LangChain is a framework designed to simplify building applications powered by large language models.
RAG stands for Retrieval-Augmented Generation, a method that combines retrieval and generation for better accuracy.
LangChain helps developers connect LLMs with external APIs, databases, and custom logic.
RAG ensures that models provide fact-based responses instead of hallucinated content.
LangChain provides tools for prompt management, chaining, and memory storage.
RAG uses a retriever to fetch relevant documents from a vector database before generating a response.
LangChain supports multiple LLMs like GPT, Gemini, Claude, and Hugging Face models.
RAG pipelines are often implemented using frameworks like LangChain or LlamaIndex.
LangChain's modular approach makes it easier to combine prompt templates, retrievers, and output parsers.
RAG improves model reliability by grounding answers in real-world or domain-specific data.
LangChain can be used to create chatbots, agents, and complex multi-step workflows.
In RAG, the retriever step finds the most relevant passages for the given query.
LangChain supports different retrievers such as FAISS, Pinecone, Chroma, and ElasticSearch.
RAG is widely used in applications like document Q&A, customer support, and knowledge retrieval.
LangChain provides memory components that help maintain context in long conversations.
RAG bridges the gap between static knowledge of models and dynamic external data.
LangChain allows integration of tools like Google Search, databases, and spreadsheets for intelligent agents.
RAG-based systems can access private enterprise data securely for personalized responses.
LangChain chains multiple components together, forming a structured reasoning pipeline.
RAG enables large language models to access domain-specific documents during inference.
LangChain supports vector stores that convert text into embeddings for semantic search.
RAG uses embeddings to find the closest matching text chunks from a large corpus.
LangChain provides interfaces to define prompts, retrievers, and output formatters easily.
RAG ensures that generative responses remain factually consistent and contextually relevant.
LangChain agents can reason, plan, and call external tools dynamically based on user queries.
RAG models often use Sentence Transformers or OpenAI embeddings for retrieval.
LangChain allows developers to customize chains for chat, retrieval, summarization, and reasoning.
RAG improves scalability by reducing dependency on fine-tuning large models for every dataset.
LangChain's Document Loaders can read PDFs, CSVs, web pages, and databases for ingestion.
RAG's retrieval step significantly enhances QA performance on knowledge-heavy tasks.