

Mern-Stack Developer POC Task

Project Overview

Project Name: AI Note-Taking App

Duration: 1-2 days

Stack: Next.js, TypeScript, Hono.js, PostgreSQL, shadcn/ui

Objective

Build a simple note-taking application with AI-powered features to demonstrate full-stack development skills.

Core Features (Must Have)

1. Authentication

- User registration and login
- Protected routes
- Simple user profile

2. Notes Management

- Create notes with title and content
- View all user notes
- Edit and delete notes
- Search notes by title

3. AI Features (Simple Integration)

- **AI Summary:** Generate summary of long notes
- **AI Improve:** Improve note content (grammar, clarity)
- **AI Tags:** Auto-generate relevant tags for notes

4. Basic UI

- Clean interface using shadcn/ui
- Responsive design
- Dark/light theme toggle

Technical Requirements

Frontend

- **Next.js 14** with App Router
- **TypeScript** (strict mode)
- **shadcn/ui** for components
- **Tailwind CSS** for styling
- **React Hook Form** for forms

Backend

- **Hono.js** for API routes
- **PostgreSQL** or **MongoDB** database
- **Drizzle ORM** or **Prisma** or **Mongoose**
- **Zod** for validation

Authentication (Choose One)

- NextAuth.js
- Clerk
- Better Auth

AI Integration

- **Any AI API** (OpenAI, Anthropic, Google, Cohere, etc.)
- Simple text processing features
- Proper error handling for API calls

Key Components

- NoteCard - Display note preview
- NoteEditor - Rich text editor for notes
- AIButton - Trigger AI features
- SearchBar - Search notes
- ThemeToggle - Switch themes

Deliverables

1. **GitHub Repository** with README
2. **Live Demo** (deployed on Vercel)
3. **Working AI features** with OpenAI integration
4. **Clean TypeScript code**

5. **Responsive UI** with shadcn/ui

Evaluation Criteria

Technical (60%)

- Clean TypeScript code
- Proper Next.js structure
- Database design
- API implementation
- Error handling

AI Integration (25%)

- Working OpenAI integration
- Proper error handling for AI calls
- User-friendly AI features

UI/UX (15%)

- Clean interface with shadcn/ui
- Responsive design
- Good user experience

AI API Options (Choose Any)

- OpenAI API (GPT-3.5/GPT-4)
- Anthropic Claude API
- Google Gemini API
- Cohere API
- Hugging Face API
- Any other AI/ML API of your choice

Note: You have complete freedom to choose any AI API or service. The goal is to demonstrate integration skills, not specific API usage.

Success Criteria

- All core features working
- AI integration functional
- Clean, readable code
- Deployed and accessible demo

- Good documentation

Focus: Build a working application with clean code rather than trying to implement every possible feature. Quality over quantity!