

# **School of Computing Sciences &Engineering**

## **Department Of Computer Science & Engineering**



**Agentic Ai - Lab**

**(CSCR3215)**

**Lab File (2025-26)**

**For**

**B.Tech. (CSE) 6<sup>th</sup> Semester**

**Submitted To:**

Mr. Ayush Singh

Department of Computer Science & Engineering  
School Of Computing Sciences & Engineering

**Submitted By:**

Kusuma M  
B.Tech. CSE 6<sup>th</sup> Semester  
2023443738 CSF G2

# RAG Based Question Answering System

## Problem Statement

The objective of this project is to build a Retrieval-Augmented Generation (RAG) system that answers questions based on a financial annual report (Apple Inc. 2025 10-K Report).

Instead of relying only on a language model's memory, the system retrieves relevant content from the uploaded PDF document and then generates an answer using that retrieved context.

This approach reduces hallucination and ensures factual accuracy.

## Dataset / Knowledge Source

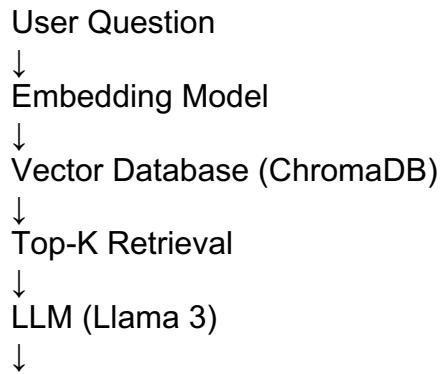
- **Dataset:** Apple Inc. 2025 Form 10-K (PDF)
- **Embedding Model:** all-MiniLM-L6-v2
- **Vector DB:** ChromaDB
- **Generator:** Llama 3 (via Groq API)

## RAG Architecture

### RAG Pipeline Steps:

1. Load PDF document
2. Chunk text
3. Generate embeddings
4. Store embeddings in ChromaDB
5. Retrieve relevant chunks
6. Pass retrieved context to LLM
7. Generate final answer

### Simple Block Diagram



Final Answer

## Text Chunking Strategy

- Chunk size: 1000 characters
- Chunk overlap: 200 characters

Reason:

- Prevents context loss
- Maintains semantic continuity
- Improves retrieval accuracy
- Avoids sentence truncation

## Embedding Details

- **Model Used:** sentence-transformers/all-MiniLM-L6-v2
- Dimension: 384

Reason:

- Lightweight
- Fast processing
- Works well on Colab
- Good semantic similarity performance

## Vector Database

- Used: ChromaDB
- Storage Type: Persistent Local Storage

Reason:

- Fast similarity search
- Easy integration with LangChain
- Efficient semantic retrieval
- Supports persistent storage

# Future Improvements

- Implement semantic chunking
- Hybrid search (BM25 + Vector Search)
- Add re-ranking layer
- Add metadata filtering
- Deploy full UI using Streamlit Cloud
- Upgrade to larger LLM models