Gen AI Developer

Case Studies

Technical Assessment for Developer Level Positions

Confidential Document

Delphi Consulting LLC

Case Study 1: RAG System for Government Knowledge Assistant

Scenario

A government entity wants to build an internal knowledge assistant for their HR department. The system should help employees find answers about policies, procedures, and entitlements without needing to contact HR directly.

Document Corpus

- 200 PDF documents (HR policies, leave guidelines, benefits handbook, SOPs)
- Total: ~5,000 pages
- Languages: English primary, some Arabic sections
- Document types: Structured (forms, tables) and unstructured (policy narratives)

Requirements

Requirement	Target
Response time	< 3 seconds
Accuracy	90%+ for factual queries
Citation	Must reference source document and section
Languages	English queries, English responses (Phase 1)
Users	500 concurrent users

Sample Queries the System Must Handle

1. *"How many annual leave days am I entitled to after 3 years of service?"*
2. *"What is the process for applying for education assistance?"*
3. *"Can I carry forward my sick leave to next year?"*
4. *"What documents do I need for a business travel request?"*

Part A: System Design (50% Weightage)

Task 1: Document Processing Pipeline

Design the complete ingestion pipeline addressing the following:

1.1 Document Preprocessing

- How will you handle different PDF qualities (scanned vs. digital)?
- How will you extract text from PDFs with tables and forms?
- How will you handle documents with mixed layouts?

1.2 Chunking Strategy

Compare and recommend an approach:

Strategy	Chunk Size	Overlap	Best For	Limitations
Fixed-size				
Semantic (paragraph/section)				
Recursive				
Document-structure aware				

Your recommendation: _______________

Justification (3-4 sentences):

1.3 Metadata Design

Define the metadata schema for each chunk. Consider: source tracking, hierarchy, categorization.

Task 2: Retrieval Architecture

2.1 Embedding Model Selection

Model	Dimensions	Multilingual	Speed	Your Assessment
text-embedding-3-small	1536	Yes	Fast	
text-embedding-3-large	3072	Yes	Medium	
UAE-Large-V1	1024	Arabic-focused	Medium	
BGE-M3	1024	Yes	Medium	

Your recommendation: _______________

Justification: _______________

2.2 Vector Database Selection

Database	Managed Option	Hybrid Search	Filtering	Your Assessment
Pinecone	Yes	Yes	Yes	
Weaviate	Yes	Yes	Yes	
Qdrant	Yes	Yes	Yes	
PostgreSQL + pgvector	Self-hosted	Needs setup	Yes	
Azure AI Search	Yes	Yes	Yes	

Your recommendation: _______________

2.3 Retrieval Strategy

Design your retrieval approach and describe:

- Will you use hybrid search (vector + keyword)? Why?
- How many chunks will you retrieve (K)?
- Will you use re-ranking? Which model?
- How will you handle queries that span multiple documents?

Task 3: Generation Architecture

3.1 Prompt Design

Write the complete system prompt and user prompt template addressing:

- How do you instruct the model to use context?
- How do you handle citations?
- How do you handle "I don't know" cases?

3.2 Response Format

Define the expected response structure. Consider: answer, citations, confidence, follow-up suggestions.

Part B: Implementation (40% Weightage)

Task 4: Core RAG Pipeline

Implement the complete RAG pipeline with the following components:

1. Query embedding generation
2. Chunk retrieval using hybrid search
3. Re-ranking of retrieved chunks
4. Context building from chunks
5. Response generation with LLM
6. Citation extraction and validation

Task 5: Edge Cases & Error Handling

Implement handlers for these common edge cases:

- Query type detection (factual, procedural, comparative, out-of-scope)
- No relevant results found - suggest alternatives without hallucinating
- Low confidence retrieval - threshold handling
- Conflicting information detection (e.g., old policy vs. new policy)

Part C: Evaluation & Testing (10% Weightage)

Task 6: Evaluation Framework

Design your testing approach:

1. Provide 5 test cases across difficulty levels (easy, medium, hard)
2. For each test case, specify: query, expected facts in answer, expected source document
3. Implement retrieval evaluation: Recall@K, Mean Reciprocal Rank (MRR)
4. Implement generation evaluation: answer correctness, citation accuracy, hallucination detection

Evaluation Rubric - Case Study 1

Criteria	Points	What We're Looking For
Chunking strategy design	10	Understanding of trade-offs, practical choice
Embedding & vector DB selection	10	Justified selection considering constraints
Retrieval pipeline design	15	Hybrid search, re-ranking awareness
Prompt engineering	10	Clear instructions, citation handling
Core implementation quality	25	Clean code, proper abstractions, completeness
Edge case handling	15	Robustness, graceful failures
Evaluation approach	10	Practical test cases, relevant metrics
Code organization & clarity	5	Readable, documented, production-minded
TOTAL	**100**	

Time Allowed: 1-2 hours

Submission Format:

1. Design document (Part A) - PDF or Markdown
2. Code files (Parts B & C) - Python files or Jupyter notebook
3. Brief README explaining how to run the code