

# Building a Modern Portfolio Website with AI Agentic Tools

---

## A Case Study in AI-Assisted Web Development

---

**Project Type:** Personal Portfolio Development

**Timeline:** Under 45 minutes (excluding content preparation)

**Tools Used:** Manus AI Agent, ChatGPT-5 (prompt refinement)

**Date:** November 2025

---

## Executive Summary

---

This case study documents the rapid development of a professional, feature-rich portfolio website using AI agentic tools. The project demonstrates how modern AI assistants can dramatically accelerate web development workflows while maintaining high code quality and design standards. Through careful prompt engineering and iterative collaboration with an AI agent, a complete portfolio website with advanced features was deployed in under 45 minutes of active development time.

The project serves as part of a broader exploration into practical applications of AI tools in professional workflows, highlighting both the capabilities and limitations of current AI-assisted development approaches.

---

## Project Overview

---

### Objective

The primary goal was to create a modern, minimalist personal portfolio landing page that would serve as a professional showcase for cybersecurity and IT transition expertise. The portfolio needed to balance aesthetic appeal with practical

functionality, including bilingual support and access control features to protect sensitive professional information.

## Key Requirements

The project demanded a sophisticated set of features that would typically require significant development time. The portfolio needed to incorporate a light violet-blue gradient theme with professional typography, fully responsive mobile-first design, and multiple interactive sections. Beyond basic presentation, the site required bilingual English-French support, PDF viewing capabilities for certifications and CV, a dynamic projects grid, and a security gate to control access via URL parameters.

---

# The Importance of Vision and Prompt Engineering

---

## Initial Prompt Development

The foundation of successful AI-assisted development lies in having a clear, specific vision of the desired outcome. Before engaging with the AI agent, the initial project requirements were refined using ChatGPT-5 to ensure clarity, completeness, and technical feasibility. This preparatory step proved crucial in achieving rapid development without extensive back-and-forth clarifications.

A well-crafted prompt serves as a blueprint that guides the AI agent through complex implementation decisions. Rather than vague requests like “build me a portfolio,” the refined prompt specified exact measurements, color schemes, layout proportions, file naming conventions, and behavioral expectations. This specificity allowed the AI to make informed decisions about technology choices, component architecture, and styling approaches without requiring constant guidance.

## The Original Prompt

The initial request provided to Manus AI was comprehensive and detailed:

*“Build a modern, minimalist personal portfolio landing page with a professional, sober aesthetic and a light violet-blue gradient theme color for accents and highlights.*

**Theme & Layout:** Background should be very light grey (1-2 shades off white) to reduce eye strain. Typography must use clean sans-serif modern font (Inter, Roboto, or similar). Layout must be fully responsive and mobile-friendly, collapsing into a centered, single-column scroll feed on smaller screens. Maintain symmetry, whitespace, and calm visual hierarchy.

**Header:** Simple white bar across the top, separated from content by a 2px theme-colored underline. Right-aligned clickable contact icons for WhatsApp, Email, GitHub, LinkedIn. Icons should be grey by default, colorize to their brand colors on hover with smooth but fast transition animation. Height around 1-2 cm, slim and clean.

**Main Section:** Left  $\frac{3}{4}$  of the page displays large Name header at top, with subheaders beneath in lighter grey. Directly below are two square theme-colored buttons side by side for ‘Certifications’ and ‘CV’ with small rounded corners. Under the buttons, an ‘About Me’ block with centered paragraph of placeholder text in dark grey. Right  $\frac{1}{3}$  of the page shows profile picture in a square frame with soft violet-blue drop-shadow, vertically centered relative to the About Me section.

**Projects Section:** Located below main content, displaying rows of three square thumbnails with theme-colored drop-shadows. Thumbnails appear only if corresponding files exist. Each links to a PDF project page. File naming convention: `project_thumb_X_Y.jpg` for thumbnails, `project_body_X_Y.pdf` for linked documents in projects subfolder.

**Code & Structure:** Project must be self-contained inside `Index/` folder with all assets. Designed for GitHub Pages deployment with no server-side rendering. Use only client-side JS when needed. Write clean, readable, fully commented code with clear section comments and inline explanations for customization. Use CSS variables in `:root` for theme colors and typography. Ensure all paths are relative for portability.

**Visual Tone:** Minimal, balanced, and professional with subtle color use, clear typography, and breathing room. No filler content except 3 placeholder images for projects section.”

This level of detail provided the AI agent with sufficient context to make appropriate architectural decisions, select suitable technologies (React, Tailwind CSS, Vite), and implement features that aligned with the overall vision.

---

# Development Process

---

## Phase 1: Project Initialization and Planning

The development began with Manus AI initializing a static web project using modern React 19 with Tailwind CSS 4 and shadcn/ui components. The agent created a comprehensive task plan breaking down the work into manageable phases: asset generation, HTML structure implementation, CSS styling with theme variables, JavaScript for dynamic features, and responsive design testing.

A critical early step was the creation of a `todo.md` file to track all planned features and implementation progress. This structured approach ensured nothing was overlooked and provided clear checkpoints throughout development.

## Phase 2: Asset Generation and Theme Implementation

The agent generated placeholder assets including a professional profile picture and three project thumbnails using AI image generation. The theme was implemented through CSS custom properties, establishing the violet-to-blue gradient color scheme, Inter typography, and light grey background that would define the visual identity.

During this phase, the gradient theme evolved from a single accent color to a sophisticated violet-to-blue gradient applied across buttons, shadows, and interactive elements. This enhancement emerged through iterative feedback, demonstrating the collaborative nature of AI-assisted development.

## Phase 3: Core Feature Implementation

The main portfolio page was constructed with semantic HTML and component-based React architecture. The header incorporated social media icons with smooth hover animations transitioning from grey to brand colors. The main content area implemented the  $\frac{3}{4}$  -  $\frac{1}{3}$  layout split with the About Me section and profile picture with gradient offset shadows.

PDF viewer pages were created for certifications, CV, and individual projects, each with embedded PDF display and download functionality. The projects grid was implemented with dynamic file detection, automatically displaying thumbnails based on file existence in the public directory.

## Phase 4: Advanced Features

### Bilingual Support

A language toggle system was implemented allowing visitors to switch between English and French content in the About Me section. The implementation used React state management with smooth transitions, storing content in a structured object for easy maintenance and expansion.

### vCard QR Code Integration

A QR code placeholder was added below the About Me section with descriptive text, sized at half the profile picture dimensions. This feature facilitates quick contact information sharing for networking purposes.

### Access Gate Security

The most sophisticated feature was a URL-based access control system. The implementation uses a secret key parameter (`?key=portfolio2025`) to grant access, with `localStorage` persistence for authenticated users. Unauthorized visitors encounter a professional “Access Denied” page, while legitimate contacts with the correct key gain seamless access with the parameter automatically removed from the URL.

This security approach provides protection against casual web crawlers and unauthorized access while maintaining a frictionless experience for intended recipients via QR codes and direct links.

## Phase 5: Documentation and Deployment Preparation

Comprehensive documentation was created to support ongoing maintenance and customization. The `CUSTOMIZATION_GUIDE.md` provides step-by-step instructions for personalizing content, replacing assets, and modifying the theme. The `ACCESS_GATE_GUIDE.md` details security setup, QR code generation, and best practices for key management. The `DEPLOYMENT_CHECKLIST.md` offers a complete pre-deployment checklist and guides for five different hosting platforms.

---

# Technical Implementation

---

## Architecture Decisions

The project utilized a modern static site architecture built on React 19 with Vite as the build tool. This choice provided fast development iteration, excellent performance, and straightforward deployment to static hosting platforms. Tailwind CSS 4 enabled rapid styling with utility classes while maintaining consistency through design tokens. The shadcn/ui component library provided accessible, customizable UI primitives that integrated seamlessly with the custom design system.

## Code Quality and Maintainability

Throughout development, the AI agent maintained high code quality standards with comprehensive inline comments explaining customization points, clear section delineation, and consistent naming conventions. CSS custom properties centralized theme management, making color and typography adjustments straightforward. The component-based architecture promoted reusability and made future enhancements simple to implement.

## Performance Considerations

The final build produces an optimized static site with minimal JavaScript, lazy-loaded images, and efficient CSS. The responsive design uses mobile-first breakpoints ensuring excellent performance across devices. Asset optimization recommendations were provided for production deployment.

---

## Key Learnings and Insights

### The Power of Specific Requirements

The most significant factor in rapid development success was the specificity of the initial prompt. Vague requirements lead to misaligned implementations and extensive revision cycles. Detailed specifications with exact measurements, color values,

behavioral descriptions, and file naming conventions enabled the AI to make appropriate decisions autonomously.

## **Iterative Refinement Works**

While the initial prompt was comprehensive, several enhancements emerged through natural conversation. The gradient theme evolution, offset shadow additions, QR code integration, and access gate implementation all resulted from iterative feedback. This demonstrates that AI-assisted development benefits from both upfront planning and flexible iteration.

## **Manual Customization Remains Essential**

AI tools excel at generating boilerplate code, implementing standard patterns, and creating initial structures. However, personalizing content, selecting specific images, writing authentic copy, and making nuanced design decisions still require human judgment. The portfolio framework was generated rapidly, but meaningful deployment requires manual content preparation including CV editing, photo selection, certification compilation, and professional copy writing.

## **Documentation Multiplies Value**

The AI agent's ability to generate comprehensive documentation alongside code significantly enhanced project value. The customization guides, deployment checklists, and security setup instructions transform a code repository into a maintainable product that can be understood and modified months or years later.

---

## **Results and Outcomes**

---

### **Development Timeline**

The entire development process from initial prompt to deployable website required less than 45 minutes of active development time. This timeline excludes content preparation (CV editing, photo selection, certification compilation) which remains user-dependent. Traditional development of equivalent functionality would typically require 8-16 hours for an experienced developer.

## **Feature Completeness**

The final portfolio includes all initially specified features plus several enhancements that emerged during development. The site is production-ready with responsive design, bilingual support, security features, PDF viewing, dynamic content detection, and comprehensive documentation.

## **Code Quality**

The generated codebase maintains professional standards with clear structure, comprehensive comments, consistent styling, and modern best practices. The code is immediately understandable to developers and readily customizable for future enhancements.

---

## **Broader Implications for AI Tool Testing**

### **Workflow Acceleration**

This project demonstrates that AI agentic tools can compress development timelines by an order of magnitude for well-defined projects. The key enabler is not just the AI's coding ability, but its capacity to handle multiple concerns simultaneously—architecture, styling, functionality, documentation, and deployment preparation.

### **Skill Amplification vs. Replacement**

The experience reinforces that AI tools amplify rather than replace developer skills. Understanding web technologies, design principles, user experience, and deployment processes remains essential for crafting effective prompts, evaluating AI outputs, and making refinement decisions. The AI accelerates execution of a clear vision but does not replace the vision itself.

### **Quality Consistency**

AI-generated code maintains consistent quality throughout a project in ways that can be challenging for human developers during extended sessions. Variable naming,

comment density, code formatting, and documentation completeness remain uniform because the AI applies the same standards systematically.

## The Prompt Engineering Skill

As AI tools become more prevalent in development workflows, prompt engineering emerges as a critical skill. The ability to articulate requirements precisely, provide appropriate context, and guide AI agents through complex implementations becomes as valuable as traditional coding skills.

---

## Conclusion

This portfolio development project successfully demonstrates the potential of AI agentic tools to dramatically accelerate web development workflows. By combining careful prompt engineering, iterative collaboration, and clear vision, a professional-grade portfolio website was created in under 45 minutes of active development time.

The experience highlights that successful AI-assisted development requires a partnership between human vision and AI execution. The human provides strategic direction, aesthetic judgment, and contextual understanding, while the AI handles implementation details, maintains consistency, and generates comprehensive documentation.

As AI tools continue to evolve, understanding their capabilities and limitations becomes increasingly important for professionals across all fields. This project serves as a practical exploration of current AI capabilities, demonstrating both the remarkable acceleration possible and the continued importance of human expertise in guiding the development process.

The resulting portfolio stands as both a professional showcase and a testament to the power of thoughtfully applied AI tools in modern workflows.

---

# Technical Specifications

| Aspect                 | Details  |
|------------------------|--|
| <b>Framework</b>       | React 19 with Vite   |
| <b>Styling</b>         | Tailwind CSS 4, CSS Custom Properties                              |
| <b>Components</b>      | shadcn/ui library  |
| <b>Routing</b>         | Wouter (lightweight React router)                                  |
| <b>Typography</b>      | Inter (Google Fonts)   |
| <b>Theme</b>           | Light violet-to-blue gradient (#8B5CF6 to #3B82F6)                 |
| <b>Responsive</b>      | Mobile-first breakpoints (sm, md, lg, xl)                          |
| <b>Security</b>        | URL parameter-based access gate with localStorage                  |
| <b>Deployment</b>      | Static site (compatible with Netlify, Vercel, GitHub Pages, Manus) |
| <b>Build Size</b>      | ~1.6MB (source), optimized for production                          |
| <b>Browser Support</b> | Modern browsers (Chrome, Firefox, Safari, Edge)                    |

# Project Timeline

| Phase                  | Duration        | Activities                                |
|------------------------|-----------------|---|
| Prompt Refinement      | Pre-development | Requirements clarification with ChatGPT-5 |
| Project Initialization | 2 minutes       | Setup, planning, todo creation            |
| Asset Generation       | 3 minutes       | Placeholder images, theme setup           |
| Core Implementation    | 15 minutes      | Layout, components, styling               |
| Feature Enhancement    | 12 minutes      | Gradients, animations, shadows            |
| Bilingual Support      | 5 minutes       | Language toggle implementation            |
| Security Features      | 8 minutes       | Access gate, documentation                |
| Documentation          | 5 minutes       | Guides, checklists, README                |
| Total Active Time      | ~45 minutes     | Excludes content preparation              |

## Acknowledgments

This project was developed in collaboration with **Manus AI**, an autonomous general AI agent capable of complex web development, documentation generation, and deployment preparation. The initial prompt was refined using **ChatGPT-5** to ensure clarity and completeness before implementation.

**Project Repository:** Available upon request

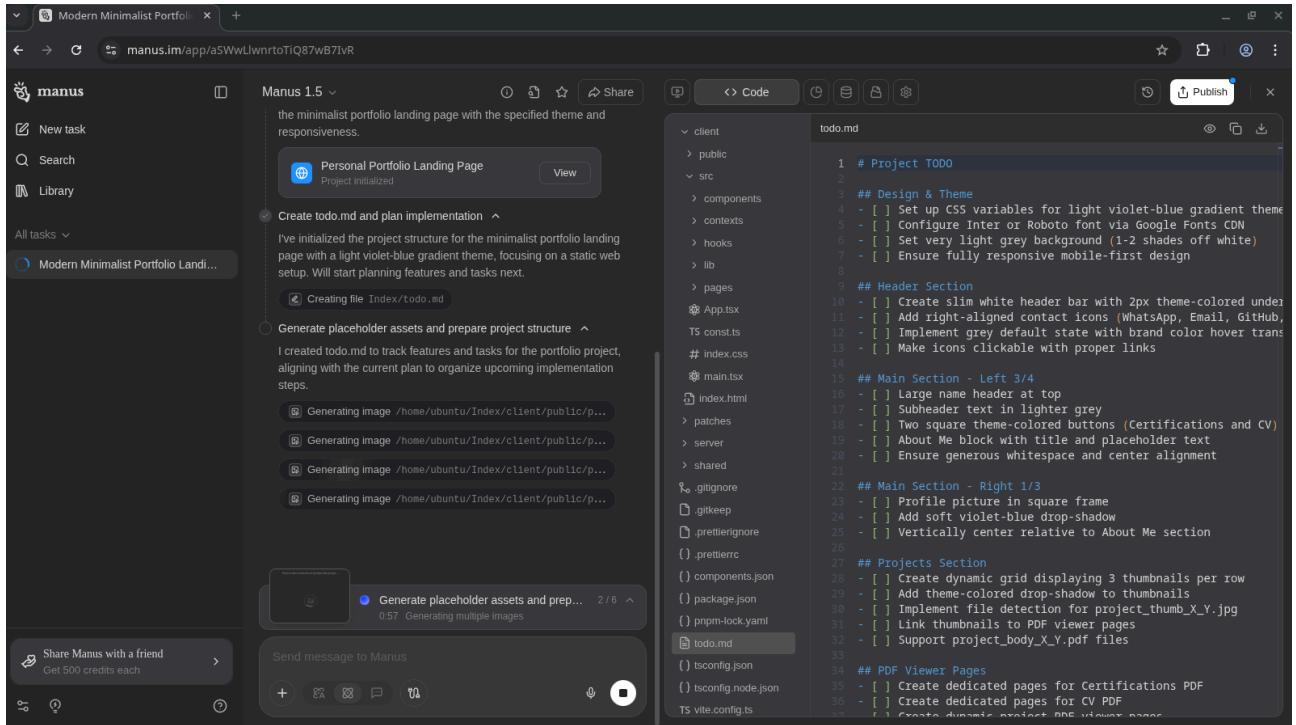
**Live Demo:** Protected by access gate (contact for access key)

**Documentation:** Comprehensive guides included in project package

*This case study was prepared as part of ongoing research into practical applications of AI tools in professional workflows. The portfolio itself serves as both a professional showcase and a demonstration of AI-assisted development capabilities.*

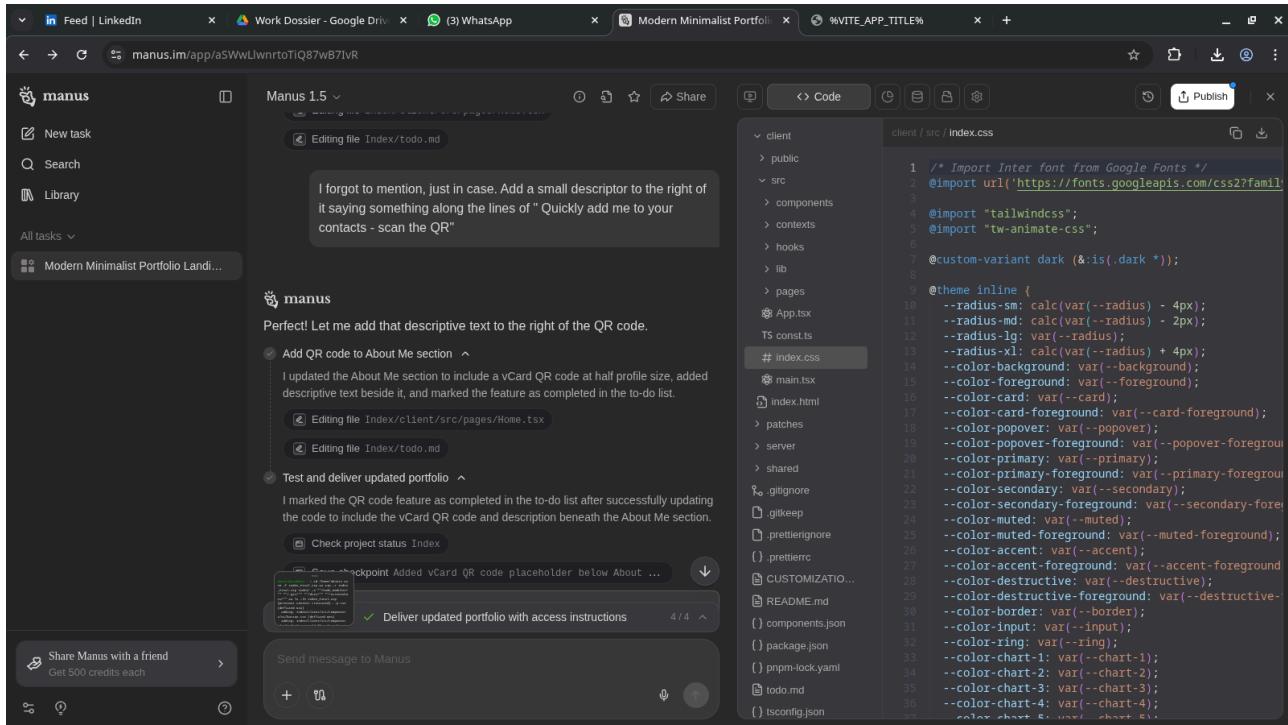
# Development Screenshots

Figure 1: Initial Project Setup and Planning



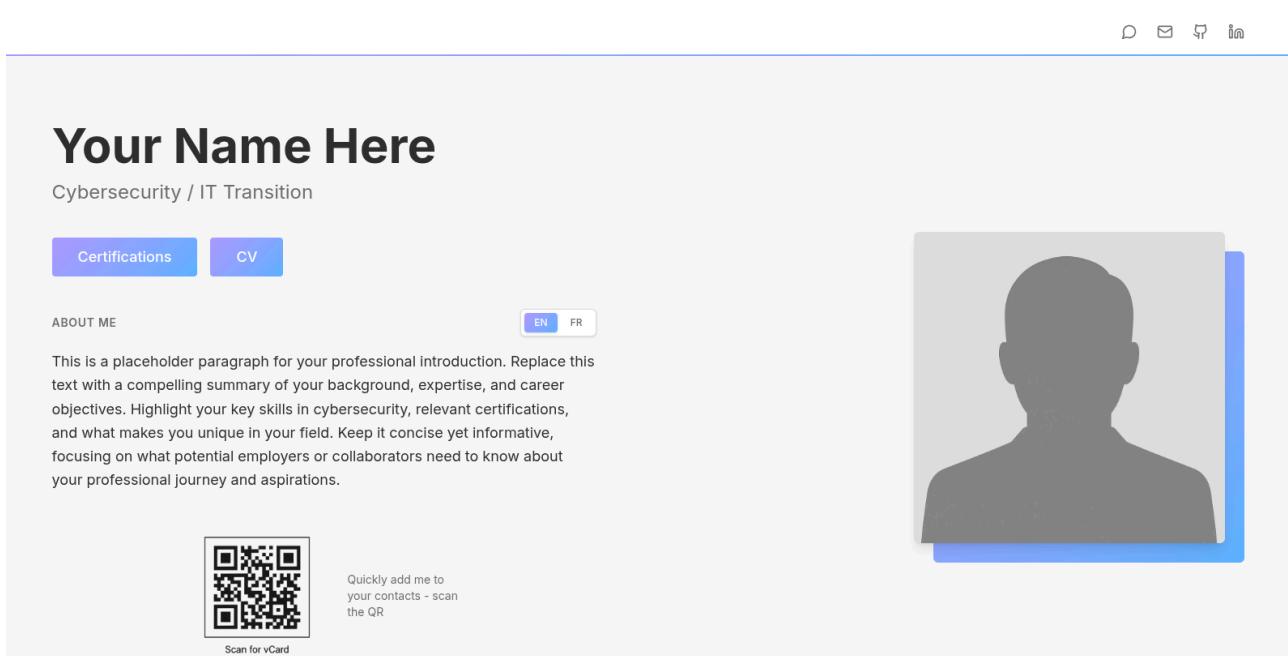
*The development process began with Manus AI creating a structured task plan and generating placeholder assets. The todo.md file tracked all features throughout development, ensuring comprehensive implementation.*

## Figure 2: Iterative Feature Development



*The iterative development process included adding features like the vCard QR code based on conversational feedback. The agent maintained clean code with comprehensive comments throughout.*

## Figure 3: Final Portfolio Result



*The finished portfolio showcases the modern minimalist design with violet-to-blue gradient theme, EN/FR language toggle, profile picture with offset shadow, and vCard*

*QR code integration.*

---