Meta-Prompting Design Migration: All Prompts Used

Meta-Prompting in Action: What Happens When You Let Claude Code Redesign Your App

A brutally honest experiment in trusting AI without human oversight

I asked Claude to write 13 detailed prompts for Claude Code, then executed them without any testing or modifications. Here's what worked brilliantly—and what broke completely.

Full case study available here

About This Document

This document contains all 14 prompts used in the design migration experiment where I transformed my Prompt Engineering Toolkit from a Neobrutalism design to Glassmorphism using meta-prompting and Claude Code.

Key Details:

- **Project:** Prompt Engineering Toolkit
- **Migration Type:** Neobrutalism → Glassmorphism
- **Approach:** Meta-prompting with zero manual intervention
- **Tool Used:** Claude Code (Terminal AI Assistant)
- **Total Prompts:** 14 (13 planned + 1 added mid-migration)
- **Execution Time:** ~1-2 hours across multiple sessions

Author: Prisca Onyebuchi
Portfolio: https://priscaonyebuchi.com
Date: October 23, 2025

How to Use These Prompts

For Your Own Projects:

1. **Adapt, Don't Copy:** These prompts are specific to my project structure. Modify file paths, component names, and design requirements to match your needs.
2. **Execute Sequentially:** These prompts build on each other. Complete and test each one before moving to the next.
3. **Test Between Prompts:** Unlike my experiment, I recommend testing after each prompt to catch issues early.
4. **Commit Frequently:** Save your progress with git commits after each successful prompt execution.
5. **Review AI Output:** Always review the code changes Claude Code makes before accepting them.

Tools You'll Need:

- **Claude Code:** Terminal-based AI coding assistant
- **Next.js Project:** Or adapt for your framework
- **Git:** For version control and rollback capability

The 14 Prompts

Phase 1: Design System Foundation

PROMPT 1: Extract and Analyze Portfolio Design System

```
I'm migrating my Prompt Engineering Toolkit from Neobrutalism to Glassmorphism to
match my main portfolio. My main portfolio is located at ../portfolio-website
relative to this project.

I need you to analyze the design system from my main portfolio. Here are the key
files:

1. The globals.css file (contains CSS variables, glassmorphism utilities,
animations)
2. The HeroSection.tsx component (shows floating orb implementation)
3. The tailwind.config file (Tailwind theme configuration)

Please:
4. Identify all CSS custom properties (--color-primary, --color-secondary, etc.)
5. Extract the glassmorphism card styles (.glass, .card classes)
6. Document the animation patterns (floating orbs, fade-in, slide-up)
7. List the gradient utilities (.text-gradient, .bg-gradient-primary)
8. Note the transition timings and easing functions

Create a comprehensive design tokens document that I can reference. Do NOT make
any code changes yet—just analyze and document.
```

What to Expect:
A markdown document with all design tokens, color values, animation specs, and utility classes.

PROMPT 2: Update Global CSS Variables

```
Now let's update the design system. Based on the analysis you just did, update
src/app/globals.css to replace the current Neobrutalism design tokens with the
Glassmorphism system from my portfolio.

Specifically:
1. Replace ALL color variables with the portfolio's color system (primary,
secondary, accent, etc.)
2. Remove all "brutal-*" class definitions
```

```
3. Add the glassmorphism utilities:
   - .glass (backdrop-filter with blur)
   - .card (with soft shadows and hover effects)
   - .text-gradient (gradient text effect)
   - .bg-gradient-primary (gradient backgrounds)
   - .bg-gradient-card (gradient card backgrounds)
4. Add animation keyframes: fadeIn, slideUp, bounce-gentle
5. Keep the existing dark mode support structure but update colors
6. Add smooth transition utilities
7. Update scrollbar styling to use new primary colors

IMPORTANT: Maintain the existing :root structure and dark mode support. Just
update the values and add new utilities.

Show me the updated globals.css file when done.
```

What to Expect:
Complete updated globals.css with new design system.

PROMPT 3: Update Tailwind Configuration

```
Update tailwind.config.ts to align with the new glassmorphism design system.

Changes needed:
1. Update the colors object to include:
   - primary with light/dark variants
   - secondary with variants
   - accent
   - Background colors that work with glass effect
   - Proper dark mode colors

2. Add custom animations:
   - bounce-gentle (6s duration)
   - fade-in (0.5s ease-in-out)
   - slide-up (0.5s ease-out)

3. Add custom box shadows for cards:
   - shadow-soft: subtle shadow for cards
   - shadow-medium: hover state
   - shadow-strong: focused/active state

4. Extend spacing if needed for micro-interactions

5. Add backdrop-blur utilities if not present

Reference the color values from the globals.css we just updated. Ensure everything
is consistent.

Show me the complete tailwind.config.ts when done.
```

What to Expect:
Updated Tailwind config matching new design system.

Phase 2: Component Updates

PROMPT 4: Update UI Component Library

```
Now let's update the base UI components in src/components/ui/ to use the new
glassmorphism design system.

For each component (Button, Card, Badge, Input, Textarea, etc.):

1. CARD COMPONENT:
   - Remove brutal borders (border-3, border-black)
   - Add glassmorphism effect: backdrop-filter, soft borders, subtle shadows
   - Add smooth hover transition (translateY, shadow increase)
   - Ensure it works in both light and dark modes

2. BUTTON COMPONENT:
   - Replace solid borders with subtle ones
   - Add gradient backgrounds for primary variant
   - Smooth hover effects (scale, shadow)
   - Add active state micro-animation
   - Maintain all existing variants (primary, secondary, outline) but with new
styling

3. BADGE COMPONENT:
   - Soften borders
   - Use gradient backgrounds where appropriate
   - Add subtle hover effects
   - Keep color variants but update to new palette

4. INPUT/TEXTAREA:
   - Remove heavy borders
   - Add focus glow effect (ring with primary color)
   - Smooth transitions on focus
   - Glassmorphism background in dark mode

IMPORTANT RULES:
- Keep all existing props and APIs—only change styling
- Maintain TypeScript types
- Ensure accessibility (focus states, contrast)
- Add smooth transitions (0.2s ease-in-out minimum)
- Test with both light and dark modes in mind

Update each component file and show me the changes.
```

What to Expect:
Updated UI components with new design system applied.

PROMPT 5: Update Layout Components

```
Update the layout components (MainLayout, Header, Footer, Navigation) to use
glassmorphism and add micro-interactions.

MAINLAYOUT (src/components/layout/MainLayout.tsx):
- Ensure background supports glassmorphism (subtle gradient if needed)
- Add smooth page transitions

HEADER (src/components/layout/Header.tsx):
- Make it glassmorphic (backdrop-filter blur on scroll)
- Add smooth scroll behavior (changes opacity/blur when scrolling)
- Navigation links should have smooth hover underline animation
- Mobile menu should slide in smoothly with backdrop

NAVIGATION:
- Hover effects on links (underline that grows from center)
- Active route indicator with smooth transition
- Subtle scale effect on hover (scale-105)

FOOTER:
- Glassmorphic card effect
- Links with smooth hover states
- Social icons with pop effect on hover

Add micro-interactions:
- Links: underline grows from 0 to full width on hover (0.3s ease)
- Buttons: slight scale (1.02) and shadow increase on hover
- Icons: rotate or bounce slightly on hover
- Smooth color transitions on all interactive elements

Show me each updated component.
```

What to Expect:
Layout components with glassmorphism and smooth interactions.

Phase 3: Page-Specific Updates

PROMPT 6: Update Homepage with Floating Orbs

```
Update src/app/page.tsx (homepage) to match my portfolio's hero section aesthetic
with floating animated orbs and glassmorphism.

Specific requirements:

1. HERO SECTION:
   - Add gradient background: from-primary/5 via-background to-secondary/5
   - Add TWO floating orbs with blur effect:
     * One top-right: large (w-72 h-72), primary color, animated (bounce-gentle,
```

```
6s)
     * One bottom-left: large, secondary color, animated (bounce-gentle, 6s delay)
   - Orbs should have: mix-blend-multiply (dark mode: mix-blend-lighten), filter
blur-xl, opacity-20-30

2. CONTENT OVERLAY:
   - Should have position relative z-10 to sit above orbs
   - Glassmorphic cards for feature showcases
   - Smooth fade-in animation on mount
   - Text gradient for main heading

3. STATS SECTION:
   - Cards should have glass effect
   - Smooth hover animation (lift + glow)
   - Numbers should count up on scroll into view
   - Icons with subtle rotation on hover

4. CTA SECTION:
   - Gradient button with smooth hover effect
   - Pulsing animation to draw attention
   - Glass card background

Ensure everything is responsive and works in dark mode.

Show me the complete updated page.tsx.
```

What to Expect:
Homepage with floating orbs, glassmorphism, and smooth animations.

PROMPT 7: Update Prompts Library Page

```
Update the Prompts Library page (src/app/prompts/page.tsx or similar) to use
glassmorphism design and add smooth interactions.

REQUIREMENTS:

1. FILTER SECTION:
   - Glass cards for filter buttons
   - Smooth active state transition
   - Hover effects on filter chips
   - Count badges with gradient

2. PROMPT CARDS:
   - Glassmorphism card design
   - Smooth hover lift effect
   - Category badges with glass effect
   - Difficulty indicators with color gradient
   - Smooth transition when clicking to view details

3. SEARCH BAR:
```

```
   - Glass effect background
   - Focus state with glow
   - Search icon animation when typing
   - Clear button appears smoothly

4. GRID LAYOUT:
   - Staggered fade-in animation on load
   - Smooth reordering when filtering
   - Responsive grid with proper spacing

5. EMPTY STATES:
   - Glass card with helpful message
   - Animated icon
   - Smooth fade-in

Show me the updated prompts page with all improvements.
```

What to Expect:
Prompts library page with glassmorphism cards and smooth filter interactions.

PROMPT 8: Update Prompt Detail Pages

```
Update individual prompt detail pages to showcase each prompt beautifully with
glassmorphism and interactive elements.

PROMPT DETAIL PAGE REQUIREMENTS:

1. HEADER SECTION:
   - Gradient hero background with subtle orb
   - Glassmorphic breadcrumb navigation
   - Prompt title with text gradient
   - Metadata (category, difficulty, date) in glass chips

2. PROMPT DISPLAY:
   - Code block with glass background
   - Copy button with smooth hover effect
   - Syntax highlighting that works in dark mode
   - Line numbers if appropriate

3. DESCRIPTION SECTION:
   - Glass card with good typography
   - Smooth scroll animations
   - Related tags as interactive chips

4. EXAMPLE OUTPUT:
   - Before/After comparison with glass cards
   - Smooth transition between views
   - Screenshots with proper shadows

5. RELATED PROMPTS:
```

```
   - Horizontal scroll of glass cards
   - Smooth hover effects
   - Staggered animation on load

6. ACTION BUTTONS:
   - "Try this prompt" with gradient
   - "Share" with social icons
   - "Bookmark" with animation feedback

Make it feel premium and polished.

Show me the complete prompt detail page.
```

What to Expect:
Prompt detail page with beautiful glassmorphism, code display, and interactive elements.

PROMPT 9: Update About/Documentation Page

```
Update the About/Documentation page to explain the toolkit with glassmorphism
design and engaging micro-interactions.

ABOUT PAGE REQUIREMENTS:

1. HERO SECTION:
   - Gradient background with floating orb
   - Glass card with mission statement
   - Animated stats counter
   - Text gradient for heading

2. FEATURES GRID:
   - Glass cards for each feature
   - Icons with hover animation (rotate/scale)
   - Smooth staggered fade-in
   - Hover lift effect

3. HOW IT WORKS:
   - Step-by-step with glass cards
   - Connecting lines or arrows (animated)
   - Number badges with gradients
   - Smooth scroll-triggered animations

4. TESTIMONIALS/QUOTES:
   - Glass cards with good typography
   - Subtle background pattern
   - Author avatars with hover effect
   - Quote icon with gradient

5. CTA SECTION:
   - Prominent glass card
   - Gradient buttons
```

```
   - Smooth hover states
   - Background orbs for visual interest

Make it informative yet visually engaging.

Show me the complete about/documentation page.
```

What to Expect:
About page with glassmorphism design, feature showcases, and smooth animations.

Phase 4: Micro-interactions & Polish

PROMPT 10: Add Comprehensive Micro-interactions

```
Add comprehensive micro-interactions across all interactive elements in the
application.

MICRO-INTERACTIONS TO ADD:

1. BUTTONS:
   - Scale to 1.02 on hover (0.2s ease)
   - Slight shadow increase
   - Active state: scale to 0.98
   - Loading state: pulsing animation
   - Success state: checkmark animation

2. LINKS:
   - Underline grows from center (0.3s ease)
   - Subtle color transition
   - Hover: slight letter-spacing increase

3. CARDS:
   - Lift on hover (translateY: -4px, 0.3s ease)
   - Shadow intensifies
   - Border glow effect appears
   - Smooth transition out

4. FORM INPUTS:
   - Focus: glow ring (primary color)
   - Label floats up when focused
   - Success: green glow
   - Error: red shake animation

5. ICONS:
   - Rotate 15deg on hover (0.3s ease)
   - Scale 1.1
   - Some icons: bounce on hover

6. BADGES/CHIPS:
   - Scale 1.05 on hover
```

```
   - Slight shadow increase
   - Smooth color transition

7. MODALS/TOASTS:
   - Slide in with backdrop fade
   - Close with smooth fade-out
   - Backdrop blur increases

8. LOADING STATES:
   - Skeleton screens with shimmer
   - Spinners with smooth rotation
   - Progress bars with gradient

Apply these consistently across the entire application.

Show me the key components with these micro-interactions added.
```

What to Expect:
Comprehensive micro-interactions added to all interactive elements for a premium feel.

PROMPT 11: Implement Page Transitions and Scroll Animations

```
Implement smooth page transitions and scroll animations throughout the
application.

PAGE TRANSITIONS:

1. ROUTE CHANGES:
   - Fade out current page (0.2s)
   - Fade in new page (0.3s with slight delay)
   - Smooth content shift
   - Loading indicator during transition

2. SCROLL ANIMATIONS:
   - Fade in elements as they enter viewport
   - Staggered animations for lists/grids
   - Parallax effect on hero sections (subtle)
   - Progress bar showing scroll depth

3. SCROLL BEHAVIOR:
   - Smooth scrolling (scroll-behavior: smooth)
   - Back-to-top button with smooth scroll
   - Anchor links smooth scroll
   - Header changes on scroll (blur, shadow)

4. INTERSECTION OBSERVER:
   - Trigger animations when elements are 20% visible
   - Fade in + slide up pattern
   - Count-up animations for numbers
   - Stagger delays for groups
```

```
5. SKELETON LOADING:
   - Show skeleton screens while content loads
   - Shimmer animation
   - Smooth transition to real content

Make navigation feel seamless and content feel alive.

Implement these animations and show me the key components.
```

What to Expect:
Smooth page transitions, scroll animations, and loading states throughout the application.

Phase 5: Testing & Optimization (1 hour)

PROMPT 12: Performance Optimization

```
Optimize the application for performance while maintaining all the smooth
animations and glassmorphism effects.

OPTIMIZATION TASKS:

1. ANIMATION PERFORMANCE:
   - Ensure animations use transform and opacity (GPU accelerated)
   - Add will-change hints where appropriate
   - Remove any layout-thrashing animations

2. GLASS EFFECT OPTIMIZATION:
   - Use backdrop-filter efficiently (not on every element)
   - Consider fallbacks for browsers without backdrop-filter support
   - Optimize blur amounts (don't go overboard)

3. IMAGE OPTIMIZATION:
   - Ensure all images use Next/Image
   - Add appropriate loading states
   - Implement blur placeholder where beneficial

4. CODE SPLITTING:
   - Lazy load heavy components (Monaco editor, etc.)
   - Ensure pages load progressively
   - Dynamic imports where appropriate

5. BUNDLE SIZE:
   - Check if any unnecessary libraries crept in
   - Tree-shaking is working properly
   - Remove unused Tailwind classes (purge)

6. ACCESSIBILITY:
   - Add prefers-reduced-motion support
   - Ensure keyboard navigation works smoothly
```

```
   - Test with screen readers (basic check)

Run performance audits and show me areas that need improvement.
```

What to Expect:
Performance optimization recommendations and implementations.

PROMPT 13 (BONUS): Add Theme Toggle

Note: This prompt was added mid-migration when I realized the theme toggle was missing.

```
Add a theme toggle button to the top navigation to allow users to switch between
light and dark mode
```

What to Expect:
A functional theme toggle button in the header that switches between light and dark modes smoothly.

PROMPT 14: Final Polish Pass

```
Do a final polish pass across the entire application. Look for:

1. CONSISTENCY:
   - All cards use same glass effect
   - All buttons have same hover behavior
   - All links have same interaction pattern
   - Colors are consistent across pages

2. EDGE CASES:
   - Empty states have glass styling
   - Error states are styled properly
   - Loading states everywhere they're needed
   - Mobile responsive on all pages

3. MICRO-INTERACTION GAPS:
   - Any buttons without hover effects?
   - Any transitions that feel abrupt?
   - Any elements that should animate but don't?

4. TYPOGRAPHY:
   - Hierarchy is clear
   - Gradients on headings are readable
   - Line heights are comfortable
   - Font weights are consistent

5. SPACING:
   - Consistent padding/margins
```

```
    - Proper breathing room
    - Elements don't feel cramped or too spread out

6. FINAL TOUCHES:
    - Add any missing loading indicators
    - Ensure all images have alt text
    - Check all links work
    - Verify forms have proper validation

Make any final adjustments needed to achieve a premium, cohesive design.
```

What to Expect:
Final refinements and polish.

Want to Learn More?

Read the full case study here

Connect with me:

- Portfolio
- LinkedIn
- GitHub

License & Usage

Feel free to use and adapt these prompts for your own projects. If you find them helpful, I'd appreciate:

- A link back to the original case study
- A mention on social media
- Sharing what you learned from your own experiments

Attribution: Prisca Onyebuchi - Meta-Prompting Design Migration Experiment (October 2025)

This document is part of the Meta-Prompting in Action case study exploring the limits of AI-assisted development without human intervention.