

# **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:**

Simran Shikha  
B.Tech. CSE 6<sup>th</sup> Semester  
2023431909 CSF G1

# 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 scientific research papers.

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

## Dataset / Knowledge Source

- Dataset: wiki\_qa
- Embedding: MiniLM
- Vector DB: FAISS
- Generator: FLAN-T5 (loaded manually as above)

## RAG Architecture

### RAG Pipeline Steps:

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

### Simple Block Diagram (You can draw this in PDF)

User Question↓

Embeddings↓

Vector Database (FAISS)↓

Top-K Retrieval↓

LLM (distilgpt2)↓

Final Answer

## Text Chunking Strategy

- Chunk size: 500 characters
- Chunk overlap: 100 characters

Reason:

- Prevents context loss
- Maintains semantic continuity
- Improves retrieval accuracy

## Embedding Details

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

Reason:

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

## Vector Database

- **Used:** FAISS
- Type: IndexFlatL2

Reason:

- Fast similarity search
- Easy to implement
- Lightweight

# Future Improvements

- Better chunking (semantic chunking)
- Hybrid search (BM25 + FAISS)
- Re-ranking
- Metadata filtering
- UI using Streamlit
- Use better LLM like Mistral or LLaMA