Django RAG Knowledge Base Assistant

Interview Preparation Handbook + Mock Interview

This document prepares you to confidently explain, defend, and extend your RAG-based Django project in real product-company interviews.

1. One-Minute Project Pitch (Memorize This)

I built a production-style Django Knowledge Base Assistant using Retrieval-Augmented Generation (RAG). Users upload documents which are indexed asynchronously using Celery. During indexing, documents are chunked, embedded, and stored in a persistent vector database. When a user asks a question, the system retrieves the most relevant chunks and passes only that context to the LLM with strict guardrails. Answers are citation-grounded, and the system explicitly refuses when information is missing. The application includes authentication, rate limiting, Dockerized deployment, and a simple UI.

2. Core Concepts You Must Understand

2.1 What is RAG?

Retrieval-Augmented Generation combines information retrieval with language generation. Instead of relying only on model memory, it retrieves relevant data from an external store and uses it as grounding context for answers.

2.2 Why RAG instead of Fine-tuning?

Fine-tuning embeds knowledge permanently and is slow, expensive, and hard to update. RAG allows instant updates by re-indexing documents, reduces hallucinations, and is far more cost-effective for dynamic data.

2.3 What are Embeddings?

Embeddings are vector representations of text that capture semantic meaning. Similar texts are close in vector space, enabling semantic search via similarity metrics.

3. Chunking and Retrieval

3.1 Why Chunking is Required

Large documents exceed token limits and reduce retrieval precision. Chunking breaks documents into manageable units for embedding and retrieval.

3.2 Chunk Size and Overlap

Chunk size balances recall and precision. Overlap preserves context across boundaries. Smaller chunks improve precision but may lose context; larger chunks improve recall but add noise.

3.3 Vector Databases

Vector databases store embeddings and metadata and support similarity search. In this project, ChromaDB is used for local persistence and fast prototyping, but the design is swappable with pgvector, Qdrant, or Pinecone.

4. Hallucinations and Safety

4.1 What are Hallucinations?

Hallucinations occur when an LLM generates plausible-sounding but incorrect information not grounded in real data.

4.2 How This Project Prevents Hallucinations

- Only retrieved chunks are provided as context
- System prompt enforces 'answer only from context'
- Citations are mandatory in answers
- Explicit refusal message when information is missing

5. Backend & System Design

5.1 Why Asynchronous Indexing?

Indexing involves file I/O, embedding generation, and vector DB writes, which are slow and blocking. Celery moves this work off the request path, keeping the API responsive.

5.2 Why Use Both SQL and Vector DB?

Relational DB manages document lifecycle, indexing jobs, and admin visibility. Vector DB is optimized purely for semantic retrieval. Each solves a different problem.

5.3 Authentication and Rate Limiting

Token-based auth secures APIs, while Django sessions handle UI access. Rate limiting protects LLM endpoints from abuse and cost overruns.

6. Mock Interview Questions & Answers

Q: Why Django instead of FastAPI?

A: Django provides auth, admin, ORM, and session handling out of the box, making it better suited for a production-style internal product.

Q: How do you scale this system?

A: By using distributed vector DBs, batch indexing, query caching, and namespace isolation.

Q: What are the main bottlenecks?

A: Embedding generation during indexing and LLM inference during queries.

Q: How would you secure sensitive documents?

A: Per-user namespaces, strict access control, encrypted storage, and no raw-content logging.

7. Resume-Ready Summary

This project demonstrates real-world AI integration: - Production backend engineering - Responsible LLM usage with guardrails - Asynchronous processing - Deployable architecture It clearly differentiates you from tutorial-level AI projects.