🌈UI/UX Design and Style Standardization

From Phong Trinh Ha Cameron Proulx Jacob Beaumont Gabriel Laboy

Date Mar 28, 2025

🎯 Goal

Our design system reflects a **dynamic and upbeat tone** tailored for a **younger audience**, while maintaining the **warmth and approachability** that defines our volunteer-driven mission.

We achieve this through:

- A **vibrant color palette** rooted in human emotion and clarity.
- **Accessible contrasts** and smooth transitions across themes.

- A **modern layout system** that feels intuitive, joyful, and inclusive.

Our aim is to create a UI that is not only beautiful but also **inclusive, adaptable, and emotionally intelligent**.

🎨 Color Schemes & Theming

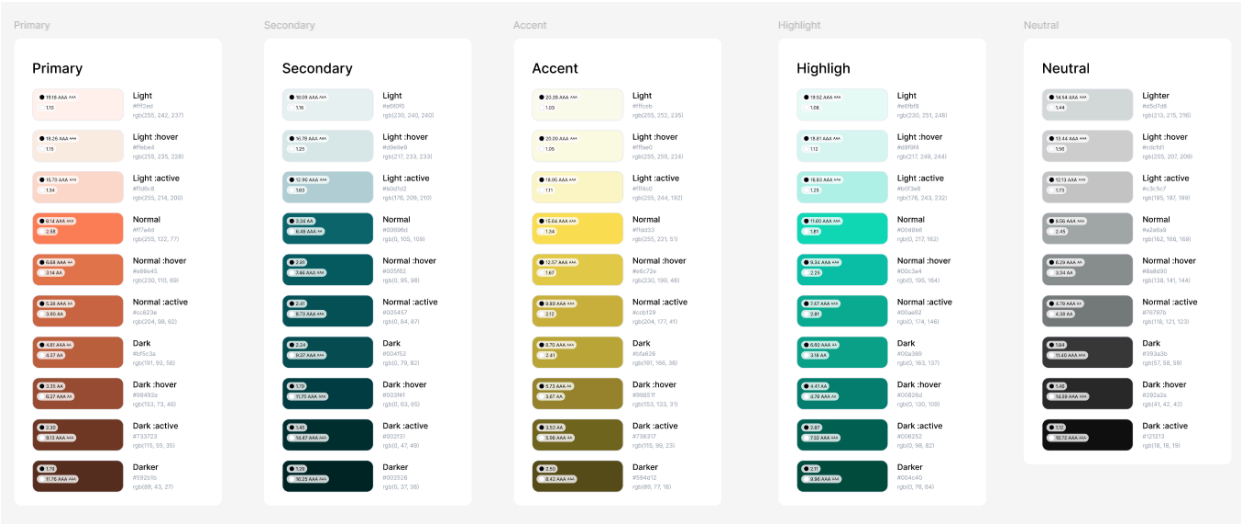
Our theme draws deep inspiration from real-world organizations that align with our mission, values, and target audience. The goal is to reflect a **sense of belonging, effortlessness, and warmth**—without compromising on clarity or usability. 🌱 Nurture Green and 🇮🇳 Volunteer for India are the main inspiration for our color palette:

Furthermore, we also took heavy inspiration from honey, its **approach to branding and interaction design** influenced our idea of what effortlessness *feels* like.

honey logo

Our logo

We've developed a comprehensive color system based on `oklch()` to optimize contrast, perceptual consistency, and accessibility. These values will be codified into our **extended `shadcn/ui` theme tokens**.



🧩 Palette Structure

Each color family (primary, secondary, accent, highlight, text) is designed with a full 50–950 scale to ensure flexibility across light/dark modes and component states. We follow the principle of **semantic theming**, where tokens represent **function, not just color**.

- `--color-primary-500`: Hero elements, calls-to-action
- `--color-secondary-500`: Tags, sub-actions, informational elements
- `--color-accent-400`: Positive highlights (e.g. achievements, badges)
- `--color-highlight-400`: Onboarding, tooltips, feature emphasis
- `--color-text-*`: Calibrated for readability and emotion

We use the **Geist font family** throughout the app for its modern, clean, and highly legible qualities.

🧱 Spacing, Sizing & Layout

To standardize spacing and maintain rhythm:

- Tailwind utility classes are used throughout: `p-*`, `m-*`, `gap-*`, `space-*`
- Consistent use of `flex` and `grid` layouts with logical gap and padding ensures cohesion.

- Design tokens will eventually be applied via `shadcn/ui`'s theme system for future scalability.
- Everything is constant based in our package so that it stay consistent throughout our development process

🧩 Component Behavior & Interactivity

We rely on the **default animations, transitions, and hover states provided by `shadcn/ui`**, which align with our tone of subtle delight and smooth interaction.

> All custom components **must match** the UX behavior (timing curves, duration, opacity, etc.) of `shadcn/ui`.

As we refine the brand aesthetic, these defaults may be tuned further—e.g., adding micro-interactions for feedback or using motion as a reward system in gamified features.

♿ Accessibility Standards

Accessibility is a **non-negotiable** aspect of our design.

- `shadcn/ui` components inherit accessibility from **Radix UI primitives**, which follow WAI-ARIA best practices.
- We maintain strong semantic HTML practices (e.g., `htmlFor`, `aria-*`, `alt`, `role`).
- Developers are encouraged to use Radix's `asChild` prop and accessibility hooks to extend functionality.

We also perform periodic **manual accessibility audits** during UI milestones.

📱 Responsive Design Strategy

Our UI must feel **native, fluid, and readable** across all devices.

🌐 For Web (React):

- **TailwindCSS** is the backbone of responsiveness via `sm`, `md`, `lg`, `xl`, etc.
- We leverage utility classes to control breakpoints, grid/flex behaviors, and content hierarchy.
- `shadcn/ui` components are **fully responsive by default** and easily customized per screen size.

📱 For Mobile (React Native):

- **NativeWind** is used in place of TailwindCSS for consistent utility-class-based styling.
- **React Native Reusables** mimic the shadcn/ui component structure, adapted for mobile ergonomics.
- Custom components follow **mobile-first principles**, prioritizing tap targets, spacing, and performance.

🔍 Testing & Tooling

- Web: Use **Vite** dev servers + Chromium DevTools to live-test responsiveness.
- Mobile: Use **Expo Go** and native device simulators (iOS/Android) for layout validation.
- Frequent QA passes on various screen sizes are key to ensuring fluid, usable experiences.

🧠 Design Philosophy

We are designing for:

- **Young volunteers and first-time contributors**
- **Mobile-first audiences**, with desktop as an enhanced experience
- **Emotionally resonant design** that feels human, not sterile

Our style language reflects **motion, clarity, and emotional warmth**—grounded in utility but always expressive.

.