

NarcoNations Brand Components – Standardisation Guide

(v1.0 – September 2025)

Purpose

To enforce consistency across all NarcoNations projects (.com, .org, Kickstarter builds, future apps), the following **core components** are standardised: **Logo, Button, NavBar, Footer**.

These must **always reference global tokens** (`globals.css` + `tailwind.config.js`) and be imported from a single `brand/` folder.

1. Logo Component

File: `/components/brand/Logo.tsx`

Rules

- Always sourced from `/public/images/Flavicon.jpg` or master SVG.
- Variants:
 - `full` → Mask emblem + “NARCO NATIONS” stacked wordmark.
 - `icon` → Mask emblem only (favicon or tight layouts).
- Never recolor outside brand palette.
- Always on black/charcoal background.

Example API

```
<BrandLogo variant="full" className="h-10" />
<BrandLogo variant="icon" className="h-8 w-8" />
```

2. Button Component

File: `/components/brand/Button.tsx`

Rules

- Base button class = `.btn` from `globals.css`.
- Variants:
 - `primary` → Blood Red background, white text (`btn-primary`).
 - `ghost` → Transparent, white border (`btn-ghost`).
 - `gold` → Gold background, black text (prestige actions).

- Sizes: `sm`, `md`, `lg` (mapped to Tailwind padding/font sizes).

Example API

```
<BrandButton variant="primary" size="md">Join the Cartel</BrandButton>
<BrandButton variant="ghost" size="sm">Learn More</BrandButton>
<BrandButton variant="gold" size="lg">Boss Edition</BrandButton>
```

3. NavBar Component

File: `/components/brand/NavBar.tsx`

Rules

- Always dark background (`bg-ink` or `bg-charcoal`).
- Left: `<BrandLogo variant="full" />`.
- Right: CTA button (`<BrandButton variant="primary" />`).
- Responsive: collapses into a mobile drawer with black background + red highlights.
- Sticky mode available for landing pages.

Example API

```
<BrandNav sticky={true} />
```

4. Footer Component

File: `/components/brand/FixedFooter.tsx`

Rules

- Background always black.
- White text, small Inter font.
- Includes CTA slot + 18+ badge.
- Supports optional `showCTA` prop to toggle final signup button.

Example API

```
<BrandFooter showCTA={true} />
```

Implementation Notes

- Place all four files in `/components/brand/`.

- Import across projects like:

```
import { BrandLogo, BrandButton, BrandNav, BrandFooter } from "@components/brand"
```

- All styling must pull from tokens in `tailwind.config.js` and `globals.css` (colors, fonts, spacing, animations).
- No hardcoded hex codes or font stacks allowed in these components.

✓ Following this guide ensures **consistent brand DNA** across .com, .org, Kickstarter, and all future NarcoNations properties.

5. Iconography (Lucide Icons)

Library: `lucide-react`

Rules

- **Stroke:** Always Lucide default (2px) for visual consistency.
- **Color Tokens:**
 - White (`text-paper`) → default on dark backgrounds.
 - Blood Red (`text-blood`) → alerts, FED actions.
 - Heat Cyan (`text-heat`) → Dark Web/tech cues.
 - Gold (`text-gold`) → prestige/wealth/NarcoCoins.
- **Sizes:**
 - `sm` = `w-4 h-4` (inline UI)
 - `md` = `w-6 h-6` (nav/footer default)
 - `lg` = `w-10 h-10` (hero/feature)
- **Placement:** Align to baseline with adjacent text; keep even spacing (Tailwind `gap-4`).

Example (Footer Social)

```
import { Instagram, Twitter, Youtube } from "lucide-react"

<div className="flex gap-4 items-center">
  <Instagram className="w-6 h-6 text-paper hover:text-blood transition" />
  <Twitter    className="w-6 h-6 text-paper hover:text-blood transition" />
  <Youtube    className="w-6 h-6 text-paper hover:text-blood transition" />
</div>
```

6. Animated Icon Patterns (Cinematic, Subtle)

Animations should be *tasteful* and respect `prefers-reduced-motion`. Use them for affordance and emphasis—not decoration spam.

****Add these utilities to **` (beneath your existing animation section):**

```
@keyframes uv-pulse { 0%,100%{ filter: drop-shadow(0 0 0 rgba(0,255,231,.0)); } 50%{ filter: drop-shadow(0 0 12px rgba(0,255,231,.35)); } }
@keyframes blood-flicker { 0%,100%{ opacity:1 } 92%{ opacity:.65 } 96%{ opacity:.9 } }
@keyframes hover-wiggle { 0%{ transform: translateY(0) rotate(0deg) } 50%{ transform: translateY(-1px) rotate(-1deg) } 100%{ transform: translateY(0) rotate(0deg) } }
@keyframes draw-stroke { to { stroke-dashoffset: 0; } }

/* Utilities */
.uv-pulse { animation: uv-pulse 3s ease-in-out infinite; }
.blood-flicker { animation: blood-flicker 1.8s steps(2,end) infinite; }
.hover-wiggle:hover { animation: hover-wiggle .6s ease-in-out; }

/* Respect motion settings */
@media (prefers-reduced-motion: reduce){
  .uv-pulse, .blood-flicker, .hover-wiggle { animation: none !important; }
}
```

Usage Patterns

• Footer Social (ambient UV pulse)

```
import { Instagram, Twitter, Youtube } from "lucide-react"

<div className="flex gap-4 items-center">
  <Instagram className="w-6 h-6 text-paper hover:text-blood transition uv-pulse" />
  <Twitter className="w-6 h-6 text-paper hover:text-blood transition uv-pulse" />
  <Youtube className="w-6 h-6 text-paper hover:text-blood transition uv-pulse" />
</div>
```

• CTA Adjacency (hover wiggle only)

```
import { ArrowRight } from "lucide-react"

<button className="btn-primary group">
  Join the Cartel
  <ArrowRight className="ml-2 w-5 h-5 text-paper hover-wiggle" />
</button>
```

• Dark Web/Digital Cue (heat cyan glow)

```
import { Cpu } from "lucide-react"

<Cpu className="w-6 h-6 text-heat uv-pulse" />
```

Do / Don't

- **Do** keep loops slow ($\geq 3s$) and subtle; add hover accents for interactivity.
- **Do** honor `prefers-reduced-motion`.
- **Don't** combine multiple looping effects on the same icon.
- **Don't** use animation for purely decorative icons in dense text.

7. BrandIcon Helper (Optional, Recommended)

Centralises size + variant mapping for Lucide imports.

```
// /components/brand/Icon.tsx
import * as Icons from "lucide-react"
import clsx from "clsx"

type Variant = "paper" | "blood" | "heat" | "gold"
const variantClass: Record<Variant, string> = {
  paper: "text-paper",
  blood: "text-blood",
  heat: "text-heat",
  gold: "text-gold",
}
const sizeClass = { sm: "w-4 h-4", md: "w-6 h-6", lg: "w-10 h-10" }

export function BrandIcon({ name, variant = "paper", size = "md",
  className }: {
  name: keyof typeof Icons; variant?: Variant; size?: keyof typeof
  sizeClass; className?: string
}){
  const Icon = Icons[name]
  if (!Icon) return null
  return <Icon className={clsx(sizeClass[size], variantClass[variant],
    "transition", className)} />
}
```

Example:

```
<BrandIcon name="Instagram" variant="paper" size="md" className="uv-pulse" />
```

8. Tagline & Provenance Lockup

Tagline: *Blood Money: Crime vs Karma*


Secondary: *Global Organised Crime. The Game.*

Provenance: *Made in Manchester · Built Worldwide*

Rules

- Taglines are set in **Bebas Neue** (uppercase), body/support in **Inter**.
- Use **paper** (white) on black backgrounds. Accents: **blood** for emphasis glyphs, **heat** for Dark Web/digital cues, **gold** for prestige.
- Pair with icons for quick semantic cues:
- **Balance** (crime vs karma), **Scale**, or **Gavel** for ethics/cause.
- **MapPin** with **Manchester** label for provenance.
- **ShieldAlert** + **18+** for age gate.
- **HandHeart** (or **HeartHandshake**) for Clean Intent pledge.

Example (hero/footer lockup)

```
<div className="flex flex-col md:flex-row md:items-center gap-3 md:gap-6">
  <div className="flex items-center gap-2">
    <BrandIcon name="Scale" variant="blood" size="md" className="blood-flicker" />
    <span className="headline text-2xl">BLOOD MONEY: CRIME VS KARMA</span>
  </div>
  <div className="flex items-center gap-2 text-smoke">
    <BrandIcon name="MapPin" variant="heat" size="md" className="uv-pulse" />
    <span className="uppercase tracking-wide text-sm">MADE IN MANCHESTER 
  </div>
  <div className="flex items-center gap-2">
    <BrandIcon name="ShieldAlert" variant="paper" size="md" />
    <span className="uppercase tracking-wide text-sm">18+</span>
  </div>
</div>
```

9. Footer & Final CTA — Standard Wiring

All footer/CTA blocks must use brand tokens + **BrandIcon** + motion utilities.

9.1 **FinalCTA.tsx** (footer CTA band)

```
// /components/brand/FinalCTA.tsx
import { BrandIcon } from "@/components/brand/Icon"

export default function FinalCTA(){
  return (
```

```

<section className="bg-charcoal text-paper py-10 md:py-14">
  <div className="max-w-6xl mx-auto px-6 flex flex-col md:flex-row items-
center justify-between gap-6">
    <div>
      <h3 className="headline text-3xl md:text-4xl">JOIN THE CARTEL</h3>
      <p className="text-smoke mt-1">Get launch alerts, dev drops, and
early-bird perks.</p>
    </div>
    <div className="flex items-center gap-4">
      <a href="#signup" className="btn-primary">Sign Up</a>
      <div className="flex items-center gap-3">
        <a aria-label="Instagram" href="https://instagram.com/
narconations">
          <BrandIcon name="Instagram" size="md" className="uv-pulse
hover:text-blood" />
        </a>
        <a aria-label="Twitter" href="https://x.com/narconations">
          <BrandIcon name="Twitter" size="md" className="uv-pulse
hover:text-blood" />
        </a>
        <a aria-label="Youtube" href="https://youtube.com/@narconations">
          <BrandIcon name="Youtube" size="md" className="uv-pulse
hover:text-blood" />
        </a>
      </div>
    </div>
  </div>
</section>
)
}

```

9.2 FixedFooter.tsx (site-wide footer)

```

// /components/brand/FixedFooter.tsx
import { BrandIcon } from "@components/brand/Icon"
import Link from "next/link"

export default function BrandFooter(){
  return (
    <footer className="bg-ink text-paper border-t border-white/10">
      <div className="max-w-6xl mx-auto px-6 py-8 grid gap-6 md:grid-cols-3">
        {/* Tagline & provenance */}
        <div className="space-y-2">
          <div className="flex items-center gap-2">
            <BrandIcon name="Scale" variant="blood" size="md"
className="blood-flicker" />
            <span className="headline text-xl">BLOOD MONEY: CRIME VS KARMA</
span>
          </div>
          <div className="flex items-center gap-2 text-sm text-smoke">

```

```
<BrandIcon name="MapPin" variant="heat" size="md" className="uv-  
pulse" />
```