Title: Therapy Use Case Senior Engineer Task – AI Session Management

Objective: Test architecture design, Node.js backend skills, and AI integration awareness relevant to therapy sessions.

Task Overview: Design and implement a backend service for therapist-client sessions, supporting: 1. Session logs (text/audio reference, timestamps) 2. Differentiating therapist vs client input 3. Providing a summary endpoint integrating with AI (mocked) 4. Scalable architecture for multiple sessions

---

1. Architecture Design (30-45 mins)
2. Draw an architecture diagram showing:
3. Session ingestion (audio → STT → diarization)
4. Storage (DB or in-memory)
5. AI summarization pipeline
6. API endpoints for session CRUD
7. Include data flow, components, and tech choices

Deliverables: - Diagram (draw.io or hand-drawn) - Explanation of design decisions (few sentences)

---

1. Backend Implementation (3-4 hours) Stack: Node.js + Express + any in-memory DB (or SQLite/ Postgres)

Endpoints: 1. Create Session POST /sessions Body: { therapistId, clientId, startTime } Response: { sessionId }

1. Add Session Entry POST /sessions/:sessionId/entries Body: { speaker: "therapist" | "client", content: string, timestamp } Response: { entryId }

2. Get Session Log GET /sessions/:sessionId Response: { sessionId, therapistId, clientId, entries: [{speaker, content, timestamp}] }

3. Summarize Session GET /sessions/:sessionId/summary Response: { summary: string }

4. Mock AI summarization (concatenate last 3 entries or placeholder text)

Bonus (Optional): - Role-based session filtering - Input validation for non-existent sessions or wrong speaker values

---

Evaluation Criteria: 1. Architecture Thinking 2. Backend Coding 3. AI Integration Awareness 4. Time Management (5-6 hours)

Deliverables: 1. Architecture diagram + brief explanation 2. Node.js project folder with working endpoints 3. README with run instructions and example requests

RAG (Embedding) + Transcription — Test Addition

Objective

Extend the take■home test by adding a Retrieval■Augmented Generation (RAG) layer. Session summaries and transcripts must be embedded and made semantically searchable so therapists can query past sessions efficiently.

Scope

- Transcribe audio session entries into text
- Generate and store embeddings for session summaries
- Enable semantic search across all sessions for a therapist

Data Model Additions

- **Session**: summary, embedding, optional transcript
- **SessionEntry**: text or audio, transcript text, timestamp

Required API Endpoints

POST /sessions/:id/transcribe
Transcribes an audio file into structured session entries. Mock transcription is acceptable for the test.

POST /sessions/:id/embed
Generates and stores embeddings for the session summary using OpenAI or a compatible embedding model.

GET /search/sessions?q=
Performs semantic search across embedded sessions and returns the most relevant sessions with similarity scores.

Acceptance Criteria

- Embeddings are generated and stored correctly
- Semantic search returns relevant sessions
- Architecture supports swapping real STT providers later

Bonus (Optional)

- Chunk long transcripts and embed per chunk
- Return highlighted text snippets with timestamps
- Use a vector database (Pinecone, Supabase, pgvector, etc.)