

AI Usage Report

Author: Ishaan Singla

Date: July 28, 2025

AI Tools Used

- **ChatGPT-4 (GPT-4o)** – for boilerplate scaffolding, prompts, design structure, and troubleshooting
 - **v0.dev** – to generate initial UI layout and design inspiration
 - **GitHub Copilot** – for inline TypeScript and utility code suggestions
-

Key Use Cases

- Scaffolding dashboard layout using Next.js + App Router structure
 - Creating charts (line/bar/pie) with sample data using Recharts
 - Building responsive card and table components with Tailwind & ShadCN
 - Fixing TypeScript issues with imports and types
 - Generating utility functions for exporting CSV and ZIP
 - Improving folder/component organization and naming conventions
 - Prompting design suggestions based on Vercel/Linear UI patterns
 - Drafting the initial README and documentation outline
-

Sample Prompts

1. *“Build a modern dashboard layout using Next.js 14 and ShadCN UI with responsive metric cards and a line chart.”*
 2. *“Fix TypeScript error for file-saver import and add type declarations.”*
 3. *“Generate a CSV + ZIP export utility using JSZip and FileSaver.”*
-

AI vs Manual Work Split

- **AI-generated (~55%)**
 - Initial layout scaffolding
 - Recharts chart components
 - Export logic for CSV/ZIP
 - DataTable boilerplate logic
 - Sample mock data structure
 - Skeleton README and AI usage outline
 - **Manual Work (~45%)**
 - Fine-tuned filters and advanced search
 - Integrated dark/light theme logic
 - Styled and refined chart responsiveness
 - Created loading skeletons and transition states
 - Made ZIP downloads work seamlessly across browsers
 - Debugged TypeScript issues and cleaned imports
-

Customization Highlights

- Unified color and spacing scale for consistent UI
 - Manual conversion of several JS code snippets to TypeScript
 - Created reusable components for charts and filters
 - Refined mobile layout to avoid crowding or overflow
 - Integrated theme switcher using ShadCN's provider system
-

Summary

AI allowed me to rapidly scaffold the dashboard, generate repetitive code, and overcome bugs quickly — giving me more time to focus on **polish, UX, and custom design work**. Instead of blindly using suggestions, I treated AI as a pair programmer and refactored every part to meet my standards.