

Studio Frontier Full-stack developer test

Build a small full-stack web application that allows users to create private notes, share them using a unique link, and use AI to summarize the note after unlocking it.

This assignment is **language and framework agnostic** for the backend, and **framework-flexible** for the frontend.

Functional Overview

1. A user creates a private note.
2. The system generates a shareable link and a password.
3. Anyone with the link can unlock the note using the password.
4. After unlocking the note, the user can click “**Summarize this note**” to get an AI-generated summary.

Requirements

1. Create Note (Backend API)

Create an endpoint that:

- Accepts a text note (up to 500 characters).
- Stores the note in any persistent store (SQL, NoSQL, file-based, etc.).
- Returns:
 - A unique URL.
 - A password required to view the note.

Validation

- Note cannot be empty.
- Note must be under the character limit.
- Return clear and structured error messages.

2. View Note (Frontend + Backend)

When a user navigates to the note URL:

- Display a password input UI.
- Verify the password using the backend API.
- On success: Show the note text.
- On failure: Show an error message.

3. AI feature (summarize the note)

- Show a button labeled “**Summarize this note**”
- On click:
 - i. Send the note text to the backend
 - ii. Backend calls an LLM API to generate a short summary
 - iii. Return the summary to the frontend
- Display the summary below the note

AI Integration Guidelines

- The LLM call **must be made from the backend**, not directly from the frontend
- The summary should be:
 - Short (3–5 bullet points or a short paragraph)
 - Based only on the note content
- Handle:
 - Loading states
 - API failures
 - Empty or invalid responses gracefully

You may use **any LLM provider** (OpenAI, Anthropic, Gemini, local model, etc.).

The goal is **correct integration and clean flow**, not AI sophistication.

4. Additional Functional Requirements

- A note should only be visible when the correct password is entered.
- If a note ID does not exist, show a proper “note not found” message.
- No login or user accounts required.

5. Frontend Styling Requirement

The frontend must use at least one modern styling or UI component framework, such as:

- Tailwind CSS
- shadcn/ui
- Bootstrap
- Material UI
- Chakra UI
- Any similar framework is acceptable.

Note: You do not need to create a visually complex interface. Basic, clean usage of a UI framework is sufficient. The purpose is to confirm familiarity with modern tooling, not design skills.

6. Optional Enhancements (Bonus - Not required, but will help your evaluation)

- Note expiry option
- Copy-to-clipboard for the shareable URL
- Unit or integration tests

Deliverables

1. GitHub repository link (frontend + backend).
2. README with setup instructions.
3. Short demo video and live deployment (Render, Vercel, Netlify, etc.)
4. One short section in the README describing potential future improvements.

Scoring

1. 50 marks for modular code implementation
2. 30 marks for the overall experience, correct success and error messages etc.
3. 10 marks for ideas on further improvements.
4. 10 marks bonus (subjective)