

Outdoor Décor by Eli - Technical Deep Dive

Executive Summary

This document provides a comprehensive technical analysis for building a **zero-lag, industry-grade website** for an important client. Every decision is evaluated against the goals of:

- **Performance:** Instant loading, smooth scrolling, no video lag
 - **SEO:** Maximum local search visibility in Detroit metro
 - **UX:** Buttery-smooth animations matching Moooi/Hajster quality
 - **Maintainability:** Clean architecture for future updates
-

SECTION 1: FRAMEWORK SELECTION

Options Evaluated

Option A: Next.js 14 (App Router)

Pros:

- Industry standard, massive ecosystem
- Hybrid rendering (SSG + SSR + ISR)
- Built-in image optimization
- Vercel deployment is seamless
- React Server Components reduce JS bundle
- Mentioned in original proposal

Cons:

- Can be overkill for purely static sites
- Larger baseline bundle than alternatives
- App Router still maturing (some edge cases)

Performance Score: 8.5/10

Option B: Astro

Pros:

- Zero JS by default (islands architecture)
- Fastest static site generator
- Can use React/Vue/Svelte components where needed
- Perfect for content-first marketing sites
- Automatic image optimization
- Smaller bundle sizes

Cons:

- Smaller ecosystem than Next.js
- Less familiar to most developers
- Limited dynamic capabilities (fine for this use case)

Performance Score: 9.5/10

Option C: Remix

Pros:

- Excellent data loading patterns
- Progressive enhancement built-in
- Great error handling

Cons:

- Overkill for static marketing site
- More suited for dynamic apps
- Smaller ecosystem

Performance Score: 7/10

Option D: Plain Vite + React

Pros:

- Fast development experience
- Full control
- Lightweight

Cons:

- No built-in SSG/SSR
- Manual SEO implementation
- No image optimization built-in

Performance Score: 6/10

🏆 RECOMMENDATION: Next.js 14 (App Router) with Static Export

Why Next.js wins for this project:

1. **Proposal Alignment:** Already specified in signed contract
2. **Static Export:** `(output: 'export')` gives us pure static files with zero server dependency
3. **Image Component:** Built-in lazy loading, WebP conversion, responsive images
4. **Vercel Edge Network:** Global CDN with automatic optimization
5. **React Ecosystem:** Access to Framer Motion, GSAP, all animation libraries
6. **Future-Proof:** If client wants dynamic features later (AI quote tool), we're ready

Configuration for Maximum Performance:

```
javascript
```

```

// next.config.js
/** @type {import('next').NextConfig} */
const nextConfig = {
  output: 'export', // Pure static generation
  images: {
    unoptimized: false,
    formats: ['image/avif', 'image/webp'],
    deviceSizes: [640, 750, 828, 1080, 1200, 1920, 2048],
    imageSizes: [16, 32, 48, 64, 96, 128, 256, 384],
  },
  experimental: {
    optimizeCss: true, // Minify CSS
  },
  compiler: {
    removeConsole: process.env.NODE_ENV === 'production',
  },
  poweredByHeader: false,
  compress: true,
}

module.exports = nextConfig

```

SECTION 2: VIDEO OPTIMIZATION (ZERO LAG)

This is CRITICAL. Video is the hero element and must be flawless.

Video Delivery Strategy

Primary: Cloudinary Video CDN

Why Cloudinary over self-hosted:

- Automatic format negotiation (WebM for Chrome, MP4 for Safari)
- Adaptive bitrate streaming
- Global CDN edge delivery
- On-the-fly compression and resizing
- Lazy loading built-in

Video Encoding Specifications

HERO VIDEO:

- └─ Resolution: 1920x1080 (desktop), 1280x720 (tablet), 854x480 (mobile)
- └─ Format: WebM (VP9) primary, MP4 (H.264) fallback
- └─ Bitrate: 2-4 Mbps desktop, 1-2 Mbps mobile
- └─ Duration: Keep under 15 seconds for hero loops
- └─ Audio: REMOVE (autoplay requires muted)
- └─ Compression: CRF 23-28 for quality/size balance

FFmpeg Encoding Commands

bash

Desktop WebM (VP9) - Best compression

```
ffmpeg -i input.mp4 -c:v libvpx-vp9 -crf 30 -b:v 0 -vf scale=1920:1080 -an hero-desktop.webm
```

Desktop MP4 (H.264) - Safari fallback

```
ffmpeg -i input.mp4 -c:v libx264 -crf 23 -preset slow -vf scale=1920:1080 -an hero-desktop.mp4
```

Mobile WebM

```
ffmpeg -i input.mp4 -c:v libvpx-vp9 -crf 32 -b:v 0 -vf scale=854:480 -an hero-mobile.webm
```

Generate poster image (first frame)

```
ffmpeg -i input.mp4 -vf scale=1920:1080 -q:v 2 hero-poster.jpg
```

React Video Component Implementation

tsx

```
// components/VideoHero.tsx
'use client';

import { useEffect, useRef, useState } from 'react';
import { useInView } from 'react-intersection-observer';

interface VideoHeroProps {
  desktopWebm: string;
  desktopMp4: string;
  mobileWebm: string;
  mobileMp4: string;
  poster: string;
  alt: string;
}

export function VideoHero({
  desktopWebm,
  desktopMp4,
  mobileWebm,
  mobileMp4,
  poster,
  alt,
}: VideoHeroProps) {
  const videoRef = useRef<HTMLVideoElement>(null);
  const [isLoaded, setIsLoaded] = useState(false);
  const [isMobile, setIsMobile] = useState(false);

  const { ref: inViewRef, inView } = useInView({
    threshold: 0.25,
    triggerOnce: false,
  });

  // Detect mobile on mount
  useEffect(() => {
    setIsMobile(window.innerWidth < 768);
    const handleResize = () => setIsMobile(window.innerWidth < 768);
    window.addEventListener('resize', handleResize);
    return () => window.removeEventListener('resize', handleResize);
  }, []);

  // Play/pause based on visibility (performance optimization)
  useEffect(() => {
    if (!videoRef.current) return;

    if (inView) {
      videoRef.current.play().catch(() => {

```

```
// Autoplay blocked - show poster instead
setIsLoaded(true);
});
} else {
  videoRef.current.pause();
}
}, [inView]);

// Combine refs
const setRefs = (el: HTMLVideoElement) => {
  videoRef.current = el;
  inViewRef(el);
};

return (
  <div className="relative w-full h-screen overflow-hidden">
    {/* Poster image shown while video loads */}
    <img
      src={poster}
      alt={alt}
      className={`absolute inset-0 w-full h-full object-cover transition-opacity duration-500 ${isLoaded ? 'opacity-0' : 'opacity-100'}`}
    >
    loading="eager" // Hero should load immediately
    fetchPriority="high"
  />

  {/* Video element */}
  <video
    ref={setRefs}
    className={`absolute inset-0 w-full h-full object-cover transition-opacity duration-500 ${isLoaded ? 'opacity-100' : 'opacity-0'}`}
    autoPlay
    muted
    loop
    playsInline // Critical for iOS
    preload="metadata" // Don't preload full video
    onLoadedData={() => setIsLoaded(true)}
    poster={poster}
  >
    {/* Serve different sources based on device */}
    <source
      src={isMobile ? mobileWebm : desktopWebm}
      type="video/webm"
    />
    <source
```

```

    src={isMobile ? mobileMp4 : desktopMp4}
    type="video/mp4"
  />
</video>

/* Overlay gradient for text readability */
<div className="absolute inset-0 bg-gradient-to-b from-black/30 via-transparent to-black/50" />
</div>
);
}

```

Video Performance Checklist

Optimization	Implementation
Lazy load off-screen videos	Intersection Observer
Poster images	First frame as JPEG/WebP
Multiple resolutions	Desktop/Tablet/Mobile variants
Modern formats	WebM (VP9) with MP4 fallback
Remove audio track	Required for autoplay, reduces size 30%+
CDN delivery	Cloudinary or Vercel Edge
Preload strategy	preload="metadata" not "auto"
Pause when off-screen	Saves CPU/battery
Hardware acceleration	CSS will-change: transform

SECTION 3: SEO TECHNICAL IMPLEMENTATION

Meta Tags Structure

Global Layout Meta Tags

tsx

```
// app/layout.tsx

import { Metadata } from 'next';

export const metadata: Metadata = {
  metadataBase: new URL('https://outdoordecorbyeliz.com'),
  title: {
    default: 'Outdoor Décor by Eli | Seasonal Planter Subscriptions | Detroit Metro',
    template: '%s | Outdoor Décor by Eli',
  },
  description: 'Premium seasonal planter subscriptions for Detroit metro homes. Beautiful year-round outdoor décor with zero effort. Premium seasonal planter subscriptions for Detroit metro homes. Beautiful year-round outdoor décor with zero effort.',
  keywords: [
    'seasonal planters Detroit',
    'planter subscription Michigan',
    'outdoor décor Birmingham MI',
    'seasonal flower delivery',
    'container gardening service',
    'front porch planters',
    'curb appeal Detroit',
  ],
  authors: [{ name: 'Outdoor Décor by Eli' }],
  creator: 'Outdoor Décor by Eli',
  publisher: 'Outdoor Décor by Eli',
  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://outdoordecorbyeliz.com',
    siteName: 'Outdoor Décor by Eli',
    title: 'Outdoor Décor by Eli | Seasonal Planter Subscriptions',
    description: 'Beautiful year-round outdoor décor with zero effort. Premium seasonal planter subscriptions for Detroit metro homes. Beautiful year-round outdoor décor with zero effort. Premium seasonal planter subscriptions for Detroit metro homes.',
    images: [
      {
        url: '/og-image.jpg',
        width: 1200,
        height: 630,
        alt: 'Outdoor Décor by Eli - Seasonal Planter Subscriptions',
      }
    ],
  }
}
```

```
  },
  ],
},
twitter: {
  card: 'summary_large_image',
  title: 'Outdoor Décor by Eli | Seasonal Planter Subscriptions',
  description: 'Beautiful year-round outdoor décor with zero effort!',
  images: ['/og-image.jpg'],
},
verification: {
  google: 'YOUR_GOOGLE_VERIFICATION_CODE',
},
alternates: {
  canonical: 'https://outdoordecorbyeliz.com',
},
};

});
```

Page-Specific Meta Tags Example

```
tsx

// app/services/residential/page.tsx
import { Metadata } from 'next';

export const metadata: Metadata = {
  title: 'Residential Planter Subscriptions',
  description: 'Premium seasonal planter subscriptions for Detroit area homes. 4 seasonal refreshes per year with professio',
  openGraph: {
    title: 'Residential Planter Subscriptions | Outdoor Décor by Eli',
    description: 'Transform your home\'s curb appeal with our turnkey seasonal planter service!',
    images: ['/images/residential-og.jpg'],
  },
  alternates: {
    canonical: 'https://outdoordecorbyeliz.com/services/residential',
  },
};
```

Schema Markup (Structured Data)

Local Business Schema

```
tsx
```

```
// components/LocalBusinessSchema.tsx

export function LocalBusinessSchema() {
  const schema = {
    '@context': 'https://schema.org',
    '@type': 'LocalBusiness',
    '@id': 'https://outdoordecorbyeliz.com/#business',
    name: 'Outdoor Décor by Eli',
    description: 'Premium seasonal planter subscription service for Detroit metro area homes and businesses.',
    url: 'https://outdoordecorbyeliz.com',
    telephone: '+1-XXX-XXX-XXXX',
    email: 'info@outdoordecorbyeliz.com',
    address: {
      '@type': 'PostalAddress',
      addressLocality: 'Detroit',
      addressRegion: 'MI',
      addressCountry: 'US',
    },
    geo: {
      '@type': 'GeoCoordinates',
      latitude: 42.3314,
      longitude: -83.0458,
    },
    areaServed: [
      { '@type': 'City', name: 'Birmingham', containedInPlace: { '@type': 'State', name: 'Michigan' } },
      { '@type': 'City', name: 'Bloomfield Hills', containedInPlace: { '@type': 'State', name: 'Michigan' } },
      { '@type': 'City', name: 'Grosse Pointe', containedInPlace: { '@type': 'State', name: 'Michigan' } },
      { '@type': 'City', name: 'Royal Oak', containedInPlace: { '@type': 'State', name: 'Michigan' } },
      { '@type': 'City', name: 'Troy', containedInPlace: { '@type': 'State', name: 'Michigan' } },
    ],
    priceRange: '$89-$249/month',
    openingHoursSpecification: {
      '@type': 'OpeningHoursSpecification',
      dayOfWeek: ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday'],
      opens: '08:00',
      closes: '18:00',
    },
    sameAs: [
      'https://www.facebook.com/outdoordecorbyeliz',
      'https://www.instagram.com/outdoordecorbyeliz',
    ],
    image: 'https://outdoordecorbyeliz.com/images/storefront.jpg',
    logo: 'https://outdoordecorbyeliz.com/logo.png',
  };
}

return (
  <script
```

```
type="application/ld+json"
dangerouslySetInnerHTML={{ __html: JSON.stringify(schema) }}
/>
);
}
```

Service Schema

tsx

```
// components/ServiceSchema.tsx
export function ServiceSchema() {
  const schema = {
    '@context': 'https://schema.org',
    '@type': 'Service',
    serviceType: 'Seasonal Planter Subscription',
    provider: {
      '@type': 'LocalBusiness',
      name: 'Outdoor Décor by Eli',
    },
    areaServed: {
      '@type': 'State',
      name: 'Michigan',
    },
    hasOfferCatalog: {
      '@type': 'OfferCatalog',
      name: 'Subscription Packages',
      itemListElement: [
        {
          '@type': 'Offer',
          itemOffered: {
            '@type': 'Service',
            name: 'Garden Glow',
            description: '2-3 planters with 4 seasonal refreshes per year',
          },
          price: '89.00',
          priceCurrency: 'USD',
          priceSpecification: {
            '@type': 'UnitPriceSpecification',
            price: '89.00',
            priceCurrency: 'USD',
            billingDuration: 'P1M',
          },
        },
        {
          '@type': 'Offer',
          itemOffered: {
            '@type': 'Service',
            name: 'Entrance Enchantment',
            description: '4-6 planters with seasonal accents and holiday touches',
          },
          price: '149.00',
          priceCurrency: 'USD',
          priceSpecification: {
            '@type': 'UnitPriceSpecification',
            price: '149.00',
          }
        }
      ]
    }
  }
}
```

```

    priceCurrency: 'USD',
    billingDuration: 'P1M',
  },
},
{
  '@type': 'Offer',
  itemOffered: {
    '@type': 'Service',
    name: 'Estate Elegance',
    description: '8+ planters with full seasonal design and premium materials',
  },
  price: '249.00',
  priceCurrency: 'USD',
  priceSpecification: {
    '@type': 'UnitPriceSpecification',
    price: '249.00',
    priceCurrency: 'USD',
    billingDuration: 'P1M',
  },
},
],
},
};

return (
<script
  type="application/ld+json"
  dangerouslySetInnerHTML={{ __html: JSON.stringify(schema) }}
/>
);
}

```

FAQ Schema

tsx

```

// components/FAQSchema.tsx

export function FAQSchema({ faqs }: { faqs: { question: string; answer: string }[] }) {
  const schema = {
    '@context': 'https://schema.org',
    '@type': 'FAQPage',
    mainEntity: faqs.map((faq) => ({
      '@type': 'Question',
      name: faq.question,
      acceptedAnswer: {
        '@type': 'Answer',
        text: faq.answer,
      },
    })),
  };

  return (
    <script
      type="application/ld+json"
      dangerouslySetInnerHTML={{ __html: JSON.stringify(schema) }}
    />
  );
}

```

Technical SEO Files

Sitemap Generation

tsx

```

// app/sitemap.ts

import { MetadataRoute } from 'next';

const BASE_URL = 'https://outdoordecorbyeliz.com';

// Detroit metro neighborhoods for service area pages

const neighborhoods = [
  'birmingham',
  'bloomfield-hills',
  'grosse-pointe',
  'royal-oak',
  'troy',
  'rochester',
  'west-bloomfield',
  'farmington-hills',
  'northville',
  'plymouth',
];

export default function sitemap(): MetadataRoute.Sitemap {
  const staticPages = [
    { url: BASE_URL, lastModified: new Date(), changeFrequency: 'weekly', priority: 1.0 },
    { url: `${BASE_URL}/about`, lastModified: new Date(), changeFrequency: 'monthly', priority: 0.8 },
    { url: `${BASE_URL}/services/residential`, lastModified: new Date(), changeFrequency: 'monthly', priority: 0.9 },
    { url: `${BASE_URL}/services/commercial`, lastModified: new Date(), changeFrequency: 'monthly', priority: 0.8 },
    { url: `${BASE_URL}/services/events`, lastModified: new Date(), changeFrequency: 'monthly', priority: 0.7 },
    { url: `${BASE_URL}/services/maintenance`, lastModified: new Date(), changeFrequency: 'monthly', priority: 0.6 },
    { url: `${BASE_URL}/gallery`, lastModified: new Date(), changeFrequency: 'weekly', priority: 0.8 },
    { url: `${BASE_URL}/pricing`, lastModified: new Date(), changeFrequency: 'monthly', priority: 0.9 },
    { url: `${BASE_URL}/faq`, lastModified: new Date(), changeFrequency: 'monthly', priority: 0.7 },
    { url: `${BASE_URL}/contact`, lastModified: new Date(), changeFrequency: 'yearly', priority: 0.6 },
  ];

  // Generate service area pages

  const serviceAreaPages = neighborhoods.map((neighborhood) => ({
    url: `${BASE_URL}/areas/${neighborhood}`,
    lastModified: new Date(),
    changeFrequency: 'monthly' as const,
    priority: 0.7,
  }));
}

return [...staticPages, ...serviceAreaPages];
}

```

Robots.txt

```
tsx

// app/robots.ts
import { MetadataRoute } from 'next';

export default function robots(): MetadataRoute.Robots {
  return {
    rules: [
      {
        userAgent: '*',
        allow: '/',
        disallow: ['/api/', '/admin/', '/_next/'],
      },
    ],
    sitemap: 'https://outdoordecorbyeliz.com/sitemap.xml',
  };
}
```

SECTION 4: PERFORMANCE OPTIMIZATION

Core Web Vitals Targets

Metric	Target	How We Achieve It
LCP (Largest Contentful Paint)	< 2.5s	Optimized hero image/video, preload critical assets
FID (First Input Delay)	< 100ms	Minimal JS, code splitting, defer non-critical
CLS (Cumulative Layout Shift)	< 0.1	Explicit dimensions on images/video, font loading
TTFB (Time to First Byte)	< 200ms	Static export + Edge CDN
FCP (First Contentful Paint)	< 1.8s	Critical CSS inlined, preload fonts

Image Optimization Strategy

Next.js Image Component Configuration

```
tsx
```

```

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

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

export function OptimizedImage({
  src,
  alt,
  priority = false,
  className,
  fill,
  width,
  height,
}: OptimizedImageProps) {
  return (
    <Image
      src={src}
      alt={alt}
      priority={priority}
      className={className}
      fill={fill}
      width={width}
      height={height}
      sizes="(max-width: 640px) 100vw, (max-width: 1024px) 50vw, 33vw"
      quality={85}
      placeholder="blur"
      blurDataURL="..." />
  );
}

```

Image Format Strategy

Priority Order:

1. AVIF (best compression, ~50% smaller than JPEG)
2. WebP (~30% smaller than JPEG)
3. JPEG (fallback)

Next.js handles this automatically with:

```
images: { formats: ['image/avif', 'image/webp'] }
```

Font Loading Strategy

tsx

```
// app/layout.tsx
import { Inter, Playfair_Display } from 'next/font/google';

// Sans-serif for body text
const inter = Inter({
  subsets: ['latin'],
  display: 'swap', // Prevent FOIT (Flash of Invisible Text)
  variable: '--font-inter',
  preload: true,
});

// Serif for headlines
const playfair = Playfair_Display({
  subsets: ['latin'],
  display: 'swap',
  variable: '--font-playfair',
  preload: true,
});

export default function RootLayout({ children }: { children: React.ReactNode }) {
  return (
    <html lang="en" className={`${inter.variable} ${playfair.variable}`}>
      <body className="font-sans">{children}</body>
    </html>
  );
}
```

Code Splitting & Bundle Optimization

Dynamic Imports for Heavy Components

tsx

```

// Lazy load animation-heavy components
import dynamic from 'next/dynamic';

const BeforeAfterSlider = dynamic(
() => import('@/components/BeforeAfterSlider'),
{
  loading: () => <div className="h-96 bg-gray-100 animate-pulse" />,
  ssr: false // Don't SSR animation-heavy components
}
);

const TestimonialCarousel = dynamic(
() => import('@/components/TestimonialCarousel'),
{ loading: () => <div className="h-64 bg-gray-100 animate-pulse" /> }
);

const SeasonalGallery = dynamic(
() => import('@/components/SeasonalGallery'),
{ loading: () => <div className="h-96 bg-gray-100 animate-pulse" /> }
);

```

Bundle Analysis

```

bash

# Add to package.json scripts
"analyze": "ANALYZE=true next build"

# Install analyzer
npm install @next/bundle-analyzer

```

```

javascript

// next.config.js
const withBundleAnalyzer = require('@next/bundle-analyzer')({
  enabled: process.env.ANALYZE === 'true',
});

module.exports = withBundleAnalyzer(nextConfig);

```

SECTION 5: ANIMATION & UI IMPLEMENTATION

Animation Library Selection

Option A: Framer Motion

Pros:

- Best React animation library
- Declarative API
- Excellent scroll animations
- Gesture support
- Layout animations

Cons:

- ~40KB bundle size
- Can cause layout shifts if not careful

Best For: Complex scroll animations, page transitions

Option B: GSAP (GreenSock)

Pros:

- Industry standard for web animation
- ScrollTrigger plugin is incredible
- Better performance for complex sequences
- Timeline control

Cons:

- ~60KB bundle size
- Less React-native API

Best For: Advanced scroll-triggered effects, timeline sequences

Option C: CSS-only with Intersection Observer

Pros:

- Zero bundle size

- Native performance
- Simple effects

Cons:

- Limited animation capabilities
- More manual code

Best For: Simple fade-ins, basic scroll reveals

 **RECOMMENDATION: Framer Motion + CSS Hybrid**

Use Framer Motion for complex animations, CSS for simple effects.

Scroll-Triggered Animations (Hajster-style)

tsx

```
// components/ScrollReveal.tsx
'use client';

import { motion, useInView } from 'framer-motion';
import { useRef } from 'react';

interface ScrollRevealProps {
  children: React.ReactNode;
  direction?: 'up' | 'down' | 'left' | 'right';
  delay?: number;
  className?: string;
}

export function ScrollReveal({
  children,
  direction = 'up',
  delay = 0,
  className,
}: ScrollRevealProps) {
  const ref = useRef(null);
  const isInView = useInView(ref, { once: true, margin: '-100px' });

  const directions = {
    up: { y: 40, x: 0 },
    down: { y: -40, x: 0 },
    left: { y: 0, x: 40 },
    right: { y: 0, x: -40 },
  };

  return (
    <motion.div
      ref={ref}
      initial={{ opacity: 0, ...directions[direction] }}
      animate={isInView ? { opacity: 1, y: 0, x: 0 } : {}}
      transition={{
        duration: 0.6,
        delay,
        ease: [0.21, 0.47, 0.32, 0.98], // Custom easing
      }}
      className={className}
    >
      {children}
    </motion.div>
  );
}
```

Text Highlight on Scroll (Hajster-style)

tsx

```
// components/ScrollHighlightText.tsx
'use client';

import { motion, useScroll, useTransform } from 'framer-motion';
import { useRef } from 'react';

interface ScrollHighlightTextProps {
  text: string;
  className?: string;
}

export function ScrollHighlightText({ text, className }: ScrollHighlightTextProps) {
  const containerRef = useRef<HTMLDivElement>(null);
  const { scrollYProgress } = useScroll({
    target: containerRef,
    offset: ['start 0.9', 'start 0.25'],
  });

  const words = text.split(' ');

  return (
    <div ref={containerRef} className={className}>
      <p className="flex flex-wrap">
        {words.map((word, i) => {
          const start = i / words.length;
          const end = start + 1 / words.length;
          return (
            <Word key={i} progress={scrollYProgress} range={[start, end]}>
              {word}
            </Word>
          );
        })}
      </p>
    </div>
  );
}

function Word({
  children,
  progress,
  range,
}: {
  children: string;
  progress: any;
  range: [number, number];
}) {
```

```
const opacity = useTransform(progress, range, [0.2, 1]);
const color = useTransform(
  progress,
  range,
  ['rgb(156, 163, 175)', 'rgb(17, 24, 39)'] // gray-400 to gray-900
);

return (
  <motion.span
    style={{ opacity, color }}
    className="mr-2 mb-1 transition-colors"
  >
  {children}
  </motion.span>
);
}
```

Split-Screen Before/After (Moooi-style)

tsx

```
// components/BeforeAfterSplit.tsx
'use client';

import { motion, useScroll, useTransform } from 'framer-motion';
import { useRef } from 'react';
import Image from 'next/image';

interface BeforeAfterSplitProps {
  beforeImage: string;
  afterImage: string;
  beforeAlt: string;
  afterAlt: string;
}

export function BeforeAfterSplit({
  beforeImage,
  afterImage,
  beforeAlt,
  afterAlt,
}: BeforeAfterSplitProps) {
  const containerRef = useRef<HTMLDivElement>(null);
  const { scrollYProgress } = useScroll({
    target: containerRef,
    offset: ['start end', 'end start'],
  });

  const clipPath = useTransform(
    scrollYProgress,
    [0, 0.5, 1],
    ['inset(0 50% 0 0)', 'inset(0 0% 0 0)', 'inset(0 0% 0 0)']
  );

  const beforeOpacity = useTransform(scrollYProgress, [0.3, 0.5], [1, 0.3]);

  return (
    <div ref={containerRef} className="relative h-[150vh]">
      <div className="sticky top-0 h-screen flex items-center overflow-hidden">
        {/* Before Image (Full width, fades) */}
        <motion.div
          className="absolute inset-0"
          style={{ opacity: beforeOpacity }}
        >
          <Image
            src={beforeImage}
            alt={beforeAlt}
            fill
          >
        
      
```

```

    className="object-cover grayscale"
  />
  <div className="absolute inset-0 bg-black/30" />
  <span className="absolute top-8 left-8 text-white text-2xl font-light tracking-wider">
    BEFORE
  </span>
</motion.div>

/* After Image (Reveals from right) */
<motion.div className="absolute inset-0" style={{ clipPath }}>
  <Image
    src={afterImage}
    alt={afterAlt}
    fill
    className="object-cover"
  />
  <span className="absolute top-8 right-8 text-white text-2xl font-light tracking-wider">
    AFTER
  </span>
</motion.div>
</div>
</div>
);
}

```

Smooth Page Transitions

tsx

```
// components/PageTransition.tsx
'use client';

import { motion, AnimatePresence } from 'framer-motion';
import { usePathname } from 'next/navigation';

export function PageTransition({ children }: { children: React.ReactNode }) {
  const pathname = usePathname();

  return (
    <AnimatePresence mode="wait">
      <motion.div
        key={pathname}
        initial={{ opacity: 0, y: 20 }}
        animate={{ opacity: 1, y: 0 }}
        exit={{ opacity: 0, y: -20 }}
        transition={{ duration: 0.3, ease: 'easeInOut' }}
      >
        {children}
      </motion.div>
    </AnimatePresence>
  );
}
```

SECTION 6: CSS & STYLING

Tailwind CSS Configuration

```
javascript
```

```
// tailwind.config.js
/** @type {import('tailwindcss').Config} */
module.exports = {
  content: [
    './app/**/*.{js,ts,jsx,tsx,mdx}',
    './components/**/*.{js,ts,jsx,tsx,mdx}'
  ],
  theme: {
    extend: {
      colors: {
        // Brand colors from proposal
        'warm-white': '#FEFCFB',
        'forest': {
          DEFAULT: '#2D6A4F',
          50: '#E8F5E9',
          100: '#C8E6C9',
          200: '#A5D6A7',
          300: '#81C784',
          400: '#66BB6A',
          500: '#2D6A4F',
          600: '#256141',
          700: '#1D5833',
          800: '#154F25',
          900: '#0D4617',
        },
        'terracotta': {
          DEFAULT: '#D4A373',
          50: '#FDF8F3',
          100: '#FAEFE3',
          200: '#F5DFC7',
          300: '#EFCFAB',
          400: '#E9BF8F',
          500: '#D4A373',
          600: '#C4895A',
          700: '#A86F42',
          800: '#8B5A33',
          900: '#6E4625',
        },
        'rich-black': '#1B1B1B',
      },
      fontFamily: {
        sans: ['var(--font-inter)', 'system-ui', 'sans-serif'],
        serif: ['var(--font-playfair)', 'Georgia', 'serif'],
      },
      animation: {
        'fade-in': 'fadeIn 0.5s ease-out',
      }
    }
  }
}
```

CSS Performance Rules

css

```
/* globals.css */

@tailwind base;
@tailwind components;
@tailwind utilities;

@layer base {
  /* Smooth scrolling */
  html {
    scroll-behavior: smooth;
  }

  /* Prevent layout shift from scrollbar */
  html {
    overflow-y: scroll;
  }

  /* Optimize text rendering */
  body {
    -webkit-font-smoothing: antialiased;
    -moz-osx-font-smoothing: grayscale;
    text-rendering: optimizeLegibility;
  }

  /* GPU acceleration for animated elements */
  .will-animate {
    will-change: transform, opacity;
    transform: translateZ(0);
  }
}

@layer utilities {
  /* Smooth transitions */
  .transition-smooth {
    @apply transition-all duration-300 ease-out;
  }

  /* Text balance for headlines */
  .text-balance {
    text-wrap: balance;
  }

  /* Aspect ratios */
  .aspect-hero {
    aspect-ratio: 16 / 9;
  }
}
```

```
.aspect-portrait {  
    aspect-ratio: 3 / 4;  
}  
}
```

SECTION 7: MOBILE OPTIMIZATION

Responsive Breakpoints Strategy

Mobile First Approach:

- └─ Base: 0px (mobile)
- └─ sm: 640px (large phones)
- └─ md: 768px (tablets)
- └─ lg: 1024px (laptops)
- └─ xl: 1280px (desktops)
- └─ 2xl: 1536px (large screens)

Mobile-Specific Video Handling

tsx

```
// hooks/useMediaQuery.ts
'use client';

import { useState, useEffect } from 'react';

export function useMediaQuery(query: string): boolean {
  const [matches, setMatches] = useState(false);

  useEffect(() => {
    const media = window.matchMedia(query);
    setMatches(media.matches);

    const listener = (e: MediaQueryListEvent) => setMatches(e.matches);
    media.addEventListener('change', listener);
    return () => media.removeEventListener('change', listener);
  }, [query]);

  return matches;
}

// Usage
const isMobile = useMediaQuery('(max-width: 768px)');
```

Mobile Video Fallback

tsx

```

// components/ResponsiveHero.tsx
'use client';

import { useMediaQuery } from '@/hooks/useMediaQuery';
import { VideoHero } from './VideoHero';
import Image from 'next/image';

export function ResponsiveHero() {
  const isMobile = useMediaQuery('(max-width: 768px)');
  const prefersReducedMotion = useMediaQuery('(prefers-reduced-motion: reduce)');

  // On mobile with slow connection or reduced motion preference, show static image
  if (prefersReducedMotion) {
    return (
      <div className="relative h-screen">
        <Image
          src="/images/hero-poster.jpg"
          alt="Beautiful seasonal planters"
          fill
          className="object-cover"
          priority
        />
      </div>
    );
  }

  return (
    <VideoHero
      desktopWebm="/video/hero-desktop.webm"
      desktopMp4="/video/hero-desktop.mp4"
      mobileWebm="/video/hero-mobile.webm"
      mobileMp4="/video/hero-mobile.mp4"
      poster="/images/hero-poster.jpg"
      alt="Seasonal planter transformations"
    />
  );
}

```

Touch-Optimized Interactions

tsx

```
// components/TouchSlider.tsx
'use client';

import { useState } from 'react';
import { motion, PanInfo } from 'framer-motion';

interface TouchSliderProps {
  images: { src: string; alt: string }[];
}

export function TouchSlider({ images }: TouchSliderProps) {
  const [currentIndex, setCurrentIndex] = useState(0);

  const handleDragEnd = (e: MouseEvent |TouchEvent, info: PanInfo) => {
    const threshold = 50;
    if (info.offset.x > threshold && currentIndex > 0) {
      setCurrentIndex(currentIndex - 1);
    } else if (info.offset.x < -threshold && currentIndex < images.length - 1) {
      setCurrentIndex(currentIndex + 1);
    }
  };

  return (
    <div className="overflow-hidden">
      <motion.div
        className="flex"
        drag="x"
        dragConstraints={{ left: 0, right: 0 }}
        dragElastic={0.1}
        onDragEnd={handleDragEnd}
        animate={{ x: -currentIndex * 100 + '%' }}
        transition={{ type: 'spring', stiffness: 300, damping: 30 }}
      >
        {images.map((image, i) =>
          <div key={i} className="min-w-full">
            <img
              src={image.src}
              alt={image.alt}
              className="w-full h-auto"
              draggable={false}
            />
          </div>
        ))}
      </motion.div>
    {/* Dots indicator */}
  );
}
```

```
<div className="flex justify-center gap-2 mt-4">
  {images.map((_, i) => (
    <button
      key={i}
      onClick={() => setCurrentIndex(i)}
      className={`w-2 h-2 rounded-full transition-colors ${(
        i === currentIndex ? 'bg-forest' : 'bg-gray-300'
      )}`}
    />
  )));
</div>
</div>
);
}
```

SECTION 8: PACKAGE.JSON & DEPENDENCIES

Complete Package.json

```
json
```

```
{
  "name": "outdoor-decor-by-eli",
  "version": "1.0.0",
  "private": true,
  "scripts": {
    "dev": "next dev",
    "build": "next build",
    "start": "next start",
    "lint": "next lint",
    "analyze": "ANALYZE=true next build",
    "lighthouse": "lighthouse http://localhost:3000 --output html --output-path ./lighthouse-report.html"
  },
  "dependencies": {
    "next": "14.2.x",
    "react": "18.2.x",
    "react-dom": "18.2.x",
    "framer-motion": "^11.0.0",
    "react-intersection-observer": "^9.8.0",
    "@tailwindcss/forms": "^0.5.7",
    "@tailwindcss/typography": "^0.5.10",
    "clsx": "^2.1.0",
    "tailwind-merge": "^2.2.0"
  },
  "devDependencies": {
    "@next/bundle-analyzer": "^14.2.0",
    "@types/node": "^20.11.0",
    "@types/react": "^18.2.0",
    "@types/react-dom": "^18.2.0",
    "autoprefixer": "^10.4.17",
    "eslint": "^8.56.0",
    "eslint-config-next": "14.2.x",
    "postcss": "^8.4.35",
    "tailwindcss": "^3.4.1",
    "typescript": "^5.3.0"
  }
}
```

Key Dependency Explanations

Package	Purpose	Bundle Impact
next	Framework	Core requirement
framer-motion	Animations	~40KB (tree-shakeable)

Package	Purpose	Bundle Impact
react-intersection-observer	Scroll detection	~3KB
clsx + tailwind-merge	Class utilities	~2KB
@tailwindcss/forms	Form styling	CSS only
@tailwindcss/typography	Prose styling	CSS only

SECTION 9: HOSTING & DEPLOYMENT

Vercel Configuration

```
json
```

```
// vercel.json
{
  "framework": "nextjs",
  "buildCommand": "next build",
  "outputDirectory": ".next",
  "regions": ["iad1"], // US East (closest to Detroit)
  "headers": [
    {
      "source": "/(.*",
      "headers": [
        {
          "key": "X-Content-Type-Options",
          "value": "nosniff"
        },
        {
          "key": "X-Frame-Options",
          "value": "DENY"
        },
        {
          "key": "X-XSS-Protection",
          "value": "1; mode=block"
        },
        {
          "key": "Referrer-Policy",
          "value": "strict-origin-when-cross-origin"
        }
      ]
    },
    {
      "source": "/video/(.*)",
      "headers": [
        {
          "key": "Cache-Control",
          "value": "public, max-age=31536000, immutable"
        }
      ]
    },
    {
      "source": "/images/(.*)",
      "headers": [
        {
          "key": "Cache-Control",
          "value": "public, max-age=31536000, immutable"
        }
      ]
    }
  ]
}
```

```
    ]  
}  
]
```

Edge Network Benefits

- **Global CDN:** Static assets served from 100+ edge locations
- **Automatic HTTPS:** SSL certificates managed automatically
- **DDoS Protection:** Built-in protection
- **Compression:** Automatic Brotli/Gzip compression
- **Image Optimization:** On-the-fly resizing and format conversion

SECTION 10: MICROSOFT CLARITY INTEGRATION

```
tsx  
  
// components/ClarityAnalytics.tsx  
'use client';  
  
import Script from 'next/script';  
  
export function ClarityAnalytics() {  
  return (  
    <Script  
      id="microsoft-clarity"  
      strategy="afterInteractive"  
    >  
      {  
        (function(c,l,a,r,i,t,y){  
          c[a]=c[a]||function(){(c[a].q=c[a].q||[]).push(arguments)};  
          t=l.createElement(r);t.async=1;t.src="https://www.clarity.ms/tag/"+i;  
          y=l.getElementsByTagName(r)[0];y.parentNode.insertBefore(t,y);  
        })(window,document,"clarity","script","YOUR_CLARITY_PROJECT_ID");  
      }  
    </Script>  
  );  
}
```

SECTION 11: FILE STRUCTURE

```
outdoor-decor-by-eli/
├── app/
│   ├── layout.tsx      # Root layout with fonts, metadata
│   ├── page.tsx        # Homepage
│   ├── globals.css     # Tailwind + custom CSS
│   ├── sitemap.ts      # Dynamic sitemap
│   ├── robots.ts        # Robots.txt
│   ├── about/
│   │   └── page.tsx
│   ├── services/
│   │   ├── residential/
│   │   │   └── page.tsx
│   │   ├── commercial/
│   │   │   └── page.tsx
│   │   ├── events/
│   │   │   └── page.tsx
│   │   ├── maintenance/
│   │   │   └── page.tsx
│   │   ├── gallery/
│   │   │   └── page.tsx
│   │   ├── pricing/
│   │   │   └── page.tsx
│   │   ├── areas/
│   │   │   └── [neighborhood]/
│   │   │       └── page.tsx    # Dynamic service area pages
│   │   ├── faq/
│   │   │   └── page.tsx
│   │   ├── contact/
│   │   │   └── page.tsx
│   ├── components/
│   │   ├── layout/
│   │   │   ├── Header.tsx
│   │   │   ├── Footer.tsx
│   │   │   ├── Navigation.tsx
│   │   │   └── MobileMenu.tsx
│   │   ├── ui/
│   │   │   ├── Button.tsx
│   │   │   ├── Card.tsx
│   │   │   ├── Input.tsx
│   │   │   └── Accordion.tsx
│   │   ├── sections/
│   │   │   ├── VideoHero.tsx
│   │   │   ├── TrustBar.tsx
│   │   │   └── HowItWorks.tsx
```

```

    ├── BeforeAfterGallery.tsx
    ├── PackageShowcase.tsx
    ├── QuoteRequestForm.tsx
    ├── OurStory.tsx
    ├── Testimonials.tsx
    ├── SeasonalGallery.tsx
    └── FAQ.tsx
  ├── animations/
    ├── ScrollReveal.tsx
    ├── ScrollHighlightText.tsx
    ├── BeforeAfterSplit.tsx
    └── PageTransition.tsx
  └── seo/
    ├── LocalBusinessSchema.tsx
    ├── ServiceSchema.tsx
    └── FAQSchema.tsx
  └── hooks/
    ├── useMediaQuery.ts
    ├── useIntersectionObserver.ts
    └── useScrollProgress.ts
  └── lib/
    ├── utils.ts      # clsx/tailwind-merge helpers
    └── constants.ts  # Site-wide constants
  └── public/
    ├── images/
    ├── video/
    ├── fonts/
    └── favicon.ico
  ├── next.config.js
  ├── tailwind.config.js
  ├── postcss.config.js
  ├── tsconfig.json
  ├── package.json
  └── vercel.json

```

SECTION 12: PERFORMANCE TESTING CHECKLIST

Pre-Launch Performance Audit

Test	Tool	Target
Lighthouse Performance	Chrome DevTools	90+
Lighthouse Accessibility	Chrome DevTools	100

Test	Tool	Target
Lighthouse SEO	Chrome DevTools	100
Core Web Vitals	PageSpeed Insights	All "Good"
Mobile Usability	Google Search Console	Pass
Schema Validation	Schema.org Validator	No errors
HTML Validation	W3C Validator	No errors
Load Time	WebPageTest	< 3s full load
Video Playback	Manual testing	Smooth on 3G

Post-Launch Monitoring

- **Microsoft Clarity:** Heatmaps, session recordings, rage clicks
 - **Google Search Console:** Indexing, Core Web Vitals, search performance
 - **Vercel Analytics:** Real user monitoring, edge function performance
-

SUMMARY: TECHNICAL STACK

Category	Choice	Rationale
Framework	Next.js 14 (App Router)	Best balance of features, performance, ecosystem
Styling	Tailwind CSS	Rapid development, small bundle, design system
Animation	Framer Motion	Best React animation library, scroll triggers
Fonts	Inter + Playfair Display	Readable + elegant, Google Fonts optimized
Images	Next.js Image + AVIF/WebP	Automatic optimization, lazy loading
Video	Cloudinary CDN + Intersection Observer	Adaptive streaming, play/pause optimization
SEO	Built-in Next.js + Schema.org	Maximum local search visibility
Hosting	Vercel	Edge CDN, automatic HTTPS, best DX
Analytics	Microsoft Clarity	Free, heatmaps, session recordings
Forms	Native + API route	Simple email notifications

This technical architecture ensures a zero-lag, industry-grade website that will impress the client and perform flawlessly.