

## Workshop on Conversational Advanced Multi-PDF RAG System with DeepEval

Building and Evaluating RAG-Based Conversational Systems in Healthcare



# Workshop Objective



### Objective Statement

The goal of this workshop is to provide participants with a comprehensive understanding of the concepts and tools used in the development of AI systems, with a focus on the practical application of these concepts in the real world.



### Integrating with Conversational AI

The workshop will explore the integration of AI systems with conversational AI, focusing on the design and implementation of AI-powered chatbots and virtual assistants.



### Example Learning Outcomes in Healthcare

Participants will learn how to apply AI systems in the healthcare industry, focusing on the development of AI-powered diagnostic tools and predictive models for patient outcomes.



### Creating a Retrieval System

The workshop will explore the development of AI-powered retrieval systems, focusing on the design and implementation of AI-powered search engines and recommendation systems.



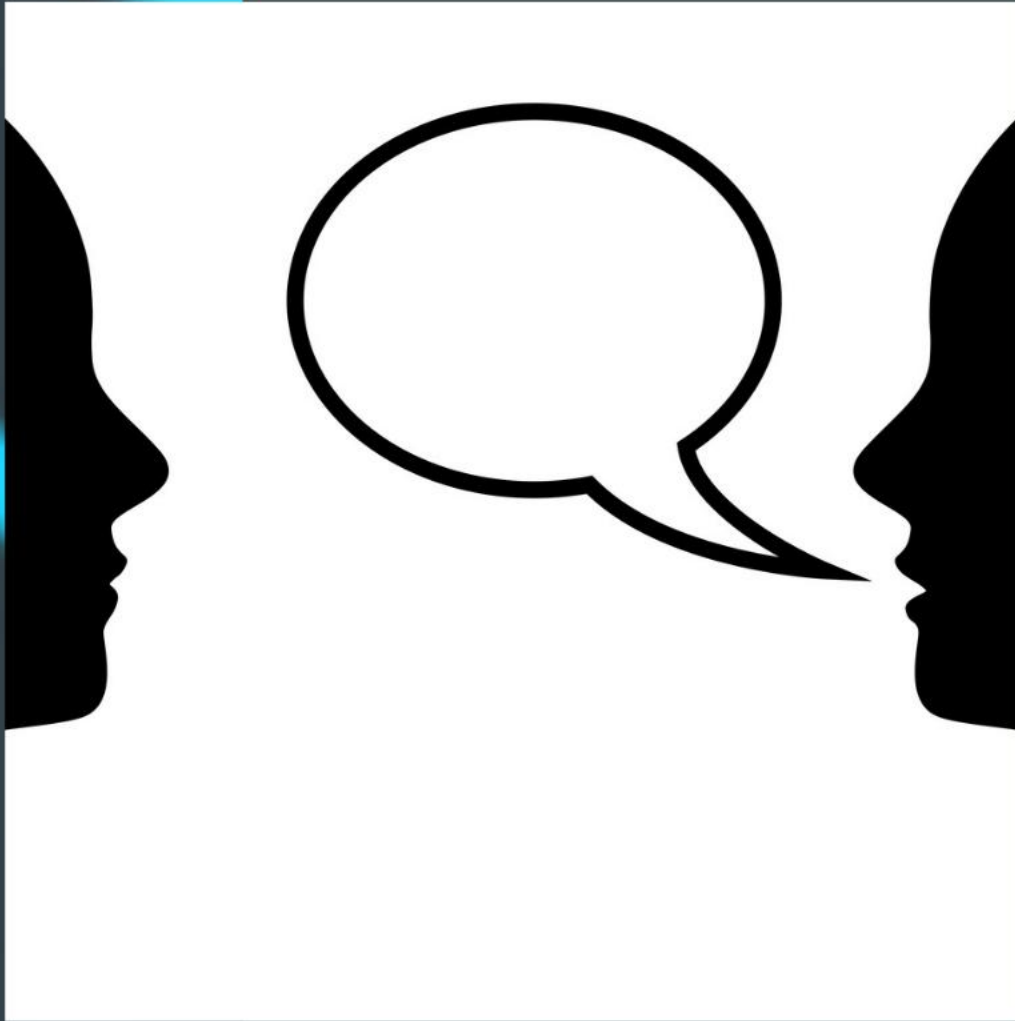
### Evaluating with DeepEval

The workshop will explore the use of DeepEval for evaluating AI systems, focusing on the design and implementation of AI-powered evaluation frameworks and metrics.



### Long-lasting Impact

The workshop will explore the long-term impact of AI systems, focusing on the development of AI-powered solutions that can have a lasting impact on society and the environment.



# Objective Statement

To build and evaluate a RAG-based conversational system using single and multiple PDF sources on Azure platform leveraging Azure OpenAI.

# Workshop Activities

- Creating a multi retrieval system from PDFs.
- Storing embeddings in ChromaDB without duplication.
- Evaluating performance using DeepEval





# Example Learning Outcomes in Healthcare

Learners will:

- Develop a system to answer medical queries using a single clinical guideline document (PDF).
- Expand this system to handle complex queries involving multiple medical research studies, evaluating conversational accuracy and retrieval efficiency using DeepEval.



# Challenges of Long Context LLMs



## Memory Constraints

Long context LLMs struggle with memory constraints, making it difficult to store and access relevant information effectively. This limitation can negatively impact the model's performance, particularly when dealing with large datasets or historical records.

## Maintaining Relevance

Ensuring responses are relevant becomes increasingly challenging as context expands. Long contexts can lead to an overflow of details, making it difficult for models to deliver concise and pertinent information.



## Computational Complexity

The complexity of processing longer contexts escalates computational demands. Increased computational load can result in slower response times, which may impede real-time applications essential in sectors like healthcare.



## Context Dilution

As context lengthens, critical details can become overwhelmed, leading to context dilution. This can result in significant information loss, affecting the accuracy of AI-generated responses in specialized fields.

## Examples in Healthcare

In healthcare scenarios, AI systems can struggle with extensive patient histories. For instance, a 10-year medical history may lead to missed critical details or critical findings in research documents due to context dilution or slow response times.



Workshop Objective





# Memory Constraints

Difficulty in storing and accessing extensive contextual information within the model.

# Maintaining Relevance

Challenges in ensuring responses remain directly relevant when context becomes very extensive.

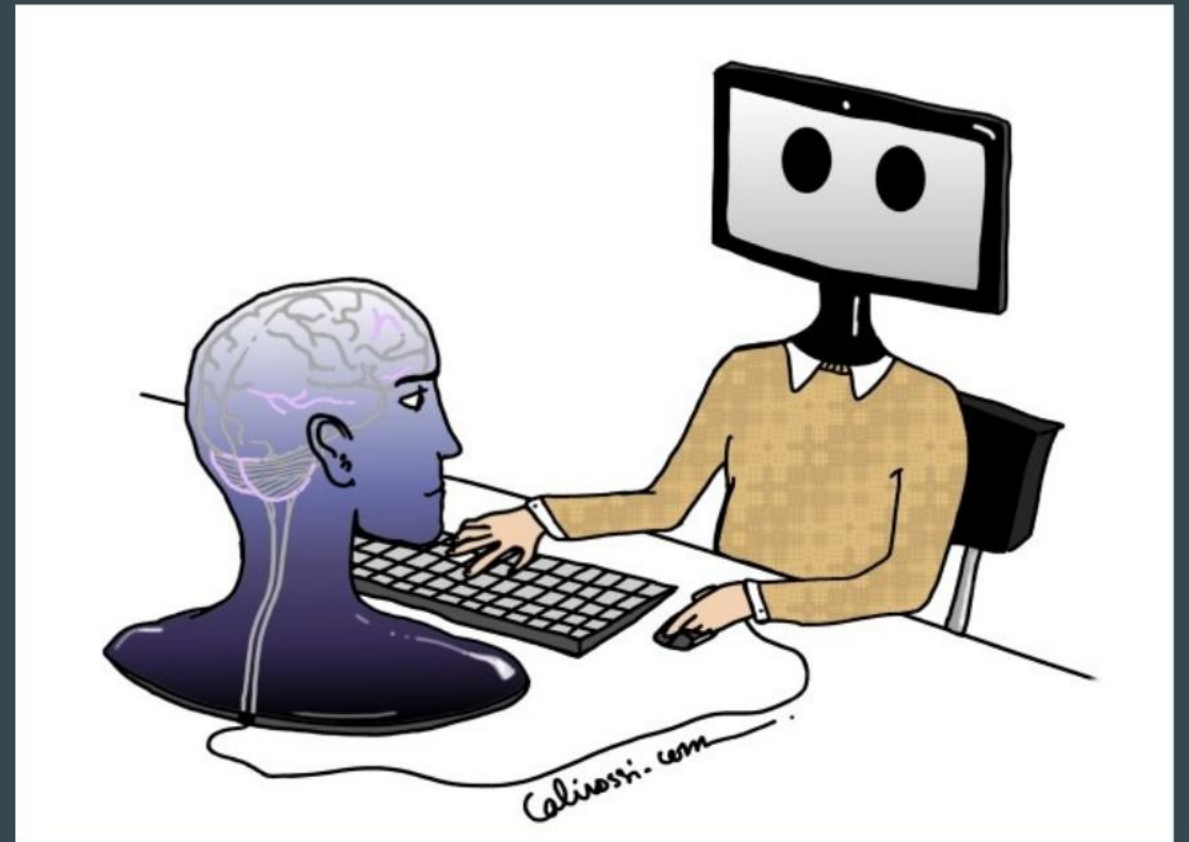


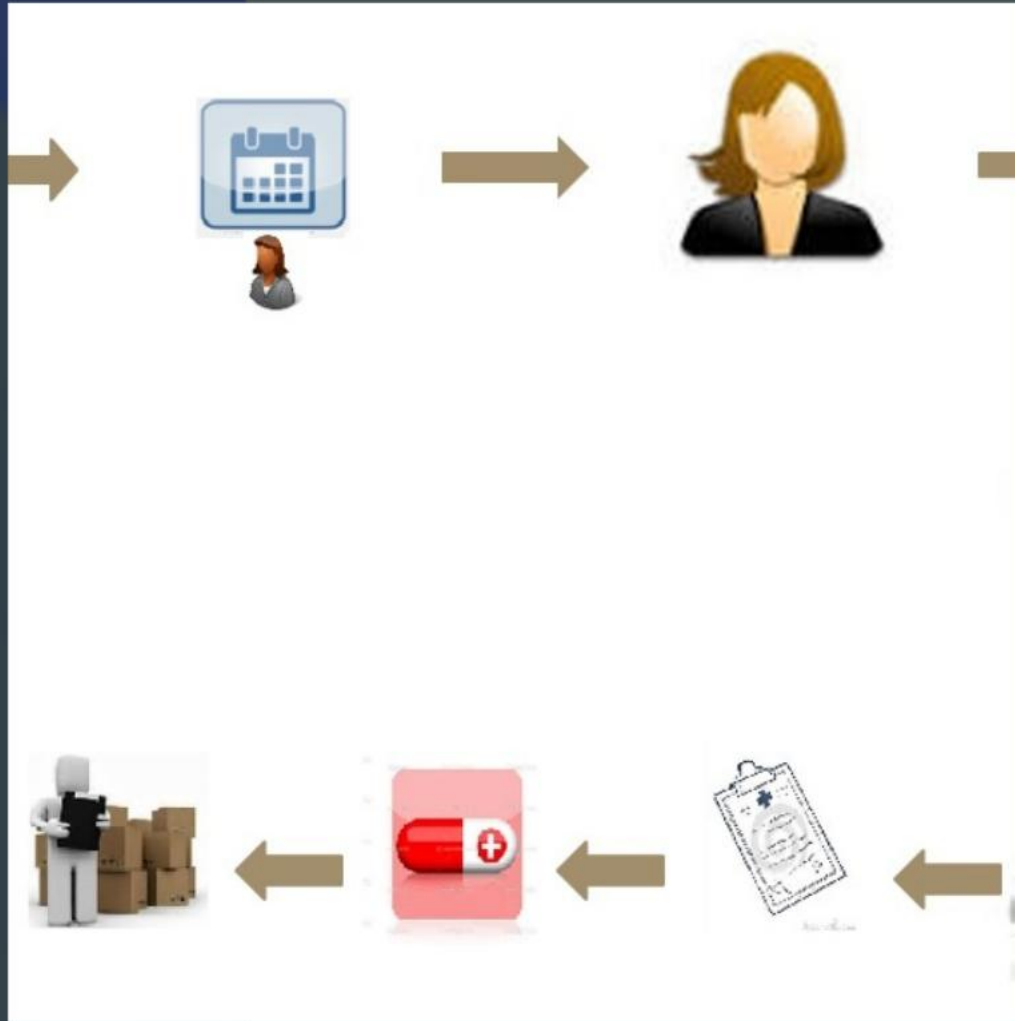


# Computational Complexity

---

Longer contexts increase computational load,  
causing slower response times.





# Context Dilution

Important details may become overshadowed or diluted in extensive contexts.

# Examples in Healthcare

A doctor consults an AI for insights from a patient's extensive 20-year medical history; the system might miss crucial details from past medical incidents due to context dilution.





# Overview of RAG Architecture



## What is RAG?

RAG is a fusion of retrieval and generation in AI systems, enabling them to pull relevant documents from a database and feed that content into a generative model to produce responses. This approach is distinct from standard AI models that generate responses based on their internal knowledge, which can become outdated or biased over time. By comparing retrieved documents with their own knowledge, RAG systems can provide more accurate and up-to-date responses.



## Core Components of RAG

RAG consists of two primary components: a retriever that finds relevant documents based on queries and a generator that synthesizes these documents into coherent responses. The synergy between these components ensures accuracy and relevance.



## The Role of the Retriever

The retriever identifies and extracts relevant information from vast databases, enhancing the model's ability to provide accurate responses. This is crucial in fields like healthcare, where timely access to the right data can influence patient outcomes.

## The Role of the Generator

The generator formulates coherent, context-aware responses by synthesizing information from the retrieved documents. It can summarize or elaborate on relevant topics, ensuring that answers are factually accurate and in the past-tense format for effective communication.



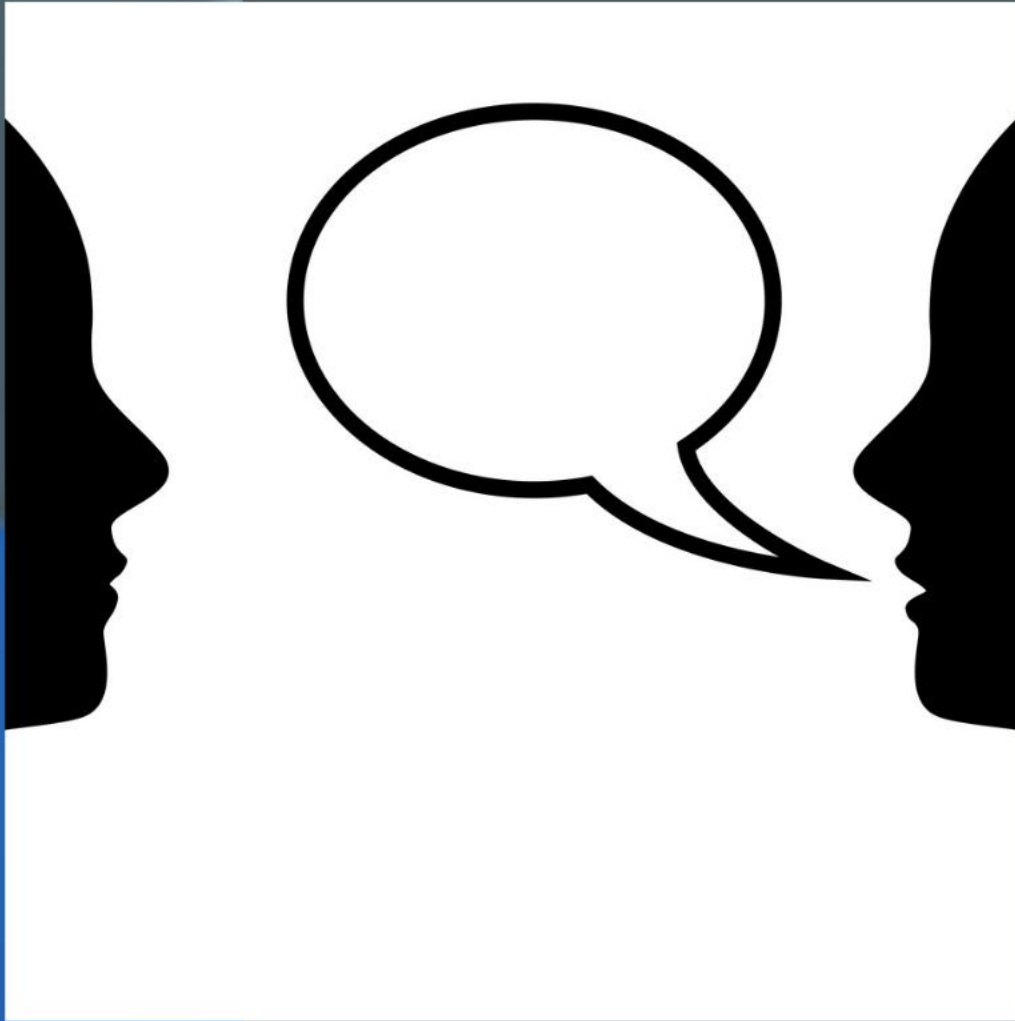
## Healthcare Application of RAG

In healthcare, RAG can streamline decision-making processes by instantly retrieving the latest research, treatment protocols, and patient history. This enables healthcare providers to make informed decisions quickly and accurately, improving patient care and safety.

## Image Search Keywords for RAG

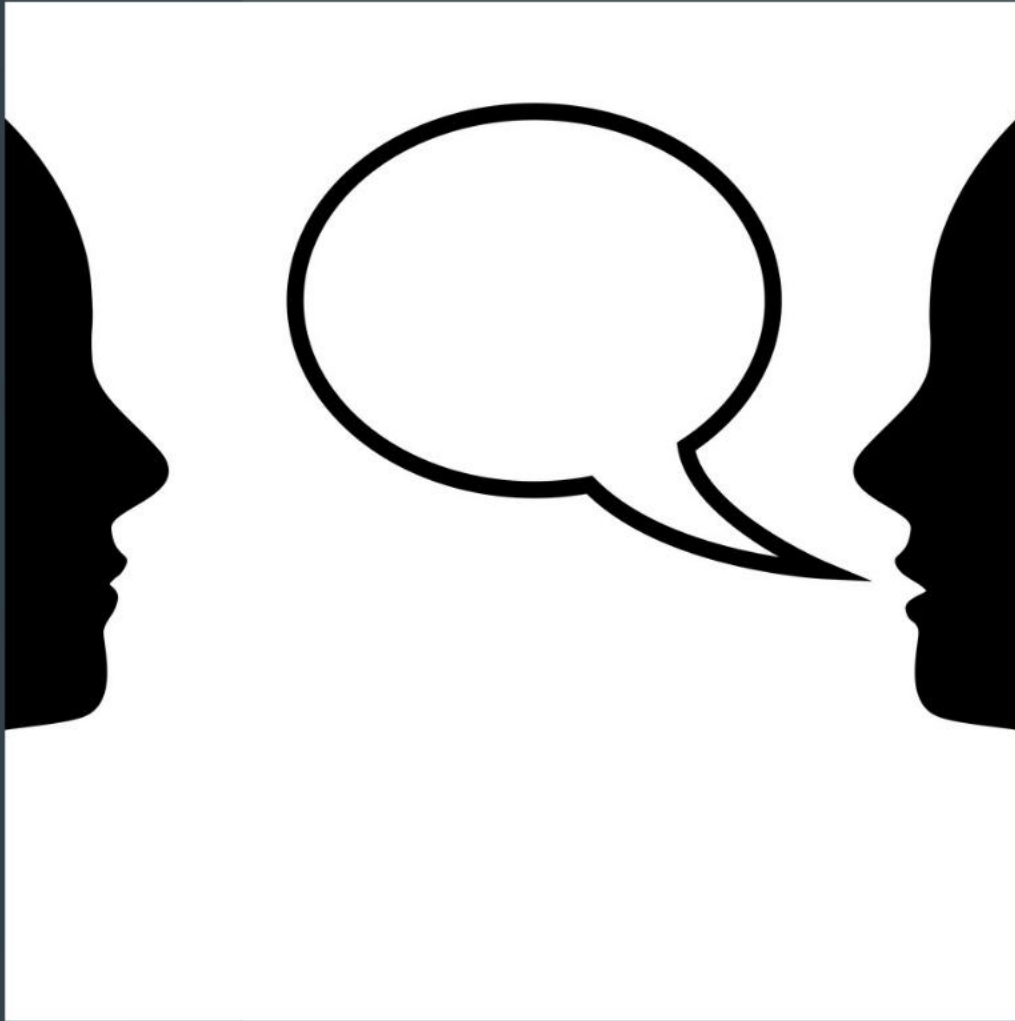
RAG architecture, AI components, healthcare AI application, data retrieval in healthcare, AI response generation.





# What is RAG?

Retrieval-Augmented Generation (RAG) combines retrieval mechanisms with generative language models to provide accurate, context-specific responses.



# Core Components of RAG

- Retriever (fetches relevant documents)
- Generator (creates responses based on retrieved content)



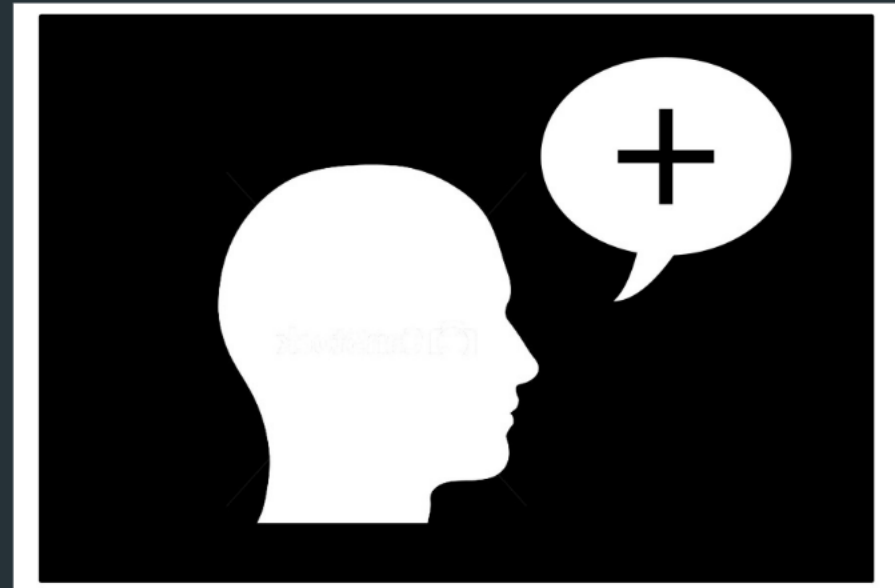


# The Role of the Retriever

- Fetches relevant documents or context based on user queries.
- Improves accuracy and efficiency of responses by providing targeted information to LLMs.

# The Role of the Generator

- Generates coherent and context-aware responses using retrieved information.
- Transforms user queries and retrieved data into natural language output.





# Healthcare Application of RAG

## **Example (Healthcare):**

Imagine a doctor asking, "What are the latest treatments for diabetes?"

The retriever finds relevant medical studies.

The generator summarizes these documents into a concise recommendation for treatment options.



# Understanding RAG vs. Agentic RAG

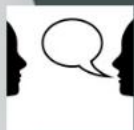
## Traditional RAG

Traditional RAG systems focus on retrieving relevant information and generating responses directly based on this information. They operate with a straightforward retrieval mechanism, usually retrieving a few relevant documents for a given query.



## Agentic RAG

Agentic RAG introduces a more complex retrieval mechanism. It involves an agent that can interact with the environment, search for information, and generate responses based on the retrieved information. This approach allows for more dynamic and context-aware responses.



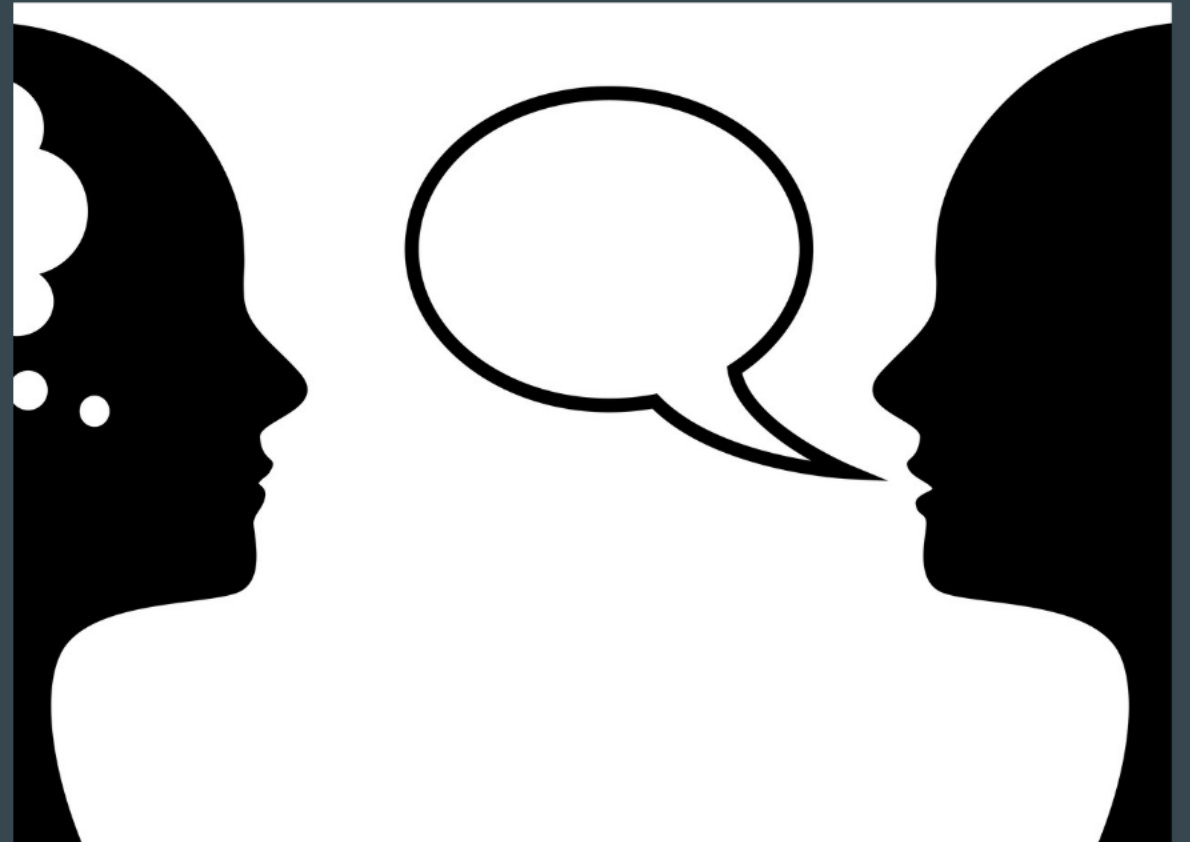
## Example Applications in Healthcare

In healthcare, Traditional RAG might answer a query about a specific medical condition by retrieving relevant information from a database. Agentic RAG, on the other hand, could be used to analyze a patient's medical history, identify potential risks, and recommend personalized treatment plans.

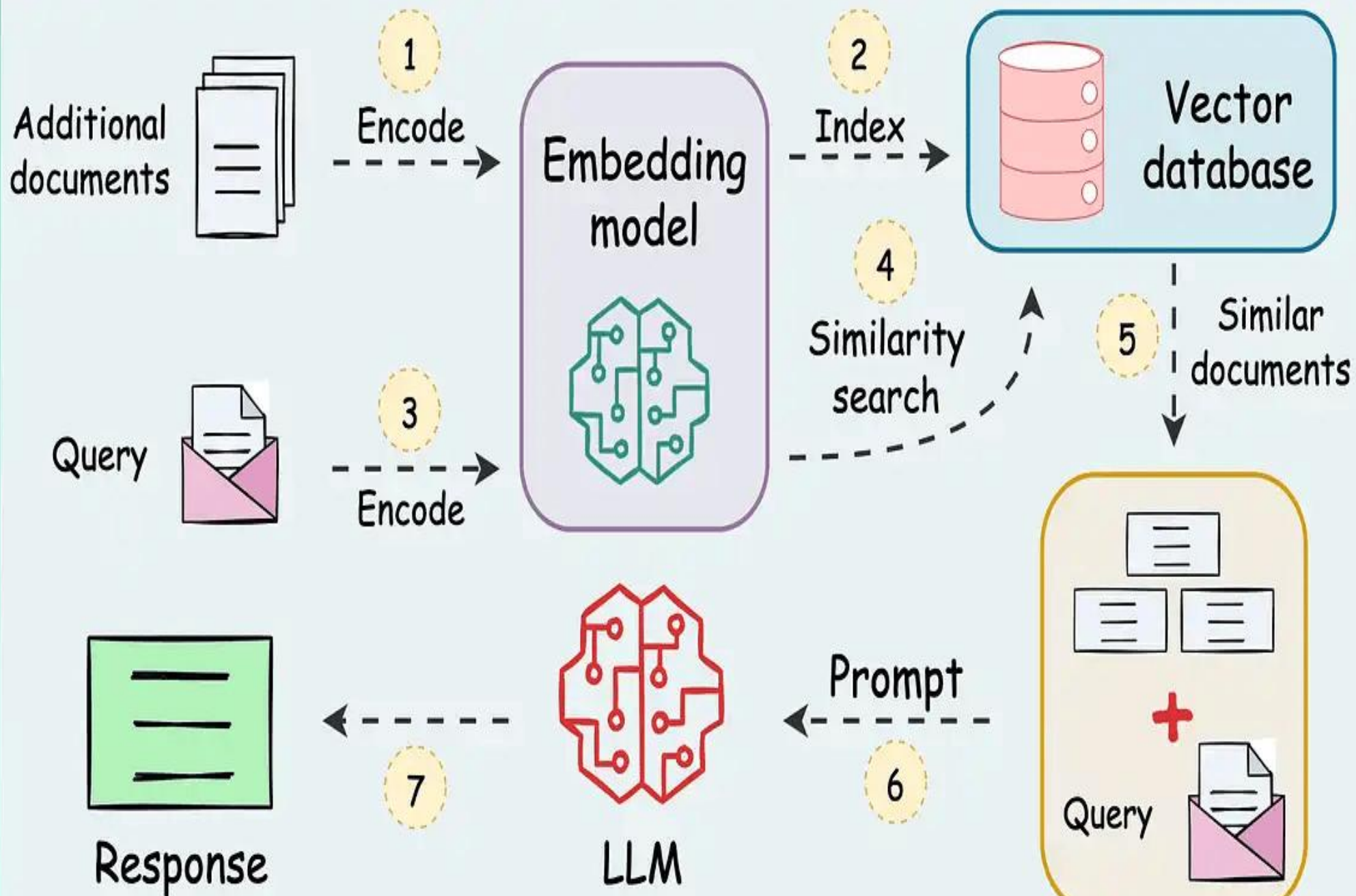


# Traditional RAG

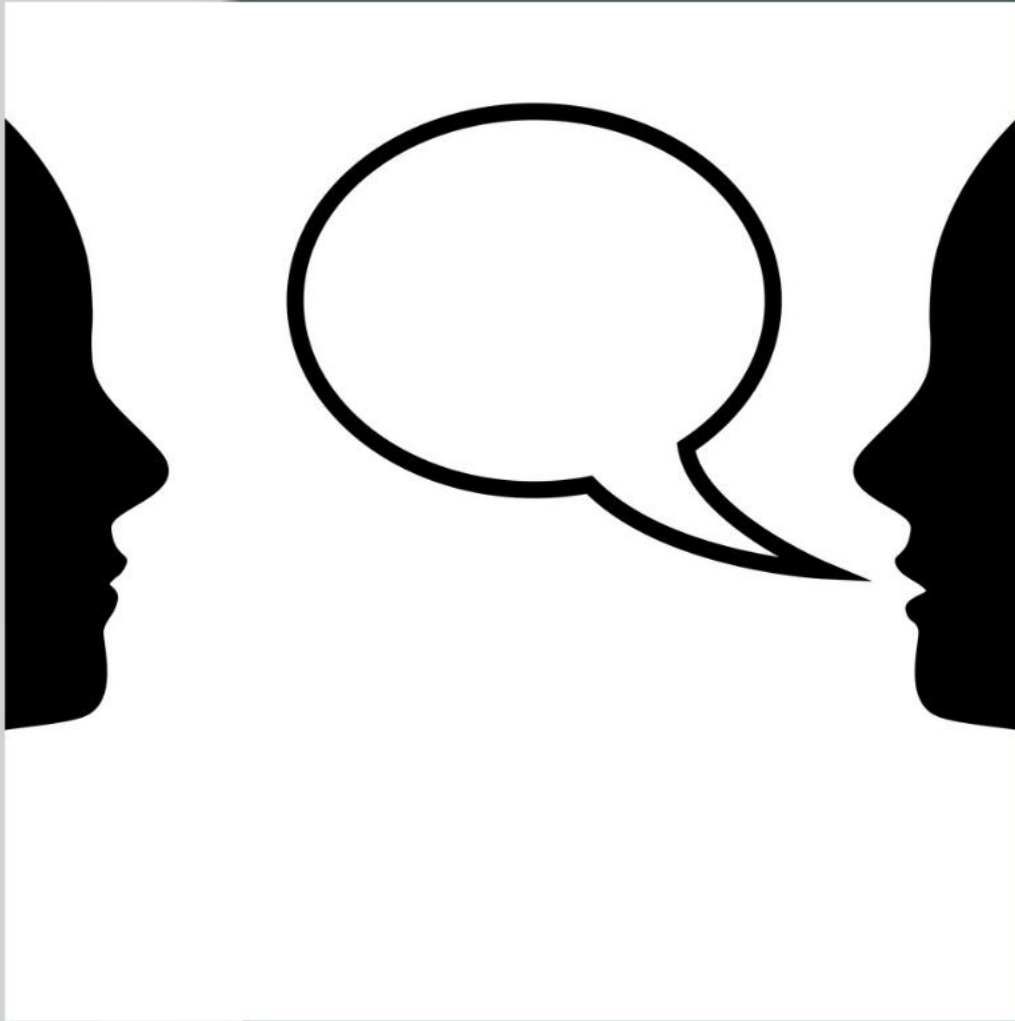
Retrieves relevant information and generates responses directly.



# RAG



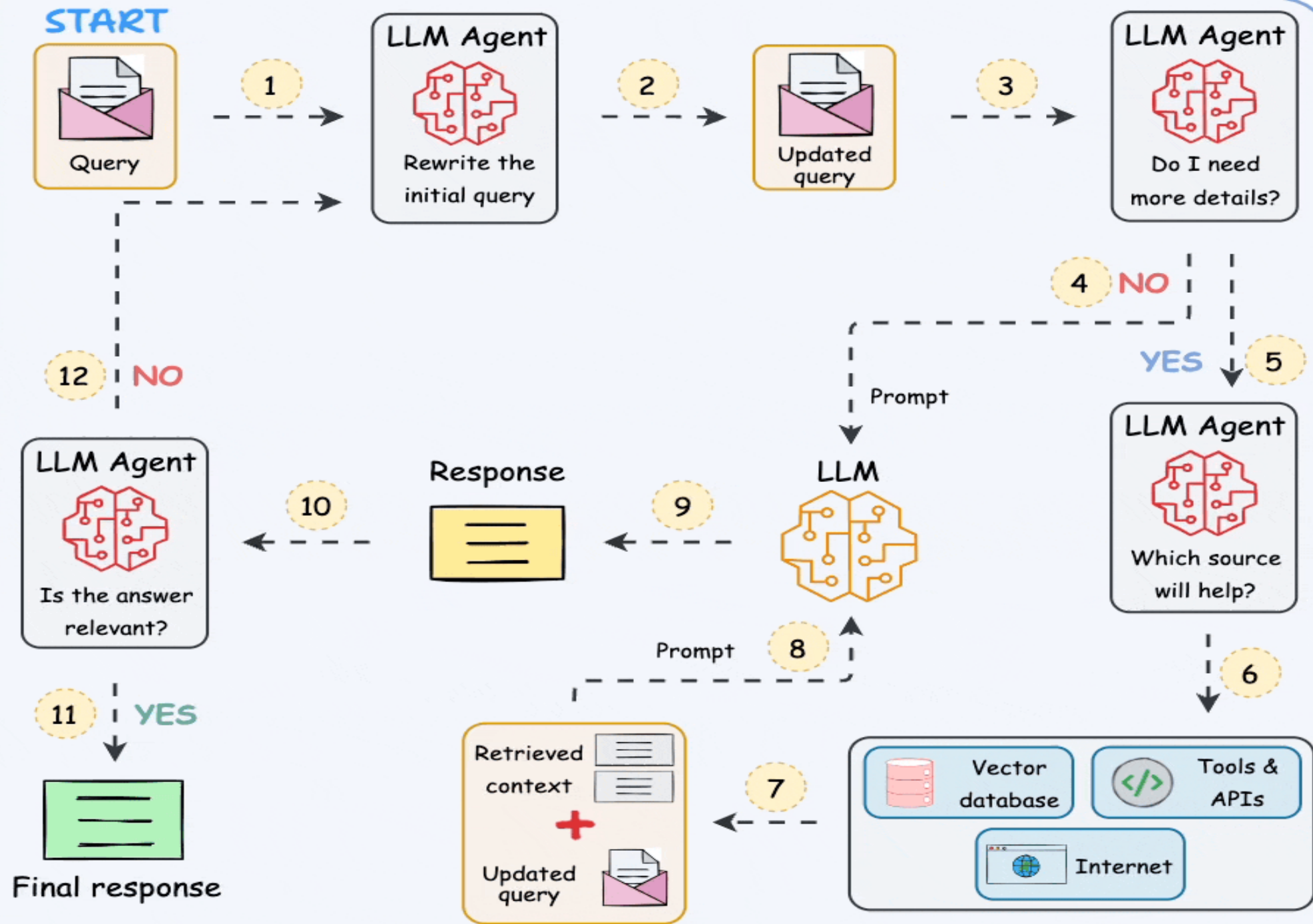




# Agentic RAG

Incorporates reasoning and decision-making, allowing iterative retrieval and deeper interaction with the user.

# Agentic RAG



# Example Applications in Healthcare

## **Traditional RAG:**

User asks, "What are the symptoms of hypertension?"  
Immediate retrieval: list of symptoms.

## **Agentic RAG:**

User (doctor) asks, "Diagnose and suggest a treatment plan for a patient exhibiting symptoms A, B, and C." The agent retrieves medical records, guidelines, and iteratively refines answers, suggesting follow-up diagnostics or personalized treatment options.



# Why Evaluation Matters in RAG-Based Systems

## Importance of Evaluation

Evaluation is essential for identifying inaccuracies, ensuring system reliability, and validating the relevance of outputs. By systematically assessing performance, improvements can be implemented, enhancing user experience and minimizing risks associated with incorrect information.



## Evaluation Method Overview

A robust evaluation method is integral to RAG performance, providing insights into retrieval accuracy and response quality. Regular evaluations help adapt systems to evolving user needs, ensuring the continuous delivery of valuable outputs.

## DeepEval Framework Description

DeepEval is an AI-driven evaluation framework tailored for RAG systems, assessing performance metrics like retrieval accuracy and generation quality. This comprehensive approach highlights error-prone elements, ensuring reliable and relevant outputs.

## Example Application in Healthcare

In healthcare, DeepEval can be instrumental in evaluating RAG systems that recommend medications for diseases. It ensures recommendations are both safe and effective, ultimately protecting patient welfare.

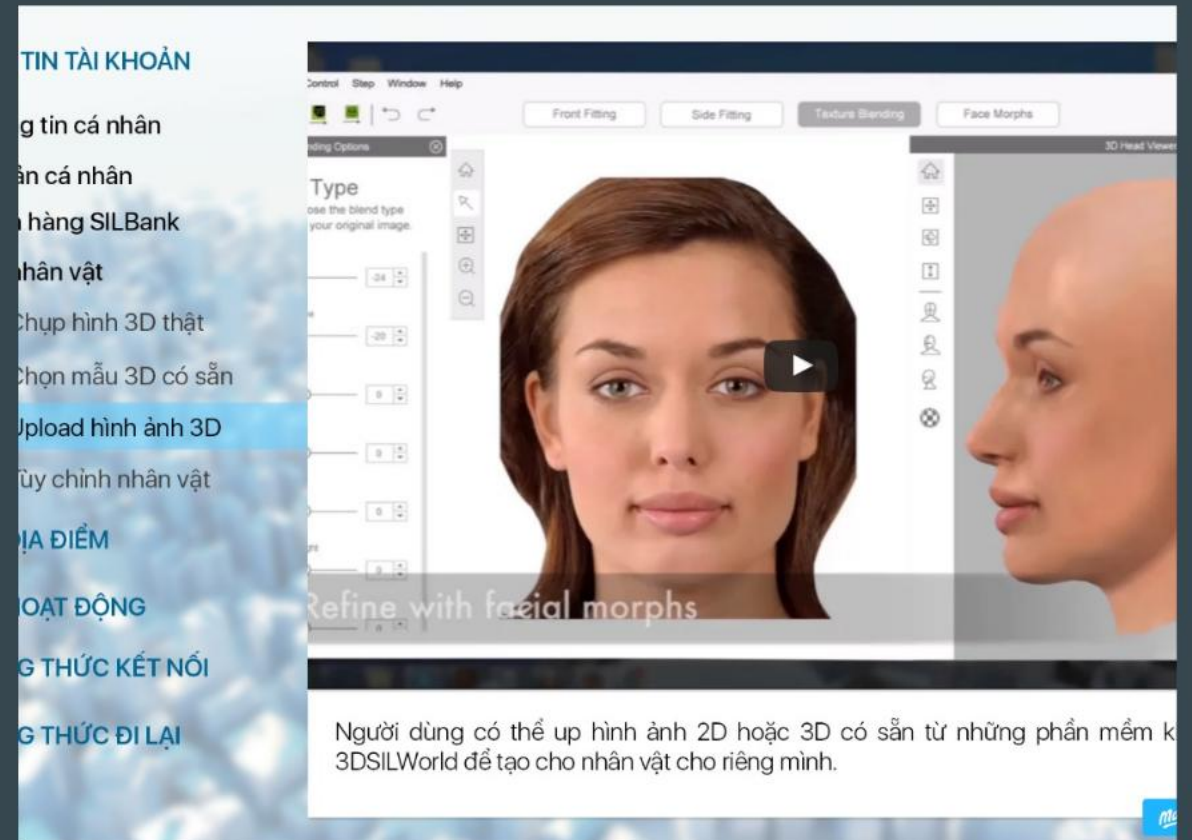


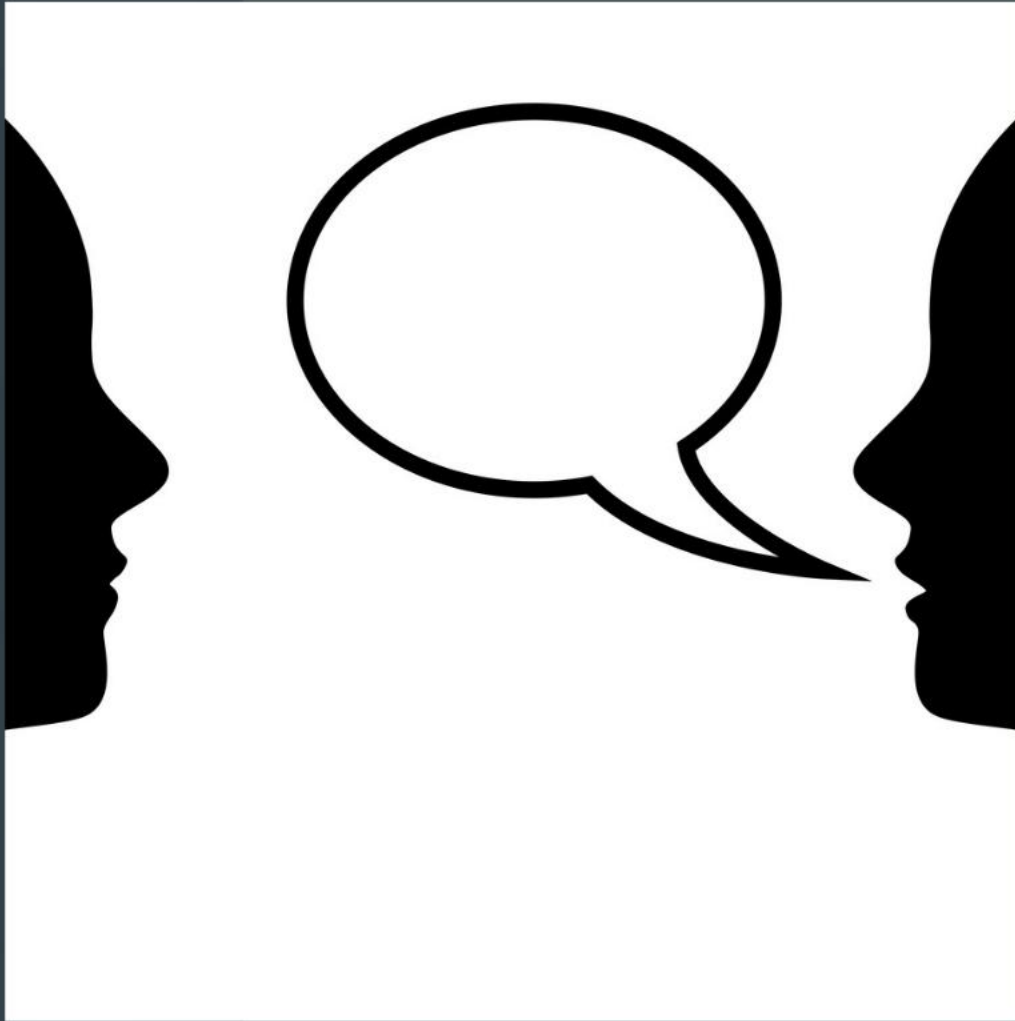


# Importance of Evaluation

Ensures accuracy, relevance, and quality of generated responses.

Highlights limitations and informs improvements.





# Evaluation Method Overview

## **DeepEval:**

An AI-driven evaluation framework that assesses retrieval accuracy, generation quality, relevance, and overall performance.

# Example Application in Healthcare

- Using DeepEval to evaluate a RAG system identifying medications for specific diseases:
- DeepEval highlights inaccuracies or outdated information, ensuring safe and effective healthcare recommendations, reducing risks to patients.

