**Embeddings, Vector DB, Retrieval, and Similarity:**

* What embedding and vector DB did you use and why?
* What was the vector size and what is the impact of vector length?
* Which vector DB did you use and why?
* What are different types of similarity search (cosine, Euclidean, Manhattan) and when to use what?
* How to perform retrieval operation?
* How do you handle metadata in vector DB?
* What is a vector DB?

**RAG (Retrieval-Augmented Generation):**

* What is RAG architecture?
* How does RAG work?
* What are RAG failures, and how do you evaluate RAG?
* Where does the evaluation module sit in a RAG pipeline?
* How to design Multi-modal RAG?
* What is RAG and Agents?
* What applications have you built using RAG, LangChain, LangGraph?

**Deterministic & Guarded Responses:**

* How to ensure deterministic response in tightly coupled guideline-based apps?
* How to define guardrails in LLM responses?

**Conversational AI:**

* How is LLM chatbot different from normal chatbot?
* How is LLM chatbot different from voice bots?
* How to build full-fledged conversational AI system?
* What is LangGraph?
* What is agentic flow and how to design it?

**Tech Stack & Infra Integration**

**Azure & Outlook Flow:**

* How system fetches PDF from Outlook?
* Why use Azure Blob Storage?
* What is Microsoft Graph API?
* Role of Azure Functions or App Services?
* Why use Azure Cosmos DB?
* What is Azure AI Search / Azure AI Studio?

These are spot-on for cloud-based GenAI apps. Keep Azure infra knowledge strong.

**OCR & Parsing**

* How OCR works (including LLM-based)?
* What happens after data extraction?
* How to parse a table split across multiple pages?
* What is document parsing — how to parse from documents and DBs?

*Smart Tip:* For multi-page table parsing: discuss layout-aware parsing (like PDFPlumber, unstructured.io, layoutLM) — **not just LLMs**.

**LLM Understanding & Comparison**

* What is BERT vs LLM? (repeated but valid)
* How LLM is different from BERT?
* Token size used in LLM input?
* Which LLMs have you used?
* Gemini vs GPT-4.0?
* Deploying Gemini 4.0-based RAG on Azure/GCP?

**Model Performance, Accuracy & Retraining**

* ML metrics for classification?
* How to check model accuracy?
* What do you do if accuracy reduces?
* How to retrain & split train-test data?

Tip: Be ready with precision, recall, F1, ROC-AUC, and confusion matrix based use-cases.

**AI System Design / XAI / Production**

* How to manage concurrency for multiple users?
* How will you implement memory management?
* How to manage cache / state?
* How to implement XAI / Responsible AI?
* How to define & enforce guardrails?
* All AI use cases you've worked on?

### **Dataset & Chunking**

* How did you profile your dataset before processing — number of rows, columns, data types, missing values?
* Why did you chunk a ~500k row dataset even though LLMs can handle small datasets?
* What chunking strategies (fixed, recursive, semantic) did you consider, and when is each ideal?
* What impact does vector size/dimension have on retrieval quality and performance?

### 

### **Embeddings, Vector DB & Retrieval**

* Which embedding model (OpenAI, BGE, etc.) and vector DB (FAISS, Pinecone, etc.) did you use and why?
* What types of vector stores exist, and when should you use FAISS, Pinecone, Weaviate, or Qdrant?
* What indexing methods (Flat, IVF, PQ, HNSW) does FAISS support, and how do they affect speed/accuracy?
* How are vectors stored internally in vector databases?
* How is a vector retrieved (via similarity search), and what happens under the hood?
* How does product quantization and inverted indexing make large-scale search more efficient?
* How did you optimize search performance with ~800k rows?
* What similarity metrics (cosine, dot product, Euclidean, Manhattan) did you explore, and when is each ideal?
* When would you choose a managed vector DB like Pinecone over a local one like FAISS?

### **RAG (Retrieval-Augmented Generation)**

* What is RAG architecture and how did you implement it in your system?
* How do you evaluate and improve a RAG pipeline when responses are inaccurate or hallucinated?
* Where does the RAG evaluation module sit, and what metrics do you use to validate responses?
* What different similarity search strategies are used in RAG, and which is best when?
* What is reranking (e.g., MMR, cross-encoder), and when is it needed in RAG?
* What is agentic RAG and how does it differ from classic retrieval pipelines?
* What models/tools (LangGraph, LangChain, FAISS, OpenAI, Azure) did you use to build the RAG system?
* What is LangGraph, and how is it different from LangChain in terms of agent orchestration?

### **Prompting, JSON Output, LLM Behavior**

* What is the token limit of GPT-4, and how does it affect chunking and prompt design?
* What’s the difference between zero-shot and few-shot prompting, and when is each ideal?
* What are the drawbacks of few-shot prompting (e.g., cost, prompt drift, token explosion)?
* How do you reduce hallucinations in LLMs when handling scientific or sensitive content?
* How do you ensure the LLM returns output in valid JSON or structured format every time?
* How do you improve chain-of-thought and reasoning quality if LLM outputs poor responses?
* How many tokens were you passing to the LLM on average, and how did you manage input limits?

### **Conversational AI & Agent Design**

* How is an LLM chatbot different from a rule-based or traditional chatbot?
* How would you implement role-based access (e.g., restrict responses based on employee pay grade)?
* Have you worked on voice bots, and how do they differ in architecture from chatbots?
* How would you design a full-fledged end-to-end conversational AI system using LangGraph or LangChain?
* What is an agentic flow and how do you design multi-agent workflows using LangGraph?
* How would you implement session memory or chat history in a multi-turn chatbot?
* How do you manage state and cache in a high-concurrency GenAI application?
* How do you scale your system for many simultaneous users (concurrency strategy)?

### **App Integration & Infra (Azure, Email, Parsing)**

* How did your system automatically detect and extract PDF files from Outlook?
* Why did you use Azure Blob Storage — what benefit did it bring to your pipeline?
* What does Microsoft Graph API do in your architecture?
* What’s the role of Azure Functions or App Services in your RAG-based solution?
* What is Azure AI Search and how does it work with vector-based search?
* Why did you use Azure Cosmos DB instead of MongoDB or SQL?
* How did you parse multi-page tables in DOCX/PDF files (cost-efficient + accurate)?
* What steps did your system follow after extracting data via OCR (structured parsing)?

### **ML Model Metrics, Accuracy & Retraining**

* What classification metrics (accuracy, precision, recall, F1) did you use and why?
* If model accuracy dropped, how did you debug and improve the pipeline?
* How do you retrain an ML model, and how do you manage train/test split to avoid leakage?
* How do you check and measure model accuracy, both for LLMs and ML models?

### **Data Structures & Algorithms (DSA)**

* What are the best and worst-case time complexities for common list operations?
* What’s the time complexity for Python list operations like append, insert, pop, etc.?
* Which is faster — list or dictionary — and in what scenarios?

### 

### **GenAI Project Discussion**

* Walk me through your latest Generative AI project (business problem, technical flow, outcomes).
* What LLM models, vector DBs, tools, and cloud services did you use?
* How did you implement Human-in-the-Loop in your system to improve quality and trust?
* How did you integrate Responsible AI principles (e.g., explainability, fairness, scientific validity)?
* How did you extract structured data from unstructured documents (e.g., research PDFs)?
* What was the structure of the tech team, and what was your exact role?
* How did your pipeline handle scale, latency, and large document parsing?

### **Behavioral + Guesstimate**

* Guesstimate: What is Netflix’s annual revenue? (Show step-by-step thinking: users × ARPU)
* If we call your manager right now, what are 3 strengths and 3 improvement areas they’d share?