ISSN: 2455 - 1597

ijergs
Open Access

International Journal of Engineering Research and Generic Science (IJERGS)
Available online at: https://www.ijergs.in

Volume - 12, Issue - 3, May – June – 2026, Page No. 40 – 45

Personal Finance Advisor Using Retrieval-Augmented Generation (RAG): An AI-Driven Approach to Reliable Financial Guidance

[1]Gagan Chaudhary, Department of Artificial Intelligence & Data Science, IIMT College of Engineering, Greater Noida, Uttar Pradesh, India

[2]Jatin Kumar, Department of Artificial Intelligence & Data Science, IIMT College of Engineering, Greater Noida, Uttar Pradesh, India

[3]Adarsh Tiwari, Department of Artificial Intelligence & Data Science, IIMT College of Engineering, Greater Noida, Uttar Pradesh, India

[4]Mr. Pragya Deep, Assistant Professor, Department of Artificial Intelligence & Data Science, IIMT College of Engineering, Greater Noida, Uttar Pradesh, India

Abstract

Hallucination in AI-generated financial content poses serious risks for individuals seeking reliable guidance. This paper presents a Personal Finance Advisor that integrates Retrieval-Augmented Generation (RAG) with a Large Language Model (LLM) to deliver factually grounded, context-aware financial advice. The system employs a FAISS vector database for semantic retrieval of curated financial documents, Groq-hosted LLaMA for inference, and LangChain for orchestration. A domain-restriction filter, dual-mode interface (Chat Mode and PDF Mode), and mandatory output disclaimers collectively enforce responsible AI behaviour. Implemented with Python 3.10, Streamlit, Sentence Transformers, and FAISS-CPU, the system is evaluated through unit, integration, accuracy, and deployment testing. The RAG-augmented approach achieves an 87.5% accuracy rating on a 40-query financial benchmark, a 25-percentage-point improvement over a non-augmented baseline, validating RAG as an effective mechanism for reducing hallucination in sensitive advisory domains.

Keywords: Retrieval-Augmented Generation, Large Language Models, FAISS, Lang Chain, LLaMA, Sentence Transformers, Hallucination Reduction, Personal Finance, Financial Literacy.

Introduction

Financial literacy is a critical determinant of individual economic well-being, yet a majority of the global population lacks access to personalized and reliable financial guidance[1]. The proliferation of digital platforms has made vast quantities of financial information readily available; however, distinguishing credible advice from misleading content remains a persistent challenge. Incorrect financial guidance can result in poor decision-making, asset misallocation, and long-term economic harm.

Contemporary AI-driven advisory systems typically depend on static pre-trained knowledge, limiting their capacity to provide accurate and verifiable information. This limitation is compounded by hallucination—wherein language models generate plausible-sounding but factually erroneous outputs[2]. In the financial domain, where precision is paramount, hallucination poses an unacceptable risk.

To address this gap, we propose a Personal Finance Advisor leveraging Retrieval-Augmented Generation (RAG)—a paradigm that conditions LLM outputs on dynamically retrieved external knowledge rather than relying solely on parametric memory[3]. By grounding each response in verified financial documents, the system substantially mitigates hallucination. The contributions of this work include: (i) a modular RAG pipeline for financial question answering; (ii) a domain-restriction mechanism preventing off-topic processing; (iii) a dual-mode interface supporting both general queries and user-specific document analysis; and (iv) comprehensive testing validating system accuracy and deployment performance.

Related Work

A. Evolution of AI in Financial Advisory

AI in financial services has progressed from rule-based systems through NLP-powered chatbots to transformer-based LLMs[4]. Early systems were brittle and incapable of handling nuanced queries. Robo-advisors demonstrated algorithmic portfolio management at scale but operated on opaque proprietary algorithms[5]. While GPT and LLaMA family models introduced unprecedented language understanding, their training-data cutoffs and susceptibility to hallucination render them unsuitable for unmediated financial advisory use.

B. Hallucination and RAG

Hallucination in LLMs arises because autoregressive generation maximises predictive probability over token sequences rather than verifying factual correspondence with ground truth[2]. Lewis et al.[3] introduced RAG to address this by conditioning outputs on retrieved external passages. Advances in dense retrieval[6], Sentence Transformers[7], and FAISS[8] collectively make RAG operationally viable at scale.

C. Research Gap

Existing financial AI tools predominantly rely on keyword matching, rule templates, or closed LLM APIs without retrieval augmentation, suffering from knowledge staleness and limited transparency. LangChain [9] and Groq have lowered the barrier to retrieval-augmented pipelines, yet a publicly documented, domain-restricted, document-aware RAG system for personal finance remains absent from the literature. This work fills that gap.

System Architecture

A. Architectural Overview

The system adopts a three-layer architecture comprising the User Interface Layer, Application Layer, and Data Layer. The User Interface Layer, built with Streamlit, provides a browser-accessible conversational interface capturing natural language queries and rendering structured responses.

The Application Layer constitutes the computational core. Upon receiving a query it: (1) embeds the query via Sentence Transformers; (2) performs FAISS nearest-neighbour search for top-k relevant chunks; (3) assembles a structured prompt from retrieved context and a role-specification instruction; and (4) forwards the prompt to the LLM via LangChain. The Data Layer hosts the FAISS index populated from curated financial documents, enabling millisecond-scale semantic retrieval.

B. Technology Stack

Component	Technology
Language	Python 3.10
UI	Streamlit
Orchestration	LangChain
LLM	Groq LLaMA API
Embeddings	Sentence Transformers
Vector DB	FAISS-CPU
PDF Parsing	PyPDFLoader
Env. Mgmt.	python-dotenv
Version Control	Git & GitHub

C. RAG Pipeline

Financial documents are pre-processed offline through noise removal and recursive text-splitting into 500-token chunks with 50-token overlap to preserve cross-boundary context. Each chunk is embedded and stored in the FAISS IVF index. At inference time the query embedding is used to retrieve the top-5 chunks by cosine similarity. A prompt template merges retrieved context with the query; a post-processing step formats the output and appends a mandatory disclaimer clarifying the system's informational nature.

D. Domain Restriction

A lightweight intent-classification heuristic filters queries prior to retrieval. Queries lacking financial terminology or classified as off-domain are rejected with an informative message, preventing misuse and reducing unnecessary load on retrieval and inference subsystems.

Methodology

A. Data Collection and Preprocessing

The knowledge base was constructed from authoritative sources including Reserve Bank of India financial literacy publications [10], government investment guides, and Investopedia resources [11]. Selection criteria were accuracy, recency, and relevance to personal finance. Documents underwent: (i) noise removal eliminating headers, footers, and non-textual artefacts; (ii) sentence-boundary normalisation; and (iii) recursive character-level splitting into 500-token chunks with 50-token overlap.

B. Embedding and Indexing

Text chunks were encoded with the all-MiniLM-L6-v2 Sentence Transformer model[7], producing 384-dimensional dense embeddings. These were indexed in a FAISS IVF structure enabling sub-linear approximate nearest-neighbour queries. The index is serialised to disk for persistent storage and rapid reload.

C. Query Processing Pipeline

User queries follow an eight-step pipeline: (1) domain intent verification; (2) query embedding; (3) FAISS similarity search (top-5 chunks); (4) context assembly and prompt construction; (5) LLM invocation via Groq API; (6) response generation; (7) post-processing and disclaimer insertion; and (8) Streamlit rendering. Chat Mode answers open-ended queries from the pre-built index; PDF Mode dynamically chunks, embeds, and indexes user-uploaded documents for session-specific analysis.

D. Hardware & Software Requirements

Parameter	Specification
Processor	Intel Core i5 / AMD equiv.
RAM	8 GB min (16 GB rec.)
Storage	10 GB SSD (min)
OS	Windows 10 / Linux / macOS
Python	3.10+
Key Libs	Streamlit, LangChain, FAISS, Sentence Transformers

Experiments and Results

A. Testing Methodology

The system was evaluated through four complementary strategies.

Unit Testing examined individual pipeline components in isolation. Embedding correctness was validated via cosine similarity rankings on controlled query-document pairs. FAISS retrieval was assessed on precision-at-k for a curated query set with known relevant chunks. The LLM module was tested on fixed prompts to verify formatting compliance and disclaimer inclusion.

Integration Testing validated end-to-end pipeline coherence from query ingestion to response rendering, with particular attention to context hand-off between retrieval and generation stages.

Accuracy Testing employed 40 financial benchmark queries spanning budgeting, savings, expense management, and investment fundamentals. Responses were rated by a domain expert on a three-point scale: accurate and complete, partially accurate, and inaccurate.

Deployment Testing confirmed responsiveness across desktop and mobile browsers, measuring end-to-end latency under standard load.

B. Results

The RAG-augmented system achieved 87.5% accuracy (35/40 queries rated accurate and complete), compared to 62.5% (25/40) for the same LLM queried without retrieval—a 25-percentage-point improvement attributable to context grounding. The domain-restriction filter rejected all off-domain test queries with zero false negatives. PDF Mode

performed reliably for documents up to approximately 50 pages. Average end-to-end latency was approximately 2.1 seconds: ~0.3 s for embedding and retrieval, ~1.8 s for Groq LLM inference.

Qualitatively, the structured output format—numbered sections with mandatory disclaimers—was rated highly by evaluators for readability and appropriate epistemic signalling. These findings are consistent with the broader RAG literature and confirm that retrieval augmentation is an effective mechanism for reducing hallucination in domain-specific advisory settings.

C. Comparative Summary

Metric	RAG System	LLM Baseline
Accuracy (40-query bench.)	87.5% (35/40)	62.5% (25/40)
Off-domain rejection	100%	N/A
Avg. response latency	~2.1 s	~1.8 s
Hallucination rate	Low	High
Source transparency	Yes	No

Conclusion And Future Work

A. Conclusion

This paper presented a Personal Finance Advisor that operationalises the RAG paradigm to deliver reliable and accessible financial guidance. Tight integration of FAISS-based semantic retrieval with Groq-hosted LLaMA inference through a LangChain pipeline effectively addresses the hallucination problem inherent in conventional LLM chatbots. Empirical evaluation confirms a substantial improvement in response accuracy over non-augmented baselines alongside strong integration, deployment, and domain-restriction performance.

The system demonstrates that transparency, factual grounding, and ethical design constraints can be embedded in a production-ready conversational AI application. Its modular architecture provides a solid foundation for future research in intelligent financial advisory platforms.

B. Future Work

Planned enhancements include: (i) integration of real-time financial data feeds; (ii) multilingual support, particularly for regional Indian languages; (iii) user profile modules for personalised, longitudinal recommendations; (iv) mobile-native application development; and (v) expansion of the knowledge base with regulatory and banking API integration.

C. Individual Contributions

- Gagan Chaudhary — RAG pipeline, embedding generation, LLM integration, documentation, testing.
- Jatin Kumar — Streamlit UI design, system deployment, front-end testing.
- Adarsh Tiwari — Data collection, FAISS implementation, system integration.

Acknowledgment

The authors sincerely thank Mr. Pragya Deep, Assistant Professor, Dept. of AI & DS, IIMT College of Engineering, for his sustained guidance. They also acknowledge IIMT College of Engineering and Dr. A.P.J. Abdul Kalam Technical University, Lucknow, for academic support. This project was undertaken in partial fulfilment of the B.Tech (AI & DS) degree requirements, Session 2025-26.

References

1. OECD, "OECD/INFE 2020 International Survey of Adult Financial Literacy," OECD, Paris, 2020.
2. Z. Ji et al., "Survey of Hallucination in Natural Language Generation," ACM Comput. Surv., vol. 55, no. 12, pp. 1–38, 2023.
3. P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," NeurIPS, vol. 33, pp. 9459–9474, 2020.
4. S. Bhatt, "AI and the Future of Financial Services," J. Financial Regulation, vol. 6, no. 1, pp. 88–100, 2020.
5. T. Baker and B. Dellaert, "Regulating Robo Advice," Iowa Law Review, vol. 103, pp. 713–750, 2018.
6. V. Karpukhin et al., "Dense Passage Retrieval for Open-Domain QA," EMNLP, pp. 6769–6781, 2020.
7. N. Reimers and I. Gurevych, "Sentence-BERT," EMNLP, pp. 3982–3992, 2019.
8. J. Johnson, M. Douze, H. Jégou, "Billion-scale similarity search with GPUs," IEEE Trans. Big Data, vol. 7, no. 3, pp. 535–547, 2021.
9. LangChain, "LangChain: Building Applications with LLMs," 2024. [Online]. Available: https://docs.langchain.com
10. Reserve Bank of India, "Financial Education Resources," 2023. [Online]. Available: https://www.rbi.org.in
11. Investopedia, "Personal Finance Resources," 2024. [Online]. Available: https://www.investopedia.com
12. Groq Inc., "Groq LLaMA API Documentation," 2024. [Online]. Available: https://console.groq.com/docs