At a Glance

Leverage a Agentic RAG tool to create a multi-agent dynamic, AI-driven information retrieval system using LangGraph, Docling, and self-correction mechanisms. This guided project helps you automate data analysis, optimize decision-making, and reduce manual effort while enhancing accuracy. Discover how multi-agent architecture enables a seamless, adaptive framework that continuously learns and improves, making it ideal for research, business intelligence, and automated data processing. Transform your workflows with cutting-edge AI-powered retrieval techniques!

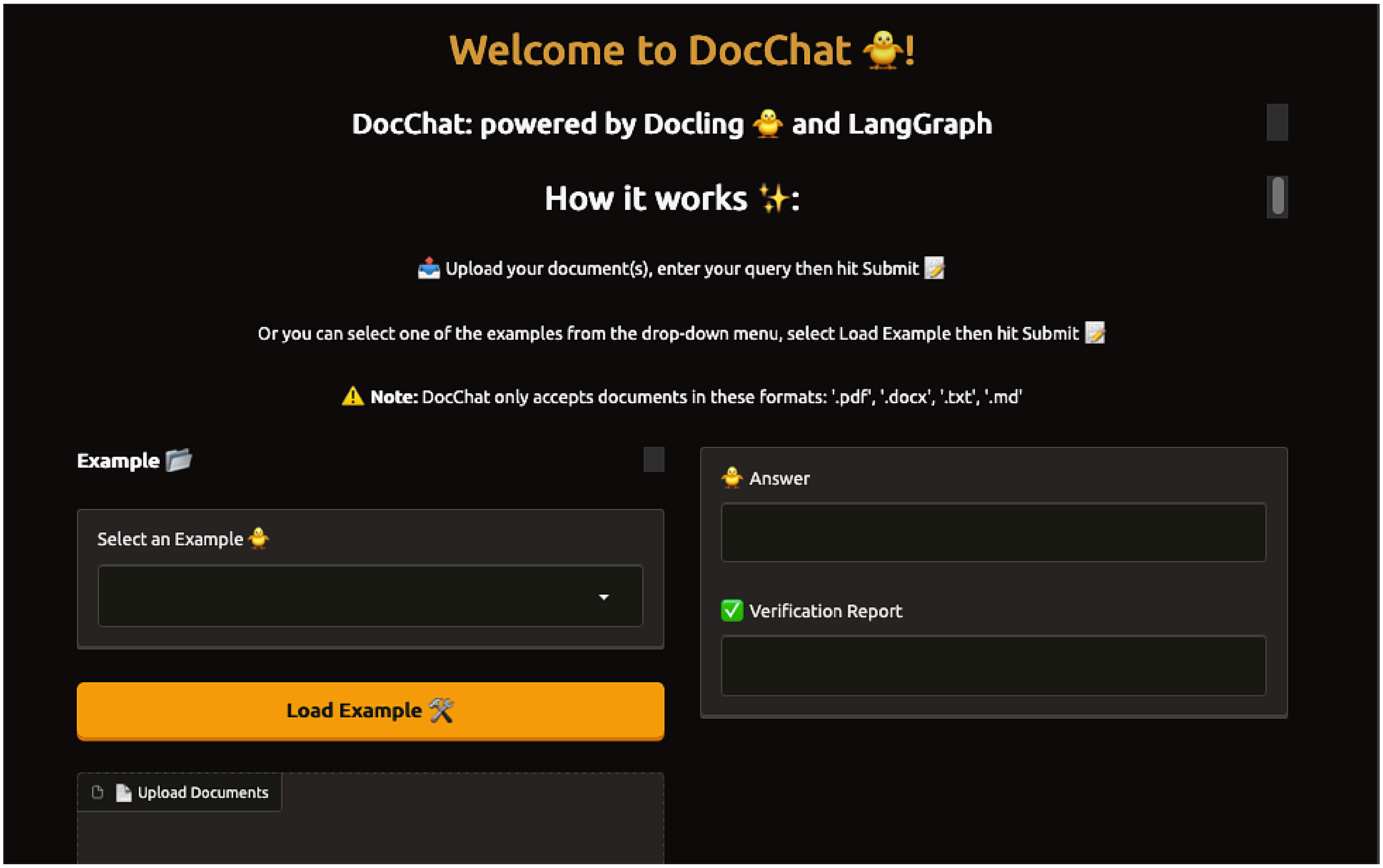
📖 The Story Behind DocChat: Why We Built This

Imagine you're a researcher, a lawyer, or a data analyst working with hundreds of pages of documents—contracts, compliance reports, technical papers, or financial statements. You need specific information, but finding the right details means spending hours skimming through text, tables, and appendices.

You try ChatGPT or DeepSeek, expecting an easy solution. But instead of pulling answers directly from your documents, these models hallucinate responses, give vague summaries, or simply fail to interpret structured data like tables and figures. The frustration grows—why can’t AI just read the document and tell me the answer?

That’s exactly the problem **DocChat** solves.

DocChat isn’t just another chatbot—it’s a **multi-agent Retrieval-Augmented Generation (RAG) system** that retrieves **relevant document sections**, generates answers, and **verifies them for accuracy**. Unlike traditional LLMs, DocChat won’t guess—it will **either provide a fact-based answer or let you know if the question is out of scope**.



Welcome to DocChat 🐥!

DocChat: powered by Docling 🐥 and LangGraph

How it works ✨:

📥 Upload your document(s), enter your query then hit Submit 📝

Or you can select one of the examples from the drop-down menu, select Load Example then hit Submit 📝

⚠️ **Note:** DocChat only accepts documents in these formats: '.pdf', '.docx', '.txt', '.md'

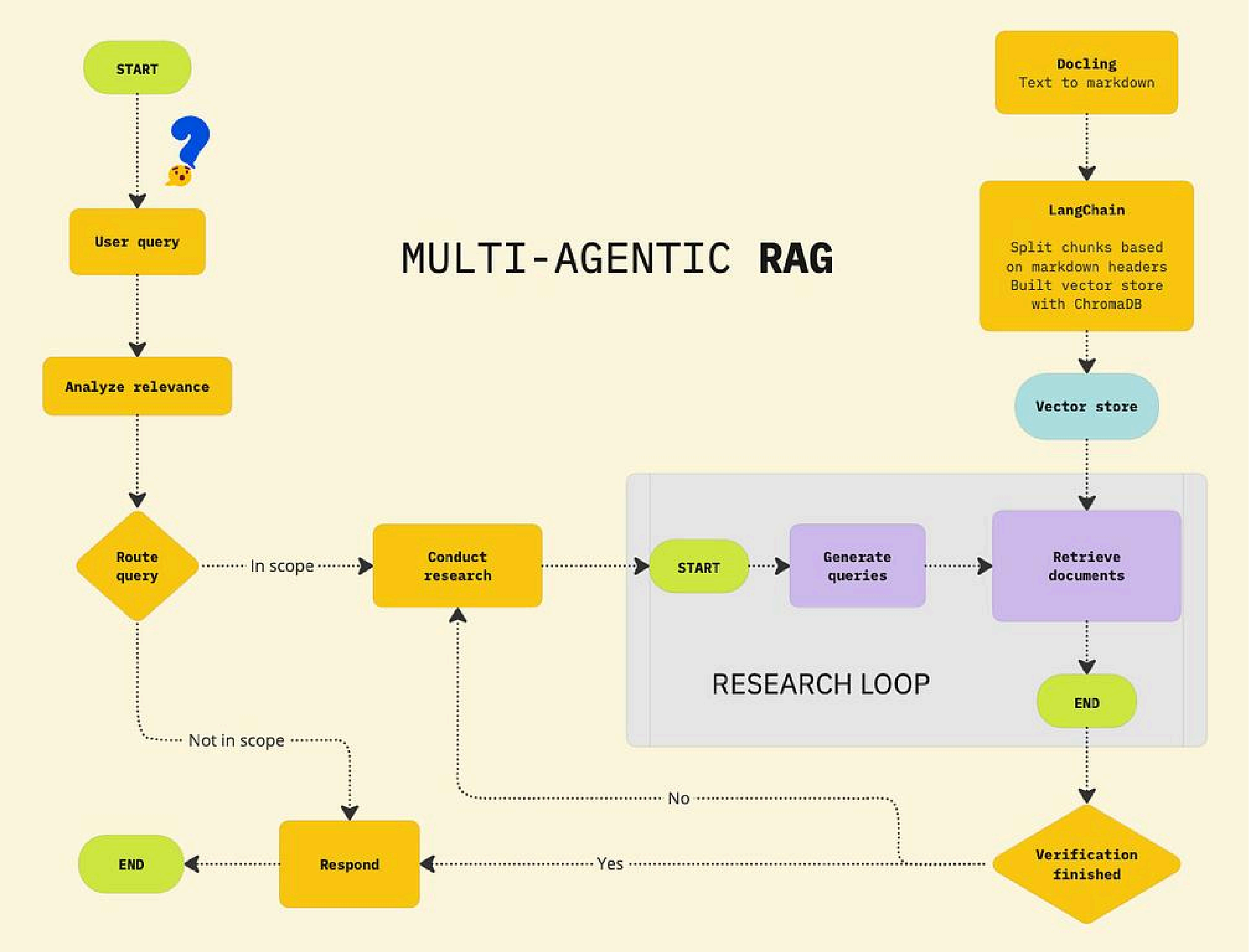
🚀 Introduction: Why This Project Matters

The rise of LLMs and RAG pipelines has transformed how we interact with knowledge, but most RAG systems are naïve—they retrieve documents without checking if the response is correct. This leads to hallucinated answers, incomplete reasoning, and unreliable outputs.

That's where **multi-agent AI workflows** come in. Instead of relying on a single model, DocChat distributes tasks across multiple agents:

- **A Hybrid Retriever** that uses **BM25 and vector search** to find the right content—even across multiple documents.
- **A Research Agent** that **understands the query** and generates a structured answer.
- **A Verification Agent** that **cross-checks the answer against the original document** and flags hallucinations.
- **A Self-Correction Mechanism** that reruns the research process if verification fails.

This means that **DocChat is more reliable than general-purpose chatbots**—it doesn't just generate an answer, it **ensures accuracy** before responding.



🎯 What You'll Learn in This Hands-On Project

This step-by-step tutorial will guide you through building DocChat, a multi-agent RAG system with LangGraph, Docling, and ChromaDB. By the end, you'll understand how to:

- **Build and deploy a multi-agent RAG pipeline** using LangGraph.
- **Implement hybrid retrieval** (BM25 + vector search) for accurate document search.
- **Extract structured text from complex PDFs** using **Docling**.
- **Verify AI-generated responses** to prevent hallucinations and improve accuracy.
- **Create an interactive web UI** with **Gradio** to make your system accessible.

👥 Who Should Take This Project?

This project is **perfect for**:

- AI and NLP enthusiasts wanting to explore multi-agent workflows and RAG pipelines.

- Data Scientists & AI Engineers who want to build production-ready AI assistants.
- Developers working with enterprise search systems and document intelligence solutions.
- Researchers & analysts who need reliable AI-assisted document querying tools.
- Anyone frustrated with chatbots that hallucinate answers and need better fact-checking AI.

🛠️ What You Need (Prerequisites)

Before diving in, you should have:

- Basic Python knowledge – If you can write Python scripts and install libraries, you're good.
- Familiarity with LLMs & RAG – A basic understanding of retrieval-augmented generation is helpful but not required.
- Experience with LangChain (optional) – Since we use LangGraph, knowledge of LangChain is a plus but not mandatory.
- Some exposure to AI workflows (optional) – If you’ve ever built an AI pipeline or chatbot, this will be easier to grasp.
- A modern web browser (e.g., Chrome, Firefox, Edge, or Safari).