Name: Rounak Raj

Email: rajrounak366@gmail.com

Problem statement- assignment 2 Solutions

1. Where would you rate yourself on (LLM, Deep Learning, AI, ML). A, B, C [A = can code independently; B = can code under supervision; C = have little or no understanding]

Solution = A

I can work independently, but sometimes I seek supervision when handling complex edge cases or advanced optimizations.

2. What are the key architectural components to create a chatbot based on LLM? Please explain the approach on a high-level

To build an LLM-based chatbot, the first step is to define the target audience and use case (for example, customer support, finance, or personalized learning system). Based on this, we select an appropriate LLM and design system prompts to control behaviour.

Next, we add conversation memory so the chatbot can maintain context and behave naturally across turn. If the chatbot needs real-time or external information, we integrate tool calling (APIs, search, databases). For higher accuracy, we use RAG or fine-tuning with domain-specific data.

Finally, A Backend layer orchestrates the model, memory, tools, and logging to make the chatbot scalable and production-ready

3. Please explain vector databases. If you were to select a vector database for a hypothetical problem (you may define the problem) which one will you choose, and why?

Problem:
We want to build a personalized learning chatbot for students.

- The chatbot should answer questions from:
    - Class notes

    - PDFs
    - Previous conversations
- When a student asks a question, the chatbot should:
    - Find the most relevant content, even if exact words don't match
    - Use that content to generate an accurate answer

Traditional databases cannot handle this well because:

- They rely on exact keyword matching
- They do not understand semantic meaning

So we need a better way to search by meaning not by words.

A vector database stores data in the form of vector (numerical embedding).

Simple explanation:

- Text (documents, questions, chats) is converted into numbers using an embedding model.
- Similar texts have similar vector
- A vector database stores these vectors and allow fast similarity search

So instead of asking:

"Find text with these exact words"

We ask:

"Find text that means something similar"

Steps to follow to build vectordatabase:

1.Convert documents into embeddings using an embedding model.

2. Store these embeddings in a vector database.

3.Convert the user's question into an embedding.

4.Search the vector database for the most similar embeddings.

5.Retrieve the relevant content.

6.Pass this content to the LLM to generate the final answer.

NOTE : To build a better vector database, we need proper document chunking with overlap use metadata (source, page number, topic) for accurate document fetching and evidence, and apply appropriate similarity metrics to improve retrieval quality.