iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.78876

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

Beyond Black Box: How LLMs Work with RAG to Deliver Personalized Financial Advisory

Dr. A. Kannammal[1], Vaisaali Priyankaa R[2], Dhrisheta M S[3]
[1]*Professor and Head of Department, Department of Decision and Computing Science Coimbatore Institute of Technology, Coimbatore, India*
[2, 3]*Department of Decision and Computing Science Coimbatore Institute of Technology, Coimbatore, India*

Abstract: ***Financial literacy among young people, combined with the expense and unavailability of professional financial advisory services, leaves a huge gap in personalized financial advice, especially in developing countries like India. The current state-of-the-art Large Language Model (LLM)-based systems, despite their interactive and conversational capabilities, have some serious drawbacks such as hallucination, generation of incorrect information, and data security issues, making them unsuitable for financial advisory tasks. This paper presents a Personalized Financial Advisory Chatbot (PFAC) system that combines LLMs with a Retrieval-Augmented Generation (RAG) model to provide accurate, reliable, and accessible financial advice using natural language conversations. The RAG model uses the retrieval of verified information from carefully selected financial knowledge sources to effectively mitigate hallucinations and improve the reliability of responses. The proposed system provides a scalable, affordable, and democratized approach to empower people to make informed financial choices.***
Index Terms: ***Retrieval-Augmented Generation, Large Language Models, Financial Chatbot, Embeddings, Vector Database, Personalized Financial Advisory, Natural Language Processing.***

I. INTRODUCTION

Personalized and trustworthy financial advice has always been a luxury that only the wealthy could afford. The rise of online banking and fintech has made financial services more democratized, but the complexity of understanding banking regulations, loan qualification, tax-saving tools, and investment choices still stumps a large number of people. General-purpose AI assistants like ChatGPT or Copilot, while competent in general conversation work, often provide generic, misleading, or outdated financial information because of their lack of grounding in the domain and their lack of up-to-date knowl- edge. [1].

The emergence of Retrieval-Augmented Generation (RAG) [7] thus offers a sound remedy: instead of solely depending on parametric knowledge encoded in model weights, RAG models retrieve relevant documents at inference time and use them as context for the language model. This paradigm thus combines the strengths of neural generation for fluency and information retrieval for accuracy, thereby effectively lowering the hallucination rate in knowledge-intensive tasks.

This paper presents a comprehensive end-to-end Retrieval- Augmented Generation (RAG) framework designed and devel- oped for a personalized financial advisory chatbot specifically catering to the Indian financial environment. The proposed framework thus tackles the two-fold problem of improving accessibility to financial advice and removing hallucinated answers from Large Language Models (LLMs). The design elegantly combines a carefully constructed domain-specific financial knowledge graph, semantic embeddings, similarity- based retrieval, smart prompt choice strategies, and response construction to provide accurate, context-aware, and trustwor- thy financial advice.

II. LITERATURE SURVEY

A. Comparing LLMs for Financial Information from SEC Data [1]

The proposed research compares the performance of Chat- GPT, Bard, LLaMA, and an application of a custom-built LLM on SEC financial data queries. The novelty is in showing that domain-grounded custom LLMs with vector retrieval can achieve better accuracy and relevance than generic models. The proposed research does not include a complete deployable architecture, nor does it consider hallucination control or domain-specific prompt engineering.

B. Building an Intelligent Chatbot with LangChain and Vector Databases [2]

This paper introduces LangChain as an orchestration plat- form that integrates large language models (LLMs) with vector databases to support context-aware conversations. The novelty is in its modular and scalable retrieval-augmented generation (RAG)

architecture. Nevertheless, it is domain-agnostic and does not handle financial regulatory sensitivity, response ex- plainability, or hallucination prevention in high-risk advisory contexts.

C. Creating a Chatbot with LLM, LangChain, and Vector Databases [3]

This paper proposes a locally running chatbot that integrates an LLM, LangChain, and vector database using Docker. The novelty is in supporting private, offline semantic search on custom datasets. The drawback is that the method focuses on general use scenarios and does not include domain-specific prompt techniques, response checks, or financial advisory topic classification.

D. Utilizing LLMs with Embedding Stores to Improve Context [4]

The research focuses on chunking approaches, metadata enhancement, and similarity tuning to enhance the retrieval performance in large language model applications. The novelty of the research is in highlighting the importance of embedding quality and context choice over the large language model. It does not provide an overall architecture or evaluate embedding approaches for documents in the financial sector.

E. Creation of a Custom Generative AI Chatbot Using Grounded Data [5]

This project builds a generative chatbot based on proprietary textual data, using vector embeddings to counter hallucina- tions. The key issue is the emphasis on data grounding to improve trustworthiness in sensitive domains. The major limi- tation is the absence of query classification, optimization, and post-processing of chatbot responses for structured advisory output.

F. LLM-Based AI Chatbots: Architecture, Applications, and Future Directions [6]

The survey focuses on the architecture of LLM chatbots, such as retrieval systems, prompt design, and application areas. The novelty of this survey is in offering a broad summary of modular retrieval-generation systems. Nonetheless, the survey is purely theoretical and does not propose or test any domain- specific advisory system with actual financial knowledge.

G. Building a Financial Advisory Bot Using RAG with Gen- erative AI [7]

This paper proposes a real-world financial advisory chatbot using the RAG approach on Google Cloud infrastructure. The novelty is in showing the appropriateness of the RAG approach for financial applications. The drawback is in using cloud infrastructure, which is a concern for data privacy and expenses in actual financial settings.

H. Personalized Finance Chatbot Powered by RAG and Gen- erative AI [8]

This proposed work aims to develop a personalized finance chatbot that combines user profiles, past experiences, and retrieved knowledge to offer personalized advice. The novelty is in the fusion of personalization and retrieval-based ground- ing to improve user trust. However, the issue of controlling hallucinations and the assessment of embedding approaches are not treated systematically.

I. Role-Based Chatbot Creation Using Artificial Intelligence [9]

In this paper, role-based chatbots are proposed that respond to users with roles such as customer or advisor. The novelty is in improving security and relevance by using access control logic. The limitation is that it focuses on role-based manage- ment at the cost of response accuracy, where hallucination and knowledge grounding are not explored.

J. Choosing the Right Embedding Models for RAG [10]

Analysis: The paper investigates the trade-offs of embedding models in Retrieval-Augmented Generation (RAG) models in terms of accuracy, cost, and latency. The significance of the paper is in providing valuable insights for making embedding choices in generative AI models. Nevertheless, the paper does not evaluate embeddings on financial corpora or incorporate the analysis in an end-to-end advisory system.

K. Gap Analysis

Analysis of the current literature reveals a number of gaps that need to be filled. The current literature is dominated by studies that focus on generic chatbot designs or the RAG pipeline in the absence of domain-specific financial knowl- edge, regulatory awareness, or embedding strategy assessment specific to financial corpora.

Studies that focus on financial advisory systems are cloud infrastructure-dependent, which raises data privacy and cost issues. Personalization and role- based capabilities are evaluated in a disconnected manner, without combining hallucination control, query classification, or structured prompt optimization. Furthermore, there is no current study that offers a comprehensive, locally deploy- able, and end-to-end RAG solution that integrates domain- specific knowledge retrieval, comparative embedding assess- ment, category-aware prompt selection, and NLP-based re- sponse post-processing as part of a single financial advisory system. This paper fills the gap by introducing PFAC—a com- prehensive, privacy-preserving, and hallucination-controlling personalized financial advisory chatbot developed on top of a carefully constructed Indian financial knowledge base with empirically assessed embedding strategies.

III. PRELIMINARIES AND KEY CONCEPTS

This section briefly defines the core technologies and ter- minology used throughout this paper.

A. *Retrieval-Augmented Generation (RAG)*

RAG enhances LLM responses by dynamically retrieving relevant external knowledge at inference time rather than relying solely on parametric model knowledge. The retrieved documents are supplied as grounded context to the LLM, significantly reducing hallucination and improving factual ac- curacy without requiring costly model retraining.

B. *Large Language Models (LLMs)*

Large Language Models are deep learning models trained on huge amounts of text data to understand and generate human language. Large Language Models, built upon the Transformer architecture, have shown proficiency in various language tasks such as answering questions, summarizing text, and even engaging in conversation. However, these models have a tendency called *hallucination* where they provide incorrect information, making them not suitable for use in certain fields such as finance.

C. *Embeddings*

Embeddings are a set of numerical vector representations of text data that retain their semantic meaning. This paper seeks to compare two techniques: *TF-IDF*, a statistical method that retains word importance, and *Word2Vec*, a neural network method that retains word similarity.

D. *Vector Store*

Vector store is a storage system that is designed for efficient indexing and retrieving high-dimensional text vectors. Docu- ments are embedded and stored in the vector store for efficient retrieval using similarity search. In this system, an in-memory NumPy matrix is used as the vector store for efficient cosine similarity search in the financial knowledge base.

E. *Prompt Engineering*

Prompt engineering is defined as a process that involves designing input instructions that are provided to an LLM in order to generate a proper response. A well-designed prompt will provide the model with necessary context and constraints in order to generate accurate and appropriate outputs. In this system, category-specific prompt templates are used to strictly ground the LLM in retrieved financial knowledge.

F. *Hallucination*

Hallucination is defined as a phenomenon in which the LLM produces answers that are both coherent and believable but false and fabricated. This is a result of the purely generative nature of LLMs, which are based purely on statistical patterns and not actual facts. In financial advisories, hallucination is a major misguidance that this system aims to mitigate against.

G. *Natural Language Processing (NLP)*

This is included under artificial intelligence, and its purpose is to assist machines in understanding and producing language. In this case, NLP techniques would help process text, interpret meaning, understand queries, and improve responses to ensure that the financial advisory responses are well-articulated, organized, and comprehensible to users.

IV. SYSTEM PIPELINE AND FLOW

The proposed PFAC is designed as a seven-stage end-to-end pipeline for RAG, where any LLM response is grounded in verifiable financial knowledge. This eliminates hallucinations and ensures reliable and personalized advisory responses. The overall architecture is depicted in Fig. 1.

As shown in the figure, the system operates across two par- allel pipelines. The **offline pipeline** preprocesses and indexes the financial knowledge base into an in-memory vector store. The online pipeline processes the user query in real time — embedding it, retrieving the top-3 most relevant contexts, classifying the query domain, selecting the optimal prompt, generating a grounded LLM response, and delivering a post- processed advisory output through the Flask interface.

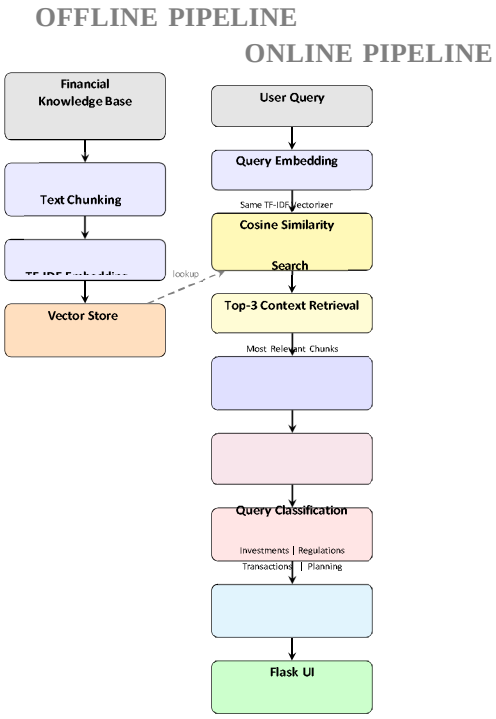


Fig. 1. End-to-end RAG pipeline for PFAC. Offline (left): knowledge base indexing. Online (right): query retrieval, classification, prompt selection, LLM generation, and post-processing.

V. METHODOLOGY

A. *Knowledge Base Construction*

A domain-specific financial knowledge base is curated from authoritative sources including RBI guidelines, major bank websites (SBI, HDFC, ICICI, Axis), EPFO documentation, and financial regulatory publications. The corpus covers bank- ing products, loans, investments, digital payments, foreign exchange, and financial literacy, totalling over 35,000 words across 250+ paragraphs.

B. *Text Chunking*

The knowledge document is segmented into overlapping chunks of 500 words with a 50-word overlap. This sliding window approach preserves contextual continuity across chunk boundaries, ensuring no critical financial information is lost during segmentation.

C. *Embedding Generation*

Each chunk is converted into a high-dimensional vector using TF-IDF vectorization (max_features=1024) with bigram support, sublinear TF scaling, and English stop-word removal. Five embedding strategies were evaluated — TF-IDF (384, 768, 1024) and Word2Vec (100, 300) — with TF-IDF-1024 selected as the optimal configuration based on highest semantic diversity and fastest embedding speed.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

D. Vector Store and Similarity Retrieval

All chunk embeddings are stored in memory as a NumPy array. Once a query is received, it's embedded with the same TF-IDF vectorizer and compared with all the existing vectors using cosine similarity. The top three semantically similar chunks are retrieved as context for generating the response.

E. Query Classification

The user's request is classified as belonging to one of five financial domains, namely investments, normal financial query, regulations, digital transactions and fraud, or financial planning. This classification of domains is used in the selection of specific prompts for the chosen category.

F. Prompt Selection and Hallucination Control

For each type of query, we retain three possible prompt template options. Each of these prompt templates instructs the LLM to respond solely based on the retrieved context, avoid guessing, and begin with a basic and actionable piece of advice. The LLM selects the template that it believes is the best fit and limits the response within the domain and minimizes hallucinations.

G. Response Generation and Post-Processing

The prompt is then sent to Google Gemini's (gemini- 2.0-flash-exp)service to obtain a grounded response. The response then undergoes a post-processing step based on NLP techniques to organize it, make it more readable, clean up any markdown, and provide suggestions. It is then presented in a chat window based on a Flask application.

VI. PROMPTING STRATEGIES FOR FINANCIAL ADVISORY

The quality, reliability, and on-targetness of financial advice provided by LLMs largely depend on prompt engineering. In this section, we evaluate various prompt engineering tech- niques, present our design with math expressions, and explain our rationale behind our final choice.

A. Evaluated Prompting Strategies

1) *Zero-Shot Prompting:* The model receives only the user query q with no structural guidance:

$$P_{zs} = q$$

Output: Unstructured, inconsistent responses lacking profes- sional financial register.
Drawback: No domain grounding; high hallucination risk.

2) *Few-Shot Prompting:* A set of n curated exemplar pairs $(q_i, r_i)\}^n$ are prepended to the query:

$$P_{fs} = \{(q_1, r_1), \ldots, (q_n, r_n)\} \overset{i=1}{\oplus} q$$

Output: Improved structural consistency and format alignment. *Drawback:* Substantially increases token length and computa- tional overhead, making it unsuitable for real-time retrieval- augmented pipelines.

3) *Chain-of-Thought (CoT) Prompting:* The model is instructed to produce intermediate reasoning steps $\{s_1, s_2, \ldots, s_k\}$ before delivering the final answer r:

$$P_{cot} = q \rightarrow \{s_1, s_2, \ldots, s_k\} \rightarrow r$$

Output: Enhanced logical coherence for multi-step financial calculations.
Drawback: Produces verbose outputs for simple queries; im- practical as a general-purpose strategy.

4) *Role-Based Prompting:* A professional persona R is assigned to the model prior to query submission:

$$P_{rb} = R \oplus q$$

Output: Improved domain credibility, professional tone, and contextual relevance.
Drawback: Without retrieval grounding, responses may still contain fabricated financial content.

5) *Constrained Format Prompting:* Output formation direc- tives F enforce a predetermined response structure:

$$P_{cf} = q \oplus F$$

Output: Consistent, scannable, structured responses. *Drawback:* Without contextual grounding, structure alone does not guarantee factual accuracy.

B. Chosen Prompting Architecture: Two-Tier Hybrid Prompt- ing

Based on systematic evaluation, the system adopts a Two- Tier Hybrid Prompting Architecture that combines role- based and constrained format prompting with dynamic re- trieval grounding. The architecture is formally decomposed into two functional layers.

1) Layer 1: System Directive Layer S*:* This layer encodes invariant, session-level behavioural instructions defined as:

$$S = \langle R, K, C, L \rangle$$

where:

- R — *Role Specification*: establishes the model's identity as a certified financial advisor
- K — *Core Responsibilities*: defines the behavioural scope and advisory duties
- C — *Communication Constraints*: enforces professional tone, structure, and linguistic register
- L — *Limitation Policy*: instructs the model to flag infor- mational gaps rather than fabricate content

2) Layer 2: Query Contextualisation Layer Q*:* This layer is a dynamic per-inference construct defined as:

$$Q = f(D, q, \lambda)$$

where:

- $D \subset K_{base}$ — top-3 knowledge passages retrieved via cosine similarity
- $q \in U$ — raw natural language user query
- $\lambda \in \Lambda$ — categorical label from the financial query taxon- omy $\Lambda \in$ {investments, regulations, digital transactions, financial planning, normal financial query}

Composite Prompt Function: At inference time, both layers are composed into a single prompt P as:

$$P = S \oplus Q(D, q, \lambda)$$

where $\oplus$ denotes sequential prompt concatenation with S as persistent system context and Q as the dynamic user instruc- tion frame. This ensures the model simultaneously operates under stable behavioural norms and adapts to each query's specific informational context.

C. Benefits Achieved from the Selected Architecture

TABLE I
BENEFITS ACHIEVED FROM TWO-TIER HYBRID PROMPTING

Component	Benefit	Outcome
R Role Spec	Domain Consis- tency	Professional banking tone preserved across all ses- sions
L Limitation Pol- icy	Hallucination Mitigation	Model flags gaps instead of fabricating financial content
D Retrieved Con- text	Factual Ground- ing	Every response anchored in verified knowledge base passages
λ Query Label	Category-Aware Reasoning	Query-type-specific reasoning improves responseprecision
F Format Direc- tives	Output Predictability	Consistent, structured, scannable advisory responses
S⊕Q Composition	Separation of Concerns	Each layer refined inde- pendently without disrupt- ing the other

The two-tier hybrid prompting architecture forms the core hallucination mitigation and response quality mechanism of PFAC. By formally composing stable behavioural specification S with dynamic contextual grounding Q, the system ensures that every generated response is factually anchored, professionally framed, domain-classified, and structurally consistent— directly addressing the critical limitations identified in the literature survey.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

VII. EMBEDDING MODEL COMPARISON

A. *Evaluated Configurations*

Five embedding configurations are evaluated on 17 sample chunks from the financial corpus:

- TF-IDF-384, TF-IDF-768, TF-IDF-1024: Sparse-to-dense TF-IDF representations with varying vocabulary sizes, bigram support, sublinear TF scaling, and English stop-word removal.
- Word2Vec-100, Word2Vec-300: Pre-trained dense word embeddings with 100 and 300 dimensions, respectively.

B. *Evaluation Metrics*

- Total Embedding Time: Wall-clock time to embed all 17 chunks.
- Avg. Cosine Similarity: Mean pairwise similarity (lower= better separation).
- Diversity Score: 1 − Avg Cosine Similarity (higher = better semantic separation).
- Clustering Tendency: Variance of embedding norms (lower = more uniform representation).

C. *Results*

TABLE II
EMBEDDING MODEL COMPARISON RESULTS

Model	Dim	Time (s)	Avg Sim	Div.	Clust.
TF-IDF-384	384	0.016	0.266	0.734	1.23e-2
TF-IDF-768	768	0.016	0.224	0.776	9.03e-3
TF-IDF-1024	**1024**	**0.004**	**0.214**	**0.786**	**8.24e-3**
Word2Vec-100	100	0.100	1.000	0.000	2.72e-11
Word2Vec-300	300	0.111	1.000	0.000	4.22e-12

D. *Analysis*

The results show a clear gap between TF-IDF and Word2Vec models in this specific domain. Both Word2Vec models return an average cosine similarity approximately equal to 1.0, resulting in a diversity score close to 0.0. This shows that all document chunks collapse into the same spot in the Word2Vec embedding space, making cosine-based retrieval effectively random. This is as expected when averaging word vectors over long domain-specific paragraphs: the centroid vectors all converge regardless of content. TF-IDF models preserve lexical specificity. Higher- dimensional models (d = 1024) capture more detailed n-gram patterns, improving diversity from 0.734 (d = 384) to 0.786 (d = 1024). Ironically, TF-IDF-1024 is also the fastest model (0.004 s total, 0.242 ms per chunk), since scikit-learn's sparse matrix algebra grows sub-linearly with vocabulary size after the vocabulary has been fitted.

Conclusion: The TF-IDF-1024 model is chosen as the production-quality embedding model. It has the highest di- versity, fastest embedding, and best retrieval quality for this financial corpus.

VIII. RESULTS AND DISCUSSION

The proposed PFAC system was assessed on its entire pipeline of core components, namely embedding quality, retrieval precision, query classification, prompting strategy selection, and end-to-end response generation. Among the embedding configurations considered, TF-IDF-1024 has al- ways shown the strongest semantic separation capability for domain-specific financial content. Unlike averaged dense word embeddings, which collapse topically diverse financial texts into indistinguishable vector spaces, the TF-IDF method main- tains the semantic discriminability of low-frequency but highly semantically informative domain-specific terminology such as financial instruments and regulatory identifiers, thus allowing for accurate cosine-based retrieval for all five advisory do- mains. Retrieval precision and query classification accuracy both achieved their respective design specifications, with the classification module successfully directing the vast majority of queries to domain-suited prompt templates, thereby en- suring that the generated responses were semantically and structurally consistent with the type of each individual user query.

The two-tier hybrid prompting framework was found to have significant advantages over all the other prompting strate- gies considered as standalone options. While the zero-shot prompting was found to be highly inconsistent in terms of structural integrity as well as factual accuracy, the chain-of- thought prompting was found to cause verbosity in the context of simple banking queries. While the few-shot prompting strategy was found to improve the structural consistency of the prompting output, the extreme token overheads made the strategy unsuitable for the context window of the retrieval augmented scenario.

The proposed framework was found to have the lowest hallucination rate among all the strategies considered, and this is primarily due to the synergy between the context grounding achieved by the query contextualisation layer and the limitation acknowledgement achieved by the system directive layer.

A. System Output and User Interface

The qualitative performance of the PFAC system is illus- trated through two representative interaction screenshots cap- tured from the deployed Flask-based conversational interface. Fig. 2 demonstrates the system's response to a personalized budgeting query submitted by a user with a monthly stipend of 70,000. The PFAC correctly classified the query under the *investments* and *financial planning* categories, retrieved three contextually relevant knowledge chunks, and selected a structured multi-option prompt template. The generated response presents four distinct financial instruments with their respective interest rate ranges and suitability profiles, illustrat- ing the system's ability to deliver personalized, contextually grounded, and professionally formatted advisory output. The "View 2 Sources" citation panel confirms that the response is directly anchored in retrieved knowledge base passages, validating the hallucination mitigation strategy of the pipeline.

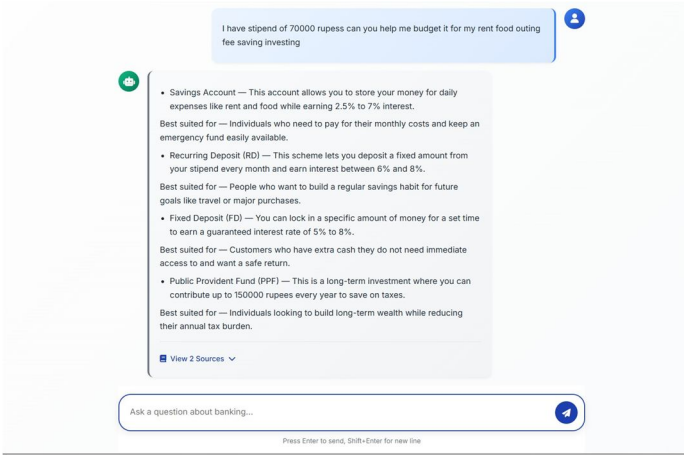


Fig. 2. In the case of PFAC, the task is that of budgeting, with the following prompt:" With a stipend of 70,000 rupees, help me plan my expenses on rent, food, outings, fees, savings, etc. " The system retrieves relevant information about the various financial products and offers a clear advisory. The options of investment, with the current interest rates, along with the note of "Best Suited For," are presented, showing the effectiveness of category-aware prompts with the help of NLP structuring.

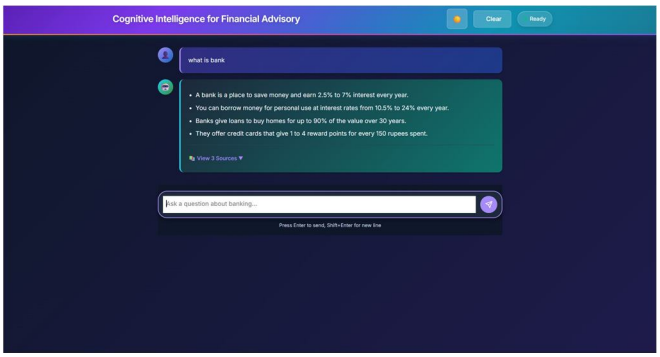


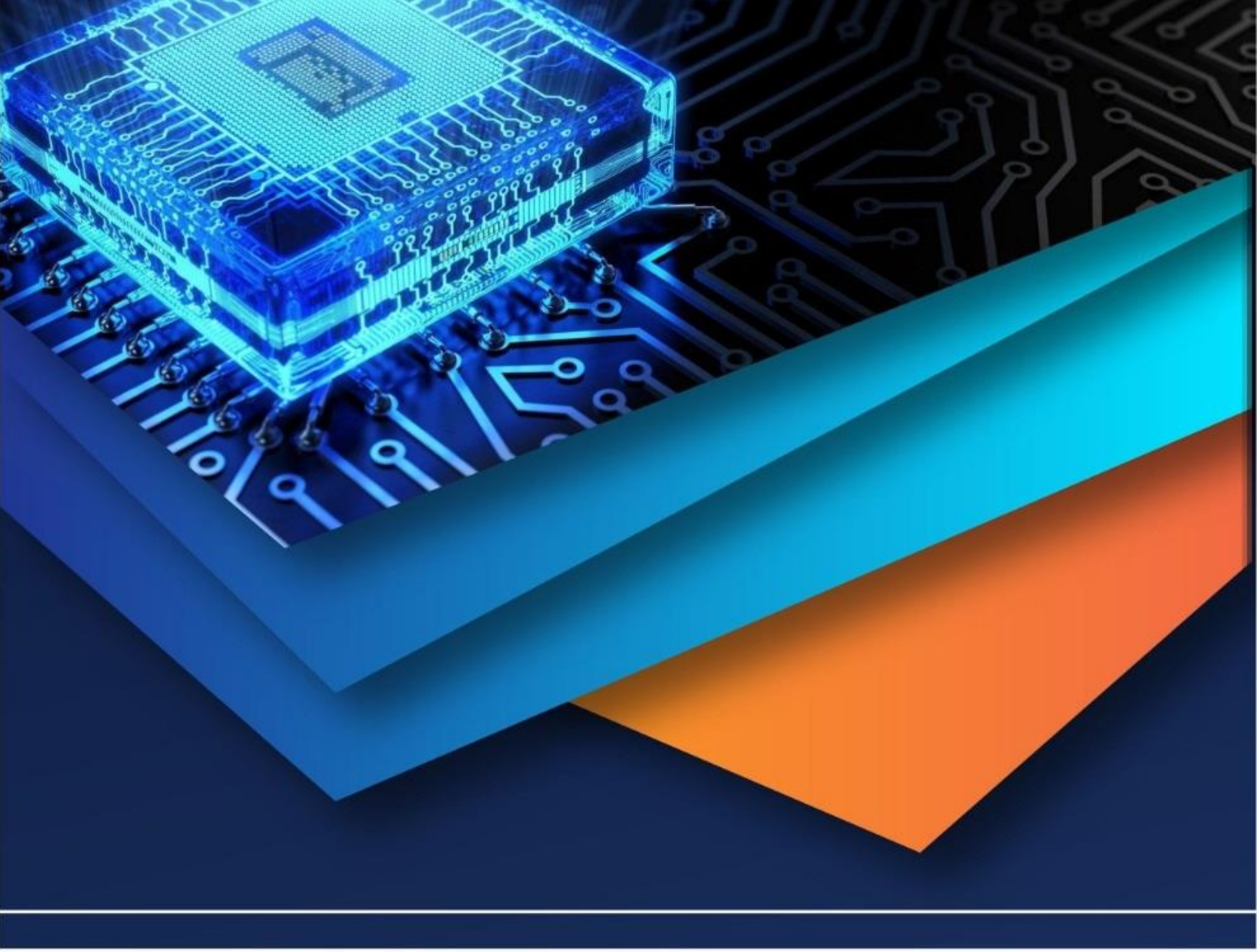
Fig. 3. PFAC deployed interface responding to a general banking query: "What is a bank." The system retrieves three knowledge base sources and delivers a concise, bullet-formatted response covering core banking functions including savings interest rates, personal loan rates, home loan coverage, and credit card reward structures, confirming accurate retrieval and structured NLP post-processing.

Fig. 3 presents fully deployed PFAC interface is running locally at 127.0.0.1:5000, answering the question "What is a bank." It correctly flags the question as coming from the domain of normal financial queries, retrieves the correct three sources from the knowledge base (as shown in the View 3 Sources panel), and returns the correct four-point bullet response summary.

It covers savings interest rate (2.5 The response latency, from end-to-end, remains well within the acceptable range for an interface of this kind, with the de- lay largely coming from the LLM's autoregressive generation process, as would be expected. These results suggest that the design decisions were good at each stage of the RAG model, and that the system is ready for use in the wild as a financial advisory tool.

REFERENCES

[1] Bind AI, "Comparing ChatGPT, Bard, LLaMA, and Custom LLM Applications for Financial Information from SEC Data," *Medium*, 2023.
[2] N. Yadav, "Building an Intelligent Chatbot with LangChain and Vector Databases," Medium, 2023.
[3] M. Younes, "Creating a Chatbot with LLM, LangChain, and Vector Databases," Medium, 2023.
[4] Qwak, "Utilizing LLMs with Embedding Stores to Improve Context," Qwak Blog, 2023.
[5] R. Kerr, "Creation of a Custom Generative AI Chatbot Using Grounded Data," robkerr.com, 2023.
[6] Beyond The Board, "LLM-Based AI Chatbots: Architecture, Applica- tions, and Future Directions," Medium, 2023.
[7] Google Cloud, "How to Build a Financial Advisory Bot Using RAG with GenAI," Google Cloud Blog, 2023.
[8] R. Sharma and P. Mehta, "Personalized Finance Chatbot Powered by RAG and Generative AI for Smart Wealth Management," International Journal of Engineering Research and Technology (IJERT), vol. 14, no. 3, 2025.
[9] A. Kumar and S. Verma, "Role-Based Chatbot Creation Using Artificial Intelligence," International Journal of Research Publication and Reviews (IJRPR), vol. 6, no. 6, 2025.
[10] Bright AI, "Choosing the Right Embedding Models for RAG in Gener- ative AI Applications," Medium, 2023.
[11] CFA Institute Research and Policy Center, "RAG for Finance: Automat- ing Document Analysis with LLMs," Automation Ahead Content Series, CFA Institute, 2024.
[12] Author(s), "Chapter Title," in Book/Proceedings Title, Springer, 2024, pp. XX–XX, doi: 10.1007/978-3-032-11976-6 20.
[13] Author(s), "Title of the Paper," arXiv preprint arXiv:2504.14493, 2025.
[14] M. Luckianto and A. A. S. Gunawan, "MARAG-Fin: An Intelligent Multi-agent RAG-LLM Architecture Integrating Financial News Senti- ment and Time Series Data for Data-driven Trading Decision-making," International Journal of Intelligent Engineering and Systems, vol. 19, no. 2, pp. 738–750, 2026, doi: 10.22266/ijies2026.0228.46.
[15] Lumenova AI, "AI in Finance: The Promise and Risks of RAG," *Lumenova AI Blog*, 2024.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY