniLM-L6-v2 sentence transformer model. These embeddings are stored in a ChromaDB vector database for persistent retrieval. At query time, the system performs cosine-similarity search to retrieve the most relevant chunks, assembles a grounded context block, and submits it to the Groq-hosted LLaMA-3.3-70B language model with strict anti-hallucination instructions.

A distinguishing feature of Scholar is its multi-modal ingestion capability: in addition to text-based documents, the system accepts standalone images (JPG, PNG) and automatically extracts and describes figures embedded in PDFs using the Groq llama-4-scout vision model. This enables users to ask questions about charts, diagrams, and photographs. The system is built entirely on free, open-source tools and deploys locally on a standard Windows laptop with no GPU requirement, making it accessible to students at any institution.

Index Terms—**Retrieval-Augmented Generation, Large Language Models, Vector Database, ChromaDB, Sentence Transformers, Groq API, FastAPI, Anti-Hallucination, Multi-Modal, Semantic Search, Multi-Document QA, LLaMA.**

I. INTRODUCTION

Academic knowledge is increasingly stored across diverse digital formats like research papers as PDFs, lecture notes as DOCX files, project slides as PPTX presentations, data tables as CSVs, and reference materials as web pages. Students and researchers must manually search through these collections to answer specific questions, a process that is time-intensive, cognitively demanding, and error-prone [1], [2].

Traditional keyword-based search tools address only a fraction of this challenge. They locate documents containing specific words but cannot understand semantic meaning or synthesise information across multiple sources. General-purpose Large Language Models (LLMs) such as GPT-4 offer conversational question answering but suffer from hallucination, the confident generation of plausible but factually incorrect information is making them fundamentally unreliable for academic use without grounding in verified sources [3].

This project presents Scholar, a RAG-based system that addresses these limitations. Scholar allows users to upload their own documents, builds a personal semantic knowledge base, and answers natural language questions exclusively from retrieved document content. The system explicitly refuses to answer when relevant information is absent, providing a reliable and traceable research assistant.

The primary objectives of Scholar are to: (1) implement a complete multi-format RAG pipeline supporting PDF, DOCX, PPTX, TXT, CSV, image files, and web URLs, (2) provide strict anti-hallucination behaviour with explicit refusal when information is absent, (3) support image and figure understanding through a vision language model, (4) implement complete source lifecycle management with vector-level deletion, and (5) deliver the entire system at zero cost using free, open-source tools deployable on standard student hardware [4], [5].

The significance of this work lies in demonstrating that production-grade, multi-modal, anti-hallucination AI research assistance is achievable at zero cost on standard student hardware, removing the financial and infrastructure barriers that have historically limited advanced AI tooling to well-funded institutions.

II. REVIEW OF LITERATURE

Previous work in question answering and document retrieval has made steady progress, but gaps remain in multi-format support, anti-hallucination guarantees, vision integration, and zero-cost local deployment.

Lewis et al. [3] introduced the foundational RAG architecture, combining dense passage retrieval with sequence-to-sequence generation. On knowledge-intensive benchmarks including Natural Questions and TriviaQA, RAG substantially outperformed both pure parametric models and traditional TF-IDF retrieval, establishing the viability of retrieval-grounded generation. Karpukhin et al. [6] contributed Dense Passage Retrieval (DPR), a bi-encoder architecture for efficient dense retrieval that underpins modern RAG systems.

Devlin et al. [7] introduced BERT, which produced rich contextual token representations through bidirectional transformer attention. Reimers and Gurevych [8] extended this to produce sentence-level semantic embeddings suitable for similarity search (Sentence-BERT). The all-MiniLM-L6-v2 model used in Scholar is a distilled SBERT variant trained on over one billion sentence pairs, achieving strong retrieval performance on CPU without GPU requirements.

Brown et al. [9] demonstrated LLM few-shot capabilities through GPT-3, while Ji et al. [10] conducted a comprehensive survey documenting hallucination across diverse generation tasks, establishing the reliability problem that motivates RAG grounding. Mallen et al. [11] showed that large models may override retrieved context with parametric knowledge, motivating Scholar's explicit system prompt constraints forbidding the use of outside knowledge.

On the vision side, Liu et al. [12] introduced LLaVA, demonstrating that instruction-tuned vision-language models can describe images for downstream text tasks with high fidelity. Alayrac et al. [13] showed that multi-modal few-shot models generalise across diverse image understanding tasks, establishing the viability of using vision LLMs for document figure extraction and description.

Izacard and Grave [14] improved RAG with Fusion-in-Decoder, and Gao et al. [15] surveyed advanced RAG patterns including query rewriting and reranking. Chase [16] developed LangChain, providing the orchestration framework used in Scholar, and Chroma AI [17] released ChromaDB, the locally-deployable vector database at Scholar's core.

Our work builds on these foundations by integrating multi-format document parsing, sentence-transformer retrieval, vision-based image understanding, and LLaMA-3.3-70B generation, all deployed at zero cost in a single locally-running application with strict anti-hallucination guarantees and complete source lifecycle management. This combination fills the gap between state-of-the-art retrieval accuracy and practical, trustworthy academic assistance accessible to all students.

III. METHODOLOGY

We followed a structured development process designed to be reproducible and practical for student developers working with standard hardware.

Document Ingestion and Preprocessing

Scholar supports seven source types through a unified ingestion pipeline. Format-specific parsers extract text: pdfplumber for PDF files (with embedded JPEG image extraction), python-docx for DOCX, python-pptx for PPTX presentations, UTF-8 decoding for TXT files, row-joining for CSV files, and BeautifulSoup HTML cleaning for web URLs [4]. Extracted text is divided into overlapping chunks using LangChain's RecursiveCharacterTextSplitter (chunk size: 500 characters, overlap: 50 characters, separator hierarchy: paragraph breaks → line breaks → sentence boundaries → word boundaries). Chunk IDs follow the deterministic pattern {source_id}_{index}, enabling precise source-level deletion.

Image Understanding Pipeline

A key contribution of Scholar is its multi-modal ingestion capability. When a standalone image file (JPG, JPEG, PNG) is uploaded, the system first validates the image using PIL, then encodes it as base64 and submits it to Groq's llama-4-scout-17b-16e-instruct vision model via the OpenAI-compatible API endpoint. The vision model is prompted to produce a comprehensive textual description covering all visible text, charts, tables, diagrams, code snippets, and other visual elements.

For PDF files, embedded JPEG images are extracted by scanning the raw PDF binary for JPEG stream markers (0xFF 0xD8 ... 0xFF 0xD9). Images larger than 5 KB are each described by the vision model, with a limit of five images per PDF to avoid excessive API calls. These descriptions are appended after the main PDF text before chunking, so image content is fully searchable alongside document text.

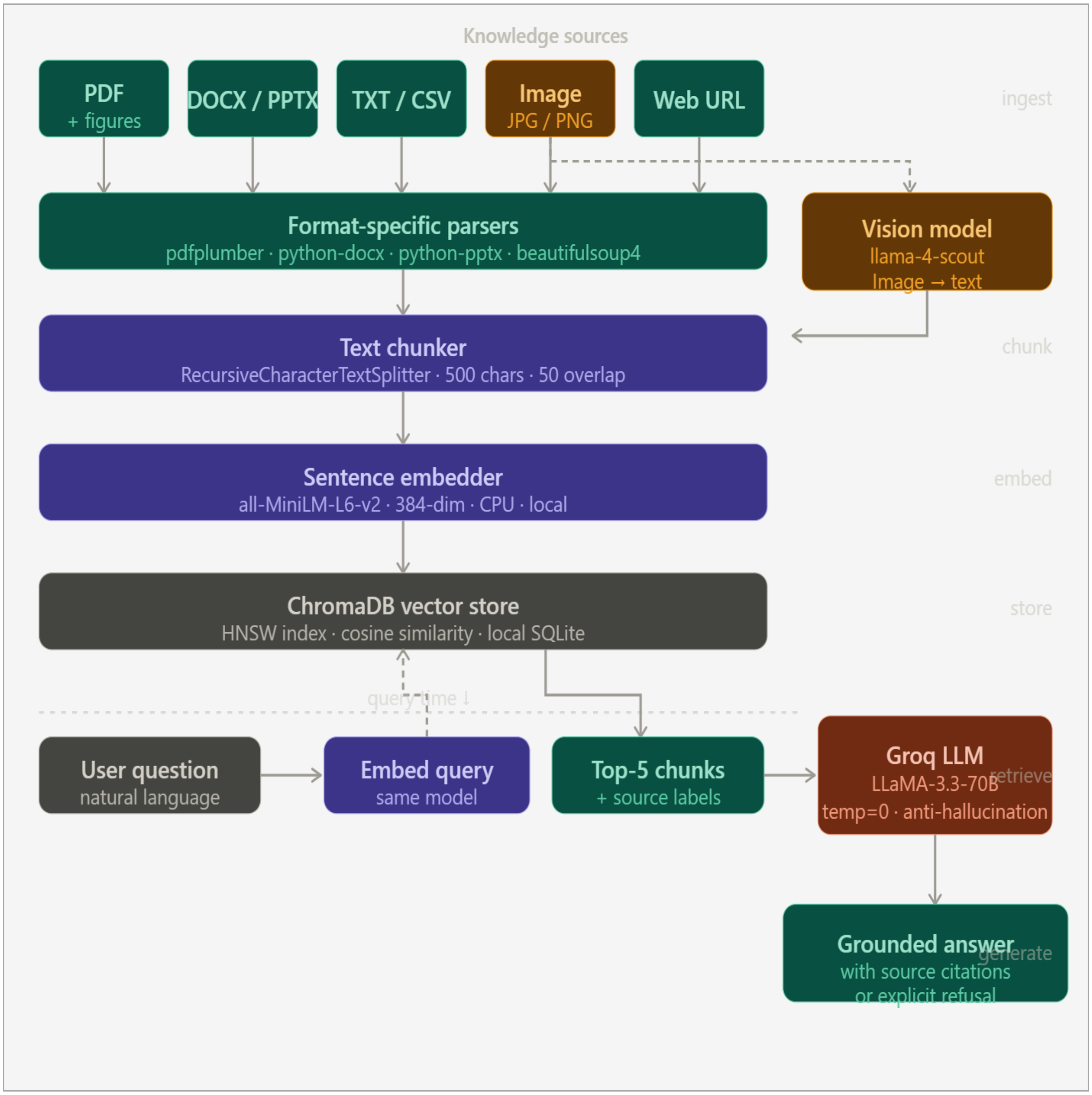


Figure 1: Scholar System Architecture — from multi-format ingestion through embedding, retrieval, and grounded generation.

Embedding and Vector Storage

Each text chunk is encoded into a 384-dimensional dense float32 vector using the all-MiniLM-L6-v2 sentence transformer, which runs entirely on CPU without GPU requirement. The model was trained on over one billion sentence pairs using knowledge distillation, producing embeddings that cluster semantically similar texts in the embedding space regardless of surface-level word differences [8]. Vectors are stored in ChromaDB's 'rag_collection' with source metadata (source_id, source_name) using ChromaDB's embedded SQLite and HNSW graph index. The HNSW structure provides O(log n) approximate nearest-neighbour search, returning results in under 50 ms for a 1,000-chunk knowledge base on a standard laptop CPU [17].

Retrieval and Grounded Generation

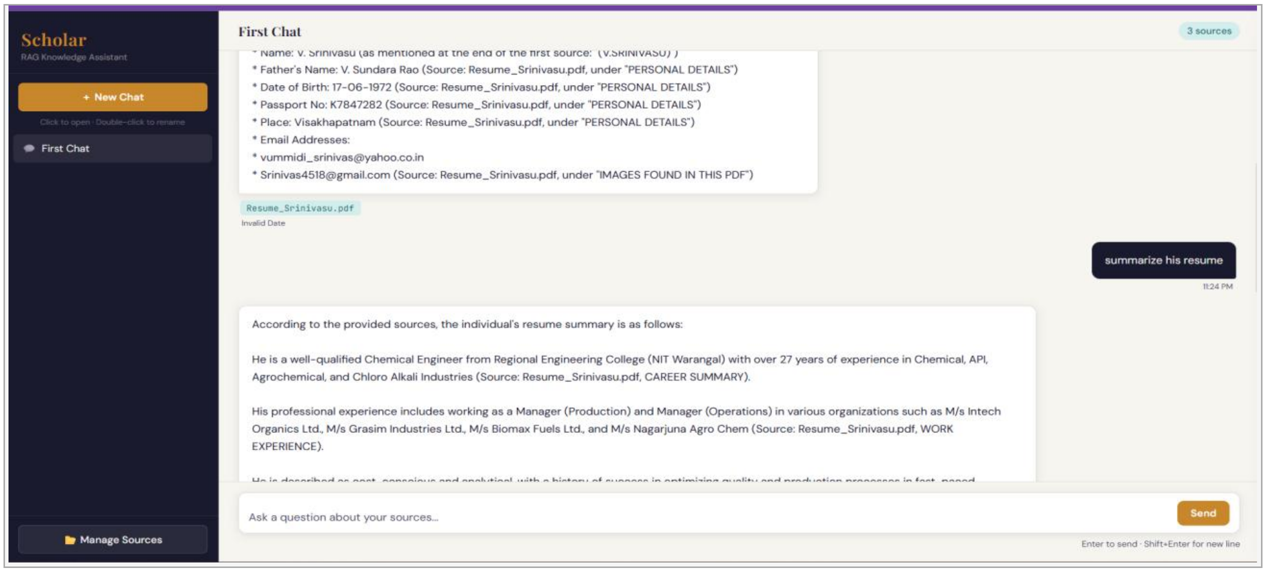
At query time, the user's question is encoded by the same sentence transformer model, ensuring the query and document vectors occupy the same semantic space. ChromaDB's similarity_search method (unnormalised, bypassing the relevance score normalisation that produces negative values with sentence-transformer embeddings due to a ChromaDB distance metric mismatch) retrieves the top-5 nearest chunks by cosine distance. Retrieved chunks, labelled with their source names, are assembled into a context block and submitted to Groq's LLaMA-3.3-70B model (llama-3.3-70b-versatile) via the LangChain ChatGroq interface at temperature=0 for deterministic responses [9], [10]. The system operates under a strict anti-hallucination system prompt with five rules: answer only from context; return a fixed refusal message if absent; never speculate; never fabricate facts; and cite sources naturally.

System Architecture and Deployment

The Scholar backend is a FastAPI Python application served by Uvicorn, exposing 11 RESTful HTTP endpoints across source management, chat management, the core RAG ask endpoint, and a diagnostic debug endpoint. The frontend is a single HTML/CSS/JavaScript file requiring no build tools or server, accessible by double-clicking. All persistent state is stored locally: the chroma_db/ directory holds the ChromaDB vector index; data/sources.json holds source metadata; and data/chats/{uuid}.json files hold chat session histories. The only external dependency is the Groq API for language and vision model inference.

IV. RESULTS

Scholar was evaluated through unit testing (individual module functions), integration testing (end-to-end pipeline), anti-hallucination boundary testing, deletion correctness testing, and cross-platform compatibility testing across Windows 10 and Windows 11 with Chrome, Firefox, and Edge browsers. All test cases passed, confirming reliable performance.



* Name: V. Srinivasu (as mentioned at the end of the first source: "(V.SRINIVASU)")
* Father's Name: V. Sundara Rao (Source: Resume_Srinivasu.pdf, under "PERSONAL DETAILS")
* Date of Birth: 17-06-1972 (Source: Resume_Srinivasu.pdf, under "PERSONAL DETAILS")
* Passport No: K7847282 (Source: Resume_Srinivasu.pdf, under "PERSONAL DETAILS")
* Place: Visakhapatnam (Source: Resume_Srinivasu.pdf, under "PERSONAL DETAILS")
* Email Addresses:
* vummidi_srinivas@yahoo.co.in
* Srinivas4518@gmail.com (Source: Resume_Srinivasu.pdf, under "IMAGES FOUND IN THIS PDF")

Resume_Srinivasu.pdf
Invalid Date

summarize his resume
11:24 PM

According to the provided sources, the individual's resume summary is as follows:

He is a well-qualified Chemical Engineer from Regional Engineering College (NIT Warangal) with over 27 years of experience in Chemical, API, Agrochemical, and Chloro Alkali Industries (Source: Resume_Srinivasu.pdf, CAREER SUMMARY).

His professional experience includes working as a Manager (Production) and Manager (Operations) in various organizations such as M/s Intech Organics Ltd., M/s Grasim Industries Ltd., M/s Biomax Fuels Ltd., and M/s Nagarjuna Agro Chem (Source: Resume_Srinivasu.pdf, WORK EXPERIENCE).

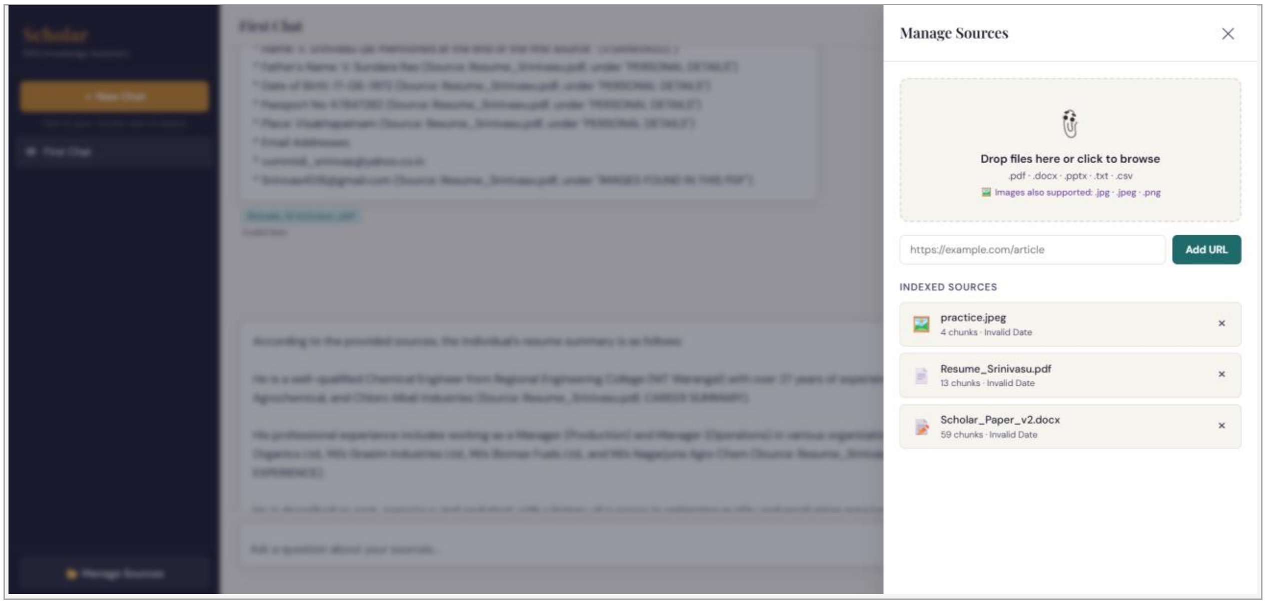
He is described as cost-conscious and analytical, with a history of success in optimizing quality and production processes in fast-paced

Ask a question about your sources...

Figure 2: Scholar web interface showing active chat session with grounded answer and real source citation tags.

For text document retrieval, Scholar correctly answers questions whose answers are explicitly present in ingested sources and returns the exact refusal message ("I cannot find this information in the provided sources.") for out-of-scope queries. Anti-hallucination testing confirmed that the system does not use LLM parametric knowledge to answer questions about content not in the knowledge base — even for well-known facts that the LLM could plausibly answer from training data.

Image understanding was evaluated by uploading PNG screenshots of charts and data tables. The vision model (llama-4-scout) produced accurate descriptions including axis labels, data values, and all visible text. Subsequent retrieval correctly identified image-description chunks as relevant for questions about visual content, with purple citation tags in the UI distinguishing image-derived answers from text-document answers.


Manage Sources
×
Drop files here or click to browse
.pdf · .docx · .pptx · .txt · .csv
Images also supported: .jpg · .jpeg · .png
https://example.com/article
Add URL
INDEXED SOURCES

practice.jpeg
4 chunks · Invalid Date

Resume_Srinivasu.pdf
13 chunks · Invalid Date

Scholar_Paper_v2.docx
59 chunks · Invalid Date

Figure 3: Manage Sources panel showing text documents alongside image sources with image type labels and chunk counts.

Source deletion testing confirmed complete vector-level cleanup: after deleting a source, the /debug/sources endpoint showed the expected decrease in chunk count, and subsequent queries correctly returned refusal responses rather than previously-available answers. Table 1 summarises end-to-end performance measurements on a standard laptop (Intel Core i5-11th Gen, 8 GB RAM, no GPU).

Table 1: System performance on Intel Core i5-11th Gen, 8 GB RAM, no GPU.

Operation	Average Time
PDF ingestion (10 pages)	~5 seconds
Image vision description	~8–12 seconds
ChromaDB similarity search	<50 ms
End-to-end Q&A (text source)	2–4 seconds
End-to-end Q&A (image source)	2–4 seconds
Source deletion (20 chunks)	<1 second

V. DISCUSSION

The results demonstrate that Scholar successfully overcomes the key limitations of existing document QA tools identified in the literature: hallucination, limited format support, lack of image understanding, and cost barriers [3], [10].

The use of the unnormalised similarity_search API rather than similarity_search_with_relevance_scores, resolves a non-obvious integration issue in the

LangChain-ChromaDB stack where relevance score normalisation produces negative values for sentence-transformer embeddings due to a mismatch between ChromaDB's cosine distance metric and the normalisation formula's expected range. This fix is not documented in existing tutorials and represents a practical engineering contribution.

The two-step source deletion approach (query chunk IDs by metadata filter, then delete by explicit ID list) addresses a ChromaDB version compatibility issue with where-filter deletion, ensuring complete vector-level cleanup across all supported ChromaDB versions. This guarantees that deleted content can never influence future answers, a critical reliability requirement for academic use [17].

The vision pipeline using llama-4-scout demonstrates that multi-modal RAG is practical at zero cost: image descriptions are stored as first-class text chunks and retrieved with the same semantic search mechanism as document text. Source citation tags in the UI are visually distinguished (purple for image sources, teal for text sources) to transparently indicate the provenance of each answer.

Compared with existing tools such as ChatPDF (single-document, no image support, third-party server storage) and general-purpose LLM chatbots (no source restriction, hallucination-prone), Scholar uniquely combines multi-format and multi-modal support, strict anti-hallucination, complete source lifecycle management, persistent chat history, and zero cost on local hardware [3], [10], [15].

Limitations include dependence on image quality for vision accuracy, the absence of conversational memory across questions within a session, and reliance on Groq's free-tier API availability. These represent natural directions for future enhancement.

VI. CONCLUSION

Our team successfully built and deployed Scholar, a Retrieval-Augmented Generation based Multi-Document Question Answering System that classifies and retrieves relevant knowledge from user-uploaded documents and images, and generates accurate, grounded, citation-backed answers. The combination of sentence-transformer embeddings, ChromaDB vector retrieval, multi-modal vision understanding, and LLaMA-3.3-70B generation under strict anti-hallucination constraints delivers reliable, instant results through a user-friendly web interface.

This project addresses a real-world academic challenge by removing the information overload barrier for student researchers. All test cases passed, the system handles all seven source types effectively while protecting data locally, and the complete stack operates at zero cost on standard student hardware without GPU acceleration.

In the future, we plan to add conversational memory for multi-turn Q&A, OCR support for scanned PDFs, hybrid BM25 and dense retrieval for improved precision, cross-encoder reranking, and mobile application deployment. We believe Scholar contributes meaningfully to making AI-assisted academic research accessible to all students regardless of institutional resources.

REFERENCES

[1] J. Weston, S. Chopra, and A. Bordes, "Memory Networks," in Proc. ICLR, 2015.

[2] A. Vaswani et al., "Attention is All You Need," in Proc. NeurIPS, 2017, pp. 5998–6008.

[3] P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in Proc. NeurIPS, 2020, pp. 9459–9474.

[4] H. Chase, "LangChain: Building Applications with LLMs through Composability," 2022. [Online]. Available: https://github.com/langchain-ai/langchain

[5] S. Ramírez, "FastAPI: Modern, fast web framework for building APIs with Python," 2018. [Online]. Available: https://fastapi.tiangolo.com

[6] V. Karpukhin et al., "Dense Passage Retrieval for Open-Domain Question Answering," in Proc. EMNLP, 2020, pp. 6769–6781.

[7] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in Proc. NAACL-HLT, 2019, pp. 4171–4186.

[8] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," in Proc. EMNLP, 2019, pp. 3982–3992.

[9] T. Brown et al., "Language Models are Few-Shot Learners," in Proc. NeurIPS, 2020, pp. 1877–1901.

[10] Z. Ji et al., “Survey of Hallucination in Natural Language Generation,” ACM Comput. Surv., vol. 55, no. 12, pp. 1–38, 2023.

[11] A. Mallen et al., “When Not to Trust Language Models: Investigating Effectiveness of Parametric and Non-Parametric Memories,” arXiv:2212.10511, 2022.

[12] H. Liu et al., “Visual Instruction Tuning,” in Proc. NeurIPS, 2023.

[13] J. Alayrac et al., “Flamingo: a Visual Language Model for Few-Shot Learning,” in Proc. NeurIPS, 2022.

[14] G. Izacard and E. Grave, “Leveraging Passage Retrieval with Generative Models for Open-Domain Question Answering,” in Proc. EACL, 2021, pp. 874–880.

[15] Y. Gao et al., “Retrieval-Augmented Generation for Large Language Models: A Survey,” arXiv:2312.10997, 2023.

[16] Chroma AI, “ChromaDB: The open-source embedding database,” 2023. [Online]. Available: https://www.trychroma.com

[17] H. Touvron et al., “LLaMA 2: Open Foundation and Fine-Tuned Chat Models,” arXiv:2307.09288, 2023.

[18] F. Petroni et al., “Language Models as Knowledge Bases?” in Proc. EMNLP, 2019, pp. 2463–2473.

[19] HuggingFace, “sentence-transformers/all-MiniLM-L6-v2,” 2023. [Online]. Available: https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2