Uni-RAG Agent

Team Name: SpongeBoB

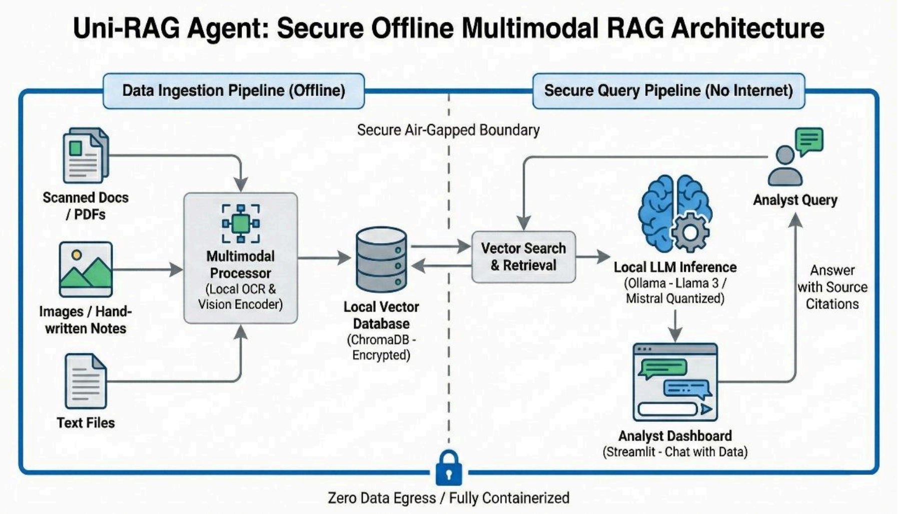
Team Members:

1.)Nuka Sumith Chandra
2.)Edupuganti Dheeraj Chandra
3.)Gongalreddy Sri Venkat Reddy
4.)V Vijay Chandra

Problem Statement:

- Organizations handling sensitive or confidential data (e.g., defense, finance, healthcare, legal) face significant challenges in leveraging modern AI for document analysis due to data privacy concerns associated with cloud-based solutions. Sending proprietary documents to external servers violates Zero-Trust policies and creates security risks.
- A large volume of existing enterprise documents consists of scanned images or includes handwritten annotations, which are inaccessible to standard text extraction tools, limiting the scope of AI analysis.
- Users often need to ask complex, multi-step questions that require synthesizing information from various parts of one or multiple documents, a task that simple Q&A systems struggle with.
- There is a need for a trustworthy AI assistant that provides verifiable answers directly linked back to the source material within the documents.

Proposed Solution:



Uni-RAG Agent: Secure Offline Multimodal RAG Architecture
Data Ingestion Pipeline (Offline)
Secure Query Pipeline (No Internet)
Secure Air-Gapped Boundary
Scanned Docs / PDFs
Images / Hand-written Notes
Text Files
Multimodal Processor (Local OCR & Vision Encoder)
Local Vector Database (ChromaDB - Encrypted)
Vector Search & Retrieval
Local LLM Inference (Ollama - Llama 3 / Mistral Quantized)
Analyst Query
Answer with Source Citations
Analyst Dashboard (Streamlit - Chat with Data)
Zero Data Egress / Fully Containerized

- We propose the Uni-RAG Agent, a comprehensive system built on a React frontend and FastAPI backend that provides advanced document interaction capabilities entirely offline.
- It utilizes a Retrieval-Augmented Generation (RAG) pipeline powered by local Large Language Models (LLMs like Phi-3 via Ollama) and a local vector database (ChromaDB).
- Crucially, it integrates Optical Character Recognition (OCR) using Tesseract to process scanned PDFs and extract text, including basic handwritten notes, making previously inaccessible documents usable.
- The system is designed with an agentic architecture in mind, where the RAG+OCR capability functions as a robust tool that a reasoning agent (like one using the ReAct framework) can leverage for multi-step task decomposition and execution on complex queries.
- Answers generated by the LLM are grounded in the retrieved document context, with support for source citations (Retrieving the document file location) . Optional Text-to-Speech output enhances accessibility.

Technologies/Tools:

1.)**AI & ML:**

- **Ollama LLM (Phi-3)** for fully local, offline language generation
- **Sentence Transformers (all-MiniLM-L6-v2)** for creating high-quality text embeddings
- **ChromaDB** for fast and efficient local vector similarity search
- **Tesseract OCR + pytesseract** for extracting text from scanned PDFs and images
- **Coqui TTS (Tacotron2-DDC)** for generating optional audio output
- **PyTorch (CPU/GPU with CUDA)** to accelerate embeddings, OCR, and TTS models

2.)**Data Processing:**

- **pdf2image + Poppler** for converting PDF pages into images
- **pytesseract + Tesseract Engine** for OCR on scanned/handwritten documents
- **Docx2txtLoader** for parsing DOCX files
- **RecursiveCharacterTextSplitter (LangChain)** for intelligent text chunking
- **FastAPI file handling + temporary storage** for managing file uploads and caching

3.)**Framework & UI:**

- **Python 3.11+** as the core backend language
- **FastAPI + Uvicorn** for backend API services
- **LangChain + langchain-chroma + langchain-huggingface + langchain-ollama** for orchestration, embeddings, vector storage, and LLM integration

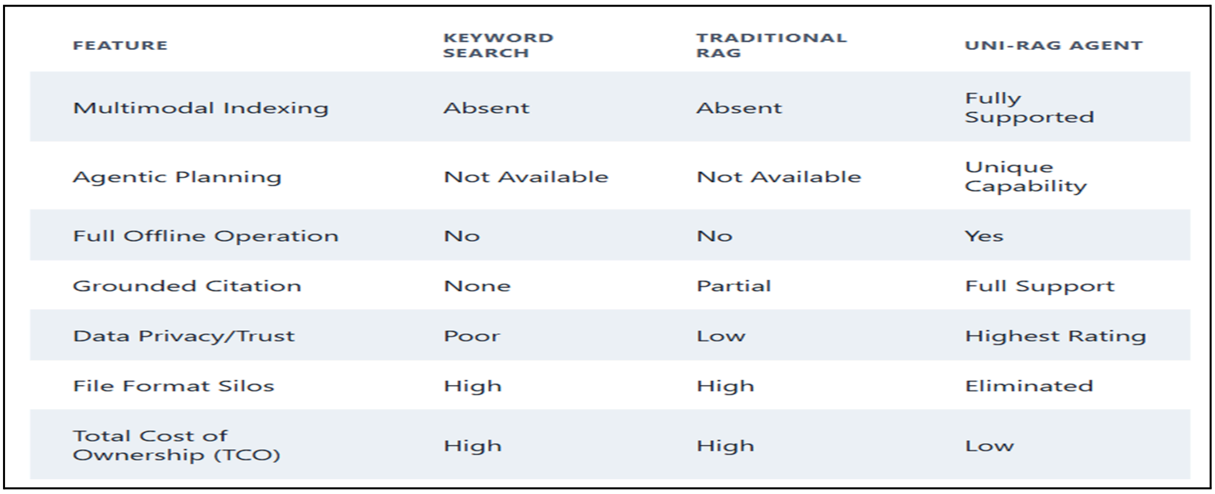
- **React + TypeScript + Vite** for a fast, modular frontend
- **Tailwind CSS + shadcn/ui** for UI styling and components
- **Axios** for frontend–backend communication
- **Framer Motion** for smooth animations
- **React Three Fiber** for optional 3D UI interactions

4.)**Agent Workflow:**

- **Retrieval-Augmented Pipeline** using context fetched from ChromaDB
- **Automated Document Processing Flow**
  PDF → Image (pdf2image) → OCR (Tesseract) → Text → Chunks → Embeddings → ChromaDB
- **Dynamic Prompt Construction** using retrieved context
- **OllamaLLM (Phi-3)** for final response generation
- **Optional Coqui TTS step** to convert model output into downloadable audio
- **JSON Response Layer** returning text, sources, and audio URL to the frontend

Expected Impact:

The solution enables the secure adoption of AI for document understanding in highly restricted and sensitive environments by operating fully offline and ensuring that all data—whether text, PDFs, images, audio, or scanned documents—remains within the organization's local infrastructure. It dramatically increases productivity by converting scanned, handwritten, or low-quality documents into searchable and queryable formats through robust OCR and cross-modal retrieval, allowing users to extract insights in seconds instead of manually reviewing pages of data. Trust is further enhanced through verifiable, grounded answers that always include transparent source citations, enabling analysts to validate every piece of information with confidence. Additionally, the platform provides an intuitive and accessible interface designed for non-technical users, with optional audio output for hands-free operation or accessibility needs, making advanced AI capabilities usable across diverse teams, including legal, research, administration, and field operations.



FEATURE	KEYWORD SEARCH	TRADITIONAL RAG	UNI-RAG AGENT
Multimodal Indexing	Absent	Absent	Fully Supported
Agentic Planning	Not Available	Not Available	Unique Capability
Full Offline Operation	No	No	Yes
Grounded Citation	None	Partial	Full Support
Data Privacy/Trust	Poor	Low	Highest Rating
File Format Silos	High	High	Eliminated
Total Cost of Ownership (TCO)	High	High	Low