

AI Usage Report

This document outlines the strategic use of AI assistants in the development of the ADmyBRAND AI Suite landing page. AI was used as a collaborative partner to accelerate development, tackle complex challenges, and ensure a high-quality final product within a tight timeframe.

AI Tools Used

- **Primary Assistant: Claude**, used for creative generation, complex component creation, code refactoring, and TypeScript assistance.
 - **Secondary Assistant: Gemini**, primarily used for logic generation and debugging tasks.
-

Key Use Cases

AI was integrated throughout the development lifecycle for various tasks:

1. **Complex Component Generation:** For highly complex visual elements like the 700-line CSS-based dashboard graphic, AI was instrumental. It would have been unfeasible to build from scratch in the given time. I provided high-level concepts, and the AI generated the intricate structure, which I then refined.
2. **Advanced Logic Implementation:** AI was tasked with writing the logic for challenging features, such as the interactive pricing calculator's feature of updating values dynamically as the user interacts with the slider.
3. **Boilerplate Code & Styling:** For standard components, I would provide specific, declarative instructions (e.g., layout, colors, effects) to generate initial JSX/TSX and Tailwind CSS code, which served as a strong starting point.
4. **TypeScript Error Resolution:** When faced with complex or tedious TypeScript errors, I would provide the component code to the AI to identify and correct type-related issues, significantly speeding up development.
5. **Code Refactoring & Best Practices:** I regularly asked the AI to review my components and suggest optimizations, ways to improve modularity, or refactor the code to align with modern React best practices.
6. **Creative Exploration:** At times, I would provide a fully functional component and ask the AI to generate a completely different design with the same content. This allowed me to explore alternative visual ideas and create a final design that was an amalgamation of my own vision and the AI's suggestions.
7. **Documentation:** This very report is a product of AI assistance. I provided rough, high-level notes about my process, and the AI helped structure, format, and refine them into a professional document.

Sample Prompts

Here are examples that reflect the types of prompts used during development:

1. **For a specific UI component:** > “Create a responsive React card component using TypeScript and Tailwind CSS with a glassmorphism effect. The background color should be `#1a1a1a` with 50% opacity and a backdrop blur. The border should be `1px solid #333333`. On hover, it should scale up slightly using Framer Motion. The card needs to display a title, a description, and a list of features passed as props.”
2. **For code optimization:** > “Here is the code for my HeroSection.tsx component. Can you review it and suggest ways to make it more modular and reusable? Specifically, can the animated text be broken out into its own component?”
3. **For debugging:** > “I’m getting a TypeScript error in this file: `Type 'string | undefined' is not assignable to type 'string'`. I’ve checked the props but can’t find the issue. Here is the complete file, please find and fix the error.”

AI vs. Manual Work Split

The project’s workload can be estimated as a **50/50 split** between AI-generated code and manual coding/customization.

- **AI-Generated (50%):** This portion primarily consists of the initial code for complex components (like the dashboard graphic), boilerplate for simpler UI elements, specific logic functions (calculator), and solutions to stubborn bugs. The heavy reliance was a strategic choice to meet the project deadline while dealing with external factors.
- **Manual Coding & Customization (50%):** My work was focused on the areas where human oversight is critical:
 - **Architecture & Integration:** Setting up the Next.js project, defining the overall folder structure, and ensuring all AI-generated components were correctly integrated and worked together seamlessly.
 - **Understanding & Refining:** I made it a point to first understand the code AI provided before implementing it. This often involved significant refactoring and tweaking to match the project’s specific needs and my own coding standards.
 - **Final Touches:** Fine-tuning all Framer Motion animations, adjusting styles for pixel-perfect alignment, and ensuring the overall user

experience felt cohesive and polished. The final product is an **amalgamation**, a collaboration between my direction and the AI's execution.