hBCS28a: Fall 2025 STUDENT HANDOUT

Build Your Own RAG System (Retrieval-Augmented Generation) Group Project — 16 Groups, 5 Students per Group

---

1. Project Purpose

In the remaing of this course you will learn one of the most important AI techniques: **Retrieval-Augmented Generation (RAG)**. You (group) will learn independently, by building a RAG system in the end yourself in Python. Each week, any group may be randomly selected to present—be prepared at all times. (Note: This document is a simple guideline and will be modified quite frequently till the end of semester).

---

2. Deliverables (in the Weeks of 4-8)

Each group must prepare and be ready all lecutre to be called for a presentation:

1. A **12–15 minute presentation**
2. A **Python notebook or script**
3. A complet **working RAG system** that includes (at the end of the semester):
    - Text chunking
    - Embedding generation
    - Vector search (FAISS, ChromaDB, or cosine similarity)
    - Retrieval
    - A custom prompt using retrieved text
    - LLM answer generation

---

3. Project Tasks (by lecture weeks)

Task 1 — Prompt Engineering (November 29, 2025. Week 4)

(Each group) Learns and demonstrates:

- **LLM Limitations:** The issues of LLMs, primarily **hallucination** and knowledge cutoff.
- **The RAG Concept:** Introduce RAG as a solution. Explain the two main stages: **Retrieval** and **Generation ➔ Then move to Prompt Engineering:**
- Zero-shot and few-shot prompting
- Chain-of-thought
- Why LLMs hallucinate
- Why retrieval improves answers

Required Output:

3–5 prompt examples, including one that fails without retrieval. By the end of Week 4, that is November 29, 2025 (having listened from 4-6 groups) everyone expected to learn or have a clear idea of:

a) Why there is a need for RAG? (What are the limitations of LLMs?)
b) What actually RAG is?
c) What is prompt engineering and how it is done best?
d) Demonstrate examples of good prompt engineering practices.

Task 2 — Embeddings & Vector Search (Week 5)

Steps:

1. Choose a dataset (PDFs, articles, documentation, etc.)
2. Split text into chunks
3. Generate embeddings
4. Store embeddings in a vector database
5. Implement similarity search
6. Retrieve top-k chunks for a query

Required Output:
A query → retrieved top-k similar text chunks.

Task 3 — Build Your RAG Pipeline (Week 6)

Connect all steps:

1. Prepare a simple user Inteface (similar to that of Gemini, OpenAI, or DeepSeek)
2. Let user enters a question (put a prompt)
3. Vector search searches the relevant chunks (similarity search)
    1. Must use at least 3 different data format (pdf, video, image, text, web-content, etc.)
4. Retrieved text is inserted into a prompt template for LLM
5. Prompt is sent to an LLM
6. A final answer is produced

Required Output:
Show one question that fails without retrieval but succeeds with RAG.

Task 4 — Use LangChain or LlamaIndex (Week 7-8)

After understanding RAG's core components, you may rebuild your pipeline with a framework.

Required Output:
Explain what the framework automates compared to your manual implementation.

---

4. Group FINAL Presentation Guidelines

Your 13–15 minute presentation must include:

1. Clear explanation of RAG concepts
2. Each member speaks at least once
3. Live demo of your RAG system
4. A failure example (no retrieval)
5. A success example (with retrieval)
6. Reflection: What was difficult or surprising?

Any group can be called at any time.

---

5. FINAL Presentation Evaluation Rubric

Area	Weight
Concept understanding	25%
Quality of implementation	30%
Presentation clarity	15%
Live demo quality	20%
Preparedness	10%

---

6. Recommended Tools

- Python + Jupyter Notebook
- FAISS or ChromaDB
- Any LLM (OpenAI, Gemini, Claude, or local)
- LangChain or LlamaIndex (optional)

---

7. Self-Study Resources (YouTube & Guides)

Prompt Engineering

- "OpenAI Prompt Engineering Overview"
- "Prompt Engineering for Beginners — DeepLearning.ai"

Embeddings & Vector Databases

- "Embeddings Explained Simply"
- "FAISS in 10 Minutes"
- "Vector Databases for Beginners"

RAG Tutorials

- "RAG From Scratch"
- "Build a RAG System in Python"

Framework Tutorials

- "LangChain Crash Course"
- "LlamaIndex for Beginners"

---

8. Summary

You will learn RAG by building it yourself.
You must be able to explain each component clearly and demonstrate your system live.
Any group may be asked to present without warning both for the weeks of 5 to 8 and for the final presenations — prepare accordingly.