



## UI Style & Accessibility Guidelines

This document outlines the design system, visual identity, and accessibility standards used in our application. It ensures consistency and usability across all components and screens.

### 1. Design Inspiration

We draw principles from the following guidelines:

- [Google's Material Design](#) for component behavior, animation, and spacing.
- **Apple's Human Interface Guidelines** for clarity, deference, and depth.
- **Microsoft Fluent UI** for accessibility and form interactions.

### 2. Color Scheme

- **Background:** #242424 (Grey)
- **Primary Text:** #FFFFFF (White)
- **Accent Green:** #7ac59b (Used for highlights, buttons, icons). Taken from Logo
- **Gray Shades:** #1f1f1f, #2e2e2e, #9ca3af (used for input borders, dividers)

All color pairings meet WCAG AA contrast ratios.

**Add Meta Tags:** SEO enhancement

### 3. Typography

- **Font Family:** System UI / sans-serif
- **Weights:** Regular (400), Medium (500), Bold (700)

- **Sizes:**
  - Titles: `text-3xl` ( $\approx 30\text{px}$ )
  - Body: `text-base` ( $\approx 16\text{px}$ )
  - Labels & Captions: `text-sm` or `text-xs`

## 4. Spacing & Layout

- Consistent padding/margins using Tailwind spacing scale:
  - Default padding: `p-4`
  - Container width: `max-w-md` ( $\approx 28\text{rem}$ )
  - Gap between elements: `gap-4` or `space-y-6`

## 5. Buttons

- Use `<Button />` component.
- Styles: `w-full bg-white text-black py-3 rounded-lg hover:bg-gray-300 transition shadow-md`
- Must be keyboard-focusable with visible focus ring (`focus:outline-white`).

## 6. Forms & Inputs

- Inputs styled using Tailwind and Material UI Autocomplete where needed.
- Use developed `InputField` component for modularity
- Style: `border: 1px solid #2b2b2b; border-radius: 4px; background-color: #2b2b2b; color: white;`
- Focus state: `focus:border-white`
- Errors: Show red text with `text-red-500 text-xs mt-1`

- Checkboxes: Allow user to select multiple options
- Validate Input: non-empty and correct format (for emails)

## 7. Components

Component	Classes/Props
Map Container	<code>w-[60%]</code> <code>h-[300px]</code> <code>rounded-xl</code> (mobile: <code>w-full</code> )
Profile Card	<code>bg-white/10</code> <code>backdrop-blur-md</code> <code>rounded-2xl</code> <code>p-8</code>
Form Wrapper	<code>w-full</code> <code>max-w-md</code> <code>p-8</code> <code>border</code> <code>border-gray-400</code>
Modal/Dialog	<code>shadow-lg</code> <code>rounded-xl</code> <code>bg-gray-800</code>
Community Tabs	<code>flex</code> <code>rounded-xl</code>

## 8. Accessibility (WCAG 2.1 AA)

- **Color Contrast:** All text meets contrast ratio of at least 4.5:1.
- **Keyboard Navigation:** All buttons and inputs are keyboard accessible.
- **ARIA Labels:** Add `aria-label` for all icons or inputs without visible labels.
- **Error Messaging:** Inline error messages for inputs, using `role="alert"` where necessary.

## 9. Navigation Bar

- Contrasting Background Color: #333
- High Z-index to make it always appear on top

## 10. Responsive Design

- Tailwind responsive breakpoints used:

- `sm: ≥640px`
- `md: ≥768px`
- `lg: ≥1024px`
- Use `w-full`, `max-w-md`, and `text-center` for mobile-friendliness.
- Scrollable panels: `overflow-y-auto`

## 11. Animations

- Subtle transitions on buttons and inputs:
  - `transition duration-200 ease-in-out`
  - `background-color 0.3s ease`
- Hover effects:
  - Buttons change color: `hover:bg-[#hex_code]`
  - Links: `hover:underline`
- Optional additions: Animate page transitions using Framer Motion in future.

## 12. Assets & Icons

- All icons should use consistent sizing (`w-5 h-5` or `w-6 h-6`)
- Prefer icons using Material Icons or FontAwesome.
- Images (if added later) should be optimized with:
  - Max size: 500KB
  - Format: `.webp` preferred
  - Compression: Use TinyPNG or Squoosh

- Effective alt text should be provided for each image. It should not start with “Image of ..”

### **13. Future Recommendations**

- Add a dark/light theme toggle using Tailwind’s `dark:` variant or Material UI.
- Migrate inline styles into centralized component classes for better maintainability.
- Consider using a design tool (e.g., Figma) to map component hierarchy visually.