

AI MASTERY RESOURCES

GenAI, Agentic AI & AI Governance

Basic / Intermediate / Advanced

Interview questions with preparation cues,
portfolio projects, and role transition guidance.

2026 Edition

Prepared for professionals building trusted AI careers.

Dr. Nabanita Sinha

How to use this question bank

Use the bank in passes. First answer aloud without notes. Then write a stronger version with assumptions, examples, and one limitation. Finally, practise applying the idea to a product, agent, governance process, or system you have built.

1. Diagnose the role

GenAI roles emphasise retrieval, evaluation, and product trade-offs. Agentic roles add tools, state, autonomy, and recovery. Governance roles add risk, accountability, evidence, and controls.

2. Answer with a frame

Clarify the user and decision, define the system boundary, state assumptions, choose a method or control, validate it, and explain what you would do next.

3. Make risk visible

Say what could go wrong, who could be affected, how you would detect it, what the fallback is, and which human or team owns the decision.

4. Practise evidence

Prepare two project stories with a baseline, measurable result, failure or limitation, and a clear explanation of what you personally did.

The three interview signals

Build	Can you turn a user need into a useful, evaluated GenAI or agentic system?
Bound	Can you set permissions, guardrails, human review, and safe failure modes?
Prove	Can you produce evidence that quality, risk, and value are understood over time?

Build useful, grounded, measurable language-model applications.

Current interviews test more than prompt writing. Be ready to explain retrieval, structured outputs, evaluation, safety, cost, latency, and how a prototype becomes a dependable product.

GenAI interview track

Progress from accurate definitions to system design and decision-quality judgment. In every answer, connect the technical choice to a user, risk, or measurable outcome.

Level 1 · Basic

Explain the core ideas simply, define assumptions, and connect them to a user outcome.

LLM foundations

1. What is a large language model?

Prepare: Describe next-token prediction, pretraining on broad text, and why fluent generation does not guarantee truth or a live connection to a database.

2. What are tokens and why do they matter?

Prepare: Explain that text is split into model-specific units that affect context limits, latency, cost, and prompt design.

3. What do temperature and top-p control?

Prepare: Describe sampling randomness, change one control at a time when tuning, and explain why lower randomness is often useful for factual or structured tasks.

4. What is an embedding?

Prepare: Explain a vector representation used to compare semantic similarity, then distinguish embedding search from the generative model that writes the answer.

5. What is hallucination in a GenAI system?

Prepare: Define an unsupported or invented claim and discuss grounding, abstention, citations, verification, and evaluation rather than treating it as only a prompt problem.

6. When would you use prompting, retrieval, or fine-tuning?

Prepare: Start with the simplest approach: prompting for behavior, retrieval for changing knowledge, and fine-tuning for repeatable behavior or format when evidence supports it.

Application patterns

7. What is retrieval-augmented generation?

Prepare: Retrieve relevant external context, place it in the model input, generate an answer, and evaluate both retrieval quality and answer quality.

8. What makes a good GenAI prompt?

Prepare: State the task, context, constraints, output format, examples, and what to do when information is missing; keep instructions testable.

9. Why use structured output or function calling?

Prepare: Constrain the response to a schema or a declared action so downstream code can validate it rather than parsing free-form prose.

10. How would you reduce a chatbot's hallucinations?

Prepare: Clarify the knowledge boundary, improve retrieval, require evidence, allow an explicit unknown response, validate high-impact claims, and measure the failure rate.

11. What should a basic GenAI evaluation contain?

Prepare: Use representative tasks, expected criteria, positive and adversarial examples, a baseline, a repeatable scorer, and a review process for ambiguous cases.

12. How would you explain an LLM failure to a non-technical stakeholder?

Prepare: Describe the user request, available evidence, model behavior, impact, likely cause, immediate mitigation, and the next experiment in plain language.

Level 2 · Intermediate

Show system thinking across data, retrieval, evaluation, and production constraints.

RAG and data quality

13. How would you choose a chunking strategy for RAG?

Prepare: Use document structure and retrieval intent, preserve useful metadata, avoid splitting critical units, and compare chunk sizes with a held-out evaluation set.

14. When would hybrid search beat vector search alone?

Prepare: Use lexical matching for exact names, codes, and rare terms alongside semantic retrieval; tune fusion and test on the queries users actually ask.

15. What is reranking and when is it useful?

Prepare: Retrieve a wider candidate set cheaply, then apply a stronger relevance model to order the smaller context passed to generation.

16. How do you evaluate retrieval separately from generation?

Prepare: Measure recall or hit rate for relevant evidence, context precision, citation support, and answer correctness; do not let a plausible answer hide retrieval failure.

17. How would you prevent stale or unauthorized data from entering a RAG answer?

Prepare: Track document versions and freshness, filter by tenant and user permissions before retrieval, log source identity, and fail closed when access cannot be verified.

18. How do you handle a query that has no good retrieved evidence?

Prepare: Set a relevance threshold, ask a clarifying question or abstain, explain the knowledge boundary, and record the miss for corpus or query improvement.

Product and operations

19. How would you reduce GenAI latency and cost?

Prepare: Measure tokens, model time, retrieval, tool calls, and retries; use smaller models, caching, streaming, batching, prompt compression, routing, and a quality floor.

20. How would you compare two LLMs for a product decision?

Prepare: Use the same representative task set, score quality and safety, measure latency and cost, inspect slices and failures, and include operational and contractual constraints.

21. What is the difference between offline and online GenAI evaluation?

Prepare: Offline tests are repeatable and fast; online signals show real user behavior but can be noisy. Use guardrails and connect both to the product decision.

22. How do you make citations trustworthy in a RAG application?

Prepare: Return source identifiers tied to retrieved passages, check that claims are supported, avoid citing irrelevant context, and test citation completeness and correctness.

23. How would you design a prompt or model change rollout?

Prepare: Version prompts and models, run regression and adversarial evaluations, canary the change, monitor quality and safety, and keep a fast rollback path.

24. What belongs in a GenAI observability trace?

Prepare: Capture request class, prompt and model versions, retrieval and tool events, latency, token usage, validation results, user feedback, and sensitive-data-safe identifiers.

Level 3 · Advanced

Connect architecture choices to reliability, safety, economics, and measurable user value.

Advanced system design

25. Design a multi-tenant enterprise RAG platform.

Prepare: Define tenant isolation, identity-aware retrieval, ingestion and deletion, indexing, freshness, model routing, audit logs, evaluation, quotas, and incident response.

26. How would you decide between a long context window and retrieval?

Prepare: Compare corpus size, freshness, relevance, cost, latency, privacy, citation needs, and failure modes; benchmark both on representative tasks.

27. How would you evaluate a GenAI copilot that drafts high-impact communications?

Prepare: Define the decision and harm model, require human review, test factuality and tone by segment, measure edit and override behavior, and set escalation rules.

28. How do you test prompt-injection resistance in a RAG application?

Prepare: Separate trusted instructions from retrieved content, create direct and indirect attack cases, test tool boundaries and exfiltration, and monitor failures without logging secrets.

29. What is a good strategy for model routing?

Prepare: Classify task complexity and risk, route to the cheapest model meeting the quality floor, preserve fallback behavior, and evaluate routing errors separately from model errors.

30. How would you investigate a quality regression after a corpus update?

Prepare: Compare retrieval distributions, document parsing, metadata filters, chunking, model and prompt versions, slice-level scores, and real traces before changing the model.

Leadership and judgment

31. When is a GenAI feature not worth deploying?

Prepare: If the user decision is unclear, evidence is weak, quality is below the cost or risk threshold, a simpler workflow works, or monitoring and ownership are missing.

32. How would you build a business case for a GenAI application?

Prepare: Quantify the decision or workflow baseline, expected time or quality improvement, adoption assumptions, model and operating costs, risk controls, and a measurable pilot.

33. How do you balance helpfulness with refusal behavior?

Prepare: Define allowed and disallowed outcomes, distinguish uncertainty from safety refusal, provide a useful safe alternative, and measure both harmful compliance and unnecessary refusal.

34. What would you include in a GenAI incident review?

Prepare: State the user impact and timeline, preserve safe evidence, identify contributing factors across data, model, prompt, and controls, assign corrective actions, and update evaluations.

35. How would you explain why benchmark scores are insufficient for launch?

Prepare: Benchmarks may not represent the domain, user distribution, tools, latency, safety, or workflow. Require task-level evidence and operational readiness.

36. How do you decide whether to fine-tune a foundation model?

Prepare: Confirm the problem is stable and repeated, collect representative high-quality examples, compare prompting and retrieval baselines, assess privacy and maintenance cost, and test for regressions.

Design tool-using systems that can act, recover, and stay within bounds.

Agent interviews increasingly focus on the boundary between a deterministic workflow and a model-directed agent. Strong answers make state, tools, permissions, stopping rules, evaluation, and human control explicit.

Agentic AI interview track

Progress from accurate definitions to system design and decision-quality judgment. In every answer, connect the technical choice to a user, risk, or measurable outcome.

Level 1 · Basic

Show that you understand the moving parts of a reliable tool-using system.

Agent foundations

1. What is an AI agent?

Prepare: Describe a system where a model helps choose actions or tools to pursue a goal, while distinguishing the model, orchestration code, tools, state, and guardrails.

2. What is the difference between a workflow and an agent?

Prepare: A workflow follows predefined code paths; an agent has more freedom to select steps dynamically. Choose the simpler predictable pattern when it is sufficient.

3. What makes a good tool definition?

Prepare: Give the tool a narrow purpose, typed inputs, clear descriptions, explicit errors, permission boundaries, and an idempotency or side-effect policy.

4. Why does an agent need a stopping condition?

Prepare: Without a bounded loop it can repeat, spend money, or take unintended actions. Use step, time, cost, confidence, and approval limits.

5. What is agent state?

Prepare: State is the information needed to continue a task, such as goal, plan, tool results, approvals, and checkpoints. Keep it explicit, scoped, and recoverable.

6. When should an agent ask a human for approval?

Prepare: Before irreversible, high-impact, ambiguous, external-facing, or privilege-changing actions, and whenever policy requires a person to review the evidence.

Reliability basics

7. Why are tool calls different from text generation?

Prepare: A tool call can change the world or expose data, so validate arguments, authorize the action, handle errors, and record what happened.

8. What is the role of tracing in agent systems?

Prepare: Trace model decisions, tool calls, state transitions, retries, approvals, and outcomes so teams can debug multi-step failures instead of inspecting only the final answer.

9. How would you prevent an agent from calling an expensive tool repeatedly?

Prepare: Set budgets and step limits, make progress observable, deduplicate requests, use timeouts and backoff, and require approval for costly actions.

10. What is a safe fallback for an agent?

Prepare: Return a clear partial result or ask for human help, preserve the state and evidence, and avoid silently guessing or repeating a failed side effect.

11. How should an agent handle a failed tool call?

Prepare: Classify the error, retry only transient failures with a limit, repair or clarify inputs when appropriate, record the failure, and stop safely when recovery is uncertain.

12. What would you measure for a simple agent?

Prepare: Task success, correct tool selection, argument validity, unnecessary steps, latency, cost, policy violations, recovery rate, and user satisfaction.

Level 2 · Intermediate

Explain orchestration, memory, tool safety, and evaluation of trajectories.

Orchestration and state

13. How would you choose between a single agent and a multi-agent design?

Prepare: Start with one agent or a workflow. Add specialized agents only when isolation, expertise, parallelism, or ownership clearly outweighs coordination and observability costs.

14. How do you design an agent loop?

Prepare: Define the input state, model decision, validated tool action, observation, state update, progress check, budget, and terminal outcomes before writing framework code.

15. What is the difference between short-term memory and long-term memory?

Prepare: Short-term memory supports the current task; long-term memory persists across tasks. Define scope, retention, consent, correction, retrieval quality, and deletion for both.

16. How do you make an agent resumable?

Prepare: Persist versioned checkpoints, tool results, approvals, and idempotency keys; recover from a known state and avoid repeating irreversible operations.

17. How would you coordinate parallel agent tasks?

Prepare: Partition work with clear contracts, bound concurrency, reconcile conflicting results, handle partial failure, and make the final decision traceable.

18. How do you prevent an agent from confusing untrusted text with instructions?

Prepare: Label data and instructions separately, constrain tool inputs, treat retrieved pages and messages as untrusted, and test indirect prompt injection.

Evaluation and integration

19. How do you evaluate an agent beyond the final answer?

Prepare: Score the trajectory: task completion, tool choice and arguments, evidence, recovery, policy compliance, unnecessary actions, and the quality of the final outcome.

20. What is an agent handoff?

Prepare: Transfer a task with explicit context, responsibility, permissions, and completion criteria from one specialist or service to another; log the handoff and preserve accountability.

21. How does a protocol such as MCP change tool integration?

Prepare: It standardizes how a client discovers and invokes tools and resources, but discovery is not trust. Apply identity, authorization, validation, and provenance controls.

22. How would you design retries for an agent?

Prepare: Classify transient versus permanent errors, use bounded exponential backoff and idempotency, preserve the original context, and avoid retrying unsafe side effects automatically.

23. How do you test an agent with nondeterministic behavior?

Prepare: Use fixed scenarios and fixtures, repeat runs, define rubric-based and code-based checks, inspect distributions and worst cases, and keep regression traces.

24. How would you expose an agent's plan to a user?

Prepare: Show the goal, intended actions, evidence, uncertainty, approvals, and progress at the right level of detail without exposing hidden chain-of-thought or sensitive internal data.

Level 3 · Advanced

Design enterprise agents for long-running work, security, interoperability, and accountable autonomy.

Enterprise agent architecture

25. Design an agent that processes a long-running procurement request.

Prepare: Define identity, policy checks, vendor data, budget and approval gates, tool permissions, durable state, retries, human escalation, audit evidence, and reconciliation.

26. How do you contain agent permissions?

Prepare: Use least privilege, scoped credentials, resource-level authorization, separate read and write tools, approval gates, sandboxing, expiry, and continuous audit.

27. How would you handle consistency when an agent updates multiple systems?

Prepare: Use idempotency keys, explicit workflow state, transactional boundaries where possible, reconciliation jobs, compensating actions, and a human path for ambiguous outcomes.

28. How do you evaluate an agent that can affect external systems?

Prepare: Use a staged environment, simulated or reversible tools, policy and safety tests, trajectory graders, canary release, blast-radius limits, and production monitoring.

29. What is the right autonomy level for a high-impact workflow?

Prepare: Match autonomy to reversibility, uncertainty, user expectation, harm potential, and evidence quality. Increase autonomy only when controls and measurements support it.

30. How would you design an agent budget?

Prepare: Budget steps, tokens, tool cost, wall-clock time, concurrency, and external side effects; surface remaining budget and choose a safe terminal action when it is exhausted.

Security and leadership

31. How would you threat-model an agentic system?

Prepare: Map assets, actors, trust boundaries, tools, data flows, and failure states; consider prompt injection, tool abuse, data exfiltration, privilege escalation, poisoning, and denial of service.

32. How do you make an agent's actions auditable without storing unnecessary sensitive data?

Prepare: Record actor, intent, policy decision, tool, target, time, result, approval, and version identifiers; redact payloads and use access-controlled evidence retention.

33. When should a multi-agent system be replaced with a workflow?

Prepare: When the task is stable, decisions are known, predictability and auditability matter more than flexibility, or the coordination overhead creates more risk than value.

34. How would you investigate an agent that appears successful but takes harmful shortcuts?

Prepare: Compare task success with policy and trajectory metrics, inspect tool traces and reward or rubric incentives, reproduce the path in a sandbox, then tighten constraints and tests.

35. How should agents interoperate across organisations?

Prepare: Use explicit capability descriptions, authenticated identities, scoped delegation, schemas, provenance, consent, rate limits, and a clear responsibility model for handoffs.

36. How do you communicate agent risk to executives?

Prepare: Frame the business outcome, autonomy boundary, plausible harms, control coverage, residual risk, monitoring signal, owner, and decision required rather than leading with model jargon.

Turn AI principles and regulation into operating controls and evidence.

Governance interviews are practical. Expect to map use cases to risk, assign accountability, review vendors and data, define human oversight, create documentation, and show how controls work across the AI lifecycle.

AI Governance interview track

Progress from accurate definitions to system design and decision-quality judgment. In every answer, connect the technical choice to a user, risk, or measurable outcome.

Level 1 · Basic

Define governance concepts clearly and show that controls support responsible delivery rather than stopping innovation.

Governance foundations

1. What is AI governance?

Prepare: Describe the policies, roles, processes, controls, and evidence used to guide AI from idea through retirement in line with law, ethics, and business risk.

2. What is the difference between AI ethics, AI safety, and AI governance?

Prepare: Ethics frames values, safety reduces harmful behavior and failure, and governance makes responsibilities and decisions operational and reviewable.

3. Why does an organisation need an AI inventory?

Prepare: It creates visibility into systems, owners, purpose, data, users, vendors, risk classification, lifecycle stage, and required controls.

4. What is a risk-based approach to AI?

Prepare: Scale scrutiny and controls to potential impact, affected people, use context, uncertainty, reversibility, and legal obligations rather than applying one checklist to every use.

5. What should an AI use-case intake form capture?

Prepare: Purpose, users, decision and intervention, data, model and vendor, autonomy, affected groups, geography, sensitivity, benefits, harms, owner, and deployment context.

6. What is human oversight in an AI system?

Prepare: A person must have the authority, information, competence, time, and ability to intervene or override when the system's impact or uncertainty requires it.

Documentation and accountability

7. What is an AI system card or model card used for?

Prepare: Document intended use, limitations, data and evaluation, risks, monitoring, ownership, and change history so people can make informed deployment decisions.

8. How would you assign accountability for an AI system?

Prepare: Name business, product, technical, data, security, privacy, legal, and operational owners; define decision rights and escalation rather than relying on a committee name.

9. What is an algorithmic impact assessment?

Prepare: A structured review of purpose, affected people, benefits, risks, alternatives, safeguards, residual risk, and monitoring before a consequential deployment.

10. How should a governance team support innovation?

Prepare: Offer clear intake, risk tiers, reusable controls, review SLAs, approved patterns, evidence templates, and early consultation so teams can move safely without surprise blocks.

11. What makes an AI policy enforceable?

Prepare: It has a clear scope, definitions, owners, required evidence, control gates, exceptions, monitoring, consequences, training, and a process for updating it.

12. How would you explain AI governance to an engineering team?

Prepare: Connect each requirement to a failure mode and delivery artifact: data lineage, evals, access control, logs, approvals, incident response, or a documented decision.

Level 2 · Intermediate

Show how to translate frameworks, regulations, and risk analysis into controls that teams can operate.

Risk and compliance operations

13. How would you apply the NIST AI RMF to a new use case?

Prepare: Use Govern, Map, Measure, and Manage as a lifecycle: establish accountability, understand context and harms, measure performance and risk, then manage and monitor decisions.

14. How does a risk-based regulation such as the EU AI Act affect governance work?

Prepare: Classify the system and role, identify prohibited or high-risk obligations where relevant, document evidence, assign responsibilities, and track timing and geography with legal partners.

15. How do you assess a third-party foundation-model vendor?

Prepare: Review data and privacy terms, security, subprocessors, model behavior and evals, service levels, change notice, audit rights, incident response, portability, and exit options.

16. How would you govern generative AI used by employees?

Prepare: Define allowed tools and data, protect confidential information, provide AI literacy, log or monitor appropriate usage, review high-impact use, and make approved alternatives easy to use.

17. What evidence would you request before approving a high-impact AI system?

Prepare: Purpose and impact assessment, data lineage, evaluation by relevant slices, human-oversight design, security and privacy review, limitations, monitoring, incident plan, and accountable sign-off.

18. How would you investigate an AI incident report?

Prepare: Triage harm and affected people, preserve evidence, stop or constrain the system if needed, identify root and contributing causes, notify owners or authorities as required, and track remediation.

Controls and measurement

19. How do you evaluate bias in an AI system?

Prepare: Define the affected groups and decision context, validate label and measurement quality, select relevant fairness and performance metrics, inspect intersections, and consider policy impact beyond a single score.

20. How do you govern explainability?

Prepare: Match the explanation to the decision, audience, and legal need; document limitations, provide meaningful factors or evidence where possible, and never imply certainty the model does not have.

21. What is a control library for AI governance?

Prepare: A reusable set of policies, technical safeguards, process gates, evidence requirements, and tests mapped to risks and system lifecycle stages.

22. How do you monitor an AI system after launch?

Prepare: Track data quality, drift, performance, fairness, incidents, overrides, user feedback, access, vendor changes, and control health with thresholds and named responders.

23. How would you handle an exception request from a product team?

Prepare: Clarify the business need and duration, assess residual risk and alternatives, define compensating controls and expiry, obtain the right approval, and record the decision.

24. What is AI literacy and why does it matter?

Prepare: People who develop, deploy, oversee, or use AI need role-appropriate understanding of capabilities, limitations, risks, instructions, and escalation paths.

Level 3 · Advanced

Demonstrate enterprise judgment across portfolios, jurisdictions, vendors, and residual risk.

Programme and portfolio leadership

25. Design an enterprise AI governance operating model.

Prepare: Define central policy and standards, business ownership, risk tiers, review forums, technical controls, evidence repositories, lifecycle gates, training, monitoring, and escalation.

26. How would you prioritise governance work across hundreds of AI use cases?

Prepare: Rank by impact, exposure, autonomy, affected population, regulatory obligations, uncertainty, control gaps, and reversibility; start with the highest-risk decisions and shared controls.

27. How do you decide whether residual AI risk is acceptable?

Prepare: Compare plausible harm and uncertainty with benefits, alternatives, affected-party protections, control effectiveness, legal requirements, reversibility, and accountable risk acceptance.

28. How would you prepare an organisation for an AI audit?

Prepare: Create an evidence map from obligations to controls, owners, system records, evaluations, approvals, monitoring, incidents, training, and change history; test retrieval and completeness.

29. How should governance handle a model or vendor change?

Prepare: Trigger impact review based on the change, rerun relevant evaluations, update documentation and contracts, assess regression and new risks, communicate changes, and preserve rollback options.

30. How would you report AI risk to a board or risk committee?

Prepare: Use a portfolio view of material systems, trend and exposure, top incidents, control coverage, exceptions, residual risk, regulatory developments, and decisions requiring oversight.

Complex judgment

31. How would you govern an agent that makes decisions across multiple jurisdictions?

Prepare: Map roles and obligations by location, constrain data and actions, document delegation and human oversight, use jurisdiction-aware controls, and obtain legal review for uncertain cases.

32. What governance challenges arise when a company builds on a foundation model it does not control?

Prepare: Separate provider, deployer, and user responsibilities; assess inherited and local risks, monitor provider changes, preserve evidence, manage data and exit, and avoid assuming vendor claims are sufficient.

33. How do you test whether human oversight is meaningful rather than ceremonial?

Prepare: Observe real workflows, time, information, override rates, incentives, training, and escalation outcomes; redesign when people cannot understand or act on the system's output.

34. How would you respond if a high-performing model fails one protected group?

Prepare: Pause or constrain the impacted decision, validate data and labels, investigate mechanisms and threshold effects, consult affected stakeholders, choose remediation, and document the trade-off and follow-up monitoring.

35. How do you measure whether an AI governance programme is effective?

Prepare: Measure inventory completeness, review cycle time, control coverage, high-risk remediation, incident response, override and complaint trends, audit findings, training effectiveness, and safe business outcomes.

36. When should an organisation stop or retire an AI system?

Prepare: When harms, legal exposure, control failure, drift, lack of evidence, or low value cannot be reduced acceptably; define a safe shutdown, data retention, user communication, and replacement path.

Projects for GenAI, agents, and governance

You do not need a previous AI job title to demonstrate readiness. A small set of finished, well-explained artifacts can show how you frame a problem, protect people and data, evaluate evidence, and communicate a decision.

What every project should include

Define the user, decision, and system boundary first. Include a baseline, data or policy assumptions, evaluation criteria, failure analysis, ownership, limitations, and a short README written for a non-technical stakeholder. A finished modest project is stronger than an unfinished advanced demo.

1. Grounded knowledge assistant

GENAI · FIRST PORTFOLIO PROJECT

Build a small RAG assistant over public documents with citations, abstention, and a test set of answerable and unanswerable questions.

Show: Ingestion and chunking choices, retrieval metrics, citation checks, failure examples, cost and latency notes, and a short user-facing demo.

2. Tool-using operations copilot

AGENTIC AI · CAREER TRANSITION

Create an agent that reads a request, uses two or three safe tools, asks for approval before a side effect, and resumes after a simulated failure.

Show: Tool schemas, state machine, permissions, approval UX, idempotency, traces, trajectory evaluation, and a clear safe fallback.

3. AI use-case risk register

AI GOVERNANCE · FIRST GOVERNANCE ARTIFACT

Create a realistic inventory of AI use cases and classify them by impact, autonomy, affected groups, data sensitivity, and regulatory context.

Show: Intake form, risk tier rationale, RACI, control mapping, evidence checklist, exception process, and a dashboard or executive memo.

4. GenAI evaluation harness

GENAI · EVALUATION-FOCUSED

Build a repeatable evaluator for a summarisation, extraction, or support task using code checks, rubric-based review, adversarial examples, and slice analysis.

Show: Dataset design, baseline, scoring rubric, inter-rater or grader checks, regression report, and an example of a change that was rejected.

5. Agent threat model and red-team lab

AGENTIC AI · SECURITY AND RELIABILITY

Threat-model a tool-using agent and demonstrate prompt injection, unauthorized tool use, data exfiltration, runaway loops, and bounded mitigations in a sandbox.

Show: Trust boundaries, attack cases, permissions, guardrails, traces, residual risk, and a test plan that can run in CI.

6. Responsible AI launch review

AI GOVERNANCE · PORTFOLIO CAPSTONE

Take a fictional high-impact AI use case from intake to launch recommendation, including impact assessment, evaluation, human oversight, vendor review, and monitoring.

Show: Decision memo, evidence map, control library, stakeholder consultation plan, incident playbook, and a defensible go / no-go recommendation.

A practical 8-week transition plan

Use the plan as a sequence, not a checklist to rush through. Keep one clean repository or evidence folder per finished project and make the decision, evidence, and limitations easy to review.

Weeks 1-2

Learn the foundations: LLMs, retrieval, structured outputs, agent loops, basic threat models, and risk-based governance. Explain each idea aloud.

Weeks 3-4

Complete the grounded knowledge assistant. Add an evaluation set, citations, abstention, and a short note on cost, latency, and data permissions.

Weeks 5-6

Complete the tool-using copilot or evaluation harness. Show state, tool validation, approval, recovery, traces, and trajectory-level tests.

Week 7

Create the AI use-case risk register or launch review. Map risks to controls, owners, evidence, monitoring, and an explicit recommendation.

Week 8

Polish two projects: README, screenshots, assumptions, limitations, reproducibility, and a two-minute walkthrough for each.

Project story checklist

Practise every project in this order. Treat each step as evidence you can explain in an interview:

- 1. Problem, user, and decision**
- 2. Data, permissions, and risks**
- 3. Baseline and success measure**
- 4. Architecture, method, and trade-offs**
- 5. Evaluation, result, and uncertainty**
- 6. What failed and what you changed**
- 7. Recommendation and owner**
- 8. Next experiment or control review**

This turns a portfolio project into strong behavioural, technical, and governance evidence.

Research basis and source notes

This question bank is an original synthesis created for interview preparation. The sources below informed the topic coverage and preparation guidance; the questions and cues are not copied from any source.

Anthropic: Building effective agents

<https://www.anthropic.com/research/building-effective-agents>

Workflows versus agents, simplicity, composable patterns, and the latency-cost trade-off of more autonomy.

OpenAI: A practical guide to building agents

<https://cdn.openai.com/business-guides-and-resources/a-practical-guide-to-building-agents.pdf>

Agent foundations, tools, orchestration, guardrails, human intervention, and production design considerations.

OpenAI: Evals guide

<https://platform.openai.com/docs/guides/evals>

Evaluation-driven development, representative datasets, graders, and regression testing for model applications.

Model Context Protocol specification

<https://modelcontextprotocol.io/specification/2025-06-18>

Standardized discovery and invocation patterns for tools and resources, with trust and authorization still left to implementers.

OWASP Top 10 for LLM Applications

<https://owasp.org/www-project-top-10-for-large-language-model-applications/>

Prompt injection, insecure output handling, supply-chain risk, data poisoning, excessive agency, and other LLM application threats.

NIST AI Risk Management Framework: GenAI Profile

<https://www.nist.gov/itl/ai-risk-management-framework/ai-risk-management-framework-generative-artificial-intelligence-profile>

Generative AI risk considerations and practical actions across Govern, Map, Measure, and Manage.

European Commission: AI Act

<https://digital-strategy.ec.europa.eu/en/policies/regulatory-framework-ai>

Risk-based regulation, prohibited practices, high-risk obligations, transparency, and responsibilities for providers and deployers.

Google: Secure AI Framework

<https://saif.google/>

Security practices for AI systems, including data, infrastructure, model, and application protections.

Coursera: How to get an AI governance job

<https://www.coursera.org/articles/ai-governance-job>

Role expectations across AI risk, compliance, privacy, model governance, and communicating controls to stakeholders.

Final reminder

Interviewers are evaluating how you reason, not whether you can recite every framework. Clarify the goal, define the system boundary, protect the data, measure the outcome, make the risk visible, and make the next decision obvious.