Assignment 2

1. Self-Rating on LLM, Deep Learning, AI, and Machine Learning

*I would rate myself as **A** in **LLM, Deep Learning, Artificial Intelligence, and Machine Learning**, which means I am able to code and work independently.*

I have a good understanding of both the theory and practical aspects of AI and ML. I am comfortable implementing machine learning models, training them on datasets, evaluating their performance, and improving them when needed. In deep learning, I understand how neural networks work and have experience using frameworks such as PyTorch or TensorFlow.

Regarding large language models, I have worked with pre-trained LLMs, understand how prompting works, and know how to build basic applications such as chatbots and question-answering systems. Overall, I feel confident working on AI/ML problems without constant supervision.

2. Key Architectural Components of an LLM-Based Chatbot (High-Level Approach)

An LLM-based chatbot is not just a single model but a system made up of multiple components that work together.

*The first component is the **user interface**. This is the part where users interact with the chatbot, such as a web page, mobile app, or chat platform. It takes the user's input and displays the chatbot's response.*

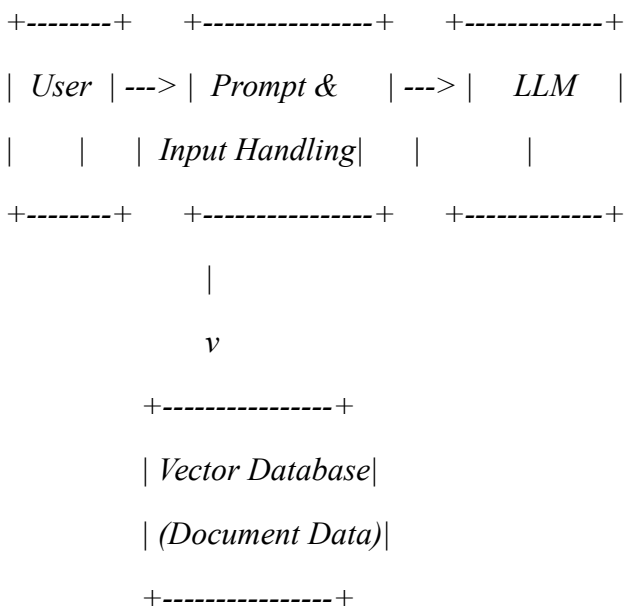
*Next comes the **input processing and prompt construction layer**. The user's question is cleaned and combined with system instructions (such as how the chatbot should behave) and previous conversation history. All of this information is put together into a prompt that is sent to the language model.*

*The **large language model (LLM)** is the core of the chatbot. It processes the prompt and generates a response based on its training and reasoning ability.*

*In many real-world systems, the chatbot also uses an external knowledge source. This is done through a **retrieval component**, where relevant documents or data are fetched and added to the prompt before sending it to the LLM. This approach is called **Retrieval-Augmented Generation (RAG)** and helps the chatbot provide more accurate and up-to-date answers.*

*Finally, the **output processing layer** formats the response and sends it back to the user in a readable way.*

High-Level Architecture Diagram



Main Components Summary (Table)

Component	***Purpose***
User Interface	*Collects user input and shows responses*
Prompt Handling	*Prepares structured input for the LLM*
LLM	*Understands queries and generates responses*
Retrieval System (Optional)	*Fetches relevant external data*
Output Processing	*Formats and returns the response*

3. Vector Databases

*A **vector database** is a type of database used to store and search numerical vectors called embeddings. These embeddings represent the meaning of unstructured data such as text documents, images, or audio.*

*Unlike traditional databases that search using keywords, vector databases allow **semantic search**, which means they return results based on meaning. For example, even if the words are different, the database can still find relevant information if the meaning is similar.*

When a user asks a question, the query is converted into a vector using an embedding model. This vector is then compared with stored vectors in the database using similarity measures like cosine similarity. The closest matches are returned.

Hypothetical Problem

Suppose I want to build a chatbot for an organization that can answer employee questions using internal documents such as HR policies, project guidelines, and technical manuals. The chatbot should respond quickly and handle a large number of documents.

Vector Database Choice: Pinecone

For this problem, I would choose ***Pinecone*** *as the vector database.*

Reasons for choosing Pinecone:

- *It can scale easily with large amounts of data*
- *It provides fast similarity search with low latency*
- *It is fully managed, so there is less infrastructure overhead*
- *It supports metadata filtering, which helps improve search accuracy*

Vector Search Flow Diagram

User Question

|

v

Embedding Model

|

v

Vector Database

|

v

Relevant Documents

|

v

LLM Response

Comparison Table (Brief)

Vector Database Use Case

FAISS	*Research and local experiments*
Chroma	*Small to medium projects*
Weaviate	*Open-source and hybrid search*
Pinecone	*Large-scale, production systems*

Conclusion

To summarize, I am confident in my skills across AI, ML, deep learning, and large language models. An LLM-based chatbot consists of multiple components including a user interface, prompt handling, an LLM, and often a retrieval system backed by a vector database. Vector databases play a crucial role in enabling semantic search, and for scalable real-world applications, Pinecone is a strong and practical choice