Interview Task: Build a Minimal “AI Knowledge Inbox”

Goal

Build a small production-style web app that lets users:

1. Save short notes or URLs
2. Ask questions over their saved content
3. Get answers powered by a simple RAG pipeline

This forces frontend + backend + async + AI integration + system design — without bloat.
Timebox expectation: 6–12 hours of real work.
Hard cap: 2–3 days calendar time.

Functional Requirements

1. Content Ingestion

1. Add text notes (plain text)
2. Add URLs (fetch page content server-side)

Store: Raw content, Metadata (timestamp, source type). No auth needed. Single-user is fine.

2. Semantic Search + RAG

1. Chunking strategy (simple is fine, but intentional)
2. Embeddings generation (OpenAI, local model, or similar)
3. Vector storage (in-memory, sqlite, or lightweight DB)
4. Enter a question and retrieve top relevant chunks
5. Pass context + question to LLM
6. Return answer with cited sources

3. Frontend

1. Add note / URL input
2. List saved items
3. Ask question interface
4. Display answer + source snippets
5. React with hooks, reasonable state management
6. Clarity over beauty

4. API Design

1. POST /ingest
2. GET /items
3. POST /query
4. Evaluate request/response shape, error handling, input validation, naming sanity

Non-Functional Expectations

Tradeoff Awareness

1. Chunking approach rationale
2. Vector store choice
3. What breaks at scale
4. Production changes

Debuggability

1. Structured logging
2. Clear error messages
3. Sensible HTTP status codes

Code Quality

1. Separation of concerns
2. No god files
3. No copy-paste soup
4. Clear naming

Tech Constraints (Aligned With Your Stack)

Backend: Node + Express or FastAPI, OpenAI or equivalent API, SQLite / in-memory / simple vector store

Frontend: React, Tailwind optional, shadcn optional

Avoid: Full auth systems, Kubernetes theater, Overengineering infra