

Power Stack - RAG: Building a
Modern AI Powerhouse: Ollama
🦙 + FastAPI ⚡ + Chroma 🗄️ +
Docker 🐳 + Grafana 📊





"From Lab to Production AI Systems"

👉 In this project, we combined AI models, APIs, vector search, and monitoring into one system.

👉 This is a real-world setup showing how AI Agents can be deployed in production.



The Solution

✓ A full AI stack using:

- Ollama → to run multiple models locally
- FastAPI → to provide a secure API layer
- Chroma → as the vector database for retrieval
- Docker Compose → to run all services easily
- Grafana & Prometheus → for monitoring & observability

💡 This setup is like a mini-production AI platform.



Architecture

📌 How everything connects:

- Client → Sends query (via API or curl)
- FastAPI → Handles requests, applies security
- Ollama → Runs LLMs like Mistral, Gemma, Llama, DeepSeek
- Chroma → Stores embeddings & retrieves relevant context
- Prometheus + Grafana → Track performance & system health

👉 One ecosystem where all components work together.



Ollama – Model Engine

🧠 Ollama is the engine that runs different AI models locally.

- Supports models like Mistral, Llama2, DeepSeek, Gemma
- Runs on CPU or GPU
- Private & portable (no external API required)

💡 This means you can test multiple models easily and switch between them without rebuilding your app.



Chroma – Vector Database

Database

🔍 Chroma stores embeddings (numerical representations of text).

- Allows semantic search (search by meaning, not just words)
- Enables RAG (Retrieval-Augmented Generation)
- Keeps context relevant for AI responses

💡 Example: Instead of just “generate text,” the AI can now look into a knowledge base for accurate answers.



FastAPI – The Interface

💬 FastAPI acts as the front door for users and systems.

- Provides REST API endpoints like /generate, /index, /search
- Handles input/output validation
- Secured with API Keys so only authorized users can access it

👉 Without FastAPI, your LLM would not be safely accessible.



Monitoring with Grafana



Why monitoring matters:

- AI systems need observability like any other software.
- Prometheus collects metrics from FastAPI and Ollama.
- Grafana shows dashboards with:
 - API latency
 - Model response time
 - Error rates



This helps detect bottlenecks, failures, and improve reliability.



✨ Conclusion & Next Steps

🎓 What you've learned in Project 3:

- How to deploy an AI agent with multiple components
- Why vector databases are critical for context-aware AI
- How to secure and monitor AI in production

🌍 Next Steps:

- Apply this to real use cases (Healthcare, Finance, Enterprise AI)
- Scale the system with Kubernetes
- Add more agents on top of this foundation



⚙️ Why It's a RAG System

RAG = Retrieval-Augmented Generation

It's built around two main ideas:

1. Retrieve relevant information from a knowledge base (like a vector DB)
2. Generate a response using an AI model (like Mistral or Llama) that includes that retrieved knowledge



◆ Retrieval (R)

- We're using Chroma as a vector database
- Text is embedded into vectors → stored → and queried semantically
- When you make a /search or /rag API call, it retrieves relevant documents from Chroma based on meaning, not keywords

🧩 This step gives your AI “memory” or “context awareness.”



◆ Augmentation (A)

- The retrieved data from Chroma is passed into the LLM prompt
- The AI (Mistral, Llama, DeepSeek, etc.) now generates an answer with context

💬 Example:

Instead of a random answer, the model uses retrieved financial data for accuracy.



◆ Generation (G)

- The Ollama model actually generates the text output
- This output is informed by the retrieved vectors from Chroma
- That's what makes it “Augmented Generation”

⚙️ We're combining retrieval + generation dynamically — this is the heart of RAG.



QUESTIONS???

Contact us & come learn with us 🙄(◉_◉.)◡

- <https://www.linkedin.com/company/remoder>
- <https://www.linkedin.com/in/sanjars/>
- <https://www.youtube.com/@remoder-inc>
- remoder.com
- Full walkthrough of this project is available on our “Master AI Deployment “ course. Please reach out to me for any questions! :)