

ZUDY BEST PRACTICES

# UI/UX Design

© Zudy

1560 Lenox Ave · Suite 203

Miami Beach, FL 33139

Phone 305.763.8287 · Fax 305.763.8452

## Introduction

This guide presents Zudy's **UI/UX Design Best Practices** to use when developing apps with the Vinyl platform. The versatility of Vinyl allows you to accomplish virtually anything when it comes to how an app looks and feels. This guide offers UI/UX best practices to follow to help achieve the best design outcome.

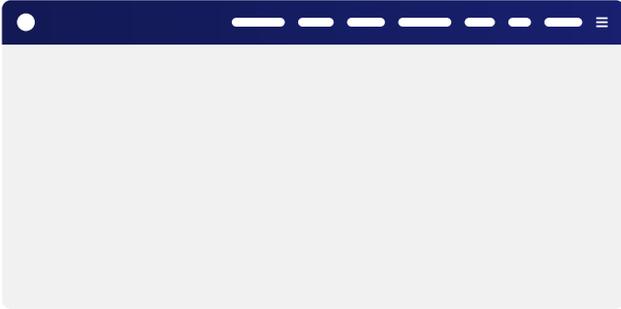
In addition to design, we strongly encourage all developers to think about accessibility when developing apps. See our [Zudy App Accessibility Checklist](#).

# Layout

# Navigation Type

Choose the most suitable navigation type based on context.

## Top Navigation

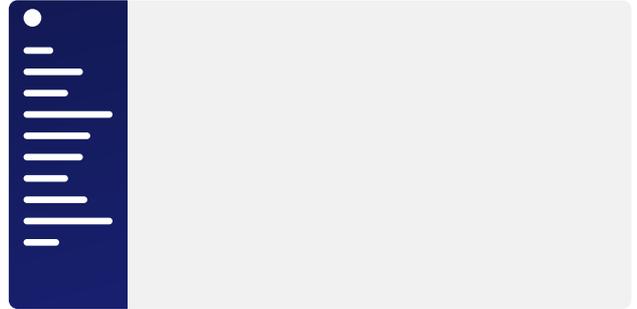


- + Takes up less space overall
- + Better suited for hover-activated sub-menus
- Can't fit as many items in
- Difficult to scale if you add more items
- Not well suited for responsive design
- Harder to scan items
- Impossible to show a folder/tree structure

### Best suited for:

- ✓ Apps with few top-level navigation items
- ✓ Apps where maximum screen space is needed
- ✓ Apps with deep sub-menu structures

## Side Navigation



- + Better suited for responsive design
- + Fits a lot more items
- + Easy to scale if you add more items
- + Easier to scan items
- + Allows for a folder/tree structure
- Takes up more space overall
- Doesn't suit hover-activated sub-menus

### Best suited for:

- ✓ Apps with many top level navigation items
- ✓ Apps with a clear folder/tree structure
- ✓ Apps where responsiveness is important

Navigation is the primary way users move around or navigate in an app, and it is critical to be as intuitive and user-friendly as possible.

When designing web applications in Vinyl, best practice is to use either **top** or **side** navigation. These are the regions a user expects to find navigation. You can find the pros and cons of each approach above.

# Visual Hierarchy

*Use size, weight & color to introduce a visual hierarchy.*

Card Title

Card Sub-Title

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

**Card Title**

**Card Sub-Title**

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.



Using **size, weight & color** to introduce a visual hierarchy is key in establishing a balanced UI that users can easily scan & navigate.

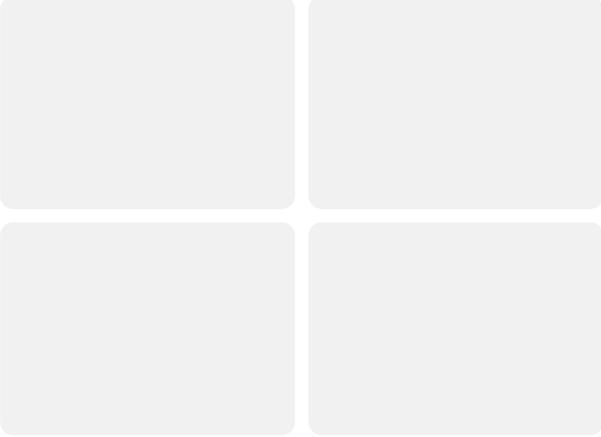
Visual hierarchy between panel headings, sub-headings, and content (e.g. rows in a table) can help **guide a user** through an interface and help them focus their attention in the right places.

Especially for users who are just scanning the page to find the heading they're looking for, giving them an easy way to do that is key.

# Generous Whitespace

*Generous spacing between elements can reduce clutter.*

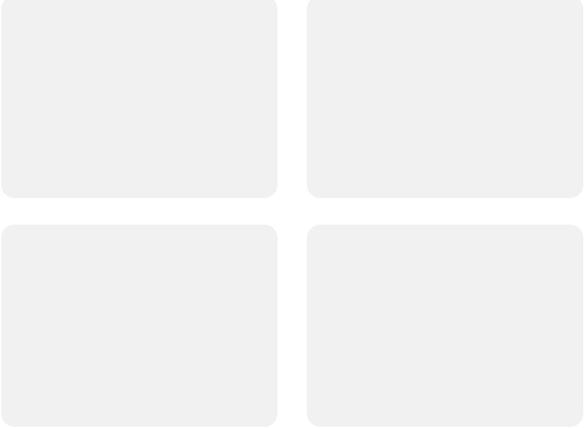
**Example Title**



Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex.



**Example Title**



Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud.



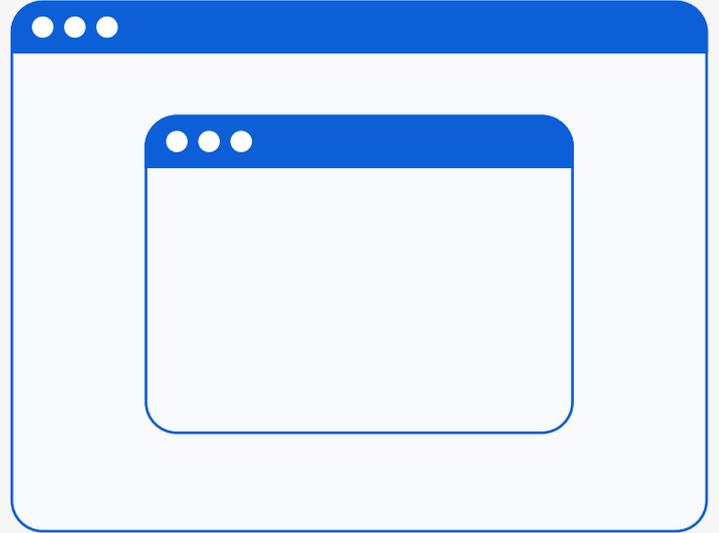
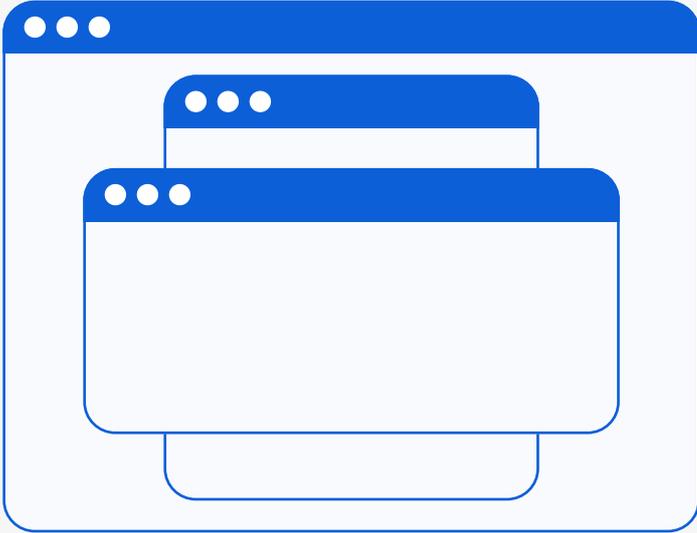
Elements need **space to breathe**, and too many UI elements in close-proximity can become overwhelming.

Lots of panels & charts on a page with minimal spacing can look incredibly cluttered, and can overwhelm a user and **increase their cognitive load**.

Using generous whitespace can significantly reduce clutter & make it easier for users to digest information.

# Multiple Popups

*Do not show multiple popups one after another.*



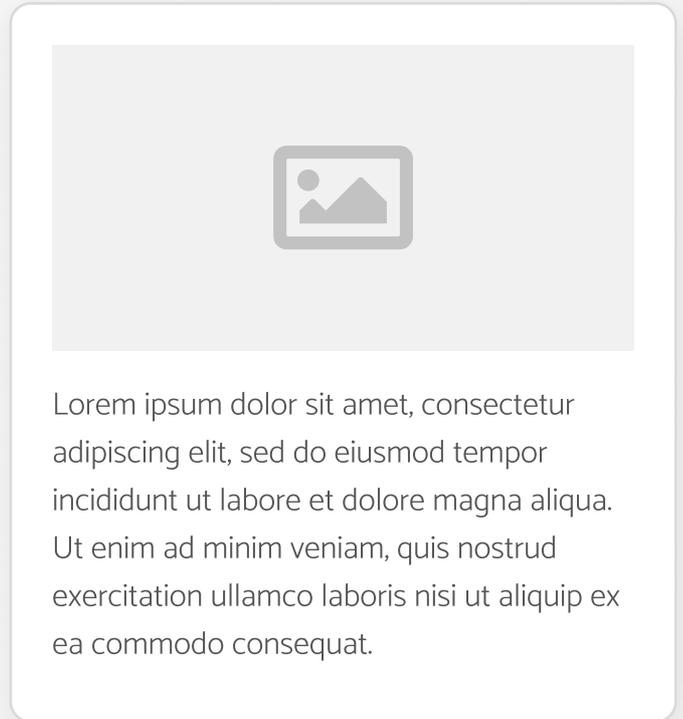
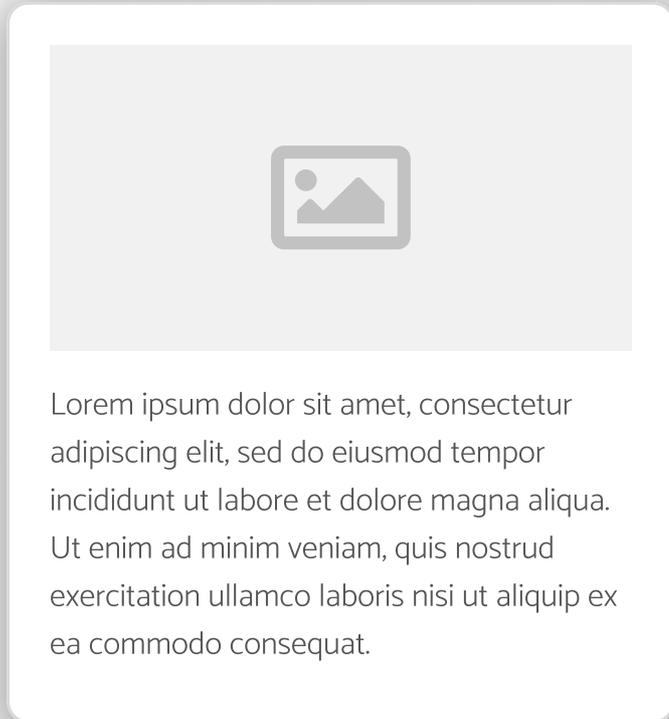
Displaying multiple popups on top of each other can make your site look unprofessional and disorganized.

It can also **overwhelm users** and force them to spend effort to close each one.

If you must present critical information like important warnings in a popup, be sure to **show only one at a time**.

# Subtle Box Shadows

*Overly dark or intense shadows can add clutter.*



Shadows can introduce a clean, modern feel to your UI – but **subtlety is key**.

Thick, dark shadows can be too bold and overwhelming, and end up detracting from a clean look & feel.

The example above uses a light grey shadow - `rgba(0,0,0,.1)` - on an off-white background - `#F6F6F6`.

Additionally, adding a **1px border** just a few shades darker than the shadow color will make your element pop more and avoid muddy/blurry looking shadows.

# Inputs

# Button Styling

*Make buttons look like buttons.*

## Inviting Action Design

Completing this action will publish your work to the public repository. You can also choose to publish the file at a later time.

Save for Later

Publish



## Inviting Action Design

Completing this action will publish your work to the public repository. You can also choose to publish the file at a later time.

Save for Later

Publish



Users will typically **recognize buttons with visual cues** that indicate that an element is clickable.

Don't get too "creative" and style a button into something that doesn't look interactive to users.

Common characteristics of buttons include rectangular shapes, rounded corners, box-shadows, bold colors and clear action labels.

# Generic Button Text

*Avoid generic commands in actionable button text.*

**Publish to Repository?**

Completing this action will publish your work to the public repository. You can also choose to publish the file at a later time.



**Publish to Repository?**

Completing this action will publish your work to the public repository. You can also choose to publish the file at a later time.



Button labels should inspire users to act with action verbs.

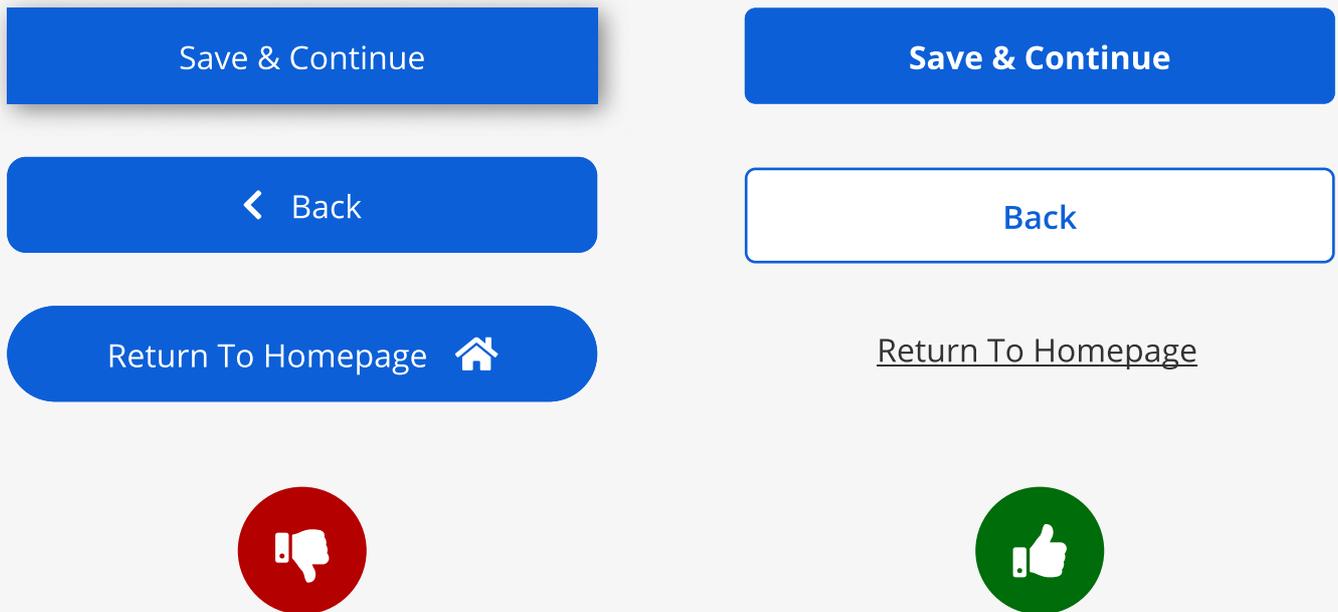
**Action verb button labels are more task-efficient**, and help users take action without reading any supporting text.

Generic “Yes/No” button labels are vague and require users to carefully read the details before taking action.

**Passive labels not only make it riskier to take action**, but also forces users to do more work.

# Commonality & Hierarchy

*Common elements create a consistent, comfortable UI for users.*



Element styling should be **consistent throughout an application**, and should have a **clear hierarchy** so users know which elements are the most important.

In the example above, a clear button hierarchy has been established, and the primary actions are designed to catch the user's attention first.

In addition, once a user is familiar with how a primary button looks, styling should be consistent and they should be able to use that knowledge across the application.

# Helpful Errors

*Error messages should be clear, concise, and not blame users.*

Email

example@zudy.com

⚠ You entered an invalid email.



Email

example@zudy.com

⚠ This email is already in use. [Log in](#) to your account or [reset your password](#).



Your UI should always aim to prevent errors from occurring, but when they do, they should use **plain language** (no error codes), **precisely identify the problem**, and constructively **suggest a solution**.

Error messages should be descriptive and useful, and should always **take blame away from the user**.

You **shouldn't ever rely solely on color** to convey information e.g. red border alone to indicate an error. Doing so can alienate users with accessibility issues, and confuse users who need descriptive context to understand the information.

# Clickable Area

*Make sure clickable elements are large enough to interact with.*

State

State

Web applications are used across a wide range of devices and screen sizes, and users will interact with elements differently across them all.

Particularly on mobile, it's important to make sure **clickable elements are large enough** to interact with.

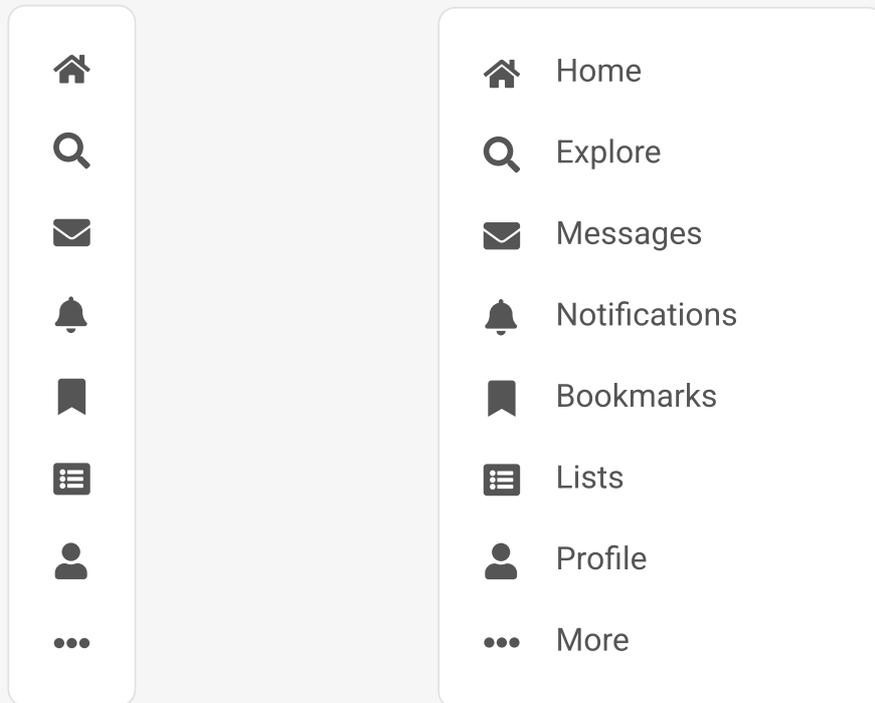
In general, elements should be **at least 32x32px** to make sure they are easily clickable across devices.

Small click areas can lead to frustration & difficulty for users when trying to achieve their goal.

# Icons

# Icon Labels

*Support icon recognition with relevant labels.*



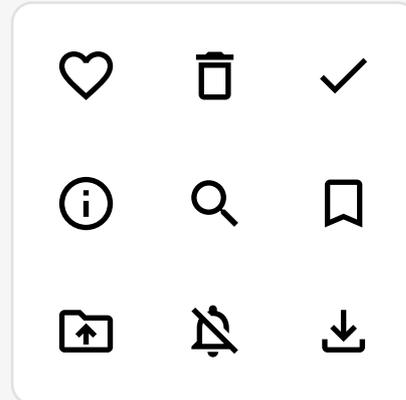
**Use icons in combination with meaningful labels** to overcome the ambiguity that icons face.

**Most icons are not widely recognized** to users due to their association with different meanings across various interfaces.

As such, using a label or supporting text will help ensure an icon meaning is understood.

# Established Icons

*Support icon recognition with relevant text.*



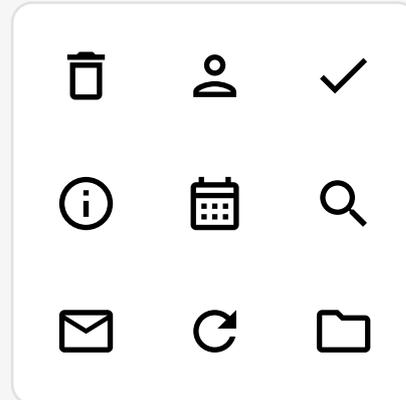
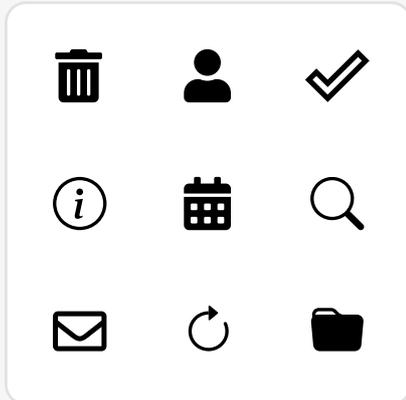
Ambiguous icons can introduce cognitive obstacles for users, and easily lead to confusion and a negative user experience.

**Using established, widely used icons** will help people recognize their meaning and become more adept in using the interface.

Icons that are **widely used across a range of web applications** are more likely to be familiar to users, for example the magnifying glass icon for search fields.

# Icon Styles

*Use consistent icon styles.*



Use icons that are all in a similar style for consistency and readability.

Keeping the **same icon parameters** (such as fill, stroke thickness, size, and shape) builds consistency throughout the application.

**Avoid mixing and matching different icon styles** as this can make the UI look cluttered and unfocused.

**Text**

# Different Font Families

*Use no more than two different font families.*

## Card Title

### Card Sub-Title

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris.

## Card Title

### Card Sub-Title

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris.



Using more than 2 different fonts makes an application look unstructured and inconsistent.

In general, limit the number of font families to a minimum – **two fonts are plenty**, and one is often sufficient.

Stick to the same fonts consistently throughout the entire application.

# All Caps

*Avoid all caps for body text.*

APPLICATION DEVELOPMENT IS THE PROCESS OF CREATING A COMPUTER PROGRAM OR A SET OF PROGRAMS TO PERFORM THE DIFFERENT TASKS THAT A BUSINESS REQUIRES. FROM CALCULATING MONTHLY EXPENSES TO SCHEDULING SALES REPORTS, APPLICATIONS HELP BUSINESSES AUTOMATE PROCESSES AND INCREASE EFFICIENCY.

Application development is the process of creating a computer program or a set of programs to perform the different tasks that a business requires. From calculating monthly expenses to scheduling sales reports, applications help businesses automate processes and increase efficiency.



All caps text should only be used for short alerts or large headings. The larger characters make it easier to glance at, but **slower to read**.

When your text involves reading, stay away from using all caps text.

If you want to emphasize body text, **bolding** it applies the same effect without sacrificing readability.

# Font Pairings

*Use one typeface for headings and a different one for body text.*

## Roboto Title

### Roboto Sub-Title

(Roboto) Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea.

## Signika Title

### Signika Sub-Title

(Hind Madurai) Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo.



Default typefaces are common, dependable, and (potentially) boring. Even **changing from the default typefaces** (e.g. Arial, Helvetica, Times etc.) can bring a design to life.

<https://fontjoy.com> is a useful tool for finding matching font families.

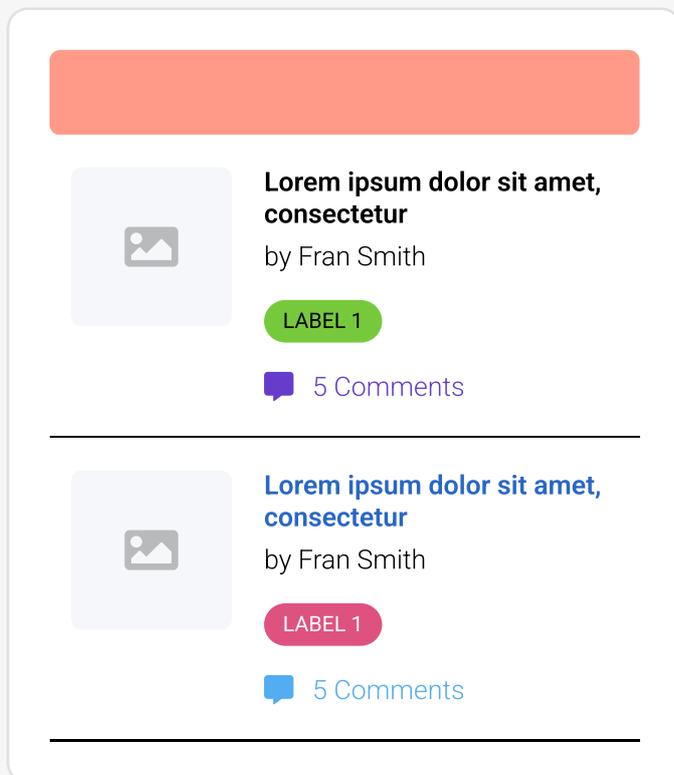
Aesthetically pleasing font pairings can go a long way in improving the look & feel of an application.

There are also typefaces that have been specifically designed for use on screens, providing optimal readability at a range of resolutions and brightness levels e.g. Poppins.

# Colors

# Color Usage

*Limit usage of colors to 3 or 4.*



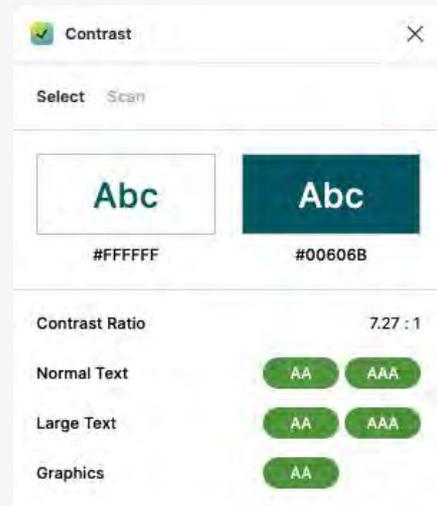
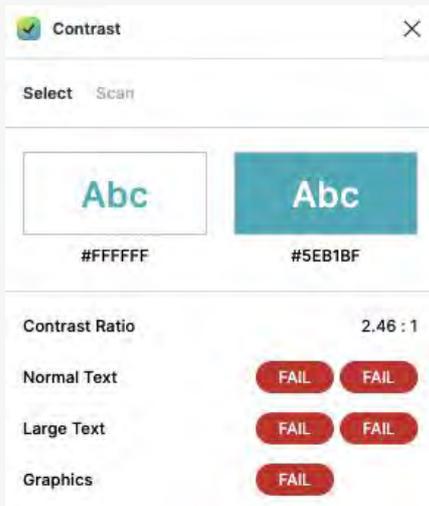
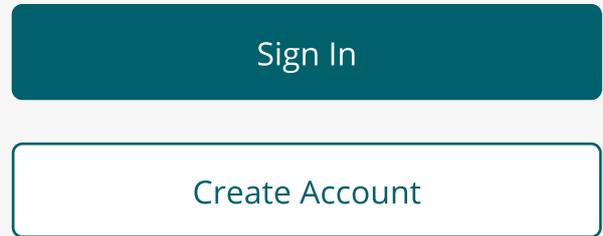
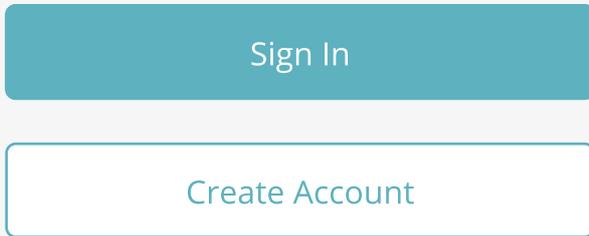
A color palette shouldn't exceed 6 colors, and a good number to aim for would be 3 or 4 (**1 dominant color, 1 secondary color, and 1-2 accent colors**).

Color is primarily used to draw attention, and too many different colors will start to diminish the ability to draw attention.

The “**60–30–10 rule**” is a rough tool for ensuring a systematic usage of color. 60% is your dominant color, 30% is a secondary color, and 10% is for the accent colors.

# Sufficient Contrast Ratios

*Text should have a high contrast ratio and be easily readable.*



To increase legibility, especially for visually impaired users, it's important to have a high enough contrast ratio to **make text easily readable**.

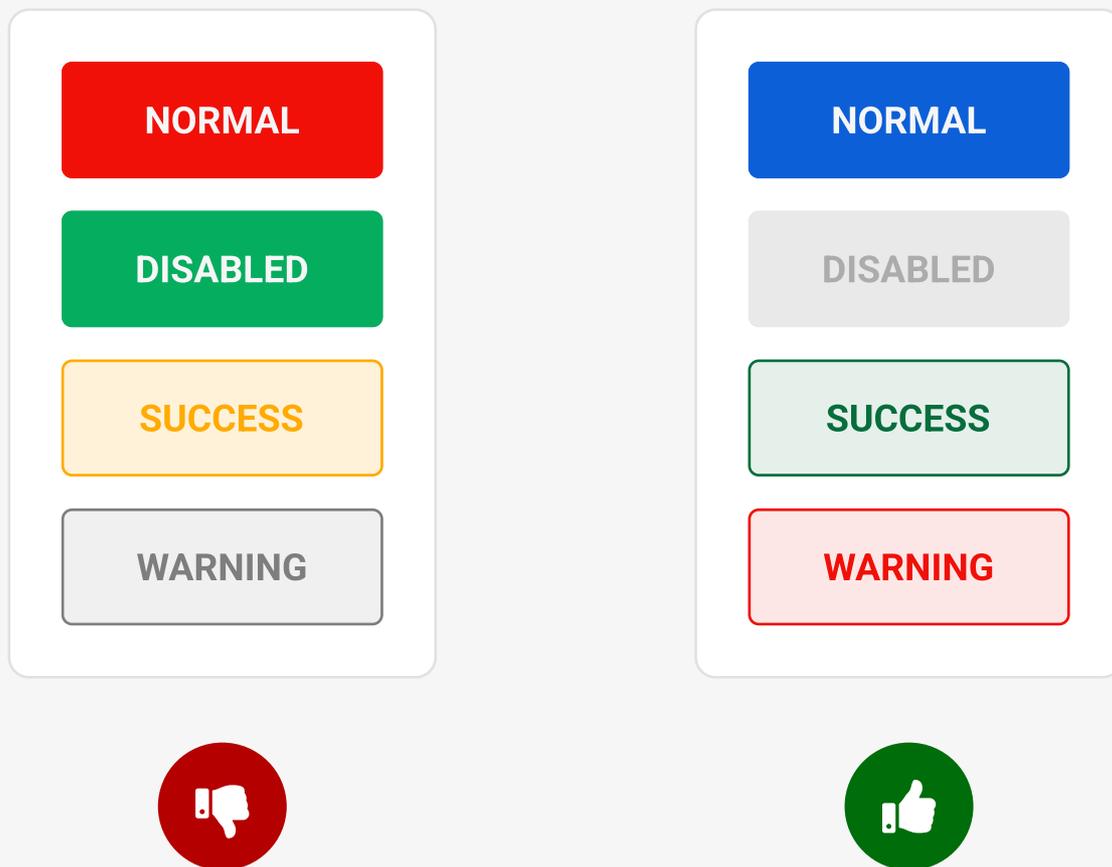
Current WCAG/ADA guidelines require a contrast ratio of **at least 4.5:1 for normal text** and 3:1 for large text.

Typical colors that have low contrast with black or white text are orange, yellow and light blue, although there are useful tools you can use to check contrast ratios as you're working: <https://colors.co/contrast-checker>

Google Chrome also has a contrast tool built into the web inspector.

# Semantic Colors

*Use commonly associated colors to provide information.*



Color can be used to convey emotion and **provide information** about the state of an application, its components, and elements.

Color is one way to **change the state of objects** in the interface.

By muting the colors of a button, it can indicate that a button is disabled or by highlighting it in red, it can signal an error.

Make sure that semantic colors do not clash with brand colors.