Building Documetech Using AI Tools - A Step-by-Step Guide

Introduction

This guide explains how I used AI tools like ChatGPT, Lovable.dev, and Curser AI to build the Documetech project, including a full authentication system and a working CRUD module. This documentation helps others replicate the process using smart prompt writing and AI tool integration.

Step 1: Writing the Initial Prompt for Authentication

I defined the authentication flow and tech stack:

- Frontend: React
- Backend: Node.js with Express and MongoDB
- Features: Login, Signup, Forgot Password with code, Guest Login, JWT-based auth, dashboard after login (image-based design)

Prompt example:

"Frontend in React for login/signup/forgot + dashboard after login using image-based design. Backend in Node.js with MongoDB for auth, guest login, code verification, JWT-based auth, and password reset logic."

Step 2: Refining the Prompt with ChatGPT

I gave the prompt to ChatGPT to improve structure, logic, and UX. It returned a polished, production-ready prompt that could be easily understood by AI code generators like Lovable.dev.

Step 3: Generating the UI with Lovable.dev

With the improved prompt and three design reference images, I used Lovable.dev to generate the authentication UI and backend. It matched 80% of my design expectations.

Step 4: Enhancing Post-Login Page & Module Creation

I requested ChatGPT to write prompts for:

- A post-login dashboard page
- A form-based CRUD module

After generating these prompts and supplying a layout image, Lovable.dev created the remaining pages and UI components.

Step 5: Downloading Code from GitHub

I downloaded the generated code from Lovable.dev's GitHub export. It included complete React frontend and Node.js backend using a custom structure.

Step 6: Backend API Creation & Integration Using Curser AI

I gave this code to Curser AI and used prompts (refined by ChatGPT) to:

- Create backend APIs for auth and module
- Integrate those APIs into the existing frontend

Curser AI successfully completed the integration.

Final Output

- Functional authentication (Signup, Login, Guest Login, Forgot Password with verification code)
- Dashboard auto-redirect after login
- One CRUD module (UI + backend)
- Built with React, Node.js, MongoDB

Tools Used

Tool	Purpose
ChatGPT	Prompt refinement, logic debugging
Lovable.dev	UI + backend generation from design/images
Curser AI	Backend API development and integration

Key Learnings

- Good prompts = better results

- Reference images help generate accurate UI
- Combining ChatGPT + Lovable + Curser can create a full-stack app
- Prompt iteration is critical for perfect results

Conclusion

Using a combination of AI tools and prompt engineering, I created a production-grade authentication system with a CRUD module. This workflow showcases how modern AI tools can accelerate full-stack development efficiently.