

1. Requirements Gathering

Prompt:

"Create a full list of functional and nonfunctional requirements for a web application called PromptPath, an AI-driven assistant that helps users craft, optimize, and save prompts for AI models like ChatGPT, MidJourney, and Stable Diffusion."

2. User Personas

Prompt:

"Generate 3 detailed user personas for the PromptPath application, including each persona's background, goals, frustrations, motivations, and how they would use the platform."

3. Use Case Diagram and Descriptions

Prompt:

"Generate a Use Case Diagram (in PlantUML or markdown format) and detailed textual descriptions for PromptPath. Include user actions like logging in, creating prompts, optimizing prompts, saving prompts, browsing a community prompt library, and rating prompts."

4. System Architecture Diagram

Prompt:

"Create a high-level system architecture for PromptPath, describing frontend (React, Next.js), backend (Node.js, Express, TypeScript), database (PostgreSQL), AI API integration (OpenAI), and Redis caching."

5. UI Wireframes

Prompt:

"Design basic low-fidelity wireframes for PromptPath's major screens:

Home/Landing page, Login/Register page, Prompt Editor page, Community Library page, and Profile page. Represent the wireframes using ASCII art or textual description."

6. Frontend Setup (React + Next.js)

Prompt:

"Generate a starter Next.js project structure for PromptPath, using TypeScript and Tailwind CSS. Include basic pages and components folders, and sample configuration files."

7. Prompt Editor Frontend Code

Prompt:

"Write a React (Next.js) component for a Prompt Editor. It should allow users to input a prompt into a textarea, click a button to send it to the backend for AI optimization, and display the optimized prompt result."

8. User Authentication Frontend

Prompt:

"Create a simple login and signup page using React and Next.js for PromptPath. Users should be able to log in via email/password or Google OAuth (assume Firebase Auth or Supabase Auth)."

9. Backend API (Node.js + Express)

Prompt:

"Write Express.js (with TypeScript) API endpoints for PromptPath:

- **POST /api/optimize-prompt:** Receives user prompt, sends it to OpenAI, returns optimized prompt.

- POST `/api/signup`: Registers a new user.
 - POST `/api/login`: Authenticates a user.
Include basic error handling and use JWTs for session management."
-

10. Database Schema (PostgreSQL)

Prompt:

"Design a PostgreSQL schema for PromptPath. Tables: Users, Prompts, Optimization History, and Community Ratings. Include field types, primary/foreign keys, and any necessary constraints."

11. OpenAI API Integration Code

Prompt:

"Generate a Node.js service function in TypeScript that calls the OpenAI API, sending a user's prompt and returning the optimized result. Handle API errors gracefully."

12. Community Prompt Library Frontend Code

Prompt:

"Create a React (Next.js) page to list community prompts in a grid layout, showing prompt title, short description, rating, and a button to view more. Fetch data from a sample backend endpoint."

13. Testing: Frontend Unit Tests

Prompt:

"Write Jest and React Testing Library test cases for the Prompt Editor component:

- Renders correctly

- Sends input to backend on submit
 - Displays optimized prompt on response."
-

14. Testing: Backend API Tests

Prompt:

"Create Jest test cases for the backend Express.js API endpoints `/api/optimize-prompt`, `/api/signup`, and `/api/login`. Test for success, failure, and invalid input cases."

15. README.md File

Prompt:

"Generate a professional `README.md` file for PromptPath. Include project overview, features list, technology stack, how to run the project locally, how to deploy, and architecture diagrams."

16. Planning.md File

Prompt:

"Generate a full `planning.md` file that contains the requirements, personas, use case diagram, system architecture, wireframes, and all early planning documents for PromptPath."

17. Deployment Instructions

Prompt:

"Write detailed deployment steps to:

- Deploy the frontend (React, Next.js) to Vercel

- Deploy the backend (Node.js, Express) to AWS Lambda using serverless
Include how to configure environment variables for the OpenAI API key, database, and authentication."