Tutor Flow

EMPOWERING POTENTIAL

Friday, 08 August 2025

Project Brief:
Mini-Note Taking Widget

Hey Josiah,

This small project is designed to test how quickly you learn, problem-solve, and deliver clean, logical work — with or without experience. You're free to use ChatGPT, blogs, documentation, YouTube, or whatever you need to figure things out. We want to see how you think, how you build, and how you explain your process.

The Task

Build a simple, clean note-taking widget that lets a student write and manage study notes by subject and grade.
This feature is inspired by real use cases inside TutorFlow. You're building a lightweight version of it, using mock data only (no backend required).

Tech Stack

- Next.js
- TypeScript
- Tailwind CSS (preferred styling)
- Mock data only (no Supabase needed)

Core Features

Your widget should allow the user to:

1. **Write a new note**
   - Title
   - Content (can be plain text)
   - Grade (dropdown: e.g. Grade 10, 11, 12)
   - Subject (dropdown: e.g. Maths, Life Sciences, Geography)

2. **View saved notes**
   - Show a list of all saved notes
   - Show title, subject, grade, and a preview of the content

3. **Delete or edit a note**
   - Ability to remove a note
   - Ability to update it (optional but great)

4. **Persist notes temporarily**
   - Store notes in memory or localStorage (no need for a database)

Deliverables

- GitHub repo (or zip file)
- A short README explaining:
  - What you built
  - Any assumptions made
  - What you would improve or do next
  - Rough estimate of time spent

Time Limit

- 2–3 hours maximum
- Please complete and submit within 24 hours

Bonus

- Filter notes by subject or grade
- Use reusable components (e.g. <NoteForm />, <NoteCard />)
- Show character count or validation
- Mobile-friendly layout (Essential)

What We're Evaluating

- Speed - Can you build a working solution quickly using the tools provided?
- Clarity - Is your UI clean and easy to follow?
- Learning - Did you clearly figure things out using AI or docs?
- Logic - Does the feature make sense? Are the steps clear and smooth?
- Code Quality - Is your code readable, organized, and logically structured?
- Ownership - Do you explain what you did and why in the README?

Don't aim for perfection — aim to deliver something usable, clean, and well thought-out in a short space of time.

Good luck — have fun with it!
Let us know if you have any questions.

TutorFlow Team 👌