

1. Explain more about the use cases or projects that you have done, specifically on the AI side.
2. Did you also get involved in the indexing/data ingestion pipeline for ADAS?
3. How did you handle tables and Excel files containing signal data in the RAC system?
4. How did you chunk Excel/table data for indexing and vectorization?
5. Did you use fixed character-length chunking, or another strategy to prevent tables from being split?
6. What other chunking strategies have you used apart from recursive character splitting?
7. In your case, how many documents were there in total?
8. Was it a fixed set of documents, or did the document corpus increase over time?
9. Where was the data ingestion pipeline running?
10. When you said Azure AI, did you mean Azure AI Search or another Azure AI service?
11. Can you explain what a production-level indexing pipeline looks like?
12. What are the different blocks/components in a production-level Azure indexing architecture?
13. How do files initially get uploaded into Azure Blob Storage?
14. If users can upload multiple files through an upload UI, how do you handle each upload?
15. Once files reach Azure Blob Storage, how does the processing continue?
16. If indexing takes time, how do you handle the process from a user-experience perspective?
17. Is embedding generation handled by the backend or by Azure AI Search?
18. What algorithms are applied at search time?
19. Do you know about BM25?
20. How does vector search work?
21. If you have a lot of data to index, how do you optimize the vector database/index?
22. Have you deployed any system to production?
23. Have you deployed any agentic or RAG system to production?

24. Was the agentic system only for a demo/POC, or were there actual users?
25. How many users were using the system?
26. You mentioned LangChain; what framework did you use for the agent?
27. What tools did the agent have in your system?
28. What is the difference between the RAG/search tool and the document retrieval tool?
29. Can you explain the different nodes in your particular agent?
30. If a user attempts prompt injection, how do you detect/capture it?
31. Is prompt-injection handling a part of the LangGraph?
32. How do you handle authentication for API calls made by the agent?
33. How is Microsoft Entra ID authentication used within the LangGraph/agent?
34. Where do you need Entra ID authentication in the agent architecture?
35. Does the agent have its own permissions, or are the permissions delegated?
36. How do delegated permissions work for API calls within tools?
37. Can multiple users with different permissions use the same chatbot/agent?
38. How would the agent behave differently for users with different permissions?
39. Can you implement RBAC for this?
40. Have you used any tools for tracing and logging?
41. Have you used any LLM tracing tools?
42. Do you have any MLOps experience?
43. How do you version prompts over time?
44. How do you manage multiple prompt versions and experiments?
45. Have you done any multimodal processing?
46. Have you processed multimodal content such as text, tables, images, or diagrams?
47. Have you used any caching technique in your RAG or agent?
48. How would you cache chat history or context generated within the same chat?
49. Have you heard of prompt caching?

50. Where does prompt caching apply?

51. Have you used KV cache?

52. How does KV cache work?

53. Can you create a Pydantic class for the chunks being created for indexing?

54. What are decorators in Python?

55. How does a decorator work when you already have a function?

56. What is the behavioral difference between async and sync?

57. What is the difference between a Python list and tuple?

58. What is the difference between shallow copy and deep copy?

59. Do you have experience with any cloud platforms other than Azure?

@RAWHUL213