

Brand Design Skill



Skill Directory Structure

You need to create a folder named brand-identity inside your .agent/skills/ directory. Inside that folder, create the following structure:

.agent/skills/brand-identity/

- |—— SKILL.md # The main entry point
- └—— resources/ # Folder for specific guidelines
 - |—— design-tokens.json # Colors, fonts, radii (machine readable)
 - |—— tech-stack.md # Frameworks and coding rules
 - └—— voice-tone.md # Copywriting guidelines

File Contents

1. .agent/skills/brand-identity/SKILL.md

This file acts as the router. It tells the agent *where* to look based on what it's trying to do.

Markdown

name: brand-identity

description: Provides the single source of truth for brand guidelines, design tokens, technology choices, and voice/tone. Use this skill whenever generating UI components, styling applications, writing copy, or creating user-facing assets to ensure brand consistency.

Brand Identity & Guidelines

****Brand Name:**** [INSERT BRAND NAME HERE]

This skill defines the core constraints for visual design and technical implementation for the brand. You must adhere to these guidelines strictly to maintain consistency.

Reference Documentation

Depending on the task you are performing, consult the specific resource files below. Do not guess brand elements; always read the corresponding file.

For Visual Design & UI Styling

If you need exact colors, fonts, border radii, or spacing values, read:

👉 ****[`resources/design-tokens.json`](resources/design-tokens.json)****

For Coding & Component Implementation

If you are generating code, choosing libraries, or structuring UI components, read the technical constraints here:

👉 `**resources/tech-stack.md**`

For Copywriting & Content Generation

If you are writing marketing copy, error messages, documentation, or user-facing text, read the persona guidelines here:

👉 `**resources/voice-tone.md**`

2. `.agent/skills/brand-identity/resources/design-tokens.json`

This is the most crucial file for design. Agents prefer JSON for exact values. **Fill in your specific hex codes and font names here.**

JSON

```
{
  "meta": {
    "brand_name": "[INSERT NAME]",
    "description": "Core design tokens for UI implementation."
  },
  "colors": {
    "primary": {
      "DEFAULT": "#000000",
      "hover": "#333333",
      "foreground": "#FFFFFF"
    },
    "secondary": {
      "DEFAULT": "#F4F4F5",
      "foreground": "#18181B"
    },
    "background": "#FFFFFF",
```

```
"foreground": "#09090B",

"muted": "#F4F4F5",

"accent": "#F4F4F5",

"destructive": "#EF4444",

"success": "#10B981"

},

"typography": {

  "font_family_headings": ["Inter", "sans-serif"],

  "font_family_body": ["Roboto", "sans-serif"],

  "font_weight_bold": "700",

  "font_weight_normal": "400"

},

"ui": {

  "border_radius_default": "0.5rem",

  "border_radius_small": "0.25rem",

  "spacing_base_unit": "4px"

}

}
```

3. [.agent/skills/brand-identity/resources/tech-stack.md](#)

Define the strict technical rules here. This stops the agent from randomly using Bootstrap when you want Tailwind.

Markdown

Preferred Tech Stack & Implementation Rules

When generating code or UI components for this brand, you ****MUST**** strictly adhere to the following technology choices.

Core Stack

- * **Framework:** React (TypeScript preferred)
- * **Styling Engine:** Tailwind CSS (Mandatory. Do not use plain CSS or styled-components unless explicitly asked.)
- * **Component Library:** shadcn/ui (Use these primitives as the base for all new components.)
- * **Icons:** Lucide React

Implementation Guidelines

1. Tailwind Usage

- * Use utility classes directly in JSX.
- * Utilize the color tokens defined in `design-tokens.json` (e.g., use `bg-primary text-primary-foreground` instead of hardcoded hex values).
- * **Dark Mode:** Support dark mode using Tailwind's `dark:` variant modifier.

2. Component Patterns

- * **Buttons:** Primary actions must use the solid Primary color. Secondary actions should use the 'Ghost' or 'Outline' variants from shadcn/ui.
- * **Forms:** Labels must always be placed *above* input fields. Use standard Tailwind spacing (e.g., `gap-4` between form items).
- * **Layout:** Use Flexbox and CSS Grid via Tailwind utilities for all layout structures.

3. Forbidden Patterns

- * Do NOT use jQuery.
- * Do NOT use Bootstrap classes.
- * Do NOT create new CSS files; keep styles located within component files via Tailwind.

4. .agent/skills/brand-identity/resources/voice-tone.md

Simple rules for how the agent should "speak" when writing on behalf of the brand.

Markdown

Copywriting: Voice & Tone Guidelines

When generating text, adhere to this brand persona.

Brand Personality Keywords

- * Professional but approachable
- * Direct and efficient
- * Tech-savvy but jargon-free
- * Empathetic

Grammar & Mechanics rules

- * **Headings:** Use Title Case for main headings (H1, H2). Use sentence case for subheadings (H3+).
- * **Punctuation:** Avoid exclamation points (!) in standard interface copy. Use periods for complete sentences.
- * **Clarity:** Prefer active voice over passive voice. Keep sentences concise.

Terminology Guide

| Do Not Use | Use Instead |

| :--- | :--- |

| "Utilize" | "Use" |

| "In order to..." | "To..." |

| [Add word] | [Add replacement] |