

Portfolio Website - Complete Design System Documentation

Table of Contents

1. [Project Overview](#)
 2. [Technical Stack](#)
 3. [Design System](#)
 4. [Page Structure](#)
 5. [Component Specifications](#)
 6. [Animation System](#)
 7. [Responsive Design](#)
 8. [Implementation Guide](#)
 9. [Development Checklist](#)
-

Project Overview

Concept

A minimalist, dark-mode portfolio website for a Full-Stack Web Developer that prioritizes:

- **Clean aesthetics** with intentional use of space
- **Performance** with fast loading and smooth interactions
- **Uniqueness** through subtle, meaningful animations
- **Professionalism** with a conversational, approachable tone

Design Philosophy

- Show, don't tell (demonstrate skills through the site itself)
- Minimalism with purpose (every element has a reason)
- Breathing room (generous spacing throughout)
- Subtle motion (animations enhance, never distract)
- Permanent dark mode (ink black dominant background)

Technical Stack

Core Framework

- **Next.js 14+** (App Router)
- **React 18+**
- **TypeScript**
- **Tailwind CSS**

Dependencies

```
bash
```

```
npm install framer-motion @web3forms/react
```

Libraries & Tools

- **Framer Motion** - All animations and transitions
- **Web3Forms** - Contact form submission handling
- **Tailwind CSS** - Utility-first styling
- **Google Fonts** - Rubik Regular 400

Optional Integrations

- **Resend** (alternative to Web3Forms for email handling)
 - **Next.js Image Optimization** (built-in)
 - **Vercel Analytics** (if deploying to Vercel)
-

Design System

Color Palette

```
css
```

```
root {  
  --ink-black: #0d1b2a; /* Main background, dominant */  
  --alabaster: #e0e1dd; /* Primary text, highlights */  
  --lavender: #778da9; /* Accents, interactive elements */  
}
```

Usage Guidelines:

- **Ink Black (#0d1b2a):**
 - All page backgrounds
 - Card/section backgrounds
 - Dominant color throughout entire site
 - Website is ALWAYS in dark mode
- **Alabaster Grey (#e0e1dd):**
 - Body text
 - Headings
 - Borders and dividers
 - Active states
 - 80-90% opacity for secondary text
- **Lavender Grey (#778da9):**
 - Links and CTAs
 - Hover states
 - Accent borders
 - Tags and labels
 - Secondary information (dates, metadata)

Background Variations:

- Pure ink black:  #0d1b2a
- Subtle variant:  #0d1b2a with 2% alabaster overlay for section alternation

Typography

Font Family:

css

```
font-family: 'Rubik', sans-serif;  
font-weight: 400; /* Regular only */
```

Type Scale:

- **Hero Heading:** 4rem (64px) desktop / 3rem (48px) mobile
- **H1:** 3rem (48px) desktop / 2.5rem (40px) mobile
- **H2:** 2.5rem (40px) desktop / 2rem (32px) mobile
- **H3:** 2rem (32px) desktop / 1.75rem (28px) mobile
- **Body Large:** 1.25rem (20px)
- **Body:** 1rem (16px)
- **Small:** 0.875rem (14px)

Line Height:

- Headings: 1.2
- Body text: 1.6-1.8
- Small text: 1.5

Letter Spacing:

- Headings: -0.02em
- Body: normal

Hierarchy Through:

- Size variations
- Opacity changes (100%, 90%, 80%)
- Color variations (alabaster vs lavender)
- Spacing/whitespace

Spacing System

Section Padding:

- Desktop: 200-300px vertical

- Mobile: 80-120px vertical
- Horizontal: 24px (mobile) / 48px (tablet) / 96px (desktop)

Component Spacing:

- Between major elements: 80-120px
- Between related elements: 40-60px
- Between text elements: 16-24px
- Inline spacing: 8-16px

Grid/Layout:

- Max content width: 1280px (7xl)
- Text content max-width: 800px (comfortable reading)
- Generous gutters: 40-80px between columns

Borders & Shadows

Borders:

- Default thickness: 1px
- Emphasis thickness: 2-4px
- Color: Lavender grey ( #778da9) at 30-50% opacity
- Style: Solid, no rounded corners (sharp, minimalist)

Shadows (Subtle):

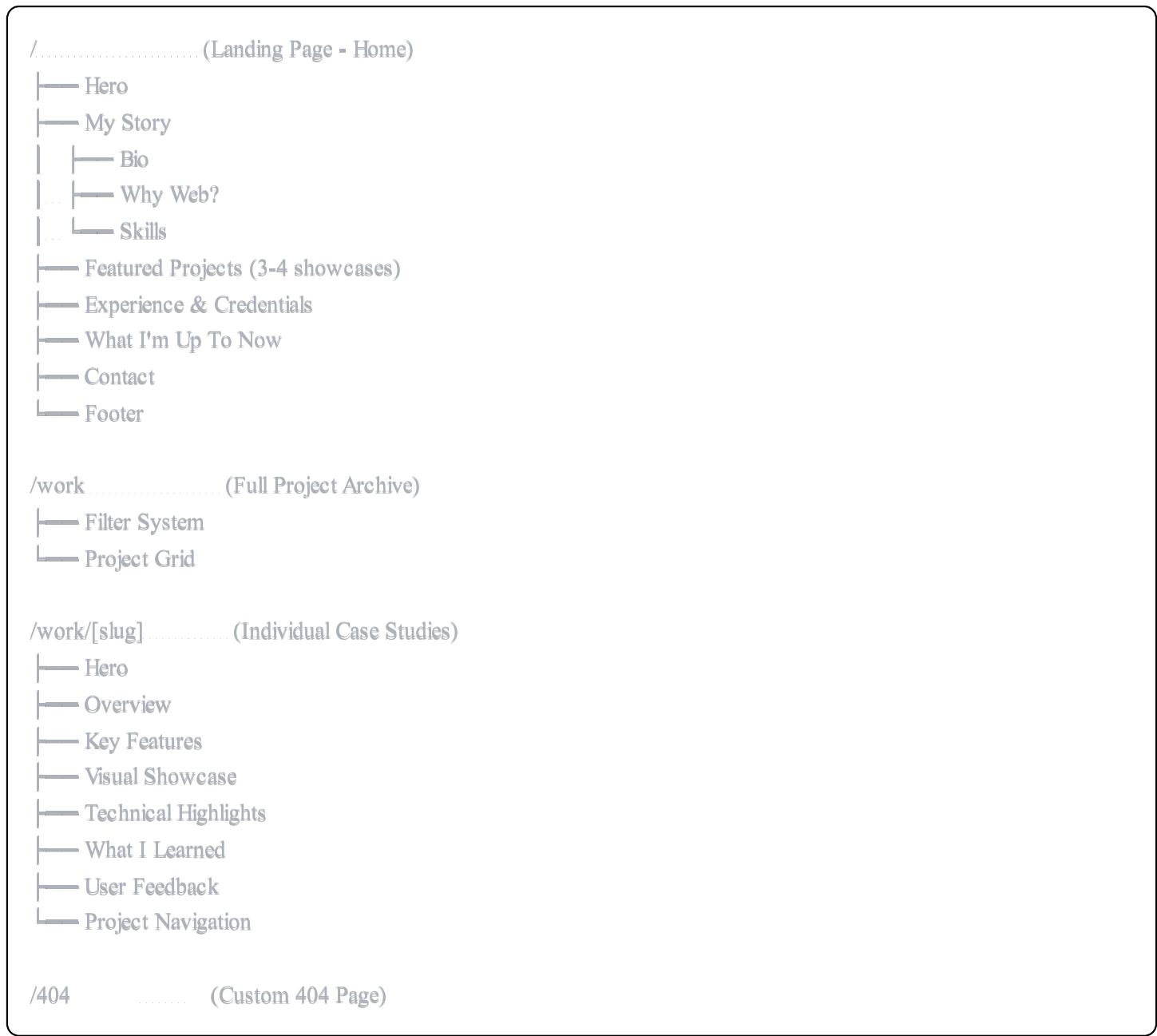
```
css

/* Hover states */
box-shadow: 0 10px 30px rgba(119, 141, 169, 0.15);

/* Lifted elements */
box-shadow: 0 20px 40px rgba(119, 141, 169, 0.2);
```

Page Structure

Site Map



Navigation Structure

Desktop Side Navigation:

- Fixed position: left side of screen
- Text-based labels (vertical alignment)
- Active indicator: vertical line that grows
- Links: Home, My Story, Work, Experience, Contact

Mobile Horizontal Navigation:

- Fixed to top of screen
 - Horizontal scrollable bar
 - Active indicator: underline that slides
 - Always accessible (no hamburger menu overlay)
-

Component Specifications

1. Loading Screen

Purpose: Initial site visit only (first load)

Animation Sequence (1.2 seconds total):

1. Three horizontal lines fade in (0.2s)
2. Lines grow left to right sequentially (0.5s)
3. Brief pause (0.1s)
4. Lines morph into checkmark + underline (0.3s)
5. Fade out (0.2s)

Visual Specs:

- Background: Ink black at 95% opacity
- Lines: Alabaster grey
- Line caps: Rounded
- Line thickness: 3px
- Optional: Subtle glow effect on lines

Implementation:

```
jsx
```

```
// Use Framer Motion for SVG path animations
<motion.svg>
  <motion.line
    initial={ { pathLength: 0 } }
    animate={ { pathLength: 1 } }
    transition={ { duration: 0.4, delay: 0.2 } }
  />
  // ... additional lines and morphing
</motion.svg>
```

2. Hero Section

Layout:

- Full viewport height
- Centered content
- Generous whitespace
- Scroll indicator at bottom

Content:

[Your Name - Large]
[Ample vertical space]
I craft web experiences with intention,
precision, and a focus on what matters.
[More vertical space]
[Scroll indicator - animated mouse]

Background Element - Breathing Smile:

- Position: Offset to upper-right quadrant
- Opacity: 10-15%
- Elements:
 - Two squares (eyes): 40-60px, alabaster grey
 - One arc (mouth): Animated, alabaster grey
- Animation loop (continuous):

1. Straight line (😐) - 2s hold
2. Curves down (😞) - 1s transition
3. Pause at frown - 0.5s
4. Curves up to smile (😊) - 1s transition
5. Hold smile + breathing effect - 3-4s
6. Return to straight - 1s transition
7. Repeat

Breathing Effect:

- Scale: $1.0 \rightarrow 1.02 \rightarrow 1.0$
- Duration: 3 seconds
- Easing: Sine wave (organic feel)

Scroll Indicator:

- Mouse icon with scroll wheel
- Wheel animates: bounce up/down
- Text: "Scroll" above icon
- Color: Lavender grey

Parallax:

- Entire hero fades out as user scrolls down
 - Opacity: 1 at top \rightarrow 0 at 300px scroll
-

3. My Story Section

Three Subsections with Visual Separation:

A. Bio Subsection

Background: Pure ink black

Layout:

- Desktop: Two columns (30% / 70%)

- Mobile: Single column, stacked

Left Column (Desktop):

- Geometric shape: Square or circle
- Size: 192px × 192px
- Style: Border 2px lavender grey at 30% opacity
- Optional: Subtle pattern inside or abstract element

Right Column:

- Heading: "About Me"
- 3 paragraphs of bio text
- Conversational tone
- Line height: 1.6-1.8

Mobile:

- Shape centered above text
 - Text flows naturally below
-

B. Why Web? Subsection

Background: Ink black with 2% alabaster overlay (subtle distinction)

Layout:

- Single column, centered content
- Max-width: 800px

Background Element:

- Animated code snippet
- Position: Offset (upper-right or lower-left)
- Opacity: 8-10%
- Style: Syntax highlighted using color palette
- Animation: Lines slowly type out or scroll through different snippets

Example Code (Meta/Playful):

```
javascript

function buildTheFuture() {
  ..return passion + code;
}

const impact = skills
..filter(s => s.useful)
..map(s => s.apply());
```

Content:

- Heading: "Why Web?"
 - Narrative format (2-3 paragraphs)
 - Personal but professional tone
-

C. Skills Subsection

Background: Pure ink black (alternates back)

Layout:

- Centered constellation diagram
- Filter buttons above diagram

Skills Constellation Diagram:

Node Design:

- Shape: **Hexagons** (not circles)
- Core skills: Larger nodes (60-80px), center position
- Supporting skills: Smaller nodes (40-50px), orbit around core
- Tech stack logos: Inside nodes (small, grayscale or monochrome)

Connections:

- Lines connect related skills (e.g., HTML-CSS, React-Next.js)

- Color: Lavender grey
- Thickness: 2px
- Skills used together are linked

Animation:

- Nodes fade in one by one on scroll
- Connection lines draw after nodes appear
- Smooth, staggered timing

Interaction - Hover:

- Hovered node scales to 1.1x
- Connected nodes highlight (alabaster grey)
- Unconnected nodes fade to 40% opacity
- Connection lines to hovered node thicken slightly

Filter System:

Categories:

1. All
2. Frontend
3. Backend
4. Tools & Workflow
5. AI-Assisted Development

Filter Pills:

- Style: Transparent bg, lavender border
- Active: Lavender bg, alabaster text
- Hover: Border brightens
- Transition: 200ms

Implementation:

- Filter buttons trigger state change

- Filtered nodes fade out/in smoothly
- Connections update accordingly

Skills to Include:

- **Core:** HTML, CSS, JavaScript, React, Next.js
 - **Backend:** Node.js, Django, Python, PostgreSQL
 - **Tools:** Git, GitHub, VS Code, Vite
 - **AI Tools:** GitHub Copilot, Claude Sonnet, Google AI Studio, Cursor
-

4. Featured Projects Section

Layout: Vertical stack of hero-style showcases

Each Project Card (70-80vh height):

Structure:

- Large floating device mockup (laptop at 15-20° angle)
- Project details beside or below mockup
- Entire card clickable → case study page

Mockup:

- Contains: Screenshot or looping GIF of key interaction
- Shadow: Soft drop shadow for depth
- Animation: Gentle floating (2-3px vertical movement over 4s)

Project Details:

- **Title:** Large, alabaster
- **Description:** 2-3 sentences max
- **Tech Stack:** Small pills (transparent bg, alabaster border, lavender text)
- **Key Metric:** Optional (e.g., "50% faster load times")
- **CTA:** "View Case Study →" link (alabaster, underline on hover)

Layout Pattern (Creates Rhythm):

- Project 1: Mockup left, text right
- Project 2: Mockup right, text left
- Project 3: Mockup left, text right
- Alternates throughout

Hover State:

- Mockup lifts: `translateY(-10px)`
- Shadow intensifies
- Duration: 200ms

Scroll Transition Between Projects:

- Current project: Scales to 0.95, fades out
- Next project: Fades in
- Smooth, cinematic feel

Mobile:

- Stack vertically: Mockup on top, text below
 - Mockup scales appropriately
 - Maintain generous spacing
-

5. Experience & Credentials Section

Layout: Combined section with two subsections

Background: Alternates (pure ink black OR 2% alabaster overlay)

Desktop: Side by side (Experience left, Certificates right) **Mobile:** Stacked (Experience first, Certificates below)

A. Experience/Positions

Format: Timeline or card-based

Timeline Structure (If 3+ positions):

- Vertical line: Thin, lavender grey
- Nodes: Small hexagons (ties to skills constellation)
- Details to right of each node

Card Structure (If 1-2 positions):

- Company name (medium size)
 - Role title (slightly smaller)
 - Date range (small, lavender grey)
 - 2-3 brief bullet points of achievements
-

B. Certificates

Format: Grid of clickable cards

Layout:

- Desktop: 2 columns
- Mobile: 1 column

Each Certificate Card:

- Border: Subtle lavender grey
- Background: Ink black (with hover state)
- Content:
 - Certificate name (medium)
 - Issuer logo (small, grayscale) OR text issuer name
 - Date earned (small, lavender grey)
 - Small external link icon (↗)
- **Entire card is clickable** (wrapped in `<a>` tag)
 - `href="certificate-link"` opens in new tab
 - `target="_blank" rel="noopener noreferrer"`

Hover State:

- Lifts: `translateY(-5px)`
 - Border brightens
 - Background lightens 2-3%
 - Transition: 200ms
-

6. What I'm Up To Now Section

Layout: Split focus (two columns)

Desktop:



Content:

- **Left:** Active project(s) with brief description
- **Right:** Technologies/skills being explored + why

Styling:

- Column headers: Medium size, lavender grey
- Content: Regular alabaster at 80%
- Clean text hierarchy
- No additional visual elements (minimal)

Mobile:

- Stack vertically
 - "Currently Building" first
 - "Currently Learning" below
-

7. Contact Section

Layout: Centered, max-width 600px

Form Fields:

- Name (text input)
- Email (email input)
- Message (textarea, 6-8 rows)
- Submit button

Envelope Animation:

Initial State:

- Open envelope illustration (outline in lavender grey)
- Form fields appear "inside" envelope
- Envelope flap open at top

On Submit (Desktop):

1. Form validation (if invalid: shake + error states)
2. If valid:
 - Form fields fade out (300ms)
 - Submit button shows loading state
 - Envelope flap closes (rotates from top, 600ms)
 - Brief pause (200ms)
 - Closed envelope slides out to right with upward arc (800ms)
 - Success message fades in: "Message sent! I'll get back to you soon."

Visual Details:

- Envelope: Outlined in lavender, filled with ink black
- Flap: CSS transform rotate from top edge
- Slide-out: `translateX(100vw) translateY(-50px)`
- Success text: Alabaster with optional checkmark icon

On Submit (Mobile - Simplified):

- Form slides out to right

- Success message slides in from left
- No envelope animation (too complex for small screens)

Implementation:

- **Framer Motion** for all animations
- **Web3Forms** for form submission
- Proper validation with helpful error messages

Form Styling:

- Fields: Border lavender grey (inactive) → alabaster (focus)
- Focus state: Subtle glow effect
- Error state: Red border + error text below
- Labels: Above fields or floating labels

Additional Elements:

- Social links: GitHub, LinkedIn (below form)
 - Style: Simple text links with hover underline
-

8. Footer

Layout: Minimal, single or two-line

Content Structure:

[GitHub] [LinkedIn] | © 2025 | [Download Resume]

Alternative Layout:

Built with Next.js, React & Framer Motion
Designed & developed by [Your Name]

[GitHub] [LinkedIn] [Download Resume]

Styling:

- Small text (14px)
- Lavender grey color
- Links hover → alabaster
- Center-aligned or split into sections
- Padding: 32px vertical

Colophon Text: "Built with Next.js, React & Framer Motion"

Resume Button:

- Style: Secondary variant (transparent, border)
 - Icon: Download symbol
 - Hover: Lift effect
-

9. Side Navigation (Desktop)

Position:

- Fixed: left side of viewport
- Distance from edge: 32px (2rem)
- Vertical center: `(top: 50%; transform: translateY(-50%);)`

Items:

- Home
- My Story
- Work
- Experience
- Contact

Styling:

- Text-based labels
- Font size: 14px
- Spacing: 24px between items

- Default color: Lavender grey
- Active color: Alabaster
- Hover: Alabaster

Active Indicator:

- Vertical line to left of active item
 - Color: Alabaster
 - Width: 2px
 - Height: Matches text height
 - Animation: Grows from 0 to full height (300ms)
 - Uses `layoutId` for smooth transitions between items
-

10. Mobile Navigation

Position:

- Fixed: top of viewport
- Width: Full screen
- Background: Ink black at 95% with backdrop blur
- Border bottom: Lavender grey at 20% opacity

Layout:

- Horizontal scrollable list
- Padding: 16px horizontal, 16px vertical
- Items have adequate spacing: 32px between

Items: Same as desktop nav

Styling:

- Text: 14px
- Default: Lavender grey
- Active: Alabaster

- Whitespace: nowrap (prevent wrapping)

Active Indicator:

- Horizontal underline beneath active item
- Color: Alabaster
- Height: 2px
- Animation: Slides to active item (300ms)
- Uses `layoutId` for smooth transitions

Scroll Behavior:

- Momentum scrolling
 - Snap to items (optional)
 - Subtle fade/gradient at edges to indicate more content
-

Animation System

Motion Design Language: "Option B - Layered Depth"

Philosophy:

- Smooth, subtle, purposeful
 - Creates sense of depth without being excessive
 - GPU-accelerated (transform + opacity only)
 - Respects `prefers-reduced-motion`
-

Page Transitions

Between Routes:

1. Outgoing page: Scales to 0.95 + fades out
2. Incoming page: Fades in from opacity 0
3. Duration: 400-500ms
4. Easing: Cubic bezier `[0.22, 1, 0.36, 1]`

```
jsx
```

```
// Framer Motion variants
const pageTransition = {
  initial: { opacity: 0, scale: 0.98 },
  animate: { opacity: 1, scale: 1 },
  exit: { opacity: 0, scale: 0.95 },
  transition: { duration: 0.4, ease: [0.22, 1, 0.36, 1] }
};
```

Scroll Animations

Element Entry (Fade + Drift):

- Elements start: `(opacity: 0, y: 20px, filter: blur(4px))`
- Elements end: `(opacity: 1, y: 0, filter: blur(0px))`
- Duration: 600ms
- Trigger: When element enters viewport (IntersectionObserver)

Parallax Effect:

- Background/images move slower than foreground content
- Speed ratio: 0.3-0.5x
- Very subtle, not exaggerated
- Applied to project mockups, background elements

Blur-to-Sharp:

- Text starts with `filter: blur(4px)`
 - Animates to `(blur(0px))` over 400ms
 - Combined with fade-in for elegant reveal
-

Micro-Interactions (Locked In)

1. Link Underline Slide-In

Trigger: Hover Animation:

- Underline grows from left to right
- Uses `::after` pseudo-element
- Width: 0 → 100%
- Duration: 200ms
- Color: Alabaster

css

```
a::after {  
  content: '';  
  position: absolute;  
  width: 0;  
  height: 1px;  
  bottom: -2px;  
  left: 0;  
  background-color: var(--alabaster);  
  transition: width 0.2s ease;  
}  
  
a:hover::after {  
  width: 100%;  
}
```

2. Button Lift on Hover

Trigger: Hover Animation:

- Translates up: `translateY(-2px)`
- Shadow intensifies slightly
- Border brightens (if outlined)
- Duration: 200ms

jsx

```
<motion.button  
.. whileHover={{ y: -2 }}  
.. transition={{ duration: 0.2 }}  
/>
```

3. Form Field Focus Glow

Trigger: Focus Animation:

- Border color: Lavender → Alabaster
- Subtle glow/shadow appears
- Label shifts color (if present)
- Duration: 200ms smooth

css

```
input:focus {  
  border-color: var(--alabaster);  
  box-shadow: 0 0 0 3px rgba(224, 225, 221, 0.1);  
  transition: all 0.2s ease;  
}
```

4. Active Nav Indicator

Desktop:

- Vertical line grows from 0 to full height
- Color: Alabaster
- Duration: 300ms
- Spring animation (Framer Motion `(layoutId)`)

Mobile:

- Horizontal underline slides beneath active item
- Same color and duration
- Spring animation

Framer Motion Presets

Standard Fade In:

```
jsx

const fadeIn = {
  hidden: { opacity: 0 },
  visible: { opacity: 1, transition: { duration: 0.6 } }
};
```

Fade In Up:

```
jsx

const fadeInUp = {
  hidden: { opacity: 0, y: 20, filter: 'blur(4px)' },
  visible: {
    opacity: 1,
    y: 0,
    filter: 'blur(0px)',
    transition: { duration: 0.6, ease: [0.22, 1, 0.36, 1] }
  }
};
```

Stagger Container:

```
jsx

const staggerContainer = {
  hidden: { opacity: 0 },
  visible: {
    opacity: 1,
    transition: { staggerChildren: 0.2 }
  }
};
```

Scale Hover:

```
jsx

whileHover={{ scale: 1.05 }}
transition={{ duration: 0.2 }}
```

Performance Optimization

GPU Acceleration:

- Only animate `transform` and `opacity`
- Avoid animating `width`, `height`, `left`, `top`
- Use `will-change` sparingly

Reduced Motion:

```
css
```

```
@media (prefers-reduced-motion: reduce) {  
  *,  
  *::before,  
  *::after {  
    animation-duration: 0.01ms !important;  
    animation-iteration-count: 1 !important;  
    transition-duration: 0.01ms !important;  
  }  
}
```

Lazy Loading:

- Use `IntersectionObserver` for scroll-triggered animations
 - Don't animate elements not in viewport
 - Debounce scroll listeners
-

/work Page - Full Project Archive

Purpose

Show all projects in a browsable, filterable grid

Layout

Header:

- Page title: "All Projects" or "My Work"
- Optional intro line: "A collection of web experiences I've built"

Filter System:

Two-Tier Filtering:

Primary Filters (Project Type):

- All
- Full-Stack
- Web
- Side Projects

Secondary Filters (Tech Stack):

- All
- React.js
- Next.js
- Vite.js
- TypeScript
- Django
- (Add all technologies you've used)

Filter Layout:

Primary: [All] [Full-Stack] [Web] [Side Projects]

Tech Stack: [All] [React] [Next.js] [Vite] [TypeScript] [Django]...

..... [Project Grid Below]

Filter Logic:

- User selects ONE primary filter
- User can select MULTIPLE tech stack filters
- Results show projects matching:

- Selected primary category AND
- ANY of selected tech filters (OR logic)

Example:

- Selected: "Full-Stack" + "TypeScript" + "React"
- Shows: All Full-Stack projects using TypeScript OR React (or both)

Visual Implementation:

- Pills: Transparent bg, lavender border (inactive)
- Active: Lavender bg, alabaster text
- Hover: Border brightens
- Transition: 200ms

Optional Results Counter:

- "Showing 8 projects"
 - Small text below filters
 - Updates as filters change
-

Project Grid

Layout:

- Desktop: 2 columns
- Mobile: 1 column
- Gap: 40-60px
- Masonry or equal height cards

Each Project Card:

- Thumbnail image (standardized aspect ratio: 16:10)
- Project title (medium size)
- One-sentence tagline
- Tech stack tags (small, subtle)

- Entire card is clickable → case study

Thumbnail:

- Grayscale by default (optional)
- Full color on hover
- Transition: Saturation 400ms

Hover State:

- Image scales: 1.05x
- Overlay appears with "View Case Study →"
- Slight shadow increase
- Duration: 300ms

Filter Animation:

- Cards fade out (200ms)
 - Grid rearranges
 - Matching cards fade in (200ms)
 - Smooth, not jarring
-

Case Study Pages ([\(/work/\[slug\]\)](#))

Purpose

Tell the complete story of each project in a 2-minute read

Page Structure

1. Hero Section

- Full-width project screenshot
- Project title (large)
- One-sentence tagline

- Tech stack tags
- Action buttons: [Live Demo] [View Code]

2. Quick Overview (4-5 lines max)

- Problem: [One sentence]
- Solution: [One sentence]
- Role: [Your role]
- Timeline: [Duration]

3. Key Features (3-4 bullet points with icons)

- Icon + Title + One sentence description
- Scannable format

4. Visual Showcase

- Embedded video (live demo recording)
 - YouTube or Vimeo embed
 - Responsive wrapper (16:9 aspect ratio)
- 2-3 screenshots with captions
- One diagram (architecture or user flow)
 - Clean, minimal vector style
 - Uses color palette (lavender lines, alabaster labels)

5. Technical Highlights (Brief paragraph or 2-3 bullets)

- Interesting technical decisions
- Challenges solved
- GitHub integration: See implementation →

6. "What I Learned" Box

- Conversational, personal tone
- 2-3 sentences max
- Styled distinctly:

- Left border: 4px lavender grey
- Slight indent
- Subtle background tint (2-3% alabaster overlay)
- Optional italic text

7. User Feedback (if available)

- Single testimonial quote
- Attribution: Name, Role
- Styled as quote card

8. Project Navigation

- Bottom of page
- Previous/Next project cards with thumbnails

[← Previous Project] [Next Project →]
 [Thumbnail + Title] [Thumbnail + Title]

Layout Guidelines

Content Width:

- Text: Max-width 800px, centered
- Images/Videos: Full-width or contained based on context
- Generous padding: 80-120px vertical between sections

Responsive:

- Desktop: Two-column layouts where appropriate (Problem | Solution)
- Mobile: Everything single column

Animations:

- Very subtle fade-in for sections (not paragraphs)
- Don't distract from reading

Tone:

- Conversational and personal (like "What I Learned")
 - Professional but approachable
 - Show personality
-

Case Study Template

markdown

[Project Title]

[One-sentence tagline]

[Hero Image - Full Width]

[Live Demo Button] [GitHub Button]

Overview

Problem: [One sentence]

Solution: [One sentence]

Role: [Your role]

Timeline: [Duration]

Key Features

⌚ **Feature 1:** Description

⚡ **Feature 2:** Description

🔧 **Feature 3:** Description

[Embedded Video Demo]

Visual Deep Dive

[Screenshot 1 with caption]

[Screenshot 2 with caption]

[Architecture/Flow Diagram]

Technical Highlights

[Brief paragraph about interesting technical decisions]

[See the implementation on GitHub →]

What I Learned

> [Your personal, conversational reflection - 2-3 sentences]

User Feedback

"[Testimonial quote or use placeholder]"

— [Client/User Name, Role]

[← Previous Project] [Back to All Projects] [Next Project →]

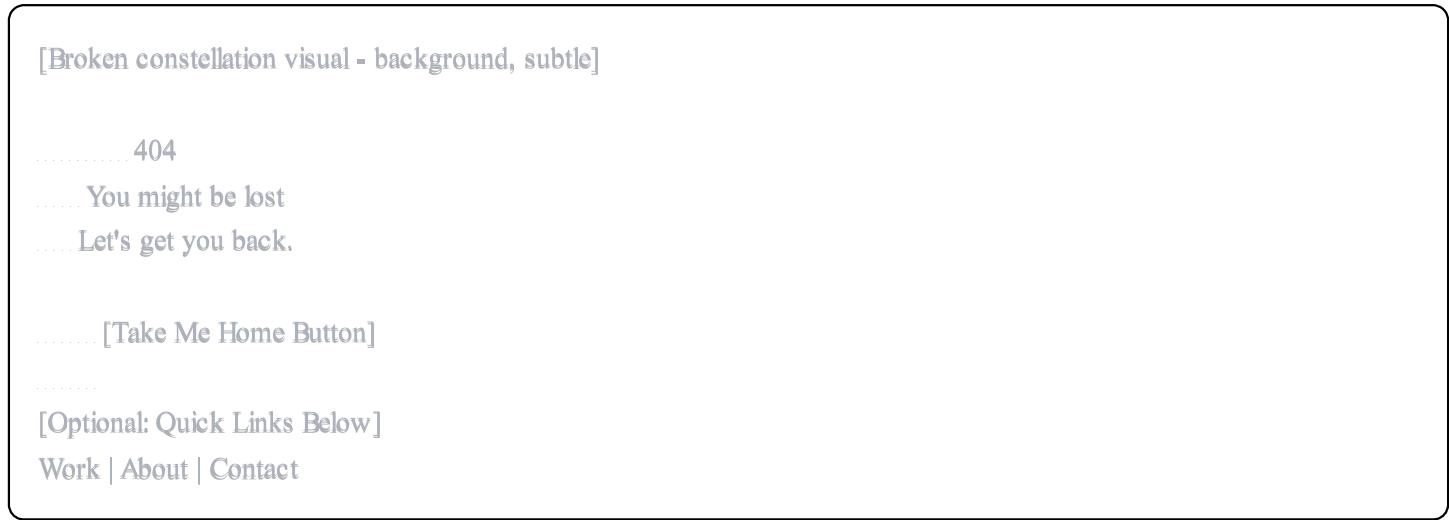
404 Page

Purpose

Custom error page with helpful navigation and personality

Layout

Centered Content:



Broken Constellation Animation

Visual:

- 5-7 hexagon nodes scattered across page
- Connection lines between some nodes (lavender grey)
- Background element, 15-20% opacity
- Positioned offset (not perfectly centered)

Animation (Calm, Melancholic):

1. Start with nodes connected
2. Connection lines fade out one by one (sequential)
3. Each line takes 2-3 seconds to disconnect
4. Nodes dim slightly as they lose connections
5. After all disconnect, nodes gently pulse/breathe

6. Very slow pace, calming
7. No reconnection (they remain disconnected)
8. Breathing effect: 2-3% scale change over 3-4 seconds

Visual Details:

- Nodes: Hexagons (ties to skills constellation)
 - Size: 40-60px
 - Color: Alabaster at 15-20% opacity
 - Lines: Lavender at 15-20% opacity
 - Smooth, organic movement
-

Content

Typography:

- "404": Very large (5-6rem), alabaster
- "You might be lost": Medium size (1.5rem)
- "Let's get you back.": Slightly smaller, lavender grey

Button:

- "Take Me Home"
- Primary style: Border, hover lift
- Center-aligned

Quick Links (Optional):

- Small text links below button
 - Work | About | Contact
 - Separated by vertical bars
 - Subtle, helpful navigation
-

Responsive Design

Breakpoints

```
css  
/* Mobile First Approach */  
/* Small Mobile: 320px - 640px */  
/* Mobile: 640px - 768px */  
/* Tablet: 768px - 1024px */  
/* Desktop: 1024px - 1280px */  
/* Large Desktop: 1280px+ */
```

Tailwind Breakpoints:

- `[sm]`: 640px
 - `[md]`: 768px
 - `[lg]`: 1024px
 - `[xl]`: 1280px
 - `[2xl]`: 1536px
-

Mobile Optimizations

Navigation:

- Side nav hidden on mobile
- Horizontal scroll nav appears at top
- Mobile nav has adequate touch targets (44px min)

Hero:

- Text size reduces proportionally
- Smile animation scales down or simplified
- Generous vertical spacing maintained

My Story:

- Two-column layouts become single column

- Geometric shapes center above text
- Code snippet background may be hidden or simplified

Skills Constellation:

- May need to scale down
- Touch interactions for hover states
- Filter pills may wrap or become scrollable

Projects:

- Floating mockups stack vertically
- Mockup comes first, then details
- Simplified animations (no complex parallax)

Contact Form:

- Envelope animation simplified (form slides, no envelope)
- Full-width fields for easy input
- Adequate spacing between fields

Typography:

- Font sizes scale down proportionally
 - Line height may increase slightly for readability
 - Maintain hierarchy through size/weight
-

Touch Interactions

Tap Targets:

- Minimum 44px × 44px for all interactive elements
- Adequate spacing between clickable items (16px min)

Hover States:

- Convert to active/tap states

- Provide visual feedback on touch
- Use `:active` pseudo-class

Gestures:

- Swipe for horizontal scrolling (mobile nav)
 - Pinch-to-zoom disabled for fixed layouts
 - Native scroll behavior maintained
-

Performance on Mobile

Image Optimization:

- Use Next.js `<Image>` component
- Responsive images with `srcset`
- WebP format with fallbacks
- Lazy loading for below-fold images

Animation Performance:

- Reduce complexity on lower-end devices
- Use `will-change` sparingly
- Test on actual devices, not just emulators

Bundle Size:

- Code splitting for routes
 - Lazy load heavy components (constellation, form)
 - Minimize JavaScript where possible
-

Implementation Guide

Project Setup

1. Create Next.js Project

```
bash
```

```
# Navigate to your project folder  
cd your-project-folder  
  
# Initialize Next.js with TypeScript and Tailwind  
npx create-next-app@latest . --typescript --tailwind --app --no-src-dir  
  
# Prompts:  
# ✓ TypeScript? Yes  
# ✓ ESLint? Yes  
# ✓ Tailwind CSS? Yes  
# ✓ `src/` directory? No  
# ✓ App Router? Yes  
# ✓ Import alias? No
```

2. Install Dependencies

```
bash
```

```
# Core animation library  
npm install framer-motion  
  
# Form handling  
npm install @web3forms/react  
  
# Optional: If using Resend for emails instead  
npm install resend
```

Project Structure

```
your-portfolio/  
  └── app/  
    ├── layout.tsx ..... # Root layout  
    ├── page.tsx       # Landing page (Home)  
    ├── globals.css..... # Global styles  
    └── work/  
      └── page.tsx     # Projects archive  
        └── [slug]/  
          └── page.tsx   # Individual case study  
    └── not-found.tsx  # 404 page
```

```
... └── favicon.ico
... └── components/
...   └── navigation/
...     ├── SideNav.tsx      # Desktop side navigation
...     └── MobileNav.tsx    # Mobile horizontal nav
...   └── sections/
...     ├── Hero.tsx        # Hero with smile
...     ├── Bio.tsx         # Bio section
...     ├── WhyWeb.tsx      # Why Web section
...     ├── Skills.tsx       # Skills constellation
...     ├── Projects.tsx     # Featured projects
...     ├── Experience.tsx   # Experience & Credentials
...     ├── WhatImUpTo.tsx   # Current activities
...     └── Contact.tsx      # Contact form
... └── ui/
...   ├── Button.tsx       # Reusable button
...   ├── Card.tsx         # Card component
...   └── Section.tsx      # Section wrapper
... └── animations/
...   ├── BreathingSmile.tsx # Hero smile animation
...   ├── LoadingScreen.tsx  # Initial loading
...   └── EnvelopeForm.tsx   # Contact form animation
... └── Footer.tsx        # Site footer
└── lib/
  ├── animations.ts      # Framer Motion variants
  ├── constants.ts        # Colors, spacing, etc.
  └── utils.ts            # Helper functions
└── public/
  ├── projects/           # Project images
  ├── certificates/       # Certificate images
  └── resume.pdf          # Downloadable resume
└── data/
  ├── projects.ts          # Project data
  ├── skills.ts            # Skills data
  └── experience.ts        # Work history data
└── tailwind.config.ts    # Tailwind configuration
└── tsconfig.json          # TypeScript config
└── next.config.js         # Next.js config
└── package.json
```

Key Files Configuration

tailwind.config.ts

```
typescript

import type { Config } from "tailwindcss";

const config: Config = {
  content: [
    ... "./pages/**/*.{js,ts,jsx,tsx,mdx}",
    ... "./components/**/*.{js,ts,jsx,tsx,mdx}",
    ... "./app/**/*.{js,ts,jsx,tsx,mdx}",
  ],
  theme: {
    extend: {
      colors: {
        'ink-black': '#0d1b2a',
        'alabaster': '#e0e1dd',
        'lavender': '#778da9',
      },
      fontFamily: {
        sans: ['Rubik', 'sans-serif'],
      },
      spacing: {
        'section': '200px',
        'section-mobile': '80px',
      },
    },
  },
  plugins: [],
};

export default config;
```

app/globals.css

```
css
```

```
@import url('https://fonts.googleapis.com/css2?family=Rubik:wght@400&display=swap');

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

:root {
  --ink-black: #0d1b2a;
  --alabaster: #e0e1dd;
  --lavender: #778da9;
}

* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

html {
  scroll-behavior: smooth;
}

body {
  font-family: 'Rubik', sans-serif;
  font-weight: 400;
  background-color: var(--ink-black);
  color: var(--alabaster);
  overflow-x: hidden;
  line-height: 1.6;
}

/* Custom Scrollbar */
::-webkit-scrollbar {
  width: 8px;
}

::-webkit-scrollbar-track {
  background: var(--ink-black);
}

::-webkit-scrollbar-thumb {
  background: var(--lavender);
  border-radius: 4px;
```

```
}

::webkit-scrollbar-thumb:hover {
  background: var(--alabaster);
}

/* Hide scrollbar for mobile nav */
.scrollbar-hide::-webkit-scrollbar {
  display: none;
}

.scrollbar-hide {
  -ms-overflow-style: none;
  scrollbar-width: none;
}

/* Link underline animation */
a {
  position: relative;
  text-decoration: none;
}

a::after {
  content: "";
  position: absolute;
  width: 0;
  height: 1px;
  bottom: -2px;
  left: 0;
  background-color: var(--alabaster);
  transition: width 0.2s ease;
}

a:hover::after {
  width: 100%;
}

/* Remove underline for nav links and buttons */
nav a::after,
button a::after {
  display: none;
}

/* Reduced motion support */
```

```
@media (prefers-reduced-motion: reduce) {  
  *,  
  *::before,  
  *::after {  
    animation-duration: 0.01ms !important;  
    animation-iteration-count: 1 !important;  
    transition-duration: 0.01ms !important;  
    scroll-behavior: auto !important;  
  }  
}  
}
```

/ Focus styles for accessibility */*

```
*:focus-visible {  
  outline: 2px solid var(--alabaster);  
  outline-offset: 2px;  
}
```

/ Selection styles */*

```
::selection {  
  background-color: var(--lavender);  
  color: var(--ink-black);  
}
```

lib/animations.ts

typescript

```
// Framer Motion animation variants
```

```
export const fadeIn = {  
  hidden: { opacity: 0 },  
  visible: {  
    opacity: 1,  
    transition: { duration: 0.6 }  
  }  
};
```

```
export const fadeInUp = {  
  hidden: {  
    opacity: 0,  
    y: 20,  
    filter: 'blur(4px)'  
  },  
  visible: {  
    opacity: 1,  
    y: 0,  
    filter: 'blur(0px)',  
    transition: {  
      duration: 0.6,  
      ease: [0.22, 1, 0.36, 1]  
    }  
  }  
};
```

```
export const fadeInDown = {  
  hidden: {  
    opacity: 0,  
    y: -20  
  },  
  visible: {  
    opacity: 1,  
    y: 0,  
    transition: { duration: 0.6 }  
  }  
};
```

```
export const scaleIn = {  
  hidden: {  
    opacity: 0,  
    scale: 0.95  
  }  
};
```

```
...},  
...visible: {  
  opacity: 1,  
  ...scale: 1,  
  ...transition: { duration: 0.4 }  
}  
};
```

```
export const staggerContainer = {  
  hidden: { opacity: 0 },  
  visible: {  
    opacity: 1,  
    ...transition: {  
      staggerChildren: 0.2,  
      delayChildren: 0.1  
    }  
  }  
};
```

```
export const slideInLeft = {  
  hidden: {  
    opacity: 0,  
    x: -30  
  },  
  visible: {  
    opacity: 1,  
    x: 0,  
    transition: { duration: 0.5 }  
  }  
};
```

```
export const slideInRight = {  
  hidden: {  
    opacity: 0,  
    x: 30  
  },  
  visible: {  
    opacity: 1,  
    x: 0,  
    transition: { duration: 0.5 }  
  }  
};
```

```
export const pageTransition = {
```

```
...initial: {
  ...opacity: 0,
  scale: 0.98
},
...animate: {
  opacity: 1,
  scale: 1
},
exit: {
  ...opacity: 0,
  ...scale: 0.95
},
transition: {
  duration: 0.4,
  ease: [0.22, 1, 0.36, 1]
}
};

// Hover animations
export const hoverLift = {
  y: -2,
  transition: { duration: 0.2 }
};

export const hoverScale = {
  scale: 1.05,
  transition: { duration: 0.2 }
};
```

lib/constants.ts

typescript

```
// Design system constants
```

```
export const colors = {  
  inkBlack: '#0d1b2a',  
  alabaster: '#e0e1dd',  
  lavender: '#778da9',  
};
```

```
export const spacing = {  
  sectionDesktop: '200px',  
  sectionMobile: '80px',  
  componentGap: '80px',  
  elementGap: '40px',  
  textGap: '24px',  
};
```

```
export const breakpoints = {  
  sm: '640px',  
  md: '768px',  
  lg: '1024px',  
  xl: '1280px',  
  '2xl': '1536px',  
};
```

```
export const typography = {  
  hero: '4rem', // 64px  
  h1: '3rem', // 48px  
  h2: '2.5rem', // 40px  
  h3: '2rem', // 32px  
  bodyLarge: '1.25rem', // 20px  
  body: '1rem', // 16px  
  small: '0.875rem', // 14px  
};
```

Web3Forms Integration

Contact Form Implementation

```
typescript
```

```
// components/sections/Contact.tsx

import { useForm } from 'react-hook-form';
import { motion } from 'framer-motion';
import { useState } from 'react';

interface FormData {
  name: string;
  email: string;
  message: string;
}

export default function Contact() {
  const [isSubmitting, setIsSubmitting] = useState(false);
  const [.isSuccess, setIsSuccess] = useState(false);

  const { register, handleSubmit, formState: { errors } } = useForm<FormData>();

  const onSubmit = async (data: FormData) => {
    setIsSubmitting(true);

    try {
      const response = await fetch('https://api.web3forms.com/submit', {
        method: 'POST',
        headers: {
          'Content-Type': 'application/json',
        },
        body: JSON.stringify({
          access_key: 'YOUR_WEB3FORMS_ACCESS_KEY', // Get this from web3forms.com
          name: data.name,
          email: data.email,
          message: data.message,
        }),
      });
    });

    if (response.ok) {
      setIsSuccess(true);
      // Trigger envelope animation here
    }
  } catch (error) {
    console.error('Error submitting form:', error);
  } finally {
    setIsSubmitting(false);
  }
}
```

```
....}
....};

...if (isSuccess) {
...return (
  <motion.div
    initial={{ opacity: 0 }}
    animate={{ opacity: 1 }}
    className="text-center py-20"
  >
    <h3 className="text-2xl mb-4">Message sent!</h3>
    <p className="text-lavender">I'll get back to you soon.</p>
  </motion.div>
);

}

return (
<form onSubmit={handleSubmit(onSubmit)} className="max-w-xl mx-auto">
  /* Form fields implementation */
  <div className="mb-6">
    <label htmlFor="name" className="block mb-2">Name</label>
    <input
      id="name"
      type="text"
      {...register('name', { required: 'Name is required' })}
      className="w-full px-4 py-3 bg-transparent border border-lavender/30
        focus:border-alabaster transition-colors rounded-sm"
    />
    {errors.name && (
      <span className="text-red-400 text-sm">{errors.name.message}</span>
    )}
  </div>

  /* Email and Message fields similar structure */

  <button
    type="submit"
    disabled={isSubmitting}
    className="w-full py-3 border border-alabaster hover:translate-y-0.5
      transition-all duration-200"
  >
    {isSubmitting ? 'Sending...' : 'Send Message'}
  </button>
</form>
```

```
...);  
}
```

Getting Web3Forms Access Key:

1. Go to <https://web3forms.com>
2. Sign up for free account
3. Create new form
4. Copy your access key
5. Add to your environment variables or directly in code (for testing)

Environment Variable (Recommended):

```
bash  
# .env.local  
NEXT_PUBLIC_WEB3FORMS_KEY=your_access_key_here
```

Then use: `(process.env.NEXT_PUBLIC_WEB3FORMS_KEY)`

Data Structure Examples

`data/projects.ts`

```
typescript
```

```
export interface Project {
  id: string;
  slug: string;
  title: string;
  tagline: string;
  description: string;
  thumbnail: string;
  category: 'full-stack' | 'web' | 'side-project';
  techStack: string[];
  featured: boolean;
  liveUrl?: string;
  githubUrl?: string;
  mockupImage?: string;
}

export const projects: Project[] = [
  {
    id: '1',
    slug: 'ecommerce-platform',
    title: 'E-Commerce Platform',
    tagline: 'Modern shopping experience with real-time inventory',
    description: 'Full-featured e-commerce platform built with Next.js...',
    thumbnail: '/projects/ecommerce-thumb.jpg',
    category: 'full-stack',
    techStack: ['Next.js', 'TypeScript', 'PostgreSQL', 'Stripe'],
    featured: true,
    liveUrl: 'https://example.com',
    githubUrl: 'https://github.com/yourusername/project',
    mockupImage: '/projects/ecommerce-mockup.png',
  },
  // Add more projects...
];


```

data/skills.ts

```
typescript
```

```

export interface Skill {
  id: string;
  name: string;
  category: 'frontend' | 'backend' | 'tools' | 'ai';
  logo?: string;
  connections: string[]; // IDs of connected skills
  core: boolean; // Is this a core skill?
}

}

```

```

export const skills: Skill[] = [
  {
    id: 'react',
    name: 'React',
    category: 'frontend',
    logo: '/logos/react.svg',
    connections: ['nextjs', 'typescript', 'tailwind'],
    core: true,
  },
  {
    id: 'nextjs',
    name: 'Next.js',
    category: 'frontend',
    logo: '/logos/nextjs.svg',
    connections: ['react', 'typescript'],
    core: true,
  },
  // Add more skills...
];

```

Development Checklist

Phase 1: Foundation (Week 1)

- Set up Next.js project with TypeScript and Tailwind
- Install dependencies (Framer Motion, Web3Forms)
- Configure design system (colors, typography, spacing)
- Set up project structure (folders, components)
- Create reusable components (Button, Card, Section)
- Implement navigation (desktop side nav, mobile horizontal)
- Build loading screen animation

- Test responsive layout basics

Phase 2: Landing Page - Core Sections (Week 2)

- Hero section with breathing smile animation
- Bio section with geometric shape
- Why Web section with animated code snippet background
- Skills constellation diagram
- Hexagon nodes
- Connection lines
- Hover interactions
- Filter system
- Footer with colophon and links

Phase 3: Landing Page - Projects & Contact (Week 3)

- Featured projects section
- Floating mockup components
- Alternating layouts
- Hover and scroll animations
- Experience & Credentials section
- Timeline or cards for experience
- Clickable certificate cards
- What I'm Up To Now section
- Contact form with envelope animation
- Form validation
- Web3Forms integration
- Desktop envelope animation
- Mobile simplified animation
- Success state

Phase 4: Projects Archive & Case Studies (Week 4)

- `/work` page structure
- Two-tier filter system
- Primary filters (project type)
- Secondary filters (tech stack)
- Filter logic implementation
- Project grid with cards
- Individual case study template

- Hero section
- Overview
- Key features
- Visual showcase (screenshots, video embeds, diagrams)
- Technical highlights with GitHub links
- "What I Learned" box
- User feedback section
- Project navigation
- Create case studies for 3-5 projects

Phase 5: 404 & Polish (Week 5)

- Custom 404 page
- Broken constellation animation
- Helpful copy and navigation
- Micro-interaction polish
- Link underlines
- Button hovers
- Form focus states
- Nav indicators
- Performance optimization
- Image optimization
- Code splitting
- Lazy loading
- Animation performance
- Accessibility audit
- Keyboard navigation
- Screen reader testing
- Color contrast verification
- Reduced motion support
- ARIA labels

Phase 6: Content & Testing (Week 6)

- Write all copy
- Bio
- Why Web narrative
- Project descriptions
- Case study content

- Gather assets
- Project screenshots
- Mockup images
- Certificate images/links
- Logo assets
- Create diagrams for case studies
- Record demo videos
- Add resume PDF
- Cross-browser testing (Chrome, Firefox, Safari, Edge)
- Mobile device testing (iOS, Android)
- Performance testing (Lighthouse)

Phase 7: Deployment (Week 7)

- Set up domain (if needed)
- Configure environment variables
- Deploy to Vercel (or hosting of choice)
- Set up analytics (optional)
- Test production build
- Set up contact form email notifications
- Final QA pass
- Launch!

Post-Launch

- Monitor Web3Forms submissions
 - Gather user feedback
 - Add user testimonials to case studies
 - Regular updates to "What I'm Up To Now"
 - Add new projects as completed
 - Performance monitoring
 - SEO optimization
-

Accessibility Guidelines

Keyboard Navigation

- All interactive elements must be keyboard accessible
- Logical tab order throughout the site

- Skip links for screen readers
- Focus indicators visible and styled

Screen Readers

- Proper heading hierarchy (h1 → h2 → h3, no skipping)
- Alt text for all images (descriptive, concise)
- ARIA labels for icon-only buttons
- ARIA live regions for dynamic content (form submissions)

Color Contrast

- Verify all text meets WCAG AA standards (4.5:1 for normal text)
- Alabaster on ink black: High contrast ✓
- Lavender on ink black: Check and adjust if needed
- Don't rely on color alone to convey information

Reduced Motion

- Respect `[prefers-reduced-motion]` media query
- Disable or reduce animations for users who prefer it
- Essential animations (loading) can stay but simplified

Forms

- Clear labels for all form fields
 - Error messages associated with fields
 - Helpful validation messages
 - Success confirmations
-

Performance Optimization

Images

- Use Next.js `(<Image>)` component everywhere

- Provide `width` and `height` attributes
- Use WebP format with fallbacks
- Lazy load images below the fold
- Optimize file sizes (aim for <200KB per image)

Fonts

- Preload critical fonts (Rubik)
- Use `font-display: swap` to prevent FOIT
- Consider subset fonts if only using Latin characters

JavaScript

- Code split by route
- Lazy load heavy components (constellation, contact form)
- Minimize client-side JavaScript where possible
- Use React Server Components where appropriate

CSS

- Inline critical CSS
- Remove unused Tailwind classes (automatic with Tailwind)
- Minimize animation complexity
- Use CSS transforms over position changes

Animations

- Only animate `transform` and `opacity` (GPU-accelerated)
- Use `will-change` sparingly (only during animation)
- Debounce scroll listeners
- Use `IntersectionObserver` for scroll-triggered animations

Lighthouse Targets

- Performance: 90+

- Accessibility: 100
 - Best Practices: 95+
 - SEO: 95+
-

SEO Recommendations

Meta Tags

```
tsx

// app/layout.tsx
export const metadata = {
  title: 'Your Name - Full-Stack Web Developer',
  description: 'Portfolio showcasing web development projects, skills, and experience in building modern web applications.',
  keywords: ['web developer', 'full-stack', 'React', 'Next.js', 'portfolio'],
  authors: [{ name: 'Your Name' }],
  openGraph: {
    title: 'Your Name - Full-Stack Web Developer',
    description: 'Portfolio showcasing web development projects and skills',
    url: 'https://yourwebsite.com',
    siteName: 'Your Name Portfolio',
    images: [
      {
        url: 'https://yourwebsite.com/og-image.jpg',
        width: 1200,
        height: 630,
      },
    ],
    locale: 'en_US',
    type: 'website',
  },
  twitter: {
    card: 'summary_large_image',
    title: 'Your Name - Full-Stack Web Developer',
    description: 'Portfolio showcasing web development projects',
    images: ['https://yourwebsite.com/og-image.jpg'],
  },
};
```

- Add JSON-LD for Person schema
- Add JSON-LD for CreativeWork for projects
- Include relevant microdata

URLs

- Clean, descriptive URLs (`/work/e-commerce-platform`) not (`/project/1`)
- Include keywords where natural
- Use hyphens, not underscores

Content

- Unique, descriptive page titles
 - Meta descriptions for each page (155-160 characters)
 - Heading hierarchy maintained
 - Alt text for images includes keywords naturally
-

Browser Support

Target Browsers

- Chrome (last 2 versions)
- Firefox (last 2 versions)
- Safari (last 2 versions)
- Edge (last 2 versions)
- Mobile Safari (iOS 14+)
- Chrome Mobile (Android 10+)

Fallbacks

- CSS Grid with flexbox fallback
- WebP images with JPG/PNG fallback
- Modern JavaScript features (use Next.js transpilation)

Maintenance & Updates

Regular Updates

- Add new projects as completed
- Update "What I'm Up To Now" monthly
- Refresh testimonials as received
- Keep dependencies up to date
- Review and optimize performance quarterly

Content Strategy

- Blog integration (optional future addition)
 - Case study deep dives
 - Update resume regularly
 - Gather metrics from projects
-

Future Enhancements (Phase 2)

Potential Additions

- Blog section for technical writing
 - Dark/Light mode toggle (though current design is dark-only)
 - Project filtering by year
 - Testimonials carousel
 - GitHub contribution graph integration
 - More interactive elements (games, experiments)
 - Analytics dashboard integration
 - Newsletter signup
 - Search functionality
-

Assets Needed

Images

- Headshot or geometric placeholder (Bio section)
- Project mockups (3-5 floating mockups for landing page)
- Project thumbnails (standardized 16:10 ratio for /work page)
- Project screenshots (2-3 per case study)
- Certificate images or logos
- Company/employer logos (if applicable)
- Tech stack logos (for skills constellation)
- Open Graph image (1200×630px)
- Favicon (multiple sizes)

Videos

- Demo videos for key projects (embedded in case studies)
- Optional: Video introduction on landing page

Documents

- Resume PDF (downloadable)
- Certificate PDFs or links

Diagrams

- Architecture diagrams for case studies (vector format)
 - User flow diagrams
 - Wireframes (if showing process)
-

Deployment Checklist

Pre-Deployment

- All content reviewed and proofread
- All links tested (internal and external)
- Images optimized and compressed
- Forms tested (contact, Web3Forms working)
- SEO meta tags added
- Favicon added

- 404 page working
- Performance audit passed
- Accessibility audit passed
- Mobile responsive on all pages
- Cross-browser tested

Deployment (Vercel Recommended)

- Connect GitHub repository to Vercel
- Configure environment variables
- Set up custom domain (if applicable)
- Enable analytics
- Deploy to production
- Test production site thoroughly
- Set up monitoring/alerts

Post-Deployment

- Submit sitemap to Google Search Console
 - Test contact form submissions
 - Monitor performance (Core Web Vitals)
 - Set up backup/version control procedures
 - Share portfolio link!
-

Contact & Support

Resources

- **Next.js Docs:** <https://nextjs.org/docs>
- **Framer Motion Docs:** <https://www.framer.com/motion/>
- **Tailwind CSS Docs:** <https://tailwindcss.com/docs>
- **Web3Forms:** <https://web3forms.com>
- **React Hook Form:** <https://react-hook-form.com>

Community

- Next.js Discord
- React Discord

- Stack Overflow
 - Dev.to community
-

Final Notes

This documentation represents a complete, production-ready design system for a minimalist portfolio website. Every section, component, animation, and interaction has been carefully specified.

Key Principles to Remember:

1. **Minimalism with purpose** - Every element should have a reason
2. **Performance first** - Fast, smooth, optimized
3. **Accessibility always** - Inclusive design for all users
4. **Show, don't tell** - Demonstrate skills through the site itself
5. **Breathing room** - Generous spacing creates elegance
6. **Dark mode only** - Ink black dominates throughout
7. **Subtle motion** - Animations enhance, never distract
8. **Conversational tone** - Professional but approachable

Development Timeline:

- Realistic timeline: 6-7 weeks for complete implementation
- Can be accelerated by focusing on essentials first
- Iterative approach: Launch with core features, enhance over time

Success Metrics:

- Fast load times (< 2 seconds)
 - High Lighthouse scores (90+ performance)
 - Positive user feedback
 - Increased project inquiries/opportunities
 - Portfolio showcases your actual skills
-

Additional Technical Specifications

Next.js Configuration

next.config.js

```
javascript

/** @type {import('next').NextConfig} */
const nextConfig = {
  images: {
    formats: ['image/webp', 'image/avif'],
    deviceSizes: [640, 750, 828, 1080, 1200, 1920, 2048, 3840],
    imageSizes: [16, 32, 48, 64, 96, 128, 256, 384],
  },
  // Enable React strict mode for better error detection
  reactStrictMode: true,
  // Optimize for static export if needed
  // output: 'export', // Uncomment for static export
};

module.exports = nextConfig;
```

TypeScript Types

types/index.ts

```
typescript
```

```
// Project Types
export interface Project {
  id: string;
  slug: string;
  title: string;
  tagline: string;
  description: string;
  longDescription?: string;
  thumbnail: string;
  heroImage?: string;
  mockupImage?: string;
  category: 'full-stack' | 'web' | 'side-project';
  techStack: string[];
  featured: boolean;
  liveUrl?: string;
  githubUrl?: string;
  videoUrl?: string;
  completedDate: string;
  duration?: string;
  role?: string;
  keyFeatures?: string[];
  challenges?: string[];
  solutions?: string[];
  whatILearned?: string;
  userFeedback?: {
    quote: string;
    author: string;
    role: string;
  };
  metrics?: {
    label: string;
    value: string;
  }[];
  screenshots?: string[];
  diagrams?: string[];
}
```

```
// Skill Types
export interface Skill {
  id: string;
  name: string;
  category: 'frontend' | 'backend' | 'tools' | 'ai';
  logo?: string;
```

```
..connections: string[];  
..core: boolean;  
proficiency?: 'beginner' | 'intermediate' | 'advanced' | 'expert';  
}
```

```
// Experience Types  
export interface Experience {  
..id: string;  
..company: string;  
..role: string;  
..startDate: string;  
endDate?: string; // Undefined means current  
..description: string;  
..achievements?: string[];  
..technologies?: string[];  
}
```

```
// Certificate Types  
export interface Certificate {  
..id: string;  
..name: string;  
..issuer: string;  
..date: string;  
..credentialUrl?: string;  
..logo?: string;  
}
```

```
// Navigation Types  
export interface NavItem {  
..id: string;  
..label: string;  
..href?: string;  
}
```

```
// Filter Types  
export interface FilterOption {  
..id: string;  
..label: string;  
..value: string;  
}
```

Environment Variables

.env.local (Create this file, add to .gitignore)

```
bash

# Web3Forms
NEXT_PUBLIC_WEB3FORMS_KEY=your_web3forms_access_key_here

# Optional: Resend (if using instead of Web3Forms)
RESEND_API_KEY=your_resend_api_key_here

# Optional: Analytics
NEXT_PUBLIC_GA_MEASUREMENT_ID=your_google_analytics_id

# Site URL (for metadata)
NEXT_PUBLIC_SITE_URL=https://yourportfolio.com
```

.env.example (Commit this for reference)

```
bash

# Web3Forms - Get your key at https://web3forms.com
NEXT_PUBLIC_WEB3FORMS_KEY=

# Optional: Resend - Get your key at https://resend.com
RESEND_API_KEY=

# Optional: Google Analytics
NEXT_PUBLIC_GA_MEASUREMENT_ID=

# Site URL
NEXT_PUBLIC_SITE_URL=
```

Git Configuration

.gitignore (Ensure these are included)

```
# dependencies
/node_modules
/.pnp
.pnp.js

# testing
/coverage

# next.js
/.next/
/out/

# production
/build

# misc
.DS_Store
*.pem

# debug
npm-debug.log*
yarn-debug.log*
yarn-error.log*

# local env files
.env*.local
.env

# vercel
.vercel

# typescript
*.tsbuildinfo
next-env.d.ts
```

Component Implementation Examples

Button Component ([components/ui/Button.tsx](#))

```
typescript
```

```
import { motion } from 'framer-motion';
import { ButtonHTMLAttributes, ReactNode } from 'react';

interface ButtonProps extends ButtonHTMLAttributes<HTMLButtonElement> {
  children: ReactNode;
  variant?: 'primary' | 'secondary' | 'ghost';
  size?: 'sm' | 'md' | 'lg';
  fullWidth?: boolean;
}

export default function Button({
  children,
  variant = 'primary',
  size = 'md',
  fullWidth = false,
  className = '',
  ...props
}: ButtonProps) {
  const baseStyles = "font-normal transition-all duration-200 rounded-sm inline-flex items-center justify-center";

  const variants = {
    primary: "bg-transparent border border-alabaster text-alabaster hover:-translate-y-0.5 hover:shadow-lg hover:shadow-lavender",
    secondary: "text-lavender hover:text-alabaster relative group",
    ghost: "text-alabaster hover:bg-alabaster/10"
  };

  const sizes = {
    sm: "px-4 py-2 text-sm",
    md: "px-6 py-3 text-base",
    lg: "px-8 py-4 text-lg"
  };

  const widthClass = fullWidth ? "w-full" : "";

  return (
    <motion.button
      className={`${baseStyles} ${variants[variant]} ${sizes[size]} ${widthClass} ${className}`}
      whileHover={{ y: -2 }}
      whileTap={{ y: 0 }}
      {...props}
    >
      {children}
      {variant === 'secondary' && (
        <div
          style={{ position: 'absolute', left: 0, top: 0, width: '100%', height: '100%', background: 'linear-gradient(45deg, transparent, black 50%)' }>
        </div>
      )}
    
  );
}
```

```
.....<span className="absolute bottom-0 left-0 w-0 h-0.5 bg-alabaster group-hover:w-full transition-all duration-200" />
.....)}
</motion.button>
...);
}
```

Section Wrapper ([components/ui/Section.tsx](#))

typescript

```
import { ReactNode } from 'react';
import { motion } from 'framer-motion';

interface SectionProps {
  children: ReactNode;
  id?: string;
  className?: string;
  background?: 'default' | 'alternate';
  noPadding?: boolean;
}

export default function Section({
  children,
  id,
  className = '',
  background = 'default',
  noPadding = false
}: SectionProps) {
  const bgClass = background === 'alternate'
    ? 'bg-ink-black bg-opacity-[0.98]'
    : 'bg-ink-black';

  const paddingClass = noPadding
    ? ''
    : 'py-32 md:py-section px-6 md:px-12';

  return (
    <section
      id={id}
      className={`${bgClass} ${paddingClass} ${className}`}
    >
      <div className="max-w-7xl mx-auto">
        {children}
      </div>
    </section>
  );
}
```

Card Component ([components/ui/Card.tsx](#))

typescript

```
import { ReactNode } from 'react';
import { motion } from 'framer-motion';

interface CardProps {
  children: ReactNode;
  className?: string;
  hoverable?: boolean;
  onClick?: () => void;
}

export default function Card({
  children,
  className = '',
  hoverable = false,
  onClick
}: CardProps) {
  const hoverProps = hoverable ? {
    whileHover: { y: -5, transition: { duration: 0.2 } },
    whileTap: { y: 0 }
  } : {};

  return (
    <motion.div
      className={'border border-lavender/30 bg-ink-black p-6 rounded-sm ${hoverable ? "cursor-pointer" : ""} ${className}`}
      onClick={onClick}
      {...hoverProps}
    >
      {children}
    </motion.div>
  );
}
```

Utility Functions

lib/utils.ts

typescript

```
import { type ClassValue, clsx } from 'clsx';
import { twMerge } from 'tailwind-merge';

// Merge Tailwind classes safely
export function cn(...inputs: ClassValue[]): string {
  return twMerge(clsx(inputs));
}

// Format date to readable string
export function formatDate(dateString: string): string {
  const date = new Date(dateString);
  return date.toLocaleDateString('en-US', {
    year: 'numeric',
    month: 'long'
  });
}

// Get reading time estimate
export function getReadingTime(text: string): number {
  const wordsPerMinute = 200;
  const wordCount = text.split(/\s+/).length;
  return Math.ceil(wordCount / wordsPerMinute);
}

// Debounce function for scroll listeners
export function debounce<T extends (...args: any[]) => any>(
  func: T,
  wait: number
): (...args: Parameters<T>) => void {
  let timeout: NodeJS.Timeout;
  return (...args: Parameters<T>) => {
    clearTimeout(timeout);
    timeout = setTimeout(() => func(...args), wait);
  };
}

// Filter projects by category and tech
export function filterProjects(
  projects: any[],
  categoryFilter: string,
  techFilters: string[]
): any[] {
  return projects.filter(project => {

```

```

....// Category filter
....const matchesCategory =
  categoryFilter === 'all' ||
  project.category === categoryFilter;

// Tech filter (OR logic)
const matchesTech =
  techFilters.length === 0 ||
  techFilters.includes('all') ||
  project.techStack.some((tech: string) =>
  techFilters.includes(tech.toLowerCase())
);

return matchesCategory && matchesTech;
});

}

// Generate slug from title
export function slugify(text: string): string {
  return text
    .toLowerCase()
    .replace(/[^w\s-]/g, "-")
    .replace(/\s+/g, '-')
    .replace(/--+|-/g, '-')
    .trim();
}

```

SEO Component

components/SEO.tsx

typescript

```
import Head from 'next/head';

interface SEOProps {
  title: string;
  description: string;
  image?: string;
  url?: string;
  type?: 'website' | 'article';
}

export default function SEO({
  title,
  description,
  image = '/og-image.jpg',
  url,
  type = 'website'
}: SEOProps) {
  const siteUrl = process.env.NEXT_PUBLIC_SITE_URL || '';
  const fullUrl = url ? `${siteUrl}${url}` : siteUrl;
  const fullImageUrl = `${siteUrl}${image}`;

  return (
    <Head>
      <title>{title}</title>
      <meta name="description" content={description} />

      {/* Open Graph */}
      <meta property="og:type" content={type} />
      <meta property="og:title" content={title} />
      <meta property="og:description" content={description} />
      <meta property="og:image" content={fullImageUrl} />
      <meta property="og:url" content={fullUrl} />

      {/* Twitter */}
      <meta name="twitter:card" content="summary_large_image" />
      <meta name="twitter:title" content={title} />
      <meta name="twitter:description" content={description} />
      <meta name="twitter:image" content={fullImageUrl} />

      {/* Canonical URL */}
      <link rel="canonical" href={fullUrl} />
    </Head>
  );
}
```

```
...);  
}  
}
```

Analytics Integration (Optional)

lib/analytics.ts

```
typescript  
// Google Analytics helper functions  
export const pageview = (url: string) => {  
    if (typeof window !== 'undefined' && (window as any).gtag) {  
        (window as any).gtag('config', process.env.NEXT_PUBLIC_GA_MEASUREMENT_ID, {  
            page_path: url,  
        });  
    }  
};  
  
export const event = ({ action, category, label, value }: {  
    action: string;  
    category: string;  
    label: string;  
    value?: number;  
}) => {  
    if (typeof window !== 'undefined' && (window as any).gtag) {  
        (window as any).gtag('event', action, {  
            event_category: category,  
            event_label: label,  
            value: value,  
        });  
    }  
};
```

Usage in app/layout.tsx

```
typescript
```

```
'use client';

import { usePathname } from 'next/navigation';
import { useEffect } from 'react';
import { pageview } from '@/lib/analytics';

export default function AnalyticsWrapper({ children }) {
  const pathname = usePathname();

  useEffect(() => {
    if (pathname) {
      pageview(pathname);
    }
  }, [pathname]);

  return <>{children}</>;
}
```

Testing Recommendations

Unit Testing (Optional)

```
bash

npm install --save-dev @testing-library/react @testing-library/jest-dom jest jest-environment-javascript
```

E2E Testing (Optional)

```
bash

npm install --save-dev @playwright/test
```

Visual Regression Testing (Optional)

- Percy.io
 - Chromatic
 - BackstopJS
-

Deployment Configurations

Vercel (Recommended)

`vercel.json` (Optional configuration)

```
json

{
  "framework": "nextjs",
  "buildCommand": "next build",
  "devCommand": "next dev",
  "installCommand": "npm install",
  "headers": [
    {
      "source": "/(.*)",
      "headers": [
        {
          "key": "X-Content-Type-Options",
          "value": "nosniff"
        },
        {
          "key": "X-Frame-Options",
          "value": "DENY"
        },
        {
          "key": "X-XSS-Protection",
          "value": "1; mode=block"
        }
      ]
    }
  ]
}
```

Deployment Steps:

1. Push code to GitHub
2. Import project in Vercel
3. Configure environment variables
4. Deploy
5. Add custom domain (optional)

Netlify (Alternative)

netlify.toml

```
toml

[[build]]
  command = "npm run build"
  publish = ".next"

[[plugins]]
  package = "@netlify/plugin-nextjs"
```

Performance Monitoring

Web Vitals Tracking

```
typescript

// app/layout.tsx
import { Analytics } from '@vercel/analytics/react';

export default function RootLayout({ children }) {
  return (
    <html lang="en">
      <body>
        {children}
        <Analytics />
      </body>
    </html>
  );
}
```

Custom Web Vitals Reporting

```
typescript
```

```
// app/web-vitals.tsx
'use client';

import { useReportWebVitals } from 'next/web-vitals';

export function WebVitals() {
  useReportWebVitals((metric) => {
    console.log(metric);
    // Send to analytics service
  });
}
```

Security Best Practices

1. Environment Variables

- Never commit `.env.local`
- Use `NEXT_PUBLIC_` prefix only for client-side vars
- Rotate API keys regularly

2. Dependencies

- Run `npm audit` regularly
- Keep dependencies updated
- Use `npm audit fix` for vulnerabilities

3. Content Security

- Sanitize user inputs (contact form)
- Validate data before processing
- Use HTTPS everywhere

4. Headers

- Set security headers (X-Frame-Options, etc.)
- Configure CORS appropriately
- Use Content Security Policy (CSP)

Backup Strategy

1. Code

- GitHub repository (primary backup)
- Regular commits with meaningful messages
- Tag releases

2. Content

- Keep content in data files (versioned)
- Export form submissions regularly
- Backup environment variables securely

3. Assets

- Keep original high-res versions
 - Use cloud storage (Google Drive, Dropbox)
 - Version control for text-based assets
-

Common Issues & Solutions

Issue: Framer Motion SSR Errors

Solution: Use dynamic imports with `ssr: false`

```
typescript
const AnimatedComponent = dynamic(
  () => import('./AnimatedComponent'),
  { ssr: false }
);
```

Issue: Hydration Mismatch

Solution: Ensure server and client render the same content initially

```
typescript
```

```
const [mounted, setMounted] = useState(false);

useEffect(() => {
  setMounted(true);
}, []);

if (!mounted) return null;
```

Issue: Large Bundle Size

Solution:

- Lazy load heavy components
- Use dynamic imports
- Optimize images
- Tree-shake unused code

Issue: Slow Animations on Mobile

Solution:

- Reduce animation complexity
- Use `transform` and `opacity` only
- Add `will-change` during animation
- Test on actual devices

Resources & Learning

Official Documentation

- [Next.js Docs](#)
- [React Docs](#)
- [Framer Motion](#)
- [Tailwind CSS](#)
- [TypeScript](#)

Design Inspiration

- [Awwwards](#)
- [Behance](#)
- [Dribbble](#)
- [SiteInspire](#)

Tools

- [Figma](#) - Design tool
- [Excalidraw](#) - Diagrams
- [Colors](#) - Color palettes
- [Google Fonts](#) - Typography

Communities

- [Next.js Discord](#)
 - [Reactiflux Discord](#)
 - [Dev.to](#)
 - [Stack Overflow](#)
-

Conclusion

This comprehensive documentation provides everything needed to build a professional, performant, and accessible portfolio website. The design system is cohesive, the technical stack is modern, and the user experience is thoughtfully crafted.

Remember:

- Start with the essentials, iterate and improve
- Test on real devices and browsers
- Gather feedback and adjust
- Keep content fresh and updated
- Showcase your personality alongside your skills

The goal is not perfection on day one, but a portfolio that:

- Represents your skills authentically
- Provides a great user experience
- Performs exceptionally well
- Stands out through thoughtful design
- Grows and evolves with you

You have all the specifications, guidelines, and code examples needed to build this portfolio. Now it's time to bring it to life!

Good luck with your development! 

Appendix: Quick Reference

Color Variables

```
css  
  
--ink-black: #0d1b2a  
--alabaster: #e0e1dd  
--lavender: #778da9
```

Spacing Scale

```
Section: 200px (desktop) / 80px (mobile)  
Component: 80px  
Element: 40px  
Text: 24px  
Inline: 8-16px
```

Typography Scale

```
Hero: 64px / 48px  
H1: 48px / 40px  
H2: 40px / 32px  
H3: 32px / 28px  
Body Large: 20px
```

Body: 16px
Small: 14px

Animation Durations

Micro: 200ms
Standard: 400ms
Slow: 600ms
Very Slow: 800ms-1000ms

Key Commands

```
bash

# Development
npm run dev

# Build
npm run build

# Start production
npm start

# Lint
npm run lint

# Type check
npm run type-check
```

End of Documentation

Last Updated: January 2026 Version: 1.0