AmanAI Lab

AI EDUCATION & MENTORSHIP

★ LIVE COHORT PROGRAM

GenAI & Agentic AI Interview Masterclass

24 live sessions to take you from core NLP foundations to production-grade Generative AI and Agentic AI systems — with theory, real production practice, and a full interview questions discussed live, session by session.

24	9	160+	1:1
LIVE SESSIONS	MODULES	TOPICS COVERED	MOCK SESSION

✓ Free 1-on-1 Mock Interview

✓ Personal Resume Review

✓ 1-on-1 Personal Discussion

✓ Job Referral Support

Detailed Program Syllabus

amanailab.com

Program Overview

Built for anyone preparing for GenAI Engineer, LLM Engineer, Agentic AI Developer, AI Solutions Architect, or Applied ML roles. Every session pairs theory revision with how the concept is used in real production systems, followed by live discussion of the interview questions actually being asked on that topic in 2026 hiring rounds.

24	90 min	4 Days	30 Days
LIVE SESSIONS	PER SESSION	PER WEEK	RECORDING ACCESS

EVERY SESSION FOLLOWS THIS STRUCTURE

1. Theory
Core concept explained simply, with examples and diagrams

2. Quick Revision
Key points, comparisons and formulas recapped fast

3. Production Lens
How the concept is actually used in real systems

4. Interview Discussion
Real questions on the topic solved and discussed live

Program Benefits

1:1

Free 1-on-1 Mock Interview

A personal mock interview scheduled just for you, with direct feedback on your answers and delivery.

CV

Personal Resume Review

Your resume reviewed live and 1-on-1 for GenAI/Agentic AI roles — projects, keywords, and ATS readiness.

1:1

1-on-1 Personal Discussion

A dedicated personal session with Aman to discuss your background, target roles, and preparation gaps.

JR

Job Referral Support

Referrals shared with the cohort whenever relevant openings come up in Aman's network.

30d

30-Day Recording Access

Revisit every session for a full month — never lose a concept because you missed a live class.

PDF

Complete PDF Notes Bundle

Branded, structured notes for every module so you revise without rewatching entire sessions.

Module 1 & 2 — Foundations

1 NLP & Deep Learning Foundations

FOUNDATIONS

S1 Tokenization & Text Representation

Word / subword / character tokenization, BPE, WordPiece, SentencePiece, vocabulary trade-offs, tokenizer cost impact at scale

S2 Embeddings

Word2Vec, GloVe, FastText, static vs contextual embeddings, similarity measures, dimensionality vs storage cost

S3 RNN

Sequential modeling, hidden states, backpropagation through time, vanishing/exploding gradients

S4 LSTM & GRU

Gating mechanisms, cell state, LSTM vs GRU vs RNN, Seq2Seq encoder-decoder architecture

2 Transformers & Foundation Models

CORE ARCHITECTURE

S5 Attention Mechanism

Bahdanau & Luong attention, self-attention, Query-Key-Value, scaled dot-product attention

S6 Transformer Architecture

Multi-head attention, positional encoding, encoder-decoder designs, BERT vs GPT vs T5

S7 LLM Training Pipeline

Pretraining, instruction tuning, RLHF, DPO, major model families (GPT, Claude, Gemini, Llama)

Module 3 & 4 — Prompting & RAG

3 Prompt Engineering & Guardrails

APPLICATION LAYER

S8 Prompt Engineering

Zero-shot, few-shot, Chain-of-Thought, ReAct prompting, prompt versioning in production

S9 Structured Outputs & Safety

Function calling, JSON mode, guardrails, jailbreak defenses, bias mitigation, content moderation layers

4 RAG Systems

MOST ASKED IN 2026

S10 RAG Fundamentals

Retriever-generator architecture, chunking strategies, chunk size/overlap trade-offs, preprocessing pipelines

S11 Vector Search & Databases

FAISS, Pinecone, Weaviate, Chroma, approximate nearest neighbor search, indexing strategies (HNSW, IVF)

S12 Advanced RAG

Hybrid search (dense + sparse/BM25), reranking, query rewriting/expansion, GraphRAG, multi-hop retrieval

S13 RAG Evaluation & Production

Faithfulness/groundedness metrics, common failure modes, monitoring retrieval quality in production

Module 5 & 6 — Fine-tuning & LLMOps

5 Fine-tuning & Evaluation

MODEL ADAPTATION

S14 Fine-tuning Techniques

Full fine-tuning vs PEFT, LoRA, QLoRA, adapters, prefix tuning, cost comparison

S15 LLM Evaluation & Compliance

Perplexity, BLEU/ROUGE, human eval, LLM-as-judge, hallucination detection, data privacy in regulated industries

6 LLMOps & Production Engineering

SYSTEM DESIGN ROUNDS

S16 Deployment & Observability

Model serving, latency optimization, caching, semantic caching, logging/tracing, cost monitoring at scale

S17 Security & Red-Teaming

Prompt injection attacks, data exfiltration risks, adversarial testing, PII handling in LLM pipelines

Module 7, 8 & 9 — Advanced Topics, Agentic AI & System Design

7 Advanced Topics & Emerging Trends

STAND OUT ROUND

S18 Model Optimization & Efficient Inference

Quantization (INT8, GPTQ, AWQ), knowledge distillation, pruning, KV cache, speculative decoding, batching strategies, Mixture of Experts (MoE) architecture

S19 Long Context & Multimodal AI

Context window management, RoPE scaling, sliding window attention, vision-language models, multimodal LLMs (image/audio/video understanding)

S20 Frameworks, Evaluation Tooling & Data

LangChain vs LlamaIndex vs Haystack, evaluation frameworks (RAGAS, TruLens, DeepEval), synthetic data generation for fine-tuning, open-weight models (Llama, Mistral, Gemma) & licensing considerations

S21 Responsible AI & Governance

Bias & fairness in LLMs, explainability, AI governance frameworks, model cards, audit trails for regulated deployments

8 Agentic AI

FASTEST GROWING AREA

S22 Agent Fundamentals

Agent architecture — planning, memory, tools, orchestration; ReAct pattern; agent vs simple LLM call

S23 Agentic RAG & Memory

Multi-step reasoning agents, short/long-term memory types, LangGraph state graphs

S24 Multi-Agent Systems

CrewAI, AutoGen, Model Context Protocol (MCP), Agent-to-Agent (A2A) protocol, orchestrator patterns

S25 Agents in Production

Cost/latency optimization, safety guardrails, Computer-Using Agents, agentic coding, human-in-the-loop design, failure recovery

9 System Design & Career Readiness

FINAL ROUND PREP

S26 End-to-End System Design + Career Prep

Designing a production RAG pipeline and a multi-agent pipeline; system design framework (requirements → architecture → trade-offs → scaling); resume & portfolio positioning; job referral discussion

What's Included

- ✓ 24 live sessions (90 minutes each)
- ✓ Complete branded PDF notes bundle
- ✓ Free 1-on-1 personal resume review
- ✓ 1-on-1 personal discussion with Aman
- ✓ Comprehensive interview question bank covered live
- ✓ Session recordings — 30 days access
- ✓ WhatsApp / Telegram doubt-solving group
- ✓ Free 1-on-1 mock interview session
- ✓ Job referral support for relevant openings
- ✓ Certificate not included — pure hands-on prep

Interview Question Bank — Topic-Wise

A representative set of the interview questions discussed live across all 24 sessions, grouped by topic.

Tokenization, Embeddings & Sequence Models — 12 Qs

- Why does GPT use Byte Pair Encoding instead of word-level tokenization?
- How does tokenizer choice affect model cost and context length?
- What is the difference between Word2Vec and GloVe?
- Why do embeddings capture analogies like king − man + woman = queen?
- Why do static embeddings fail compared to contextual embeddings?
- Explain cosine similarity vs Euclidean distance for embeddings.
- Why do RNNs struggle with long sequences?
- What is backpropagation through time?
- Explain the three gates in an LSTM cell.
- Why does LSTM handle long-term dependencies better than RNN?
- When would you choose GRU over LSTM?
- What is a Seq2Seq encoder-decoder architecture used for?

Attention & Transformer Architecture — 12 Qs

- Explain self-attention in simple terms.
- Why is attention better than RNN for long sequences?
- What are Query, Key and Value vectors in attention?
- Why use multi-head attention instead of single-head?
- Why do transformers need positional encoding?
- What is the difference between encoder-only, decoder-only and encoder-decoder models?
- Compare BERT and GPT architectures.
- What is masked language modeling vs causal language modeling?
- What is RLHF and why is it needed?
- Explain DPO and how it differs from RLHF.
- Why don't bigger models always perform better in production?
- How would you choose between GPT, Claude, Gemini and Llama for a use case?

Prompt Engineering & Guardrails — 10 Qs

- What is Chain-of-Thought prompting and when does it help?
- Difference between zero-shot, few-shot and instruction prompting.
- How does prompting reduce hallucination?
- What is the ReAct prompting pattern?
- How do you design prompts for structured JSON outputs?
- What is prompt injection and how do you defend against it?
- How do content moderation layers work in an LLM application?
- How do you version and test prompts in production?
- What is the difference between system, user and developer prompts?
- How would you reduce bias in an LLM's outputs?

RAG & Vector Search

14 Qs

- Why use RAG instead of fine-tuning?
- How do you decide chunk size and overlap for a document set?
- What are the common failure modes of a naive RAG pipeline?
- How does approximate nearest neighbor search work?
- Compare FAISS, Pinecone, Weaviate and Chroma.
- What is HNSW indexing and why is it used?
- When would you use hybrid search over pure vector search?
- What role does reranking play after initial retrieval?
- What is query rewriting and when is it needed?
- Explain GraphRAG and how it differs from standard RAG.
- What is multi-hop retrieval and when is it required?
- How do you evaluate the faithfulness of a RAG pipeline's outputs?
- How do you monitor retrieval quality once a RAG system is live?
- How would you debug a RAG system that keeps giving wrong answers?

Fine-tuning & Evaluation

12 Qs

- Explain LoRA in simple terms.
- How does QLoRA reduce memory requirements further?
- When would you fine-tune instead of using RAG?
- What is the cost/performance trade-off of full fine-tuning vs PEFT?
- What are adapters and prefix tuning?
- How do you measure hallucination in an LLM's output?
- What is LLM-as-judge evaluation and when is it useful?
- Compare perplexity, BLEU and ROUGE as evaluation metrics.
- How would you build an evaluation pipeline for a GenAI product?
- How do you ensure HIPAA/financial compliance in a GenAI system?
- How do you anonymize or de-identify data used by an LLM?
- How do you prevent unauthorized data from reaching the model?

LLMOps, Deployment & Security

10 Qs

- How do you reduce latency and cost for an LLM application at scale?
- What is semantic caching and how does it help?
- How do you design logging and tracing for an LLM pipeline?
- What is prompt caching and when is it effective?
- How do you monitor cost across thousands of LLM calls?
- How would you red-team an LLM application before launch?
- What is data exfiltration risk in a GenAI system and how do you prevent it?
- How do you handle PII safely inside an LLM pipeline?
- What layers of guardrails would you put around a production LLM app?
- How do you roll out a new model version safely in production?

Model Optimization, Multimodal & Responsible AI

14 Qs

- What is quantization and how do INT8, GPTQ and AWQ differ?
- Explain knowledge distillation and when you would use it.
- What is a KV cache and why does it speed up inference?
- What is speculative decoding and how does it reduce latency?
- Explain the Mixture of Experts (MoE) architecture.
- How do you extend an LLM's context window without retraining from scratch?
- What is RoPE scaling used for?
- How do vision-language models process image and text together?
- Compare LangChain, LlamaIndex and Haystack for building LLM applications.
- What is RAGAS and how is it used to evaluate RAG pipelines?
- How would you generate synthetic data for fine-tuning?
- What licensing considerations matter when using open-weight models like Llama or Mistral?
- How do you detect and mitigate bias in an LLM's outputs?
- What belongs in a model card for a regulated deployment?

Agentic AI & Multi-Agent Systems

16 Qs

- What is agentic AI and how is it different from a chatbot or basic GenAI pipeline?
- What are the four components interviewers check for in agent architecture?
- Explain the ReAct pattern for agents.
- How does an agent decide which tool to call?
- What is the difference between short-term and long-term agent memory?
- How do you give an agent long-term memory?
- What is Agentic RAG and how does it differ from traditional RAG?
- Why use LangGraph over a simple LangChain chain?
- How do multiple agents communicate in a multi-agent system?
- Compare CrewAI and AutoGen for multi-agent orchestration.
- What is the Model Context Protocol (MCP) and why does it matter?
- What is the Agent-to-Agent (A2A) protocol used for?
- What are Computer-Using Agents and what risks do they introduce?
- How is agentic coding different from Copilot-style autocomplete?
- How do you evaluate an autonomous agent's performance?
- How do you prevent an agent from taking harmful or unintended actions?

System Design & Scenario-Based

12 Qs

- Design a production RAG pipeline for a customer support system.
- Design a multi-tenant RAG system serving multiple clients.
- Design an agentic pipeline for a research assistant use case.
- How would you design an agent for a long-horizon task without it getting stuck?
- Why choose hybrid search over vector-only search for a given system?
- When does fine-tuning beat prompting in a real project?
- Why choose LangGraph over LangChain for a specific workflow?
- How would you design an evaluation platform for a GenAI product?
- How would you design human-in-the-loop checkpoints for a critical agent workflow?
- How would you scale a GenAI system from 100 to 100,000 users?
- How would you approach a case study involving a hospital or bank's GenAI compliance needs?
- How do you decide between building in-house vs using a managed GenAI platform?