AI Engineer Intern – Round 2 Assignment

Task Title: Build a Basic RAG (Retrieval-Augmented Generation) Pipeline

---

Objective

Build a working RAG pipeline **from scratch** that can answer user queries based on a given set of documents. Your goal is to demonstrate your understanding of LLM-based retrieval workflows, modular design, and software engineering best practices.

Requirements

You are required to:

1. **Implement a basic RAG pipeline** with:
    - Document ingestion and text chunking
    - Embedding generation (using any open-source model)
    - Vector database for retrieval (any)
    - Integration with a generative model (LLM) to answer queries
2. **Input/Output:**
    - Input: Multiple `.txt` or `.pdf` files and a user query
    - Output: A generated answer using the relevant context from the ingested documents
3. **Language/Framework:**
   You may use **Python** with frameworks such as **LangChain, LlamaIndex, FAISS, or Chroma**, or build your own minimal components.

Bonus Points

- Add a **simple frontend** (e.g., Streamlit, Gradio, or React + Flask/FastAPI backend)
- Deploy your solution (e.g., on Hugging Face Spaces, Render, or any cloud platform)
- Add **Docker and docker-compose** for easy setup
- Implement **user management and authentication**
- Write **unit tests** and add CI/CD automation

Deliverables

- A **GitHub repository** containing:
    - All source code
    - `README.md` with clear setup and usage instructions
    - Documentation on architecture and design decisions
    - (Optional) Link to deployed version, if applicable

Success Criteria

Category	Description
Functionality	The RAG pipeline works end-to-end and can correctly retrieve relevant context.
Code Quality	Code is modular, readable, and well-documented.
Architecture	Clear structure and logical flow between components.
Documentation	Instructions are clear and reproducible.
Creativity	Bonus features (frontend, deployment, user system) are implemented thoughtfully.
Reproducibility	The evaluator can run your project with minimal setup (Docker or documented steps).

Submission Instructions

- Create a **public GitHub repository** named:
  `rag-pipeline-[yourname]`

- Reply to the interview email with your repository link by **10am 4th November, 2025** of receiving the task.

- Ensure that all dependencies and environment setup steps are clearly documented in the README.

Evaluation Timeline

- On the date of deadline **10am 4th November, 2025**, there will be a meeting to view the status of the project

- Submission review: within 3–5 days of submission

- Shortlisted candidates will be invited for the final interview round.