Technical Specification - Atelier Agency Theme v2.1

Project Overview

Project Name: Atelier Agency Theme
Version: 2.1
Framework: Next.js 14 with TypeScript
Philosophy: Digital Stonemasonry
Target: Creative agencies, architects, design studios

Architecture

Framework & Technology Stack

```
Core Framework:
├── Next.js 14 (App Router)
├── TypeScript 5.2+
├── React 18
└── Node.js 18+

Styling:
├── SCSS/CSS Modules
├── CSS Custom Properties
├── Tailwind CSS (utility-first)
└── PostCSS

Animation & Interaction:
├── GSAP 3.12+
├── Lenis (smooth scroll)
├── Framer Motion (optional)
└── CSS Transitions

Build Tools:
├── ESLint + Prettier
├── Husky (git hooks)
└── TypeScript Compiler
```

Project Structure

```
atelier-agency-theme/
├── app/                        # Next.js App Router
│   ├── layout.tsx              # Root layout
│   ├── page.tsx                # Home page
│   ├── globals.css             # Global styles
│   ├── about/
│   │   └── page.tsx            # About page
│   ├── work/
│   │   ├── page.tsx            # Work listing
│   │   └── [slug]/
│   │       └── page.tsx        # Individual project
│   ├── contact/
│   │   └── page.tsx            # Contact page
│   └── api/                    # API routes
├── components/
│   ├── layout/                 # Layout components
│   │   ├── Header.tsx
│   │   ├── Footer.tsx
│   │   └── Navigation.tsx
│   ├── interactive/            # Interactive components
│   │   ├── CookieConsent.tsx
│   │   ├── LiveSearch.tsx
│   │   ├── ThemeToggle.tsx
│   │   ├── InteractiveCursor.tsx
│   │   ├── DynamicMarquee.tsx
│   │   ├── ImageHotspots.tsx
│   │   └── 3DModelViewer.tsx
│   ├── showcase/               # Portfolio components
│   │   ├── ProjectGrid.tsx
│   │   ├── ProjectCard.tsx
│   │   ├── TeamGrid.tsx
│   │   ├── ServiceCards.tsx
│   │   └── TestimonialDisplay.tsx
│   ├── forms/                  # Form components
│   │   ├── ContactForm.tsx
│   │   └── NewsletterForm.tsx
│   ├── animation/              # Animation components
│   │   ├── ScrollReveal.tsx
│   │   ├── ParallaxSection.tsx
│   │   └── StaggeredGrid.tsx
│   └── ui/                     # Base UI components
│       ├── Button.tsx
│       ├── Card.tsx
│       ├── Modal.tsx
│       └── ...
├── hooks/                      # Custom React hooks
│   ├── use-gsap-animations.ts
│   ├── use-smooth-scroll.ts
│   ├── use-theme-toggle.ts
│   ├── use-live-search.ts
│   └── ...
├── lib/                        # Utility functions
│   ├── utils.ts
│   ├── constants.ts
│   ├── animations.ts
│   └── content.ts
├── data/                       # Content data
│   ├── projects.ts
│   ├── team.ts
│   ├── services.ts
│   └── site-config.ts
├── styles/                     # SCSS stylesheets
```

```
│   ├── globals.scss
│   ├── components.scss
│   ├── animations.scss
│   └── utilities.scss
├── public/                      # Static assets
│   ├── images/
│   ├── icons/
│   ├── textures/
│   └── models/
└── docs/                        # Documentation
    ├── Getting-Started.md
    ├── Customization.md
    ├── Content-Guide.md
    └── SPECIFICATION.md
```

Design System

Color System

The theme implements a sophisticated color palette inspired by architectural materials:

```
:root {
  /* Primary Palette - "Quarried Stone" */
  --color-background: #191919;       /* Charcoal stone */
  --color-surface: #212121;          /* Elevated surfaces */
  --color-text: #A3A3A3;            /* Neutral text */
  --color-heading: #E5E5E5;         /* High contrast headings */
  --color-primary: #C2A878;         /* Gold vein accent */

  /* Extended Palette */
  --color-secondary: #8B9AAF;       /* Steel blue */
  --color-tertiary: #6B7280;        /* Slate gray */

  /* Semantic Colors */
  --color-success: #10B981;         /* Emerald */
  --color-warning: #F59E0B;         /* Amber */
  --color-error: #EF4444;           /* Rose */
  --color-info: #3B82F6;            /* Blue */

  /* Transparency Variants */
  --color-text-muted: rgba(163, 163, 163, 0.7);
  --color-border: rgba(255, 255, 255, 0.1);
  --color-overlay: rgba(0, 0, 0, 0.8);
}
```

Typography System

Fluid Typography Scale using CSS `clamp()` functions:

```
:root {
  /* Font Families */
  --font-family-sans: 'Inter', ui-sans-serif, system-ui, sans-serif;
  --font-family-serif: 'Lora', ui-serif, Georgia, serif;
  --font-family-mono: 'JetBrains Mono', 'Consolas', monospace;

  /* Fluid Type Scale (Utopia methodology) */
  --step--2: clamp(0.72rem, 0.70rem + 0.10vw, 0.78rem);    /* 11.5px → 12.5px */
  --step--1: clamp(0.83rem, 0.80rem + 0.18vw, 0.94rem);    /* 13.3px → 15px */
  --step-0:  clamp(1.00rem, 0.95rem + 0.25vw, 1.13rem);    /* 16px → 18px */
  --step-1:  clamp(1.20rem, 1.10rem + 0.35vw, 1.35rem);    /* 19.2px → 21.6px */
  --step-2:  clamp(1.44rem, 1.28rem + 0.48vw, 1.62rem);    /* 23px → 26px */
  --step-3:  clamp(1.73rem, 1.48rem + 0.64vw, 1.94rem);    /* 27.7px → 31px */
  --step-4:  clamp(2.07rem, 1.71rem + 0.85vw, 2.33rem);    /* 33.1px → 37.3px */
  --step-5:  clamp(2.49rem, 1.98rem + 1.12vw, 2.80rem);    /* 39.8px → 44.8px */
  --step-6:  clamp(2.99rem, 2.30rem + 1.47vw, 3.36rem);    /* 47.8px → 53.8px */
  --step-7:  clamp(3.59rem, 2.66rem + 1.92vw, 4.03rem);    /* 57.4px → 64.5px */
  --step-8:  clamp(4.31rem, 3.07rem + 2.50vw, 4.84rem);    /* 68.9px → 77.4px */

  /* Font Weights */
  --font-weight-light: 300;
  --font-weight-normal: 400;
  --font-weight-medium: 500;
  --font-weight-semibold: 600;
  --font-weight-bold: 700;

  /* Line Heights */
  --line-height-tight: 1.25;
  --line-height-normal: 1.5;
  --line-height-relaxed: 1.75;
}
```

Spacing System

Fluid Spacing Scale for consistent rhythm:

```
:root {
  /* Fluid Spacing (Mobile → Desktop) */
  --space-2xs: clamp(0.25rem, 0.20rem + 0.20vw, 0.5rem);   /* 4px → 8px */
  --space-xs:  clamp(0.5rem, 0.40rem + 0.30vw, 0.75rem);   /* 8px → 12px */
  --space-s:   clamp(0.75rem, 0.60rem + 0.40vw, 1rem);     /* 12px → 16px */
  --space-m:   clamp(1rem, 0.80rem + 0.60vw, 1.5rem);      /* 16px → 24px */
  --space-l:   clamp(1.5rem, 1.20rem + 0.80vw, 2.25rem);   /* 24px → 36px */
  --space-xl:  clamp(2rem, 1.60rem + 1.00vw, 3rem);        /* 32px → 48px */
  --space-2xl: clamp(3rem, 2.40rem + 1.50vw, 4.5rem);      /* 48px → 72px */
  --space-3xl: clamp(4rem, 3.20rem + 2.50vw, 6rem);        /* 64px → 96px */
  --space-4xl: clamp(6rem, 4.80rem + 3.00vw, 9rem);        /* 96px → 144px */
  --space-5xl: clamp(8rem, 6.40rem + 4.00vw, 12rem);       /* 128px → 192px */
}
```

Design Tokens

```
:root {
  /* Border Radius */
  --radius-sm: 0.375rem;      /* 6px */
  --radius: 0.5rem;           /* 8px */
  --radius-md: 0.75rem;       /* 12px */
  --radius-lg: 1rem;          /* 16px */
  --radius-xl: 1.5rem;        /* 24px */

  /* Shadows */
  --shadow-sm: 0 1px 2px rgba(0, 0, 0, 0.05);
  --shadow: 0 1px 3px rgba(0, 0, 0, 0.1), 0 1px 2px rgba(0, 0, 0, 0.06);
  --shadow-md: 0 4px 6px rgba(0, 0, 0, 0.07), 0 2px 4px rgba(0, 0, 0, 0.06);
  --shadow-lg: 0 10px 15px rgba(0, 0, 0, 0.1), 0 4px 6px rgba(0, 0, 0, 0.05);
  --shadow-xl: 0 20px 25px rgba(0, 0, 0, 0.1), 0 10px 10px rgba(0, 0, 0, 0.04);
  --shadow-2xl: 0 25px 50px rgba(0, 0, 0, 0.25);

  /* Transitions */
  --transition-fast: 0.15s ease-out;
  --transition-normal: 0.3s ease-out;
  --transition-slow: 0.5s ease-out;

  /* Z-Index Scale */
  --z-dropdown: 1000;
  --z-sticky: 1020;
  --z-fixed: 1030;
  --z-modal-backdrop: 1040;
  --z-modal: 1050;
  --z-popover: 1060;
  --z-tooltip: 1070;
  --z-toast: 1080;
}
```

Component Architecture

Core Component Patterns

1. Layout Components

Header Component

```
interface HeaderProps {
  transparent?: boolean;
  fixed?: boolean;
  className?: string;
}

export function Header({ transparent = false, fixed = true, className }: HeaderProps) {
  return (
    <header
      className={`header ${transparent ? 'header--transparent' : ''} ${fixed ? 'header--fixed' : ''} ${className || ''}`}
    >
      <div className="header__container">
        <Logo />
        <Navigation />
        <div className="header__actions">
          <ThemeToggle />
          <LiveSearch />
          <MenuTrigger />
        </div>
      </div>
    </header>
  );
}
```

Navigation Component

```
interface NavigationItem {
  label: string;
  href: string;
  children?: NavigationItem[];
}

export function Navigation() {
  const [isOpen, setIsOpen] = useState(false);

  return (
    <nav className={`navigation ${isOpen ? 'navigation--open' : ''}`}>
      <div className="navigation__backdrop" onClick={() => setIsOpen(false)} />
      <div className="navigation__content">
        <NavigationItems />
        <NavigationFooter />
      </div>
    </nav>
  );
}
```

2. Interactive Components

CookieConsent Component

```
export function CookieConsent() {
  const { hasConsented, acceptCookies } = useCookieConsent();

  if (hasConsented) return null;

  return (
    <div className="cookie-consent">
      <div className="cookie-consent__content">
        <p>We use cookies to enhance your experience.</p>
        <div className="cookie-consent__actions">
          <Button onClick={acceptCookies} variant="primary">
            Accept All
          </Button>
        </div>
      </div>
    </div>
  );
}
```

LiveSearch Component

```
export function LiveSearch() {
  const [query, setQuery] = useState('');
  const [isOpen, setIsOpen] = useState(false);
  const { results, loading } = useSearch(query);

  return (
    <div className={`live-search ${isOpen ? 'live-search--open' : ''}`}>
      <SearchTrigger onClick={() => setIsOpen(true)} />
      <SearchModal isOpen={isOpen} onClose={() => setIsOpen(false)}>
        <SearchInput value={query} onChange={setQuery} />
        <SearchResults results={results} loading={loading} />
      </SearchModal>
    </div>
  );
}
```

3. Showcase Components

ProjectGrid Component

```
interface ProjectGridProps {
  projects: Project[];
  columns?: 2 | 3 | 4;
  showFilter?: boolean;
}

export function ProjectGrid({ projects, columns = 3, showFilter = true }: ProjectGridProps) {
  const [filteredProjects, setFilteredProjects] = useState(projects);

  return (
    <section className="project-grid">
      {showFilter && <ProjectFilter onFilter={setFilteredProjects} />}
      <div className={`project-grid__items project-grid__items--columns-${columns}`}>
        {filteredProjects.map((project) => (
          <ProjectCard key={project.id} project={project} />
        ))}
      </div>
    </section>
  );
}
```

Animation System

GSAP Integration

Core Animation Utilities

```
// lib/animations.ts
import { gsap } from 'gsap';
import { ScrollTrigger } from 'gsap/ScrollTrigger';

gsap.registerPlugin(ScrollTrigger);

export const animations = {
  // Scroll reveal animation
  scrollReveal: (
    selector: string,
    options: {
      duration?: number;
      stagger?: number;
      blur?: boolean;
      scale?: boolean;
    } = {}
  ) => {
    const { duration = 0.8, stagger = 0.1, blur = true, scale = false } = options;

    const fromVars: any = {
      opacity: 0,
      y: 50,
    };

    const toVars: any = {
      opacity: 1,
      y: 0,
      duration,
      ease: 'power2.out',
      stagger,
    };

    if (blur) {
      fromVars.filter = 'blur(5px)';
      toVars.filter = 'blur(0px)';
    }

    if (scale) {
      fromVars.scale = 0.95;
      toVars.scale = 1;
    }

    gsap.fromTo(selector, fromVars, toVars);
  },

  // Page transition
  pageTransition: {
    enter: (element: HTMLElement) => {
      return gsap.fromTo(element,
        {
          opacity: 0,
          y: 20,
          scale: 0.98
        },
        {
          opacity: 1,
          y: 0,
          scale: 1,
          duration: 0.6,
          ease: 'power2.out'
        }
      );
    },
    exit: (element: HTMLElement) => {
      return gsap.to(element, {
        opacity: 0,
        y: -20,
        scale: 0.98,
        duration: 0.4,
        ease: 'power2.in'
      });
    }
  },

  // Hover animations
  hover: {
    lift: (selector: string) => {
      gsap.set(selector, { transformOrigin: 'center bottom' });

      return {
        enter: () => gsap.to(selector, {
          y: -8,
          scale: 1.02,
          duration: 0.3,
          ease: 'power2.out'
        }),
        leave: () => gsap.to(selector, {
          y: 0,
          scale: 1,
          duration: 0.3,
          ease: 'power2.out'
        })
      };
    }
  }
};
```

Custom Animation Hooks

```ts
// hooks/use-gsap-animations.ts
import { useEffect, useRef } from 'react';
import { animations } from '@/lib/animations';

export function useScrollReveal(options = {}) {
  const ref = useRef<HTMLElement>(null);

  useEffect(() => {
    if (ref.current) {
      const elements = ref.current.querySelectorAll('.scroll-reveal');
      animations.scrollReveal('.scroll-reveal', options);
    }
  }, [options]);

  return ref;
}

export function useHoverLift() {
  const ref = useRef<HTMLElement>(null);

  useEffect(() => {
    if (!ref.current) return;

    const hoverAnimation = animations.hover.lift(ref.current);

    const handleMouseEnter = () => hoverAnimation.enter();
    const handleMouseLeave = () => hoverAnimation.leave();

    ref.current.addEventListener('mouseenter', handleMouseEnter);
    ref.current.addEventListener('mouseleave', handleMouseLeave);

    return () => {
      ref.current?.removeEventListener('mouseenter', handleMouseEnter);
      ref.current?.removeEventListener('mouseleave', handleMouseLeave);
    };
  }, []);

  return ref;
}
```

Smooth Scrolling with Lenis

```
// hooks/use-smooth-scroll.ts
import { useEffect } from 'react';
import Lenis from '@studio-freight/lenis';

export function useSmoothScroll(options = {}) {
  useEffect(() => {
    const lenis = new Lenis({
      duration: 1.2,
      easing: (t: number) => Math.min(1, 1.001 - Math.pow(2, -10 * t)),
      direction: 'vertical',
      gestureDirection: 'vertical',
      smooth: true,
      smoothTouch: false,
      touchMultiplier: 2,
      ...options
    });

    function raf(time: number) {
      lenis.raf(time);
      requestAnimationFrame(raf);
    }

    requestAnimationFrame(raf);

    return () => {
      lenis.destroy();
    };
  }, [options]);
}
```

Performance Specifications

Core Web Vitals Targets

- **LCP (Largest Contentful Paint)**: < 2.5s
- **FID (First Input Delay)**: < 100ms
- **CLS (Cumulative Layout Shift)**: < 0.1
- **FCP (First Contentful Paint)**: < 1.8s
- **TTI (Time to Interactive)**: < 3.5s

Optimization Strategies

Image Optimization

```tsx
// components/ui/OptimizedImage.tsx
import Image from 'next/image';

interface OptimizedImageProps {
  src: string;
  alt: string;
  width?: number;
  height?: number;
  priority?: boolean;
  className?: string;
}

export function OptimizedImage({
  src,
  alt,
  width,
  height,
  priority = false,
  className
}: OptimizedImageProps) {
  return (
    <Image
      src={src}
      alt={alt}
      width={width}
      height={height}
      priority={priority}
      className={className}
      sizes="(max-width: 768px) 100vw, (max-width: 1200px) 50vw, 33vw"
      style={{ objectFit: 'cover' }}
    />
  );
}
```

Code Splitting

```tsx
// Dynamic imports for heavy components
const DynamicMarquee = dynamic(() => import('@/components/interactive/DynamicMarquee'), {
  ssr: false,
  loading: () => <MarqueeSkeleton />
});

const ThreeDViewer = dynamic(() => import('@/components/interactive/3DModelViewer'), {
  ssr: false,
  loading: () => <ViewerSkeleton />
});
```

SEO & Accessibility

SEO Configuration

```
// app/layout.tsx
export const metadata: Metadata = {
  title: {
    template: '%s | Atelier Agency',
    default: 'Atelier Agency - Digital Stonemasonry'
  },
  description: 'Premium web design and development with architectural precision.',
  keywords: ['web design', 'digital agency', 'branding', 'web development'],
  authors: [{ name: 'Atelier Agency' }],
  creator: 'Atelier Agency',
  publisher: 'Atelier Agency',
  robots: {
    index: true,
    follow: true,
    googleBot: {
      index: true,
      follow: true,
      'max-video-preview': -1,
      'max-image-preview': 'large',
      'max-snippet': -1,
    },
  },
  openGraph: {
    type: 'website',
    locale: 'en_US',
    url: 'https://atelier-agency.com',
    siteName: 'Atelier Agency',
    title: 'Atelier Agency - Digital Stonemasonry',
    description: 'Premium web design and development with architectural precision.',
    images: [
      {
        url: '/images/og-image.jpg',
        width: 1200,
        height: 630,
        alt: 'Atelier Agency',
      },
    ],
  },
  twitter: {
    card: 'summary_large_image',
    title: 'Atelier Agency - Digital Stonemasonry',
    description: 'Premium web design and development with architectural precision.',
    creator: '@atelier_agency',
    images: ['/images/twitter-image.jpg'],
  },
};
```

Accessibility Standards

WCAG 2.1 AA Compliance

- Color contrast ratio minimum 4.5:1
- Keyboard navigation support
- Screen reader compatibility
- Focus indicators

- Alt text for all images
- Semantic HTML structure

```
// Accessibility utilities
export const a11y = {
  // Skip to main content
  skipLink: () => (
    <a
      href="#main-content"
      className="skip-link"
      onFocus={(e) => e.target.scrollIntoView()}
    >
      Skip to main content
    </a>
  ),

  // Visually hidden content
  visuallyHidden: (text: string) => (
    <span className="sr-only">{text}</span>
  ),

  // ARIA live region
  liveRegion: (message: string, politeness: 'polite' | 'assertive' = 'polite') => (
    <div
      aria-live={politeness}
      aria-atomic="true"
      className="sr-only"
    >
      {message}
    </div>
  )
};
```

Testing Strategy

Unit Testing

```
// __tests__/components/Button.test.tsx
import { render, fireEvent } from '@testing-library/react';
import { Button } from '@/components/ui/Button';

describe('Button', () => {
  it('renders correctly', () => {
    const { getByRole } = render(<Button>Click me</Button>);
    expect(getByRole('button')).toBeInTheDocument();
  });

  it('handles click events', () => {
    const handleClick = jest.fn();
    const { getByRole } = render(<Button onClick={handleClick}>Click me</Button>);

    fireEvent.click(getByRole('button'));
    expect(handleClick).toHaveBeenCalledTimes(1);
  });
});
```

E2E Testing

```typescript
// e2e/navigation.spec.ts
import { test, expect } from '@playwright/test';

test('navigation works correctly', async ({ page }) => {
  await page.goto('/');

  await page.click('[data-testid="menu-trigger"]');
  await expect(page.locator('[data-testid="navigation"]')).toBeVisible();

  await page.click('[href="/work"]');
  await expect(page).toHaveURL('/work');
  await expect(page.locator('h1')).toContainText('Our Work');
});
```

Browser Support

Supported Browsers

- **Chrome**: Latest 2 versions
- **Firefox**: Latest 2 versions
- **Safari**: Latest 2 versions
- **Edge**: Latest 2 versions

Progressive Enhancement

- Core functionality works without JavaScript
- Enhanced experiences with modern browser features
- Graceful degradation for older browsers

```css
/* Feature detection with @supports */
@supports (backdrop-filter: blur(10px)) {
  .navigation {
    backdrop-filter: blur(10px);
  }
}

@supports not (backdrop-filter: blur(10px)) {
  .navigation {
    background-color: rgba(0, 0, 0, 0.9);
  }
}
```

Security Considerations

Content Security Policy

```
// next.config.js
const ContentSecurityPolicy = `
  default-src 'self';
  script-src 'self' 'unsafe-eval' 'unsafe-inline';
  style-src 'self' 'unsafe-inline';
  img-src 'self' data: https:;
  font-src 'self' https://fonts.gstatic.com;
  connect-src 'self' https://api.example.com;
`;

module.exports = {
  async headers() {
    return [
      {
        source: '/(.*)',
        headers: [
          {
            key: 'Content-Security-Policy',
            value: ContentSecurityPolicy.replace(/\s{2,}/g, ' ').trim()
          }
        ]
      }
    ];
  }
};
```

Data Validation

```
// lib/validation.ts
import { z } from 'zod';

export const ContactFormSchema = z.object({
  name: z.string().min(2, 'Name must be at least 2 characters'),
  email: z.string().email('Please enter a valid email'),
  message: z.string().min(10, 'Message must be at least 10 characters'),
  consent: z.boolean().refine(val => val === true, 'Consent is required')
});
```

Deployment Configuration

Vercel Configuration

```
{
  "buildCommand": "npm run build",
  "outputDirectory": ".next",
  "installCommand": "npm install",
  "framework": "nextjs",
  "regions": ["all"],
  "functions": {
    "app/api/**/*.ts": {
      "maxDuration": 30
    }
  },
  "headers": [
    {
      "source": "/(.*)",
      "headers": [
        {
          "key": "X-Frame-Options",
          "value": "DENY"
        },
        {
          "key": "X-Content-Type-Options",
          "value": "nosniff"
        }
      ]
    }
  ]
}
```

Environment Variables

```
# Required
DATABASE_URL=""
NEXTAUTH_URL=""
NEXTAUTH_SECRET=""

# Optional
NEXT_PUBLIC_ANALYTICS_ID=""
NEXT_PUBLIC_GTM_ID=""
SMTP_HOST=""
SMTP_PORT=""
SMTP_USER=""
SMTP_PASS=""
```

Version History

v2.1 (Current)

- ✅ Complete design system implementation
- ✅ All interactive components (Phases 3A, 3B, 3C)
- ✅ GSAP animation system
- ✅ Professional documentation suite
- ✅ Performance optimizations

- ✅ SEO & accessibility compliance

v2.0

- ✅ Next.js 14 migration
- ✅ TypeScript integration
- ✅ Core components (Phases 1 & 2)
- ✅ Basic animation system

v1.0

- ✅ Initial Nuxt.js implementation
- ✅ Basic design system
- ✅ Core layout components

This specification serves as the definitive technical reference for the Atelier Agency Theme, ensuring consistency, quality, and maintainability across all implementations.