AI - Driven Internal Data Intelligence Assistant

Sampada Kulkarni[1], Shivi Verma, Apeksha Hatle[2], Shreeja Rajput[3], Pratiksha Jadhav[4]

[1,2,3,4]Department of Information Technology, Progressive Education Society's Modern College of Engineering, Pune, India

DOI: https://doi.org/10.51244/IJRSI.2025.1210000288

Received: 08 November 2025; Accepted: 17 November 2025; Published: 19 November 2025

ABSTRACT

The proliferation of generative AI presents significant productivity opportunities for software companies, but the practice of employees using public AI chatbots poses severe data security risks. This research paper demonstrates the application of a Retrieval-Augmented Generation (RAG) architecture into building a dedicated AI assistant for secure, internal corporate use by reasoning its responses exclusively in a company's private document repository, unlike a general-purpose model. This will ensure that sensitive internal data such as project specifications, internal wikis, project codes, knowledge documents, policy documents etc do not leave the company firewall during the AI operation cycle while also letting the company entity use AI to compare and access the needful resources from huge internal data.Our methodology involves processing and vectorizing internal documents, enabling semantic search for precise information retrieval, and leveraging a large language model (LLM) solely for generating context-aware responses from the retrieved data. We argue that this system provides a practical, secure, and efficient solution for knowledge management and employee assistance, balancing the power of modern AI with the non-negotiable demands of corporate data privacy.

Keywords: Retrieval-Augmented Generation (RAG), AI data security, Knowledge management, vector database, code generation, Natural language Processing (NLP), Document intelligence

INTRODUCTION

Today's software development environment is distinctly information-intensive. Development teams, project managers, and quality assurance engineers often deal with the challenge of locating and understanding internal documentation, the context of complicated legacy codebases which can be poorly documented, and changing company policy or security standards. Having to locate, integrate, apply, and translate all of this knowledge is a drag on productivity and operational efficiency. Concurrently, large language models (LLMs) have caused a disruptive entry point for natural language-based interaction, including the potential to change how humans engage with and obtain information. Examples of public-facing artificial intelligence (AI) assistants demonstrate impressive capabilities, however the opportunity for corporate environments is constrained by a serious security risk, specifically, they require proprietary data— intellectual property, trade secrets or sensitive customer data to be sent to external, third-party servers

Accordingly, establishing this type of an attack vector is unacceptable, it puts organizations at risk of data breach, compliance with data protection obligations and puts them at a competitive disadvantage. In summary, the promise of AI efficiency is intriguing, but often the levels of risk associated with public tools just does not allow for use in enterprise. This article proposes the architecture for a private, self-contained AI assistant to combat this significant conflict. The fundamental challenge is to leverage the analytical and interpretive capabilities of AI to amplify human capacity for information retrieval and understanding while also providing data confidentiality in perpetuity. The solution proposed here removes dependence on third-party AI services by implementing a Retrieval-Augmented Generation (RAG) system that runs exclusively on a company's private document repository. This solution creates a closed-loop, secure environment where the AI's knowledge is intentionally limited to internal vetted/approved information. By interrogating and processing information entirely inside the organizational perimeter, the RAG solution provides the benefit of intelligent assistance without compromising data sovereignty.The forthcoming sections will outline the limitations of other systems, describe our approach to building a local RAG-based assistant, and offer a full architectural schema. The paper will conclude with a review of the effectiveness of the system in achieving its bi-focal goal

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue X October 2025

of providing augmented efficiency with absolute data security.

Objective

The primary goal of our project is to design, implement, evaluate, test and assess a completely local AI assistant that provides secure and effective document intelligence for organisations. Our project will use a Retrieval-Augmented Generation (RAG) architecture to keep sensitive company information bounded by a private infrastructure. This will eliminate the need to use a publicly-available AI service with which sharing the information can risk leaking sensitive data. To accomplish this overall goal, the specific objectives shown below have been developed:

1. To design and build a securely working Retrieval-Augmented Generation (RAG) pipeline. The project will focus on developing a system that can read documents (e.g., PDFs, Word documents, text files in all formats), convert the documents into a vectorized format by using technology that operates locally.

2. To create a user interface for easy interaction. This objective is to create a web interface where users can upload a document and ask their question in the chat tab in natural language easily such that the assistant can be instructed to complete a task (for e.g. write a code to add two numbers). The interface should include a way to present AI-generated answers from its own data citations, so the answers are transparent to the users

3. To ensure that we maintain data sovereignty and privacy through local LLM use. This critical objective highlights integrating a locally-hosted large language model (e.g., via Ollama) for the final response generation. By processing all data and AI inferences entirely on local hardware, the system will guarantee that no sensitive information is transmitted to or processed by third-party APIs.

4. To evaluate the system's performance based on statistical measures: accuracy, response time, and usability. The final objective of our project is to conduct a detailed assessment of the developed system. This will involve measuring the accuracy of the answers against a validated set of queries, standardizing the response latency for a satisfactory user experience, and gathering user feedback on platform use.

LITERATURE SURVEY

In recent years, to develop the AI-Driven Internal Data Intelligence Assistant, we reviewed several studies related to privacy-focused large language models (LLMs), retrieval-augmented generation (RAG), and secure enterprise data management. These studies offered valuable understanding of how to protect sensitive data, enhance information retrieval, and implement AI solutions safely within organizational environments.

Earlier research focused either on performance or data security, but very few combined both in one complete framework. This review helped identify that gap and guided the design of the proposed system.

1. DB-GPT: Empowering Database Interactions with Private LLMs (Xue et al., 2024) – Introduced a privacy-focused LLM using RAG for database queries. While effective for structured data, it required heavy computation and was less suitable for unstructured or lightweight local setups.

2. Confidential Prompting (Zhang et al., 2023) – Proposed using Trusted Execution Environments (TEEs) to secure prompts during cloud inference. Though privacy improved, dependence on cloud infrastructure raised costs and limited offline use.

3. E2E Data Extraction Framework (Kumar & Singh, 2025) – Developed a deep learning-based system to structure unorganized enterprise data. However, it did not include security or access control features needed for sensitive information.

4. Fine-Tuning LLMs for Enterprise Applications (Raj et al., 2024) – Explored efficient fine-tuning methods like LoRA and Q-LoRA to improve model performance. These methods, though accurate, were

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue X October 2025

costly and required powerful infrastructure.

From these studies, it was clear that there is a need for a system that combines local deployment, secure authentication, and retrieval-based intelligence. The proposed AI-Driven Internal Data Intelligence Assistant addresses this by blending the privacy ideas of DB-GPT, the secure access concepts of Confidential Prompting, and adaptable enterprise fine-tuning — all under a RAG-based setup using ChromaDB, Ollama, and Supabase to ensure security, accuracy, and scalability.

Existing solutions:

Numerous enterprise-grade solutions that help document analysis and knowledge search have been developed in response to the growing demand for incorporating artificial intelligence into corporate workflows [4][5]. These systems typically make an effort to strike a balance between the strict data governance requirements of contemporary businesses and the potent capabilities of large language models (LLMs).

Microsoft Copilot for Microsoft 365:

It is a highly integrated assistant Microsoft Copilot for Microsoft 365, which functions inside the Microsoft ecosystem. The company's "Commercial Data Protection" guarantee, which guarantees that organizational data and customer prompts are kept separate within the tenant's compliance boundary and aren't used to train foundational models, as the company claims. It offers a smooth but vendor-locked experience by using the Microsoft Graph to contextualize responses in user-specific emails, documents, and calendars. This makes the service very expensive hence small organizations cannot always afford this system.

Glean:

An enterprise search and discovery platform powered by AI, creates a unified, searchable knowledge graph that links together a company's diverse data sources ranging from Google Drive and Confluence, to GitHub and Salesforce. Glean adheres to existing access permissions, so users will only receive answers from the documents and data they already have access to. Regardless of these features, we must note that Glean is a cloud-based SaaS model and it is not suitable for domain specific industries.

Bloomberg GPT:

This offers a different solution. It is a domain specific model that has been trained from scratch on a very large body of financial data. This allows for maximizing performance and accuracy relative to the financial domain, and because the model is trained and deployed internally, they have access to the data, and thus full data sovereignty, minimizes privacy concerns regarding external APIs. The drawback lies in the trained model that is domain specific and cannot be reached for general corporate use.

Even though these solutions can be innovative, they typically require substantial financial resources, force platform locking, or are only for specific industries. There is also no check-in for the data security that they might claim which cannot compete with the security against a model that only processes the data locally, in your own system and doesn’t export the data. Thus, there is no generalizable, affordable, and deployable solution in an effective manner for small to mid-sized businesses.

Proposed Solution

Our AI assistant is a standalone web-based solution that allows a company’s employees to interact with their corporate documentation in natural language. The system is designed using a "Retrieval First, Generation Second" construct inspired by RAG [1][3].

Key Principles:

1. **Data Isolation:** All company data remains inside the company's control infrastructure.
2. **Contextual Grounding:** All AI generated content is grounded in retrieved content from the company's internal document store.

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue X October 2025

3. **Transparency:** The system provides citations, referencing all documents used to generate every answer.

Main Components:

1. **Document Ingestion Component:** Accepts and pre-processes various document formats (PDF, DOCX, TXT).

2. **Vector Embedding and Storage Component:** Embeds text into numerical representations (embeddings) and stores in a vector database.

3. **Query Processing and Retrieval Component:** Understand user intent and retrieves the most semantically relevant text chunks. The assistant responds based on retrieved context and prior interactions instead of follow up questions.

4. **Response Generation Component:** Generates a natural language content answer based on retrieved context only.

Advantages Of Proposed Solution

The proposed project "AI driven Internal Data Intelligence Assistant" brings advantages over existing solutions [4][5] and as a stand-alone system that operates independently, unrestricted by dependency on data sharing with public tools.

1. By using a local LLM inference engine (Ollama), our system refuses to compromise the company data. This makes the system fit perfectly with organisations with sensitive data or security conscious values.

2. The project uses open-source technologies and doesn't use any paid API or services from the web, making the solution highly cost effective and affordable. This way we can serve small organisations by providing economic advantage.

3. The proposed project supports the elimination of vendor lock-in by using open-source technologies like python, Ollama, ChromaDB. Hence the system is not exclusively tied to any specific vendor, helping the system gain flexibility and independent operations.

4. The system architecture itself provides for the high relevance and accuracy in outputs from the assistant. This is because the system AI's knowledge context is based around company data itself, hence the system delivers answers that provide high accuracy and relevance. Hence rated higher against public tools that are inflicted with noise, hallucination and unnecessary generic information.

Working Details

1. At the center of this system is something called a **Retrieval-Augmented Generation (RAG)** pipeline —it is a smart way to make AI give accurate answers using real information instead of random guesses.

2. Here's what actually happens behind the scenes: First, all the documents are cleaned up and broken into smaller, meaningful pieces (usually around 500–1000 characters long) so they're easier for the AI to work with. Then, each of these pieces is turned into a special kind of digital "fingerprint" using a model like **Sentence-BERT**.

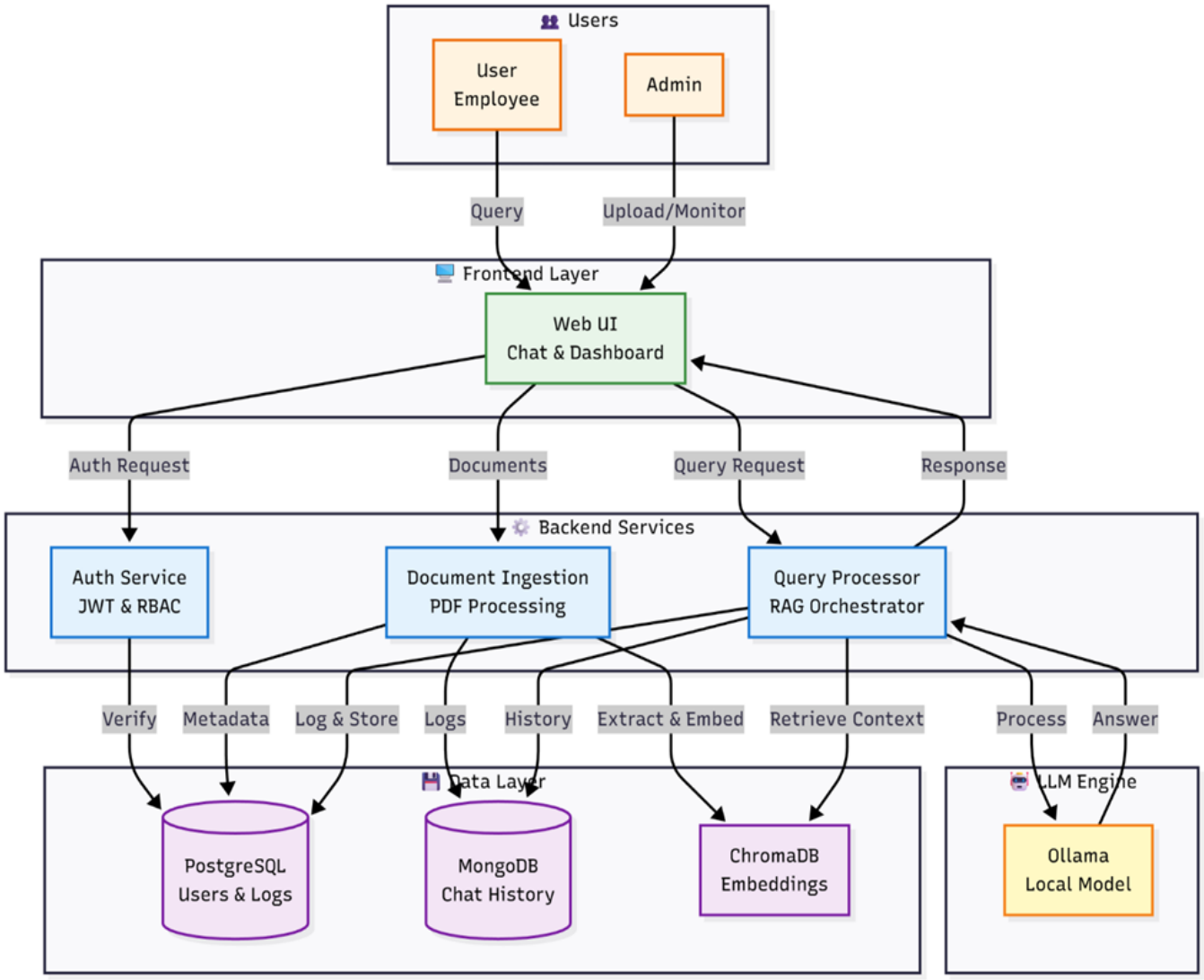
3. These fingerprints are stored in a **vector database** (like **ChromaDB**), which is designed to quickly find text pieces that have similar meanings. Hence, when someone asks a question, that question is also converted into a fingerprint, and the system searches for the few most relevant text chunks — usually the top 3 to 5.

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue X October 2025

4. Once those pieces are found, they're placed into a prompt that basically says: "Use only this information to answer the question." The whole thing is then sent to a **Large Language Model (LLM)**, which reads the context and writes a clear, accurate answer — as if it's pulling the answer straight from the documents.

System Architecture

Fig 1 - System Architecture Diagram



We aim to build a system that is capable of finding relevant and accurate information effortlessly — a place where users can simply ask questions in natural language and get results pulled straight from their own documents.

User Interaction: There are two types of users that contribute to the system. Regular employees, who are direct beneficiaries of the project. They can use queries and ask questions to the assistant; and admins, who handle uploading and managing the documents. Both use the same web dashboard — a single space that works like a chat and a control center rolled into one.

Frontend Layer: The web interface is responsible for direct interaction of the user with the system. It is simple and intuitive — users log in, upload files, type in their questions, and instantly get answers. Behind the scenes, it quietly manages authentication, routes every request to the right service, and brings the AI's response back to the user.

Backend Services: Backend controls the system working, manages the logic, data, and server-operations. The backend is split into three main parts:

1. **Auth Service** makes sure everything stays secure. It checks every user's identity using JWT tokens and role-based access control before letting them in.
2. **Document Ingestion** takes care of the uploaded PDFs. It extracts all the text, splits it into smaller chunks, generates metadata, and stores everything neatly in MongoDB — from the raw documents to chat history.
3. **The Query Processor** is the system's decision-maker. When a user asks a question, it searches for relevant information from the vector database, prepares the context, and coordinates with the AI model to generate a meaningful, accurate answer.

Data Layer: Each database is responsible for specific tasks in our system.

1. **PostgreSQL** manages the structured stuff — like user accounts and system logs.
2. **MongoDB** handles the flexible data of the system— chat records, document logs, etc.
3. **ChromaDB** stores the document embeddings — the numerical "understandings" of text that allow the system to match meaning, not just exact words.

LLM Engine: The LLM Engine that powers our system is **Ollama. Ollama** supports the language model to understand query and generate results. Once the Query Processor finds the right pieces of information, Ollama blends them with the user's question and generates the response that is completely based on the documents (company data) uploaded to the system by the admins. Hence, this ensures that the results are accurate and relevant, and not degraded by unnecessary generic information.

In simple terms: You upload your PDFs, the system processes and stores them intelligently, and later, when you ask something, it digs into your own data to give you the right answer — fast, reliable, and backed by facts.

CONCLUSIONS

This research offers a thorough explanation of the system "**AI - Driven Internal Data Intelligence Assistant**", the application of RAG and in-depth functionality of the project. The paper explains how our project is an appropriate solution to resolve the conflict between inclusion of artificial intelligence to promote productivity and risking the company's data security. By using the RAG pipeline, we demonstrate how AI and open-source technologies benefit the users without being under control of public tool providers.

The proposed solution successfully completes the necessary checks to make a closed-loop environment for data processing. The integration of Ollama as a local LLM inference engine is one of the most crucial foundations of our work. It holds the centre of the system architecture and ensures that no company data leaves the organisational firewall. The assistant provides the employee with a natural language-based interface that gives accurate and relevant results and values transparency.

This paper confirms that development of secure corporate tools is needed and can be built with design explained in this article. The project has been carefully developed in accordance with industry standards. Future work will include the expansion of features in terms of scalability, role-based access control, optimizing the document retrieval mechanism, using image/video-based information uploads, and addition of multi-hop queries. This project stands as evidence to the idea that AI capabilities and data security can coexist.

REFERENCES

1. Xue, L., Chen, M., and Li, Y., "DB-GPT: Empowering Database Interactions with Private LLMs," IEEE International Conference on Data Engineering, 2024.
2. Zhang, Y., Wu, H., and Zhao, T., "Confidential Prompting: Privacy-Preserving LLM Inference on Cloud," IEEE Conference on Secure and Trustworthy Machine Learning, 2023.
3. Kumar, R., and Singh, A., "E2E Data Extraction Framework from Unstructured Data: Integration of Deep Learning and Text Mining Techniques," Journal of Intelligent Information Systems, 2025.
4. Raj, S., Mehta, P., and Bansal, V., "Fine-Tuning Large Language Models for Enterprise Applications," IEEE Conference on Artificial Intelligence and Data Science, 2024.