Cracking Generative AI Interviews: The Ultimate Playbook

Welcome to "Cracking Generative AI Interviews: The Ultimate Playbook", a comprehensive guide curated by Maninder Kaur, Associate Director at Talentify Global in collaboration with Leading Gen AI Experts. This playbook is designed for aspiring professionals seeking roles in Generative AI and Large Language Models (LLMs)—whether you're a recent graduate, a working professional, or transitioning into an AI role.

AI Vanguard Panel

Expert Contributors to This Guide

This guide is enriched with expert insights from top contributors, referred to as the "AI Vanguard Panel", who bring real-world experience, project exposure, and tactical understanding:

Nishant Niranjan	Amar Misra	Harsimar Singh
Co-founder – SolvusAI	Senior Director (Chief Architect)Transversal - Capgemini	AI-ML Consultant | Generative AI - KPMG

Your roadmap to breaking into one of the fastest-growing domains in tech

This guide is designed for professionals across roles—engineers, analysts, product folks, and AI-curious learners—who are exploring a pivot into the world of Generative AI.

Whether you're a developer learning about LLMs, a data scientist exploring model tuning, or a beginner navigating prompt engineering—this guide provides a **structured, real-world interview toolkit** to help you stand out.

What makes this playbook unique?
It combines deep technical knowledge and real hiring insights, contributed by AI practitioners and recruiters actively shaping the GenAI hiring landscape.

✨ **Special thanks to:**

- **Amar** – for sharing in-depth technical scenarios and real-world GenAI challenges
- **Nishant** – for outlining advanced evaluation strategies and fine-tuning frameworks
- **Harsimar** – for simplifying LLM fundamentals and tokenization for beginner-friendly learning

Let this be your launchpad into the world of Generative AI—where skills, strategy, and story matter.

What you can expect

Technical Deep Dives

Core GenAI and LLM concepts explained

Real Interview Q&A

- 25+ curated GenAI interview questions (with recruiter insights)

Case-Based Thinking

Ethical frameworks, project scenarios, and evaluation strategies

Best Practices

For prompt engineering, fine-tuning & ethics

Practical Resources

Books, communities, and projects-Transition roadmap for upskilling from non-AI roles

Why This Guide Matters a Recruiter's POV

As a recruiter working closely with fast-scaling AI and tech firms, I often notice that while candidates possess theoretical knowledge, they struggle to articulate practical GenAI concepts confidently during interviews. This guide bridges that gap by offering well-structured, real-world questions along with detailed sample responses. It's a resource I'd recommend to anyone preparing for GenAI roles — whether you're just pivoting into AI or aiming for senior-level technical interviews.

Understanding Generative AI

What is Generative AI?

Generative AI systems are capable of creating new content—such as text, images, music, or code—based on patterns learned from training data. This ability is powered by models like RNNs, LSTMs, and more recently, Transformers.

RNNs in GenAI

Recurrent Neural Networks (RNNs) are designed for sequence data and were foundational in early GenAI development. They process tokens one at a time, maintaining a hidden state to preserve context. However, they struggle with long-term dependencies and have largely been replaced by LSTMs, GRUs, and Transformers.

✨ Real-World Example

Training an RNN on movie scripts can allow it to complete dialogue like:

- **Input: "I'll be back..."**
- **Output: "...as soon as I finish what I started."**

Components of Generative AI

1. **Prompt Engineering**

Crafting eective prompts to guide AI models towards desired outputs.

1. **Fine-Tuning**

Adapting pre-trained models to specific tasks or domains using additional data.

1. **Evaluation Metrics**

- **Text:** BLEU, ROUGE, METEOR, and human evaluations.
- **Images:** Fréchet Inception Distance (FID), Inception Score.

4. Ethical Considerations

Addressing biases, ensuring data privacy, and mitigating misinformation.

Insights from the Frontlines of GenAI- **Nishant Niranjan** — Co-founder, SolvusAI

A visionary in AI product development and research, Nishant leads cutting-edge projects in large language models, driving innovation in scalable GenAI solutions. The following questions, shared by Nishant, highlight critical aspects of Large Language Models including their limitations, fine-tuning techniques, hallucination mitigation, AI agents, pipeline design, ethics, model evaluation, and document processing best practices:

1. What are the main limitations of Large Language Models (LLMs)?

LLMs like GPT have limitations such as:

- Hallucination: Confidently generating incorrect information.
- Context length constraints: Struggling with long documents.
- Lack of up-to-date knowledge unless fine-tuned or augmented.
- Computational cost and latency.
- Infrastructure Cost
- Research Cost
- Experimentation cost

2. What are LoRA, QLoRA, and PEFT techniques in fine-tuning?

- **LoRA** (Low-Rank Adaptation): Freezes most of the model and fine-tunes only a small set of low-rank weights → ecient and lightweight.
- **QLoRA**: Combines quantization with LoRA for running large models on lower-end hardware.
- **PEFT** (Parameter-Ecient Fine-Tuning): Umbrella for techniques like LoRA, Adapter, Prefix Tuning.

3. What causes hallucinations in LLMs, and how can they be mitigated?

Hallucinations happen when a model generates plausible but incorrect or fabricated responses. This is often due to lack of grounding, poor-quality data, or misaligned objectives. Mitigation strategies include using Retrieval-Augmented Generation (RAG), applying Reinforcement Learning with Human Feedback (RLHF), implementing fact-checking layers, and fine-tuning with curated datasets.

4. What are AI Agents in the context of GenAI?

AI Agents are systems that autonomously plan and execute tasks using tools, memory, and reasoning across steps.

Example: CrewAI: Frameworks for autonomous agents using LLMs.

5. How would you design a GenAI pipeline for real-time document summarization?

- OCR (if input is scanned image/PDF)
- Preprocessing (cleaning text, chunking)
- Embedding (if using RAG) and storage in a vector DB
- Retrieval + generation with summarization prompt
- Post-processing (fact-check, formatting)

6. How have you handled ethical dilemmas in AI-related projects?

- S: Built a resume screening system using NLP.
- T: Realized the model was biased toward certain universities and names.
- A: Paused deployment, introduced diverse training data, bias detection, and adversarial testing.
- R: Reduced bias significantly, and system was approved by compliance and deployed responsibly.

7. How to research and select an appropriate LLM for a task?

Start by defining the task type and constraints (latency, accuracy, domain). Explore model performance on benchmarks via sources like Hugging Face, Open LLM Leaderboard, or Papers with Code. Compare architecture, context length, tokenizer compatibility, and licensing. Evaluate shortlisted models using pilot tests and taskspecific metrics.

8. Why are chunk size and overlap important in document processing for LLMs?

Chunk size aects context length—too small loses meaning, too large may be truncated. Overlap preserves continuity between chunks. Together, they ensure coherent input for retrieval or generation.

9. How do you evaluate the performance of a large language model (LLM) beyond accuracy or BLEU?

Use a mix of automatic (ROUGE, METEOR, F1) and human evaluations (fluency, coherence, factuality). For truthfulness, tools like TruthfulQA or HELM help. In production, apply A/B testing and user feedback. Calibration and uncertainty are key in high-stakes use cases.

From Practitioners to You: GenAI Wisdom - **Amar Misra** — Senior Director & Chief Architect, Transversal (Capgemini)

Amar brings extensive experience architecting enterprise AI systems, specializing in generative models, NLP, and ethical AI deployment. His strategic vision helps organizations harness AI responsibly and effectively. Below are some insightful questions and explanations shared by Amar, showcasing his deep understanding of Generative AI architectures, project challenges, prompt engineering, ethics, and high-resolution content generation:

1.What is the Architecture of Generative AI

Generative AI works through the use of neural networks, specifically Recurrent Neural Networks (RNNs) and more recently, Transformers. Here's a simplified breakdown of how it functions:

- **Data Collection:** To begin, a substantial amount of data related to the specific task is gathered. For instance, if you want to generate text, the model needs a massive text corpus to learn from.
- **Training:** The neural network is then trained on this data. During training, the model learns the underlying patterns, structures, and relationships within the data. It learns to predict the next word, character, or element in a sequence.
- **Generation:** Once trained, the model can generate content by taking a seed input and predicting the subsequent elements. For instance, if you give it the start of a sentence, it can complete the sentence in a coherent and contextually relevant manner.
- **Fine-Tuning:** Generative AI models can be further fine-tuned for specific tasks or domains to improve the quality of generated content.

2.Describe a challenging project involving generative AI that you've tackled. What were the key challenges, and how did you overcome them?

Answering this question is really subjective to your projects and experiences. You can, however, keep these points in mind when answering questions like this:

- Select a specific project with clear AI challenges like bias, model accuracy, or hallucination.
- Clarify the challenge and explain the technical or operational diculty.
- Show your approach by mentioning key strategies you leveraged like data augmentation, model tuning, or collaboration with experts.
- Highlight results and quantify the impact—improved accuracy, better user engagement, or solving a business problem

3.How can prompts be strategically designed to elicit desired behaviours or outputs from the model? What are some best practices for effective prompt engineering?

Prompting is important in directing LLMs to respond to specific tasks. Eective prompts can even mitigate the need for fine-tuning models by using techniques such as few-shot learning, task decomposition, and prompt templates. Some best practices for prompt engineering include:

- **Be clear and concise:** Provide specific instructions so the model knows exactly what task you want it to perform. Be straightforward and to-the-point.
- **Use examples:** For in-context learning, showing a few input-output pairs helps the model understand the task the way you would like.
- **Break down complex tasks:** If the task is complicated, breaking it into smaller steps can improve the quality of the response.
- **Set constraints or formats:** If you need a specific output style, format, or length, clearly state those requirements within the prompt

4. What are some ethical considerations surrounding the use of generative AI?

The widespread use of GenAI and its use cases requires a thorough evaluation of their performance in terms of **ethics.** Some examples include:

- **Deepfakes:** Creating fake but hyper-realistic media can spread misinformation or defame individuals.
- **Biased generation:** Amplifying historical and societal biases in the training data.
- **Intellectual property:** Unauthorized use of copyrighted material in the data.

5. Can you discuss the challenges of generating high-resolution or long-form content using generative AI?

As you increase the complexity of AI generation, you should also tackle:

- **Computational cost:** High-resolution outputs require bigger networks and more computational power.
- **Multi-GPU training:** Larger models may not fit into a single GPU, requiring multi-GPU training. Online platforms can mitigate the complexity of implementing such systems.
- **Training stability:** Bigger networks and more complex architectures make it more challenging to maintain a stable training procedure.
- **Data quality:** Higher resolution and longer-form content require higher-quality data

Voice Powering the Future of GenAI-**Harsimar Singh** — AI-ML Consultant (Generative AI), KPMG

Harsimar is a specialist in multimodal AI and local language models, focusing on bridging technical complexity with business impact. He advises on AI alignment, evaluation, and real-world implementation challenges. Below are some detailed technical questions and insights shared by Harsimar, reflecting his expertise in Generative AI and Large Language Models

1. What are Dierent Methods of Tokenization?

Tokenization is the process of converting input (like text) into smaller units—tokens—which a model can process. Common methods include:

- **a. Whitespace Tokenization**
- Splits text on spaces.
- Simple but misses nuances like punctuation.
- "ChatGPT is great!" → ["ChatGPT", "is", "great!"]

b. **Word-level Tokenization**

- Splits text into words, handling punctuation better.
- Fails on out-of-vocabulary (OOV) words.

c. Character-level Tokenization

- Breaks text into individual characters.
- Useful for handling rare words or typos.
- "AI" → ["A", "I"]

d. Subword Tokenization (e.g., Byte Pair Encoding - BPE, WordPiece)

- Balances vocabulary size and coverage.
- Breaks rare words into known subwords.
- "unhappiness" → ["un", "happi", "ness"]

e. Byte-Level Tokenization

- Works at raw byte level (used in GPT-3 and beyond).
- Supports multilingual text and symbols without OOV issues.

2. What Are the Challenges in LLMs Based on Local Language Models?

a. Data Scarcity

- Limited high-quality datasets in regional languages.
- Lack of standardized corpora for pretraining.

b. Tokenization Complexity

- Complex scripts (e.g., Devanagari, Tamil) make tokenization harder.
- Multilingual tokenizers may be sub-optimal.

c. Cultural & Linguistic Diversity

- Local languages often have dialectal variations.
- Direct translation may lose contextual meaning.

d. Evaluation Metrics

- English-centric benchmarks don't translate well.
- Need for culturally relevant NLP evaluation datasets.

e. Bias and Representation

Underrepresented communities may be left out, leading to biased outputs.

3. How Do You Compare Different Models?

Model comparison typically considers:

Criteria	Description
Performance	Accuracy, F1-score, BLEU (for text), perplexity (for LLMs)
Inference Speed	How fast the model generates output
Training Time & Cost	GPU hours, compute eciency
Size & Architecture	Number of parameters, layers, attention heads
Generalization	Ability to adapt across domains or tasks
Multilingual or Multimodal Support	Extent of language or data type coverage
Alignment & Safety	Quality of outputs and toxicity filtering

Benchmarks like MMLU, HELLASWAG, and TruthfulQA are used for standardized comparison.

4. What Are Multimodal Capabilities of a Model?

Multimodal models can understand and generate across multiple data types like:

- Text + Image (e.g., ChatGPT with vision, GPT-4V)
- Text + Audio (e.g., Whisper, AudioLM)
- Text + Code (e.g., Codex)
- Text + Video (e.g., Sora, Flamingo)

Examples of tasks:

- Generating image captions
- Visual question answering
- Talking avatars
- Generating code from natural language

These models combine multiple input "modalities" in a single neural architecture for richer contextual understanding.

Aspect	Text Generation	Image Generation
Output	Sequences of tokens (words/subwords)	Pixel data (or latent representations)
Model Type	Transformer-based LLMs (e.g., GPT, BERT)	Diusion Models, GANs, Vision Transformers
Training Objective	Predict next token in sequence	Learn pixel or latent distribution from data
Tokenization	Subwords, bytes	Pixels, patches, or VQ-tokens
Evaluation	Perplexity, BLEU, ROUGE	FID, IS, human judgment
Challenges	Coherence, factuality	Realism, consistency, diversity

Resources for Continued Learning

Online Courses & Certifications

- **DeepLearning.AI** – Generative AI with Large Language Models (Coursera) A practical course covering fundamentals and applications of generative AI models like GPT.
- **AI For Everyone by Andrew Ng (Coursera)** A beginner-friendly introduction to AI concepts, helping build foundational knowledge.
- **Fast.ai** – Practical Deep Learning for Coders A hands-on course emphasizing practical coding skills in deep learning and generative models.
- **Udacity** – AI Programming with Python Nanodegree Covers key AI programming skills and foundational machine learning techniques.
- **OpenAI's API Documentation & Tutorials** Directly learn to build with OpenAI's models through ocial docs and code examples.

Books & Reading Materials

- ***Deep Learning*** **by Ian Goodfellow, Yoshua Bengio, and Aaron Courville** The definitive book for understanding neural networks and deep learning.
- ***Generative Deep Learning*** by David Foster Focused specifically on generative models like GANs, VAEs, and autoregressive models.
- **Research Papers & ArXiv.org** Stay current by reading the latest papers on Generative AI breakthroughs and advancements.

Tools & Practice Platforms

- **Hugging Face Transformers Library Practice** with state-of-the-art transformer models and experiment with fine-tuning.
- **Google Colab Notebooks Use free cloud GPUs** to experiment with generative AI code samples and projects.
- **Kaggle Competitions** Engage with real-world AI problems and datasets to apply generative modeling techniques.

OpenAI Community

Connect with developers using OpenAI tools and share projects or questions

Roadmap to Transition into a GenAI Role

Whether you're a software engineer, data analyst, product manager, or researcher — here's a phased roadmap to break into GenAI and land interviews confidently.

Phase 1: Build Solid Foundations (Weeks 1–3)

- Learn Python basics
- Understand AI/ML fundamentals
- Explore how Large Language Models (LLMs) work
  Tools: Coursera, DeepLearning.AI, Khan Academy, YouTube, ChatGPT

Phase 2: Hands-On Projects (Weeks 4–7)

- Build GenAI mini-projects
- Practice prompt engineering
- Try OpenAI & Hugging Face APIs
  Tools: Google Colab, PromptHero, GitHub, Hugging Face

Phase 3: Specialize Based on Your Background (Weeks 8–10)

- **Designers:** Try text-to-image tools (Midjourney, DALL·E)
- **Developers:** Learn RAG, LangChain, fine-tuning basics
- **Marketers/Content:** Use AI tools like Jasper, Writesonic
  Tools: Sora, LangChain, Pinecone, Claude

Phase 4: Interview & Portfolio Prep (Weeks 11–12)

- Review sample Q&A (like in this guide)
- Build a simple portfolio (GitHub, personal blog)
- Practice interviews with peers or mentors
  Tools: Topmate mock sessions, LinkedIn Projects, GitHub