Noteless Technical Case (Software Engineer)

Time estimate: 3-4 hours

Task 1: Scalable & International Authentication and User Management

Scenario:
Noteless is expanding internationally and must support various local authentication systems (e.g., Norwegian BankID, Danish MitID, Email OTP, and additional local 2FA methods via Signicat). The platform also needs to manage segmented user profiles for different professions (e.g, GPs, Physiotherapists, Psychologists) and countries (Norway, Denmark, Europe).

Objectives:
Design a scalable authentication service that cleanly abstracts multiple methods while ensuring security, data privacy, and smooth integration with our Django REST Framework backend. Incorporate local compliance and language-specific flows to support internationalization and future B2B requirements.

Deliverables:

- A high-level architecture diagram showing how the authentication service interacts with the React/TypeScript frontend and Django backend, how third-party auth providers are integrated, and how user segmentation/localization is managed.
- A brief explanation of your design decisions and outlining strategies for scaling as the customer base grows from 1,500 to 15,000+.

Task 2: Integration of AI Models for Enhanced Medical Note Generation

Scenario:
Noteless wants to build a modular architecture for AI features to be able to quickly iterate and implement new models and features. The company currently uses different AI models hosted on providers like Azure ML and GCP Vertex. The goal is to be able to quickly implement, test, and roll out new models and features for specific user groups.

Objectives:
Propose a strategy that allows easy addition or swapping of AI models and services. Also, design the solution to support A/B testing, feature flags, and KPI monitoring (for example, user feedback and note accuracy).

Deliverables:

- An integration diagram that illustrates how our Django/DRF backend will call the various AI services, the data flow from raw audio to structured notes, and the control points for testing and feature management.
- A short explanation describing your approach to balancing performance, cost, and reliability when using multiple cloud AI providers.