

Smart Notes - A Notes Application

Built with Next.js, TypeScript, SQLite, Prisma

Task 7

WNP. Karunarathna

Explain the Problem & User

The Problem

- Digital note-taking is fragmented - users switch between multiple apps
- Poor search capabilities - finding specific notes is time-consuming
- Lack of organization - no efficient tagging or categorization

Target Users

- Students organizing lecture notes and research
- Professionals managing meeting notes and project ideas
- Content creators collecting ideas and inspiration

Key Features Solving These Problems

- CRUD Operations: Create, read, update, delete notes seamlessly
- Smart Search: Find notes by title, content, or tags instantly
- Tag System: Organize notes with flexible categorization

Design

1. User inputs data to the Frontend.
2. Frontend sends this data to the API Route.
3. API Route queries the Database.
4. Database returns data to the API Route.
5. API Route delivers this data back to the Frontend.
6. Frontend displays the data to the User.

Demonstrate a Feature

- Feature Showcase: Smart Note Creation
- Feature Showcase: Advanced Search
- Feature Showcase: Note Management

Challenges Encountered

- Real-time Search Performance
- Tag Management Complexity

AI Tools Integration

- Cursor AI: Used for rapid component generation and debugging
- Vibe Coding Approach: Conversational development with natural language prompts

Future Development Ideas

- Authentication: User accounts with secure note privacy
- Note Templates: Pre-defined structures for meeting notes, lists, etc.
- Note Sharing: Public links for note sharing
- Version History: Track note changes over time