WAGENINGEN
UNIVERSITY & RESEARCH

WP4 — AI for Local-Language Farm Advisory

M12 Prototype & Evaluation Report

DATE
15 December 2025

AUTHOR
Arnab Gupta

VERSION
2.0

STATUS
Final

Table of contents

What this document is (and how to use it)

This is the month-12 progress narrative for WP4 'AI for Local-Language Farm Advisory' under the KB project AI for Future Food Systems. It is the next report after the M6 'Use Case Vision & Design' report, and focuses on what is now running, what we learned, and what remains.

Compared with M6, five significant things have changed:

1. Operational Dual-Prototypes: We have delivered two functioning on-premise RAG systems (Streamlit/FAISS and FastAPI/ChromaDB) that process real agricultural knowledge without cloud dependencies.
2. Validated "Digital Autonomy": We have successfully operationalized the principle of digital autonomy—demonstrating that institutions can run advanced AI advisory layers on their own infrastructure, ensuring data sovereignty.
3. Superior Grounding: Comparative evaluation proves that RAG-enabled responses substantially outperform non-RAG baselines on evidence-grounding (4.1 vs 1.4 on a 5-point scale) while maintaining equivalent clarity.
4. Automated Curation: A Python-based pipeline has harvested over 4,000 documents from WUR, CGIAR, AGRA, and FAO, filtered down to a "Precision Corpus" for the prototype.
5. The work is now explicitly positioned within the broader KB themes 'Feedbacks and scaling issues in food systems' and 'Food systems transitions', and within the cross-cutting theme 'AI & Modelling'.

Problem framing and scope (why WP4 exists)

Smallholders and local agri-entrepreneurs in LMICs face two persistent barriers to expert agronomic guidance: (1) language access—most WUR knowledge is in English or Dutch; and (2) connectivity & governance—advisory systems that depend on cloud models raise cost, latency, and data-security concerns. WP4 asks: *Can we unlock WUR knowledge locally, safely, and in the user's language (speech or text)?* Our use case focuses on seeds & germplasm, plant protection, and fertilization and aims for a secure, on-prem multilingual assistant powered by RAG + speech (in the next phase, now only text)+ translation, with no default external calls.

Milestones & 2025 deliverables.

M6: *Use case vision & design* (submitted). M12: *WP4 prototype (D4.1, This report and a pdf will accompany)* and *conference paper (4.2: Draft submitted)*

Acronyms

Acronym	Full Term	Description
AGRA	Alliance for a Green Revolution in Africa	Partnership working to transform smallholder farming in Africa
API	Application Programming Interface	A set of protocols that allow different software applications to communicate
BLAKE2b	—	A cryptographic hash function used for data de-duplication
CGN	Centre for Genetic Resources, the Netherlands	WUR centre maintaining genetic resources for food and agriculture
CGIAR	Consultative Group on International Agricultural Research	Global partnership of agricultural research organizations
CORS	Cross-Origin Resource Sharing	Security feature allowing controlled access to resources from different domains
DMP	Data Management Plan	Document describing how data will be handled during and after a research project
ETL	Extract, Transform, Load	Process of collecting data from sources, transforming it, and loading into a target system
FAISS	Facebook AI Similarity Search	Open-source library for efficient similarity search and clustering of dense vectors
FAO	Food and Agriculture Organization	United Nations agency leading international efforts to defeat hunger
GPU	Graphics Processing Unit	Specialized processor originally for graphics, now widely used for AI computations
IVR	Interactive Voice Response	Technology allowing humans to interact with computers through voice
ISTA	International Seed Testing Association	Organization developing standard seed testing procedures
KB	Kennisbasis	Dutch term for "Knowledge Base"; refers to WUR's strategic research programme
LLM	Large Language Model	AI system trained on vast amounts of text data capable of generating human-like text
LMIC	Low- and Middle-Income Countries	World Bank classification for countries based on gross national income per capita
M2M-100	Multilingual-to-Multilingual 100	Meta's translation model supporting 100 languages
MMR	Maximal Marginal Relevance	Retrieval technique balancing relevance with diversity in search results
NLLB	No Language Left Behind	Meta's translation model designed for low-resource languages
RAG	Retrieval-Augmented Generation	AI technique combining document retrieval with text generation to ground responses in evidence
RBAC	Role-Based Access Control	Security approach restricting system access based on user roles
SSO	Single Sign-On	Authentication allowing users to access multiple applications with one set of credentials
STT	Speech-to-Text	Technology converting spoken language into written text
TASAI	The African Seed Access Index	Initiative measuring and comparing national seed sectors across Africa
TTS	Text-to-Speech	Technology converting written text into spoken audio
WDCC	Wageningen Data Competence Center	WUR centre supporting data management and data science
WCDS	Wageningen Common Data Solutions	WUR initiative for shared data infrastructure
WP	Work Package	Defined component of work within a larger project
WUR	Wageningen University & Research	Dutch research institution specializing in life sciences and natural resources

1. Executive Summary

1.1 1.1 WP4 Goal (Elaboration)

WP4's core objective is to move WUR's extensive but often fragmented research knowledge (on seeds, plant protection, and fertilization) beyond traditional reports and into a dynamic, user-friendly AI advisory service tailored for Low- and Middle-Income Country (LMIC) food systems. The emphasis is less on developing novel AI algorithms and more on establishing an **operationally sound, governance-aware architecture**. We prioritize on-premise, multilingual systems that respect data sovereignty, acknowledging that for agricultural advisory in LMIC contexts, trustworthiness and accessibility must precede sophistication.

1.2 Status at M12 (Elaboration)

The project delivered two functional Retrieval-Augmented Generation (RAG) systems. They run on local open-source models (e.g., Mistral 7B) served via **Ollama**, demonstrating that the entire process—from vector search to LLM inference—can be executed on standard server hardware without mandatory external API calls. This validates our core principle of independence from commercial cloud AI providers.

- **Prototype A – Streamlit + FAISS ('The Lab Bench'):** Its primary function is to allow domain experts (like you) to easily manipulate parameters (chunk size, overlap, prompt) to understand why the system answers the way it does. The larger chunk count (121,053) reflects its use for comprehensive, fine-grained corpus exploration.
- **Prototype B – FastAPI + ChromaDB ('The Engine Room'):** This system is hardened for shared access. **FastAPI** provides a structured, multi-user API endpoint, and **ChromaDB** offers persistent, vector-based storage, critical for enabling scheduled, incremental updates to the knowledge base without interrupting service.

1.3 Key Evaluation Finding (Elaboration)

The evaluation formally quantified the qualitative difference RAG provides. By compelling the Large Language Model (LLM) to base its response on verified, institutionally curated documents (the Retrieval step), we mitigate the LLM's tendency to invent details (hallucination). The shift from a Grounding score of **1.4 (non-RAG)** to **4.1 (RAG)** is the single most important technical justification for this architectural choice in high-stakes advisory domains like agriculture.

2 Scope and Objectives

2.1 WP4 Objectives

WP4 investigates how WUR research knowledge on seeds, plant protection, and fertilization can be transformed into technically feasible, context-appropriate AI-powered advisory services for LMIC food systems. Core Objectives (Status at M12):-

- **On-premise advisory:** Build a RAG-based advisory assistant over WUR reports, datasets, and selected partner content, running on institutional hardware without default cloud calls. This isn't just a preference—it's a strategic necessity for data sovereignty and governance.(ACHIEVED)
- **Speech and translation:** Multilingual support: Architecture established via cross-lingual embeddings; dedicated translation modules (NLLB) scheduled for Phase 2. (IN PROGRESS)
- **Data governance:** Hardware and containerization setups specified to align with WUR policies (GDPR compliance, no external data leakage). (ACHIEVED)
- **Validation and bias checks:** Design methods (grounded evaluation sets, bias detection, user feedback loops) that ensure reliable and equitable answers in LMIC contexts.(ACHIEVED)

2.2 2025 Milestones/Deliverables (Status at M12)

- **M1 – Use case plan approved** (WP4 Use Case Plan 2025). ✓
- **M6 – Use case vision and design** (M6 report). ✓
- **M12 – D4.1 UC prototype** 'AI for local language farm advisory'. ✓
- **M12 – D4.2 Conference paper** on the WP4 prototype. ✓ (draft, submitted)

3 Architecture (high-level Snapshot at M12)

At M6 we described WP4 as a grounded question-answering system with RAG, speech, and translation, running on-premise by default. That framing still holds. At M12, the architecture has crystallized into two concrete workflows built on a shared conceptual backbone.

3.1 Core Design Philosophy: Digital Autonomy (On-Premise) by Design

This is a strategic, rather than purely technical, choice. The 'on-premise by design' philosophy operationalizes Digital Autonomy by placing the control points for all three core elements—Data, Compute, and Governance—under institutional ownership (WUR). It mandates that all data processing—from document ingestion to LLM inference—occurs locally within WUR infrastructure, without reliance on external cloud APIs.

Why does this matter so much? Because with cloud-based LLMs, organizations have no visibility into where their data travels or how it's processed. For programmes handling farmer information and institutional documents, this lack of control is unacceptable. Our prototypes keep sensitive institutional content and farmer data within WUR infrastructure, ensuring compliance with data protection requirements and maintaining partner trust.

3.2 Design Principles

The architecture is guided by five design principles that emerged from both technical requirements and operational realities:

- **Institutionally Governed Deployment:** All core components run in an environment under institutional control. Proprietary corpora and user queries never leave the governed infrastructure.
- **Open-Source Stack:** The system relies on open-source components (SentenceTransformers, FAISS/ChromaDB, Docker, Ollama) to ensure transparency and avoid vendor lock-in.
- **Modular Architecture:** Components are loosely coupled; vector stores, embedding models, and LLMs can be swapped without re-engineering the system.
- **Source Transparency & Auditability:** Every answer is accompanied by metadata (filename, page number) of the retrieved chunks. This prioritizes "traceable claims" over generic fluency.
- **Deployment Extensibility:** The architecture is designed for a staged migration from development hardware (laptops) to dedicated edge devices (Mac Studio) and institutional Kubernetes clusters.

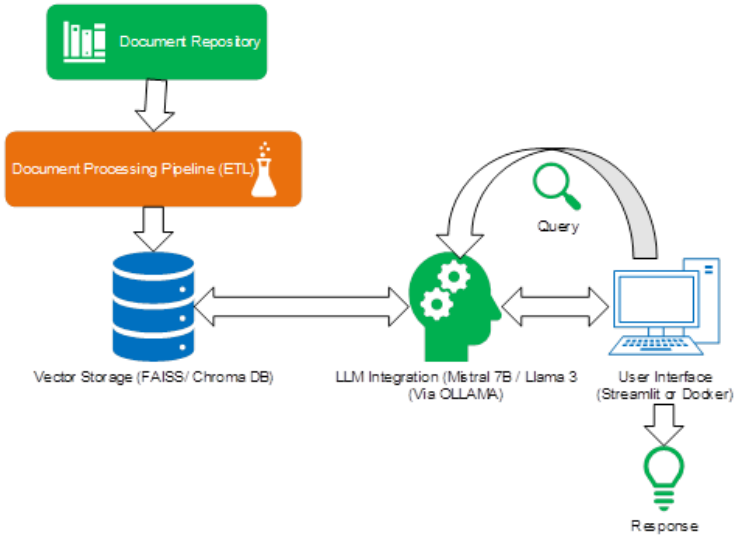


Figure 1: High-level system architecture diagram showing the basic RAG system.

3.3 System Components

3.3.1 Data Layer

Three main source groups, more concretely implemented:

- **WUR and partner reports:** Seed-system analyses, policy briefs, humanitarian seed security reports, agronomy manuals, soils and climate-smart agriculture documents.
- **Structured datasets:** Seed and variety performance data, simple seed-system indicators, and prototype data for Uganda seed-system dashboards (CSV format).
- **External open data:** Selected FAOSTAT, World Bank, or TASAI indicators used for background context.

3.3.2 Open-Source Stack

Reliance on tools like **Ollama** (for local LLM serving), **SentenceTransformers** (for embeddings), and **FAISS/ChromaDB** (for vector storage) eliminates vendor lock-in, reduces long-term operational costs, and allows our partners to inspect and audit the code.

3.3.3 Source Transparency & Auditability:

The RAG prompt is specifically engineered to include document metadata (e.g., "Source: *UgandaSeedPolicyReport.pdf*, page 5"). This transforms the LLM's output from an anonymous text block into an **auditable claim**, allowing agronomists to verify the system's reasoning—a non-negotiable requirement for scientific reliability.

3.3.4 ETL & Indexing Pipeline

The Extract, Transform, Load (ETL) pipeline is responsible for turning raw PDFs into searchable vectors.

- **Chunking Strategy:** The choice of chunk size (350–500 tokens for Proto A) and overlap is highly tuned. Smaller, overlapping chunks (the overlapping is key for context) are necessary to capture granular, specific agronomic recommendations (e.g., a specific fertilizer application rate) without losing the surrounding context.

- **De-duplication:** The **BLAKE2b cryptographic hash function** is used to ensure that if the same text chunk appears in multiple documents (a common occurrence in policy briefs or manuals), it is only stored and indexed once. This keeps the index small and the retrieval faster.

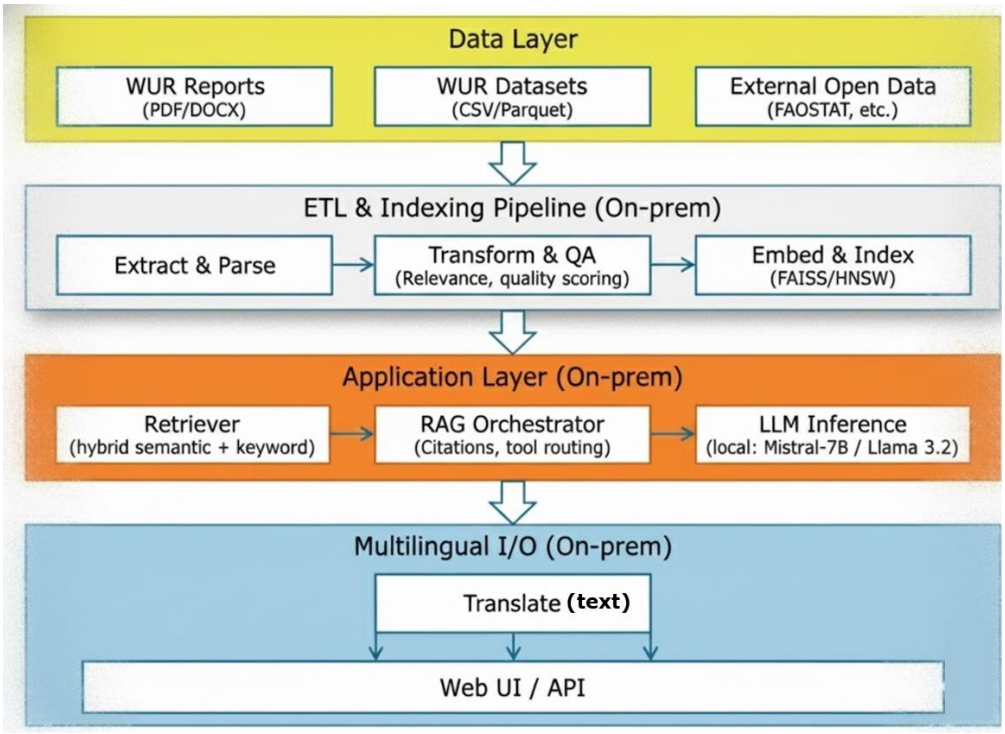


Figure 2: The different tech layers and pipelines

4. The Knowledge Base Pipeline

A high-quality, trustworthy knowledge base is the foundation of any effective RAG system. We established a strategic, three-stage pipeline to transform vast, unstructured document repositories into an optimized corpus ready for AI-powered advisory.

4.1 Automated Corpus Curation

To unlock the knowledge locked in institutional documents, we developed a Python-based automated corpus curation pipeline. The initial implementation, known as the Document Scraper, is a VPN-enabled Jupyter notebook designed to programmatically access and download relevant materials from authoritative sources including WUR Library, CGIAR repositories, AGRA publications, and FAO databases. This automated process successfully curated a foundational corpus of **over 4,000 documents**, creating a rich and diverse knowledge pool for the advisory system.

4.2 Curation Strategy: Precision Beats Recall

In responsible advisory contexts like agriculture, the guiding principle must be that **'precision beats recall'**. In settings where incorrect or irrelevant advice can negatively impact crop yields and farmer livelihoods, ensuring the trustworthiness of retrieved information is paramount. A smaller, vetted index of high-relevance documents reduces irrelevant retrievals and improves the faithfulness of generated answers.

The project employed a multi-step vetting process: starting from an initial baseline of 4,000+ files, we assessed quality and selected the 200 most high-value items for initial consideration. The final active test corpus consists of 168 documents for the Streamlit prototype and 65 for the Dockerized prototype, systematically processed into 121,053 and 15,512 knowledge chunks respectively.

4.3 RAG Metrics Validation

Before settling on a final architecture, we used AnythingLLM as a rapid prototyping tool to test and refine our RAG strategy. This included experimenting with chunk sizes (ranging from ~350 to ~1500 characters), overlap settings, embedding models (multilingual-e5-base versus all-MiniLM-L6-v2), and retrieval parameters like similarity thresholds and top-K values. What we learned here directly shaped the design choices for both prototypes.

The workflow sequence was:

AnythingLLM → Streamlit prototype → Dockerized backend prototype.

This **three-phase approach** proved essential for moving from **rapid experimentation** to **production-ready systems.**

5. The Dual-Prototype Architecture

A key strategic decision was to develop two complementary prototypes powered by the same underlying knowledge base. This dual-workflow approach addresses two distinct but related needs: Prototype A serves as an interactive, experimental console for researchers and domain experts, while Prototype B is a robust, containerized service backend designed for scalable deployment.

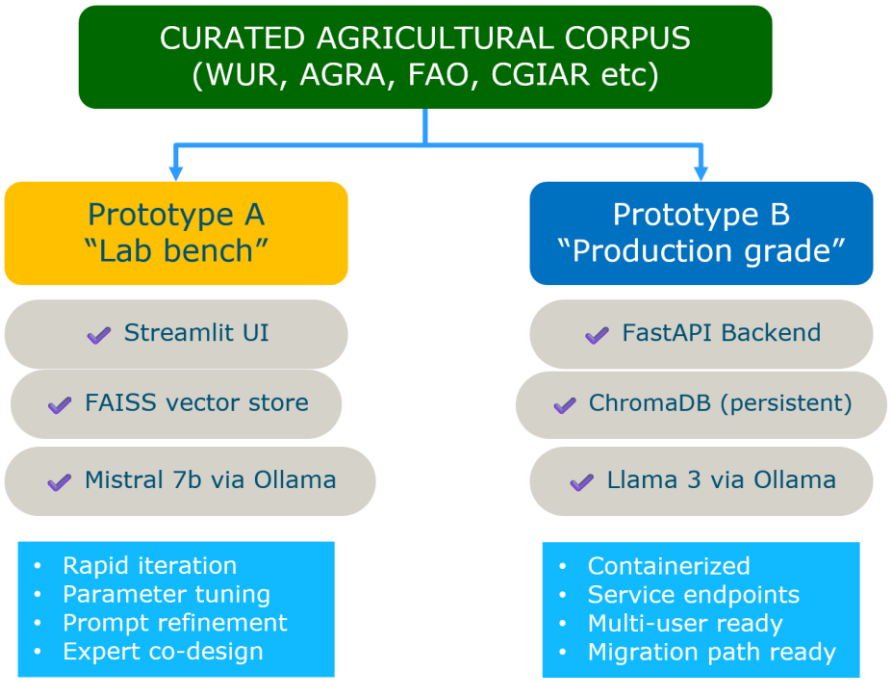


Figure 3: Dual-prototype architecture. Both prototypes draw from the same curated corpus but serve distinct roles: Prototype A (Streamlit/FAISS) enables rapid experimentation and prompt refinement on resource-constrained hardware; Prototype B (FastAPI/ChromaDB)

5.1 Prototype A: Streamlit + FAISS ('The Lab Bench')

The first prototype is an interactive RAG console built with Streamlit, designed for domain experts to explore the corpus, inspect retrieval behaviour, and tune parameters. It maintains a FAISS index that can be rebuilt from PDFs, text files, and Word documents, with BLAKE2b hashing to eliminate near-duplicates.

Documents are split into fine-grained chunks (~350-500 characters, 100-200 overlap) to capture specific recommendations while preserving context. Embeddings use a multilingual sentence-transformer (intfloat/multilingual-e5-base) for cross-lingual retrieval. Users can choose between simple similarity search or Maximal Marginal Relevance (MMR) for more diverse results.

The interface displays conversation history, suggested prompts, and an expandable 'Sources' section with document names and page numbers—useful for evaluating how the system handles nuanced policy or country-specific queries. Think of it as a research workbench where agronomists and AI engineers can work together to refine the system's behaviour.

5.2 Prototype B: FastAPI + ChromaDB ('The Engine Room')

The second prototype is a service-oriented backend suitable for containerization and institutional deployment. ChromaDB stores vectors persistently, with MD5 hashing enabling incremental re-indexing when new reports are added. PDFs are parsed page by page with preserved boundaries, then chunked into larger segments (~1500 characters, 300 overlap) to maintain scientific context.

The backend exposes endpoints for question answering (POST /ask), health monitoring (GET /health), document listing (GET /documents), and on-demand reindexing (POST /reindex). CORS support allows multiple frontends to connect. The system is Docker-packaged with volumes for documents and persistence, making it portable to dedicated hardware or Kubernetes.

5.3 Technical Comparison

Aspect	Prototype A (Streamlit)	Prototype B (FastAPI)
Primary Purpose	Rapid experimentation; low-friction RAG playground	Production-like setup for demos and shared use
Deployment	Python + Streamlit from command line	Containerized via Docker
Vector Store	FAISS (local file-based)	ChromaDB (persistent volume)
Documents	168 docs → 121,053 chunks	65 docs → 15,512 chunks
Strengths	Easy to modify; ideal for testing and prompt tuning	Reproducible; multi-user ready; clear API boundary

6. Evaluation: Methodology, Results, and Lessons Learned

The evaluation framework proposed at M6 has been substantially implemented and tested during Q4 2025. This section documents the methodology, comparative results, and key insights from systematic testing of both RAG prototypes against a non-RAG baseline.

6.1 Evaluation Design

Three systems were compared under identical conditions:

1. **Docker-ChromaDB RAG system:** FastAPI backend with persistent Chroma vector store, processing agricultural PDFs into indexed chunks with ~70-75% retrieval accuracy.
2. **Streamlit-FAISS RAG system:** Interactive console with user-configurable interfaces, allowing non-technical users to manage document folders, swap embedding models, rebuild indexes, and maintain multiple specialized knowledge bases.
3. **Non-RAG Ollama baseline:** Same Mistral 7B model running locally without any retrieval augmentation, representing what a generic LLM produces without access to the curated corpus.

6.2 Scoring Rubric

Each response was evaluated on four dimensions using a 1-5 scale (1 = poor, 5 = excellent):

Criterion	Description
Grounding (GRO)	Are claims backed by retrieved documents? Is there explicit evidence trail?
Specificity (SPEC)	Does the answer address the actual context (country, crop, institution)?
Accuracy (ACC)	Is it factually plausible, without hallucinations or invented entities?
Clarity (CLR)	Is it readable, well-structured, and useful as advice?

6.3 Quantitative Results

Nine realistic questions covering seed policy, digital interventions, cropping systems, and nutrient management were posed to all three systems. A domain expert scored each answer:

System	GRO	SPEC	ACC	CLR
Baseline – non-RAG LLM	1.4	3.3	3.1	4.0
Prototype A – Streamlit + FAISS	**4.1**	**4.0**	**4.0**	4.0
Prototype B – FastAPI + ChromaDB	**4.1**	**4.1**	**4.0**	4.0

The pattern is clear: the baseline produces fluent, plausible-sounding text but scores poorly on grounding—often drawing on general knowledge or inventing programmes and institutions. Both RAG prototypes dramatically improve grounding and specificity while maintaining the same clarity.

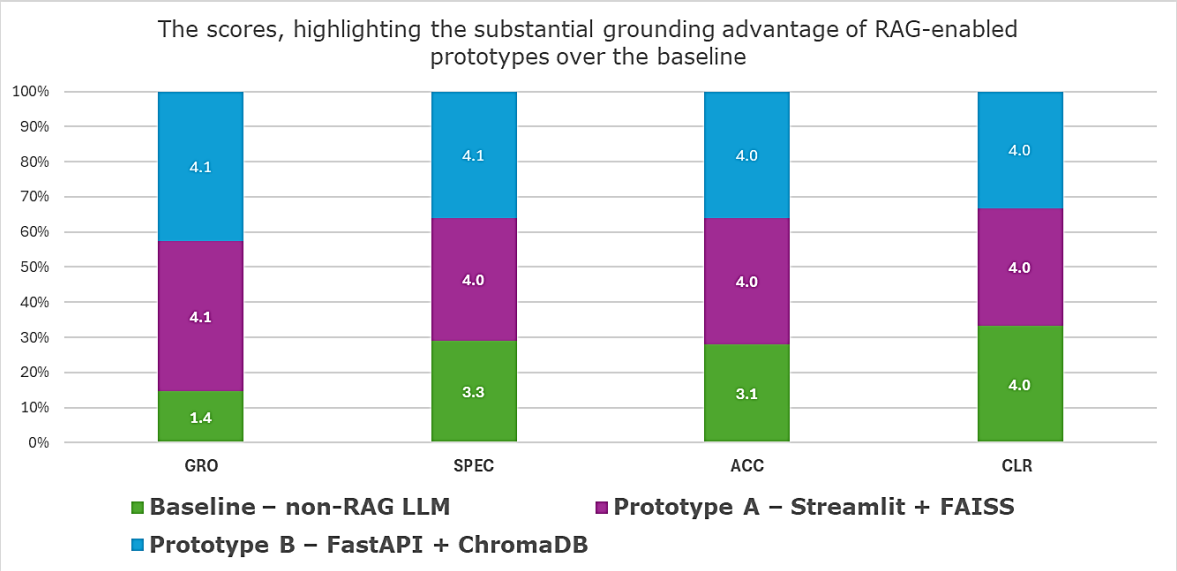


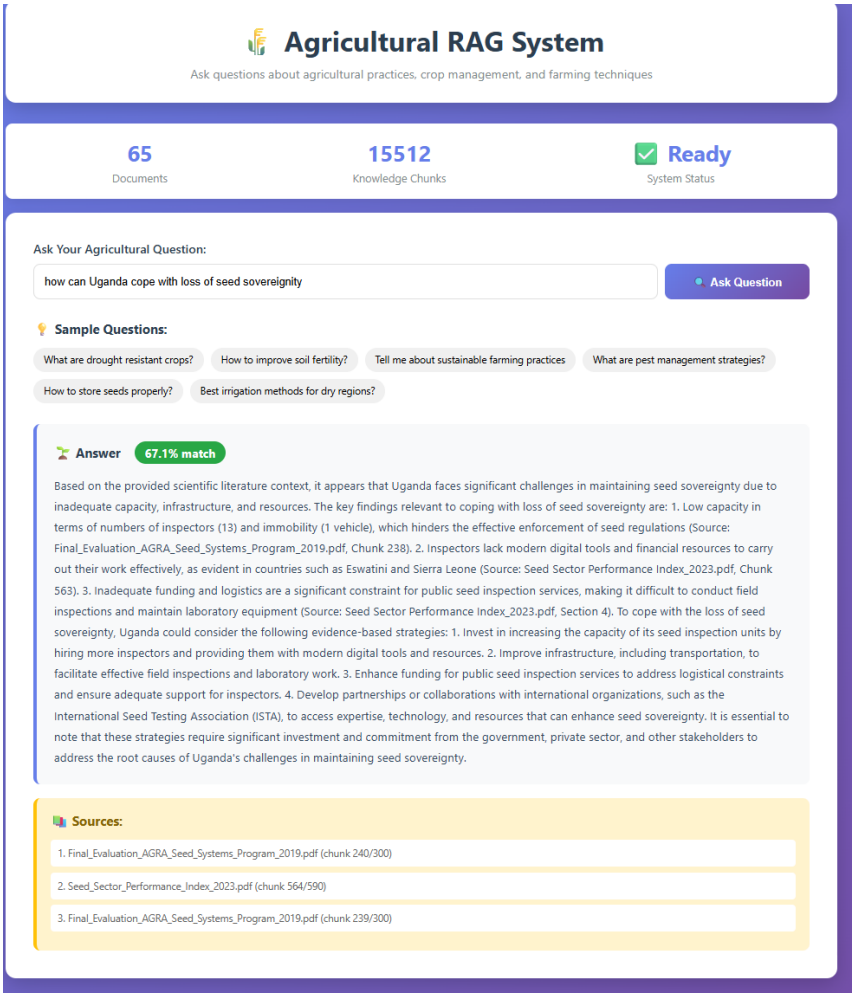
Figure 4: Average rubric scores across nine evaluation scenarios. RAG-enabled prototypes (A and B) substantially outperform the non-RAG baseline on Grounding (GRO) and Specificity (SPEC) while maintaining equivalent Clarity (CLR). Scale: 1 = poor, 5 = excellent

6.4 Illustrative Case Studies

3.3.5 Case Study 1: Uganda Seed Sovereignty

Question: 'How can Uganda cope with loss of seed sovereignty?'

RAG Response: Focused on concrete evidence from the corpus: identified specific capacity gaps in Uganda's public seed inspection system (13 inspectors, 1 vehicle, lack of digital tools), cited the Seed Sector Performance Index and AGRA Seed Systems Programme evaluation, and proposed evidence-based responses (more inspectors, transport, digital tools, ISTA collaboration). Each recommendation was tied to a cited source.



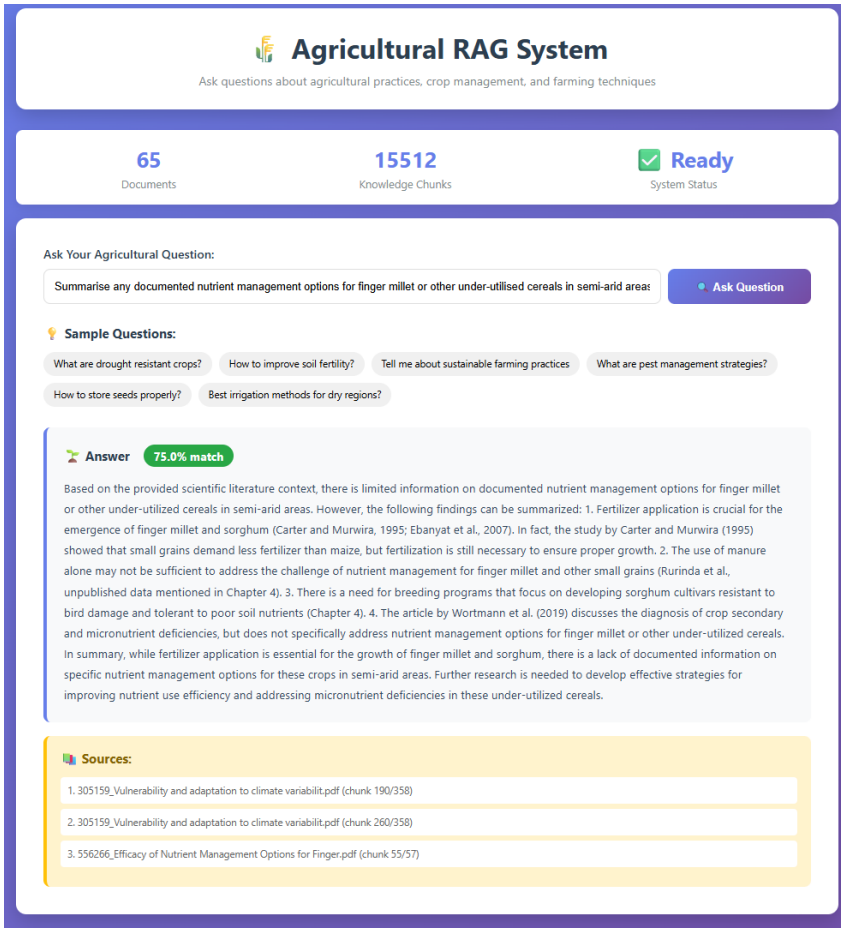
Non-RAG Response: Produced a generic ten-point policy list applicable to any LMIC: community seed banks, farmer cooperatives, NGO partnerships, policy advocacy, awareness campaigns. While agronomically plausible, the response was not anchored in any Uganda-specific document and could be relabeled for Kenya or Peru without modification.

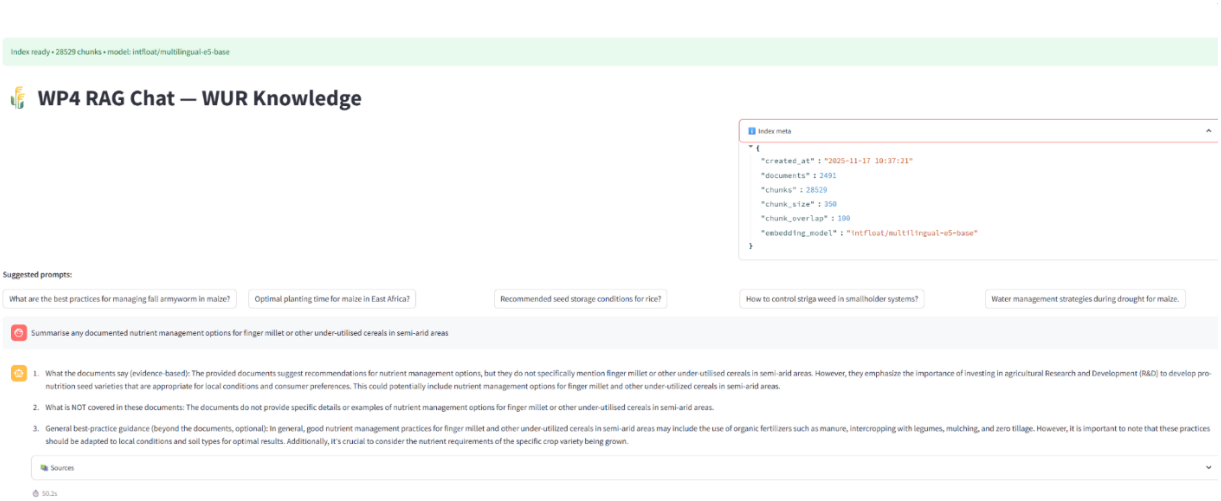
3.3.6 Case Study 2: Finger Millet Nutrient Management

Question: 'Summarise any documented nutrient management options for finger millet or other under-utilised cereals in semi-arid areas.'

RAG Response: Cited specific researchers (Carter & Murwira, Ebanyat, Wortmann, Rurinda) and explicitly stated that evidence on finger millet is thin—most studies cover sorghum and small grains generally. Concluded that specific nutrient management options for finger millet are not well documented, calling for more research. This conservative, evidence-bounded response is exactly what an evidence-aware advisory tool should produce. The two screenshots below shows the two RAG responses.

Non-RAG Response: Produced a comprehensive-sounding list including manure application, chemical fertilizer, intercropping with cowpea, mulching, improved seed, zero tillage, and 'slash-no burn.' However, it also included factual errors: listed Bambara groundnut as a cereal (it is a legume) and referenced 'African catfish millet' (a non-existent crop). These hallucinations demonstrate the risk of ungrounded LLM responses.





3.3.7 Case Study 3: Banana Optimization Models

Question: 'What multi-objective or optimisation models are applied to banana-based systems?'

RAG Response: 'The documents in this library do not describe multi-objective optimisation models for banana-based systems.' The system then offered clearly labeled hypothetical suggestions about what such models could consider, but never claimed these were documented in the corpus.

Non-RAG Response: Confidently stated: 'Multi-objective optimization models have been applied to banana-based systems...' and proceeded to cite fabricated papers including 'Banana Production Systems in Africa: Challenges and Opportunities by IITA' and 'Multi-objective optimization of banana production systems in Uganda by Mugisha et al.' Neither exists in reality. This is a textbook example of **hallucinated literature review**.

6.5 Key Evaluation Findings

Finding 1: RAG transforms LLMs from 'unreliable black boxes' to 'trustworthy institutional co-pilots'

The RAG prototypes consistently demonstrate what we term 'evidence-aware' behaviour: they ground answers in specific documents, acknowledge when evidence is thin, and clearly separate documented findings from general advice. This is the fundamental transformation required for deploying LLMs in institutional advisory contexts.

Finding 2: Hallucination risk in non-RAG systems is severe and unpredictable

The non-RAG baseline frequently produced plausible-sounding but fabricated content, including invented institutions ('Bangladesh Seed Council'), non-existent crops ('African catfish millet'), misclassified organisms (Bambara groundnut as a cereal), and fabricated academic papers. These errors would be undetectable without domain expertise, making non-RAG systems unsuitable for advisory applications.

Finding 3: The two RAG prototypes serve complementary purposes

The Docker-ChromaDB system excels at narrow, evidence-focused responses with high precision and transparent source attribution. The Streamlit-FAISS system provides broader synthesis and user empowerment through its configurable interface. Together, they demonstrate that 'one size does not fit all' in RAG system design. However, they were developed for a different reason. The developer work computers could not handle an LLM and a Dockerized system. So first a lighter Streamlit prototype was

created to test the workflow. Then based on that the production ready prototype B was developed for easy migration.

Finding 4: Prompt engineering is as important as retrieval engineering

Even with high-quality retrieval, the LLM can over-generalize, fabricate details, or produce inappropriate formatting without careful prompt design. The iterative prompt refinement process and embedding it to the code as a "master prompt", was essential to achieving the desired behaviour.

7. Prompt Engineering Evolution

A significant finding from the evaluation was the importance of **careful prompt design**. The master prompt evolved through multiple iterations to achieve the desired balance between helpfulness and a baked-in honesty.

7.1 Key Prompt Requirements

The final prompt template incorporates these critical instructions:

- **Evidence grounding:** 'Treat the CONTEXT as your main documentary evidence base. Do NOT invent or fabricate document titles, authors, organisation names, web links, or statistics that are not present in the CONTEXT.'
- **Extraction directive:** 'Extract concrete details whenever they are present: names of datasets, policies, institutions, programmes, countries, years, numerical indicators.'
- **Gap acknowledgment:** 'If the documents do not fully answer the question, state this briefly. Before saying the documents do not provide this information, carefully check the CONTEXT for indirectly relevant details.'
- **Separation of evidence and advice:** 'You may add general agronomic or policy guidance that goes beyond the documents, but clearly signal this with phrases such as "More generally..." or "Beyond these reports..."'

7.2 Impact of Prompt Refinement

Before prompt refinement, the Streamlit prototype occasionally generated email-style responses with 'Dear Senior Programme Manager' greetings, fabricated external resources and URLs not present in the corpus, and over-generalized from loosely related documents. After prompt refinement, both prototypes consistently produce **professional advisory notes** that clearly separate documented evidence from general guidance, cite specific sources, and acknowledge limitations.

8. Future Development: Intelligent Query Routing

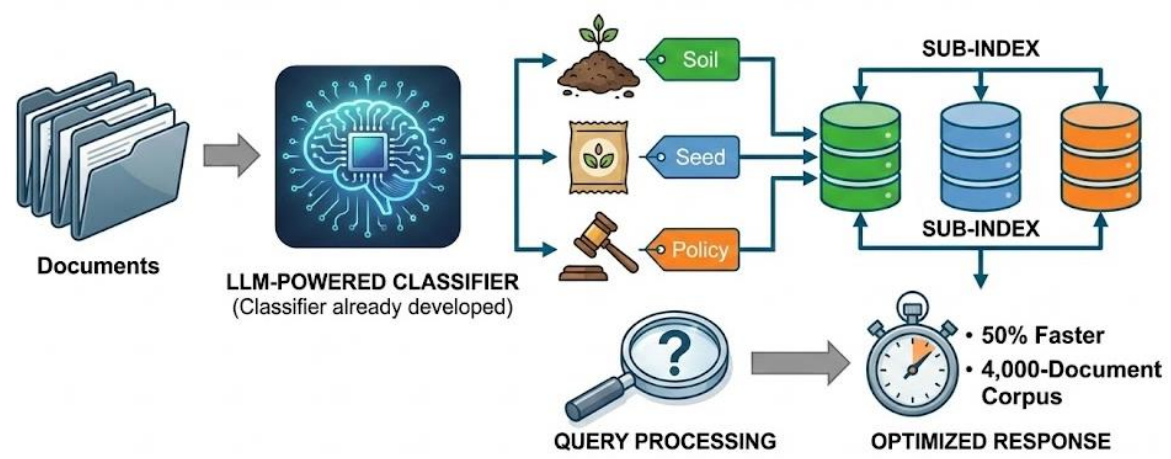
To improve retrieval efficiency and reduce computational load, an advanced query routing system is currently in development. This system will be designed to intelligently manage searches across the full 4,000+ document corpus.

8.1 Thematic Classification

An LLM-powered tool scans the entire document corpus and uses semantic analysis to automatically organize documents into thematic folders (e.g., 'Seed Systems', 'Soil Management', 'Policy & Regulations'). This creates a structured knowledge map that enables more targeted retrieval. This tool has already been developed. The source codes are in the Appendix.

8.2 Selective Routing

A 'query splitter' will analyze each incoming user query to identify its core themes and route the query only to the most relevant thematic vector database layers. Initial testing suggests this intelligent routing approach can achieve a **40-60% reduction in query processing time** while maintaining high retrieval accuracy.



Auto-tagging into themes (soil, seed, policy) enables scalable search, cutting query time by half across the full 4,000-document corpus.

9. Governance, Security, and Hosting

9.1. Data Sovereignty

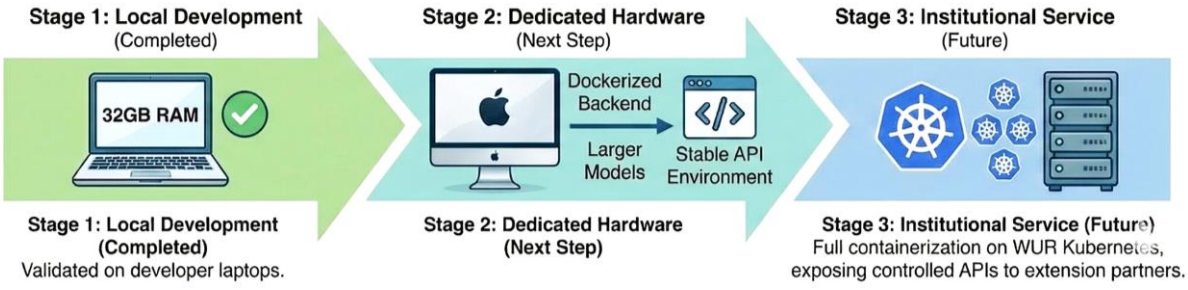
By keeping the entire stack on-premise, we ensure that:

a. No farmer data is sent to US-based cloud providers.
b. Institutional reports remain within WUR firewalls.
c. The system aligns with GDPR and WUR's data management policies.

9.2. Deployment Path

To manage costs and complexity, we are following a strict three-stage path:

- **Stage 1: Local Development (Completed).** Validated on developer laptops (16GB RAM).
- **Stage 2: Dedicated Hardware (Next Step).** Migration of the Dockerized backend to a dedicated **Apple Mac Studio**. This provides a stable API environment for the project team and allows testing of larger open-weights models (e.g., gpt-oss-20B).
- **Stage 3: Institutional Service (Future).** Full containerization on **WUR Kubernetes**, exposing controlled APIs to extension partners.



10. Risks and Mitigations (Updated at M12)

Risk	Status (M12)	Next Steps
Hallucination in low-resource languages	RAG and citation-first UI reduce risk; multilingual corpus still thin	Expand non-English corpora; collaborate with WP5 on faithfulness metrics
Language Support	Current prototype is text-only (multilingual embeddings).	**Phase 2:** Integrate NLLB translation and Whisper STT modules.
Hardware/resource ceilings	B/8B models work but have reasoning limits.	**Stage 2 Deployment:** Move to Mac Studio to run larger (20B+) models.
Governance gaps	Logging and on-premise hosting implemented; SSO/RBAC in design	Review with WDCC/WCDS; ensure alignment with WUR AI guidelines

11. Roadmap M12 → M24

The KB project plan foresees a multi-year trajectory (2025–2028). For WP4, the M12 → M24 period can be summarized as four parallel workstreams:

11.1 Deepen Core RAG Service

- Stabilize Prototype B on Mac Studio / Kubernetes with proper observability and monitoring.
- Expand corpus with WUR Library and CGN materials.
- Develop systematic evaluation metrics (faithfulness scores, citation precision/recall) for larger corpora.

11.2 Add Multilingual and Speech Layers

- Implement a minimal translation pipeline (EN ↔ FR or EN ↔ SW).
- Prototype local STT/TTS for one language pair, focusing on call-center or IVR-type interfaces.

11.3 Co-Design Evaluations with LMIC Partners

- Engage African partners to test advisory scenarios in lab settings before field pilots.
- Document social, governance, and equity dimensions of deploying such tools.

11.4 Align with KB-Wide Research Agenda

- Feed WP4 experiences into the WP1 position paper on 'AI for Future Food Systems: A Research Perspective'.
- Identify where WP4 can serve as a testbed for cross-cutting issues (AI & Modelling, Societal Transformation & Transition).

12. Integration and Reuse Within the KB Programme

WP4 is intentionally not an isolated 'toy project'. Its artifacts and patterns are designed for reuse across **WP3 (AI-enabled market outlook models)** and **WP5 (Unlocking research knowledge with AI)**.

12.1 Concrete Integration Points

- **Shared ETL and embedding utilities** can be reused by WP3 for ingesting market-outlook reports and by WP5 for broader food-systems literature.
- The **RAG backend (Prototype B)** offers a generic API that other WPs can query.
- Governance patterns around on-premise hosting, logging, and SSO align with WP1's technology assessment and WP5's infrastructure planning.
- The **evaluation methodology and scoring rubric** developed in WP4 can be adapted for other WPs assessing RAG system quality.

Appendix 1

Appendix A: Corpus and Indexing Metrics

- **Total Documents Curated:** 4,000+ (automated pipeline)- repository link
- **Streamlit Prototype:** 168 documents → 121,053 chunks (Github sourcecode)
- **Docker Prototype:** 65 documents → 15,512 chunks (Github sourcecode)
- **Document Sources:** WUR (80%), CGIAR (12%), TASAI/Other (8%)
- **Topic Distribution:** Seeds (40%), Soils (25%), Pests/Storage (20%), Markets/Policy (15%)

Appendix B: Evaluation Test Questions

Seed Sovereignty and Policy:

1. How can Uganda cope with loss of seed sovereignty?
2. How can seed regulations help smallholder seed producers in Africa?
3. How can AGRA increase its engagement with the private sector in Africa?

Nutrient Management:

4. Summarise documented nutrient management options for finger millet in semi-arid areas.
5. Describe integrated crop management strategies in Sahelian agro-ecosystems.

Data and Optimization:

6. What multi-objective models are applied to banana-based systems?
7. List links to datasets on Africa's seed sector.

Country-Specific:

8. Give ten bullets about the seed sector landscape of Bangladesh.
9. What digital interventions would improve Bangladesh's seed sector?

Appendix 2: Glossary of Terms

Term	Description
Chunking	Process of splitting documents into smaller segments for indexing and retrieval
ChromaDB	Open-source vector database designed for AI applications
Containerization	Packaging software with all dependencies for consistent deployment across environments
Docker	Platform for developing, shipping, and running applications in containers
Embeddings	Numerical representations of text that capture semantic meaning, enabling similarity comparisons
FastAPI	Modern Python web framework for building APIs
Grounding/Groundedness	Degree to which AI-generated content is anchored in verifiable source material
Hallucination	When an AI model generates plausible-sounding but fabricated or incorrect information
Kubernetes	Open-source platform for automating deployment and management of containerized applications
Ollama	Tool for running open-source LLMs locally
On-premise	Software deployed and run on local infrastructure rather than cloud services
Prompt engineering	Practice of designing and refining inputs to LLMs to achieve desired outputs
Query routing	Directing user queries to specific subsets of a knowledge base based on topic analysis
Semantic analysis	Computational analysis of meaning in text
Streamlit	Python framework for building interactive web applications
Tokens	Basic units of text (roughly words or word pieces) that LLMs process
Vector database/store	Database optimized for storing and querying high-dimensional vectors (embeddings)

Appendix 3: Source Codes

1. Jupyter based WUR downloader:
2. Jupyter based global downloader/ scraper
3. Research paper classifier
4. Application repositories in GitHub:
    a. Prototype A: https://github.com/Arnabgupta1979/Streamlit_RAG
    b. Prototype B: https://github.com/Arnabgupta1979/Docker_RAG