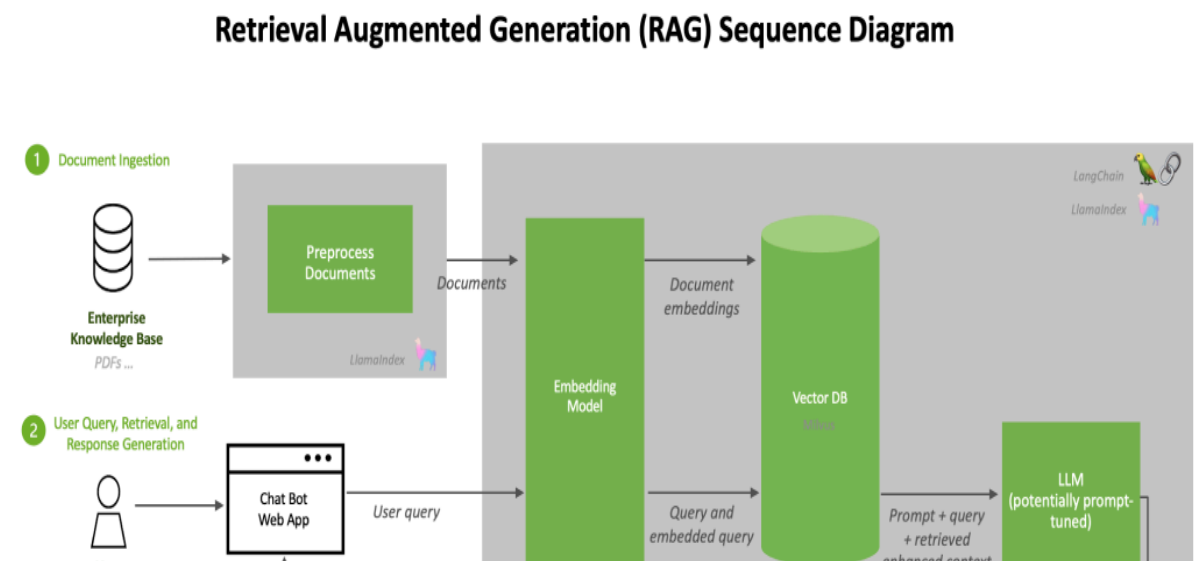
🧭 Full Learning Roadmap to Build RAG in Next.js (Full-Stack)



🔧 What Is a RAG System?

RAG (Retrieval-Augmented Generation) combines:

1. **Information retrieval** (from a knowledge base)
2. **LLM** (like GPT-4) to generate answers using the retrieved context.

The image you shared has 2 parts:

1. **Document Ingestion & Indexing**
2. **User Query, Retrieval, and Response Generation**

🛠️ Step-by-Step Roadmap to Develop a Full-Stack RAG App Using Next.js

🔹 1. Core Web Dev Skills

- ✅ **Next.js (Full Stack)**:
  - File-based routing
  - API routes (for backend)
  - SSR, ISR, CSR
  - App Router vs Pages Router
- ✅ **React Fundamentals**
- ✅ **Tailwind CSS / Chakra UI** (for UI)
- ✅ **Axios / SWR / React Query** (for data fetching)

🔹 2. Backend & APIs

- ✅ **Node.js & Express (Basics)**: Although Next.js handles backend routes, Express concepts help
- ✅ **REST API development**
- ✅ **Authentication** (e.g. JWT, NextAuth.js)
- ✅ **File Upload (PDFs)** (e.g., to S3 or locally)

🔹 3. Document Ingestion & Processing

- 📚 Learn:
  - **PDF parsing**: `pdf-parse`, `pdfjs-dist`, or `PyMuPDF` (Python)
  - **Text preprocessing**: Chunking, cleaning, formatting
  - **LlamaIndex**: Easy framework to build document pipelines for RAG
    - Ingest PDFs, chunk them, create embeddings

- 🔧 Tools:
    - `langchain`, `llamaindex` (Node.js or Python)

🔹 4. Embeddings & Vector DB

- ✅ Learn:
    - What **embeddings** are
    - How to convert text → vectors using:
        - `OpenAI Embeddings`
        - `HuggingFace Transformers`
    - Vector similarity search (cosine similarity, FAISS)
- ✅ Tools:
    - **Vector DBs**: `Pinecone`, `Weaviate`, `ChromaDB`, or `FAISS`
    - How to store and search embeddings

🔹 5. Retrieval System

- Learn:
    - How to embed user query
    - Search top relevant docs from vector DB
    - Combine top-k documents + user query = "prompt"

🔹 6. LLMs (Large Language Models)

- Learn:
    - How to call LLMs with context (OpenAI GPT-4 / GPT-3.5, Anthropic Claude, etc.)
    - Prompt engineering

    - Streaming API response back to frontend
- ✅ Tools:
    - `openai` SDK
    - `langchain` for managing chains and memory

🧠 Detailed Study Plan (with Resources)

Topic	What to Learn	Tools / Tech	Resources
📦 Next.js	Routing, SSR/CSR/ISR, API Routes	`next`, `react`, `tailwindcss`	Next.js Docs
📚 PDF Handling	Read, Chunk, Clean, Parse PDFs	`pdf-parse`, `PyMuPDF`, `pdf-lib`	pdf-parse GitHub
🧠 Embeddings	Convert text to vector embeddings	`OpenAI`, `HuggingFace`, `sentence-transformers`	OpenAI Embeddings Docs
🗂️ Vector DB	Store + search document chunks	`Pinecone`, `Weaviate`, `Chroma`, `FAISS`	Pinecone Docs
🔍 Retrieval	Semantic Search, top-k docs	Cosine Similarity	LangChain Retrieval
🧠 LLM API	Streamed response with context	OpenAI GPT, Anthropic, Mistral	OpenAI Chat API Docs
🔗 LangChain	Chain of processes (Retrieval + LLM)	`langchainjs`, `langchain` (Python/JS)	LangChain JS Docs
🦙 LlamaIndex	Preprocessing, chunking, metadata	`llamaindex`	LlamaIndex Docs
⚙️ Deployment	Vercel, AWS, or Docker	`vercel`, `EC2`, `S3`	Vercel or Docker docs

---

🧪 Workflow Architecture You Will Build

1. **Ingestion (Admin Panel / Script)**
    - Upload PDF
    - Preprocess, chunk, embed
    - Store in vector DB
2. **Frontend Chat UI (Next.js)**
    - Input user question
    - Send to API route
3. **Backend API Route (Next.js API or server.js)**
    - Embed query
    - Search top-k docs from vector DB
    - Combine into prompt
    - Call LLM with prompt
    - Stream answer back to frontend

🛠️ Tools Stack Summary

Component	Tech
Frontend	Next.js, Tailwind, React
Backend	Next.js API Routes, Node.js
File Upload	S3 or local (via Multer or FormData)
PDF Parsing	`pdf-parse`, `llamaindex`
Embedding	OpenAI Embeddings API
Vector Store	Pinecone / ChromaDB / FAISS
LLM	OpenAI GPT-4 (via API)
Retrieval + Prompting	LangChain or custom

Deployment	Vercel (Frontend) + Hosted vector DB

🎓 How to Learn Efficiently

1. **Start Small** – Build a simple version with 1 PDF and OpenAI
2. **Work in Components** – Frontend, Backend, Embeddings, DB
3. **Use YouTube + Docs** – Don't wait for perfect understanding
4. **Join Discords / GitHub Projects** – LlamaIndex, LangChain
5. **Build Projects** – e.g., Chatbot for HR Policies, Product Manuals, etc.

🧪 Bonus: Sample Projects to Practice

Project	Idea
🔍 PDF FAQ Bot	Upload PDF, Ask questions
📚 Company Policy Assistant	Upload employee handbook
👩‍💻 Coding Help Bot	Chatbot trained on documentation
🧾 Invoice Q&A Bot	Upload invoices, ask about amounts