Keeping content in chat

01/07/2026 19:49:22

I'm ready to begin! Let's dive into your comprehensive Tailwind CSS...
MODULE 2: CORE CONCEPTS & UTILITIES How Tailwind Utility Classes Wo...
MODULE 3: COLORS & BACKGROUNDS Color Palette System (50-900...
MODULE 4: TYPOGRAPHY Font Family, Size, Weight Font Family Tailwind...
MODULE 5: LAYOUT WITH FLEXBOX & GRID Flex Utilities (Justify, Align,...
MODULE 6: RESPONSIVE DESIGN Breakpoint System (sm, md, lg, xl, 2xl)...
MODULE 7: STATE VARIANTS & INTERACTIONS Hover States (hover:) The...
MODULE 8: BORDERS, SHADOWS & EFFECTS Border Width, Style, Radius...
MODULE 9: CUSTOMIZATION tailwind.config.js Deep Dive The...
MODULE 10: COMPONENTS & PATTERNS Building Reusable Components...
MODULE 11: BEST PRACTICES & OPTIMIZATION Class Organization...
MODULE 12: REAL-WORLD PROJECTS In this final module, we'll build...
Project 2: Dashboard Layout A complete admin dashboard with sidebar,...
Project 3: E-commerce Product Card A complete e-commerce product car...
Project 4: Blog Post Layout A complete blog layout with article header,...
You're absolutely right! I apologize for missing these important propertie...

I'm ready to begin! Let's dive into your comprehensive Tailwind CSS learning journey. Since you have a solid CSS foundation, I'll make frequent connections to vanilla CSS concepts you already know.

MODULE 1: INTRODUCTION & SETUP

What is Tailwind CSS and Why Use It?

Tailwind CSS is a **utility-first CSS framework** that provides low-level utility classes to build custom designs directly in your HTML. Unlike component frameworks (Bootstrap, Material UI), Tailwind doesn't impose design decisions on you.

Think of it this way:

- **Vanilla CSS**: You write custom classes and define their styles
- **Bootstrap**: You use pre-made components (`.btn-primary`, `.card`)
- **Tailwind**: You compose utility classes to build anything (`.bg-blue-500`, `.p-4`, `.rounded`)

Why Developers Choose Tailwind:

1. **Speed**: No context switching between HTML and CSS files
2. **Consistency**: Built-in design system (spacing, colors) prevents arbitrary values
3. **Maintainability**: Changes are localized to components
4. **Smaller CSS**: Only ships utilities you actually use
5. **Responsive Design**: Inline responsive modifiers (`md:flex`, `lg:grid`)

6. **No Naming Fatigue**: No need to invent class names like `.user-profile-card-header-title`

The Trade-off:

Pros:

- Rapid prototyping
- Design system out of the box
- No CSS file bloat from unused styles
- Easy to see what's applied without checking CSS files

Cons:

- HTML can look cluttered with many classes
- Learning curve for utility class names
- Requires build step for production (usually)
- Less semantic HTML on the surface

Utility-First Approach vs Traditional CSS

Let's compare building the same button:

Traditional CSS Approach:

```
<button class="btn btn-primary">Click Me</button>
```

```
.btn {
  padding: 0.5rem 1rem;
  border-radius: 0.25rem;
  font-weight: 600;
  border: none;
```

```
  cursor: pointer;
}

.btn-primary {
  background-color: #3b82f6;
  color: white;
}

.btn-primary:hover {
  background-color: #2563eb;
}
```

Tailwind Approach:

```
<button class="px-4 py-2 bg-blue-500 text-white font-semibold
rounded hover:bg-blue-700">
  Click Me
</button>
```

Breaking it down:

- `px-4` = padding-left and padding-right: 1rem
- `py-2` = padding-top and padding-bottom: 0.5rem
- `bg-blue-500` = background-color: #3b82f6
- `text-white` = color: white
- `font-semibold` = font-weight: 600
- `rounded` = border-radius: 0.25rem
- `hover:bg-blue-700` = darker blue on hover

Key Insight: You're essentially writing inline styles, but with constraints. Tailwind's spacing scale (4, 8, 12, 16px, etc.) prevents you from using arbitrary values like `padding: 13px`.

Installation Methods

1. CDN (Quick Start - Not for Production)

Perfect for learning and prototyping:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Tailwind CDN</title>
  <script src="https://cdn.tailwindcss.com"></script>
</head>
<body>
  <h1 class="text-3xl font-bold text-blue-600">
    Hello Tailwind!
  </h1>
</body>
</html>
```

Limitations:

- Larger file size (includes ALL utilities)
- No customization
- No build-time optimizations
- Not suitable for production

---

2. NPM with PostCSS (Recommended)

Step-by-step setup:

```
# Create project folder
mkdir my-tailwind-project
cd my-tailwind-project

# Initialize npm
npm init -y

# Install Tailwind
npm install -D tailwindcss postcss autoprefixer

# Generate config files
npx tailwindcss init
```

This creates `tailwind.config.js`:

```
/** @type {import('tailwindcss').Config} */
export default {
  content: [
    "./index.html",
    "./src/**/*.{js,ts,jsx,tsx}",
  ],
  theme: {
    extend: {},
  },
  plugins: [],
}
```

Create your CSS file (`src/input.css`):

```
@tailwind base;
@tailwind components;
```

```
@tailwind utilities;
```

What these directives do:

- `@tailwind base` - Resets and base styles (like Normalize.css)
- `@tailwind components` - Component classes (if you create any)
- `@tailwind utilities` - All utility classes

Add build script to `package.json`:

```
{
  "scripts": {
    "dev": "npx tailwindcss -i ./src/input.css -o
./dist/output.css --watch"
  }
}
```

Run the build:

```
npm run dev
```

Use in HTML:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <link href="/dist/output.css" rel="stylesheet">
</head>
<body>
  <h1 class="text-3xl font-bold">Hello Tailwind!</h1>
</body>
</html>
```

3. Vite + Tailwind (Modern & Fast)

Vite is a modern build tool that's incredibly fast:

```
# Create Vite project
npm create vite@latest my-app
cd my-app
npm install

# Install Tailwind
npm install -D tailwindcss postcss autoprefixer
npx tailwindcss init -p
```

The `-p` flag creates both `tailwind.config.js` and `postcss.config.js`.

Configure `tailwind.config.js`:

```
export default {
  content: [
    "./index.html",
    "./src/**/*.{js,ts,jsx,tsx}",
  ],
  theme: {
    extend: {},
  },
  plugins: [],
}
```

Update `src/index.css`:

```
@tailwind base;
@tailwind components;
```

```
@tailwind utilities;
```

Import in `src/main.js`:

```
import './index.css'
```

Start dev server:

```
npm run dev
```

Why Vite?

- Instant hot reload
- Optimized builds
- Modern ES modules
- Great developer experience

Understanding the Build Process

Here's what happens when you build Tailwind:

```
Your HTML/JSX files
       ↓
Tailwind scans for classes (content config)
       ↓
Generates only the CSS you need
       ↓
PostCSS processes @tailwind directives
       ↓
Autoprefixer adds vendor prefixes
```

```
        ↓
Output CSS file (small & optimized)
```

The `content` configuration is crucial:

```
content: [
  "./src/**/*.{html,js,jsx,tsx}",
]
```

This tells Tailwind: "Scan these files for class names." Any class found generates CSS. Classes not found are **purged** (removed), keeping your CSS tiny.

Example:

If your HTML only uses `bg-blue-500` and `text-white`, Tailwind only generates CSS for those two utilities, not all 1000+ possible utilities.

First Tailwind Project Setup

Let's create a simple landing page hero section:

1. Create `index.html`:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>My First Tailwind Project</title>
  <script src="https://cdn.tailwindcss.com"></script>
</head>
<body class="bg-gray-100">
```

```
  <!-- Hero Section -->
  <div class="min-h-screen flex items-center justify-center px-
4">
    <div class="text-center">
      <h1 class="text-5xl font-bold text-gray-900 mb-4">
        Welcome to Tailwind CSS
      </h1>
      <p class="text-xl text-gray-600 mb-8">
        Build amazing designs without leaving your HTML
      </p>
      <button class="bg-blue-600 text-white px-8 py-3 rounded-lg
font-semibold hover:bg-blue-700 transition">
        Get Started
      </button>
    </div>
  </div>

</body>
</html>
```

Breaking down the classes:

Body:

- `bg-gray-100` - Light gray background

Container div:

- `min-h-screen` - Minimum height of 100vh (full viewport)
- `flex items-center justify-center` - Center content vertically & horizontally
- `px-4` - Horizontal padding (prevents edge touch on mobile)

Heading:

- `text-5xl` - Very large font size (3rem / 48px)

- `font-bold` - font-weight: 700
- `text-gray-900` - Very dark gray (almost black)
- `mb-4` - Margin bottom: 1rem

Paragraph:

- `text-xl` - Larger text (1.25rem / 20px)
- `text-gray-600` - Medium gray
- `mb-8` - Margin bottom: 2rem

Button:

- `bg-blue-600` - Blue background
- `text-white` - White text
- `px-8 py-3` - Generous padding
- `rounded-lg` - Large border radius
- `font-semibold` - font-weight: 600
- `hover:bg-blue-700` - Darker on hover
- `transition` - Smooth color transition

Common Mistakes to Avoid

1. Not configuring the `content` array properly

```
// ❌ Wrong - won't scan your files
content: [],

// ✅ Correct
content: ["./src/**/*.{html,js,jsx,tsx}"],
```

2. Forgetting the @tailwind directives

Your CSS file MUST include:

```
@tailwind base;
@tailwind components;
@tailwind utilities;
```

3. Using arbitrary values too early

```
<!-- ❌ Defeats the purpose of Tailwind's system -->
<div class="p-[13px] m-[27px]">

<!-- ✅ Use the spacing scale -->
<div class="p-3 m-6">
```

4. Not running the build process

Remember: changes to `tailwind.config.js` require restarting the build.

Key Takeaways

✅ Tailwind is a **utility-first framework** that provides low-level building blocks
✅ You write classes directly in HTML instead of writing custom CSS
✅ The **build process** only includes CSS for classes you actually use
✅ **CDN** is great for learning, **npm** is better for production
✅ The `content` configuration tells Tailwind where to find your classes
✅ Tailwind enforces a **design system** through its spacing scale and color palette
✅ You'll write more HTML classes but way less CSS

Practice Assignments

Assignment 1 (Easy): CDN Setup

Create an HTML file using the Tailwind CDN and build a simple card with:

- A colored background
- Padding
- Rounded corners
- A heading and paragraph
- A button

Assignment 2 (Easy): Spacing Exploration

Create 5 divs with different padding values (`p-2`, `p-4`, `p-6`, `p-8`, `p-10`). Add a background color to see the spacing clearly. Observe how the spacing scale works.

Assignment 3 (Medium): NPM Setup

Set up a Tailwind project using npm (not Vite). Create the following file structure:

```
my-project/
  ├── src/
  │   └── input.css
  ├── dist/
  │   └── output.css
  ├── index.html
  ├── package.json
  └── tailwind.config.js
```

Get the build process working with `--watch` mode.

Assignment 4 (Medium): Vite + Tailwind

Set up a new project using Vite and Tailwind. Create a simple page with a navigation bar and hero section. Ensure hot reload is working (changes appear instantly).

Assignment 5 (Hard): Configuration Deep Dive

Set up a Tailwind project and modify `tailwind.config.js` to:

- Add a custom color called `brand` with shades 500 and 700
- Use your custom color in the HTML (`bg-brand-500`)
- Verify it works by checking the generated CSS

Hint for Assignment 5:

```
theme: {
  extend: {
    colors: {
      brand: {
        500: '#your-color-here',
        700: '#your-darker-color-here',
      }
    }
  }
}
```

Ready for **Module 2: Core Concepts & Utilities**? Let me know when you want to continue!

12/30/2025, 9:54:23 AM

MODULE 2: CORE CONCEPTS & UTILITIES

How Tailwind Utility Classes Work

Tailwind utilities are **single-purpose classes** that typically map 1:1 with CSS properties. Think of them as CSS properties in class form.

The Pattern:

```
[property]-[value]
```

Examples:

- `text-center` → `text-align: center`
- `font-bold` → `font-weight: 700`
- `bg-blue-500` → `background-color: #3b82f6`
- `p-4` → `padding: 1rem`

Comparison to Vanilla CSS:

```
/* Vanilla CSS */
.my-element {
  margin-top: 1rem;
  margin-bottom: 2rem;
  padding: 1.5rem;
  display: flex;
}
```

```
<!-- Tailwind -->
<div class="mt-4 mb-8 p-6 flex">
```

Each Tailwind class does **one thing**. You combine them to achieve complex designs.

Naming Conventions and Patterns

Tailwind follows predictable patterns. Once you learn them, you can guess class names:

1. Directional Modifiers

For properties that have directions (margin, padding, border):

Modifier	Meaning	CSS Property
`t`	top	`*-top`
`r`	right	`*-right`
`b`	bottom	`*-bottom`
`l`	left	`*-left`
`x`	horizontal (left + right)	`*-left` + `*-right`
`y`	vertical (top + bottom)	`*-top` + `*-bottom`
(none)	all sides	`*`

Examples:

```
<div class="mt-4">   <!-- margin-top: 1rem -->
<div class="px-6">   <!-- padding-left: 1.5rem; padding-right: 1.5rem -->
<div class="border-b-2">  <!-- border-bottom-width: 2px -->
```

2. Negative Values

Use `-` prefix for negative margins:

```
<div class="-mt-4">   <!-- margin-top: -1rem -->
<div class="-ml-8">   <!-- margin-left: -2rem -->
```

Note: Only margins support negative values (not padding).

3. Value Scale

Most utilities use a numeric scale. The number roughly corresponds to `0.25rem` increments:

```
0   = 0
1   = 0.25rem  (4px)
2   = 0.5rem   (8px)
3   = 0.75rem  (12px)
4   = 1rem     (16px)
5   = 1.25rem  (20px)
6   = 1.5rem   (24px)
8   = 2rem     (32px)
10  = 2.5rem   (40px)
12  = 3rem     (48px)
16  = 4rem     (64px)
20  = 5rem     (80px)
24  = 6rem     (96px)
```

Why this scale? It creates visual harmony. Each step is intentional, preventing random pixel values.

Spacing System (Margin & Padding)

Understanding the Spacing Scale

Tailwind's spacing scale is based on a **4px base unit**. This creates consistent, harmonious spacing:

```
Visual representation of spacing scale:
```

```
p-0   |
p-1   |▌
p-2   |▌▌
p-3   |▌▌▌
p-4   |▌▌▌▌          ← Most commonly used
p-6   |▌▌▌▌▌▌
p-8   |▌▌▌▌▌▌▌▌      ← Very commonly used
p-12  |▌▌▌▌▌▌▌▌▌▌▌▌
p-16  |▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌
```

Margin Utilities

Syntax: `m[direction]-[size]`

```
<!-- All sides -->
<div class="m-4">        <!-- margin: 1rem -->

<!-- Individual sides -->
<div class="mt-2">       <!-- margin-top: 0.5rem -->
<div class="mr-4">       <!-- margin-right: 1rem -->
<div class="mb-6">       <!-- margin-bottom: 1.5rem -->
<div class="ml-8">       <!-- margin-left: 2rem -->

<!-- Horizontal (x-axis) -->
<div class="mx-4">       <!-- margin-left: 1rem; margin-right:
1rem -->
<div class="mx-auto">    <!-- margin-left: auto; margin-right:
auto → Centers block elements -->

<!-- Vertical (y-axis) -->
<div class="my-8">       <!-- margin-top: 2rem; margin-bottom:
2rem -->

<!-- Negative margins -->
```

```
<div class="-mt-4">      <!-- margin-top: -1rem -->
<div class="-ml-2">      <!-- margin-left: -0.5rem -->
```

Real-world example:

```
<!-- Card with spacing -->
<div class="bg-white p-6 mb-4">
  <h2 class="text-xl font-bold mb-2">Card Title</h2>
  <p class="text-gray-600">Card content here</p>
</div>
```

Comparison to CSS:

```
/* Vanilla CSS */
.card {
  background-color: white;
  padding: 1.5rem;
  margin-bottom: 1rem;
}

.card h2 {
  font-size: 1.25rem;
  font-weight: 700;
  margin-bottom: 0.5rem;
}
```

Padding Utilities

Syntax: `p[direction]-[size]`

```
<!-- All sides -->
<div class="p-4">        <!-- padding: 1rem -->

<!-- Individual sides -->
```

```
<div class="pt-2">       <!-- padding-top: 0.5rem -->
<div class="pr-4">       <!-- padding-right: 1rem -->
<div class="pb-6">       <!-- padding-bottom: 1.5rem -->
<div class="pl-8">       <!-- padding-left: 2rem -->

<!-- Horizontal -->
<div class="px-6">       <!-- padding-left: 1.5rem; padding-
right: 1.5rem -->

<!-- Vertical -->
<div class="py-3">       <!-- padding-top: 0.75rem; padding-
bottom: 0.75rem -->
```

Button example showing padding patterns:

```
<!-- Small button -->
<button class="px-3 py-1 text-sm">
  Small
</button>

<!-- Medium button (most common) -->
<button class="px-4 py-2">
  Medium
</button>

<!-- Large button -->
<button class="px-6 py-3 text-lg">
  Large
</button>
```

Space Between Utilities

For spacing between sibling elements, use `space-x-*` and `space-y-*`:

```
<!-- Horizontal spacing between items -->
<div class="flex space-x-4">
  <button>One</button>
  <button>Two</button>
  <button>Three</button>
</div>

<!-- Equivalent to: -->
<div class="flex">
  <button class="mr-4">One</button>
  <button class="mr-4">Two</button>
  <button>Three</button>
</div>

<!-- Vertical spacing (for stacked elements) -->
<div class="space-y-4">
  <p>Paragraph one</p>
  <p>Paragraph two</p>
  <p>Paragraph three</p>
</div>
```

This is cleaner than adding margins to each child!

Sizing Utilities (Width & Height)

Width Utilities

Tailwind provides multiple ways to set width:

1. Fixed Widths (using spacing scale)

```
<div class="w-4">    <!-- width: 1rem (16px) -->
<div class="w-12">   <!-- width: 3rem (48px) -->
```

```
<div class="w-64">   <!-- width: 16rem (256px) -->
<div class="w-96">   <!-- width: 24rem (384px) -->
```

2. Percentage Widths

```
<div class="w-1/2">    <!-- width: 50% -->
<div class="w-1/3">    <!-- width: 33.333% -->
<div class="w-2/3">    <!-- width: 66.666% -->
<div class="w-1/4">    <!-- width: 25% -->
<div class="w-3/4">    <!-- width: 75% -->
<div class="w-full">   <!-- width: 100% -->
```

Real-world layout example:

```
<!-- Two-column layout -->
<div class="flex gap-4">
  <div class="w-2/3 bg-gray-100 p-4">
    Main content (66%)
  </div>
  <div class="w-1/3 bg-gray-200 p-4">
    Sidebar (33%)
  </div>
</div>
```

3. Viewport Width

```
<div class="w-screen">   <!-- width: 100vw -->
```

4. Content-based Width

```
<div class="w-auto">     <!-- width: auto (default) -->
<div class="w-min">      <!-- width: min-content -->
<div class="w-max">      <!-- width: max-content -->
<div class="w-fit">      <!-- width: fit-content -->
```

When to use each:

```
<!-- w-auto: Let content determine width -->
<button class="w-auto px-4 py-2">
  Dynamic Width Button
</button>

<!-- w-full: Fill parent container -->
<input class="w-full px-3 py-2 border" placeholder="Full width
input">

<!-- w-min: Shrink to minimum content width -->
<div class="w-min whitespace-nowrap">
  This won't wrap
</div>

<!-- w-max: Expand to maximum content width -->
<div class="w-max">
  As wide as content needs
</div>
```

Height Utilities

Height follows the same patterns as width:

```
<!-- Fixed heights -->
<div class="h-8">       <!-- height: 2rem (32px) -->
<div class="h-16">      <!-- height: 4rem (64px) -->
<div class="h-64">      <!-- height: 16rem (256px) -->

<!-- Percentage heights -->
<div class="h-full">    <!-- height: 100% -->
<div class="h-1/2">     <!-- height: 50% -->
```

```
<!-- Viewport height -->
<div class="h-screen">  <!-- height: 100vh -->

<!-- Min/Max height -->
<div class="min-h-screen">  <!-- min-height: 100vh → Great for hero sections -->
<div class="max-h-96">      <!-- max-height: 24rem → Prevent overflow -->
```

Common pattern - Full-height hero:

```
<div class="min-h-screen flex items-center justify-center">
  <div class="text-center">
    <h1>Hero Content</h1>
  </div>
</div>
```

Comparison to CSS:

```
/* Vanilla CSS */
.hero {
  min-height: 100vh;
  display: flex;
  align-items: center;
  justify-content: center;
}
```

Max-Width for Readable Content

Tailwind provides max-width utilities for constraining content:

```
<div class="max-w-xs">    <!-- max-width: 20rem (320px) -->
<div class="max-w-sm">    <!-- max-width: 24rem (384px) -->
<div class="max-w-md">    <!-- max-width: 28rem (448px) -->
```

```
<div class="max-w-lg">    <!-- max-width: 32rem (512px) -->
<div class="max-w-xl">    <!-- max-width: 36rem (576px) -->
<div class="max-w-2xl">   <!-- max-width: 42rem (672px) -->
<div class="max-w-4xl">   <!-- max-width: 56rem (896px) -->
<div class="max-w-7xl">   <!-- max-width: 80rem (1280px) -->
```

Readable content container:

```
<article class="max-w-2xl mx-auto px-4">
  <h1>Blog Post Title</h1>
  <p>Content is constrained to a comfortable reading width and
centered.</p>
</article>
```

This is a **very common pattern** for blog posts and articles!

Display Utilities

These map directly to CSS `display` property:

```
<div class="block">        <!-- display: block -->
<div class="inline-block">  <!-- display: inline-block -->
<div class="inline">       <!-- display: inline -->
<div class="flex">         <!-- display: flex -->
<div class="inline-flex">  <!-- display: inline-flex -->
<div class="grid">         <!-- display: grid -->
<div class="inline-grid">  <!-- display: inline-grid -->
<div class="hidden">       <!-- display: none -->
```

Practical Examples

1. Hiding elements:

```
<!-- Hide on all screens -->
<div class="hidden">
  This won't display
</div>

<!-- Show/hide responsively (we'll cover this more in Module 6) -
->
<div class="hidden md:block">
  Hidden on mobile, visible on tablets+
</div>
```

2. Inline-block for buttons:

```
<div>
  <button class="inline-block px-4 py-2 bg-blue-500 text-white">
    Button 1
  </button>
  <button class="inline-block px-4 py-2 bg-blue-500 text-white">
    Button 2
  </button>
</div>
```

3. Flex for layouts:

```
<div class="flex justify-between items-center">
  <div>Logo</div>
  <nav>Menu</nav>
</div>
```

Position Utilities

Position utilities map to CSS positioning:

```
<div class="static">      <!-- position: static (default) -->
<div class="relative">    <!-- position: relative -->
<div class="absolute">    <!-- position: absolute -->
<div class="fixed">       <!-- position: fixed -->
<div class="sticky">      <!-- position: sticky -->
```

Positioning Values

Combined with position, you use inset utilities:

```
<!-- Individual sides -->
<div class="top-0">       <!-- top: 0 -->
<div class="right-4">     <!-- right: 1rem -->
<div class="bottom-0">    <!-- bottom: 0 -->
<div class="left-0">      <!-- left: 0 -->

<!-- All sides at once -->
<div class="inset-0">     <!-- top: 0; right: 0; bottom: 0; left:
0 -->

<!-- Horizontal or vertical -->
<div class="inset-x-0">   <!-- left: 0; right: 0 -->
<div class="inset-y-0">   <!-- top: 0; bottom: 0 -->
```

Real-World Examples

1. Centered overlay (modal backdrop):

```
<div class="fixed inset-0 bg-black bg-opacity-50 flex items-
center justify-center">
```

```
  <div class="bg-white p-8 rounded-lg">
    Modal content
  </div>
</div>
```

Breaking it down:

- `fixed` - Stay in place when scrolling
- `inset-0` - Cover entire viewport (top:0, right:0, bottom:0, left:0)
- `bg-black bg-opacity-50` - Semi-transparent black background
- `flex items-center justify-center` - Center the modal

2. Notification badge:

```
<div class="relative inline-block">
  <button class="px-4 py-2 bg-blue-500 text-white">
    Inbox
  </button>
  <span class="absolute -top-2 -right-2 bg-red-500 text-white
text-xs rounded-full h-5 w-5 flex items-center justify-center">
    3
  </span>
</div>
```

3. Sticky header:

```
<header class="sticky top-0 bg-white shadow z-10">
  <nav class="max-w-7xl mx-auto px-4 py-4">
    Navigation
  </nav>
</header>
```

Z-Index

Control stacking order:

```
<div class="z-0">     <!-- z-index: 0 -->
<div class="z-10">    <!-- z-index: 10 -->
<div class="z-20">    <!-- z-index: 20 -->
<div class="z-30">    <!-- z-index: 30 -->
<div class="z-40">    <!-- z-index: 40 -->
<div class="z-50">    <!-- z-index: 50 -->
<div class="z-auto">  <!-- z-index: auto -->
```

Common z-index pattern:

```
<!-- Page layers from bottom to top -->
<main class="z-0">Content</main>
<header class="z-10">Sticky header</header>
<div class="z-20">Dropdown menu</div>
<div class="z-30">Modal backdrop</div>
<div class="z-40">Modal content</div>
<div class="z-50">Tooltip</div>
```

Understanding the Spacing Scale Deep Dive

Let's visualize why Tailwind's spacing scale matters:

```
❌ With arbitrary values:
margin: 13px
padding: 27px
gap: 19px
→ Inconsistent, hard to maintain

✅ With Tailwind's scale:
```

```
m-3  (12px)
p-6  (24px)
gap-4 (16px)
→ Consistent, harmonious
```

The Scale in Practice

Micro spacing (0-2): For tight spacing

```
<div class="space-x-1">   <!-- 4px between items -->
<div class="space-x-2">   <!-- 8px between items -->
```

Small spacing (2-4): Most common for UI elements

```
<button class="px-3 py-1.5">  <!-- Compact button -->
<button class="px-4 py-2">    <!-- Standard button -->
```

Medium spacing (4-8): For section spacing

```
<section class="py-8">        <!-- 32px vertical padding -->
<div class="mb-6">            <!-- 24px margin bottom -->
```

Large spacing (12-24): For major sections

```
<section class="py-16">       <!-- 64px vertical padding -->
<section class="py-24">       <!-- 96px vertical padding -->
```

Visual spacing example:

```
<!-- Well-spaced card -->
<div class="bg-white rounded-lg shadow-lg p-6 space-y-4">
```

```
  <h2 class="text-2xl font-bold">
    Card Title
  </h2>

  <p class="text-gray-600">
    This card uses consistent spacing throughout.
  </p>

  <div class="flex space-x-2">
    <button class="px-4 py-2 bg-blue-500 text-white rounded">
      Action
    </button>
    <button class="px-4 py-2 bg-gray-200 text-gray-700 rounded">
      Cancel
    </button>
  </div>

</div>
```

Common Mistakes to Avoid

1. Mixing spacing systems

```
<!-- ❌ Inconsistent -->
<div class="p-3 mb-5 mt-7">

<!-- ✅ Use multiples of the base scale -->
<div class="p-4 mb-4 mt-8">
```

2. Forgetting mx-auto for centering

```
<!-- ❌ Won't center -->
<div class="w-1/2">Content</div>

<!-- ✅ Centers block element -->
<div class="w-1/2 mx-auto">Content</div>
```

3. Using padding when margin is better (or vice versa)

```
<!-- ❌ Pushes content inside -->
<button class="pt-20">Weird spacing</button>

<!-- ✅ Creates space outside -->
<button class="mt-20">Better spacing</button>
```

4. *Not using space- utilities**

```
<!-- ❌ Repetitive -->
<div class="flex">
  <div class="mr-4">One</div>
  <div class="mr-4">Two</div>
  <div>Three</div>
</div>

<!-- ✅ Cleaner -->
<div class="flex space-x-4">
  <div>One</div>
  <div>Two</div>
  <div>Three</div>
</div>
```

Key Takeaways

✅ Utility classes follow predictable **naming patterns**: `property-value`
✅ **Directional modifiers** (t, r, b, l, x, y) control which sides are affected
✅ The **spacing scale** (0, 1, 2, 4, 8...) creates visual consistency
✅ Use `space-x-*` and `space-y-*` for spacing between siblings
✅ **Width/height** have multiple scales: fixed, percentage, viewport
✅ `max-w-*` utilities are perfect for **readable content widths**
✅ **Position utilities** + **inset values** = powerful layout control
✅ Each class does **one thing** - combine them for complex designs

---

Practice Assignments

Assignment 1 (Easy): Spacing Experiments

Create a page with 5 boxes. Each box should have:

- A different background color
- Different padding values (p-2, p-4, p-6, p-8, p-12)
- Different margin values
- Observe how the spacing changes visually

Assignment 2 (Easy): Button Variations

Create 4 buttons with different sizes:

- Extra small: `px-2 py-1 text-xs`
- Small: `px-3 py-1.5 text-sm`
- Medium: `px-4 py-2`
- Large: `px-6 py-3 text-lg`

Add colors and observe the spacing patterns.

Assignment 3 (Medium): Card Component

Build a card with proper spacing:

- 6 units of padding
- Heading with 2 units margin bottom
- Paragraph with 4 units margin bottom
- Button with 4 units horizontal padding and 2 units vertical
- Use `space-y-*` where appropriate

Assignment 4 (Medium): Two-Column Layout

Create a responsive layout:

- Container with max-w-6xl centered with mx-auto
- Two columns: 2/3 width and 1/3 width
- Gap between columns
- Each column has padding and a background

Assignment 5 (Hard): Complex Positioning

Build a notification card with:

- Relative positioned container
- Absolute positioned close button (top-right corner)
- Absolute positioned icon (top-left, slightly outside the card using negative positioning)
- Proper z-index layering
- The icon should be a circle that overlaps the card edge

Hint: Use `relative`, `absolute`, negative top/left values, and z-index utilities.

Ready for **Module 3: Colors & Backgrounds**? Let me know!

12/30/2025, 9:57:12 AM

MODULE 3: COLORS & BACKGROUNDS

Color Palette System (50-900 Shades)

Tailwind comes with a **comprehensive color palette** where each color has 10 shades ranging from very light (50) to very dark (900). This creates consistency and prevents "random color syndrome."

The Shade Scale

```
Visual representation of blue shades:

50  ██████████ (Lightest - almost white)
100 ██████████
200 ██████████
300 ██████████
400 ██████████
500 ██████████ ← Base color (most saturated)
600 ██████████
700 ██████████
800 ██████████
900 ██████████ (Darkest - almost black)
```

The Pattern:

- **50-200**: Very light tints (backgrounds, subtle highlights)
- **300-400**: Light to medium (borders, secondary elements)
- **500**: The "true" color (buttons, primary actions)

- **600-700**: Darker shades (hover states, emphasis)
- **800-900**: Very dark (text on light backgrounds, headers)

Available Color Families

```
slate      gray       zinc       neutral    stone
red        orange     amber      yellow     lime
green      emerald    teal       cyan       sky
blue       indigo     violet     purple     fuchsia
pink       rose
```

Plus special colors:

```
black      white      transparent    current
```

Text Colors

Syntax: `text-[color]-[shade]`

```
<!-- Basic text colors -->
<p class="text-gray-900">Very dark gray text (almost black)</p>
<p class="text-gray-700">Medium dark gray</p>
<p class="text-gray-500">Medium gray</p>
<p class="text-gray-300">Light gray</p>

<!-- Colored text -->
<p class="text-blue-600">Blue text</p>
<p class="text-red-500">Red text</p>
<p class="text-green-700">Dark green text</p>

<!-- Special colors -->
```

```
<p class="text-black">Pure black</p>
<p class="text-white">Pure white</p>
```

Common Text Color Patterns

1. Hierarchical text (importance levels):

```
<article class="bg-white p-6">
  <h1 class="text-3xl font-bold text-gray-900">
    Primary Heading (darkest)
  </h1>

  <h2 class="text-xl font-semibold text-gray-800">
    Secondary Heading
  </h2>

  <p class="text-gray-700">
    Body text (readable but not too dark)
  </p>

  <p class="text-sm text-gray-500">
    Muted text (captions, metadata)
  </p>

  <p class="text-xs text-gray-400">
    Very subtle text (disclaimers)
  </p>
</article>
```

This creates visual hierarchy! Darker = more important.

2. Status colors:

```
<div class="space-y-2">
  <p class="text-green-600">✓ Success message</p>
  <p class="text-red-600">✗ Error message</p>
  <p class="text-yellow-600">⚠ Warning message</p>
  <p class="text-blue-600">ℹ Info message</p>
</div>
```

3. Links:

```
<a href="#" class="text-blue-600 hover:text-blue-800">
  Click here
</a>

<!-- With underline -->
<a href="#" class="text-blue-600 hover:text-blue-800 underline">
  Underlined link
</a>
```

Background Colors

Syntax: `bg-[color]-[shade]`

```
<!-- Solid backgrounds -->
<div class="bg-white">White background</div>
<div class="bg-gray-100">Light gray background</div>
<div class="bg-blue-500">Blue background</div>
<div class="bg-red-600">Darker red background</div>

<!-- Transparent -->
<div class="bg-transparent">No background</div>
```

Background Color Patterns

1. Page backgrounds (usually very light):

```
<body class="bg-gray-50">
  <!-- Very subtle gray, easier on eyes than pure white -->
</body>

<body class="bg-slate-100">
  <!-- Slightly cooler gray -->
</body>
```

2. Card/panel backgrounds:

```
<!-- Card on gray background -->
<div class="bg-gray-50 min-h-screen p-8">
  <div class="bg-white rounded-lg shadow p-6">
    <h2 class="text-xl font-bold text-gray-900">Card Title</h2>
    <p class="text-gray-600">Card content</p>
  </div>
</div>
```

The contrast creates depth!

3. Alert/notification backgrounds:

```
<!-- Success alert -->
<div class="bg-green-100 border border-green-400 text-green-700
px-4 py-3 rounded">
  <strong>Success!</strong> Your changes have been saved.
</div>

<!-- Error alert -->
<div class="bg-red-100 border border-red-400 text-red-700 px-4
py-3 rounded">
```

```
  <strong>Error!</strong> Something went wrong.
</div>

<!-- Info alert -->
<div class="bg-blue-100 border border-blue-400 text-blue-700 px-4
py-3 rounded">
  <strong>Info:</strong> Please read this carefully.
</div>
```

Pattern: Light background (100) + medium border (400) + dark text (700)

4. Button backgrounds:

```
<!-- Primary button -->
<button class="bg-blue-600 hover:bg-blue-700 text-white px-4 py-2
rounded">
  Primary Action
</button>

<!-- Secondary button -->
<button class="bg-gray-200 hover:bg-gray-300 text-gray-800 px-4
py-2 rounded">
  Secondary Action
</button>

<!-- Danger button -->
<button class="bg-red-600 hover:bg-red-700 text-white px-4 py-2
rounded">
  Delete
</button>
```

Notice the hover pattern: Typically 100 shades darker on hover.

Border Colors

Syntax: `border-[color]-[shade]`

```
<!-- You need border-width first! -->
<div class="border border-gray-300">Default border</div>
<div class="border-2 border-blue-500">Thicker blue border</div>
<div class="border-4 border-red-600">Very thick red border</div>

<!-- Individual sides -->
<div class="border-b border-gray-200">Bottom border only</div>
<div class="border-t-2 border-blue-400">Top border, 2px</div>
```

Border Patterns

1. Subtle dividers:

```
<!-- List with dividers -->
<ul class="divide-y divide-gray-200">
  <li class="py-3">Item 1</li>
  <li class="py-3">Item 2</li>
  <li class="py-3">Item 3</li>
</ul>
```

`divide-y` adds borders between siblings automatically!

2. Input borders:

```
<!-- Default input -->
<input
  class="border border-gray-300 rounded px-3 py-2 w-full"
  placeholder="Enter text"
>
```

```
<!-- Focused input (we'll cover this more in Module 7) -->
<input
  class="border border-gray-300 rounded px-3 py-2 w-full
focus:border-blue-500 focus:outline-none"
  placeholder="Focused border becomes blue"
>
```

3. Card borders:

```
<!-- Subtle card border -->
<div class="border border-gray-200 rounded-lg p-6">
  <h3 class="font-bold">Card with border</h3>
</div>

<!-- Colored accent border -->
<div class="border-l-4 border-blue-500 bg-blue-50 p-4">
  <p class="text-blue-800">Note with left accent</p>
</div>
```

Opacity Modifiers

Tailwind lets you modify the opacity of colors using the `/` syntax:

Syntax: `[utility]-[color]-[shade]/[opacity]`

```
<!-- Background opacity -->
<div class="bg-blue-500/100">100% opacity (default)</div>
<div class="bg-blue-500/75">75% opacity</div>
<div class="bg-blue-500/50">50% opacity</div>
<div class="bg-blue-500/25">25% opacity</div>
<div class="bg-blue-500/10">10% opacity</div>
<div class="bg-blue-500/0">0% opacity (invisible)</div>
```

```
<!-- Text opacity -->
<p class="text-gray-900/100">Fully opaque text</p>
<p class="text-gray-900/60">60% opacity text</p>
<p class="text-gray-900/30">30% opacity text</p>

<!-- Border opacity -->
<div class="border-2 border-gray-500/50">Semi-transparent
border</div>
```

Real-World Opacity Examples

1. Overlay (modal backdrop):

```
<div class="fixed inset-0 bg-black/50 flex items-center justify-
center">
  <div class="bg-white p-8 rounded-lg">
    Modal content
  </div>
</div>
```

`bg-black/50` = 50% transparent black overlay

2. Disabled state:

```
<button class="bg-blue-500 text-white px-4 py-2 rounded opacity-
50 cursor-not-allowed">
  Disabled Button
</button>
```

3. Subtle backgrounds:

```
<!-- Very subtle blue tint -->
<div class="bg-blue-500/5 p-4">
```

```
  <p>Content with barely visible blue background</p>
</div>
```

This is powerful! You can create subtle tints without defining new colors.

Gradients

Tailwind supports CSS gradients with utility classes.

Linear Gradients

Syntax:

1. Set gradient direction: `bg-gradient-to-[direction]`
2. Set start color: `from-[color]-[shade]`
3. Set end color: `to-[color]-[shade]`
4. Optional middle: `via-[color]-[shade]`

```
<!-- Left to right -->
<div class="bg-gradient-to-r from-blue-500 to-purple-500 p-8">
  Blue to purple gradient
</div>

<!-- Top to bottom -->
<div class="bg-gradient-to-b from-gray-50 to-gray-200 p-8">
  Subtle gray gradient
</div>

<!-- Diagonal -->
<div class="bg-gradient-to-br from-pink-500 to-orange-500 p-8">
  Bottom-right diagonal gradient
</div>
```

```
<!-- Three colors with via -->
<div class="bg-gradient-to-r from-blue-500 via-purple-500 to-pink-500 p-8">
  Three-color gradient
</div>
```

Gradient Directions

```
to-t    → Top
to-tr   → Top-right
to-r    → Right
to-br   → Bottom-right (diagonal)
to-b    → Bottom
to-bl   → Bottom-left
to-l    → Left
to-tl   → Top-left
```

Practical Gradient Examples

1. Hero section gradient:

```
<div class="bg-gradient-to-br from-blue-600 via-blue-700 to-purple-800 min-h-screen flex items-center justify-center">
  <div class="text-center text-white">
    <h1 class="text-5xl font-bold mb-4">Welcome</h1>
    <p class="text-xl">Beautiful gradient background</p>
  </div>
</div>
```

2. Button with gradient:

```
<button class="bg-gradient-to-r from-blue-500 to-blue-600
hover:from-blue-600 hover:to-blue-700 text-white px-6 py-3
rounded-lg font-semibold">
  Gradient Button
</button>
```

3. Text gradient (requires extra trick):

```
<h1 class="text-6xl font-bold bg-gradient-to-r from-blue-500 to-
purple-600 bg-clip-text text-transparent">
  Gradient Text
</h1>
```

Breaking down text gradient:

- `bg-gradient-to-r from-blue-500 to-purple-600` - Create gradient
- `bg-clip-text` - Clip background to text shape
- `text-transparent` - Make text transparent so gradient shows through

4. Subtle overlay gradient:

```
<div class="relative">
  <img src="hero.jpg" class="w-full h-96 object-cover">

  <!-- Gradient overlay -->
  <div class="absolute inset-0 bg-gradient-to-t from-black/70 to-
transparent"></div>

  <!-- Content on top -->
  <div class="absolute bottom-8 left-8 text-white">
    <h2 class="text-3xl font-bold">Image Title</h2>
  </div>
</div>
```

This creates a **dark gradient at the bottom** making white text readable!

Custom Colors in Config

Sometimes you need brand colors not in Tailwind's default palette.

Extending the Theme

Open `tailwind.config.js`:

```
/** @type {import('tailwindcss').Config} */
export default {
  content: ["./src/**/*.{html,js}"],
  theme: {
    extend: {
      colors: {
        // Add a single color
        'brand': '#5B21B6',

        // Add a color with shades (recommended)
        'brand': {
          50: '#F5F3FF',
          100: '#EDE9FE',
          200: '#DDD6FE',
          300: '#C4B5FD',
          400: '#A78BFA',
          500: '#8B5CF6',  // Base shade
          600: '#7C3AED',
          700: '#6D28D9',
          800: '#5B21B6',
          900: '#4C1D95',
        },
      },
```

```
    },
  },
  plugins: [],
}
```

Now use it in HTML:

```
<div class="bg-brand-500 text-white p-4">
  Using custom brand color
</div>

<button class="bg-brand-600 hover:bg-brand-700 text-white px-4
py-2">
  Brand Button
</button>
```

Generating Color Shades

You can use online tools to generate shades from a single color:

- **UI Colors**: uicolors.app
- **Tailwind Shades**: tailwindshades.com

Example: Your brand color is `#FF6B6B` (coral red). Use a tool to generate:

```
colors: {
  'coral': {
    50: '#FFF5F5',
    100: '#FFE3E3',
    200: '#FFC9C9',
    300: '#FFA8A8',
    400: '#FF8787',
    500: '#FF6B6B',  // Your original color
    600: '#FA5252',
```

```
    700: '#F03E3E',
    800: '#E03131',
    900: '#C92A2A',
  },
}
```

Replacing Default Colors

If you want to **replace** instead of extend:

```
theme: {
  colors: {
    // This replaces ALL default colors
    transparent: 'transparent',
    current: 'currentColor',
    'white': '#ffffff',
    'black': '#000000',
    'gray': {
      100: '#f7fafc',
      // ... your custom gray scale
    },
    'blue': {
      // ... your custom blue scale
    },
  },
}
```

⚠️ **Warning:** This removes all default Tailwind colors. Usually you want `extend` instead.

Color Comparison: Vanilla CSS vs Tailwind

Vanilla CSS:

```
/* Define variables */
:root {
  --color-primary: #3b82f6;
  --color-primary-dark: #2563eb;
  --color-text: #1f2937;
  --color-text-muted: #6b7280;
}

.button {
  background-color: var(--color-primary);
  color: white;
}

.button:hover {
  background-color: var(--color-primary-dark);
}

.heading {
  color: var(--color-text);
}

.caption {
  color: var(--color-text-muted);
}
```

Tailwind:

```
<!-- Colors are inline, but systematic -->
<button class="bg-blue-500 hover:bg-blue-700 text-white">
  Button
</button>
```

```
<h1 class="text-gray-900">Heading</h1>
<p class="text-gray-600">Caption</p>
```

Tailwind's advantage: You don't define variables, the design system is built-in. But you can still customize via config if needed.

Common Mistakes to Avoid

1. Using too many different shades

```
<!-- ❌ Inconsistent - too many shades -->
<div>
  <p class="text-gray-500">Text 1</p>
  <p class="text-gray-550">Text 2</p> <!-- This doesn't exist! --
>
  <p class="text-gray-600">Text 3</p>
  <p class="text-gray-650">Text 4</p> <!-- This doesn't exist! --
>
</div>

<!-- ✅ Stick to the scale -->
<div>
  <p class="text-gray-900">Primary text</p>
  <p class="text-gray-700">Secondary text</p>
  <p class="text-gray-500">Muted text</p>
</div>
```

2. Forgetting text color on colored backgrounds

```
<!-- ❌ Dark text on dark background (invisible!) -->
<div class="bg-blue-900">
  <p class="text-gray-900">Can't read this!</p>
```

```
</div>

<!-- ✅ Light text on dark background -->
<div class="bg-blue-900">
  <p class="text-white">Readable!</p>
</div>
```

3. Not considering contrast (accessibility)

```
<!-- ❌ Poor contrast -->
<button class="bg-yellow-300 text-yellow-100">
  Hard to read
</button>

<!-- ✅ Good contrast -->
<button class="bg-yellow-500 text-gray-900">
  Easy to read
</button>
```

Tip: Aim for at least 4.5:1 contrast ratio for text (WCAG AA standard).

4. Using gradients everywhere

```
<!-- ❌ Too much -->
<div class="bg-gradient-to-r from-pink-500 to-purple-500">
  <h1 class="bg-gradient-to-r from-blue-500 to-green-500 bg-clip-
text text-transparent">
    Gradient overload!
  </h1>
</div>

<!-- ✅ Subtle and purposeful -->
<div class="bg-gradient-to-b from-gray-50 to-white">
```

```
  <h1 class="text-gray-900">Clean and readable</h1>
</div>
```

Key Takeaways

✅ Each color has **10 shades** (50-900) for consistency
✅ **50-200** = backgrounds, **500** = primary, **800-900** = text
✅ Use **opacity modifiers** (`/50`, `/75`) for transparent colors
✅ **Text hierarchy**: darker text = more important
✅ **Gradients**: `bg-gradient-to-[dir]` + `from-` + `to-` (+ optional `via-`)
✅ **Custom colors** go in `tailwind.config.js` under `theme.extend.colors`
✅ Always consider **color contrast** for accessibility
✅ Stick to the **color scale** - don't use arbitrary values

Practice Assignments

Assignment 1 (Easy): Color Palette Exploration

Create a page showing all shades of a single color (e.g., blue). Create 10 divs, each with:

- Background: `bg-blue-50` through `bg-blue-900`
- Text showing the shade number
- Appropriate text color (white for dark backgrounds, dark for light backgrounds)

Assignment 2 (Easy): Alert Components

Create 4 alert boxes for different states:

- Success (green)

- Error (red)
- Warning (yellow)
- Info (blue)

Each should have: light background (100), medium border (400), dark text (700).

Assignment 3 (Medium): Gradient Hero Section

Build a full-screen hero section with:

- A gradient background (3 colors using `via`)
- White text centered vertically and horizontally
- A heading and subheading
- A button with a different gradient
- The button should have a hover state with darker gradients

Assignment 4 (Medium): Card with Hover State

Create a card that changes colors on hover:

- Default: white background, gray border
- Hover: light blue background, blue border
- Text color should adjust accordingly
- Include a heading, paragraph, and button

Hint: Use `hover:` prefix (we'll dive deeper into this in Module 7).

Assignment 5 (Hard): Custom Brand Colors

1. Choose your own brand color (any hex code)
2. Use an online tool to generate a full shade palette (50-900)
3. Add it to `tailwind.config.js`
4. Build a landing page using ONLY your custom brand colors + grays
5. Include: navigation, hero section, feature cards, and footer

Extra challenge: Make the text gradient heading using your custom colors.

Ready for **Module 4: Typography**? Let me know!

1/2/2026, 1:40:15 PM

MODULE 4: TYPOGRAPHY

Font Family, Size, Weight

Font Family

Tailwind provides three default font stacks:

```
<!-- Sans-serif (default) -->
<p class="font-sans">
  System UI, Segoe UI, Roboto, Helvetica, Arial, sans-serif
</p>

<!-- Serif -->
<p class="font-serif">
  Georgia, Cambria, "Times New Roman", Times, serif
</p>

<!-- Monospace (code) -->
<p class="font-mono">
  Menlo, Monaco, Consolas, "Courier New", monospace
</p>
```

When to use each:

- `font-sans` - Body text, UI elements, modern websites (DEFAULT)

- `font-serif` - Blog posts, articles, formal documents, print-style layouts
- `font-mono` - Code snippets, technical documentation, fixed-width needs

Example usage:

```
<article class="max-w-2xl mx-auto p-8">
  <!-- Article title in serif for elegance -->
  <h1 class="font-serif text-4xl font-bold mb-4">
    The Art of Typography
  </h1>

  <!-- Body text in sans-serif for readability -->
  <p class="font-sans text-gray-700 mb-4">
    Typography is the art and technique of arranging type.
  </p>

  <!-- Code example in monospace -->
  <pre class="font-mono bg-gray-100 p-4 rounded">
    const greeting = "Hello World";
  </pre>
</article>
```

Font Size

Tailwind uses a **type scale** that creates harmonious size relationships:

```
<p class="text-xs">Extra small (0.75rem / 12px)</p>
<p class="text-sm">Small (0.875rem / 14px)</p>
<p class="text-base">Base (1rem / 16px) - DEFAULT</p>
<p class="text-lg">Large (1.125rem / 18px)</p>
<p class="text-xl">Extra large (1.25rem / 20px)</p>
<p class="text-2xl">2XL (1.5rem / 24px)</p>
```

```
<p class="text-3xl">3XL (1.875rem / 30px)</p>
<p class="text-4xl">4XL (2.25rem / 36px)</p>
<p class="text-5xl">5XL (3rem / 48px)</p>
<p class="text-6xl">6XL (3.75rem / 60px)</p>
<p class="text-7xl">7XL (4.5rem / 72px)</p>
<p class="text-8xl">8XL (6rem / 96px)</p>
<p class="text-9xl">9XL (8rem / 128px)</p>
```

Visual hierarchy guide:

```
text-9xl  ████████  Hero headlines (rare)
text-6xl  ██████    Page titles
text-4xl  ████      Section headings
text-2xl  ██        Subsection headings
text-xl   ▌         Lead paragraphs
text-base ▌         Body text (DEFAULT)
text-sm   ▓         Captions, labels
text-xs   ▒         Fine print, metadata
```

Common Typography Patterns

1. Blog post hierarchy:

```
<article class="max-w-2xl mx-auto">
  <!-- Main title -->
  <h1 class="text-4xl font-bold text-gray-900 mb-2">
    Understanding Typography
  </h1>

  <!-- Subtitle/lead -->
  <p class="text-xl text-gray-600 mb-8">
    A comprehensive guide to type on the web
  </p>
```

```
  <!-- Section heading -->
  <h2 class="text-2xl font-semibold text-gray-900 mb-4">
    Why Typography Matters
  </h2>

  <!-- Body text -->
  <p class="text-base text-gray-700 mb-4">
    Typography is the foundation of good design...
  </p>

  <!-- Subsection heading -->
  <h3 class="text-xl font-semibold text-gray-900 mb-3">
    Key Principles
  </h3>

  <!-- Metadata/caption -->
  <p class="text-sm text-gray-500">
    Published on January 2, 2026
  </p>
</article>
```

2. Landing page hero:

```
<div class="text-center py-20">
  <!-- Large hero headline -->
  <h1 class="text-6xl font-bold text-gray-900 mb-6">
    Build Faster
  </h1>

  <!-- Subheadline -->
  <p class="text-2xl text-gray-600 mb-8">
    Ship products your customers will love
  </p>

  <!-- Call to action text -->
```

```
  <button class="text-lg font-semibold">
    Get Started
  </button>
</div>
```

Font Weight

Control how bold your text appears:

```
<p class="font-thin">Thin (100)</p>
<p class="font-extralight">Extra Light (200)</p>
<p class="font-light">Light (300)</p>
<p class="font-normal">Normal (400) - DEFAULT</p>
<p class="font-medium">Medium (500)</p>
<p class="font-semibold">Semibold (600)</p>
<p class="font-bold">Bold (700)</p>
<p class="font-extrabold">Extra Bold (800)</p>
<p class="font-black">Black (900)</p>
```

Most commonly used weights:

- `font-normal` (400) - Body text
- `font-medium` (500) - Slightly emphasized text
- `font-semibold` (600) - Subheadings, buttons
- `font-bold` (700) - Headings, strong emphasis

Real-world example:

```
<div class="space-y-4">
  <!-- Primary heading -->
  <h1 class="text-3xl font-bold">
    Product Features
```

```
  </h1>

  <!-- Secondary heading -->
  <h2 class="text-xl font-semibold">
    Why Choose Us
  </h2>

  <!-- Body with emphasis -->
  <p class="font-normal">
    Our platform is <span class="font-semibold">fast</span>,
    <span class="font-semibold">reliable</span>, and
    <span class="font-semibold">secure</span>.
  </p>

  <!-- Button text -->
  <button class="font-semibold">
    Learn More
  </button>

  <!-- Fine print -->
  <p class="text-sm font-light text-gray-500">
    *Terms and conditions apply
  </p>
</div>
```

Comparison to vanilla CSS:

```
/* Vanilla CSS */
h1 {
  font-size: 1.875rem;
  font-weight: 700;
}

.lead {
  font-size: 1.25rem;
```

```
  font-weight: 500;
}
```

```
<!-- Tailwind -->
<h1 class="text-3xl font-bold">Heading</h1>
<p class="text-xl font-medium">Lead paragraph</p>
```

Text Alignment and Decoration

Text Alignment

```
<p class="text-left">Left aligned (default)</p>
<p class="text-center">Center aligned</p>
<p class="text-right">Right aligned</p>
<p class="text-justify">Justified text spreads evenly</p>
```

Common patterns:

```
<!-- Centered hero section -->
<div class="text-center py-16">
  <h1 class="text-5xl font-bold">Welcome</h1>
  <p class="text-xl text-gray-600">Centered subheading</p>
</div>

<!-- Article with justified text -->
<article class="max-w-2xl mx-auto">
  <p class="text-justify leading-relaxed">
    Lorem ipsum dolor sit amet...
  </p>
</article>
```

```html
<!-- Right-aligned metadata -->
<div class="flex justify-between">
  <span>Article Title</span>
  <span class="text-right text-sm text-gray-500">
    Jan 2, 2026
  </span>
</div>
```

Text Decoration

```html
<!-- Underline -->
<p class="underline">Underlined text</p>

<!-- Overline (line above) -->
<p class="overline">Overlined text</p>

<!-- Line through -->
<p class="line-through">Strikethrough text</p>

<!-- No decoration -->
<a href="#" class="no-underline">Link without underline</a>
```

Practical examples:

```html
<!-- Link styles -->
<a href="#" class="text-blue-600 hover:underline">
  Underline appears on hover
</a>

<!-- Pricing (show discount) -->
<div>
  <span class="line-through text-gray-500">$99</span>
  <span class="text-2xl font-bold text-green-600">$49</span>
</div>
```

```
<!-- Task list -->
<ul class="space-y-2">
  <li class="line-through text-gray-500">Completed task</li>
  <li>Incomplete task</li>
</ul>
```

Text Decoration Style & Color

```
<!-- Decoration style -->
<p class="underline decoration-solid">Solid underline</p>
<p class="underline decoration-double">Double underline</p>
<p class="underline decoration-dotted">Dotted underline</p>
<p class="underline decoration-dashed">Dashed underline</p>
<p class="underline decoration-wavy">Wavy underline</p>

<!-- Decoration color -->
<p class="underline decoration-blue-500">Blue underline</p>
<p class="underline decoration-red-500">Red underline</p>

<!-- Decoration thickness -->
<p class="underline decoration-1">Thin underline</p>
<p class="underline decoration-2">Medium underline</p>
<p class="underline decoration-4">Thick underline</p>
```

Creative example:

```
<h2 class="text-3xl font-bold underline decoration-wavy
decoration-4 decoration-pink-500">
  Creative Heading
</h2>
```

Line Height and Letter Spacing

Line Height (Leading)

Line height controls vertical spacing between lines of text:

```
<p class="leading-none">Line height: 1 (tight)</p>
<p class="leading-tight">Line height: 1.25</p>
<p class="leading-snug">Line height: 1.375</p>
<p class="leading-normal">Line height: 1.5 (DEFAULT)</p>
<p class="leading-relaxed">Line height: 1.625</p>
<p class="leading-loose">Line height: 2</p>
```

Visual representation:

```
leading-tight:
Line one is close
Line two is close

leading-normal:
Line one has normal spacing

Line two has normal spacing

leading-loose:
Line one is far apart



Line two is far apart
```

When to use different line heights:

```
<!-- Headlines: tighter line-height -->
<h1 class="text-5xl font-bold leading-tight">
```

```
  Big Headlines
  Look Better Tight
</h1>

<!-- Body text: normal to relaxed -->
<p class="text-base leading-relaxed">
  Body text is more readable with generous line spacing.
  This is especially true for longer paragraphs where
  readers need breathing room between lines.
</p>

<!-- Code blocks: snug to normal -->
<pre class="font-mono leading-snug">
  function hello() {
    return "world";
  }
</pre>

<!-- Fine print: normal -->
<p class="text-xs leading-normal">
  Terms and conditions apply
</p>
```

Real-world article example:

```
<article class="max-w-2xl mx-auto">
  <h1 class="text-4xl font-bold leading-tight mb-4">
    The Perfect Line Height
  </h1>

  <p class="text-xl text-gray-600 leading-relaxed mb-8">
    Learn how line spacing affects readability
  </p>

  <p class="text-base text-gray-700 leading-relaxed mb-4">
```

```
    Line height is one of the most important factors in
typography.
    Too tight and text feels cramped. Too loose and it loses
cohesion.
  </p>
</article>
```

Letter Spacing (Tracking)

Letter spacing controls horizontal space between characters:

```
<p class="tracking-tighter">Tighter (-0.05em)</p>
<p class="tracking-tight">Tight (-0.025em)</p>
<p class="tracking-normal">Normal (0em) - DEFAULT</p>
<p class="tracking-wide">Wide (0.025em)</p>
<p class="tracking-wider">Wider (0.05em)</p>
<p class="tracking-widest">Widest (0.1em)</p>
```

When to adjust letter spacing:

```
<!-- All caps headings: increase tracking -->
<h2 class="text-sm font-semibold uppercase tracking-wide text-
gray-500">
  Features
</h2>

<!-- Logo/brand text: custom spacing -->
<h1 class="text-3xl font-bold tracking-tight">
  BRAND
</h1>

<!-- Fine print: slight increase for readability -->
<p class="text-xs tracking-wide text-gray-600">
  © 2026 Company Name
```

```
</p>

<!-- Button text: slightly wider feels more open -->
<button class="tracking-wide uppercase text-sm font-semibold">
  Get Started
</button>
```

Real-world navigation example:

```
<nav class="flex space-x-8">
  <a href="#" class="text-sm uppercase tracking-wider font-semibold text-gray-700 hover:text-gray-900">
    Home
  </a>
  <a href="#" class="text-sm uppercase tracking-wider font-semibold text-gray-700 hover:text-gray-900">
    About
  </a>
  <a href="#" class="text-sm uppercase tracking-wider font-semibold text-gray-700 hover:text-gray-900">
    Contact
  </a>
</nav>
```

Text Transform and Overflow

Text Transform

```
<!-- Uppercase -->
<p class="uppercase">this becomes uppercase</p>

<!-- Lowercase -->
```

```
<p class="lowercase">THIS BECOMES LOWERCASE</p>

<!-- Capitalize (first letter of each word) -->
<p class="capitalize">this becomes This Becomes Capitalize</p>

<!-- Normal case -->
<p class="normal-case">Preserves Original Case</p>
```

Common patterns:

```
<!-- Section labels -->
<h3 class="text-xs uppercase tracking-wide font-semibold text-
gray-500 mb-4">
  Latest Articles
</h3>

<!-- Button text -->
<button class="uppercase text-sm tracking-wider font-bold">
  Shop Now
</button>

<!-- Form labels -->
<label class="text-sm font-medium capitalize">
  first name
</label>

<!-- Display user input as-is -->
<p class="normal-case">
  {userInput}
</p>
```

Text Overflow

Control how text behaves when it's too long:

```
<!-- Truncate with ellipsis -->
<p class="truncate">
  This is a very long line of text that will be cut off with an
ellipsis...
</p>

<!-- Text ellipsis (requires specific setup) -->
<p class="overflow-hidden text-ellipsis">
  Long text...
</p>

<!-- Clip (no ellipsis) -->
<p class="text-clip">
  Long text gets clipped
</p>
```

For truncate to work, you need a width constraint:

```
<!-- Card with truncated title -->
<div class="w-64 bg-white p-4 shadow rounded">
  <h3 class="font-bold truncate">
    This is a very long article title that would normally wrap
  </h3>
  <p class="text-sm text-gray-600">
    Preview text...
  </p>
</div>
```

Whitespace Control

```
<!-- Normal wrapping -->
<p class="whitespace-normal">
  Text wraps normally
```

```
</p>

<!-- No wrapping -->
<p class="whitespace-nowrap">
  This text will never wrap to a new line
</p>

<!-- Preserve whitespace and line breaks -->
<pre class="whitespace-pre">
  This    preserves    spaces
  and line breaks
</pre>

<!-- Preserve line breaks but collapse spaces -->
<p class="whitespace-pre-line">
  This    collapses    spaces
  but preserves
  line breaks
</p>

<!-- Preserve spaces, allow wrapping -->
<p class="whitespace-pre-wrap">
  This    preserves    spaces
  and allows wrapping
</p>
```

Real-world examples:

```
<!-- Price tag that shouldn't wrap -->
<span class="whitespace-nowrap">
  $1,299.99 USD
</span>

<!-- User-generated content (preserve formatting) -->
<div class="whitespace-pre-line bg-gray-50 p-4 rounded">
```

```
  {userComment}
</div>

<!-- Navigation links (prevent awkward wrapping) -->
<a href="#" class="whitespace-nowrap">
  View All Products →
</a>
```

Word Break

```
<!-- Normal breaking -->
<p class="break-normal">
  Veryveryveryverylongwordthatwontbreak
</p>

<!-- Break anywhere -->
<p class="break-all">
  Veryveryveryverylongwordthatwillbreak
</p>

<!-- Break at word boundaries, but break long words if needed -->
<p class="break-words">
  Veryveryveryverylongwordthatwillbreaknicely
</p>
```

Use cases:

```
<!-- URLs or emails that might overflow -->
<div class="max-w-sm p-4 bg-gray-100">
  <p class="break-all text-sm">
    https://example.com/very/long/url/path/that/needs/to/break
  </p>
</div>
```

```
<!-- User-generated content -->
<div class="break-words">
  <p>{userContent}</p>
</div>
```

Custom Fonts Setup

Using Google Fonts

Method 1: Link in HTML (quick)

```
<!DOCTYPE html>
<html>
<head>
  <!-- Google Fonts -->
  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com"
crossorigin>
  <link href="https://fonts.googleapis.com/css2?
family=Inter:wght@400;500;600;700&display=swap" rel="stylesheet">

  <script src="https://cdn.tailwindcss.com"></script>
  <script>
    tailwind.config = {
      theme: {
        extend: {
          fontFamily: {
            sans: ['Inter', 'sans-serif'],
          },
        },
      },
    }
```

```
  </script>
</head>
<body>
  <h1 class="font-sans text-4xl">Now using Inter font!</h1>
</body>
</html>
```

Method 2: Import in CSS (better for production)

```
/* In your main CSS file */
@import url('https://fonts.googleapis.com/css2?
family=Inter:wght@400;500;600;700&display=swap');

@tailwind base;
@tailwind components;
@tailwind utilities;
```

Then configure in `tailwind.config.js`:

```
export default {
  theme: {
    extend: {
      fontFamily: {
        sans: ['Inter', 'sans-serif'],
        display: ['Playfair Display', 'serif'],
      },
    },
  },
}
```

Use in HTML:

```
<h1 class="font-sans">Body text in Inter</h1>
```

```
<h2 class="font-display">Headings in Playfair Display</h2>
```

Using Local/Self-Hosted Fonts

1. Add fonts to your project:

```
/public
  /fonts
    Inter-Regular.woff2
    Inter-Bold.woff2
```

2. Define @font-face in CSS:

```
@font-face {
  font-family: 'Inter';
  src: url('/fonts/Inter-Regular.woff2') format('woff2');
  font-weight: 400;
  font-style: normal;
  font-display: swap;
}

@font-face {
  font-family: 'Inter';
  src: url('/fonts/Inter-Bold.woff2') format('woff2');
  font-weight: 700;
  font-style: normal;
  font-display: swap;
}

@tailwind base;
@tailwind components;
@tailwind utilities;
```

3. Configure Tailwind:

```
// tailwind.config.js
export default {
  theme: {
    extend: {
      fontFamily: {
        sans: ['Inter', 'system-ui', 'sans-serif'],
      },
    },
  },
}
```

Multiple Custom Fonts

```
export default {
  theme: {
    extend: {
      fontFamily: {
        sans: ['Inter', 'sans-serif'],          // Body text
        display: ['Playfair Display', 'serif'],  // Headings
        mono: ['Fira Code', 'monospace'],        // Code
      },
    },
  },
}
```

Usage:

```
<article>
  <h1 class="font-display text-5xl font-bold">
    Elegant Heading
  </h1>

  <p class="font-sans text-base">
```

```
    Clean, readable body text
  </p>

  <code class="font-mono bg-gray-100 px-2 py-1">
    const x = 42;
  </code>
</article>
```

Typography Plugin

The official `@tailwindcss/typography` plugin provides beautiful typographic defaults for content you don't control (like markdown or CMS content).

Installation

```
npm install -D @tailwindcss/typography
```

Configuration

```
// tailwind.config.js
export default {
  plugins: [
    require('@tailwindcss/typography'),
  ],
}
```

Usage

Simply add the `prose` class to a container:

```html
<article class="prose lg:prose-xl">
  <h1>This heading gets automatic styling</h1>
  <p>Paragraphs get proper spacing, line-height, and color.</p>
  <ul>
    <li>Lists are styled automatically</li>
    <li>No need for individual classes</li>
  </ul>
  <blockquote>
    Blockquotes look great too
  </blockquote>
  <pre><code>Code blocks are formatted nicely</code></pre>
</article>
```

What `prose` does automatically:

- Sensible font sizes and spacing
- Proper line heights
- Styled lists, quotes, code blocks
- Links, strong, em tags
- Headings hierarchy
- Images and figures

Prose Modifiers

```html
<!-- Size variants -->
<div class="prose prose-sm">Small typography</div>
<div class="prose prose-base">Base size (default)</div>
<div class="prose prose-lg">Large typography</div>
<div class="prose prose-xl">Extra large typography</div>
<div class="prose prose-2xl">2XL typography</div>

<!-- Color variants -->
<div class="prose prose-gray">Gray links and accents</div>
```

```
<div class="prose prose-slate">Slate theme</div>
<div class="prose prose-zinc">Zinc theme</div>
<div class="prose prose-stone">Stone theme</div>
<div class="prose prose-blue">Blue theme</div>
<div class="prose prose-indigo">Indigo theme</div>

<!-- Invert for dark backgrounds -->
<div class="bg-gray-900 p-8">
  <article class="prose prose-invert">
    <h1>White text on dark background</h1>
  </article>
</div>
```

Real-World Blog Example

```
<div class="bg-gray-50 min-h-screen py-12">
  <article class="prose lg:prose-xl mx-auto px-4">
    <h1>The Complete Guide to Tailwind Typography</h1>

    <p class="lead">
      Typography is the foundation of good design. Here's
everything you need to know.
    </p>

    <h2>Why Typography Matters</h2>
    <p>
      Good typography improves readability, establishes
hierarchy,
      and creates visual harmony.
    </p>

    <h3>Key Principles</h3>
    <ul>
      <li>Choose appropriate font sizes</li>
```

```
      <li>Maintain consistent spacing</li>
      <li>Consider line length</li>
    </ul>

    <blockquote>
      Typography is what language looks like.
    </blockquote>

    <p>
      And here's some <code>inline code</code> within a
paragraph.
    </p>
  </article>
</div>
```

The magic: No classes needed on internal elements! Everything is styled perfectly.

Customizing Prose

You can customize the prose plugin in your config:

```
export default {
  theme: {
    extend: {
      typography: {
        DEFAULT: {
          css: {
            color: '#333',
            h1: {
              color: '#111',
              fontWeight: '800',
            },
            a: {
              color: '#3b82f6',
```

```
                '&:hover': {
                  color: '#2563eb',
                },
              },
            },
          },
        },
      },
    },
  },
  plugins: [
    require('@tailwindcss/typography'),
  ],
}
```

Key Takeaways

✅ Use **type scale** (text-sm, text-base, text-xl, etc.) for consistent sizing
✅ **Font weight** creates hierarchy: normal (body), semibold (buttons), bold (headings)
✅ **Line height**: tight for headlines, relaxed for body text
✅ **Letter spacing**: wider for uppercase, normal for regular text
✅ Use `truncate` with width constraints to handle long text
✅ **Custom fonts** via Google Fonts or self-hosted files
✅ **Typography plugin** (`prose`) for styled content blocks
✅ Combine size + weight + spacing for effective hierarchy

Practice Assignments

Assignment 1 (Easy): Type Scale Exploration

Create a page showing the entire type scale (text-xs through text-9xl). For each size, show:

- The size name
- The actual rendered size
- Use appropriate font weights for different sizes

Assignment 2 (Easy): Heading Hierarchy

Create a blog post structure with:

- h1: Main title (text-4xl, font-bold, leading-tight)
- h2: Section heading (text-2xl, font-semibold)
- h3: Subsection (text-xl, font-semibold)
- Body text (text-base, leading-relaxed)
- Caption (text-sm, text-gray-500)

Ensure proper spacing between elements.

Assignment 3 (Medium): Custom Font Integration

1. Choose a Google Font (like Inter, Poppins, or Roboto)
2. Integrate it into your Tailwind project
3. Configure it in `tailwind.config.js`
4. Build a landing page using your custom font
5. Use different weights (400, 600, 700)

Assignment 4 (Medium): Card with Text Overflow

Create a product card that:

- Has a fixed width (w-64)
- Product title that truncates if too long
- Price that never wraps (whitespace-nowrap)

- Description with line clamping
- Proper typography hierarchy

Assignment 5 (Hard): Typography Plugin Blog

1. Install `@tailwindcss/typography`
2. Create a full blog post layout with:
    - Centered prose container
    - Responsive sizing (prose-lg on desktop)
    - All standard markdown elements (h1-h6, p, ul, ol, blockquote, code)
    - Dark mode variant using prose-invert
3. Add a toggle to switch between light and dark modes

Bonus: Customize the prose styles in your config to match a specific brand.

Ready for **Module 5: Layout with Flexbox & Grid**? Let me know!

1/2/2026, 4:31:15 PM

MODULE 5: LAYOUT WITH FLEXBOX & GRID

Flex Utilities (Justify, Align, Direction, Wrap)

Since you already understand Flexbox from vanilla CSS, let's see how Tailwind maps those concepts to utility classes.

Display: Flex

```html
<div class="flex">
  <!-- display: flex -->
</div>

<div class="inline-flex">
  <!-- display: inline-flex -->
</div>
```

Flex Direction

```html
<!-- Row (default) -->
<div class="flex flex-row">
  <div>1</div>
  <div>2</div>
  <div>3</div>
</div>

<!-- Row reversed -->
<div class="flex flex-row-reverse">
  <div>1</div>  <!-- Appears on right -->
  <div>2</div>
  <div>3</div>  <!-- Appears on left -->
</div>

<!-- Column -->
<div class="flex flex-col">
  <div>1</div>
  <div>2</div>
  <div>3</div>
</div>

<!-- Column reversed -->
```

```
<div class="flex flex-col-reverse">
  <div>1</div>  <!-- Appears on bottom -->
  <div>2</div>
  <div>3</div>  <!-- Appears on top -->
</div>
```

Comparison to vanilla CSS:

```
/* Vanilla CSS */
.container {
  display: flex;
  flex-direction: row;
}
```

```
<!-- Tailwind -->
<div class="flex flex-row">
```

Justify Content (Main Axis)

Controls alignment along the **main axis** (horizontal in flex-row, vertical in flex-col):

```
<!-- Start (default) -->
<div class="flex justify-start">
  <div>1</div>
  <div>2</div>
  <div>3</div>
</div>

<!-- Center -->
<div class="flex justify-center">
```

```
  <div>1</div>
  <div>2</div>
  <div>3</div>
</div>

<!-- End -->
<div class="flex justify-end">
  <div>1</div>
  <div>2</div>
  <div>3</div>
</div>

<!-- Space between -->
<div class="flex justify-between">
  <div>1</div>  <!-- Far left -->
  <div>2</div>  <!-- Center -->
  <div>3</div>  <!-- Far right -->
</div>

<!-- Space around -->
<div class="flex justify-around">
  <div>1</div>  <!-- Equal space around each -->
  <div>2</div>
  <div>3</div>
</div>

<!-- Space evenly -->
<div class="flex justify-evenly">
  <div>1</div>  <!-- Equal space between all (including edges) --
>
  <div>2</div>
  <div>3</div>
</div>
```

Visual representation:

```
justify-start:
[1][2][3]

justify-center:
    [1][2][3]

justify-end:
         [1][2][3]

justify-between:
[1]     [2]     [3]

justify-around:
  [1]   [2]   [3]

justify-evenly:
 [1]   [2]   [3]
```

Real-world example - Navigation bar:

```
<nav class="flex justify-between items-center px-6 py-4 bg-white
shadow">
  <!-- Logo on left -->
  <div class="text-xl font-bold">
    Logo
  </div>

  <!-- Links in center (using nested flex) -->
  <div class="flex space-x-6">
    <a href="#" class="hover:text-blue-600">Home</a>
    <a href="#" class="hover:text-blue-600">About</a>
    <a href="#" class="hover:text-blue-600">Contact</a>
  </div>
```

```
  <!-- Button on right -->
  <button class="bg-blue-600 text-white px-4 py-2 rounded">
    Sign In
  </button>
</nav>
```

Align Items (Cross Axis)

Controls alignment along the **cross axis** (vertical in flex-row, horizontal in flex-col):

```
<!-- Stretch (default) -->
<div class="flex items-stretch h-32">
  <div>1</div>  <!-- Stretches to fill height -->
  <div>2</div>
  <div>3</div>
</div>

<!-- Start -->
<div class="flex items-start h-32">
  <div>1</div>  <!-- Aligned to top -->
  <div>2</div>
  <div>3</div>
</div>

<!-- Center -->
<div class="flex items-center h-32">
  <div>1</div>  <!-- Vertically centered -->
  <div>2</div>
  <div>3</div>
</div>
```

```
<!-- End -->
<div class="flex items-end h-32">
  <div>1</div>  <!-- Aligned to bottom -->
  <div>2</div>
  <div>3</div>
</div>

<!-- Baseline -->
<div class="flex items-baseline">
  <div class="text-4xl">A</div>  <!-- Text baselines align -->
  <div class="text-sm">B</div>
  <div class="text-2xl">C</div>
</div>
```

Most common pattern - vertical centering:

```
<!-- Center content vertically and horizontally -->
<div class="flex items-center justify-center h-screen">
  <div class="text-center">
    <h1 class="text-4xl font-bold">Perfectly Centered</h1>
    <p class="text-gray-600">Both vertically and horizontally</p>
  </div>
</div>
```

This is **MUCH EASIER** than vanilla CSS's old-school centering techniques!

Align Content (Multiple Lines)

Only applies when flex items wrap to multiple lines:

```
<!-- Normal (default) -->
<div class="flex flex-wrap content-normal h-64">
```

```
  <!-- Multiple rows -->
</div>

<!-- Start -->
<div class="flex flex-wrap content-start h-64">
  <!-- Rows packed to start -->
</div>

<!-- Center -->
<div class="flex flex-wrap content-center h-64">
  <!-- Rows centered -->
</div>

<!-- End -->
<div class="flex flex-wrap content-end h-64">
  <!-- Rows packed to end -->
</div>

<!-- Between -->
<div class="flex flex-wrap content-between h-64">
  <!-- Rows distributed with space between -->
</div>

<!-- Around -->
<div class="flex flex-wrap content-around h-64">
  <!-- Rows distributed with space around -->
</div>

<!-- Evenly -->
<div class="flex flex-wrap content-evenly h-64">
  <!-- Rows distributed evenly -->
</div>
```

Align Self (Individual Item)

Override `items-*` for a specific child:

```
<div class="flex items-start h-32">
  <div class="bg-blue-500 p-4">Normal (start)</div>
  <div class="bg-green-500 p-4 self-center">Centered</div>
  <div class="bg-red-500 p-4 self-end">End</div>
  <div class="bg-yellow-500 p-4 self-stretch">Stretched</div>
</div>
```

Flex Wrap

```
<!-- No wrap (default) -->
<div class="flex flex-nowrap">
  <!-- Items will shrink or overflow -->
</div>

<!-- Wrap -->
<div class="flex flex-wrap">
  <!-- Items wrap to new line if needed -->
</div>

<!-- Wrap reverse -->
<div class="flex flex-wrap-reverse">
  <!-- Items wrap in reverse order -->
</div>
```

Real-world example - responsive tag list:

```
<div class="flex flex-wrap gap-2">
  <span class="px-3 py-1 bg-blue-100 text-blue-800 rounded-full
```

```
text-sm">
    React
  </span>
  <span class="px-3 py-1 bg-green-100 text-green-800 rounded-full
text-sm">
    Node.js
  </span>
  <span class="px-3 py-1 bg-purple-100 text-purple-800 rounded-
full text-sm">
    TypeScript
  </span>
  <span class="px-3 py-1 bg-yellow-100 text-yellow-800 rounded-
full text-sm">
    JavaScript
  </span>
  <span class="px-3 py-1 bg-red-100 text-red-800 rounded-full
text-sm">
    HTML/CSS
  </span>
</div>
```

Tags will wrap to multiple lines on narrow screens!

Gap Utilities

Gap creates spacing between flex/grid children (MUCH cleaner than margins):

```
<!-- Horizontal gap in flex row -->
<div class="flex gap-4">
  <div>1</div>  <!-- 1rem gap between items -->
  <div>2</div>
  <div>3</div>
```

```
</div>

<!-- Vertical gap in flex column -->
<div class="flex flex-col gap-6">
  <div>1</div>  <!-- 1.5rem gap between items -->
  <div>2</div>
  <div>3</div>
</div>

<!-- Different horizontal and vertical gaps -->
<div class="flex flex-wrap gap-x-4 gap-y-8">
  <div>1</div>  <!-- 1rem horizontal, 2rem vertical -->
  <div>2</div>
  <div>3</div>
</div>
```

Comparison to the old way:

```
<!-- ❌ Old way: margin on each child -->
<div class="flex">
  <div class="mr-4">1</div>
  <div class="mr-4">2</div>
  <div>3</div>  <!-- Last one has no margin -->
</div>

<!-- ✅ New way: gap on parent -->
<div class="flex gap-4">
  <div>1</div>
  <div>2</div>
  <div>3</div>
</div>
```

Gap scale:

```
gap-0   = 0
gap-1   = 0.25rem (4px)
gap-2   = 0.5rem (8px)
gap-4   = 1rem (16px)
gap-6   = 1.5rem (24px)
gap-8   = 2rem (32px)
gap-12  = 3rem (48px)
gap-16  = 4rem (64px)
```

Flex Grow/Shrink

Flex Grow

Controls how items expand to fill available space:

```
<!-- Don't grow (default) -->
<div class="flex">
  <div class="flex-none bg-blue-500 p-4">Fixed</div>
  <div class="flex-none bg-green-500 p-4">Fixed</div>
</div>

<!-- Grow to fill space -->
<div class="flex">
  <div class="flex-none bg-blue-500 p-4">Fixed</div>
  <div class="flex-1 bg-green-500 p-4">Grows to fill</div>
  <div class="flex-none bg-red-500 p-4">Fixed</div>
</div>

<!-- Multiple growing items (equal distribution) -->
<div class="flex">
  <div class="flex-1 bg-blue-500 p-4">Equal</div>
  <div class="flex-1 bg-green-500 p-4">Equal</div>
```

```
  <div class="flex-1 bg-red-500 p-4">Equal</div>
</div>
```

Visual representation:

```
flex-none, flex-none:
[Fixed][Fixed]

flex-none, flex-1, flex-none:
[Fixed][========Grows========][Fixed]

flex-1, flex-1, flex-1:
[===Equal===][===Equal===][===Equal===]
```

Flex Shrink

Controls whether items can shrink:

```
<!-- Shrink (default) -->
<div class="flex">
  <div class="flex-shrink bg-blue-500 p-4">Can shrink</div>
  <div class="flex-shrink bg-green-500 p-4">Can shrink</div>
</div>

<!-- Don't shrink -->
<div class="flex">
  <div class="flex-shrink-0 bg-blue-500 p-4 w-64">Won't shrink
below 64</div>
  <div class="flex-1 bg-green-500 p-4">Shrinks first</div>
</div>
```

Flex Basis

Set initial size before growing/shrinking:

```
<div class="flex">
  <div class="basis-1/4 bg-blue-500 p-4">25% basis</div>
  <div class="basis-1/2 bg-green-500 p-4">50% basis</div>
  <div class="basis-1/4 bg-red-500 p-4">25% basis</div>
</div>
```

Common Flex Patterns

1. Sidebar layout:

```
<div class="flex h-screen">
  <!-- Fixed sidebar -->
  <aside class="flex-none w-64 bg-gray-800 text-white p-6">
    <h2 class="text-xl font-bold mb-4">Sidebar</h2>
    <nav class="space-y-2">
      <a href="#" class="block hover:bg-gray-700 px-3 py-2
rounded">Home</a>
      <a href="#" class="block hover:bg-gray-700 px-3 py-2
rounded">About</a>
    </nav>
  </aside>

  <!-- Growing main content -->
  <main class="flex-1 bg-gray-100 p-6 overflow-auto">
    <h1 class="text-3xl font-bold mb-4">Main Content</h1>
    <p>This area grows to fill available space</p>
  </main>
</div>
```

2. Sticky footer:

```
<div class="flex flex-col min-h-screen">
  <!-- Header -->
  <header class="flex-none bg-white shadow p-4">
```

```
    Header
  </header>

  <!-- Main content grows -->
  <main class="flex-1 bg-gray-50 p-8">
    Main content (pushes footer down)
  </main>

  <!-- Footer -->
  <footer class="flex-none bg-gray-800 text-white p-4 text-
center">
    Footer always at bottom
  </footer>
</div>
```

3. Card with header, body, and footer:

```
<div class="flex flex-col h-96 bg-white shadow rounded-lg">
  <!-- Fixed header -->
  <div class="flex-none px-6 py-4 border-b">
    <h3 class="text-xl font-bold">Card Title</h3>
  </div>

  <!-- Scrollable content -->
  <div class="flex-1 overflow-auto px-6 py-4">
    <p>Content that can scroll...</p>
    <p>More content...</p>
    <p>Even more content...</p>
  </div>

  <!-- Fixed footer -->
  <div class="flex-none px-6 py-4 border-t bg-gray-50">
    <button class="w-full bg-blue-600 text-white py-2 rounded">
      Action
    </button>
```

```
  </div>
</div>
```

Grid Utilities (Columns, Rows, Spans)

Tailwind's Grid utilities map to CSS Grid properties.

Basic Grid

```
<!-- Define columns -->
<div class="grid grid-cols-3 gap-4">
  <div>1</div>
  <div>2</div>
  <div>3</div>
  <div>4</div>
  <div>5</div>
  <div>6</div>
</div>
```

Grid column count:

```
<div class="grid grid-cols-1">1 column</div>
<div class="grid grid-cols-2">2 columns</div>
<div class="grid grid-cols-3">3 columns</div>
<div class="grid grid-cols-4">4 columns</div>
<div class="grid grid-cols-5">5 columns</div>
<div class="grid grid-cols-6">6 columns</div>
<div class="grid grid-cols-12">12 columns (like Bootstrap)</div>
```

Column Spanning

Control how many columns an item spans:

```
<div class="grid grid-cols-6 gap-4">
  <div class="col-span-1 bg-blue-500">1</div>
  <div class="col-span-2 bg-green-500">2 cols</div>
  <div class="col-span-3 bg-red-500">3 cols</div>
  <div class="col-span-4 bg-yellow-500">4 cols</div>
  <div class="col-span-2 bg-purple-500">2</div>
</div>
```

Visual representation:

```
6-column grid:
[1][==2==][====3====]
[======4======][==2==]
```

Spanning all columns:

```
<div class="grid grid-cols-4 gap-4">
  <div class="bg-blue-500">1</div>
  <div class="bg-blue-500">2</div>
  <div class="bg-blue-500">3</div>
  <div class="bg-blue-500">4</div>

  <!-- Full-width element -->
  <div class="col-span-4 bg-red-500">
    Spans entire width
  </div>
</div>
```

Column Start/End

Precise placement:

```html
<div class="grid grid-cols-6 gap-4">
  <!-- Start at column 2, span 3 columns -->
  <div class="col-start-2 col-span-3 bg-blue-500">
    Offset and span
  </div>

  <!-- Start at column 1, end at column 4 -->
  <div class="col-start-1 col-end-4 bg-green-500">
    Explicit start and end
  </div>
</div>
```

Grid Rows

```html
<!-- Define rows -->
<div class="grid grid-rows-3 gap-4 h-64">
  <div class="bg-blue-500">Row 1</div>
  <div class="bg-green-500">Row 2</div>
  <div class="bg-red-500">Row 3</div>
</div>

<!-- Row spanning -->
<div class="grid grid-cols-3 grid-rows-3 gap-4 h-96">
  <div class="row-span-2 bg-blue-500">Spans 2 rows</div>
  <div class="bg-green-500">1</div>
  <div class="bg-red-500">2</div>
  <div class="bg-yellow-500">3</div>
  <div class="bg-purple-500">4</div>
</div>
```

Auto-Fit and Auto-Fill

```
<!-- Auto-fit: columns adjust to fit content -->
<div class="grid grid-cols-[repeat(auto-fit,minmax(200px,1fr))]
gap-4">
  <div class="bg-blue-500 p-4">Card 1</div>
  <div class="bg-green-500 p-4">Card 2</div>
  <div class="bg-red-500 p-4">Card 3</div>
</div>
```

This is responsive without breakpoints! Cards automatically wrap based on available width.

Grid Template Columns (Custom)

For more control, use arbitrary values:

```
<!-- Sidebar + main + sidebar layout -->
<div class="grid grid-cols-[200px_1fr_200px] gap-4">
  <aside class="bg-gray-200">Left Sidebar</aside>
  <main class="bg-white">Main Content</main>
  <aside class="bg-gray-200">Right Sidebar</aside>
</div>

<!-- Fractional units -->
<div class="grid grid-cols-[1fr_2fr_1fr] gap-4">
  <div class="bg-blue-500">1fr</div>
  <div class="bg-green-500">2fr (twice as wide)</div>
  <div class="bg-red-500">1fr</div>
</div>
```

Grid Alignment

Similar to Flexbox alignment:

Justify Items (horizontal alignment of grid items)

```
<div class="grid grid-cols-3 gap-4 justify-items-start">
  <!-- Items aligned to start of their cells -->
</div>

<div class="grid grid-cols-3 gap-4 justify-items-center">
  <!-- Items centered in their cells -->
</div>

<div class="grid grid-cols-3 gap-4 justify-items-end">
  <!-- Items aligned to end of their cells -->
</div>

<div class="grid grid-cols-3 gap-4 justify-items-stretch">
  <!-- Items stretch to fill cells (default) -->
</div>
```

Align Items (vertical alignment of grid items)

```
<div class="grid grid-cols-3 gap-4 h-64 items-start">
  <!-- Items aligned to top of their cells -->
</div>

<div class="grid grid-cols-3 gap-4 h-64 items-center">
  <!-- Items centered vertically in their cells -->
</div>

<div class="grid grid-cols-3 gap-4 h-64 items-end">
  <!-- Items aligned to bottom of their cells -->
</div>
```

Place Items (shorthand)

```
<!-- Center both horizontally and vertically -->
<div class="grid grid-cols-3 gap-4 h-64 place-items-center">
  <div>Centered</div>
  <div>Centered</div>
  <div>Centered</div>
</div>
```

Individual Item Alignment

```
<div class="grid grid-cols-3 gap-4 h-64">
  <div class="justify-self-start">Start</div>
  <div class="justify-self-center">Center</div>
  <div class="justify-self-end">End</div>

  <div class="self-start">Top</div>
  <div class="self-center">Middle</div>
  <div class="self-end">Bottom</div>
</div>
```

Common Layout Patterns

1. Responsive Card Grid

```
<div class="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3
xl:grid-cols-4 gap-6">
  <!-- Card -->
  <div class="bg-white shadow rounded-lg p-6">
    <h3 class="text-xl font-bold mb-2">Card Title</h3>
    <p class="text-gray-600">Card content here</p>
```

```
  </div>

  <!-- More cards... -->
</div>
```

Breakdown:

- Mobile: 1 column
- Tablet (md): 2 columns
- Desktop (lg): 3 columns
- Large desktop (xl): 4 columns

2. Dashboard Layout

```
<div class="grid grid-cols-1 lg:grid-cols-4 gap-6">
  <!-- Sidebar (full width on mobile, 1 col on desktop) -->
  <aside class="lg:col-span-1 bg-gray-800 text-white p-6">
    Navigation
  </aside>

  <!-- Main content (3 cols on desktop) -->
  <main class="lg:col-span-3 space-y-6">
    <!-- Stats grid -->
    <div class="grid grid-cols-1 md:grid-cols-3 gap-4">
      <div class="bg-white p-6 shadow rounded">Stat 1</div>
      <div class="bg-white p-6 shadow rounded">Stat 2</div>
      <div class="bg-white p-6 shadow rounded">Stat 3</div>
    </div>

    <!-- Chart area -->
    <div class="bg-white p-6 shadow rounded">
      Chart goes here
    </div>
```

```
  </main>
</div>
```

3. Holy Grail Layout (Header, 2 Sidebars, Footer)

```
<div class="grid grid-rows-[auto_1fr_auto] min-h-screen">
  <!-- Header -->
  <header class="bg-white shadow p-4">
    Header
  </header>

  <!-- Middle section with sidebars -->
  <div class="grid grid-cols-[200px_1fr_200px]">
    <aside class="bg-gray-100 p-4">Left Sidebar</aside>
    <main class="bg-white p-8">Main Content</main>
    <aside class="bg-gray-100 p-4">Right Sidebar</aside>
  </div>

  <!-- Footer -->
  <footer class="bg-gray-800 text-white p-4">
    Footer
  </footer>
</div>
```

4. Masonry-Style Grid (Pinterest-like)

```
<!-- Using columns (CSS columns, not grid) -->
<div class="columns-1 md:columns-2 lg:columns-3 gap-6">
  <div class="break-inside-avoid mb-6 bg-white shadow rounded p-
4">
    <img src="image1.jpg" class="w-full rounded mb-2">
    <h3 class="font-bold">Title 1</h3>
  </div>
```

```
  <div class="break-inside-avoid mb-6 bg-white shadow rounded p-
4">
    <img src="image2.jpg" class="w-full rounded mb-2">
    <h3 class="font-bold">Title 2</h3>
  </div>

  <!-- More cards with varying heights... -->
</div>
```

5. Feature Section (Alternating Layout)

```
<div class="space-y-16">
  <!-- Feature 1: Image left, text right -->
  <div class="grid grid-cols-1 lg:grid-cols-2 gap-8 items-
center">
    <div>
      <img src="feature1.jpg" class="rounded-lg shadow-lg">
    </div>
    <div>
      <h2 class="text-3xl font-bold mb-4">Feature One</h2>
      <p class="text-gray-600">Description...</p>
    </div>
  </div>

  <!-- Feature 2: Text left, image right -->
  <div class="grid grid-cols-1 lg:grid-cols-2 gap-8 items-
center">
    <div class="lg:order-2">
      <img src="feature2.jpg" class="rounded-lg shadow-lg">
    </div>
    <div class="lg:order-1">
      <h2 class="text-3xl font-bold mb-4">Feature Two</h2>
      <p class="text-gray-600">Description...</p>
    </div>
```

```
  </div>
</div>
```

Comparing Vanilla CSS Flexbox/Grid to Tailwind

Vanilla CSS Flexbox:

```
.container {
  display: flex;
  justify-content: space-between;
  align-items: center;
  gap: 1rem;
}

.item {
  flex: 1;
}
```

```
<div class="container">
  <div class="item">1</div>
  <div class="item">2</div>
  <div class="item">3</div>
</div>
```

Tailwind Flexbox:

```
<div class="flex justify-between items-center gap-4">
  <div class="flex-1">1</div>
  <div class="flex-1">2</div>
```

```
  <div class="flex-1">3</div>
</div>
```

Advantage: Everything is visible in HTML. No need to check CSS file.

Vanilla CSS Grid:

```
.grid {
  display: grid;
  grid-template-columns: repeat(3, 1fr);
  gap: 1.5rem;
}

.grid-item:first-child {
  grid-column: span 2;
}
```

```
<div class="grid">
  <div class="grid-item">1</div>
  <div>2</div>
  <div>3</div>
</div>
```

Tailwind Grid:

```
<div class="grid grid-cols-3 gap-6">
  <div class="col-span-2">1</div>
  <div>2</div>
  <div>3</div>
</div>
```

Key insight: Tailwind's utilities are slightly more verbose but **completely self-documenting**.

Common Mistakes to Avoid

1. Forgetting to set display

```
<!-- ❌ Won't work - no display set -->
<div class="justify-center items-center">
  Content
</div>

<!-- ✅ Correct -->
<div class="flex justify-center items-center">
  Content
</div>
```

2. Confusing justify and align

```
<!-- ❌ Wrong - these are swapped -->
<div class="flex justify-center">
  <!-- This centers horizontally, not vertically -->
</div>

<!-- ✅ For vertical centering in flex-row -->
<div class="flex items-center">
  <!-- This centers vertically -->
</div>

<!-- ✅ For both -->
<div class="flex items-center justify-center">
  <!-- Centered both ways -->
</div>
```

3. Using margin instead of gap

```
<!-- ❌ Repetitive -->
<div class="flex">
  <div class="mr-4">1</div>
  <div class="mr-4">2</div>
  <div>3</div>
</div>

<!-- ✅ Cleaner -->
<div class="flex gap-4">
  <div>1</div>
  <div>2</div>
  <div>3</div>
</div>
```

4. Not considering responsive layouts

```
<!-- ❌ Fixed 3 columns (breaks on mobile) -->
<div class="grid grid-cols-3 gap-4">
  <!-- Too cramped on mobile -->
</div>

<!-- ✅ Responsive -->
<div class="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-
4">
  <!-- 1 col mobile, 2 tablet, 3 desktop -->
</div>
```

Key Takeaways

✅ **Flexbox**: Use for **one-dimensional** layouts (rows or columns)
✅ **Grid**: Use for **two-dimensional** layouts (rows AND columns)

✅ **Gap** replaces margins for spacing between flex/grid items
✅ `flex-1` makes items grow to fill space equally
✅ `items-center justify-center` = perfect centering
✅ Grid **column spanning** creates complex layouts easily
✅ **Responsive grids**: Change column count at breakpoints
✅ Combine Flex and Grid for powerful nested layouts

Practice Assignments

Assignment 1 (Easy): Flex Navigation Bar

Create a navigation bar with:

- Logo on the left
- Navigation links in the center (using flex)
- Sign-in button on the right
- Vertically centered items
- Use `justify-between` and `items-center`

Assignment 2 (Easy): Card Grid

Create a responsive grid of cards:

- 1 column on mobile
- 2 columns on tablet (md)
- 3 columns on desktop (lg)
- Each card has image, title, description, and button
- Use consistent gap spacing

Assignment 3 (Medium): Dashboard Layout

Build a dashboard with:

- Fixed sidebar (w-64) on the left
- Main content area that grows (flex-1)
- Grid of stat cards (3 columns) in the main area
- Everything should work on mobile (sidebar becomes top bar or hidden)

Assignment 4 (Medium): Complex Grid Layout

Create a blog layout using grid:

- Header spans full width
- Featured post spans 2 columns
- 4 smaller posts in remaining space
- Sidebar spans full height on the right
- 12-column grid system

Hint: Use `col-span-*` and `row-span-*`

Assignment 5 (Hard): Responsive Landing Page

Build a complete landing page section with:

1. Hero section with centered content (flex)
2. Feature section with alternating image/text layout (grid)
3. Stats section (4 items in grid, responsive)
4. Testimonial cards (3 columns, responsive)
5. Footer with 4 columns on desktop, stacked on mobile

Requirements:

- Mobile-first responsive design
- Proper use of flex and grid
- Gap instead of margins
- Semantic HTML structure

Ready for **Module 6: Responsive Design**? Let me know!

1/2/2026, 7:02:45 PM

MODULE 6: RESPONSIVE DESIGN

Breakpoint System (sm, md, lg, xl, 2xl)

Tailwind uses **five default breakpoints** that correspond to common device sizes:

```
sm   = 640px  and up (landscape phones, small tablets)
md   = 768px  and up (tablets)
lg   = 1024px and up (laptops, desktops)
xl   = 1280px and up (large desktops)
2xl  = 1536px and up (larger desktops)
```

Visual representation:

```
Mobile        sm          md          lg          xl         2xl
(0-639px)   (640px+)    (768px+)   (1024px+)  (1280px+)
(1536px+)
   📱           📱            📱💻         💻          🖥️         🖥️
🖥️
```

Comparison to vanilla CSS:

```
/* Vanilla CSS */
.element {
  font-size: 1rem;
}
```

```
@media (min-width: 768px) {
  .element {
    font-size: 1.5rem;
  }
}

@media (min-width: 1024px) {
  .element {
    font-size: 2rem;
  }
}
```

```
<!-- Tailwind (much more concise!) -->
<div class="text-base md:text-2xl lg:text-4xl">
  Responsive text
</div>
```

Mobile-First Approach in Tailwind

Tailwind is **mobile-first**, meaning unprefixed utilities apply to all screen sizes, and you use breakpoint prefixes to override at larger sizes.

The pattern:

```
base class → applies to ALL sizes (mobile first)
sm: → applies from 640px and up
md: → applies from 768px and up
lg: → applies from 1024px and up
xl: → applies from 1280px and up
2xl: → applies from 1536px and up
```

Example: Understanding the Cascade

```
<div class="text-sm md:text-base lg:text-xl">
  Responsive text
</div>
```

What happens:

- **Mobile (0-767px)**: `text-sm` (14px)
- **Tablet (768-1023px)**: `text-base` (16px) - `md:` overrides
- **Desktop (1024px+)**: `text-xl` (20px) - `lg:` overrides

Important: Each breakpoint overrides the previous one, but stays until the next breakpoint changes it.

```
<div class="bg-blue-500 md:bg-green-500">
  Blue on mobile, green from tablet up
</div>
```

What happens:

- **Mobile**: Blue background
- **Tablet and above**: Green background (no further changes)

Responsive Utility Syntax (md:, lg:, etc.)

Basic Syntax

Pattern: `[breakpoint]:[utility]`

```
<!-- Hidden on mobile, visible on tablet+ -->
<div class="hidden md:block">
```

```
  Visible from tablet up
</div>

<!-- Full width on mobile, half width on desktop -->
<div class="w-full lg:w-1/2">
  Responsive width
</div>

<!-- Stacked on mobile, flex row on desktop -->
<div class="flex flex-col lg:flex-row">
  <div>Item 1</div>
  <div>Item 2</div>
</div>
```

Combining Multiple Breakpoints

```
<!-- Different styles at each breakpoint -->
<div class="text-sm sm:text-base md:text-lg lg:text-xl xl:text-
2xl">
  Progressively larger text
</div>

<!-- Grid columns change at each breakpoint -->
<div class="grid grid-cols-1 sm:grid-cols-2 md:grid-cols-3
lg:grid-cols-4 gap-4">
  <div>Card 1</div>
  <div>Card 2</div>
  <div>Card 3</div>
  <div>Card 4</div>
</div>
```

Breakdown:

- **Mobile**: 1 column

- **Small (640px+)**: 2 columns
- **Medium (768px+)**: 3 columns
- **Large (1024px+)**: 4 columns

Real-World Navigation Example

```
<nav class="bg-white shadow">
  <div class="max-w-7xl mx-auto px-4">
    <div class="flex justify-between items-center h-16">

      <!-- Logo -->
      <div class="flex-shrink-0">
        <h1 class="text-xl font-bold">Logo</h1>
      </div>

      <!-- Desktop Navigation (hidden on mobile) -->
      <div class="hidden md:flex space-x-8">
        <a href="#" class="text-gray-700 hover:text-gray-
900">Home</a>
        <a href="#" class="text-gray-700 hover:text-gray-
900">About</a>
        <a href="#" class="text-gray-700 hover:text-gray-
900">Services</a>
        <a href="#" class="text-gray-700 hover:text-gray-
900">Contact</a>
      </div>

      <!-- Mobile Menu Button (visible only on mobile) -->
      <button class="md:hidden p-2">
        <svg class="h-6 w-6" fill="none" stroke="currentColor"
viewBox="0 0 24 24">
          <path stroke-linecap="round" stroke-linejoin="round"
stroke-width="2" d="M4 6h16M4 12h16M4 18h16"></path>
        </svg>
```

```
      </button>

    </div>
  </div>
</nav>
```

Key techniques:

- `hidden md:flex` - Hidden on mobile, flex on tablet+
- `md:hidden` - Visible on mobile, hidden on tablet+

Container Utility

The `container` class creates a responsive container that centers content with max-widths at each breakpoint:

```
<div class="container mx-auto">
  Content is centered and constrained
</div>
```

Container breakpoints:

```
None       100% (full width)
sm (640px) 640px
md (768px) 768px
lg (1024px) 1024px
xl (1280px) 1280px
2xl (1536px) 1536px
```

Container with Padding

```
<div class="container mx-auto px-4">
  <!-- px-4 adds horizontal padding on mobile -->
  Content with breathing room on edges
</div>

<!-- Responsive padding -->
<div class="container mx-auto px-4 sm:px-6 lg:px-8">
  <!-- More padding on larger screens -->
</div>
```

Customizing Container

In `tailwind.config.js`:

```
export default {
  theme: {
    container: {
      center: true, // Centers by default (no need for mx-auto)
      padding: {
        DEFAULT: '1rem',
        sm: '2rem',
        lg: '4rem',
        xl: '5rem',
        '2xl': '6rem',
      },
    },
  },
}
```

Now just use:

```
<div class="container">
  Automatically centered with responsive padding
```

```
</div>
```

Common Container Patterns

1. Full-width colored section with constrained content:

```
<section class="bg-blue-600 py-16">
  <div class="container mx-auto px-4">
    <h2 class="text-3xl font-bold text-white mb-4">Section
Title</h2>
    <p class="text-blue-100">Content is constrained</p>
  </div>
</section>
```

2. Multiple sections:

```
<div class="space-y-16">
  <section class="container mx-auto px-4">
    <h2>Section 1</h2>
  </section>

  <section class="bg-gray-100 py-16">
    <div class="container mx-auto px-4">
      <h2>Section 2</h2>
    </div>
  </section>
</div>
```

Responsive Typography

Font Size

```
<!-- Small on mobile, progressively larger -->
<h1 class="text-2xl md:text-4xl lg:text-5xl xl:text-6xl font-
bold">
  Responsive Heading
</h1>

<!-- Body text adjustment -->
<p class="text-sm md:text-base lg:text-lg">
  Comfortable reading size on each device
</p>
```

Line Height

```
<!-- Tighter on mobile for space, relaxed on desktop -->
<p class="leading-snug md:leading-normal lg:leading-relaxed">
  Line height adjusts for screen size and readability
</p>
```

Letter Spacing

```
<h2 class="tracking-tight md:tracking-normal lg:tracking-wide
uppercase">
  Adjust tracking by screen size
</h2>
```

Complete Responsive Typography Example

```
<article class="max-w-4xl mx-auto px-4 py-8">
  <!-- Headline -->
  <h1 class="text-3xl sm:text-4xl md:text-5xl lg:text-6xl font-
bold leading-tight mb-4">
    The Art of Responsive Typography
```

```
  </h1>

  <!-- Subtitle -->
  <p class="text-lg sm:text-xl md:text-2xl text-gray-600 leading-relaxed mb-8">
    Creating readable text at every screen size
  </p>

  <!-- Metadata -->
  <div class="text-xs sm:text-sm text-gray-500 mb-8">
    Published on January 3, 2026
  </div>

  <!-- Body text -->
  <div class="prose prose-sm sm:prose-base md:prose-lg lg:prose-xl">
    <p>
      Typography on the web must adapt to countless screen sizes. The text you're reading now adjusts its size based on your viewport width.
    </p>
  </div>
</article>
```

Show/Hide at Breakpoints

Hiding Elements

```
<!-- Show only on mobile -->
<div class="block md:hidden">
  Mobile only content
</div>
```

```
<!-- Hide on mobile, show on tablet+ -->
<div class="hidden md:block">
  Desktop content
</div>

<!-- Show only on desktop -->
<div class="hidden lg:block">
  Large screen only
</div>

<!-- Complex visibility -->
<div class="block md:hidden lg:block">
  Visible on mobile and large desktop, hidden on tablet
</div>
```

Common Visibility Patterns

1. Mobile vs Desktop Navigation:

```
<nav>
  <!-- Mobile hamburger menu -->
  <button class="md:hidden">
    <svg><!-- Menu icon --></svg>
  </button>

  <!-- Desktop navigation links -->
  <div class="hidden md:flex space-x-6">
    <a href="#">Home</a>
    <a href="#">About</a>
    <a href="#">Contact</a>
  </div>
</nav>
```

2. Different content for different screens:

```
<div>
  <!-- Mobile image (vertical) -->
  <img src="vertical.jpg" class="md:hidden w-full">

  <!-- Desktop image (horizontal) -->
  <img src="horizontal.jpg" class="hidden md:block w-full">
</div>
```

3. Sidebar that becomes top bar:

```
<div class="flex flex-col lg:flex-row">
  <!-- Sidebar: full width on mobile, fixed width on desktop -->
  <aside class="w-full lg:w-64 bg-gray-800 p-4">
    Sidebar
  </aside>

  <!-- Main content -->
  <main class="flex-1 p-4">
    Content
  </main>
</div>
```

Real Responsive Examples

Example 1: Responsive Card Grid

```
<div class="container mx-auto px-4 py-8">
  <h2 class="text-2xl md:text-3xl lg:text-4xl font-bold mb-8">
    Our Products
```

```
  </h2>

  <!-- Grid: 1 col mobile, 2 cols tablet, 3 cols desktop, 4 cols
large -->
  <div class="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3
xl:grid-cols-4 gap-4 md:gap-6">

    <!-- Product Card -->
    <div class="bg-white shadow rounded-lg overflow-hidden">
      <img src="product.jpg" class="w-full h-48 object-cover">
      <div class="p-4 sm:p-6">
        <h3 class="text-lg sm:text-xl font-bold mb-2">
          Product Name
        </h3>
        <p class="text-sm sm:text-base text-gray-600 mb-4">
          Product description goes here
        </p>
        <button class="w-full sm:w-auto px-4 py-2 bg-blue-600
text-white rounded">
          Buy Now
        </button>
      </div>
    </div>

    <!-- More cards... -->

  </div>
</div>
```

Responsive features:

- Grid columns adapt: 1 → 2 → 3 → 4
- Gap spacing increases on larger screens
- Card padding increases on tablet+

- Button full-width on mobile, auto on desktop
- Text size increases

Example 2: Responsive Hero Section

```
<section class="relative bg-gradient-to-r from-blue-600 to-
purple-600 text-white">
  <div class="container mx-auto px-4 py-16 sm:py-20 md:py-24
lg:py-32">

    <!-- Flex column on mobile, row on desktop -->
    <div class="flex flex-col lg:flex-row items-center gap-8
lg:gap-12">

      <!-- Text content -->
      <div class="flex-1 text-center lg:text-left">
        <h1 class="text-3xl sm:text-4xl md:text-5xl lg:text-6xl
font-bold leading-tight mb-4 md:mb-6">
          Build Amazing Products
        </h1>

        <p class="text-base sm:text-lg md:text-xl lg:text-2xl mb-
6 md:mb-8 opacity-90">
          Ship faster with our comprehensive toolkit
        </p>

        <!-- Buttons: stacked on mobile, row on tablet -->
        <div class="flex flex-col sm:flex-row gap-4 justify-
center lg:justify-start">
          <button class="px-6 py-3 md:px-8 md:py-4 bg-white text-
blue-600 font-semibold rounded-lg text-sm md:text-base">
            Get Started
```

```
          </button>
          <button class="px-6 py-3 md:px-8 md:py-4 border-2 border-white text-white font-semibold rounded-lg text-sm md:text-base">
            Learn More
          </button>
        </div>
      </div>

      <!-- Hero image: hidden on small mobile, shown on tablet+ -->
      <div class="hidden sm:block flex-1">
        <img src="hero.png" class="w-full max-w-md lg:max-w-lg mx-auto">
      </div>

    </div>
  </div>
</section>
```

Responsive features:

- Vertical padding adjusts: 64px → 80px → 96px → 128px
- Heading size: 30px → 36px → 48px → 60px
- Layout switches from column to row at `lg`
- Buttons stack on mobile, side-by-side on tablet
- Image hidden on small mobile
- Text centered on mobile, left-aligned on desktop

Example 3: Responsive Dashboard

```
<div class="min-h-screen bg-gray-100">

  <!-- Header: fixed height -->
  <header class="bg-white shadow h-16 flex items-center px-4
md:px-6">
    <h1 class="text-lg md:text-xl font-bold">Dashboard</h1>
  </header>

  <!-- Main layout -->
  <div class="flex flex-col md:flex-row">

    <!-- Sidebar: full width on mobile, fixed width on desktop --
>
    <aside class="w-full md:w-64 bg-gray-800 text-white p-4
md:min-h-[calc(100vh-4rem)]">
      <nav class="space-y-2">
        <a href="#" class="block px-3 py-2 rounded hover:bg-gray-
700">
          Dashboard
        </a>
        <a href="#" class="block px-3 py-2 rounded hover:bg-gray-
700">
          Analytics
        </a>
        <a href="#" class="block px-3 py-2 rounded hover:bg-gray-
700">
          Settings
        </a>
      </nav>
    </aside>

    <!-- Main content -->
    <main class="flex-1 p-4 md:p-6 lg:p-8">

      <!-- Stats grid -->
```

```
      <div class="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-4
gap-4 md:gap-6 mb-8">

        <div class="bg-white p-4 md:p-6 rounded-lg shadow">
          <h3 class="text-sm md:text-base text-gray-500 mb-
2">Total Users</h3>
          <p class="text-2xl md:text-3xl font-bold">12,345</p>
        </div>

        <div class="bg-white p-4 md:p-6 rounded-lg shadow">
          <h3 class="text-sm md:text-base text-gray-500 mb-
2">Revenue</h3>
          <p class="text-2xl md:text-3xl font-bold">$45,678</p>
        </div>

        <div class="bg-white p-4 md:p-6 rounded-lg shadow">
          <h3 class="text-sm md:text-base text-gray-500 mb-
2">Orders</h3>
          <p class="text-2xl md:text-3xl font-bold">789</p>
        </div>

        <div class="bg-white p-4 md:p-6 rounded-lg shadow">
          <h3 class="text-sm md:text-base text-gray-500 mb-
2">Conversion</h3>
          <p class="text-2xl md:text-3xl font-bold">3.45%</p>
        </div>

      </div>

      <!-- Chart -->
      <div class="bg-white p-4 md:p-6 rounded-lg shadow">
        <h2 class="text-lg md:text-xl font-bold mb-
4">Analytics</h2>
        <div class="h-64 md:h-96 bg-gray-100 rounded flex items-
center justify-center">
```

```
            Chart goes here
          </div>
        </div>

      </main>

    </div>
  </div>
```

Responsive features:

- Sidebar: full-width on mobile, becomes side panel on tablet+
- Stats grid: 1 col → 2 cols → 4 cols
- Padding increases at larger breakpoints
- Chart height adjusts
- Text sizes adapt

Example 4: Responsive Form

```
<div class="min-h-screen bg-gray-50 py-8 md:py-12 lg:py-16">
  <div class="container mx-auto px-4">

    <!-- Form container: wider on desktop -->
    <div class="max-w-md md:max-w-2xl mx-auto bg-white shadow-lg
rounded-lg p-6 md:p-8 lg:p-10">

      <h2 class="text-2xl md:text-3xl font-bold mb-6 md:mb-8">
        Contact Us
      </h2>

      <form class="space-y-4 md:space-y-6">
```

```
        <!-- Single column on mobile, two columns on desktop -->
        <div class="grid grid-cols-1 md:grid-cols-2 gap-4 md:gap-
6">

          <div>
            <label class="block text-sm font-medium mb-2">First
Name</label>
            <input
              type="text"
              class="w-full px-3 py-2 md:px-4 md:py-3 border
border-gray-300 rounded-lg focus:ring-2 focus:ring-blue-500
focus:border-transparent"
            >
          </div>

          <div>
            <label class="block text-sm font-medium mb-2">Last
Name</label>
            <input
              type="text"
              class="w-full px-3 py-2 md:px-4 md:py-3 border
border-gray-300 rounded-lg focus:ring-2 focus:ring-blue-500
focus:border-transparent"
            >
          </div>

        </div>

        <!-- Full width fields -->
        <div>
          <label class="block text-sm font-medium mb-
2">Email</label>
          <input
            type="email"
            class="w-full px-3 py-2 md:px-4 md:py-3 border
```

```
border-gray-300 rounded-lg focus:ring-2 focus:ring-blue-500
focus:border-transparent"
          >
        </div>

        <div>
          <label class="block text-sm font-medium mb-
2">Message</label>
          <textarea
            rows="4"
            class="w-full px-3 py-2 md:px-4 md:py-3 border
border-gray-300 rounded-lg focus:ring-2 focus:ring-blue-500
focus:border-transparent"
          ></textarea>
        </div>

        <!-- Button: full width on mobile, auto on desktop -->
        <button
          type="submit"
          class="w-full md:w-auto px-6 py-3 md:px-8 md:py-4 bg-
blue-600 hover:bg-blue-700 text-white font-semibold rounded-lg
text-sm md:text-base"
        >
          Send Message
        </button>

      </form>

    </div>

  </div>
</div>
```

Responsive features:

- Form width constrained at each breakpoint
- Two-column layout on desktop
- Input padding increases on larger screens
- Button full-width on mobile, auto on desktop
- Spacing increases on larger screens

Common Responsive Patterns

1. Responsive Spacing

```
<!-- Padding -->
<div class="p-4 md:p-6 lg:p-8 xl:p-10">
  Progressive padding
</div>

<!-- Margin -->
<div class="mb-4 md:mb-6 lg:mb-8">
  Progressive margin
</div>

<!-- Gap -->
<div class="flex gap-2 md:gap-4 lg:gap-6">
  Progressive gap
</div>
```

2. Responsive Images

```
<!-- Responsive height -->
<img
  src="hero.jpg"
  class="w-full h-48 sm:h-64 md:h-80 lg:h-96 object-cover"
```

```
>

<!-- Different images for different screens -->
<picture>
  <source media="(min-width: 1024px)" srcset="desktop.jpg">
  <source media="(min-width: 768px)" srcset="tablet.jpg">
  <img src="mobile.jpg" class="w-full">
</picture>
```

3. Responsive Order

```
<!-- Reverse order on mobile -->
<div class="flex flex-col-reverse md:flex-row">
  <div class="md:w-1/2">Content (second on mobile, first on
desktop)</div>
  <div class="md:w-1/2">Image (first on mobile, second on
desktop)</div>
</div>

<!-- Specific ordering -->
<div class="flex flex-col md:flex-row">
  <div class="order-2 md:order-1">First on desktop</div>
  <div class="order-1 md:order-2">Second on desktop</div>
  <div class="order-3 md:order-3">Third everywhere</div>
</div>
```

4. Responsive Max-Width

```
<!-- Content gets progressively wider -->
<article class="max-w-sm sm:max-w-md md:max-w-lg lg:max-w-xl
xl:max-w-2xl mx-auto">
  <h1>Article Title</h1>
```

```
  <p>Content that's readable at all sizes</p>
</article>
```

Common Mistakes to Avoid

1. Not thinking mobile-first

```
<!-- ❌ Wrong approach (desktop-first thinking) -->
<div class="text-xl md:text-base">
  This gets SMALLER on tablets, which is backwards
</div>

<!-- ✅ Correct (mobile-first) -->
<div class="text-base md:text-xl">
  This gets LARGER on tablets
</div>
```

2. Forgetting to test on actual devices

```
<!-- Looks good in browser dev tools, but... -->
<button class="text-xs">Tap me</button>
<!-- Text is too small on actual phones! -->

<!-- ✅ Better -->
<button class="text-sm md:text-base py-3 px-6">
  Tap me (touch-friendly size)
</button>
```

3. Over-engineering breakpoints

```
<!-- ❌ Too many breakpoints (confusing) -->
<div class="text-xs sm:text-sm md:text-base lg:text-lg xl:text-xl 2xl:text-2xl">
  Overkill!
</div>

<!-- ✅ Just what you need -->
<div class="text-base lg:text-xl">
  Simple and effective
</div>
```

4. Inconsistent spacing

```
<!-- ❌ Random spacing values -->
<div class="p-3 md:p-7 lg:p-11">
  Inconsistent scaling
</div>

<!-- ✅ Follow the scale -->
<div class="p-4 md:p-6 lg:p-8">
  Harmonious progression
</div>
```

5. Not considering content reflow

```
<!-- ❌ Fixed height breaks on mobile -->
<div class="h-64">
  <p>Too much text for mobile viewport...</p>
</div>

<!-- ✅ Let content determine height -->
<div class="min-h-64">
```

```
  <p>Content can expand naturally</p>
</div>
```

Key Takeaways

✅ Tailwind is **mobile-first** - base styles apply to all sizes
✅ Five breakpoints: **sm, md, lg, xl, 2xl**
✅ Syntax: `[breakpoint]:[utility]` (e.g., `md:text-xl`)
✅ Use `hidden` and `block` to show/hide at different sizes
✅ `container` class provides responsive max-widths
✅ Grid columns should adapt: `grid-cols-1 md:grid-cols-2 lg:grid-cols-3`
✅ Test on real devices, not just browser dev tools
✅ Keep it simple - don't over-specify breakpoints

Practice Assignments

Assignment 1 (Easy): Responsive Typography

Create a simple page with:

- Heading that scales from text-2xl to text-6xl
- Paragraph text from text-sm to text-lg
- Different line heights at different breakpoints
- Test by resizing your browser

Assignment 2 (Easy): Show/Hide Elements

Build a navigation bar that:

- Shows hamburger menu icon on mobile

- Shows horizontal menu links on desktop
- Hides hamburger on desktop
- Hides links on mobile Use `hidden`, `block`, and `md:` prefix.

Assignment 3 (Medium): Responsive Card Grid

Create a product grid:

- 1 column on mobile
- 2 columns on small tablets (sm)
- 3 columns on tablets (md)
- 4 columns on desktop (lg)
- Each card has image, title, description, and button
- Button should be full-width on mobile, auto on desktop
- Gap should increase at larger breakpoints

Assignment 4 (Medium): Responsive Hero Section

Build a hero section with:

- Vertical layout (stacked) on mobile
- Horizontal layout (side-by-side) on desktop
- Image on right for desktop, above text on mobile
- Different heading sizes at each breakpoint
- Buttons stacked on mobile, side-by-side on tablet
- Responsive padding and spacing

Assignment 5 (Hard): Full Responsive Landing Page

Create a complete landing page with:

1. **Navigation**: Mobile menu vs desktop links
2. **Hero**: Image/text layout changes

3. **Features**: 3-column grid (responsive)
4. **Testimonials**: Slider on mobile, grid on desktop
5. **Footer**: 4 columns on desktop, stacked on mobile

Requirements:

- Must work perfectly from 320px to 1920px wide
- Smooth transitions between breakpoints
- Proper touch targets on mobile (44px minimum)
- Readable text at all sizes
- No horizontal scrolling on any size

Bonus challenges:

- Add a mobile menu that slides in from the side
- Use different images for mobile vs desktop
- Add "scroll to top" button that only shows on mobile

Ready for **Module 7: State Variants & Interactions**? Let me know!

1/3/2026, 8:47:38 PM

MODULE 7: STATE VARIANTS & INTERACTIONS

Hover States (hover:)

The `hover:` prefix applies styles when the user hovers over an element (primarily for devices with a mouse/cursor).

Basic Hover

```
<!-- Background color change -->
<button class="bg-blue-500 hover:bg-blue-700 text-white px-4 py-2
rounded">
  Hover me
</button>

<!-- Text color change -->
<a href="#" class="text-blue-600 hover:text-blue-800">
  Hover link
</a>

<!-- Multiple properties -->
<div class="bg-white hover:bg-gray-50 hover:shadow-lg p-6 rounded
transition">
  Hover card
</div>
```

Comparison to vanilla CSS:

```
/* Vanilla CSS */
.button {
  background-color: #3b82f6;
}

.button:hover {
  background-color: #1d4ed8;
}
```

```
<!-- Tailwind -->
<button class="bg-blue-500 hover:bg-blue-700">
  Button
</button>
```

Hover with Transforms

```
<!-- Scale up on hover -->
<button class="transform hover:scale-105 transition bg-blue-500
text-white px-6 py-3 rounded">
  Scale on hover
</button>

<!-- Translate up -->
<div class="transform hover:-translate-y-1 transition bg-white
shadow p-6 rounded">
  Lifts up on hover
</div>

<!-- Rotate -->
<button class="transform hover:rotate-6 transition bg-purple-500
text-white p-4 rounded">
  Rotate me
</button>
```

Visual effects:

```
Normal:      [Button]
Hover:       [Button] ↑ (lifted)

Normal:      [Card]
Scale:       [Card] (slightly larger)

Normal:      [Icon]
Rotate:      [Icon]↻ (rotated)
```

Hover on Links

```
<!-- Underline appears on hover -->
<a href="#" class="text-blue-600 hover:underline">
  Underline on hover
</a>

<!-- Underline disappears on hover -->
<a href="#" class="underline hover:no-underline">
  Underline disappears
</a>

<!-- Color and underline -->
<a href="#" class="text-gray-700 hover:text-blue-600
hover:underline transition">
  Color change + underline
</a>
```

Hover on Cards

```
<div class="bg-white shadow hover:shadow-xl rounded-lg overflow-
hidden transition-shadow duration-300">
  <img src="product.jpg" class="w-full h-48 object-cover">
  <div class="p-6">
    <h3 class="text-xl font-bold mb-2">Product Name</h3>
    <p class="text-gray-600 mb-4">Description here</p>
    <button class="bg-blue-500 hover:bg-blue-600 text-white px-4
py-2 rounded transition">
      Buy Now
    </button>
  </div>
</div>
```

Effect: Card shadow intensifies when hovering anywhere on the card.

Hover on Images

```
<!-- Image zoom effect -->
<div class="overflow-hidden rounded-lg">
  <img
    src="photo.jpg"
    class="w-full h-64 object-cover transform hover:scale-110
transition duration-500"
  >
</div>

<!-- Overlay appears on hover -->
<div class="relative group">
  <img src="photo.jpg" class="w-full h-64 object-cover">
  <div class="absolute inset-0 bg-black bg-opacity-0 group-
hover:bg-opacity-50 transition-opacity duration-300 flex items-
center justify-center">
    <span class="text-white opacity-0 group-hover:opacity-100
transition-opacity">
      View Details
    </span>
  </div>
</div>
```

Focus States (focus:)

The `focus:` prefix applies when an element receives keyboard focus (important for accessibility!).

Focus on Inputs

```
<!-- Change border color on focus -->
<input
  type="text"
  class="border border-gray-300 focus:border-blue-500
focus:outline-none px-4 py-2 rounded"
  placeholder="Focus me"
>

<!-- Add ring on focus -->
<input
  type="text"
  class="border border-gray-300 focus:ring-2 focus:ring-blue-500
focus:border-transparent px-4 py-2 rounded"
  placeholder="Ring appears on focus"
>

<!-- Multiple focus effects -->
<input
  type="text"
  class="border border-gray-300 focus:border-blue-500 focus:ring-
2 focus:ring-blue-200 focus:outline-none px-4 py-2 rounded
transition"
  placeholder="Multiple effects"
>
```

Why `focus:outline-none`? Removes the default browser outline so you can apply custom focus styles.

Focus on Buttons

```
<!-- Focus ring for keyboard navigation -->
<button class="bg-blue-500 hover:bg-blue-600 focus:ring-4
focus:ring-blue-300 text-white px-6 py-3 rounded">
  Keyboard accessible
```

```
</button>

<!-- Focus with outline offset -->
<button class="bg-green-500 hover:bg-green-600 focus:outline-none
focus:ring-2 focus:ring-green-400 focus:ring-offset-2 text-white
px-6 py-3 rounded">
  Ring with offset
</button>
```

Accessibility tip: Always provide visible focus indicators for keyboard users!

Focus Within

`focus-within:` applies styles when any child element has focus:

```
<div class="border-2 border-gray-300 focus-within:border-blue-500
p-4 rounded">
  <label class="block mb-2">Name</label>
  <input
    type="text"
    class="w-full border border-gray-300 px-3 py-2 rounded
focus:outline-none"
  >
</div>
```

Effect: The container's border changes color when the input inside is focused.

Complete Form Example with Focus States

```
<form class="space-y-4 max-w-md">

  <!-- Text input -->
  <div>
    <label class="block text-sm font-medium mb-2">Email</label>
```

```
    <input
      type="email"
      class="w-full border border-gray-300 rounded-lg px-4 py-2
             focus:border-blue-500 focus:ring-2 focus:ring-blue-
200
             focus:outline-none transition"
      placeholder="you@example.com"
    >
  </div>

  <!-- Textarea -->
  <div>
    <label class="block text-sm font-medium mb-2">Message</label>
    <textarea
      rows="4"
      class="w-full border border-gray-300 rounded-lg px-4 py-2
             focus:border-blue-500 focus:ring-2 focus:ring-blue-
200
             focus:outline-none transition"
      placeholder="Your message..."
    ></textarea>
  </div>

  <!-- Submit button -->
  <button
    type="submit"
    class="w-full bg-blue-500 hover:bg-blue-600
           focus:ring-4 focus:ring-blue-300 focus:outline-none
           text-white font-semibold px-6 py-3 rounded-lg
transition"
  >
    Send Message
  </button>

</form>
```

Active States (active:)

The `active:` prefix applies when an element is being clicked/pressed.

```
<!-- Button press effect -->
<button class="bg-blue-500 hover:bg-blue-600 active:bg-blue-700
text-white px-6 py-3 rounded">
  Press me
</button>

<!-- Scale down when pressed -->
<button class="bg-green-500 hover:bg-green-600 active:scale-95
transform text-white px-6 py-3 rounded transition">
  Squish on click
</button>

<!-- Inset shadow (pressed appearance) -->
<button class="bg-purple-500 hover:bg-purple-600 active:shadow-
inner text-white px-6 py-3 rounded shadow-lg transition">
  Press effect
</button>
```

Visual progression:

```
Normal:  [Button]
Hover:   [Button] (lighter)
Active:  [Button] (darker, pressed down)
```

Complete Button States

```
<button class="
  bg-blue-500
```

```
  hover:bg-blue-600
  active:bg-blue-700
  focus:ring-4
  focus:ring-blue-300
  transform
  active:scale-95
  transition
  text-white font-semibold px-6 py-3 rounded-lg
">
  Full State Button
</button>
```

States in order:

1. Default: `bg-blue-500`
2. Hover: `bg-blue-600` (mouse over)
3. Focus: `focus:ring-4` (keyboard navigation)
4. Active: `bg-blue-700` + `active:scale-95` (being clicked)

Group Hover Patterns

The `group` and `group-hover:` pattern allows child elements to react when the parent is hovered.

Basic Group Hover

```
<div class="group bg-white p-6 rounded-lg shadow hover:shadow-xl
transition">

  <h3 class="text-xl font-bold text-gray-900 group-hover:text-
blue-600 transition">
    Hover the card
```

```
  </h3>

  <p class="text-gray-600 group-hover:text-gray-900 transition">
    Text changes color when you hover anywhere on the card
  </p>

  <button class="mt-4 text-blue-600 group-hover:text-blue-800
transition">
    Learn More →
  </button>

</div>
```

Effect: Hovering anywhere on the card changes the text colors inside.

Group Hover with Images

```
<a href="#" class="group block relative overflow-hidden rounded-
lg">

  <!-- Image scales on group hover -->
  <img
    src="product.jpg"
    class="w-full h-64 object-cover transform group-hover:scale-
110 transition duration-500"
  >

  <!-- Overlay appears on group hover -->
  <div class="absolute inset-0 bg-black bg-opacity-0 group-
hover:bg-opacity-40 transition-opacity">
  </div>

  <!-- Text content -->
  <div class="absolute bottom-0 left-0 right-0 p-6 text-white
transform translate-y-2 group-hover:translate-y-0 transition">
```

```
    <h3 class="text-2xl font-bold">Product Name</h3>
    <p class="opacity-0 group-hover:opacity-100 transition">
      View details
    </p>
  </div>

</a>
```

Effect:

- Hover anywhere on the card
- Image zooms
- Overlay darkens
- Text slides up
- Description fades in

Navigation with Group Hover

```
<nav class="bg-white shadow">
  <div class="flex space-x-8 px-6 py-4">

    <!-- Nav item with dropdown -->
    <div class="group relative">
      <button class="text-gray-700 group-hover:text-blue-600
transition">
        Products
        <svg class="inline w-4 h-4 ml-1 transform group-
hover:rotate-180 transition">
          <!-- Chevron icon -->
        </svg>
      </button>

      <!-- Dropdown (hidden by default, shows on group hover) -->
      <div class="absolute left-0 mt-2 w-48 bg-white shadow-lg
```

```
rounded-lg opacity-0 invisible group-hover:opacity-100 group-
hover:visible transition-all">
        <a href="#" class="block px-4 py-2 hover:bg-gray-
100">Product 1</a>
        <a href="#" class="block px-4 py-2 hover:bg-gray-
100">Product 2</a>
        <a href="#" class="block px-4 py-2 hover:bg-gray-
100">Product 3</a>
      </div>
    </div>

  </div>
</nav>
```

Card Grid with Group Hover

```
<div class="grid grid-cols-1 md:grid-cols-3 gap-6">

  <!-- Card 1 -->
  <div class="group bg-white rounded-lg shadow overflow-hidden
cursor-pointer">
    <div class="relative overflow-hidden">
      <img
        src="card1.jpg"
        class="w-full h-48 object-cover transform group-
hover:scale-105 transition duration-300"
      >
      <div class="absolute top-4 right-4 bg-blue-500 text-white
px-3 py-1 rounded-full text-sm opacity-0 group-hover:opacity-100
transition">
        New
      </div>
    </div>
```

```
    <div class="p-6">
      <h3 class="text-xl font-bold mb-2 group-hover:text-blue-600
transition">
        Card Title
      </h3>
      <p class="text-gray-600 mb-4">
        Card description goes here
      </p>
      <div class="flex items-center text-blue-600 group-
hover:text-blue-800 transition">
        <span>Learn More</span>
        <svg class="w-4 h-4 ml-2 transform group-hover:translate-
x-2 transition">
          <!-- Arrow icon -->
        </svg>
      </div>
    </div>
  </div>

  <!-- More cards... -->

</div>
```

Peer Modifiers

The `peer` pattern allows an element to affect its siblings based on its state.

Key difference from `group`:

- `group` / `group-hover`: Parent affects children
- `peer` / `peer-*`: Sibling affects siblings

Peer Checked (Checkbox/Radio)

```
<!-- Custom checkbox -->
<label class="flex items-center cursor-pointer">
  <input
    type="checkbox"
    class="peer sr-only"
  >
  <div class="w-11 h-6 bg-gray-300 rounded-full peer-checked:bg-
blue-500 transition relative">
    <div class="w-4 h-4 bg-white rounded-full absolute top-1
left-1 peer-checked:translate-x-5 transition"></div>
  </div>
  <span class="ml-3 text-gray-700 peer-checked:text-blue-600">
    Toggle me
  </span>
</label>
```

Effect: When checkbox is checked, the background turns blue and the dot slides right.

Peer Focus (Floating Labels)

```
<div class="relative">
  <input
    type="text"
    id="email"
    class="peer w-full border border-gray-300 rounded-lg px-4 pt-
6 pb-2 focus:border-blue-500 focus:outline-none transition"
    placeholder=" "
  >
  <label
    for="email"
    class="absolute left-4 top-2 text-xs text-gray-500 peer-
placeholder-shown:text-base peer-placeholder-shown:top-4 peer-
```

```
focus:text-xs peer-focus:top-2 peer-focus:text-blue-500
transition-all"
  >
    Email Address
  </label>
</div>
```

Effect: Label floats up when input is focused or has content.

Toggle Switch with Peer

```
<label class="inline-flex items-center cursor-pointer">
  <input type="checkbox" class="peer sr-only">

  <!-- Toggle background -->
  <div class="
    relative w-14 h-8
    bg-gray-300 peer-checked:bg-green-500
    rounded-full
    transition-colors
    peer-focus:ring-4 peer-focus:ring-green-200
  ">
    <!-- Toggle circle -->
    <div class="
      absolute top-1 left-1
      w-6 h-6
      bg-white
      rounded-full
      shadow-md
      transform peer-checked:translate-x-6
      transition-transform
    "></div>
  </div>

  <span class="ml-3 text-gray-700 peer-checked:text-green-600
```

```
peer-checked:font-medium">
    Enabled
  </span>
</label>
```

Radio Button Group with Peer

```
<div class="space-y-3">

  <!-- Option 1 -->
  <label class="flex items-center p-4 border-2 border-gray-300
rounded-lg cursor-pointer peer-checked:border-blue-500 peer-
checked:bg-blue-50">
    <input
      type="radio"
      name="plan"
      class="peer sr-only"
    >
    <div class="flex-1">
      <div class="font-semibold peer-checked:text-blue-600">Basic
Plan</div>
      <div class="text-sm text-gray-500">$9/month</div>
    </div>
    <svg class="w-6 h-6 text-blue-500 hidden peer-checked:block">
      <!-- Checkmark icon -->
    </svg>
  </label>

  <!-- Option 2 -->
  <label class="flex items-center p-4 border-2 border-gray-300
rounded-lg cursor-pointer peer-checked:border-blue-500 peer-
checked:bg-blue-50">
    <input
      type="radio"
```

```
      name="plan"
      class="peer sr-only"
    >
    <div class="flex-1">
      <div class="font-semibold peer-checked:text-blue-600">Pro
Plan</div>
      <div class="text-sm text-gray-500">$29/month</div>
    </div>
    <svg class="w-6 h-6 text-blue-500 hidden peer-checked:block">
      <!-- Checkmark icon -->
    </svg>
  </label>

</div>
```

Note: `sr-only` (screen reader only) hides the input visually but keeps it accessible.

Disabled, Checked, Required States

Disabled State

```
<!-- Disabled button -->
<button
  disabled
  class="bg-blue-500 text-white px-6 py-3 rounded disabled:bg-
gray-300 disabled:text-gray-500 disabled:cursor-not-allowed"
>
  Disabled Button
</button>

<!-- Disabled input -->
```

```
<input
  type="text"
  disabled
  class="border border-gray-300 px-4 py-2 rounded disabled:bg-
gray-100 disabled:text-gray-500 disabled:cursor-not-allowed"
  value="Can't edit this"
>
```

Checked State

```
<!-- Styled checkbox -->
<label class="inline-flex items-center">
  <input
    type="checkbox"
    class="w-5 h-5 text-blue-500 border-gray-300 rounded
focus:ring-2 focus:ring-blue-500 checked:bg-blue-500"
  >
  <span class="ml-2">Agree to terms</span>
</label>

<!-- Custom checkbox appearance -->
<label class="flex items-center cursor-pointer">
  <div class="relative">
    <input type="checkbox" class="peer sr-only">
    <div class="w-6 h-6 border-2 border-gray-300 rounded peer-
checked:bg-blue-500 peer-checked:border-blue-500 transition">
</div>
    <svg class="absolute top-0.5 left-0.5 w-5 h-5 text-white
hidden peer-checked:block">
      <path d="M20 6L9 17l-5-5" stroke="currentColor" stroke-
width="2" fill="none"/>
    </svg>
  </div>
```

```
  <span class="ml-3">Custom styled</span>
</label>
```

Required State

```
<form class="space-y-4">

  <div>
    <label class="block mb-2">
      Email
      <span class="text-red-500">*</span>
    </label>
    <input
      type="email"
      required
      class="w-full border border-gray-300 px-4 py-2 rounded
             focus:border-blue-500 focus:ring-2 focus:ring-blue-
200
             invalid:border-red-500 invalid:text-red-600
             focus:invalid:border-red-500 focus:invalid:ring-red-
200"
      placeholder="you@example.com"
    >
    <p class="mt-1 text-sm text-red-500 hidden peer-
invalid:block">
      Please enter a valid email
    </p>
  </div>

  <button
    type="submit"
    class="bg-blue-500 hover:bg-blue-600 text-white px-6 py-3
rounded"
  >
```

```
    Submit
  </button>

</form>
```

Valid/Invalid States

```
<!-- Valid state (green) -->
<input
  type="email"
  class="border border-gray-300 px-4 py-2 rounded
         valid:border-green-500 valid:text-green-600
         invalid:border-red-500 invalid:text-red-600"
>

<!-- With icons -->
<div class="relative">
  <input
    type="email"
    required
    class="w-full border border-gray-300 pl-4 pr-10 py-2 rounded
           valid:border-green-500 invalid:border-red-500"
  >
  <!-- Validation icon -->
  <svg class="absolute right-3 top-3 w-5 h-5 text-green-500
hidden valid:block">
    <!-- Checkmark -->
  </svg>
  <svg class="absolute right-3 top-3 w-5 h-5 text-red-500 hidden
invalid:block">
    <!-- X mark -->
  </svg>
</div>
```

Dark Mode Support (dark:)

Tailwind provides dark mode support out of the box (we'll cover this more in future modules, but here's an intro).

Enabling Dark Mode

In `tailwind.config.js`:

```
export default {
  darkMode: 'class', // or 'media' for system preference
  // rest of config...
}
```

Using Dark Mode Classes

```
<!-- Light background, dark in dark mode -->
<div class="bg-white dark:bg-gray-900">

  <!-- Text changes in dark mode -->
  <h1 class="text-gray-900 dark:text-white">
    Heading
  </h1>

  <p class="text-gray-600 dark:text-gray-300">
    Body text
  </p>

  <!-- Button adapts -->
  <button class="bg-blue-500 dark:bg-blue-600 hover:bg-blue-600
dark:hover:bg-blue-700 text-white px-4 py-2 rounded">
    Button
  </button>
```

```
</div>
```

Complete Dark Mode Card

```
<div class="bg-white dark:bg-gray-800 shadow-lg dark:shadow-gray-900/50 rounded-lg p-6">

  <h2 class="text-2xl font-bold text-gray-900 dark:text-white mb-4">
    Card Title
  </h2>

  <p class="text-gray-600 dark:text-gray-300 mb-4">
    This card works in both light and dark mode seamlessly.
  </p>

  <div class="flex space-x-4">
    <button class="bg-blue-500 dark:bg-blue-600 hover:bg-blue-600 dark:hover:bg-blue-700 text-white px-4 py-2 rounded">
      Primary
    </button>
    <button class="bg-gray-200 dark:bg-gray-700 hover:bg-gray-300 dark:hover:bg-gray-600 text-gray-900 dark:text-white px-4 py-2 rounded">
      Secondary
    </button>
  </div>

</div>
```

To activate dark mode, add `class="dark"` to your `<html>` tag:

```
<html class="dark">
```

Transition and Animation Utilities

Transition

Controls which properties transition and the duration:

```
<!-- Default transition (all properties, 150ms) -->
<button class="bg-blue-500 hover:bg-blue-700 transition">
  Smooth color change
</button>

<!-- Specific properties -->
<div class="transition-colors">Colors only</div>
<div class="transition-opacity">Opacity only</div>
<div class="transition-transform">Transforms only</div>
<div class="transition-all">All properties</div>

<!-- Duration -->
<div class="transition duration-75">75ms</div>
<div class="transition duration-150">150ms (default)</div>
<div class="transition duration-300">300ms</div>
<div class="transition duration-500">500ms</div>
<div class="transition duration-700">700ms</div>
<div class="transition duration-1000">1000ms (1s)</div>
```

Easing (Timing Functions)

```
<div class="transition ease-linear">Linear</div>
<div class="transition ease-in">Ease in</div>
```

```
<div class="transition ease-out">Ease out</div>
<div class="transition ease-in-out">Ease in-out (default)</div>
```

Delay

```
<div class="transition delay-75">75ms delay</div>
<div class="transition delay-150">150ms delay</div>
<div class="transition delay-300">300ms delay</div>
<div class="transition delay-500">500ms delay</div>
```

Complete Transition Example

```
<button class="
  bg-blue-500 hover:bg-blue-600
  transform hover:scale-105 hover:-translate-y-1
  shadow-md hover:shadow-xl
  text-white px-6 py-3 rounded-lg
  transition-all duration-300 ease-in-out
">
  Smooth Button
</button>
```

Animations

Tailwind includes some built-in animations:

```
<!-- Spin -->
<div class="animate-spin">
  <svg class="w-8 h-8"><!-- Loading spinner --></svg>
</div>

<!-- Ping (expanding circle) -->
<div class="relative">
```

```
  <span class="absolute animate-ping inline-flex h-3 w-3 rounded-full bg-blue-400 opacity-75"></span>
  <span class="relative inline-flex rounded-full h-3 w-3 bg-blue-500"></span>
</div>

<!-- Pulse (fade in/out) -->
<div class="animate-pulse bg-gray-200 h-4 w-full rounded"></div>

<!-- Bounce -->
<div class="animate-bounce">
  ↓
</div>
```

Loading Skeleton with Animation

```
<div class="animate-pulse space-y-4">
  <div class="h-4 bg-gray-200 rounded w-3/4"></div>
  <div class="h-4 bg-gray-200 rounded"></div>
  <div class="h-4 bg-gray-200 rounded w-5/6"></div>
</div>
```

Notification Badge with Ping

```
<button class="relative">
  <svg class="w-6 h-6"><!-- Bell icon --></svg>

  <!-- Badge with ping animation -->
  <span class="absolute -top-1 -right-1">
    <span class="absolute inline-flex h-full w-full rounded-full bg-red-400 opacity-75 animate-ping"></span>
    <span class="relative inline-flex rounded-full h-3 w-3 bg-red-500"></span>
```

```
  </span>
</button>
```

Common Mistakes to Avoid

1. Forgetting transitions

```
<!-- ❌ Abrupt change -->
<button class="bg-blue-500 hover:bg-blue-700">
  Instant change
</button>

<!-- ✅ Smooth change -->
<button class="bg-blue-500 hover:bg-blue-700 transition">
  Smooth change
</button>
```

2. Not considering keyboard users

```
<!-- ❌ No focus state -->
<button class="bg-blue-500 hover:bg-blue-600">
  No focus indicator
</button>

<!-- ✅ Includes focus state -->
<button class="bg-blue-500 hover:bg-blue-600 focus:ring-4
focus:ring-blue-300">
  Keyboard accessible
</button>
```

3. Using hover on touch devices

```
<!-- ❌ Hover stays "stuck" on mobile -->
<div class="hover:bg-blue-500">
  Mobile issues
</div>

<!-- ✅ Better approach -->
<div class="active:bg-blue-500 md:hover:bg-blue-500">
  Active on mobile, hover on desktop
</div>
```

4. Confusing group and peer

```
<!-- ❌ Won't work - peer affects siblings, not children -->
<div class="peer">
  <span class="peer-hover:text-blue-500">Wrong</span>
</div>

<!-- ✅ Correct - use group -->
<div class="group">
  <span class="group-hover:text-blue-500">Correct</span>
</div>
```

Key Takeaways

✅ `hover:` for mouse interactions, **always add transitions**
✅ `focus:` for keyboard navigation - **critical for accessibility**
✅ `active:` for button press effects
✅ `group` + `group-hover:` parent affects children
✅ `peer` + `peer-*:` sibling affects siblings

✅ `disabled:`, `checked:`, `required:` for form states
✅ `dark:` for dark mode support
✅ Use `transition` + `duration` for smooth animations
✅ Combine multiple states: `hover:`, `focus:`, `active:`

Practice Assignments

Assignment 1 (Easy): Button States

Create 3 buttons with complete state management:

- Default state
- Hover state (color change)
- Focus state (ring)
- Active state (scale down)
- Include smooth transitions

Assignment 2 (Easy): Card with Group Hover

Build a product card where:

- Image zooms on hover
- Title color changes
- "Learn More" arrow slides right
- Shadow intensifies
- All triggered by hovering anywhere on the card

Assignment 3 (Medium): Custom Toggle Switch

Create a custom toggle switch using `peer`:

- Looks like iOS toggle

- Smooth slide animation
- Background changes color when checked
- Label text changes color
- Include focus state for accessibility

Assignment 4 (Medium): Interactive Form

Build a form with:

- Floating labels (using peer-focus)
- Custom styled checkboxes/radios
- Validation states (valid/invalid)
- Focus rings on all inputs
- Disabled state for submit button until form is valid

Assignment 5 (Hard): Mega Menu with Group Hover

Create a navigation with dropdown mega menu:

- Desktop: Dropdown appears on hover
- Dropdowns have fade-in animation
- Icons rotate when dropdown is open
- Nested items with group hover effects
- Mobile: Click to open (no hover)
- Smooth transitions throughout

Bonus: Add dark mode support to all interactive elements.

Ready for **Module 8: Borders, Shadows & Effects**? Let me know!

1/4/2026, 8:19:09 PM

MODULE 8: BORDERS, SHADOWS & EFFECTS

Border Width, Style, Radius

Border Width

Tailwind provides utilities for border width on all sides or individual sides:

```
<!-- All sides -->
<div class="border">1px border (default)</div>
<div class="border-0">No border</div>
<div class="border-2">2px border</div>
<div class="border-4">4px border</div>
<div class="border-8">8px border</div>

<!-- Individual sides -->
<div class="border-t">Top border only</div>
<div class="border-r">Right border only</div>
<div class="border-b">Bottom border only</div>
<div class="border-l">Left border only</div>

<!-- Specific widths on specific sides -->
<div class="border-t-2">2px top border</div>
<div class="border-b-4">4px bottom border</div>

<!-- Horizontal and vertical -->
<div class="border-x">Left and right borders</div>
<div class="border-y">Top and bottom borders</div>
<div class="border-x-2">2px left and right</div>
```

Visual representation:

```
border:       ┌──────┐
              │      │
              └──────┘

border-t:     ┌──────┐
              │      │
              └      ┘

border-b:     ┌      ┐
              │      │
              └──────┘

border-x:     ┌│    │┐
              │|    |│
              └│    │┘
```

Border Style

```
<div class="border-solid">Solid border (default)</div>
<div class="border-dashed">Dashed border</div>
<div class="border-dotted">Dotted border</div>
<div class="border-double">Double border</div>
<div class="border-none">No border</div>
```

Common pattern - dividers:

```
<!-- Solid divider -->
<div class="border-b border-gray-200"></div>

<!-- Dashed divider -->
<div class="border-b border-dashed border-gray-300"></div>
```

```
<!-- Dotted divider -->
<div class="border-b border-dotted border-gray-400"></div>
```

Border Radius (Rounded Corners)

```
<!-- Uniform radius -->
<div class="rounded-none">No rounding (square)</div>
<div class="rounded-sm">Small radius (2px)</div>
<div class="rounded">Default radius (4px)</div>
<div class="rounded-md">Medium radius (6px)</div>
<div class="rounded-lg">Large radius (8px)</div>
<div class="rounded-xl">Extra large (12px)</div>
<div class="rounded-2xl">2XL radius (16px)</div>
<div class="rounded-3xl">3XL radius (24px)</div>
<div class="rounded-full">Full circle/pill shape</div>

<!-- Individual corners -->
<div class="rounded-t-lg">Top corners rounded</div>
<div class="rounded-r-lg">Right corners rounded</div>
<div class="rounded-b-lg">Bottom corners rounded</div>
<div class="rounded-l-lg">Left corners rounded</div>

<!-- Specific corners -->
<div class="rounded-tl-lg">Top-left only</div>
<div class="rounded-tr-lg">Top-right only</div>
<div class="rounded-br-lg">Bottom-right only</div>
<div class="rounded-bl-lg">Bottom-left only</div>
```

Real-World Border Examples

1. Card with border:

```
<div class="border border-gray-200 rounded-lg p-6 bg-white
shadow-sm">
```

```
  <h3 class="text-xl font-bold mb-2">Card Title</h3>
  <p class="text-gray-600">Card with subtle border and rounded
corners</p>
</div>
```

2. Accent border:

```
<div class="border-l-4 border-blue-500 bg-blue-50 p-4 rounded">
  <h4 class="font-bold text-blue-900">Info</h4>
  <p class="text-blue-800">Important information with left
accent</p>
</div>
```

3. Avatar/profile image:

```
<!-- Circle avatar -->
<img
  src="avatar.jpg"
  class="w-16 h-16 rounded-full border-2 border-white shadow"
>

<!-- Square with rounded corners -->
<img
  src="avatar.jpg"
  class="w-16 h-16 rounded-lg border-2 border-gray-300"
>
```

4. Button variations:

```
<!-- Rounded button -->
<button class="bg-blue-500 text-white px-6 py-2 rounded-lg">
  Standard
</button>
```

```
<!-- Pill button -->
<button class="bg-green-500 text-white px-6 py-2 rounded-full">
  Pill Shape
</button>

<!-- Square button -->
<button class="bg-red-500 text-white px-6 py-2 rounded-none">
  Square
</button>
```

5. Input with border:

```
<input
  type="text"
  class="w-full px-4 py-2 border-2 border-gray-300 rounded-lg
focus:border-blue-500 focus:outline-none"
  placeholder="Enter text"
>
```

6. Tabs with borders:

```
<div class="border-b border-gray-200">
  <nav class="flex space-x-8">
    <a href="#" class="py-4 px-1 border-b-2 border-blue-500 text-
blue-600 font-medium">
      Active Tab
    </a>
    <a href="#" class="py-4 px-1 border-b-2 border-transparent
hover:border-gray-300 text-gray-500 hover:text-gray-700">
      Tab 2
    </a>
    <a href="#" class="py-4 px-1 border-b-2 border-transparent
hover:border-gray-300 text-gray-500 hover:text-gray-700">
      Tab 3
```

```
    </a>
  </nav>
</div>
```

Box Shadows

Tailwind provides a range of shadow utilities for depth and elevation.

Shadow Scale

```
<div class="shadow-sm">Small shadow (subtle)</div>
<div class="shadow">Default shadow (base)</div>
<div class="shadow-md">Medium shadow</div>
<div class="shadow-lg">Large shadow</div>
<div class="shadow-xl">Extra large shadow</div>
<div class="shadow-2xl">2XL shadow (dramatic)</div>
<div class="shadow-none">No shadow</div>
```

Visual representation (elevation):

```
shadow-sm:    Card barely lifted
shadow:       Card slightly lifted
shadow-md:    Card moderately lifted
shadow-lg:    Card clearly lifted
shadow-xl:    Card floating
shadow-2xl:   Card far above surface
```

Inner Shadow

```
<div class="shadow-inner">
  Inset shadow (pressed appearance)
```

```
</div>
```

Use case - pressed button:

```
<button class="bg-gray-200 shadow-inner px-6 py-3 rounded-lg">
  Looks pressed
</button>
```

Shadow Colors

You can customize shadow colors (requires Tailwind v3.0+):

```
<!-- Colored shadows -->
<div class="shadow-lg shadow-blue-500/50">Blue shadow</div>
<div class="shadow-lg shadow-red-500/50">Red shadow</div>
<div class="shadow-xl shadow-purple-500/30">Purple shadow
(lighter)</div>
```

The `/50` means 50% opacity.

Real-World Shadow Examples

1. Card elevation system:

```
<!-- Resting card -->
<div class="bg-white shadow rounded-lg p-6">
  Default elevation
</div>

<!-- Hovered card (lifts) -->
<div class="bg-white shadow hover:shadow-lg transition-shadow
rounded-lg p-6">
  Lifts on hover
</div>
```

```
<!-- Active/clicked card -->
<div class="bg-white shadow-lg hover:shadow-2xl transition-shadow
rounded-lg p-6">
  High elevation
</div>
```

2. Button with shadow:

```
<button class="bg-blue-500 text-white px-6 py-3 rounded-lg
shadow-md hover:shadow-lg transition-shadow">
  Elevated button
</button>
```

3. Floating action button (FAB):

```
<button class="fixed bottom-8 right-8 w-14 h-14 bg-blue-500 text-
white rounded-full shadow-lg hover:shadow-2xl transition-shadow
flex items-center justify-center">
  +
</button>
```

4. Modal with shadow:

```
<div class="fixed inset-0 bg-black bg-opacity-50 flex items-
center justify-center">
  <div class="bg-white rounded-xl shadow-2xl p-8 max-w-md w-
full">
    <h2 class="text-2xl font-bold mb-4">Modal Title</h2>
    <p class="text-gray-600 mb-6">Modal content here</p>
    <button class="bg-blue-500 text-white px-4 py-2
rounded">Close</button>
```

```
  </div>
</div>
```

5. Image with shadow:

```
<img
  src="photo.jpg"
  class="w-full rounded-lg shadow-xl"
>
```

6. Dropdown with shadow:

```
<div class="absolute top-full left-0 mt-2 w-48 bg-white rounded-
lg shadow-lg py-2">
  <a href="#" class="block px-4 py-2 hover:bg-gray-100">Option
1</a>
  <a href="#" class="block px-4 py-2 hover:bg-gray-100">Option
2</a>
  <a href="#" class="block px-4 py-2 hover:bg-gray-100">Option
3</a>
</div>
```

Drop Shadows

Drop shadows are applied to the **shape** of an element, not its box. Useful for SVGs and images with transparency.

```
<!-- Drop shadow scale -->
<img src="logo.png" class="drop-shadow-sm">
<img src="logo.png" class="drop-shadow">
<img src="logo.png" class="drop-shadow-md">
```

```
<img src="logo.png" class="drop-shadow-lg">
<img src="logo.png" class="drop-shadow-xl">
<img src="logo.png" class="drop-shadow-2xl">
<img src="logo.png" class="drop-shadow-none">
```

Comparison:

```
<!-- Box shadow: shadow around the rectangle -->
<img src="star.png" class="shadow-xl">
<!-- Shadow is a rectangle around the image -->

<!-- Drop shadow: shadow follows the star shape -->
<img src="star.png" class="drop-shadow-xl">
<!-- Shadow is shaped like the star -->
```

Best use cases:

- PNG images with transparency
- SVG icons
- Images with irregular shapes
- Text (when using filters)

Example - icon with drop shadow:

```
<svg class="w-12 h-12 text-blue-500 drop-shadow-lg">
  <!-- Icon paths -->
</svg>
```

Opacity

Control the opacity (transparency) of entire elements:

```
<div class="opacity-0">Invisible (0%)</div>
<div class="opacity-5">5% opacity</div>
<div class="opacity-10">10% opacity</div>
<div class="opacity-20">20% opacity</div>
<div class="opacity-25">25% opacity</div>
<div class="opacity-30">30% opacity</div>
<div class="opacity-40">40% opacity</div>
<div class="opacity-50">50% opacity</div>
<div class="opacity-60">60% opacity</div>
<div class="opacity-70">70% opacity</div>
<div class="opacity-75">75% opacity</div>
<div class="opacity-80">80% opacity</div>
<div class="opacity-90">90% opacity</div>
<div class="opacity-95">95% opacity</div>
<div class="opacity-100">Fully visible (100%)</div>
```

Opacity with Hover

```
<!-- Fade in on hover -->
<img
  src="photo.jpg"
  class="opacity-0 hover:opacity-100 transition-opacity duration-
300"
>

<!-- Image overlay -->
<div class="relative">
  <img src="photo.jpg" class="w-full">
  <div class="absolute inset-0 bg-black opacity-0 hover:opacity-
50 transition-opacity">
  </div>
</div>
```

Opacity for Loading States

```
<!-- Loading skeleton -->
<div class="space-y-3">
  <div class="h-4 bg-gray-200 rounded animate-pulse opacity-70">
</div>
  <div class="h-4 bg-gray-200 rounded animate-pulse opacity-70">
</div>
  <div class="h-4 bg-gray-200 rounded animate-pulse opacity-70">
</div>
</div>

<!-- Disabled state -->
<button class="bg-blue-500 text-white px-4 py-2 rounded
disabled:opacity-50 disabled:cursor-not-allowed">
  Submit
</button>
```

Remember: You can also use opacity modifiers on colors:

```
<!-- These are different! -->
<div class="bg-blue-500 opacity-50">
  Entire element is 50% transparent
</div>

<div class="bg-blue-500/50">
  Only the background is 50% transparent (text stays opaque)
</div>
```

Blur Effects

Apply blur filters to elements:

```
<!-- Blur scale -->
<img src="photo.jpg" class="blur-none">No blur</img>
<img src="photo.jpg" class="blur-sm">Small blur (4px)</img>
<img src="photo.jpg" class="blur">Default blur (8px)</img>
<img src="photo.jpg" class="blur-md">Medium blur (12px)</img>
<img src="photo.jpg" class="blur-lg">Large blur (16px)</img>
<img src="photo.jpg" class="blur-xl">XL blur (24px)</img>
<img src="photo.jpg" class="blur-2xl">2XL blur (40px)</img>
<img src="photo.jpg" class="blur-3xl">3XL blur (64px)</img>
```

Backdrop Blur (Glass Effect)

Creates frosted glass effect - blurs content **behind** the element:

```
<div class="backdrop-blur-none">No blur</div>
<div class="backdrop-blur-sm">Small backdrop blur</div>
<div class="backdrop-blur">Default backdrop blur</div>
<div class="backdrop-blur-md">Medium backdrop blur</div>
<div class="backdrop-blur-lg">Large backdrop blur</div>
<div class="backdrop-blur-xl">XL backdrop blur</div>
<div class="backdrop-blur-2xl">2XL backdrop blur</div>
<div class="backdrop-blur-3xl">3XL backdrop blur</div>
```

Real-World Blur Examples

1. Glassmorphism card:

```
<div class="relative min-h-screen bg-gradient-to-br from-purple-
400 via-pink-500 to-red-500 p-8">

  <!-- Background content -->
  <div class="text-white text-9xl">BLUR</div>

  <!-- Glass card -->
```

```
  <div class="absolute top-1/2 left-1/2 transform -translate-x-
1/2 -translate-y-1/2">
    <div class="backdrop-blur-md bg-white/30 rounded-xl p-8
shadow-xl border border-white/20">
      <h2 class="text-3xl font-bold text-white mb-4">
        Glass Effect
      </h2>
      <p class="text-white/90">
        Background is blurred, creating a frosted glass
appearance
      </p>
    </div>
  </div>

</div>
```

2. Modal with blurred backdrop:

```
<div class="fixed inset-0 backdrop-blur-sm bg-black/30 flex
items-center justify-center">
  <div class="bg-white rounded-xl shadow-2xl p-8 max-w-md">
    <h2 class="text-2xl font-bold mb-4">Modal</h2>
    <p class="text-gray-600">Background is blurred when modal is
open</p>
  </div>
</div>
```

3. Navigation bar with glass effect:

```
<nav class="sticky top-0 backdrop-blur-lg bg-white/70 shadow-sm
z-10">
  <div class="max-w-7xl mx-auto px-4 py-4 flex justify-between
items-center">
    <h1 class="text-xl font-bold">Logo</h1>
```

```
    <div class="flex space-x-6">
      <a href="#" class="hover:text-blue-600">Home</a>
      <a href="#" class="hover:text-blue-600">About</a>
      <a href="#" class="hover:text-blue-600">Contact</a>
    </div>
  </div>
</nav>
```

4. Image with blur on hover:

```
<div class="relative overflow-hidden rounded-lg">
  <img
    src="photo.jpg"
    class="w-full transition-all duration-300 hover:blur-sm
hover:scale-110"
  >
  <div class="absolute inset-0 flex items-center justify-center
opacity-0 hover:opacity-100 transition-opacity">
    <button class="bg-white text-gray-900 px-6 py-3 rounded-lg
font-semibold">
      View Details
    </button>
  </div>
</div>
```

5. Spoiler text (blur until hover):

```
<p class="blur hover:blur-none transition-all cursor-pointer
select-none">
  This text is blurred until you hover over it. Great for
spoilers!
</p>
```

Ring Utilities

Rings are outline-style borders that don't affect layout (great for focus states).

Ring Width

```
<div class="ring">Default ring (3px)</div>
<div class="ring-0">No ring</div>
<div class="ring-1">1px ring</div>
<div class="ring-2">2px ring</div>
<div class="ring-4">4px ring</div>
<div class="ring-8">8px ring</div>

<!-- Inset ring (inside element) -->
<div class="ring-inset">Inset ring</div>
```

Ring Color

```
<div class="ring-2 ring-blue-500">Blue ring</div>
<div class="ring-2 ring-red-500">Red ring</div>
<div class="ring-2 ring-green-500">Green ring</div>

<!-- With opacity -->
<div class="ring-2 ring-blue-500/50">50% opacity blue ring</div>
```

Ring Offset

Creates space between the element and the ring:

```
<div class="ring-2 ring-blue-500 ring-offset-0">No offset</div>
<div class="ring-2 ring-blue-500 ring-offset-2">2px offset</div>
<div class="ring-2 ring-blue-500 ring-offset-4">4px offset</div>
<div class="ring-2 ring-blue-500 ring-offset-8">8px offset</div>
```

```
<!-- Offset color (for dark backgrounds) -->
<div class="bg-gray-800 ring-2 ring-white ring-offset-2 ring-offset-gray-800">
  Ring with colored offset
</div>
```

Real-World Ring Examples

1. Focus states:

```
<!-- Input focus -->
<input
  type="text"
  class="border border-gray-300 rounded px-4 py-2 focus:outline-none focus:ring-2 focus:ring-blue-500"
>

<!-- Button focus -->
<button class="bg-blue-500 text-white px-6 py-3 rounded focus:outline-none focus:ring-4 focus:ring-blue-300">
  Focusable Button
</button>
```

2. Selected state:

```
<div class="grid grid-cols-3 gap-4">
  <button class="border-2 border-gray-300 p-4 rounded-lg hover:border-blue-500">
    Option 1
  </button>

  <!-- Selected -->
  <button class="border-2 border-blue-500 p-4 rounded-lg ring-4 ring-blue-200">
```

```
    Option 2 (Selected)
  </button>

  <button class="border-2 border-gray-300 p-4 rounded-lg
hover:border-blue-500">
    Option 3
  </button>
</div>
```

3. Avatar with ring:

```
<img
  src="avatar.jpg"
  class="w-16 h-16 rounded-full ring-4 ring-blue-500 ring-offset-
2"
>
```

4. Status indicator:

```
<div class="relative inline-block">
  <img src="avatar.jpg" class="w-12 h-12 rounded-full">
  <span class="absolute bottom-0 right-0 block h-3 w-3 rounded-
full bg-green-500 ring-2 ring-white"></span>
</div>
```

Divide Utilities

Adds borders between child elements automatically (saves repeating border classes).

Divide Styles

```
<!-- Vertical dividers (for flex-row) -->
<div class="flex divide-x divide-gray-300">
  <div class="px-4">Item 1</div>
  <div class="px-4">Item 2</div>
  <div class="px-4">Item 3</div>
</div>

<!-- Horizontal dividers (for flex-col or stacked) -->
<div class="divide-y divide-gray-200">
  <div class="py-4">Item 1</div>
  <div class="py-4">Item 2</div>
  <div class="py-4">Item 3</div>
</div>

<!-- Different widths -->
<div class="divide-y-2 divide-gray-300">
  <div class="py-4">Thicker divider</div>
  <div class="py-4">Between items</div>
</div>

<!-- Dashed/dotted -->
<div class="divide-y divide-dashed divide-gray-300">
  <div class="py-4">Dashed</div>
  <div class="py-4">Dividers</div>
</div>
```

Comparison to manual borders:

```
<!-- ❌ Repetitive -->
<div>
  <div class="border-b border-gray-200 py-4">Item 1</div>
  <div class="border-b border-gray-200 py-4">Item 2</div>
  <div class="py-4">Item 3</div> <!-- Last one has no border -->
</div>
```

```
<!-- ✅ Clean with divide -->
<div class="divide-y divide-gray-200">
  <div class="py-4">Item 1</div>
  <div class="py-4">Item 2</div>
  <div class="py-4">Item 3</div>
</div>
```

Real-World Divide Examples

1. Navigation breadcrumbs:

```
<nav class="flex divide-x divide-gray-400 text-sm">
  <a href="#" class="px-3 hover:text-blue-600">Home</a>
  <a href="#" class="px-3 hover:text-blue-600">Products</a>
  <a href="#" class="px-3 hover:text-blue-600">Laptops</a>
  <span class="px-3 text-gray-500">MacBook Pro</span>
</nav>
```

2. List with dividers:

```
<ul class="divide-y divide-gray-200 border border-gray-200
rounded-lg">
  <li class="p-4 hover:bg-gray-50">
    <div class="font-medium">Item 1</div>
    <p class="text-sm text-gray-600">Description</p>
  </li>
  <li class="p-4 hover:bg-gray-50">
    <div class="font-medium">Item 2</div>
    <p class="text-sm text-gray-600">Description</p>
  </li>
  <li class="p-4 hover:bg-gray-50">
    <div class="font-medium">Item 3</div>
    <p class="text-sm text-gray-600">Description</p>
```

```
  </li>
</ul>
```

3. Stats row:

```
<div class="flex divide-x divide-gray-300 bg-white shadow
rounded-lg">
  <div class="flex-1 p-6 text-center">
    <div class="text-3xl font-bold text-gray-900">12,345</div>
    <div class="text-sm text-gray-500">Users</div>
  </div>
  <div class="flex-1 p-6 text-center">
    <div class="text-3xl font-bold text-gray-900">$45,678</div>
    <div class="text-sm text-gray-500">Revenue</div>
  </div>
  <div class="flex-1 p-6 text-center">
    <div class="text-3xl font-bold text-gray-900">789</div>
    <div class="text-sm text-gray-500">Orders</div>
  </div>
</div>
```

4. Timeline:

```
<div class="divide-y divide-gray-200">
  <div class="py-4 flex gap-4">
    <div class="text-sm text-gray-500">2:30 PM</div>
    <div class="flex-1">
      <div class="font-medium">Meeting started</div>
      <p class="text-sm text-gray-600">Team sync</p>
    </div>
  </div>
  <div class="py-4 flex gap-4">
    <div class="text-sm text-gray-500">3:15 PM</div>
    <div class="flex-1">
```

```
      <div class="font-medium">Document uploaded</div>
      <p class="text-sm text-gray-600">Q4 Report.pdf</p>
    </div>
  </div>
</div>
```

Combining Effects

The real power comes from combining multiple effects:

Elevated Card

```
<div class="bg-white rounded-xl shadow-lg hover:shadow-2xl border
border-gray-200 p-6 transition-all duration-300 hover:-translate-
y-1">
  <h3 class="text-xl font-bold mb-2">Interactive Card</h3>
  <p class="text-gray-600">Combines shadow, border, and
transform</p>
</div>
```

Glass Button

```
<button class="backdrop-blur-md bg-white/20 border border-
white/30 rounded-lg px-6 py-3 text-white shadow-xl hover:bg-
white/30 transition-all">
  Glass Button
</button>
```

Notification Badge

```
<div class="relative inline-block">
  <button class="bg-gray-800 text-white p-3 rounded-lg">
    <svg class="w-6 h-6"><!-- Bell icon --></svg>
  </button>

  <!-- Badge with multiple effects -->
  <span class="absolute -top-2 -right-2 bg-red-500 text-white
text-xs rounded-full h-6 w-6 flex items-center justify-center
ring-4 ring-white shadow-lg">
    3
  </span>
</div>
```

Complex Input

```
<div class="relative">
  <input
    type="text"
    class="w-full px-4 py-3 pl-12 border-2 border-gray-300
rounded-lg
           shadow-sm
           focus:border-blue-500 focus:ring-4 focus:ring-blue-200
focus:outline-none
           transition-all"
    placeholder="Search..."
  >
  <svg class="absolute left-4 top-3.5 w-5 h-5 text-gray-400">
    <!-- Search icon -->
  </svg>
</div>
```

Common Mistakes to Avoid

1. Overusing shadows

```
<!-- ❌ Shadow overload -->
<div class="shadow-2xl">
  <div class="shadow-xl">
    <button class="shadow-lg">Too much!</button>
  </div>
</div>

<!-- ✅ Subtle and hierarchical -->
<div class="shadow">
  <div class="p-4">
    <button class="shadow-sm hover:shadow
transition">Better</button>
  </div>
</div>
```

2. Forgetting transitions

```
<!-- ❌ Abrupt change -->
<div class="shadow hover:shadow-xl">
  No transition
</div>

<!-- ✅ Smooth -->
<div class="shadow hover:shadow-xl transition-shadow duration-
300">
  Smooth transition
</div>
```

3. Ring without focus

```
<!-- ❌ Always visible ring (weird) -->
<button class="ring-4 ring-blue-500">
  Why is there always a ring?
</button>

<!-- ✅ Ring on focus -->
<button class="focus:ring-4 focus:ring-blue-500 focus:outline-
none">
  Ring appears on focus
</button>
```

4. Blur performance

```
<!-- ❌ Blurring large areas (slow) -->
<div class="blur-3xl w-full h-screen">
  <!-- Performance issues -->
</div>

<!-- ✅ Blur smaller elements -->
<div class="backdrop-blur-sm">
  <!-- Much better performance -->
</div>
```

Key Takeaways

✅ **Borders**: Individual sides, widths, and styles
✅ **Border radius**: `rounded-lg` for corners, `rounded-full` for circles
✅ **Shadows**: Create elevation, use `shadow-lg` to `shadow-2xl`
✅ **Drop shadows**: Follow element shape (for transparent images)
✅ **Blur**: Regular blur vs `backdrop-blur` (glass effect)
✅ **Ring**: Perfect for focus states, doesn't affect layout

✅ **Divide**: Automatic borders between siblings
✅ Always add **transitions** when animating effects

Practice Assignments

Assignment 1 (Easy): Border Variations

Create 5 boxes demonstrating different border styles:

- Solid border with rounded corners
- Dashed border
- Border on specific sides only
- Border with accent color on one side
- Double border

Assignment 2 (Easy): Shadow Elevation

Create a card that demonstrates elevation on hover:

- Default state: subtle shadow
- Hover: larger shadow + slight lift (transform)
- Include smooth transition
- Add a button inside the card

Assignment 3 (Medium): Glass Card

Create a glassmorphism card:

- Colorful gradient background
- Card with `backdrop-blur` effect
- Semi-transparent white background
- Border with opacity

- Shadow
- Content that's readable

Assignment 4 (Medium): Form with Effects

Build a form with:

- Inputs with border and focus ring
- Custom styled checkboxes with rings
- Buttons with shadows that change on hover
- Divide utilities between form sections
- Proper focus states throughout

Assignment 5 (Hard): Product Card Gallery

Create a grid of product cards with:

- Image that zooms on hover
- Overlay that appears with backdrop blur
- Shadow that intensifies on hover
- Ring around selected card
- Status badge with ring
- Smooth transitions on all effects
- Responsive grid (1, 2, 3 columns)

Bonus: Add a "glassmorphic" navigation bar that blurs content behind it as you scroll.

Ready for **Module 9: Customization**? Let me know!

1/7/2026, 1:26:21 PM

MODULE 9: CUSTOMIZATION

tailwind.config.js Deep Dive

The `tailwind.config.js` file is where you customize Tailwind to match your design system. Let's explore its structure and capabilities.

Basic Structure

```
/** @type {import('tailwindcss').Config} */
export default {
  content: [
    "./index.html",
    "./src/**/*.{js,ts,jsx,tsx}",
  ],
  theme: {
    extend: {
      // Your customizations here
    },
  },
  plugins: [],
}
```

Key sections:

1. `content` - Where to look for class names (for purging unused CSS)
2. `theme` - Design system configuration (colors, spacing, fonts, etc.)
3. `plugins` - Add third-party plugins

Content Configuration

The `content` array tells Tailwind which files to scan for class names:

```
export default {
  content: [
    "./index.html",
    "./src/**/*.{js,jsx,ts,tsx}",      // React/TypeScript files
    "./components/**/*.{vue,js}",       // Vue components
    "./pages/**/*.{html,js}",           // Static pages
    "./app/**/*.{js,ts,jsx,tsx}",       // Next.js app directory
    "./*.{html,js}",                    // Root level files
  ],
  // ...
}
```

Important: Only classes found in these files will be included in your final CSS!

Safelist (Forcing Classes to Stay)

Sometimes you generate classes dynamically and need to prevent them from being purged:

```
export default {
  content: ["./src/**/*.{html,js}"],
  safelist: [
    'bg-red-500',
    'bg-green-500',
    'bg-blue-500',
    // Pattern matching
    {
      pattern: /bg-(red|green|blue)-(100|200|300)/,
    },
  ],
  // ...
}
```

Use case: When class names are constructed dynamically:

```
// ❌ This won't work reliably (Tailwind can't detect these)
const color = 'blue';
const className = `bg-${color}-500`; // May be purged!

// ✅ Better: Use complete class names
const className = color === 'blue' ? 'bg-blue-500' : 'bg-red-
500';

// ✅ Or add to safelist
```

Extending the Theme

The `extend` key lets you **add to** Tailwind's defaults without overwriting them.

Extend vs Replace

```
export default {
  theme: {
    // ❌ This REPLACES all default colors
    colors: {
      blue: '#1e40af',
    },

    // ✅ This ADDS to default colors
    extend: {
      colors: {
        brand: '#1e40af',
      },
    },
  },
}
```

Rule of thumb: Almost always use `extend` unless you want to replace Tailwind's entire design system.

Custom Colors, Spacing, Fonts

Custom Colors

Add brand colors while keeping Tailwind's defaults:

```
export default {
  theme: {
    extend: {
      colors: {
        // Single color
        brand: '#5B21B6',

        // Color with shades
        primary: {
          50: '#f5f3ff',
          100: '#ede9fe',
          200: '#ddd6fe',
          300: '#c4b5fd',
          400: '#a78bfa',
          500: '#8b5cf6',  // Base
          600: '#7c3aed',
          700: '#6d28d9',
          800: '#5b21b6',
          900: '#4c1d95',
        },

        // Semantic colors
        success: '#10b981',
        warning: '#f59e0b',
```

```
        danger: '#ef4444',

        // Using CSS variables (dynamic theming)
        background: 'var(--color-background)',
        foreground: 'var(--color-foreground)',
      },
    },
  },
}
```

Usage:

```
<div class="bg-brand text-white">Brand color</div>
<div class="bg-primary-500 hover:bg-primary-600">Primary
button</div>
<div class="text-success">Success message</div>
```

Custom Spacing

Add custom spacing values:

```
export default {
  theme: {
    extend: {
      spacing: {
        '72': '18rem',     // 288px
        '84': '21rem',     // 336px
        '96': '24rem',     // 384px
        '128': '32rem',    // 512px
        '144': '36rem',    // 576px

        // Fractional values
        '1/10': '10%',
        '2/10': '20%',
```

```
        // Custom named sizes
        'card': '320px',
        'sidebar': '280px',
      },
    },
  },
}
```

Usage:

```
<div class="w-sidebar">Custom sidebar width</div>
<div class="h-128">Extra tall element</div>
<div class="p-72">Large padding</div>
```

Custom Fonts

Add custom font families:

```
export default {
  theme: {
    extend: {
      fontFamily: {
        // Using system fonts
        sans: ['Inter', 'system-ui', 'sans-serif'],
        serif: ['Georgia', 'serif'],
        mono: ['Fira Code', 'monospace'],

        // Custom fonts
        display: ['Playfair Display', 'serif'],
        body: ['Open Sans', 'sans-serif'],

        // Multiple fallbacks
        heading: ['Montserrat', 'Helvetica', 'Arial', 'sans-
```

```
serif'],
      },
    },
  },
}
```

Usage:

```
<h1 class="font-display text-5xl">Heading in Playfair</h1>
<p class="font-body">Body text in Open Sans</p>
<code class="font-mono">Code in Fira Code</code>
```

Custom Font Sizes

Add custom text sizes:

```
export default {
  theme: {
    extend: {
      fontSize: {
        'xxs': '0.625rem',      // 10px
        'huge': '10rem',         // 160px

        // With line-height
        'display': ['6rem', { lineHeight: '1' }],

        // With line-height and letter-spacing
        'hero': ['8rem', {
          lineHeight: '1',
          letterSpacing: '-0.02em',
          fontWeight: '700',
        }],
      },
    },
```

```
  },
}
```

Usage:

```
<p class="text-xxs">Extra extra small</p>
<h1 class="text-display">Display heading</h1>
<h1 class="text-hero">Hero text</h1>
```

Custom Breakpoints

Modify or add breakpoints:

```
export default {
  theme: {
    extend: {
      screens: {
        'xs': '475px',      // New extra-small breakpoint
        '3xl': '1920px',    // New extra-large breakpoint

        // Custom breakpoint
        'tablet': '640px',
        'laptop': '1024px',
        'desktop': '1280px',

        // Max-width breakpoint (mobile-last)
        'max-md': {'max': '767px'},

        // Range breakpoint
        'md-only': {'min': '768px', 'max': '1023px'},
      },
    },
  },
}
```

Usage:

```
<div class="hidden xs:block">Visible from 475px</div>
<div class="text-sm tablet:text-base laptop:text-lg">Custom
breakpoints</div>
<div class="block max-md:hidden">Hidden on mobile</div>
```

Custom Container

Customize the container utility:

```
export default {
  theme: {
    container: {
      center: true,        // Center by default
      padding: {
        DEFAULT: '1rem',
        sm: '2rem',
        lg: '4rem',
        xl: '5rem',
        '2xl': '6rem',
      },
      screens: {
        sm: '640px',
        md: '768px',
        lg: '1024px',
        xl: '1280px',
        '2xl': '1400px',  // Custom max-width
      },
    },
  },
}
```

Usage:

```
<!-- No need for mx-auto anymore -->
<div class="container">
  Automatically centered with responsive padding
</div>
```

Adding Custom Utilities

Sometimes you need utilities that don't exist in Tailwind.

Using the `@layer` Directive

Add custom utilities in your CSS file:

```
@tailwind base;
@tailwind components;
@tailwind utilities;

@layer utilities {
  /* Custom text utilities */
  .text-shadow {
    text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.3);
  }

  .text-shadow-lg {
    text-shadow: 4px 4px 8px rgba(0, 0, 0, 0.4);
  }

  /* Custom scrollbar */
  .scrollbar-hide {
    -ms-overflow-style: none;
    scrollbar-width: none;
  }
```

```
  .scrollbar-hide::-webkit-scrollbar {
    display: none;
  }

  /* Custom gradient text */
  .gradient-text {
    background: linear-gradient(to right, #3b82f6, #8b5cf6);
    -webkit-background-clip: text;
    -webkit-text-fill-color: transparent;
    background-clip: text;
  }

  /* Custom aspect ratios */
  .aspect-video-vertical {
    aspect-ratio: 9 / 16;
  }
}
```

Usage:

```
<h1 class="text-5xl font-bold text-shadow-lg">
  Text with shadow
</h1>

<div class="overflow-auto scrollbar-hide h-64">
  Scrollable without visible scrollbar
</div>

<h2 class="text-6xl font-bold gradient-text">
  Gradient Text
</h2>
```

Adding Utilities via Config

You can also add utilities in `tailwind.config.js`:

```
import plugin from 'tailwindcss/plugin'

export default {
  theme: {
    extend: {},
  },
  plugins: [
    plugin(function({ addUtilities }) {
      const newUtilities = {
        '.text-shadow-sm': {
          textShadow: '1px 1px 2px rgba(0, 0, 0, 0.3)',
        },
        '.text-shadow': {
          textShadow: '2px 2px 4px rgba(0, 0, 0, 0.3)',
        },
        '.text-shadow-lg': {
          textShadow: '4px 4px 8px rgba(0, 0, 0, 0.4)',
        },
        '.rotate-y-180': {
          transform: 'rotateY(180deg)',
        },
        '.backface-hidden': {
          backfaceVisibility: 'hidden',
        },
      }

      addUtilities(newUtilities)
    })
  ],
}
```

Responsive and State Variants

Make your custom utilities work with responsive and state variants:

```
plugins: [
  plugin(function({ addUtilities }) {
    const newUtilities = {
      '.text-stroke': {
        '-webkit-text-stroke': '1px black',
      },
    }

    addUtilities(newUtilities, {
      variants: ['responsive', 'hover'], // Enable responsive and hover
    })
  })
]
```

Usage:

```
<h1 class="text-stroke hover:text-stroke-none md:text-stroke-2">
  Responsive text stroke
</h1>
```

Using CSS Variables with Tailwind

CSS variables (custom properties) enable dynamic theming and work great with Tailwind.

Setup in CSS

```
@tailwind base;
@tailwind components;
@tailwind utilities;

@layer base {
  :root {
    --color-primary: 59 130 246;      /* RGB values (space-separated) */
    --color-secondary: 139 92 246;
    --color-accent: 236 72 153;

    --spacing-section: 4rem;
    --border-radius: 0.5rem;
  }

  .dark {
    --color-primary: 96 165 250;     /* Lighter blue for dark mode */
    --color-secondary: 167 139 250;
    --color-accent: 244 114 182;
  }
}
```

Why RGB format? So you can use opacity modifiers:

Configure in Tailwind

```
export default {
  theme: {
    extend: {
      colors: {
        primary: 'rgb(var(--color-primary) / <alpha-value>)',
        secondary: 'rgb(var(--color-secondary) / <alpha-value>)',
        accent: 'rgb(var(--color-accent) / <alpha-value>)',
```

```
      },
      spacing: {
        'section': 'var(--spacing-section)',
      },
      borderRadius: {
        'custom': 'var(--border-radius)',
      },
    },
  },
}
```

Usage with opacity:

```
<div class="bg-primary">Solid primary color</div>
<div class="bg-primary/50">50% transparent primary</div>
<div class="bg-primary/20">20% transparent primary</div>

<div class="py-section">Dynamic section padding</div>
```

Dynamic Theming Example

```
<!DOCTYPE html>
<html>
<head>
  <style>
    :root {
      --color-theme: 59 130 246; /* Blue */
    }

    [data-theme="purple"] {
      --color-theme: 139 92 246;
    }

    [data-theme="pink"] {
```

```
      --color-theme: 236 72 153;
    }
  </style>
</head>
<body>
  <div class="p-8">
    <!-- Theme switcher -->
    <div class="flex space-x-4 mb-8">
      <button onclick="setTheme('')" class="px-4 py-2 bg-blue-500
text-white rounded">
        Blue
      </button>
      <button onclick="setTheme('purple')" class="px-4 py-2 bg-
purple-500 text-white rounded">
        Purple
      </button>
      <button onclick="setTheme('pink')" class="px-4 py-2 bg-
pink-500 text-white rounded">
        Pink
      </button>
    </div>

    <!-- Content that uses the theme color -->
    <div class="bg-theme text-white p-8 rounded-lg">
      <h1 class="text-3xl font-bold mb-4">Dynamic Theme</h1>
      <p>Click the buttons to change the theme color</p>
      <button class="mt-4 bg-white text-theme px-4 py-2 rounded">
        Themed Button
      </button>
    </div>
  </div>

  <script>
    function setTheme(theme) {
      document.documentElement.setAttribute('data-theme', theme);
```

```
    }
  </script>
</body>
</html>
```

Advanced: Theming System

Create a complete theming system:

```
@layer base {
  :root {
    /* Light theme (default) */
    --background: 255 255 255;
    --foreground: 17 24 39;
    --card: 255 255 255;
    --card-foreground: 17 24 39;
    --primary: 59 130 246;
    --primary-foreground: 255 255 255;
    --secondary: 243 244 246;
    --secondary-foreground: 17 24 39;
    --muted: 243 244 246;
    --muted-foreground: 107 114 128;
    --border: 229 231 235;
    --radius: 0.5rem;
  }

  .dark {
    /* Dark theme */
    --background: 17 24 39;
    --foreground: 243 244 246;
    --card: 31 41 55;
    --card-foreground: 243 244 246;
    --primary: 96 165 250;
    --primary-foreground: 17 24 39;
    --secondary: 55 65 81;
```

```
    --secondary-foreground: 243 244 246;
    --muted: 55 65 81;
    --muted-foreground: 156 163 175;
    --border: 55 65 81;
  }
}
```

```
// tailwind.config.js
export default {
  theme: {
    extend: {
      colors: {
        background: 'rgb(var(--background) / <alpha-value>)',
        foreground: 'rgb(var(--foreground) / <alpha-value>)',
        card: {
          DEFAULT: 'rgb(var(--card) / <alpha-value>)',
          foreground: 'rgb(var(--card-foreground) / <alpha-
value>)',
        },
        primary: {
          DEFAULT: 'rgb(var(--primary) / <alpha-value>)',
          foreground: 'rgb(var(--primary-foreground) / <alpha-
value>)',
        },
        secondary: {
          DEFAULT: 'rgb(var(--secondary) / <alpha-value>)',
          foreground: 'rgb(var(--secondary-foreground) / <alpha-
value>)',
        },
        muted: {
          DEFAULT: 'rgb(var(--muted) / <alpha-value>)',
          foreground: 'rgb(var(--muted-foreground) / <alpha-
value>)',
        },
        border: 'rgb(var(--border) / <alpha-value>)',
```

```
      },
      borderRadius: {
        lg: 'var(--radius)',
        md: 'calc(var(--radius) - 2px)',
        sm: 'calc(var(--radius) - 4px)',
      },
    },
  },
}
```

Usage:

```
<div class="bg-background text-foreground">
  <div class="bg-card text-card-foreground border border-border
rounded-lg p-6">
    <h2 class="text-xl font-bold mb-4">Card</h2>
    <p class="text-muted-foreground mb-4">Some muted text</p>
    <button class="bg-primary text-primary-foreground px-4 py-2
rounded-lg">
      Primary Button
    </button>
  </div>
</div>
```

Plugins (forms, typography, etc.)

Tailwind has official and community plugins that extend functionality.

Official Plugins

1. Typography Plugin

We covered this briefly in Module 4. It styles prose content:

```
npm install -D @tailwindcss/typography
```

```
// tailwind.config.js
export default {
  plugins: [
    require('@tailwindcss/typography'),
  ],
}
```

```
<article class="prose lg:prose-xl">
  <h1>This gets styled automatically</h1>
  <p>No need for individual classes</p>
</article>
```

2. Forms Plugin

Provides better default styles for form elements:

```
npm install -D @tailwindcss/forms
```

```
export default {
  plugins: [
    require('@tailwindcss/forms'),
  ],
}
```

Before forms plugin:

```
<!-- Ugly, unstyled inputs -->
<input type="text">
<select></select>
<textarea></textarea>
```

After forms plugin:

```
<!-- Beautiful, consistent styling -->
<input type="text" class="rounded-lg">
<select class="rounded-lg"></select>
<textarea class="rounded-lg"></textarea>
```

3. Aspect Ratio Plugin (now built-in)

Maintains aspect ratios:

```
<div class="aspect-video">16:9 video container</div>
<div class="aspect-square">1:1 square container</div>
<div class="aspect-[4/3]">4:3 aspect ratio</div>
```

4. Container Queries Plugin

Responsive design based on container size, not viewport:

```
npm install -D @tailwindcss/container-queries
```

```
export default {
  plugins: [
    require('@tailwindcss/container-queries'),
  ],
}
```

```
<div class="@container">
  <div class="@lg:grid @lg:grid-cols-2">
    <!-- Layout changes based on container size, not viewport -->
  </div>
</div>
```

Custom Plugin Example

Create your own plugin:

```
// tailwind.config.js
import plugin from 'tailwindcss/plugin'

export default {
  plugins: [
    // Custom animations plugin
    plugin(function({ addUtilities, theme }) {
      const animations = {
        '.animate-fade-in': {
          animation: 'fadeIn 0.5s ease-in',
        },
        '.animate-slide-up': {
          animation: 'slideUp 0.5s ease-out',
        },
      }

      addUtilities(animations)
    }),

    // Add keyframes
    plugin(function({ addBase }) {
      addBase({
        '@keyframes fadeIn': {
          '0%': { opacity: '0' },
          '100%': { opacity: '1' },
        },
        '@keyframes slideUp': {
          '0%': {
            transform: 'translateY(20px)',
            opacity: '0',
          },
          '100%': {
            transform: 'translateY(0)',
```

```
            opacity: '1',
          },
        },
      })
    }),
  ],
}
```

Usage:

```
<div class="animate-fade-in">Fades in</div>
<div class="animate-slide-up">Slides up</div>
```

Complete Config Example

Here's a comprehensive config showcasing various customizations:

```
/** @type {import('tailwindcss').Config} */
import plugin from 'tailwindcss/plugin'

export default {
  content: [
    "./index.html",
    "./src/**/*.{js,ts,jsx,tsx}",
  ],

  darkMode: 'class',

  theme: {
    // Global container config
    container: {
```

```
      center: true,
      padding: {
        DEFAULT: '1rem',
        sm: '2rem',
        lg: '4rem',
        xl: '5rem',
      },
    },

    extend: {
      // Custom colors
      colors: {
        brand: {
          50: '#f0f9ff',
          100: '#e0f2fe',
          200: '#bae6fd',
          300: '#7dd3fc',
          400: '#38bdf8',
          500: '#0ea5e9',
          600: '#0284c7',
          700: '#0369a1',
          800: '#075985',
          900: '#0c4a6e',
        },
        primary: 'rgb(var(--color-primary) / <alpha-value>)',
        secondary: 'rgb(var(--color-secondary) / <alpha-value>)',
      },

      // Custom fonts
      fontFamily: {
        sans: ['Inter', 'system-ui', 'sans-serif'],
        display: ['Playfair Display', 'serif'],
        mono: ['Fira Code', 'monospace'],
      },
```

```
      // Custom font sizes
      fontSize: {
        'xxs': '0.625rem',
        'hero': ['4rem', { lineHeight: '1', letterSpacing:
'-0.02em' }],
      },

      // Custom spacing
      spacing: {
        '128': '32rem',
        '144': '36rem',
        'sidebar': '280px',
      },

      // Custom breakpoints
      screens: {
        'xs': '475px',
        '3xl': '1920px',
      },

      // Custom animations
      animation: {
        'fade-in': 'fadeIn 0.5s ease-in',
        'slide-up': 'slideUp 0.5s ease-out',
        'slide-down': 'slideDown 0.5s ease-out',
      },

      keyframes: {
        fadeIn: {
          '0%': { opacity: '0' },
          '100%': { opacity: '1' },
        },
        slideUp: {
          '0%': { transform: 'translateY(20px)', opacity: '0' },
          '100%': { transform: 'translateY(0)', opacity: '1' },
```

```
        },
        slideDown: {
          '0%': { transform: 'translateY(-20px)', opacity: '0' },
          '100%': { transform: 'translateY(0)', opacity: '1' },
        },
      },

      // Custom border radius
      borderRadius: {
        '4xl': '2rem',
      },

      // Custom shadows
      boxShadow: {
        'glow': '0 0 20px rgba(59, 130, 246, 0.5)',
      },
    },
  },

  plugins: [
    require('@tailwindcss/typography'),
    require('@tailwindcss/forms'),

    // Custom utilities plugin
    plugin(function({ addUtilities }) {
      const newUtilities = {
        '.text-shadow': {
          textShadow: '2px 2px 4px rgba(0, 0, 0, 0.3)',
        },
        '.scrollbar-hide': {
          '-ms-overflow-style': 'none',
          'scrollbar-width': 'none',
          '&::-webkit-scrollbar': {
            display: 'none',
          },
```

```
        },
      }

      addUtilities(newUtilities)
    }),
  ],
}
```

Common Mistakes to Avoid

1. Overriding instead of extending

```
// ❌ Removes all default colors
theme: {
  colors: {
    brand: '#5B21B6',
  },
}

// ✅ Keeps defaults, adds brand
theme: {
  extend: {
    colors: {
      brand: '#5B21B6',
    },
  },
}
```

2. Not using proper content paths

```
// ❌ Too specific (misses files)
content: ["./src/App.jsx"]

// ✅ Catches all files
content: ["./src/**/*.{js,jsx,ts,tsx}"]
```

3. Generating classes dynamically

```
// ❌ Tailwind can't see this
const size = 'lg';
<div className={`text-${size}`}>

// ✅ Use complete class names
<div className={size === 'lg' ? 'text-lg' : 'text-base'}>
```

4. Not restarting build after config changes

Config changes require restarting your build process!

Key Takeaways

✅ Use `extend` to **add to** defaults, not replace them
✅ `content` array must include all files with Tailwind classes
✅ CSS variables enable **dynamic theming**
✅ Create **custom utilities** for repeated patterns
✅ Use **plugins** for complex functionality (typography, forms)
✅ **Safelist** dynamically generated classes
✅ Customize **breakpoints, colors, spacing, fonts** to match your design
✅ **Restart build** after config changes

Practice Assignments

Assignment 1 (Easy): Custom Colors

1. Add 3 brand colors to your config (primary, secondary, accent)
2. Create a color palette with shades (50-900) for your primary color
3. Use them in a card component
4. Test opacity modifiers (`bg-primary/50`)

Assignment 2 (Easy): Custom Spacing

1. Add custom spacing values (e.g., `sidebar: '280px'`, `hero: '32rem'`)
2. Add custom breakpoints (e.g., `xs: '475px'`)
3. Create a layout using your custom values
4. Make it responsive with your custom breakpoints

Assignment 3 (Medium): CSS Variables Theme

1. Set up CSS variables for colors in `:root` and `.dark`
2. Configure Tailwind to use these variables
3. Create a theme switcher (light/dark)
4. Build a card that adapts to both themes
5. Ensure opacity modifiers work

Assignment 4 (Medium): Custom Plugin

Create a plugin that adds:

1. Text shadow utilities (`.text-shadow-sm`, `.text-shadow-md`, `.text-shadow-lg`)
2. Custom animation (e.g., fade-in with slide)
3. Scrollbar hide utility
4. Use all three in a demo page

Assignment 5 (Hard): Complete Design System

Build a complete design system config:

1. Brand color palette (primary, secondary, accent with shades)
2. Custom fonts (heading, body, mono)
3. Custom spacing scale
4. Custom breakpoints
5. CSS variables for dynamic theming
6. Custom utilities (at least 3)
7. Typography plugin configured
8. Build a multi-page website using ONLY your custom design system

Deliverables:

- `tailwind.config.js` with all customizations
- CSS file with variables and custom utilities
- 3 pages demonstrating your design system
- Theme switcher
- Documentation of your design tokens

Ready for **Module 10: Components & Patterns**? Let me know!

1/6/2026, 1:10:46 PM

MODULE 10: COMPONENTS & PATTERNS

Building Reusable Components

When using Tailwind, you'll often repeat the same class combinations. Let's explore strategies for creating reusable components without losing Tailwind's utility-first benefits.

Component Approaches

Three main strategies:

1. **HTML/Template components** (copy-paste patterns)
2. **JavaScript components** (React, Vue, etc.)
3. **CSS components** (using `@apply` directive - use sparingly!)

Button Variations

Let's build a comprehensive button system.

Base Button Pattern

```
<!-- Primary button -->
<button class="px-6 py-3 bg-blue-600 hover:bg-blue-700 active:bg-blue-800 text-white font-semibold rounded-lg shadow-md hover:shadow-lg transform active:scale-95 transition-all duration-150 focus:outline-none focus:ring-4 focus:ring-blue-300">
  Primary Button
</button>

<!-- Secondary button -->
<button class="px-6 py-3 bg-gray-200 hover:bg-gray-300 active:bg-gray-400 text-gray-900 font-semibold rounded-lg shadow-md hover:shadow-lg transform active:scale-95 transition-all duration-150 focus:outline-none focus:ring-4 focus:ring-gray-300">
```

```
  Secondary Button
</button>

<!-- Outline button -->
<button class="px-6 py-3 border-2 border-blue-600 hover:bg-blue-
50 active:bg-blue-100 text-blue-600 font-semibold rounded-lg
transition-colors duration-150 focus:outline-none focus:ring-4
focus:ring-blue-300">
  Outline Button
</button>

<!-- Ghost button -->
<button class="px-6 py-3 hover:bg-gray-100 active:bg-gray-200
text-gray-700 font-semibold rounded-lg transition-colors
duration-150 focus:outline-none focus:ring-4 focus:ring-gray-
200">
  Ghost Button
</button>
```

Button Sizes

```
<!-- Extra small -->
<button class="px-3 py-1.5 text-sm bg-blue-600 hover:bg-blue-700
text-white font-medium rounded transition">
  Extra Small
</button>

<!-- Small -->
<button class="px-4 py-2 text-sm bg-blue-600 hover:bg-blue-700
text-white font-semibold rounded-lg transition">
  Small
</button>

<!-- Medium (default) -->
```

```
<button class="px-6 py-3 bg-blue-600 hover:bg-blue-700 text-white font-semibold rounded-lg transition">
  Medium
</button>

<!-- Large -->
<button class="px-8 py-4 text-lg bg-blue-600 hover:bg-blue-700 text-white font-semibold rounded-lg transition">
  Large
</button>

<!-- Extra large -->
<button class="px-10 py-5 text-xl bg-blue-600 hover:bg-blue-700 text-white font-bold rounded-xl transition">
  Extra Large
</button>
```

Button States

```
<!-- Loading state -->
<button class="px-6 py-3 bg-blue-600 text-white font-semibold rounded-lg flex items-center space-x-2" disabled>
  <svg class="animate-spin h-5 w-5" viewBox="0 0 24 24">
    <circle class="opacity-25" cx="12" cy="12" r="10" stroke="currentColor" stroke-width="4" fill="none"></circle>
    <path class="opacity-75" fill="currentColor" d="M4 12a8 8 0 018-8V0C5.373 0 0 5.373 0 12h4zm2 5.291A7.962 7.962 0 014 12H0c0 3.042 1.135 5.824 3 7.938l3-2.647z"></path>
  </svg>
  <span>Loading...</span>
</button>

<!-- Disabled state -->
<button class="px-6 py-3 bg-blue-600 text-white font-semibold
```

```
rounded-lg opacity-50 cursor-not-allowed" disabled>
  Disabled
</button>

<!-- With icon -->
<button class="px-6 py-3 bg-blue-600 hover:bg-blue-700 text-white
font-semibold rounded-lg flex items-center space-x-2 transition">
  <svg class="w-5 h-5" fill="none" stroke="currentColor"
viewBox="0 0 24 24">
    <path stroke-linecap="round" stroke-linejoin="round" stroke-
width="2" d="M12 4v16m8-8H4"></path>
  </svg>
  <span>Add Item</span>
</button>

<!-- Icon only -->
<button class="p-3 bg-blue-600 hover:bg-blue-700 text-white
rounded-lg transition">
  <svg class="w-5 h-5" fill="none" stroke="currentColor"
viewBox="0 0 24 24">
    <path stroke-linecap="round" stroke-linejoin="round" stroke-
width="2" d="M6 18L18 6M6 6l12 12"></path>
  </svg>
</button>
```

React Button Component Example

```
// Button.jsx
export function Button({
  children,
  variant = 'primary',
  size = 'md',
  disabled = false,
  loading = false,
```

```
  onClick,
  className = '',
}) {
  const baseStyles = "font-semibold rounded-lg transition-all duration-150 focus:outline-none focus:ring-4 inline-flex items-center justify-center"

  const variants = {
    primary: "bg-blue-600 hover:bg-blue-700 active:bg-blue-800 text-white focus:ring-blue-300",
    secondary: "bg-gray-200 hover:bg-gray-300 active:bg-gray-400 text-gray-900 focus:ring-gray-300",
    outline: "border-2 border-blue-600 hover:bg-blue-50 text-blue-600 focus:ring-blue-300",
    ghost: "hover:bg-gray-100 active:bg-gray-200 text-gray-700 focus:ring-gray-200",
    danger: "bg-red-600 hover:bg-red-700 active:bg-red-800 text-white focus:ring-red-300",
  }

  const sizes = {
    sm: "px-4 py-2 text-sm",
    md: "px-6 py-3",
    lg: "px-8 py-4 text-lg",
  }

  const classes = `
    ${baseStyles}
    ${variants[variant]}
    ${sizes[size]}
    ${disabled || loading ? 'opacity-50 cursor-not-allowed' : 'active:scale-95'}
    ${className}
  `.trim()
```

```
  return (
    <button
      className={classes}
      disabled={disabled || loading}
      onClick={onClick}
    >
      {loading && (
        <svg className="animate-spin -ml-1 mr-2 h-5 w-5"
viewBox="0 0 24 24">
          <circle className="opacity-25" cx="12" cy="12" r="10"
stroke="currentColor" strokeWidth="4" fill="none"></circle>
          <path className="opacity-75" fill="currentColor" d="M4
12a8 8 0 018-8V0C5.373 0 0 5.373 0 12h4zm2 5.291A7.962 7.962 0
014 12H0c0 3.042 1.135 5.824 3 7.938l3-2.647z"></path>
        </svg>
      )}
      {children}
    </button>
  )
}
```

Usage:

```
<Button variant="primary" size="md">Click Me</Button>
<Button variant="secondary" size="lg">Secondary</Button>
<Button variant="outline" loading>Loading...</Button>
<Button variant="danger" disabled>Disabled</Button>
```

Card Components

Cards are one of the most common UI patterns.

Basic Card

```
<div class="bg-white rounded-lg shadow-md overflow-hidden">
  <!-- Image -->
  <img src="card-image.jpg" class="w-full h-48 object-cover">

  <!-- Content -->
  <div class="p-6">
    <h3 class="text-xl font-bold text-gray-900 mb-2">
      Card Title
    </h3>
    <p class="text-gray-600 mb-4">
      This is a description of the card content. It provides
context and information.
    </p>
    <button class="bg-blue-600 hover:bg-blue-700 text-white px-4
py-2 rounded transition">
      Learn More
    </button>
  </div>
</div>
```

Card Variations

1. Horizontal Card:

```
<div class="bg-white rounded-lg shadow-md overflow-hidden flex
flex-col md:flex-row">
  <!-- Image -->
  <img src="card-image.jpg" class="w-full md:w-64 h-48 md:h-auto
object-cover">

  <!-- Content -->
  <div class="p-6 flex-1">
    <h3 class="text-xl font-bold mb-2">Horizontal Card</h3>
```

```
    <p class="text-gray-600 mb-4">Content side by side with image
on larger screens.</p>
    <button class="bg-blue-600 hover:bg-blue-700 text-white px-4
py-2 rounded transition">
      Action
    </button>
  </div>
</div>
```

2. Card with Header & Footer:

```
<div class="bg-white rounded-lg shadow-md overflow-hidden">
  <!-- Header -->
  <div class="bg-gray-50 border-b border-gray-200 px-6 py-4">
    <h3 class="text-lg font-semibold text-gray-900">Card
Header</h3>
  </div>

  <!-- Body -->
  <div class="p-6">
    <p class="text-gray-600">
      Main content goes here. This is the body of the card.
    </p>
  </div>

  <!-- Footer -->
  <div class="bg-gray-50 border-t border-gray-200 px-6 py-4 flex
justify-between items-center">
    <span class="text-sm text-gray-500">Last updated: 2 hours
ago</span>
    <button class="text-blue-600 hover:text-blue-800 text-sm
font-medium">
      View Details →
    </button>
```

```
  </div>
</div>
```

3. Hoverable Card:

```
<div class="bg-white rounded-lg shadow-md hover:shadow-xl
transition-shadow duration-300 overflow-hidden cursor-pointer
group">
  <!-- Image with zoom effect -->
  <div class="overflow-hidden">
    <img src="card-image.jpg" class="w-full h-48 object-cover
transform group-hover:scale-110 transition-transform duration-
500">
  </div>

  <!-- Content -->
  <div class="p-6">
    <h3 class="text-xl font-bold text-gray-900 group-hover:text-
blue-600 transition-colors mb-2">
      Interactive Card
    </h3>
    <p class="text-gray-600 mb-4">
      Entire card is clickable with hover effects.
    </p>
    <div class="flex items-center text-blue-600 group-hover:text-
blue-800 transition-colors">
      <span class="text-sm font-medium">Read More</span>
      <svg class="w-4 h-4 ml-2 transform group-hover:translate-x-
2 transition-transform" fill="none" stroke="currentColor"
viewBox="0 0 24 24">
        <path stroke-linecap="round" stroke-linejoin="round"
stroke-width="2" d="M9 5l7 7-7 7"></path>
      </svg>
    </div>
```

```
  </div>
</div>
```

4. Stat Card:

```
<div class="bg-white rounded-lg shadow-md p-6">
  <div class="flex items-center justify-between mb-4">
    <h3 class="text-sm font-medium text-gray-500">Total
Revenue</h3>
    <svg class="w-8 h-8 text-green-500" fill="none"
stroke="currentColor" viewBox="0 0 24 24">
      <path stroke-linecap="round" stroke-linejoin="round"
stroke-width="2" d="M12 8c-1.657 0-3 .895-3 2s1.343 2 3 2 3 .895
3 2-1.343 2-3 2m0-8c1.11 0 2.08.402 2.599 1M12 8V7m0 1v8m0 0v1m0-
1c-1.11 0-2.08-.402-2.599-1M21 12a9 9 0 11-18 0 9 9 0 0118 0z">
</path>
    </svg>
  </div>
  <div class="text-3xl font-bold text-gray-900 mb-
2">$45,678</div>
  <div class="flex items-center text-sm">
    <span class="text-green-600 font-medium flex items-center">
      <svg class="w-4 h-4 mr-1" fill="currentColor" viewBox="0 0
20 20">
        <path fill-rule="evenodd" d="M5.293 9.707a1 1 0 010-
1.414l4-4a1 1 0 011.414 0l4 4a1 1 0 01-1.414 1.414L11 7.414V15a1
1 0 11-2 0V7.414L6.707 9.707a1 1 0 01-1.414 0z" clip-
rule="evenodd"></path>
      </svg>
      12.5%
    </span>
    <span class="text-gray-500 ml-2">from last month</span>
  </div>
</div>
```

5. Profile Card:

```
<div class="bg-white rounded-lg shadow-md overflow-hidden">
  <!-- Cover image -->
  <div class="h-32 bg-gradient-to-r from-blue-400 to-purple-500">
</div>

  <!-- Profile content -->
  <div class="px-6 pb-6">
    <!-- Avatar (overlapping cover) -->
    <div class="flex justify-center -mt-16 mb-4">
      <img src="avatar.jpg" class="w-32 h-32 rounded-full border-
4 border-white shadow-lg">
    </div>

    <!-- Info -->
    <div class="text-center">
      <h3 class="text-2xl font-bold text-gray-900 mb-1">John
Doe</h3>
      <p class="text-gray-600 mb-4">Product Designer</p>

      <!-- Stats -->
      <div class="flex justify-center space-x-8 mb-6">
        <div>
          <div class="text-xl font-bold text-gray-900">1.2K</div>
          <div class="text-sm text-gray-500">Followers</div>
        </div>
        <div>
          <div class="text-xl font-bold text-gray-900">345</div>
          <div class="text-sm text-gray-500">Following</div>
        </div>
        <div>
          <div class="text-xl font-bold text-gray-900">89</div>
          <div class="text-sm text-gray-500">Posts</div>
        </div>
```

```
      </div>

      <!-- Action button -->
      <button class="w-full bg-blue-600 hover:bg-blue-700 text-
white font-semibold py-2 rounded-lg transition">
        Follow
      </button>
    </div>
  </div>
</div>
```

Navigation Bars

Simple Navigation

```
<nav class="bg-white shadow">
  <div class="max-w-7xl mx-auto px-4 sm:px-6 lg:px-8">
    <div class="flex justify-between items-center h-16">
      <!-- Logo -->
      <div class="flex-shrink-0">
        <h1 class="text-2xl font-bold text-blue-600">Logo</h1>
      </div>

      <!-- Navigation links -->
      <div class="hidden md:flex space-x-8">
        <a href="#" class="text-gray-700 hover:text-blue-600 px-3
py-2 text-sm font-medium transition">
          Home
        </a>
        <a href="#" class="text-gray-700 hover:text-blue-600 px-3
py-2 text-sm font-medium transition">
          About
```

```
        </a>
        <a href="#" class="text-gray-700 hover:text-blue-600 px-3
py-2 text-sm font-medium transition">
          Services
        </a>
        <a href="#" class="text-gray-700 hover:text-blue-600 px-3
py-2 text-sm font-medium transition">
          Contact
        </a>
      </div>

      <!-- CTA Button -->
      <div class="hidden md:block">
        <button class="bg-blue-600 hover:bg-blue-700 text-white
px-4 py-2 rounded-lg text-sm font-medium transition">
          Get Started
        </button>
      </div>

      <!-- Mobile menu button -->
      <div class="md:hidden">
        <button class="text-gray-700 hover:text-blue-600 p-2">
          <svg class="w-6 h-6" fill="none" stroke="currentColor"
viewBox="0 0 24 24">
            <path stroke-linecap="round" stroke-linejoin="round"
stroke-width="2" d="M4 6h16M4 12h16M4 18h16"></path>
          </svg>
        </button>
      </div>
    </div>
  </div>

  <!-- Mobile menu (hidden by default) -->
  <div class="md:hidden hidden" id="mobile-menu">
    <div class="px-2 pt-2 pb-3 space-y-1">
```

```
      <a href="#" class="block px-3 py-2 text-gray-700 hover:bg-
gray-100 rounded-md">Home</a>
      <a href="#" class="block px-3 py-2 text-gray-700 hover:bg-
gray-100 rounded-md">About</a>
      <a href="#" class="block px-3 py-2 text-gray-700 hover:bg-
gray-100 rounded-md">Services</a>
      <a href="#" class="block px-3 py-2 text-gray-700 hover:bg-
gray-100 rounded-md">Contact</a>
    </div>
  </div>
</nav>
```

Navigation with Dropdown

```
<nav class="bg-white shadow">
  <div class="max-w-7xl mx-auto px-4">
    <div class="flex justify-between items-center h-16">
      <div class="flex-shrink-0">
        <h1 class="text-2xl font-bold text-blue-600">Logo</h1>
      </div>

      <div class="hidden md:flex space-x-8">
        <a href="#" class="text-gray-700 hover:text-blue-600 px-3
py-2 font-medium">Home</a>

        <!-- Dropdown menu -->
        <div class="relative group">
          <button class="text-gray-700 group-hover:text-blue-600
px-3 py-2 font-medium flex items-center">
            Products
            <svg class="w-4 h-4 ml-1 transform group-
hover:rotate-180 transition-transform" fill="none"
stroke="currentColor" viewBox="0 0 24 24">
              <path stroke-linecap="round" stroke-
```

```
linejoin="round" stroke-width="2" d="M19 9l-7 7-7-7"></path>
            </svg>
          </button>

          <!-- Dropdown content -->
          <div class="absolute left-0 mt-2 w-48 bg-white rounded-lg shadow-lg opacity-0 invisible group-hover:opacity-100 group-hover:visible transition-all duration-200 z-10">
            <a href="#" class="block px-4 py-2 text-gray-700 hover:bg-gray-100 rounded-t-lg">Product 1</a>
            <a href="#" class="block px-4 py-2 text-gray-700 hover:bg-gray-100">Product 2</a>
            <a href="#" class="block px-4 py-2 text-gray-700 hover:bg-gray-100 rounded-b-lg">Product 3</a>
          </div>
        </div>

        <a href="#" class="text-gray-700 hover:text-blue-600 px-3 py-2 font-medium">About</a>
        <a href="#" class="text-gray-700 hover:text-blue-600 px-3 py-2 font-medium">Contact</a>
      </div>
    </div>
  </div>
</nav>
```

Sidebar Navigation

```
<div class="flex h-screen">
  <!-- Sidebar -->
  <aside class="w-64 bg-gray-800 text-white">
    <!-- Logo -->
    <div class="p-6 border-b border-gray-700">
      <h1 class="text-2xl font-bold">Dashboard</h1>
```

```
    </div>

    <!-- Navigation -->
    <nav class="p-4 space-y-2">
      <!-- Active item -->
      <a href="#" class="flex items-center space-x-3 px-4 py-3
bg-blue-600 rounded-lg">
        <svg class="w-5 h-5" fill="none" stroke="currentColor"
viewBox="0 0 24 24">
          <path stroke-linecap="round" stroke-linejoin="round"
stroke-width="2" d="M3 12l2-2m0 0l7-7 7 7M5 10v10a1 1 0 001
1h3m10-11l2 2m-2-2v10a1 1 0 01-1 1h-3m-6 0a1 1 0 001-1v-4a1 1 0
011-1h2a1 1 0 011 1v4a1 1 0 001 1m-6 0h6"></path>
        </svg>
        <span>Dashboard</span>
      </a>

      <!-- Regular items -->
      <a href="#" class="flex items-center space-x-3 px-4 py-3
hover:bg-gray-700 rounded-lg transition">
        <svg class="w-5 h-5" fill="none" stroke="currentColor"
viewBox="0 0 24 24">
          <path stroke-linecap="round" stroke-linejoin="round"
stroke-width="2" d="M9 19v-6a2 2 0 00-2-2H5a2 2 0 00-2 2v6a2 2 0
002 2h2a2 2 0 002-2zm0 0V9a2 2 0 012-2h2a2 2 0 012 2v10m-6 0a2 2
0 002 2h2a2 2 0 002-2m0 0V5a2 2 0 012-2h2a2 2 0 012 2v14a2 2 0
01-2 2h-2a2 2 0 01-2-2z"></path>
        </svg>
        <span>Analytics</span>
      </a>

      <a href="#" class="flex items-center space-x-3 px-4 py-3
hover:bg-gray-700 rounded-lg transition">
        <svg class="w-5 h-5" fill="none" stroke="currentColor"
viewBox="0 0 24 24">
```

```
          <path stroke-linecap="round" stroke-linejoin="round"
stroke-width="2" d="M12 4.354a4 4 0 110 5.292M15 21H3v-1a6 6 0
0112 0v1zm0 0h6v-1a6 6 0 00-9-5.197M13 7a4 4 0 11-8 0 4 4 0 018
0z"></path>
        </svg>
        <span>Users</span>
      </a>

      <a href="#" class="flex items-center space-x-3 px-4 py-3
hover:bg-gray-700 rounded-lg transition">
        <svg class="w-5 h-5" fill="none" stroke="currentColor"
viewBox="0 0 24 24">
          <path stroke-linecap="round" stroke-linejoin="round"
stroke-width="2" d="M10.325 4.317c.426-1.756 2.924-1.756 3.35
0a1.724 1.724 0 002.573 1.066c1.543-.94 3.31.826 2.37 2.37a1.724
1.724 0 001.065 2.572c1.756.426 1.756 2.924 0 3.35a1.724 1.724 0
00-1.066 2.573c.94 1.543-.826 3.31-2.37 2.37a1.724 1.724 0 00-
2.572 1.065c-.426 1.756-2.924 1.756-3.35 0a1.724 1.724 0 00-
2.573-1.066c-1.543.94-3.31-.826-2.37-2.37a1.724 1.724 0 00-1.065-
2.572c-1.756-.426-1.756-2.924 0-3.35a1.724 1.724 0 001.066-
2.573c-.94-1.543.826-3.31 2.37-2.37.996.608 2.296.07 2.572-
1.065z"></path>
          <path stroke-linecap="round" stroke-linejoin="round"
stroke-width="2" d="M15 12a3 3 0 11-6 0 3 3 0 016 0z"></path>
        </svg>
        <span>Settings</span>
      </a>
    </nav>
  </aside>

  <!-- Main content -->
  <main class="flex-1 bg-gray-100 p-8">
    <h2 class="text-3xl font-bold mb-4">Main Content</h2>
    <p class="text-gray-600">Content goes here...</p>
```

```
  </main>
</div>
```

Forms and Inputs

Complete Form Example

```
<form class="max-w-lg mx-auto bg-white shadow-lg rounded-lg p-8 space-y-6">
  <h2 class="text-2xl font-bold text-gray-900 mb-6">Contact Us</h2>

  <!-- Text input -->
  <div>
    <label for="name" class="block text-sm font-medium text-gray-700 mb-2">
      Full Name *
    </label>
    <input
      type="text"
      id="name"
      required
      class="w-full px-4 py-2 border border-gray-300 rounded-lg focus:ring-2 focus:ring-blue-500 focus:border-transparent transition"
      placeholder="John Doe"
    >
  </div>

  <!-- Email input -->
  <div>
    <label for="email" class="block text-sm font-medium text-
```

```
gray-700 mb-2">
      Email Address *
    </label>
    <input
      type="email"
      id="email"
      required
      class="w-full px-4 py-2 border border-gray-300 rounded-lg
focus:ring-2 focus:ring-blue-500 focus:border-transparent
transition"
      placeholder="you@example.com"
    >
  </div>

  <!-- Select dropdown -->
  <div>
    <label for="subject" class="block text-sm font-medium text-
gray-700 mb-2">
      Subject
    </label>
    <select
      id="subject"
      class="w-full px-4 py-2 border border-gray-300 rounded-lg
focus:ring-2 focus:ring-blue-500 focus:border-transparent
transition bg-white"
    >
      <option>General Inquiry</option>
      <option>Support</option>
      <option>Sales</option>
      <option>Partnership</option>
    </select>
  </div>

  <!-- Textarea -->
  <div>
```

```
    <label for="message" class="block text-sm font-medium text-gray-700 mb-2">
      Message *
    </label>
    <textarea
      id="message"
      rows="4"
      required
      class="w-full px-4 py-2 border border-gray-300 rounded-lg focus:ring-2 focus:ring-blue-500 focus:border-transparent transition resize-none"
      placeholder="Tell us more..."
    ></textarea>
  </div>

  <!-- Checkbox -->
  <div class="flex items-start">
    <input
      type="checkbox"
      id="terms"
      class="mt-1 w-4 h-4 text-blue-600 border-gray-300 rounded focus:ring-2 focus:ring-blue-500"
    >
    <label for="terms" class="ml-2 text-sm text-gray-700">
      I agree to the <a href="#" class="text-blue-600 hover:underline">Terms and Conditions</a>
    </label>
  </div>

  <!-- Submit button -->
  <button
    type="submit"
    class="w-full bg-blue-600 hover:bg-blue-700 active:bg-blue-800 text-white font-semibold py-3 rounded-lg shadow-md hover:shadow-lg transition-all duration-150 focus:outline-none
```

```
focus:ring-4 focus:ring-blue-300"
  >
    Send Message
  </button>
</form>
```

Input Variations

```
<!-- Input with icon (left) -->
<div class="relative">
  <svg class="absolute left-3 top-3 w-5 h-5 text-gray-400"
fill="none" stroke="currentColor" viewBox="0 0 24 24">
    <path stroke-linecap="round" stroke-linejoin="round" stroke-
width="2" d="M21 21l-6-6m2-5a7 7 0 11-14 0 7 7 0 0114 0z"></path>
  </svg>
  <input
    type="text"
    class="w-full pl-10 pr-4 py-2 border border-gray-300 rounded-
lg focus:ring-2 focus:ring-blue-500 focus:border-transparent"
    placeholder="Search..."
  >
</div>

<!-- Input with icon (right) -->
<div class="relative">
  <input
    type="password"
    class="w-full px-4 py-2 pr-10 border border-gray-300 rounded-
lg focus:ring-2 focus:ring-blue-500 focus:border-transparent"
    placeholder="Password"
  >
  <button class="absolute right-3 top-3 text-gray-400 hover:text-
gray-600">
    <svg class="w-5 h-5" fill="none" stroke="currentColor"
```

```
viewBox="0 0 24 24">
      <path stroke-linecap="round" stroke-linejoin="round"
stroke-width="2" d="M15 12a3 3 0 11-6 0 3 3 0 016 0z"></path>
      <path stroke-linecap="round" stroke-linejoin="round"
stroke-width="2" d="M2.458 12C3.732 7.943 7.523 5 12 5c4.478 0
8.268 2.943 9.542 7-1.274 4.057-5.064 7-9.542 7-4.477 0-8.268-
2.943-9.542-7z"></path>
    </svg>
  </button>
</div>

<!-- Input with error -->
<div>
  <input
    type="email"
    class="w-full px-4 py-2 border-2 border-red-500 rounded-lg
focus:ring-2 focus:ring-red-300 focus:border-red-500"
    placeholder="Email"
    value="invalid-email"
  >
  <p class="mt-1 text-sm text-red-600">Please enter a valid email
address</p>
</div>

<!-- Input with success -->
<div>
  <input
    type="email"
    class="w-full px-4 py-2 border-2 border-green-500 rounded-lg
focus:ring-2 focus:ring-green-300 focus:border-green-500"
    value="user@example.com"
  >
  <p class="mt-1 text-sm text-green-600 flex items-center">
    <svg class="w-4 h-4 mr-1" fill="currentColor" viewBox="0 0 20
20">
```

```
      <path fill-rule="evenodd" d="M10 18a8 8 0 100-16 8 8 0 000
16zm3.707-9.293a1 1 0 00-1.414-1.414L9 10.586 7.707 9.293a1 1 0
00-1.414 1.414l2 2a1 1 0 001.414 0l4-4z" clip-rule="evenodd">
</path>
    </svg>
    Email is valid
  </p>
</div>

<!-- Floating label input -->
<div class="relative">
  <input
    type="text"
    id="floating"
    class="peer w-full px-4 pt-6 pb-2 border border-gray-300
rounded-lg focus:ring-2 focus:ring-blue-500 focus:border-
transparent placeholder-transparent"
    placeholder="Name"
  >
  <label
    for="floating"
    class="absolute left-4 top-2 text-xs text-gray-500
transition-all peer-placeholder-shown:text-base peer-placeholder-
shown:top-4 peer-focus:text-xs peer-focus:top-2 peer-focus:text-
blue-500"
  >
    Full Name
  </label>
</div>
```

Modals and Overlays

Basic Modal

```
<!-- Modal backdrop -->
<div class="fixed inset-0 bg-black bg-opacity-50 flex items-
center justify-center p-4 z-50">

  <!-- Modal content -->
  <div class="bg-white rounded-xl shadow-2xl max-w-md w-full p-6
relative animate-fade-in">

    <!-- Close button -->
    <button class="absolute top-4 right-4 text-gray-400
hover:text-gray-600 transition">
      <svg class="w-6 h-6" fill="none" stroke="currentColor"
viewBox="0 0 24 24">
        <path stroke-linecap="round" stroke-linejoin="round"
stroke-width="2" d="M6 18L18 6M6 6l12 12"></path>
      </svg>
    </button>

    <!-- Modal header -->
    <div class="mb-4">
      <h3 class="text-2xl font-bold text-gray-900">Modal
Title</h3>
      <p class="text-gray-600 mt-2">This is a modal dialog with
some content.</p>
    </div>

    <!-- Modal body -->
    <div class="mb-6">
      <p class="text-gray-700">
        Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Sed do eiusmod tempor incididunt ut labore et dolore magna
aliqua.
      </p>
```

```
    </div>

    <!-- Modal footer -->
    <div class="flex space-x-4">
      <button class="flex-1 px-4 py-2 border border-gray-300
rounded-lg hover:bg-gray-50 transition">
        Cancel
      </button>
      <button class="flex-1 px-4 py-2 bg-blue-600 hover:bg-blue-
700 text-white rounded-lg transition">
        Confirm
      </button>
    </div>

  </div>
</div>
```

Confirmation Modal

```
<div class="fixed inset-0 bg-black bg-opacity-50 flex items-
center justify-center p-4 z-50">
  <div class="bg-white rounded-xl shadow-2xl max-w-sm w-full p-
6">

    <!-- Icon -->
    <div class="flex justify-center mb-4">
      <div class="w-12 h-12 rounded-full bg-red-100 flex items-
center justify-center">
        <svg class="w-6 h-6 text-red-600" fill="none"
stroke="currentColor" viewBox="0 0 24 24">
          <path stroke-linecap="round" stroke-linejoin="round"
stroke-width="2" d="M12 9v2m0 4h.01m-6.938 4h13.856c1.54 0 2.502-
1.667 1.732-3L13.732 4c-.77-1.333-2.694-1.333-3.464 0L3.34
16c-.77 1.333.192 3 1.732 3z"></path>
```

```
        </svg>
      </div>
    </div>

    <!-- Content -->
    <h3 class="text-xl font-bold text-gray-900 text-center mb-2">
      Delete Account?
    </h3>
    <p class="text-gray-600 text-center mb-6">
      This action cannot be undone. All your data will be
permanently deleted.
    </p>

    <!-- Actions -->
    <div class="flex flex-col space-y-3">
      <button class="w-full px-4 py-2 bg-red-600 hover:bg-red-700
text-white font-semibold rounded-lg transition">
        Yes, Delete
      </button>
      <button class="w-full px-4 py-2 border border-gray-300
hover:bg-gray-50 rounded-lg transition">
        Cancel
      </button>
    </div>

  </div>
</div>
```

Slide-over Panel

```
<div class="fixed inset-0 z-50">
  <!-- Backdrop -->
  <div class="fixed inset-0 bg-black bg-opacity-50"></div>
```

```
  <!-- Panel -->
  <div class="fixed inset-y-0 right-0 max-w-md w-full bg-white
shadow-2xl overflow-y-auto">

    <!-- Header -->
    <div class="sticky top-0 bg-white border-b border-gray-200
px-6 py-4 flex items-center justify-between z-10">
      <h2 class="text-xl font-bold text-gray-900">Panel
Title</h2>
      <button class="text-gray-400 hover:text-gray-600
transition">
        <svg class="w-6 h-6" fill="none" stroke="currentColor"
viewBox="0 0 24 24">
          <path stroke-linecap="round" stroke-linejoin="round"
stroke-width="2" d="M6 18L18 6M6 6l12 12"></path>
        </svg>
      </button>
    </div>

    <!-- Content -->
    <div class="p-6">
      <p class="text-gray-600 mb-4">Panel content goes here...
</p>
      <!-- More content -->
    </div>

    <!-- Footer -->
    <div class="sticky bottom-0 bg-white border-t border-gray-200
px-6 py-4">
      <button class="w-full bg-blue-600 hover:bg-blue-700 text-
white py-2 rounded-lg transition">
        Save Changes
      </button>
    </div>
```

```
  </div>
</div>
```

Common UI Patterns

Alert/Toast Notifications

```
<!-- Success toast -->
<div class="fixed top-4 right-4 bg-green-50 border-l-4 border-green-500 rounded-lg shadow-lg p-4 max-w-sm animate-slide-down">
  <div class="flex items-start">
    <svg class="w-6 h-6 text-green-500 mt-0.5" fill="currentColor" viewBox="0 0 20 20">
      <path fill-rule="evenodd" d="M10 18a8 8 0 100-16 8 8 0 000 16zm3.707-9.293a1 1 0 00-1.414-1.414L9 10.586 7.707 9.293a1 1 0 00-1.414 1.414l2 2a1 1 0 001.414 0l4-4z" clip-rule="evenodd">
</path>
    </svg>
    <div class="ml-3 flex-1">
      <p class="text-sm font-medium text-green-900">Success!</p>
      <p class="text-sm text-green-700 mt-1">Your changes have been saved.</p>
    </div>
    <button class="ml-4 text-green-500 hover:text-green-700">
      <svg class="w-5 h-5" fill="currentColor" viewBox="0 0 20 20">
        <path fill-rule="evenodd" d="M4.293 4.293a1 1 0 011.414 0L10 8.586l4.293-4.293a1 1 0 111.414 1.414L11.414 10l4.293 4.293a1 1 0 01-1.414 1.414L10 11.414l-4.293 4.293a1 1 0 01-1.414-1.414L8.586 10 4.293 5.707a1 1 0 010-1.414z" clip-rule="evenodd">
</path>
      </svg>
```

```
    </button>
  </div>
</div>

<!-- Error toast -->
<div class="fixed top-4 right-4 bg-red-50 border-l-4 border-red-
500 rounded-lg shadow-lg p-4 max-w-sm">
  <div class="flex items-start">
    <svg class="w-6 h-6 text-red-500 mt-0.5" fill="currentColor"
viewBox="0 0 20 20">
      <path fill-rule="evenodd" d="M10 18a8 8 0 100-16 8 8 0 000
16zM8.707 7.293a1 1 0 00-1.414 1.414L8.586 10l-1.293 1.293a1 1 0
101.414 1.414L10 11.414l1.293 1.293a1 1 0 001.414-1.414L11.414
10l1.293-1.293a1 1 0 00-1.414-1.414L10 8.586 8.707 7.293z" clip-
rule="evenodd"></path>
    </svg>
    <div class="ml-3 flex-1">
      <p class="text-sm font-medium text-red-900">Error!</p>
      <p class="text-sm text-red-700 mt-1">Something went wrong.
</p>
    </div>
    <button class="ml-4 text-red-500 hover:text-red-700">
      <svg class="w-5 h-5" fill="currentColor" viewBox="0 0 20
20">
        <path fill-rule="evenodd" d="M4.293 4.293a1 1 0 011.414
0L10 8.586l4.293-4.293a1 1 0 111.414 1.414L11.414 10l4.293
4.293a1 1 0 01-1.414 1.414L10 11.414l-4.293 4.293a1 1 0 01-1.414-
1.414L8.586 10 4.293 5.707a1 1 0 010-1.414z" clip-rule="evenodd">
</path>
      </svg>
    </button>
  </div>
</div>
```

Pagination

```
<div class="flex items-center justify-center space-x-2">
  <!-- Previous button -->
  <button class="px-3 py-2 border border-gray-300 rounded-lg
hover:bg-gray-50 disabled:opacity-50 disabled:cursor-not-allowed
transition">
    <svg class="w-5 h-5" fill="none" stroke="currentColor"
viewBox="0 0 24 24">
      <path stroke-linecap="round" stroke-linejoin="round"
stroke-width="2" d="M15 19l-7-7 7-7"></path>
    </svg>
  </button>

  <!-- Page numbers -->
  <button class="px-4 py-2 bg-blue-600 text-white rounded-
lg">1</button>
  <button class="px-4 py-2 border border-gray-300 rounded-lg
hover:bg-gray-50 transition">2</button>
  <button class="px-4 py-2 border border-gray-300 rounded-lg
hover:bg-gray-50 transition">3</button>
  <span class="px-2">...</span>
  <button class="px-4 py-2 border border-gray-300 rounded-lg
hover:bg-gray-50 transition">10</button>

  <!-- Next button -->
  <button class="px-3 py-2 border border-gray-300 rounded-lg
hover:bg-gray-50 transition">
    <svg class="w-5 h-5" fill="none" stroke="currentColor"
viewBox="0 0 24 24">
      <path stroke-linecap="round" stroke-linejoin="round"
stroke-width="2" d="M9 5l7 7-7 7"></path>
    </svg>
  </button>
</div>
```

Loading Skeleton

```
<div class="animate-pulse space-y-4">
  <!-- Header skeleton -->
  <div class="h-4 bg-gray-200 rounded w-3/4"></div>

  <!-- Content skeletons -->
  <div class="space-y-3">
    <div class="h-4 bg-gray-200 rounded"></div>
    <div class="h-4 bg-gray-200 rounded w-5/6"></div>
    <div class="h-4 bg-gray-200 rounded w-4/6"></div>
  </div>

  <!-- Image skeleton -->
  <div class="h-64 bg-gray-200 rounded"></div>

  <!-- Button skeleton -->
  <div class="h-10 bg-gray-200 rounded w-32"></div>
</div>
```

Badge/Tag

```
<!-- Basic badges -->
<span class="inline-flex items-center px-3 py-1 rounded-full
text-sm font-medium bg-blue-100 text-blue-800">
  New
</span>

<span class="inline-flex items-center px-3 py-1 rounded-full
text-sm font-medium bg-green-100 text-green-800">
  Active
</span>

<span class="inline-flex items-center px-3 py-1 rounded-full
```

```
text-sm font-medium bg-red-100 text-red-800">
  Urgent
</span>

<!-- Badge with icon -->
<span class="inline-flex items-center px-3 py-1 rounded-full
text-sm font-medium bg-purple-100 text-purple-800">
  <svg class="w-4 h-4 mr-1" fill="currentColor" viewBox="0 0 20
20">
    <path fill-rule="evenodd" d="M10 18a8 8 0 100-16 8 8 0 000
16zm3.707-9.293a1 1 0 00-1.414-1.414L9 10.586 7.707 9.293a1 1 0
00-1.414 1.414l2 2a1 1 0 001.414 0l4-4z" clip-rule="evenodd">
</path>
  </svg>
  Verified
</span>

<!-- Badge with remove button -->
<span class="inline-flex items-center px-3 py-1 rounded-full
text-sm font-medium bg-gray-200 text-gray-800">
  React
  <button class="ml-2 hover:text-gray-600">
    <svg class="w-4 h-4" fill="currentColor" viewBox="0 0 20 20">
      <path fill-rule="evenodd" d="M4.293 4.293a1 1 0 011.414
0L10 8.586l4.293-4.293a1 1 0 111.414 1.414L11.414 10l4.293
4.293a1 1 0 01-1.414 1.414L10 11.414l-4.293 4.293a1 1 0 01-1.414-
1.414L8.586 10 4.293 5.707a1 1 0 010-1.414z" clip-rule="evenodd">
</path>
    </svg>
  </button>
</span>
```

When to Use @apply Directive

The `@apply` directive lets you extract repeated utility patterns into custom CSS classes.

When to Use @apply

✅ **Good use cases:**

- Component libraries where you need consistent, reusable styles
- Extremely repetitive patterns used dozens of times
- Third-party integrations that require CSS classes
- Team prefers traditional CSS workflows

❌ **Avoid @apply for:**

- One-off components (just use utilities)
- Prematurely optimizing (only extract when you see repetition)
- Everything (defeats the purpose of utility-first)

@apply Example

```
/* In your CSS file */
@layer components {
  .btn {
    @apply px-6 py-3 font-semibold rounded-lg transition-all
duration-150 focus:outline-none focus:ring-4;
  }

  .btn-primary {
    @apply bg-blue-600 hover:bg-blue-700 active:bg-blue-800 text-
white focus:ring-blue-300;
  }

  .btn-secondary {
```

```
    @apply bg-gray-200 hover:bg-gray-300 active:bg-gray-400 text-gray-900 focus:ring-gray-300;
  }

  .card {
    @apply bg-white rounded-lg shadow-md overflow-hidden;
  }

  .card-body {
    @apply p-6;
  }
}
```

Usage:

```
<button class="btn btn-primary">Primary</button>
<button class="btn btn-secondary">Secondary</button>

<div class="card">
  <div class="card-body">
    <h3 class="text-xl font-bold">Card Title</h3>
  </div>
</div>
```

Important: This should be your **last resort**. Prefer component frameworks (React, Vue) or copy-paste patterns first.

Key Takeaways

✅ **Components** can be HTML patterns, JS components, or CSS with @apply
✅ Build **button systems** with variants, sizes, and states
✅ **Cards** are versatile - use for products, profiles, stats

✅ **Navigation** should be responsive (mobile menu)
✅ **Forms** need focus states, validation, and error handling
✅ **Modals** require backdrop, positioning, and animations
✅ Use **@apply sparingly** - only for heavily repeated patterns
✅ Prefer **JS components** over @apply when possible

Practice Assignments

Assignment 1 (Easy): Button Component System

Create a complete button system with:

- 4 variants (primary, secondary, outline, ghost)
- 3 sizes (small, medium, large)
- All states (hover, focus, active, disabled, loading)
- With and without icons
- Document usage examples

Assignment 2 (Medium): Card Gallery

Build a responsive card gallery with:

- 3 different card types (basic, horizontal, profile)
- Hover effects (shadow, image zoom, content reveal)
- Responsive grid (1, 2, 3, 4 columns)
- Interactive elements (buttons, links)
- Loading skeletons

Assignment 3 (Medium): Complete Form

Create a multi-step form with:

- Various input types (text, email, select, textarea, checkbox, radio)
- Input validation states (error, success, default)
- Floating labels
- Icon inputs
- Step indicator
- Submit with loading state

Assignment 4 (Hard): Dashboard Component Library

Build a dashboard with reusable components:

- Sidebar navigation (collapsible)
- Top navbar with dropdowns
- Stat cards (4 variations)
- Data table with pagination
- Modal for actions
- Toast notifications
- All fully responsive

Assignment 5 (Hard): Design System

Create a complete design system documentation site:

1. **Components page** showing all your components
2. **Color system** with all variants
3. **Typography scale** demonstrated
4. **Spacing system** visualized
5. **Interactive examples** with code snippets
6. **React/Vue components** (if applicable)
7. Each component should have:
    - Visual example

- Code snippet
- Props/options documentation
- Usage guidelines

Deliverable: A fully functional design system website that other developers could use as a reference.

Ready for **Module 11: Best Practices & Optimization**? Let me know!

1/6/2026, 1:23:21 PM

MODULE 11: BEST PRACTICES & OPTIMIZATION

Class Organization Strategies

Long lists of utility classes can become hard to read. Let's explore strategies to keep them manageable.

Logical Grouping

Group related utilities together for better readability:

```
<!-- ❌ Unorganized (hard to scan) -->
<div class="text-white p-6 rounded-lg hover:shadow-xl bg-blue-600
flex transition-shadow items-center shadow-lg space-x-4">

<!-- ✅ Organized by category -->
<div class="
  flex items-center space-x-4
  p-6 rounded-lg
```

```
  bg-blue-600 text-white
  shadow-lg hover:shadow-xl
  transition-shadow
">
```

Recommended grouping order:

1. **Layout** (display, position, flex/grid)
2. **Sizing** (width, height, padding, margin)
3. **Typography** (font, text size, color)
4. **Visual** (background, border, shadow)
5. **Interactive** (hover, focus, active)
6. **Responsive** (breakpoint variants)
7. **State** (group-hover, peer)

Example with Comments

```
<button class="
  /* Layout */
  flex items-center justify-center

  /* Sizing */
  px-6 py-3

  /* Typography */
  text-white font-semibold

  /* Visual */
  bg-blue-600 rounded-lg shadow-md

  /* Interactive */
  hover:bg-blue-700 hover:shadow-lg
  active:bg-blue-800 active:scale-95
```

```
  focus:outline-none focus:ring-4 focus:ring-blue-300

  /* Animation */
  transition-all duration-150
">
  Click Me
</button>
```

Multi-line Formatting

For complex components, use multiple lines:

```
<div class="
  relative
  flex flex-col
  w-full max-w-sm
  p-6
  bg-white
  rounded-xl shadow-lg
  hover:shadow-2xl
  transform hover:-translate-y-1
  transition-all duration-300
  group
">
  <img
    src="product.jpg"
    class="
      w-full h-48
      object-cover
      rounded-lg
      mb-4
      transform group-hover:scale-105
      transition-transform duration-500
    "
  >
```

```
  <h3 class="text-xl font-bold text-gray-900 group-hover:text-
blue-600 transition-colors">
    Product Name
  </h3>
</div>
```

Why this works:

- Easy to see what classes are applied
- Easy to add/remove classes
- Easy to spot duplicate classes
- Better for code reviews
- Easier to maintain

Using Prettier Plugin

Install the official Tailwind CSS Prettier plugin for automatic formatting:

```
npm install -D prettier prettier-plugin-tailwindcss
```

Create `.prettierrc`:

```
{
  "plugins": ["prettier-plugin-tailwindcss"],
  "tailwindConfig": "./tailwind.config.js"
}
```

It automatically sorts classes in a logical order:

```
<!-- Before -->
<div class="text-white bg-blue-600 p-4 rounded">
```

```
<!-- After (automatically sorted) -->
<div class="rounded bg-blue-600 p-4 text-white">
```

Avoiding Class Bloat

1. Don't Over-Specify

```
<!-- ❌ Redundant classes -->
<div class="flex flex-row items-center justify-start">
  <!-- flex-row is default, justify-start is default -->
</div>

<!-- ✅ Only what you need -->
<div class="flex items-center">
  <!-- Cleaner and does the same thing -->
</div>
```

2. Use Shorthand When Possible

```
<!-- ❌ Long form -->
<div class="padding-top: 1rem padding-bottom: 1rem padding-left:
1rem padding-right: 1rem">

<!-- ✅ Shorthand -->
<div class="p-4">
```

```
<!-- ❌ Individual sides -->
<div class="mt-4 mb-4">
```

```
<!-- ✅ Use y-axis shorthand -->
<div class="my-4">
```

3. Avoid Arbitrary Values Unless Necessary

```
<!-- ❌ Arbitrary values (defeats design system) -->
<div class="p-[13px] m-[27px] text-[#3b7af4]">

<!-- ✅ Use the scale -->
<div class="p-3 m-6 text-blue-600">
```

When arbitrary values ARE okay:

- One-off design requirements
- Matching exact brand specifications
- Prototyping before adding to config

4. Extract to Components

Instead of repeating long class strings:

```
<!-- ❌ Repeated everywhere -->
<button class="px-6 py-3 bg-blue-600 hover:bg-blue-700 text-white
font-semibold rounded-lg shadow-md transition">Button 1</button>
<button class="px-6 py-3 bg-blue-600 hover:bg-blue-700 text-white
font-semibold rounded-lg shadow-md transition">Button 2</button>
<button class="px-6 py-3 bg-blue-600 hover:bg-blue-700 text-white
font-semibold rounded-lg shadow-md transition">Button 3</button>
```

✅ **Extract to React/Vue component:**

```
// Button.jsx
function Button({ children, ...props }) {
```

```
  return (
    <button
      className="px-6 py-3 bg-blue-600 hover:bg-blue-700 text-
white font-semibold rounded-lg shadow-md transition"
      {...props}
    >
      {children}
    </button>
  )
}

// Usage
<Button>Button 1</Button>
<Button>Button 2</Button>
<Button>Button 3</Button>
```

5. Use CSS Variables for Dynamic Values

```
<!-- ❌ Inline styles break Tailwind's design system -->
<div style="color: #ff0000; background: #00ff00;">

<!-- ✅ Use CSS variables + Tailwind -->
<div class="text-[var(--primary)] bg-[var(--secondary)]">
```

Or better yet, configure them in `tailwind.config.js`.

PurgeCSS and Production Builds

Tailwind generates thousands of utility classes. In production, you only want the ones you actually use.

How Purging Works

Development mode:

- All utilities available
- Large CSS file (~3-4MB uncompressed)
- Fast rebuilds

Production mode:

- Only used utilities included
- Tiny CSS file (~10-50KB)
- Optimized and minified

Configuration

In `tailwind.config.js`, the `content` array tells Tailwind where to look:

```
export default {
  content: [
    "./index.html",
    "./src/**/*.{js,ts,jsx,tsx,vue}",
    "./components/**/*.{js,ts,jsx,tsx,vue}",
  ],
  // ...
}
```

Tailwind scans these files and includes only the classes it finds.

Build Commands

Vite:

```
{
  "scripts": {
```

```
    "dev": "vite",
    "build": "vite build",  // Automatically purges
    "preview": "vite preview"
  }
}
```

PostCSS CLI:

```
{
  "scripts": {
    "dev": "tailwindcss -i ./src/input.css -o ./dist/output.css -
-watch",
    "build": "NODE_ENV=production tailwindcss -i ./src/input.css
-o ./dist/output.css --minify"
  }
}
```

What Gets Purged

✅ **Keeps:**

```
<div class="bg-blue-500">  <!-- Static class -->
<div className="text-xl">  <!-- JSX -->
```

⚠️ **May purge (unsafe):**

```
// Dynamic classes can be missed
const color = 'blue'
<div className={`bg-${color}-500`}>  // Might be purged!

// String concatenation
<div className={'text-' + size}>  // Might be purged!
```

✅ **Safe alternatives:**

```
// Use complete class names
const className = color === 'blue' ? 'bg-blue-500' : 'bg-red-500'

// Or use safelist
```

Safelisting Classes

If you generate classes dynamically, add them to safelist:

```
export default {
  content: ["./src/**/*.{html,js}"],
  safelist: [
    'bg-red-500',
    'bg-green-500',
    'bg-blue-500',
    // Pattern matching
    {
      pattern: /bg-(red|green|blue)-(400|500|600)/,
    },
  ],
}
```

Checking Bundle Size

After building, check the size:

```
npm run build
ls -lh dist/assets/*.css
```

Good targets:

- Simple site: 5-15KB

- Complex app: 20-50KB
- Very complex: 50-100KB

If your CSS is >100KB, you likely have:

- Too broad `content` patterns
- Classes in `safelist` that aren't needed
- Unused plugins
- Missing purge configuration

Performance Optimization

1. Optimize the Build

Enable compression:

```
// vite.config.js
export default {
  build: {
    cssCodeSplit: true,
    minify: 'esbuild',
    rollupOptions: {
      output: {
        manualChunks: {
          vendor: ['react', 'react-dom'],
        },
      },
    },
  },
}
```

Use `@layer` for custom styles:

```
@layer components {
  .btn {
    @apply px-4 py-2 rounded;
  }
}
```

This ensures proper CSS ordering and tree-shaking.

2. Reduce Redundant Utilities

Avoid this pattern:

```
<!-- ❌ Too many hover states -->
<div class="
  bg-blue-500 hover:bg-blue-600
  text-white hover:text-white
  border-blue-500 hover:border-blue-600
  shadow-md hover:shadow-lg
  scale-100 hover:scale-105
  opacity-100 hover:opacity-100
">
```

Better:

```
<!-- ✅ Only specify what changes -->
<div class="
  bg-blue-500 hover:bg-blue-600
  text-white
  shadow-md hover:shadow-lg
  transform hover:scale-105
  transition
">
```

3. Use @apply Wisely

Don't do this:

```
/* ❌ Just recreating Bootstrap */
@layer components {
  .container {
    @apply max-w-7xl mx-auto px-4;
  }

  .row {
    @apply flex flex-wrap;
  }

  .col {
    @apply flex-1;
  }

  /* ... 50 more components */
}
```

This defeats the purpose of utility-first CSS!

Do this:

```
/* ✅ Only for truly repeated patterns */
@layer components {
  .btn-primary {
    @apply px-6 py-3 bg-blue-600 hover:bg-blue-700 text-white
rounded-lg transition;
  }
}
```

4. Lazy Load Non-Critical CSS

For very large apps, consider code-splitting CSS:

```
// In React with lazy loading
const Dashboard = lazy(() => import('./Dashboard'))

// Dashboard.css only loads when Dashboard component is rendered
```

5. Use JIT Mode (Default in v3+)

Tailwind v3+ uses Just-In-Time compilation by default:

Benefits:

- Instant build times
- All variants available (no configuration needed)
- Arbitrary values work out of the box
- Smaller CSS in development

It's automatic - no configuration needed!

Accessibility Considerations

Tailwind makes it easy to build, but you're responsible for accessibility.

1. Focus States

❌ **Never do this:**

```
<button class="focus:outline-none">
  <!-- No visible focus indicator! -->
</button>
```

✅ **Always provide focus indication:**

```
<button class="focus:outline-none focus:ring-4 focus:ring-blue-300">
  Accessible button
</button>
```

2. Color Contrast

Use tools to check contrast ratios:

- WebAIM Contrast Checker
- Browser DevTools (Chrome, Firefox)

❌ **Poor contrast:**

```
<div class="bg-yellow-300 text-yellow-100">
  <!-- Can't read this! -->
</div>
```

✅ **Good contrast:**

```
<div class="bg-yellow-500 text-gray-900">
  <!-- Clear and readable -->
</div>
```

WCAG Standards:

- **AA**: 4.5:1 for normal text, 3:1 for large text
- **AAA**: 7:1 for normal text, 4.5:1 for large text

3. Screen Reader Support

Use Tailwind's `sr-only` utility for screen-reader-only content:

```
<button>
  <svg class="w-5 h-5" aria-hidden="true">
    <!-- Icon -->
  </svg>
  <span class="sr-only">Close menu</span>
</button>
```

The `sr-only` class:

```
.sr-only {
  position: absolute;
  width: 1px;
  height: 1px;
  padding: 0;
  margin: -1px;
  overflow: hidden;
  clip: rect(0, 0, 0, 0);
  white-space: nowrap;
  border-width: 0;
}
```

4. Semantic HTML

Tailwind doesn't replace semantic HTML:

❌ **Bad:**

```
<div class="text-3xl font-bold mb-4">Page Title</div>
<div class="cursor-pointer" onclick="doSomething()">Click
me</div>
```

✅ **Good:**

```
<h1 class="text-3xl font-bold mb-4">Page Title</h1>
<button class="cursor-pointer" onclick="doSomething()">Click me</button>
```

5. Keyboard Navigation

Ensure all interactive elements are keyboard accessible:

```
<!-- ✅ Native button - keyboard accessible by default -->
<button class="px-4 py-2 bg-blue-600 text-white rounded">
  Click me
</button>

<!-- ❌ Div as button - not keyboard accessible -->
<div class="px-4 py-2 bg-blue-600 text-white rounded cursor-pointer" onclick="handleClick()">
  Don't do this
</div>

<!-- ✅ If you must use div, add proper ARIA -->
<div
  role="button"
  tabindex="0"
  class="px-4 py-2 bg-blue-600 text-white rounded cursor-pointer"
  onclick="handleClick()"
  onkeypress="handleKeyPress(event)"
>
  Better, but just use <button>
</div>
```

6. Hidden Content

Hide visually but keep for screen readers:

```
<span class="sr-only">Loading...</span>
<svg class="animate-spin">
<!-- Spinner -->
</svg>
```

Hide from everyone:

```
<div class="hidden">Completely hidden</div>
```

Hide from screen readers only:

```
<div aria-hidden="true">
  Visible but screen readers ignore it
</div>
```

7. Form Labels

❌ **Bad:**

```
<input type="text" placeholder="Enter your name" class="border p-
2 rounded">
<!-- No label! -->
```

✅ **Good:**

```
<label for="name" class="block mb-2">Name</label>
<input
  id="name"
  type="text"
  class="border p-2 rounded"
>
```

Or use `sr-only` for visual-only designs:

```
<label for="search" class="sr-only">Search</label>
<input
  id="search"
  type="text"
  placeholder="Search..."
  class="border p-2 rounded"
>
```

When to Use Custom CSS vs Utilities

Use Utilities When:

✅ Building standard UI components
✅ Prototyping rapidly
✅ The design follows common patterns
✅ You want maximum flexibility
✅ Classes are used 1-5 times

Example:

```
<div class="flex items-center space-x-4 p-6 bg-white rounded-lg
shadow">
  <!-- Perfect use of utilities -->
</div>
```

Use Custom CSS When:

✅ Complex animations not in Tailwind
✅ Very specific design requirements
✅ Exact same pattern repeated 20+ times
✅ Working with third-party libraries
✅ Pseudo-elements (::before, ::after) with complex content

Example:

```
@layer components {
  .custom-scrollbar::-webkit-scrollbar {
    width: 8px;
  }

  .custom-scrollbar::-webkit-scrollbar-track {
    @apply bg-gray-100;
  }

  .custom-scrollbar::-webkit-scrollbar-thumb {
    @apply bg-gray-400 rounded-full;
  }

  .custom-scrollbar::-webkit-scrollbar-thumb:hover {
    @apply bg-gray-600;
  }
}
```

```
@keyframes wiggle {
  0%, 100% { transform: rotate(-3deg); }
  50% { transform: rotate(3deg); }
}

@layer utilities {
  .animate-wiggle {
    animation: wiggle 1s ease-in-out infinite;
  }
}
```

Hybrid Approach (Recommended)

Most projects use both:

Base utilities: 90% of styling
Custom CSS: 10% for specific needs

```
<!-- Mostly utilities -->
<div class="flex items-center p-4 custom-pattern">
  <!-- custom-pattern handles the 10% that's unique -->
</div>
```

Component Extraction Strategies

When to Extract

Extract when you have:

- Same pattern used 5+ times
- Complex class combinations (10+ classes)
- Need variants (sizes, colors)
- Team collaboration requires consistency

Don't extract:

- One-off components
- Simple, 3-4 class combinations
- Prototypes and experiments

Extraction Hierarchy

Level 1: Copy-paste pattern (no extraction)

```
<!-- Use 2-3 times: Just copy -->
<button class="px-4 py-2 bg-blue-600 text-white
```

```
rounded">Button</button>
```

Level 2: Component (React/Vue)

```
// Used 5+ times: Extract to component
function Button({ children }) {
  return (
    <button className="px-6 py-3 bg-blue-600 hover:bg-blue-700
text-white font-semibold rounded-lg transition">
      {children}
    </button>
  )
}
```

Level 3: Component with Variants

```
// Many variations needed
function Button({ variant = 'primary', size = 'md', children }) {
  const baseClasses = "font-semibold rounded-lg transition"

  const variants = {
    primary: "bg-blue-600 hover:bg-blue-700 text-white",
    secondary: "bg-gray-200 hover:bg-gray-300 text-gray-900",
    danger: "bg-red-600 hover:bg-red-700 text-white",
  }

  const sizes = {
    sm: "px-4 py-2 text-sm",
    md: "px-6 py-3",
    lg: "px-8 py-4 text-lg",
  }

  return (
    <button className={`${baseClasses} ${variants[variant]}
```

```
${sizes[size]}`}>
      {children}
    </button>
  )
}
```

Level 4: CSS with @apply (last resort)

```
/* Used everywhere, across multiple projects */
@layer components {
  .btn {
    @apply px-6 py-3 font-semibold rounded-lg transition;
  }

  .btn-primary {
    @apply bg-blue-600 hover:bg-blue-700 text-white;
  }
}
```

Decision Tree

```
Is it used more than once?
├─ No → Use inline utilities
└─ Yes → Is it used 5+ times?
    ├─ No → Use inline utilities (copy-paste is ok)
    └─ Yes → Using a component framework?
        ├─ Yes → Extract to component
        └─ No → Is it used 20+ times across many files?
            ├─ Yes → Consider @apply
            └─ No → Extract to component or keep inline
```

Common Mistakes to Avoid

1. Over-engineering Early

```
<!-- ❌ Creating components for everything immediately -->
<!-- You've only used it once! -->
<MyButton variant="primary" size="lg">Click</MyButton>

<!-- ✅ Start simple -->
<button class="px-6 py-3 bg-blue-600 text-white rounded-
lg">Click</button>
<!-- Extract later if you use it often -->
```

2. Fighting the Framework

```
<!-- ❌ Using custom CSS for everything -->
<div class="my-custom-flex-center">
  <!-- Why? Tailwind has utilities for this -->
</div>

<!-- ✅ Use utilities first -->
<div class="flex items-center justify-center">
  <!-- Clear what it does -->
</div>
```

3. Not Using the Design System

```
<!-- ❌ Arbitrary values everywhere -->
<div class="p-[13px] m-[27px] text-[#3b7af4]">

<!-- ✅ Use the scale -->
<div class="p-3 m-6 text-blue-600">
```

4. Ignoring Purge Configuration

```
// ❌ Incorrect content paths
export default {
  content: ["./src/index.html"],  // Only one file!
}

// ✅ Correct
export default {
  content: [
    "./index.html",
    "./src/**/*.{js,jsx,ts,tsx}",
  ],
}
```

5. Poor Responsive Design

```
<!-- ❌ Desktop-first thinking -->
<div class="text-2xl md:text-base">
  Gets smaller on tablets? Backwards!
</div>

<!-- ✅ Mobile-first -->
<div class="text-base md:text-2xl">
  Gets larger on tablets
</div>
```

6. Forgetting Accessibility

```
<!-- ❌ No focus state -->
<button class="bg-blue-600 text-white px-4 py-2 rounded
focus:outline-none">
  Bad for keyboard users
```

```
</button>

<!-- ✅ Visible focus state -->
<button class="bg-blue-600 text-white px-4 py-2 rounded
focus:outline-none focus:ring-4 focus:ring-blue-300">
  Accessible
</button>
```

Key Takeaways

✅ **Organize classes** logically: layout, sizing, typography, visual, interactive
✅ Use **Prettier plugin** for automatic class sorting
✅ **Avoid class bloat**: don't over-specify, use shorthand
✅ **Purging works automatically** if `content` is configured correctly
✅ Always provide **focus states** for accessibility
✅ Check **color contrast** (4.5:1 minimum for AA)
✅ Use **semantic HTML** (buttons, headings, etc.)
✅ **Extract to components** when used 5+ times
✅ Use **custom CSS** for complex animations and specific needs
✅ **Mobile-first** approach with responsive utilities

Practice Assignments

Assignment 1 (Easy): Class Organization

Take a complex component with 20+ utility classes and:

1. Reorganize into logical groups (layout, sizing, typography, etc.)
2. Format with line breaks for readability
3. Remove any redundant classes

4. Add comments explaining each group

Assignment 2 (Easy): Bundle Size Audit

1. Build a project with Tailwind
2. Check the CSS file size
3. Identify any unused utilities (use Chrome DevTools Coverage)
4. Optimize the `content` configuration
5. Document the before/after sizes

Assignment 3 (Medium): Accessibility Audit

Audit a landing page for accessibility:

1. Check all focus states (keyboard navigation)
2. Test color contrast ratios
3. Verify semantic HTML usage
4. Add `sr-only` labels where needed
5. Test with a screen reader (NVDA, JAWS, or VoiceOver)
6. Document and fix all issues

Assignment 4 (Medium): Component Extraction

Take a project with repeated patterns:

1. Identify components used 5+ times
2. Extract to React/Vue components with variants
3. Replace all instances with the component
4. Document props/variants
5. Compare class count before/after

Assignment 5 (Hard): Performance Optimization

Optimize a Tailwind project:

1. **Audit** current bundle size
2. **Configure** proper content paths
3. **Remove** unused plugins
4. **Optimize** images and assets
5. **Enable** compression in build config
6. **Implement** lazy loading for large components
7. **Document** all optimizations with before/after metrics

Target metrics:

- CSS: <50KB gzipped
- First Contentful Paint: <1.5s
- Lighthouse Performance: 90+

Deliverables:

- Performance report (before/after)
- Optimized codebase
- Documentation of changes
- Accessibility score of 100

Ready for **Module 12: Real-World Projects**? Let me know!

1/7/2026, 12:27:03 PM

MODULE 12: REAL-WORLD PROJECTS

In this final module, we'll build complete, production-ready components and pages. Each project will combine everything you've learned.

Project 1: Complete Responsive Landing Page

Let's build a full landing page with hero section, features, pricing, testimonials, and footer.

```
<!DOCTYPE html>
<html lang="en" class="scroll-smooth">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>SaaS Landing Page</title>
  <script src="https://cdn.tailwindcss.com"></script>
</head>
<body class="bg-gray-50">

  <!-- Navigation -->
  <nav class="bg-white shadow-sm sticky top-0 z-50">
    <div class="max-w-7xl mx-auto px-4 sm:px-6 lg:px-8">
      <div class="flex justify-between items-center h-16">

        <!-- Logo -->
        <div class="flex-shrink-0">
          <h1 class="text-2xl font-bold text-blue-
600">SaaSify</h1>
        </div>

        <!-- Desktop Navigation -->
        <div class="hidden md:flex space-x-8">
          <a href="#features" class="text-gray-700 hover:text-
```

```
blue-600 transition">Features</a>
          <a href="#pricing" class="text-gray-700 hover:text-
blue-600 transition">Pricing</a>
          <a href="#testimonials" class="text-gray-700
hover:text-blue-600 transition">Testimonials</a>
          <a href="#contact" class="text-gray-700 hover:text-
blue-600 transition">Contact</a>
        </div>

        <!-- CTA Buttons -->
        <div class="hidden md:flex items-center space-x-4">
          <button class="text-gray-700 hover:text-blue-600 font-
medium transition">
            Sign In
          </button>
          <button class="bg-blue-600 hover:bg-blue-700 text-white
px-6 py-2 rounded-lg font-medium transition">
            Get Started
          </button>
        </div>

        <!-- Mobile Menu Button -->
        <button id="mobile-menu-btn" class="md:hidden p-2 text-
gray-700 hover:text-blue-600">
          <svg class="w-6 h-6" fill="none" stroke="currentColor"
viewBox="0 0 24 24">
            <path stroke-linecap="round" stroke-linejoin="round"
stroke-width="2" d="M4 6h16M4 12h16M4 18h16"></path>
          </svg>
        </button>

      </div>
    </div>

    <!-- Mobile Menu (hidden by default) -->
```

```
    <div id="mobile-menu" class="hidden md:hidden border-t
border-gray-200">
      <div class="px-4 py-3 space-y-3">
        <a href="#features" class="block text-gray-700
hover:text-blue-600 py-2">Features</a>
        <a href="#pricing" class="block text-gray-700 hover:text-
blue-600 py-2">Pricing</a>
        <a href="#testimonials" class="block text-gray-700
hover:text-blue-600 py-2">Testimonials</a>
        <a href="#contact" class="block text-gray-700 hover:text-
blue-600 py-2">Contact</a>
        <button class="w-full text-left text-gray-700 hover:text-
blue-600 py-2 font-medium">
          Sign In
        </button>
        <button class="w-full bg-blue-600 hover:bg-blue-700 text-
white py-2 rounded-lg font-medium">
          Get Started
        </button>
      </div>
    </div>
  </nav>

  <!-- Hero Section -->
  <section class="relative bg-gradient-to-br from-blue-50 to-
indigo-100 py-20 md:py-32 overflow-hidden">
    <!-- Background decoration -->
    <div class="absolute top-0 left-0 w-64 h-64 bg-blue-200
rounded-full mix-blend-multiply filter blur-3xl opacity-70
animate-blob"></div>
    <div class="absolute top-0 right-0 w-64 h-64 bg-purple-200
rounded-full mix-blend-multiply filter blur-3xl opacity-70
animate-blob animation-delay-2000"></div>
    <div class="absolute bottom-0 left-1/2 w-64 h-64 bg-pink-200
rounded-full mix-blend-multiply filter blur-3xl opacity-70
```

```
animate-blob animation-delay-4000"></div>

    <div class="relative max-w-7xl mx-auto px-4 sm:px-6 lg:px-8">
      <div class="flex flex-col lg:flex-row items-center gap-12">

        <!-- Hero Text -->
        <div class="flex-1 text-center lg:text-left">
          <h1 class="text-4xl sm:text-5xl md:text-6xl font-bold
text-gray-900 mb-6 leading-tight">
            Build Your SaaS Product
            <span class="text-blue-600">Faster Than Ever</span>
          </h1>
          <p class="text-lg sm:text-xl text-gray-600 mb-8 max-w-
2xl mx-auto lg:mx-0">
            Launch your startup in days, not months. Everything
you need to build, ship, and scale your SaaS application.
          </p>

          <!-- CTA Buttons -->
          <div class="flex flex-col sm:flex-row gap-4 justify-
center lg:justify-start">
            <button class="bg-blue-600 hover:bg-blue-700 text-
white px-8 py-4 rounded-lg font-semibold shadow-lg hover:shadow-
xl transform hover:scale-105 transition-all">
              Start Free Trial
            </button>
            <button class="bg-white hover:bg-gray-50 text-gray-
900 px-8 py-4 rounded-lg font-semibold border-2 border-gray-200
transition">
              Watch Demo
            </button>
          </div>

          <!-- Social Proof -->
          <div class="mt-8 flex items-center justify-center
```

```
lg:justify-start space-x-2 text-sm text-gray-600">
            <div class="flex -space-x-2">
              <img src="https://i.pravatar.cc/40?img=1" class="w-
8 h-8 rounded-full border-2 border-white" alt="User">
              <img src="https://i.pravatar.cc/40?img=2" class="w-
8 h-8 rounded-full border-2 border-white" alt="User">
              <img src="https://i.pravatar.cc/40?img=3" class="w-
8 h-8 rounded-full border-2 border-white" alt="User">
            </div>
            <span>Join 10,000+ developers building with us</span>
          </div>
        </div>

        <!-- Hero Image/Graphic -->
        <div class="flex-1 w-full max-w-lg">
          <div class="relative">
            <div class="bg-white rounded-2xl shadow-2xl p-8
transform hover:rotate-1 transition-transform">
              <div class="space-y-4">
                <div class="h-4 bg-gray-200 rounded w-3/4
animate-pulse"></div>
                <div class="h-4 bg-gray-200 rounded w-full
animate-pulse"></div>
                <div class="h-4 bg-gray-200 rounded w-5/6
animate-pulse"></div>
                <div class="h-32 bg-gradient-to-br from-blue-400
to-purple-500 rounded-lg"></div>
                <div class="flex space-x-2">
                  <div class="h-10 bg-blue-500 rounded flex-1">
</div>
                  <div class="h-10 bg-gray-200 rounded flex-1">
</div>
                </div>
              </div>
            </div>
```

```
            <!-- Floating badge -->
            <div class="absolute -top-4 -right-4 bg-green-500
text-white px-4 py-2 rounded-full shadow-lg font-semibold text-
sm">
              ✓ Verified
            </div>
          </div>
        </div>

      </div>
    </div>
  </section>

  <!-- Features Section -->
  <section id="features" class="py-20 bg-white">
    <div class="max-w-7xl mx-auto px-4 sm:px-6 lg:px-8">

      <!-- Section Header -->
      <div class="text-center mb-16">
        <h2 class="text-3xl md:text-4xl font-bold text-gray-900
mb-4">
          Everything You Need to Succeed
        </h2>
        <p class="text-xl text-gray-600 max-w-2xl mx-auto">
          Powerful features designed to help you build, launch,
and scale your business
        </p>
      </div>

      <!-- Features Grid -->
      <div class="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3
gap-8">

        <!-- Feature 1 -->
        <div class="group p-8 rounded-xl border-2 border-gray-100
```

```
hover:border-blue-500 hover:shadow-xl transition-all">
          <div class="w-12 h-12 bg-blue-100 rounded-lg flex
items-center justify-center mb-4 group-hover:bg-blue-600
transition-colors">
            <svg class="w-6 h-6 text-blue-600 group-hover:text-
white" fill="none" stroke="currentColor" viewBox="0 0 24 24">
              <path stroke-linecap="round" stroke-
linejoin="round" stroke-width="2" d="M13 10V3L4 14h7v7l9-11h-7z">
</path>
            </svg>
          </div>
          <h3 class="text-xl font-bold text-gray-900 mb-
2">Lightning Fast</h3>
          <p class="text-gray-600">Build and deploy in minutes
with our optimized infrastructure and developer tools.</p>
        </div>

        <!-- Feature 2 -->
        <div class="group p-8 rounded-xl border-2 border-gray-100
hover:border-blue-500 hover:shadow-xl transition-all">
          <div class="w-12 h-12 bg-green-100 rounded-lg flex
items-center justify-center mb-4 group-hover:bg-green-600
transition-colors">
            <svg class="w-6 h-6 text-green-600 group-hover:text-
white" fill="none" stroke="currentColor" viewBox="0 0 24 24">
              <path stroke-linecap="round" stroke-
linejoin="round" stroke-width="2" d="M9 12l2 2 4-4m5.618-
4.016A11.955 11.955 0 0112 2.944a11.955 11.955 0 01-8.618
3.04A12.02 12.02 0 003 9c0 5.591 3.824 10.29 9 11.622 5.176-1.332
9-6.03 9-11.622 0-1.042-.133-2.052-.382-3.016z"></path>
            </svg>
          </div>
          <h3 class="text-xl font-bold text-gray-900 mb-2">Secure
by Default</h3>
          <p class="text-gray-600">Enterprise-grade security with
```

```
automatic updates and 99.9% uptime guarantee.</p>
        </div>

        <!-- Feature 3 -->
        <div class="group p-8 rounded-xl border-2 border-gray-100
hover:border-blue-500 hover:shadow-xl transition-all">
          <div class="w-12 h-12 bg-purple-100 rounded-lg flex
items-center justify-center mb-4 group-hover:bg-purple-600
transition-colors">
            <svg class="w-6 h-6 text-purple-600 group-hover:text-
white" fill="none" stroke="currentColor" viewBox="0 0 24 24">
              <path stroke-linecap="round" stroke-
linejoin="round" stroke-width="2" d="M7 21a4 4 0 01-4-4V5a2 2 0
012-2h4a2 2 0 012 2v12a4 4 0 01-4 4zm0 0h12a2 2 0 002-2v-4a2 2 0
00-2-2h-2.343M11 7.343l1.657-1.657a2 2 0 012.828 0l2.829 2.829a2
2 0 010 2.828l-8.486 8.485M7 17h.01"></path>
            </svg>
          </div>
          <h3 class="text-xl font-bold text-gray-900 mb-2">Fully
Customizable</h3>
          <p class="text-gray-600">Tailor every aspect to match
your brand and workflow requirements.</p>
        </div>

        <!-- Feature 4 -->
        <div class="group p-8 rounded-xl border-2 border-gray-100
hover:border-blue-500 hover:shadow-xl transition-all">
          <div class="w-12 h-12 bg-yellow-100 rounded-lg flex
items-center justify-center mb-4 group-hover:bg-yellow-600
transition-colors">
            <svg class="w-6 h-6 text-yellow-600 group-hover:text-
white" fill="none" stroke="currentColor" viewBox="0 0 24 24">
              <path stroke-linecap="round" stroke-
linejoin="round" stroke-width="2" d="M9 19v-6a2 2 0 00-2-2H5a2 2
0 00-2 2v6a2 2 0 002 2h2a2 2 0 002-2zm0 0V9a2 2 0 012-2h2a2 2 0
```

```
012 2v10m-6 0a2 2 0 002 2h2a2 2 0 002-2m0 0V5a2 2 0 012-2h2a2 2 0
012 2v14a2 2 0 01-2 2h-2a2 2 0 01-2-2z"></path>
            </svg>
          </div>
          <h3 class="text-xl font-bold text-gray-900 mb-
2">Advanced Analytics</h3>
          <p class="text-gray-600">Track performance with real-
time insights and detailed reporting.</p>
        </div>

        <!-- Feature 5 -->
        <div class="group p-8 rounded-xl border-2 border-gray-100
hover:border-blue-500 hover:shadow-xl transition-all">
          <div class="w-12 h-12 bg-red-100 rounded-lg flex items-
center justify-center mb-4 group-hover:bg-red-600 transition-
colors">
            <svg class="w-6 h-6 text-red-600 group-hover:text-
white" fill="none" stroke="currentColor" viewBox="0 0 24 24">
              <path stroke-linecap="round" stroke-
linejoin="round" stroke-width="2" d="M12 4.354a4 4 0 110 5.292M15
21H3v-1a6 6 0 0112 0v1zm0 0h6v-1a6 6 0 00-9-5.197M13 7a4 4 0 11-8
0 4 4 0 018 0z"></path>
            </svg>
          </div>
          <h3 class="text-xl font-bold text-gray-900 mb-2">Team
Collaboration</h3>
          <p class="text-gray-600">Work together seamlessly with
built-in collaboration tools.</p>
        </div>

        <!-- Feature 6 -->
        <div class="group p-8 rounded-xl border-2 border-gray-100
hover:border-blue-500 hover:shadow-xl transition-all">
          <div class="w-12 h-12 bg-indigo-100 rounded-lg flex
items-center justify-center mb-4 group-hover:bg-indigo-600
```

```
transition-colors">
            <svg class="w-6 h-6 text-indigo-600 group-hover:text-
white" fill="none" stroke="currentColor" viewBox="0 0 24 24">
              <path stroke-linecap="round" stroke-
linejoin="round" stroke-width="2" d="M18.364 5.636l-3.536 3.536m0
5.656l3.536 3.536M9.172 9.172L5.636 5.636m3.536 9.192l-3.536
3.536M21 12a9 9 0 11-18 0 9 9 0 0118 0zm-5 0a4 4 0 11-8 0 4 4 0
018 0z"></path>
            </svg>
          </div>
          <h3 class="text-xl font-bold text-gray-900 mb-2">24/7
Support</h3>
          <p class="text-gray-600">Get help whenever you need it
from our dedicated support team.</p>
        </div>

      </div>
    </div>
  </section>

  <!-- Pricing Section -->
  <section id="pricing" class="py-20 bg-gray-50">
    <div class="max-w-7xl mx-auto px-4 sm:px-6 lg:px-8">

      <!-- Section Header -->
      <div class="text-center mb-16">
        <h2 class="text-3xl md:text-4xl font-bold text-gray-900
mb-4">
          Simple, Transparent Pricing
        </h2>
        <p class="text-xl text-gray-600">
          Choose the perfect plan for your needs
        </p>
      </div>
```

```
      <!-- Pricing Cards -->
      <div class="grid grid-cols-1 md:grid-cols-3 gap-8 max-w-6xl
mx-auto">

        <!-- Starter Plan -->
        <div class="bg-white rounded-2xl shadow-lg p-8 border-2
border-gray-100 hover:border-blue-500 transition-all">
          <h3 class="text-2xl font-bold text-gray-900 mb-
2">Starter</h3>
          <p class="text-gray-600 mb-6">Perfect for trying out
our platform</p>
          <div class="mb-6">
            <span class="text-5xl font-bold text-gray-
900">$29</span>
            <span class="text-gray-600">/month</span>
          </div>
          <ul class="space-y-4 mb-8">
            <li class="flex items-center text-gray-700">
              <svg class="w-5 h-5 text-green-500 mr-3"
fill="currentColor" viewBox="0 0 20 20">
                <path fill-rule="evenodd" d="M16.707 5.293a1 1 0
010 1.414l-8 8a1 1 0 01-1.414 0l-4-4a1 1 0 011.414-1.414L8
12.586l7.293-7.293a1 1 0 011.414 0z" clip-rule="evenodd"></path>
              </svg>
              Up to 3 projects
            </li>
            <li class="flex items-center text-gray-700">
              <svg class="w-5 h-5 text-green-500 mr-3"
fill="currentColor" viewBox="0 0 20 20">
                <path fill-rule="evenodd" d="M16.707 5.293a1 1 0
010 1.414l-8 8a1 1 0 01-1.414 0l-4-4a1 1 0 011.414-1.414L8
12.586l7.293-7.293a1 1 0 011.414 0z" clip-rule="evenodd"></path>
              </svg>
              5 GB storage
            </li>
```

```
            <li class="flex items-center text-gray-700">
              <svg class="w-5 h-5 text-green-500 mr-3"
fill="currentColor" viewBox="0 0 20 20">
                <path fill-rule="evenodd" d="M16.707 5.293a1 1 0
010 1.414l-8 8a1 1 0 01-1.414 0l-4-4a1 1 0 011.414-1.414L8
12.586l7.293-7.293a1 1 0 011.414 0z" clip-rule="evenodd"></path>
              </svg>
              Basic support
            </li>
            <li class="flex items-center text-gray-700">
              <svg class="w-5 h-5 text-green-500 mr-3"
fill="currentColor" viewBox="0 0 20 20">
                <path fill-rule="evenodd" d="M16.707 5.293a1 1 0
010 1.414l-8 8a1 1 0 01-1.414 0l-4-4a1 1 0 011.414-1.414L8
12.586l7.293-7.293a1 1 0 011.414 0z" clip-rule="evenodd"></path>
              </svg>
              Community access
            </li>
          </ul>
          <button class="w-full bg-gray-900 hover:bg-gray-800
text-white py-3 rounded-lg font-semibold transition">
            Get Started
          </button>
        </div>

        <!-- Professional Plan (Featured) -->
        <div class="bg-blue-600 rounded-2xl shadow-2xl p-8
border-2 border-blue-600 transform md:scale-105 relative">
          <div class="absolute top-0 right-0 bg-yellow-400 text-
gray-900 px-4 py-1 rounded-bl-lg rounded-tr-2xl text-sm font-
bold">
            POPULAR
          </div>
          <h3 class="text-2xl font-bold text-white mb-
2">Professional</h3>
```

```
          <p class="text-blue-100 mb-6">For growing
businesses</p>
          <div class="mb-6">
            <span class="text-5xl font-bold text-
white">$99</span>
            <span class="text-blue-100">/month</span>
          </div>
          <ul class="space-y-4 mb-8">
            <li class="flex items-center text-white">
              <svg class="w-5 h-5 text-yellow-400 mr-3"
fill="currentColor" viewBox="0 0 20 20">
                <path fill-rule="evenodd" d="M16.707 5.293a1 1 0
010 1.414l-8 8a1 1 0 01-1.414 0l-4-4a1 1 0 011.414-1.414L8
12.586l7.293-7.293a1 1 0 011.414 0z" clip-rule="evenodd"></path>
              </svg>
              Unlimited projects
            </li>
            <li class="flex items-center text-white">
              <svg class="w-5 h-5 text-yellow-400 mr-3"
fill="currentColor" viewBox="0 0 20 20">
                <path fill-rule="evenodd" d="M16.707 5.293a1 1 0
010 1.414l-8 8a1 1 0 01-1.414 0l-4-4a1 1 0 011.414-1.414L8
12.586l7.293-7.293a1 1 0 011.414 0z" clip-rule="evenodd"></path>
              </svg>
              100 GB storage
            </li>
            <li class="flex items-center text-white">
              <svg class="w-5 h-5 text-yellow-400 mr-3"
fill="currentColor" viewBox="0 0 20 20">
                <path fill-rule="evenodd" d="M16.707 5.293a1 1 0
010 1.414l-8 8a1 1 0 01-1.414 0l-4-4a1 1 0 011.414-1.414L8
12.586l7.293-7.293a1 1 0 011.414 0z" clip-rule="evenodd"></path>
              </svg>
              Priority support
            </li>
```

```
            <li class="flex items-center text-white">
              <svg class="w-5 h-5 text-yellow-400 mr-3"
fill="currentColor" viewBox="0 0 20 20">
                <path fill-rule="evenodd" d="M16.707 5.293a1 1 0
010 1.414l-8 8a1 1 0 01-1.414 0l-4-4a1 1 0 011.414-1.414L8
12.586l7.293-7.293a1 1 0 011.414 0z" clip-rule="evenodd"></path>
              </svg>
              Advanced analytics
            </li>
            <li class="flex items-center text-white">
              <svg class="w-5 h-5 text-yellow-400 mr-3"
fill="currentColor" viewBox="0 0 20 20">
                <path fill-rule="evenodd" d="M16.707 5.293a1 1 0
010 1.414l-8 8a1 1 0 01-1.414 0l-4-4a1 1 0 011.414-1.414L8
12.586l7.293-7.293a1 1 0 011.414 0z" clip-rule="evenodd"></path>
              </svg>
              API access
            </li>
          </ul>
          <button class="w-full bg-white hover:bg-gray-50 text-
blue-600 py-3 rounded-lg font-semibold transition">
            Get Started
          </button>
        </div>

        <!-- Enterprise Plan -->
        <div class="bg-white rounded-2xl shadow-lg p-8 border-2
border-gray-100 hover:border-blue-500 transition-all">
          <h3 class="text-2xl font-bold text-gray-900 mb-
2">Enterprise</h3>
          <p class="text-gray-600 mb-6">For large
organizations</p>
          <div class="mb-6">
            <span class="text-5xl font-bold text-gray-
900">$299</span>
```

```
            <span class="text-gray-600">/month</span>
          </div>
          <ul class="space-y-4 mb-8">
            <li class="flex items-center text-gray-700">
              <svg class="w-5 h-5 text-green-500 mr-3"
fill="currentColor" viewBox="0 0 20 20">
                <path fill-rule="evenodd" d="M16.707 5.293a1 1 0
010 1.414l-8 8a1 1 0 01-1.414 0l-4-4a1 1 0 011.414-1.414L8
12.586l7.293-7.293a1 1 0 011.414 0z" clip-rule="evenodd"></path>
              </svg>
              Unlimited everything
            </li>
            <li class="flex items-center text-gray-700">
              <svg class="w-5 h-5 text-green-500 mr-3"
fill="currentColor" viewBox="0 0 20 20">
                <path fill-rule="evenodd" d="M16.707 5.293a1 1 0
010 1.414l-8 8a1 1 0 01-1.414 0l-4-4a1 1 0 011.414-1.414L8
12.586l7.293-7.293a1 1 0 011.414 0z" clip-rule="evenodd"></path>
              </svg>
              1 TB storage
            </li>
            <li class="flex items-center text-gray-700">
              <svg class="w-5 h-5 text-green-500 mr-3"
fill="currentColor" viewBox="0 0 20 20">
                <path fill-rule="evenodd" d="M16.707 5.293a1 1 0
010 1.414l-8 8a1 1 0 01-1.414 0l-4-4a1 1 0 011.414-1.414L8
12.586l7.293-7.293a1 1 0 011.414 0z" clip-rule="evenodd"></path>
              </svg>
              24/7 dedicated support
            </li>
            <li class="flex items-center text-gray-700">
              <svg class="w-5 h-5 text-green-500 mr-3"
fill="currentColor" viewBox="0 0 20 20">
                <path fill-rule="evenodd" d="M16.707 5.293a1 1 0
010 1.414l-8 8a1 1 0 01-1.414 0l-4-4a1 1 0 011.414-1.414L8
```

```
12.586l7.293-7.293a1 1 0 011.414 0z" clip-rule="evenodd"></path>
              </svg>
              Custom integrations
            </li>
            <li class="flex items-center text-gray-700">
              <svg class="w-5 h-5 text-green-500 mr-3"
fill="currentColor" viewBox="0 0 20 20">
                <path fill-rule="evenodd" d="M16.707 5.293a1 1 0
010 1.414l-8 8a1 1 0 01-1.414 0l-4-4a1 1 0 011.414-1.414L8
12.586l7.293-7.293a1 1 0 011.414 0z" clip-rule="evenodd"></path>
              </svg>
              SLA guarantee
            </li>
          </ul>
          <button class="w-full bg-gray-900 hover:bg-gray-800
text-white py-3 rounded-lg font-semibold transition">
            Contact Sales
          </button>
        </div>

      </div>
    </div>
  </section>

  <!-- Testimonials Section -->
  <section id="testimonials" class="py-20 bg-white">
    <div class="max-w-7xl mx-auto px-4 sm:px-6 lg:px-8">

      <!-- Section Header -->
      <div class="text-center mb-16">
        <h2 class="text-3xl md:text-4xl font-bold text-gray-900
mb-4">
          Loved by Developers Worldwide
        </h2>
        <p class="text-xl text-gray-600">
```

```
          See what our customers have to say
        </p>
      </div>

      <!-- Testimonials Grid -->
      <div class="grid grid-cols-1 md:grid-cols-3 gap-8">

        <!-- Testimonial 1 -->
        <div class="bg-gray-50 rounded-xl p-8">
          <div class="flex items-center mb-4">
            <div class="text-yellow-400 flex">
              <svg class="w-5 h-5" fill="currentColor" viewBox="0
0 20 20"><path d="M9.049 2.927c.3-.921 1.603-.921 1.902 0l1.07
3.292a1 1 0 00.95.69h3.462c.969 0 1.371 1.24.588 1.81l-2.8
2.034a1 1 0 00-.364 1.118l1.07 3.292c.3.921-.755 1.688-1.54
1.118l-2.8-2.034a1 1 0 00-1.175 0l-2.8 2.034c-.784.57-1.838-.197-
1.539-1.118l1.07-3.292a1 1 0 00-.364-1.118L2.98
8.72c-.783-.57-.38-1.81.588-1.81h3.461a1 1 0 00.951-.69l1.07-
3.292z"></path></svg>
              <svg class="w-5 h-5" fill="currentColor" viewBox="0
0 20 20"><path d="M9.049 2.927c.3-.921 1.603-.921 1.902 0l1.07
3.292a1 1 0 00.95.69h3.462c.969 0 1.371 1.24.588 1.81l-2.8
2.034a1 1 0 00-.364 1.118l1.07 3.292c.3.921-.755 1.688-1.54
1.118l-2.8-2.034a1 1 0 00-1.175 0l-2.8 2.034c-.784.57-1.838-.197-
1.539-1.118l1.07-3.292a1 1 0 00-.364-1.118L2.98
8.72c-.783-.57-.38-1.81.588-1.81h3.461a1 1 0 00.951-.69l1.07-
3.292z"></path></svg>
              <svg class="w-5 h-5" fill="currentColor" viewBox="0
0 20 20"><path d="M9.049 2.927c.3-.921 1.603-.921 1.902 0l1.07
3.292a1 1 0 00.95.69h3.462c.969 0 1.371 1.24.588 1.81l-2.8
2.034a1 1 0 00-.364 1.118l1.07 3.292c.3.921-.755 1.688-1.54
1.118l-2.8-2.034a1 1 0 00-1.175 0l-2.8 2.034c-.784.57-1.838-.197-
1.539-1.118l1.07-3.292a1 1 0 00-.364-1.118L2.98
8.72c-.783-.57-.38-1.81.588-1.81h3.461a1 1 0 00.951-.69l1.07-
3.292z"></path></svg>
```

```
              <svg class="w-5 h-5" fill="currentColor" viewBox="0
0 20 20"><path d="M9.049 2.927c.3-.921 1.603-.921 1.902 0l1.07
3.292a1 1 0 00.95.69h3.462c.969 0 1.371 1.24.588 1.81l-2.8
2.034a1 1 0 00-.364 1.118l1.07 3.292c.3.921-.755 1.688-1.54
1.118l-2.8-2.034a1 1 0 00-1.175 0l-2.8 2.034c-.784.57-1.838-.197-
1.539-1.118l1.07-3.292a1 1 0 00-.364-1.118L2.98
8.72c-.783-.57-.38-1.81.588-1.81h3.461a1 1 0 00.951-.69l1.07-
3.292z"></path></svg>
              <svg class="w-5 h-5" fill="currentColor" viewBox="0
0 20 20"><path d="M9.049 2.927c.3-.921 1.603-.921 1.902 0l1.07
3.292a1 1 0 00.95.69h3.462c.969 0 1.371 1.24.588 1.81l-2.8
2.034a1 1 0 00-.364 1.118l1.07 3.292c.3.921-.755 1.688-1.54
1.118l-2.8-2.034a1 1 0 00-1.175 0l-2.8 2.034c-.784.57-1.838-.197-
1.539-1.118l1.07-3.292a1 1 0 00-.364-1.118L2.98
8.72c-.783-.57-.38-1.81.588-1.81h3.461a1 1 0 00.951-.69l1.07-
3.292z"></path></svg>
            </div>
          </div>
          <p class="text-gray-700 mb-6">
            "This platform has completely transformed how we
build and ship products. The speed and reliability are
unmatched."
          </p>
          <div class="flex items-center">
            <img src="https://i.pravatar.cc/50?img=5" class="w-12
h-12 rounded-full mr-4" alt="Sarah">
            <div>
              <div class="font-semibold text-gray-900">Sarah
Johnson</div>
              <div class="text-sm text-gray-600">CTO,
TechCorp</div>
            </div>
          </div>
        </div>
```

```
        <!-- Testimonial 2 -->
        <div class="bg-gray-50 rounded-xl p-8">
          <div class="flex items-center mb-4">
            <div class="text-yellow-400 flex">
              <svg class="w-5 h-5" fill="currentColor" viewBox="0
0 20 20"><path d="M9.049 2.927c.3-.921 1.603-.921 1.902 0l1.07
3.292a1 1 0 00.95.69h3.462c.969 0 1.371 1.24.588 1.81l-2.8
2.034a1 1 0 00-.364 1.118l1.07 3.292c.3.921-.755 1.688-1.54
1.118l-2.8-2.034a1 1 0 00-1.175 0l-2.8 2.034c-.784.57-1.838-.197-
1.539-1.118l1.07-3.292a1 1 0 00-.364-1.118L2.98
8.72c-.783-.57-.38-1.81.588-1.81h3.461a1 1 0 00.951-.69l1.07-
3.292z"></path></svg>
              <svg class="w-5 h-5" fill="currentColor" viewBox="0
0 20 20"><path d="M9.049 2.927c.3-.921 1.603-.921 1.902 0l1.07
3.292a1 1 0 00.95.69h3.462c.969 0 1.371 1.24.588 1.81l-2.8
2.034a1 1 0 00-.364 1.118l1.07 3.292c.3.921-.755 1.688-1.54
1.118l-2.8-2.034a1 1 0 00-1.175 0l-2.8 2.034c-.784.57-1.838-.197-
1.539-1.118l1.07-3.292a1 1 0 00-.364-1.118L2.98
8.72c-.783-.57-.38-1.81.588-1.81h3.461a1 1 0 00.951-.69l1.07-
3.292z"></path></svg>
              <svg class="w-5 h-5" fill="currentColor" viewBox="0
0 20 20"><path d="M9.049 2.927c.3-.921 1.603-.921 1.902 0l1.07
3.292a1 1 0 00.95.69h3.462c.969 0 1.371 1.24.588 1.81l-2.8
2.034a1 1 0 00-.364 1.118l1.07 3.292c.3.921-.755 1.688-1.54
1.118l-2.8-2.034a1 1 0 00-1.175 0l-2.8 2.034c-.784.57-1.838-.197-
1.539-1.118l1.07-3.292a1 1 0 00-.364-1.118L2.98
8.72c-.783-.57-.38-1.81.588-1.81h3.461a1 1 0 00.951-.69l1.07-
3.292z"></path></svg>
              <svg class="w-5 h-5" fill="currentColor" viewBox="0
0 20 20"><path d="M9.049 2.927c.3-.921 1.603-.921 1.902 0l1.07
3.292a1 1 0 00.95.69h3.462c.969 0 1.371 1.24.588 1.81l-2.8
2.034a1 1 0 00-.364 1.118l1.07 3.292c.3.921-.755 1.688-1.54
1.118l-2.8-2.034a1 1 0 00-1.175 0l-2.8 2.034c-.784.57-1.838-.197-
1.539-1.118l1.07-3.292a1 1 0 00-.364-1.118L2.98
8.72c-.783-.57-.38-1.81.588-1.81h3.461a1 1 0 00.951-.69l1.07-
```

```
3.292z"></path></svg>
              <svg class="w-5 h-5" fill="currentColor" viewBox="0
0 20 20"><path d="M9.049 2.927c.3-.921 1.603-.921 1.902 0l1.07
3.292a1 1 0 00.95.69h3.462c.969 0 1.371 1.24.588 1.81l-2.8
2.034a1 1 0 00-.364 1.118l1.07 3.292c.3.921-.755 1.688-1.54
1.118l-2.8-2.034a1 1 0 00-1.175 0l-2.8 2.034c-.784.57-1.838-.197-
1.539-1.118l1.07-3.292a1 1 0 00-.364-1.118L2.98
8.72c-.783-.57-.38-1.81.588-1.81h3.461a1 1 0 00.951-.69l1.07-
3.292z"></path></svg>
            </div>
          </div>
          <p class="text-gray-700 mb-6">
            "Best investment we've made. The ROI was immediate
and the support team is incredibly responsive."
          </p>
          <div class="flex items-center">
            <img src="https://i.pravatar.cc/50?img=8" class="w-12
h-12 rounded-full mr-4" alt="Mike">
            <div>
              <div class="font-semibold text-gray-900">Mike
Chen</div>
              <div class="text-sm text-gray-600">Founder,
StartupXYZ</div>
            </div>
          </div>
        </div>

        <!-- Testimonial 3 -->
        <div class="bg-gray-50 rounded-xl p-8">
          <div class="flex items-center mb-4">
            <div class="text-yellow-400 flex">
              <svg class="w-5 h-5" fill="currentColor" viewBox="0
0 20 20"><path d="M9.049 2.927c.3-.921 1.603-.921 1.902 0l1.07
3.292a1 1 0 00.95.69h3.462c.969 0 1.371 1.24.588 1.81l-2.8
2.034a1 1 0 00-.364 1.118l1.07 3.292c.3.921-.755 1.688-1.54
```

```
1.118l-2.8-2.034a1 1 0 00-1.175 0l-2.8 2.034c-.784.57-1.838-.197-
1.539-1.118l1.07-3.292a1 1 0 00-.364-1.118L2.98
8.72c-.783-.57-.38-1.81.588-1.81h3.461a1 1 0 00.951-.69l1.07-
3.292z"></path></svg>
              <svg class="w-5 h-5" fill="currentColor" viewBox="0
0 20 20"><path d="M9.049 2.927c.3-.921 1.603-.921 1.902 0l1.07
3.292a1 1 0 00.95.69h3.462c.969 0 1.371 1.24.588 1.81l-2.8
2.034a1 1 0 00-.364 1.118l1.07 3.292c.3.921-.755 1.688-1.54
1.118l-2.8-2.034a1 1 0 00-1.175 0l-2.8 2.034c-.784.57-1.838-.197-
1.539-1.118l1.07-3.292a1 1 0 00-.364-1.118L2.98
8.72c-.783-.57-.38-1.81.588-1.81h3.461a1 1 0 00.951-.69l1.07-
3.292z"></path></svg>
              <svg class="w-5 h-5" fill="currentColor" viewBox="0
0 20 20"><path d="M9.049 2.927c.3-.921 1.603-.921 1.902 0l1.07
3.292a1 1 0 00.95.69h3.462c.969 0 1.371 1.24.588 1.81l-2.8
2.034a1 1 0 00-.364 1.118l1.07 3.292c.3.921-.755 1.688-1.54
1.118l-2.8-2.034a1 1 0 00-1.175 0l-2.8 2.034c-.784.57-1.838-.197-
1.539-1.118l1.07-3.292a1 1 0 00-.364-1.118L2.98
8.72c-.783-.57-.38-1.81.588-1.81h3.461a1 1 0 00.951-.69l1.07-
3.292z"></path></svg>
              <svg class="w-5 h-5" fill="currentColor" viewBox="0
0 20 20"><path d="M9.049 2.927c.3-.921 1.603-.921 1.902 0l1.07
3.292a1 1 0 00.95.69h3.462c.969 0 1.371 1.24.588 1.81l-2.8
2.034a1 1 0 00-.364 1.118l1.07 3.292c.3.921-.755 1.688-1.54
1.118l-2.8-2.034a1 1 0 00-1.175 0l-2.8 2.034c-.784.57-1.838-.197-
1.539-1.118l1.07-3.292a1 1 0 00-.364-1.118L2.98
8.72c-.783-.57-.38-1.81.588-1.81h3.461a1 1 0 00.951-.69l1.07-
3.292z"></path></svg>
              <svg class="w-5 h-5" fill="currentColor" viewBox="0
0 20 20"><path d="M9.049 2.927c.3-.921 1.603-.921 1.902 0l1.07
3.292a1 1 0 00.95.69h3.462c.969 0 1.371 1.24.588 1.81l-2.8
2.034a1 1 0 00-.364 1.118l1.07 3.292c.3.921-.755 1.688-1.54
1.118l-2.8-2.034a1 1 0 00-1.175 0l-2.8 2.034c-.784.57-1.838-.197-
1.539-1.118l1.07-3.292a1 1 0 00-.364-1.118L2.98
8.72c-.783-.57-.38-1.81.588-1.81h3.461a1 1 0 00.951-.69l1.07-
```

```
3.292z"></path></svg>
            </div>
          </div>
          <p class="text-gray-700 mb-6">
            "Intuitive, powerful, and exactly what we needed. Our
development time has been cut in half."
          </p>
          <div class="flex items-center">
            <img src="https://i.pravatar.cc/50?img=9" class="w-12
h-12 rounded-full mr-4" alt="Emma">
            <div>
              <div class="font-semibold text-gray-900">Emma
Wilson</div>
              <div class="text-sm text-gray-600">Lead Developer,
DevShop</div>
            </div>
          </div>
        </div>

      </div>
    </div>
  </section>

  <!-- CTA Section -->
  <section class="py-20 bg-gradient-to-r from-blue-600 to-indigo-
600">
    <div class="max-w-4xl mx-auto px-4 sm:px-6 lg:px-8 text-
center">
      <h2 class="text-3xl md:text-4xl font-bold text-white mb-6">
        Ready to Get Started?
      </h2>
      <p class="text-xl text-blue-100 mb-8">
        Join thousands of developers building amazing products
      </p>
      <div class="flex flex-col sm:flex-row gap-4 justify-
```

```
center">
        <button class="bg-white hover:bg-gray-50 text-blue-600
px-8 py-4 rounded-lg font-semibold shadow-lg hover:shadow-xl
transition-all">
          Start Free Trial
        </button>
        <button class="bg-blue-700 hover:bg-blue-800 text-white
px-8 py-4 rounded-lg font-semibold border-2 border-blue-500
transition-all">
          Schedule Demo
        </button>
      </div>
    </div>
  </section>

  <!-- Footer -->
  <footer class="bg-gray-900 text-gray-300">
    <div class="max-w-7xl mx-auto px-4 sm:px-6 lg:px-8 py-12">
      <div class="grid grid-cols-2 md:grid-cols-4 gap-8 mb-8">

        <!-- Company -->
        <div>
          <h3 class="text-white font-semibold mb-4">Company</h3>
          <ul class="space-y-2">
            <li><a href="#" class="hover:text-white
transition">About</a></li>
            <li><a href="#" class="hover:text-white
transition">Blog</a></li>
            <li><a href="#" class="hover:text-white
transition">Careers</a></li>
            <li><a href="#" class="hover:text-white
transition">Press</a></li>
          </ul>
        </div>
```

```
        <!-- Product -->
        <div>
          <h3 class="text-white font-semibold mb-4">Product</h3>
          <ul class="space-y-2">
            <li><a href="#" class="hover:text-white
transition">Features</a></li>
            <li><a href="#" class="hover:text-white
transition">Pricing</a></li>
            <li><a href="#" class="hover:text-white
transition">Security</a></li>
            <li><a href="#" class="hover:text-white
transition">Roadmap</a></li>
          </ul>
        </div>

        <!-- Resources -->
        <div>
          <h3 class="text-white font-semibold mb-
4">Resources</h3>
          <ul class="space-y-2">
            <li><a href="#" class="hover:text-white
transition">Documentation</a></li>
            <li><a href="#" class="hover:text-white
transition">Guides</a></li>
            <li><a href="#" class="hover:text-white
transition">API Reference</a></li>
            <li><a href="#" class="hover:text-white
transition">Support</a></li>
          </ul>
        </div>

        <!-- Legal -->
        <div>
          <h3 class="text-white font-semibold mb-4">Legal</h3>
          <ul class="space-y-2">
```

```
            <li><a href="#" class="hover:text-white
transition">Privacy</a></li>
            <li><a href="#" class="hover:text-white
transition">Terms</a></li>
            <li><a href="#" class="hover:text-white
transition">Cookie Policy</a></li>
            <li><a href="#" class="hover:text-white
transition">Licenses</a></li>
          </ul>
        </div>

      </div>

      <!-- Bottom Footer -->
      <div class="border-t border-gray-800 pt-8 flex flex-col
md:flex-row justify-between items-center">
        <p class="text-sm mb-4 md:mb-0">
          © 2026 SaaSify. All rights reserved.
        </p>
        <div class="flex space-x-6">
          <a href="#" class="hover:text-white transition">
            <svg class="w-6 h-6" fill="currentColor" viewBox="0 0
24 24"><path d="M24 12.073c0-6.627-5.373-12-12-12s-12 5.373-12
12c0 5.99 4.388 10.954 10.125 11.854v-8.385H7.078v-
3.47h3.047V9.43c0-3.007 1.792-4.669 4.533-4.669 1.312 0 2.686.235
2.686.235v2.953H15.83c-1.491 0-1.956.925-1.956
1.874v2.25h3.328l-.532 3.47h-2.796v8.385C19.612 23.027 24 18.062
24 12.073z"/></svg>
          </a>
          <a href="#" class="hover:text-white transition">
            <svg class="w-6 h-6" fill="currentColor" viewBox="0 0
24 24"><path d="M23.953 4.57a10 10 0 01-2.825.775 4.958 4.958 0
002.163-2.723c-.951.555-2.005.959-3.127 1.184a4.92 4.92 0 00-
8.384 4.482C7.69 8.095 4.067 6.13 1.64 3.162a4.822 4.822 0
00-.666 2.475c0 1.71.87 3.213 2.188 4.096a4.904 4.904 0 01-
```

```
2.228-.616v.06a4.923 4.923 0 003.946 4.827 4.996 4.996 0 01-
2.212.085 4.936 4.936 0 004.604 3.417 9.867 9.867 0 01-6.102
2.105c-.39 0-.779-.023-1.17-.067a13.995 13.995 0 007.557
2.209c9.053 0 13.998-7.496 13.998-13.985 0-.21
0-.42-.015-.63A9.935 9.935 0 0024 4.59z"/></svg>
          </a>
          <a href="#" class="hover:text-white transition">
            <svg class="w-6 h-6" fill="currentColor" viewBox="0 0
24 24"><path d="M12 0C5.374 0 0 5.373 0 12c0 5.302 3.438 9.8
8.207 11.387.599.111.793-.261.793-.577v-2.234c-3.338.726-4.033-
1.416-4.033-1.416-.546-1.387-1.333-1.756-1.333-1.756-
1.089-.745.083-.729.083-.729 1.205.084 1.839 1.237 1.839 1.237
1.07 1.834 2.807 1.304 3.492.997.107-.775.418-1.305.762-1.604-
2.665-.305-5.467-1.334-5.467-5.931 0-1.311.469-2.381 1.236-
3.221-.124-.303-.535-1.524.117-3.176 0 0 1.008-.322 3.301
1.23A11.509 11.509 0 0112 5.803c1.02.005 2.047.138 3.006.404
2.291-1.552 3.297-1.23 3.297-1.23.653 1.653.242 2.874.118
3.176.77.84 1.235 1.911 1.235 3.221 0 4.609-2.807 5.624-5.479
5.921.43.372.823 1.102.823 2.222v3.293c0
.319.192.694.801.576C20.566 21.797 24 17.3 24 12c0-6.627-5.373-
12-12-12z"/></svg>
          </a>
        </div>
      </div>
    </div>
  </footer>

  <!-- JavaScript for Mobile Menu -->
  <script>
    const mobileMenuBtn = document.getElementById('mobile-menu-
btn');
    const mobileMenu = document.getElementById('mobile-menu');

    mobileMenuBtn.addEventListener('click', () => {
      mobileMenu.classList.toggle('hidden');
```

```
    });
  </script>

</body>
</html>
```

This landing page includes all the essential sections and demonstrates:

✅ Responsive navigation with mobile menu
✅ Hero section with gradient background
✅ Features grid with hover effects
✅ Pricing cards with featured plan
✅ Testimonials section
✅ CTA section
✅ Complete footer
✅ Smooth animations and transitions
✅ Mobile-first responsive design

Due to character limits, I'll provide the remaining projects (Dashboard, E-commerce, Blog, Navigation patterns, Forms) as separate responses. Would you like me to continue with **Project 2: Dashboard Layout** next?

1/7/2026, 12:45:17 PM

Project 2: Dashboard Layout

A complete admin dashboard with sidebar, top bar, stats, charts, and tables.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
```

```
scale=1.0">
  <title>Admin Dashboard</title>
  <script src="https://cdn.tailwindcss.com"></script>
</head>
<body class="bg-gray-100">

  <div class="flex h-screen overflow-hidden">

    <!-- Sidebar -->
    <aside id="sidebar" class="fixed inset-y-0 left-0 z-50 w-64
bg-gray-900 text-white transform -translate-x-full lg:translate-
x-0 lg:static transition-transform duration-300 ease-in-out">

      <!-- Logo -->
      <div class="flex items-center justify-between h-16 px-6
border-b border-gray-800">
        <h1 class="text-xl font-bold">AdminPanel</h1>
        <button id="sidebar-close" class="lg:hidden text-gray-400
hover:text-white">
          <svg class="w-6 h-6" fill="none" stroke="currentColor"
viewBox="0 0 24 24">
            <path stroke-linecap="round" stroke-linejoin="round"
stroke-width="2" d="M6 18L18 6M6 6l12 12"></path>
          </svg>
        </button>
      </div>

      <!-- Navigation -->
      <nav class="flex-1 overflow-y-auto p-4 space-y-2">

        <!-- Active Item -->
        <a href="#" class="flex items-center space-x-3 px-4 py-3
bg-blue-600 rounded-lg">
          <svg class="w-5 h-5" fill="none" stroke="currentColor"
viewBox="0 0 24 24">
```

```
            <path stroke-linecap="round" stroke-linejoin="round"
stroke-width="2" d="M3 12l2-2m0 0l7-7 7 7M5 10v10a1 1 0 001
1h3m10-11l2 2m-2-2v10a1 1 0 01-1 1h-3m-6 0a1 1 0 001-1v-4a1 1 0
011-1h2a1 1 0 011 1v4a1 1 0 001 1m-6 0h6"></path>
          </svg>
          <span class="font-medium">Dashboard</span>
        </a>

        <!-- Regular Items -->
        <a href="#" class="flex items-center space-x-3 px-4 py-3
text-gray-400 hover:bg-gray-800 hover:text-white rounded-lg
transition">
          <svg class="w-5 h-5" fill="none" stroke="currentColor"
viewBox="0 0 24 24">
            <path stroke-linecap="round" stroke-linejoin="round"
stroke-width="2" d="M9 19v-6a2 2 0 00-2-2H5a2 2 0 00-2 2v6a2 2 0
002 2h2a2 2 0 002-2zm0 0V9a2 2 0 012-2h2a2 2 0 012 2v10m-6 0a2 2
0 002 2h2a2 2 0 002-2m0 0V5a2 2 0 012-2h2a2 2 0 012 2v14a2 2 0
01-2 2h-2a2 2 0 01-2-2z"></path>
          </svg>
          <span>Analytics</span>
        </a>

        <a href="#" class="flex items-center space-x-3 px-4 py-3
text-gray-400 hover:bg-gray-800 hover:text-white rounded-lg
transition">
          <svg class="w-5 h-5" fill="none" stroke="currentColor"
viewBox="0 0 24 24">
            <path stroke-linecap="round" stroke-linejoin="round"
stroke-width="2" d="M16 11V7a4 4 0 00-8 0v4M5 9h14l1 12H4L5 9z">
</path>
          </svg>
          <span>Products</span>
        </a>
```

```
        <a href="#" class="flex items-center space-x-3 px-4 py-3
text-gray-400 hover:bg-gray-800 hover:text-white rounded-lg
transition">
          <svg class="w-5 h-5" fill="none" stroke="currentColor"
viewBox="0 0 24 24">
            <path stroke-linecap="round" stroke-linejoin="round"
stroke-width="2" d="M3 3h2l.4 2M7 13h10l4-8H5.4M7 13L5.4 5M7 13l-
2.293 2.293c-.63.63-.184 1.707.707 1.707H17m0 0a2 2 0 100 4 2 2 0
000-4zm-8 2a2 2 0 11-4 0 2 2 0 014 0z"></path>
          </svg>
          <span>Orders</span>
        </a>

        <a href="#" class="flex items-center space-x-3 px-4 py-3
text-gray-400 hover:bg-gray-800 hover:text-white rounded-lg
transition">
          <svg class="w-5 h-5" fill="none" stroke="currentColor"
viewBox="0 0 24 24">
            <path stroke-linecap="round" stroke-linejoin="round"
stroke-width="2" d="M12 4.354a4 4 0 110 5.292M15 21H3v-1a6 6 0
0112 0v1zm0 0h6v-1a6 6 0 00-9-5.197M13 7a4 4 0 11-8 0 4 4 0 018
0z"></path>
          </svg>
          <span>Customers</span>
        </a>

        <a href="#" class="flex items-center space-x-3 px-4 py-3
text-gray-400 hover:bg-gray-800 hover:text-white rounded-lg
transition">
          <svg class="w-5 h-5" fill="none" stroke="currentColor"
viewBox="0 0 24 24">
            <path stroke-linecap="round" stroke-linejoin="round"
stroke-width="2" d="M9 17v-2m3 2v-4m3 4v-6m2 10H7a2 2 0 01-2-
2V5a2 2 0 012-2h5.586a1 1 0 01.707.293l5.414 5.414a1 1 0
01.293.707V19a2 2 0 01-2 2z"></path>
```

```
          </svg>
          <span>Reports</span>
        </a>

        <!-- Divider -->
        <div class="pt-4 pb-2">
          <div class="px-4 text-xs font-semibold text-gray-500
uppercase tracking-wider">
            Settings
          </div>
        </div>

        <a href="#" class="flex items-center space-x-3 px-4 py-3
text-gray-400 hover:bg-gray-800 hover:text-white rounded-lg
transition">
          <svg class="w-5 h-5" fill="none" stroke="currentColor"
viewBox="0 0 24 24">
            <path stroke-linecap="round" stroke-linejoin="round"
stroke-width="2" d="M10.325 4.317c.426-1.756 2.924-1.756 3.35
0a1.724 1.724 0 002.573 1.066c1.543-.94 3.31.826 2.37 2.37a1.724
1.724 0 001.065 2.572c1.756.426 1.756 2.924 0 3.35a1.724 1.724 0
00-1.066 2.573c.94 1.543-.826 3.31-2.37 2.37a1.724 1.724 0 00-
2.572 1.065c-.426 1.756-2.924 1.756-3.35 0a1.724 1.724 0 00-
2.573-1.066c-1.543.94-3.31-.826-2.37-2.37a1.724 1.724 0 00-1.065-
2.572c-1.756-.426-1.756-2.924 0-3.35a1.724 1.724 0 001.066-
2.573c-.94-1.543.826-3.31 2.37-2.37.996.608 2.296.07 2.572-
1.065z"></path>
            <path stroke-linecap="round" stroke-linejoin="round"
stroke-width="2" d="M15 12a3 3 0 11-6 0 3 3 0 016 0z"></path>
          </svg>
          <span>Settings</span>
        </a>

        <a href="#" class="flex items-center space-x-3 px-4 py-3
text-gray-400 hover:bg-gray-800 hover:text-white rounded-lg
```

```
transition">
          <svg class="w-5 h-5" fill="none" stroke="currentColor"
viewBox="0 0 24 24">
            <path stroke-linecap="round" stroke-linejoin="round"
stroke-width="2" d="M17 16l4-4m0 0l-4-4m4 4H7m6 4v1a3 3 0 01-3
3H6a3 3 0 01-3-3V7a3 3 0 013-3h4a3 3 0 013 3v1"></path>
          </svg>
          <span>Logout</span>
        </a>

      </nav>

      <!-- User Profile (at bottom) -->
      <div class="border-t border-gray-800 p-4">
        <div class="flex items-center space-x-3">
          <img src="https://i.pravatar.cc/40?img=12" class="w-10
h-10 rounded-full" alt="User">
          <div class="flex-1 min-w-0">
            <p class="text-sm font-medium text-white
truncate">John Doe</p>
            <p class="text-xs text-gray-400
truncate">john@example.com</p>
          </div>
        </div>
      </div>

    </aside>

    <!-- Main Content -->
    <div class="flex-1 flex flex-col overflow-hidden">

      <!-- Top Bar -->
      <header class="bg-white shadow-sm z-10">
        <div class="flex items-center justify-between h-16 px-4
sm:px-6 lg:px-8">
```

```
          <!-- Mobile menu button + Search -->
          <div class="flex items-center space-x-4 flex-1">
            <button id="sidebar-open" class="lg:hidden text-gray-
500 hover:text-gray-700">
              <svg class="w-6 h-6" fill="none"
stroke="currentColor" viewBox="0 0 24 24">
                <path stroke-linecap="round" stroke-
linejoin="round" stroke-width="2" d="M4 6h16M4 12h16M4 18h16">
</path>
              </svg>
            </button>

            <!-- Search Bar -->
            <div class="hidden sm:block flex-1 max-w-md">
              <div class="relative">
                <svg class="absolute left-3 top-2.5 w-5 h-5 text-
gray-400" fill="none" stroke="currentColor" viewBox="0 0 24 24">
                  <path stroke-linecap="round" stroke-
linejoin="round" stroke-width="2" d="M21 21l-6-6m2-5a7 7 0 11-14
0 7 7 0 0114 0z"></path>
                </svg>
                <input
                  type="text"
                  placeholder="Search..."
                  class="w-full pl-10 pr-4 py-2 border border-
gray-300 rounded-lg focus:ring-2 focus:ring-blue-500
focus:border-transparent"
                >
              </div>
            </div>
          </div>

          <!-- Right side actions -->
          <div class="flex items-center space-x-4">
```

```
            <!-- Notifications -->
            <button class="relative p-2 text-gray-400 hover:text-
gray-600">
              <svg class="w-6 h-6" fill="none"
stroke="currentColor" viewBox="0 0 24 24">
                <path stroke-linecap="round" stroke-
linejoin="round" stroke-width="2" d="M15 17h5l-1.405-1.405A2.032
2.032 0 0118 14.158V11a6.002 6.002 0 00-4-5.659V5a2 2 0 10-4
0v.341C7.67 6.165 6 8.388 6 11v3.159c0 .538-.214 1.055-.595
1.436L4 17h5m6 0v1a3 3 0 11-6 0v-1m6 0H9"></path>
              </svg>
              <span class="absolute top-1 right-1 w-2 h-2 bg-red-
500 rounded-full"></span>
            </button>

            <!-- Profile -->
            <div class="hidden sm:flex items-center space-x-3">
              <img src="https://i.pravatar.cc/32?img=12"
class="w-8 h-8 rounded-full" alt="User">
              <span class="text-sm font-medium text-gray-
700">John Doe</span>
            </div>

          </div>

        </div>
      </header>

      <!-- Main Content Area -->
      <main class="flex-1 overflow-y-auto bg-gray-100 p-4 sm:p-6
lg:p-8">

        <!-- Page Header -->
        <div class="mb-8">
```

```
          <h1 class="text-3xl font-bold text-gray-
900">Dashboard</h1>
          <p class="text-gray-600 mt-1">Welcome back! Here's
what's happening.</p>
        </div>

        <!-- Stats Grid -->
        <div class="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-
4 gap-6 mb-8">

          <!-- Stat Card 1 -->
          <div class="bg-white rounded-lg shadow p-6">
            <div class="flex items-center justify-between mb-4">
              <div class="text-sm font-medium text-gray-
500">Total Revenue</div>
              <div class="p-2 bg-blue-50 rounded-lg">
                <svg class="w-6 h-6 text-blue-600" fill="none"
stroke="currentColor" viewBox="0 0 24 24">
                  <path stroke-linecap="round" stroke-
linejoin="round" stroke-width="2" d="M12 8c-1.657 0-3 .895-3
2s1.343 2 3 2 3 .895 3 2-1.343 2-3 2m0-8c1.11 0 2.08.402 2.599
1M12 8V7m0 1v8m0 0v1m0-1c-1.11 0-2.08-.402-2.599-1M21 12a9 9 0
11-18 0 9 9 0 0118 0z"></path>
                </svg>
              </div>
            </div>
            <div class="text-3xl font-bold text-gray-900 mb-
2">$45,678</div>
            <div class="flex items-center text-sm">
              <span class="text-green-600 font-medium flex items-
center">
                <svg class="w-4 h-4 mr-1" fill="currentColor"
viewBox="0 0 20 20">
                  <path fill-rule="evenodd" d="M5.293 9.707a1 1 0
010-1.414l4-4a1 1 0 011.414 0l4 4a1 1 0 01-1.414 1.414L11
```

```
7.414V15a1 1 0 11-2 0V7.414L6.707 9.707a1 1 0 01-1.414 0z" clip-
rule="evenodd"></path>
                </svg>
                12.5%
              </span>
              <span class="text-gray-500 ml-2">from last
month</span>
            </div>
          </div>

          <!-- Stat Card 2 -->
          <div class="bg-white rounded-lg shadow p-6">
            <div class="flex items-center justify-between mb-4">
              <div class="text-sm font-medium text-gray-500">New
Customers</div>
              <div class="p-2 bg-green-50 rounded-lg">
                <svg class="w-6 h-6 text-green-600" fill="none"
stroke="currentColor" viewBox="0 0 24 24">
                  <path stroke-linecap="round" stroke-
linejoin="round" stroke-width="2" d="M17 20h5v-2a3 3 0 00-5.356-
1.857M17 20H7m10 0v-2c0-.656-.126-1.283-.356-1.857M7 20H2v-2a3 3
0 015.356-1.857M7 20v-2c0-.656.126-1.283.356-1.857m0 0a5.002
5.002 0 019.288 0M15 7a3 3 0 11-6 0 3 3 0 016 0zm6 3a2 2 0 11-4 0
2 2 0 014 0zM7 10a2 2 0 11-4 0 2 2 0 014 0z"></path>
                </svg>
              </div>
            </div>
            <div class="text-3xl font-bold text-gray-900 mb-
2">1,234</div>
            <div class="flex items-center text-sm">
              <span class="text-green-600 font-medium flex items-
center">
                <svg class="w-4 h-4 mr-1" fill="currentColor"
viewBox="0 0 20 20">
                  <path fill-rule="evenodd" d="M5.293 9.707a1 1 0
```

```
010-1.414l4-4a1 1 0 011.414 0l4 4a1 1 0 01-1.414 1.414L11
7.414V15a1 1 0 11-2 0V7.414L6.707 9.707a1 1 0 01-1.414 0z" clip-
rule="evenodd"></path>
                </svg>
                8.2%
              </span>
              <span class="text-gray-500 ml-2">from last
month</span>
            </div>
          </div>

          <!-- Stat Card 3 -->
          <div class="bg-white rounded-lg shadow p-6">
            <div class="flex items-center justify-between mb-4">
              <div class="text-sm font-medium text-gray-
500">Total Orders</div>
              <div class="p-2 bg-yellow-50 rounded-lg">
                <svg class="w-6 h-6 text-yellow-600" fill="none"
stroke="currentColor" viewBox="0 0 24 24">
                  <path stroke-linecap="round" stroke-
linejoin="round" stroke-width="2" d="M16 11V7a4 4 0 00-8 0v4M5
9h14l1 12H4L5 9z"></path>
                </svg>
              </div>
            </div>
            <div class="text-3xl font-bold text-gray-900 mb-
2">789</div>
            <div class="flex items-center text-sm">
              <span class="text-red-600 font-medium flex items-
center">
                <svg class="w-4 h-4 mr-1" fill="currentColor"
viewBox="0 0 20 20">
                  <path fill-rule="evenodd" d="M14.707 10.293a1 1
0 010 1.414l-4 4a1 1 0 01-1.414 0l-4-4a1 1 0 111.414-1.414L9
12.586V5a1 1 0 012 0v7.586l2.293-2.293a1 1 0 011.414 0z" clip-
```

```
rule="evenodd"></path>
                </svg>
                3.1%
              </span>
              <span class="text-gray-500 ml-2">from last
month</span>
            </div>
          </div>

          <!-- Stat Card 4 -->
          <div class="bg-white rounded-lg shadow p-6">
            <div class="flex items-center justify-between mb-4">
              <div class="text-sm font-medium text-gray-
500">Conversion Rate</div>
              <div class="p-2 bg-purple-50 rounded-lg">
                <svg class="w-6 h-6 text-purple-600" fill="none"
stroke="currentColor" viewBox="0 0 24 24">
                  <path stroke-linecap="round" stroke-
linejoin="round" stroke-width="2" d="M9 19v-6a2 2 0 00-2-2H5a2 2
0 00-2 2v6a2 2 0 002 2h2a2 2 0 002-2zm0 0V9a2 2 0 012-2h2a2 2 0
012 2v10m-6 0a2 2 0 002 2h2a2 2 0 002-2m0 0V5a2 2 0 012-2h2a2 2 0
012 2v14a2 2 0 01-2 2h-2a2 2 0 01-2-2z"></path>
                </svg>
              </div>
            </div>
            <div class="text-3xl font-bold text-gray-900 mb-
2">3.45%</div>
            <div class="flex items-center text-sm">
              <span class="text-green-600 font-medium flex items-
center">
                <svg class="w-4 h-4 mr-1" fill="currentColor"
viewBox="0 0 20 20">
                  <path fill-rule="evenodd" d="M5.293 9.707a1 1 0
010-1.414l4-4a1 1 0 011.414 0l4 4a1 1 0 01-1.414 1.414L11
7.414V15a1 1 0 11-2 0V7.414L6.707 9.707a1 1 0 01-1.414 0z" clip-
```

```
rule="evenodd"></path>
                </svg>
                2.3%
              </span>
              <span class="text-gray-500 ml-2">from last
month</span>
            </div>
          </div>

        </div>

        <!-- Charts Row -->
        <div class="grid grid-cols-1 lg:grid-cols-2 gap-6 mb-8">

          <!-- Chart 1 - Revenue -->
          <div class="bg-white rounded-lg shadow p-6">
            <div class="flex items-center justify-between mb-6">
              <h2 class="text-lg font-semibold text-gray-
900">Revenue Overview</h2>
              <select class="text-sm border border-gray-300
rounded-lg px-3 py-1.5 focus:ring-2 focus:ring-blue-500
focus:border-transparent">
                <option>Last 7 days</option>
                <option>Last 30 days</option>
                <option>Last 90 days</option>
              </select>
            </div>
            <!-- Placeholder for chart -->
            <div class="h-64 bg-gradient-to-br from-blue-50 to-
indigo-100 rounded-lg flex items-center justify-center">
              <div class="text-center">
                <svg class="w-16 h-16 text-blue-400 mx-auto mb-2"
fill="none" stroke="currentColor" viewBox="0 0 24 24">
                  <path stroke-linecap="round" stroke-
linejoin="round" stroke-width="2" d="M9 19v-6a2 2 0 00-2-2H5a2 2
```

```
0 00-2 2v6a2 2 0 002 2h2a2 2 0 002-2zm0 0V9a2 2 0 012-2h2a2 2 0
012 2v10m-6 0a2 2 0 002 2h2a2 2 0 002-2m0 0V5a2 2 0 012-2h2a2 2 0
012 2v14a2 2 0 01-2 2h-2a2 2 0 01-2-2z"></path>
                </svg>
                <p class="text-gray-500 text-sm">Chart goes
here</p>
              </div>
            </div>
          </div>

          <!-- Chart 2 - Traffic Sources -->
          <div class="bg-white rounded-lg shadow p-6">
            <div class="flex items-center justify-between mb-6">
              <h2 class="text-lg font-semibold text-gray-
900">Traffic Sources</h2>
              <button class="text-sm text-blue-600 hover:text-
blue-800 font-medium">View All</button>
            </div>
            <!-- Pie chart placeholder -->
            <div class="h-64 flex items-center justify-center">
              <div class="w-48 h-48 rounded-full bg-gradient-to-
br from-blue-400 via-purple-400 to-pink-400 flex items-center
justify-center">
                <div class="w-32 h-32 rounded-full bg-white flex
items-center justify-center">
                  <div class="text-center">
                    <div class="text-2xl font-bold text-gray-
900">45%</div>
                    <div class="text-xs text-gray-
500">Organic</div>
                  </div>
                </div>
              </div>
            </div>
            <!-- Legend -->
```

```
            <div class="grid grid-cols-2 gap-4 mt-6">
              <div class="flex items-center">
                <div class="w-3 h-3 bg-blue-500 rounded-full mr-
2"></div>
                <span class="text-sm text-gray-600">Organic (45%)
</span>
              </div>
              <div class="flex items-center">
                <div class="w-3 h-3 bg-purple-500 rounded-full
mr-2"></div>
                <span class="text-sm text-gray-600">Direct (25%)
</span>
              </div>
              <div class="flex items-center">
                <div class="w-3 h-3 bg-pink-500 rounded-full mr-
2"></div>
                <span class="text-sm text-gray-600">Referral
(20%)</span>
              </div>
              <div class="flex items-center">
                <div class="w-3 h-3 bg-yellow-500 rounded-full
mr-2"></div>
                <span class="text-sm text-gray-600">Social (10%)
</span>
              </div>
            </div>
          </div>

        </div>

        <!-- Recent Orders Table -->
        <div class="bg-white rounded-lg shadow">
          <div class="px-6 py-4 border-b border-gray-200 flex
items-center justify-between">
            <h2 class="text-lg font-semibold text-gray-
```

```
900">Recent Orders</h2>
            <button class="text-sm text-blue-600 hover:text-blue-
800 font-medium">View All</button>
          </div>

          <div class="overflow-x-auto">
            <table class="w-full">
              <thead class="bg-gray-50 border-b border-gray-200">
                <tr>
                  <th class="px-6 py-3 text-left text-xs font-
medium text-gray-500 uppercase tracking-wider">Order ID</th>
                  <th class="px-6 py-3 text-left text-xs font-
medium text-gray-500 uppercase tracking-wider">Customer</th>
                  <th class="px-6 py-3 text-left text-xs font-
medium text-gray-500 uppercase tracking-wider">Date</th>
                  <th class="px-6 py-3 text-left text-xs font-
medium text-gray-500 uppercase tracking-wider">Amount</th>
                  <th class="px-6 py-3 text-left text-xs font-
medium text-gray-500 uppercase tracking-wider">Status</th>
                  <th class="px-6 py-3 text-right text-xs font-
medium text-gray-500 uppercase tracking-wider">Actions</th>
                </tr>
              </thead>
              <tbody class="bg-white divide-y divide-gray-200">

                <tr class="hover:bg-gray-50">
                  <td class="px-6 py-4 whitespace-nowrap">
                    <div class="text-sm font-medium text-gray-
900">#ORD-001</div>
                  </td>
                  <td class="px-6 py-4 whitespace-nowrap">
                    <div class="flex items-center">
                      <img src="https://i.pravatar.cc/32?img=1"
class="w-8 h-8 rounded-full mr-3" alt="User">
                      <div>
```

```
                        <div class="text-sm font-medium text-
gray-900">Sarah Johnson</div>
                        <div class="text-sm text-gray-
500">sarah@example.com</div>
                      </div>
                    </div>
                  </td>
                  <td class="px-6 py-4 whitespace-nowrap">
                    <div class="text-sm text-gray-900">Jan 2,
2026</div>
                  </td>
                  <td class="px-6 py-4 whitespace-nowrap">
                    <div class="text-sm font-medium text-gray-
900">$249.00</div>
                  </td>
                  <td class="px-6 py-4 whitespace-nowrap">
                    <span class="px-2 py-1 inline-flex text-xs
leading-5 font-semibold rounded-full bg-green-100 text-green-
800">
                      Completed
                    </span>
                  </td>
                  <td class="px-6 py-4 whitespace-nowrap text-
right text-sm font-medium">
                    <button class="text-blue-600 hover:text-blue-
900 mr-3">View</button>
                    <button class="text-gray-600 hover:text-gray-
900">Edit</button>
                  </td>
                </tr>

                <tr class="hover:bg-gray-50">
                  <td class="px-6 py-4 whitespace-nowrap">
                    <div class="text-sm font-medium text-gray-
900">#ORD-002</div>
```

```
                  </td>
                  <td class="px-6 py-4 whitespace-nowrap">
                    <div class="flex items-center">
                      <img src="https://i.pravatar.cc/32?img=2"
class="w-8 h-8 rounded-full mr-3" alt="User">
                      <div>
                        <div class="text-sm font-medium text-
gray-900">Mike Chen</div>
                        <div class="text-sm text-gray-
500">mike@example.com</div>
                      </div>
                    </div>
                  </td>
                  <td class="px-6 py-4 whitespace-nowrap">
                    <div class="text-sm text-gray-900">Jan 3,
2026</div>
                  </td>
                  <td class="px-6 py-4 whitespace-nowrap">
                    <div class="text-sm font-medium text-gray-
900">$149.00</div>
                  </td>
                  <td class="px-6 py-4 whitespace-nowrap">
                    <span class="px-2 py-1 inline-flex text-xs
leading-5 font-semibold rounded-full bg-yellow-100 text-yellow-
800">
                      Processing
                    </span>
                  </td>
                  <td class="px-6 py-4 whitespace-nowrap text-
right text-sm font-medium">
                    <button class="text-blue-600 hover:text-blue-
900 mr-3">View</button>
                    <button class="text-gray-600 hover:text-gray-
900">Edit</button>
                  </td>
```

```
                </tr>

                <tr class="hover:bg-gray-50">
                  <td class="px-6 py-4 whitespace-nowrap">
                    <div class="text-sm font-medium text-gray-
900">#ORD-003</div>
                  </td>
                  <td class="px-6 py-4 whitespace-nowrap">
                    <div class="flex items-center">
                      <img src="https://i.pravatar.cc/32?img=3"
class="w-8 h-8 rounded-full mr-3" alt="User">
                      <div>
                        <div class="text-sm font-medium text-
gray-900">Emma Wilson</div>
                        <div class="text-sm text-gray-
500">emma@example.com</div>
                      </div>
                    </div>
                  </td>
                  <td class="px-6 py-4 whitespace-nowrap">
                    <div class="text-sm text-gray-900">Jan 3,
2026</div>
                  </td>
                  <td class="px-6 py-4 whitespace-nowrap">
                    <div class="text-sm font-medium text-gray-
900">$349.00</div>
                  </td>
                  <td class="px-6 py-4 whitespace-nowrap">
                    <span class="px-2 py-1 inline-flex text-xs
leading-5 font-semibold rounded-full bg-blue-100 text-blue-800">
                      Shipped
                    </span>
                  </td>
                  <td class="px-6 py-4 whitespace-nowrap text-
right text-sm font-medium">
```

```
                    <button class="text-blue-600 hover:text-blue-
900 mr-3">View</button>
                    <button class="text-gray-600 hover:text-gray-
900">Edit</button>
                  </td>
                </tr>

                <tr class="hover:bg-gray-50">
                  <td class="px-6 py-4 whitespace-nowrap">
                    <div class="text-sm font-medium text-gray-
900">#ORD-004</div>
                  </td>
                  <td class="px-6 py-4 whitespace-nowrap">
                    <div class="flex items-center">
                      <img src="https://i.pravatar.cc/32?img=4"
class="w-8 h-8 rounded-full mr-3" alt="User">
                      <div>
                        <div class="text-sm font-medium text-
gray-900">David Lee</div>
                        <div class="text-sm text-gray-
500">david@example.com</div>
                      </div>
                    </div>
                  </td>
                  <td class="px-6 py-4 whitespace-nowrap">
                    <div class="text-sm text-gray-900">Jan 4,
2026</div>
                  </td>
                  <td class="px-6 py-4 whitespace-nowrap">
                    <div class="text-sm font-medium text-gray-
900">$99.00</div>
                  </td>
                  <td class="px-6 py-4 whitespace-nowrap">
                    <span class="px-2 py-1 inline-flex text-xs
leading-5 font-semibold rounded-full bg-red-100 text-red-800">
```

```
                      Cancelled
                    </span>
                  </td>
                  <td class="px-6 py-4 whitespace-nowrap text-
right text-sm font-medium">
                    <button class="text-blue-600 hover:text-blue-
900 mr-3">View</button>
                    <button class="text-gray-600 hover:text-gray-
900">Edit</button>
                  </td>
                </tr>

                <tr class="hover:bg-gray-50">
                  <td class="px-6 py-4 whitespace-nowrap">
                    <div class="text-sm font-medium text-gray-
900">#ORD-005</div>
                  </td>
                  <td class="px-6 py-4 whitespace-nowrap">
                    <div class="flex items-center">
                      <img src="https://i.pravatar.cc/32?img=5"
class="w-8 h-8 rounded-full mr-3" alt="User">
                      <div>
                        <div class="text-sm font-medium text-
gray-900">Lisa Brown</div>
                        <div class="text-sm text-gray-
500">lisa@example.com</div>
                      </div>
                    </div>
                  </td>
                  <td class="px-6 py-4 whitespace-nowrap">
                    <div class="text-sm text-gray-900">Jan 5,
2026</div>
                  </td>
                  <td class="px-6 py-4 whitespace-nowrap">
                    <div class="text-sm font-medium text-gray-
```

```
900">$199.00</div>
                  </td>
                  <td class="px-6 py-4 whitespace-nowrap">
                    <span class="px-2 py-1 inline-flex text-xs
leading-5 font-semibold rounded-full bg-green-100 text-green-
800">
                      Completed
                    </span>
                  </td>
                  <td class="px-6 py-4 whitespace-nowrap text-
right text-sm font-medium">
                    <button class="text-blue-600 hover:text-blue-
900 mr-3">View</button>
                    <button class="text-gray-600 hover:text-gray-
900">Edit</button>
                  </td>
                </tr>

              </tbody>
            </table>
          </div>

          <!-- Pagination -->
          <div class="px-6 py-4 border-t border-gray-200 flex
items-center justify-between">
            <div class="text-sm text-gray-700">
              Showing <span class="font-medium">1</span> to <span
class="font-medium">5</span> of <span class="font-
medium">97</span> results
            </div>
            <div class="flex space-x-2">
              <button class="px-3 py-1 border border-gray-300
rounded hover:bg-gray-50 disabled:opacity-50 disabled:cursor-not-
allowed" disabled>
                Previous
```

```
              </button>
              <button class="px-3 py-1 bg-blue-600 text-white
rounded">1</button>
              <button class="px-3 py-1 border border-gray-300
rounded hover:bg-gray-50">2</button>
              <button class="px-3 py-1 border border-gray-300
rounded hover:bg-gray-50">3</button>
              <button class="px-3 py-1 border border-gray-300
rounded hover:bg-gray-50">
                Next
              </button>
            </div>
          </div>

        </div>

      </main>

    </div>

  </div>

  <!-- Sidebar Overlay (mobile) -->
  <div id="sidebar-overlay" class="fixed inset-0 bg-black bg-
opacity-50 z-40 lg:hidden hidden"></div>

  <!-- JavaScript -->
  <script>
    const sidebar = document.getElementById('sidebar');
    const sidebarOpen = document.getElementById('sidebar-open');
    const sidebarClose = document.getElementById('sidebar-
close');
    const sidebarOverlay = document.getElementById('sidebar-
overlay');
```

```
    function openSidebar() {
      sidebar.classList.remove('-translate-x-full');
      sidebarOverlay.classList.remove('hidden');
    }

    function closeSidebar() {
      sidebar.classList.add('-translate-x-full');
      sidebarOverlay.classList.add('hidden');
    }

    sidebarOpen.addEventListener('click', openSidebar);
    sidebarClose.addEventListener('click', closeSidebar);
    sidebarOverlay.addEventListener('click', closeSidebar);
  </script>

</body>
</html>
```

This dashboard demonstrates:

✅ **Responsive sidebar** (collapsible on mobile)
✅ **Top navigation bar** with search and notifications
✅ **Stat cards** with icons and trend indicators
✅ **Chart placeholders** (can be replaced with Chart.js, etc.)
✅ **Data table** with pagination
✅ **Status badges** with color coding
✅ **Mobile-friendly** with overlay menu
✅ **Professional styling** throughout

Would you like me to continue with **Project 3: E-commerce Product Card** next?

1/7/2026, 1:10:18 PM

Project 3: E-commerce Product Card

A complete e-commerce product card with multiple variations, image gallery, and shopping cart functionality.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>E-commerce Product Cards</title>
  <script src="https://cdn.tailwindcss.com"></script>
</head>
<body class="bg-gray-50 p-8">

  <div class="max-w-7xl mx-auto">

    <h1 class="text-3xl font-bold text-gray-900 mb-8">E-commerce
Product Cards</h1>

    <!-- Product Grid -->
    <div class="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3
gap-8 mb-16">

      <!-- Product Card 1: Basic -->
      <div class="bg-white rounded-xl shadow-md overflow-hidden
group hover:shadow-xl transition-shadow">

        <!-- Image Container -->
        <div class="relative overflow-hidden bg-gray-100">
          <img
            src="https://images.unsplash.com/photo-1572635196237-
14b3f281503f?w=400&h=400&fit=crop"
```

```
            alt="Product"
            class="w-full h-64 object-cover transform group-
hover:scale-110 transition-transform duration-500"
          >

          <!-- Badges -->
          <div class="absolute top-4 left-4 flex flex-col space-
y-2">
            <span class="bg-red-500 text-white text-xs font-bold
px-3 py-1 rounded-full">
              -20%
            </span>
            <span class="bg-blue-500 text-white text-xs font-bold
px-3 py-1 rounded-full">
              New
            </span>
          </div>

          <!-- Wishlist Button -->
          <button class="absolute top-4 right-4 w-10 h-10 bg-
white rounded-full shadow-md flex items-center justify-center
text-gray-400 hover:text-red-500 transition-colors">
            <svg class="w-5 h-5" fill="none"
stroke="currentColor" viewBox="0 0 24 24">
              <path stroke-linecap="round" stroke-
linejoin="round" stroke-width="2" d="M4.318 6.318a4.5 4.5 0 000
6.364L12 20.364l7.682-7.682a4.5 4.5 0 00-6.364-6.364L12 7.636l-
1.318-1.318a4.5 4.5 0 00-6.364 0z"></path>
            </svg>
          </button>

          <!-- Quick View (appears on hover) -->
          <div class="absolute inset-x-0 bottom-0 transform
translate-y-full group-hover:translate-y-0 transition-transform
duration-300">
```

```
            <button class="w-full bg-white bg-opacity-90
backdrop-blur-sm text-gray-900 py-3 font-semibold hover:bg-
opacity-100 transition">
              Quick View
            </button>
          </div>
        </div>

        <!-- Product Info -->
        <div class="p-6">

          <!-- Category -->
          <div class="text-xs text-gray-500 uppercase tracking-
wider mb-2">
            Electronics
          </div>

          <!-- Product Name -->
          <h3 class="text-lg font-semibold text-gray-900 mb-2
group-hover:text-blue-600 transition-colors">
            Wireless Headphones
          </h3>

          <!-- Rating -->
          <div class="flex items-center mb-3">
            <div class="flex text-yellow-400">
              <svg class="w-4 h-4" fill="currentColor" viewBox="0
0 20 20"><path d="M9.049 2.927c.3-.921 1.603-.921 1.902 0l1.07
3.292a1 1 0 00.95.69h3.462c.969 0 1.371 1.24.588 1.81l-2.8
2.034a1 1 0 00-.364 1.118l1.07 3.292c.3.921-.755 1.688-1.54
1.118l-2.8-2.034a1 1 0 00-1.175 0l-2.8 2.034c-.784.57-1.838-.197-
1.539-1.118l1.07-3.292a1 1 0 00-.364-1.118L2.98
8.72c-.783-.57-.38-1.81.588-1.81h3.461a1 1 0 00.951-.69l1.07-
3.292z"></path></svg>
              <svg class="w-4 h-4" fill="currentColor" viewBox="0
```

```
0 20 20"><path d="M9.049 2.927c.3-.921 1.603-.921 1.902 0l1.07
3.292a1 1 0 00.95.69h3.462c.969 0 1.371 1.24.588 1.81l-2.8
2.034a1 1 0 00-.364 1.118l1.07 3.292c.3.921-.755 1.688-1.54
1.118l-2.8-2.034a1 1 0 00-1.175 0l-2.8 2.034c-.784.57-1.838-.197-
1.539-1.118l1.07-3.292a1 1 0 00-.364-1.118L2.98
8.72c-.783-.57-.38-1.81.588-1.81h3.461a1 1 0 00.951-.69l1.07-
3.292z"></path></svg>
              <svg class="w-4 h-4" fill="currentColor" viewBox="0
0 20 20"><path d="M9.049 2.927c.3-.921 1.603-.921 1.902 0l1.07
3.292a1 1 0 00.95.69h3.462c.969 0 1.371 1.24.588 1.81l-2.8
2.034a1 1 0 00-.364 1.118l1.07 3.292c.3.921-.755 1.688-1.54
1.118l-2.8-2.034a1 1 0 00-1.175 0l-2.8 2.034c-.784.57-1.838-.197-
1.539-1.118l1.07-3.292a1 1 0 00-.364-1.118L2.98
8.72c-.783-.57-.38-1.81.588-1.81h3.461a1 1 0 00.951-.69l1.07-
3.292z"></path></svg>
              <svg class="w-4 h-4" fill="currentColor" viewBox="0
0 20 20"><path d="M9.049 2.927c.3-.921 1.603-.921 1.902 0l1.07
3.292a1 1 0 00.95.69h3.462c.969 0 1.371 1.24.588 1.81l-2.8
2.034a1 1 0 00-.364 1.118l1.07 3.292c.3.921-.755 1.688-1.54
1.118l-2.8-2.034a1 1 0 00-1.175 0l-2.8 2.034c-.784.57-1.838-.197-
1.539-1.118l1.07-3.292a1 1 0 00-.364-1.118L2.98
8.72c-.783-.57-.38-1.81.588-1.81h3.461a1 1 0 00.951-.69l1.07-
3.292z"></path></svg>
              <svg class="w-4 h-4" fill="currentColor" viewBox="0
0 20 20"><path d="M9.049 2.927c.3-.921 1.603-.921 1.902 0l1.07
3.292a1 1 0 00.95.69h3.462c.969 0 1.371 1.24.588 1.81l-2.8
2.034a1 1 0 00-.364 1.118l1.07 3.292c.3.921-.755 1.688-1.54
1.118l-2.8-2.034a1 1 0 00-1.175 0l-2.8 2.034c-.784.57-1.838-.197-
1.539-1.118l1.07-3.292a1 1 0 00-.364-1.118L2.98
8.72c-.783-.57-.38-1.81.588-1.81h3.461a1 1 0 00.951-.69l1.07-
3.292z"></path></svg>
            </div>
            <span class="text-sm text-gray-600 ml-2">(128
reviews)</span>
          </div>
```

```
          <!-- Price -->
          <div class="flex items-center justify-between mb-4">
            <div class="flex items-center space-x-2">
              <span class="text-2xl font-bold text-gray-
900">$79.99</span>
              <span class="text-sm text-gray-500 line-
through">$99.99</span>
            </div>
            <span class="text-sm text-green-600 font-medium">In
Stock</span>
          </div>

          <!-- Add to Cart Button -->
          <button class="w-full bg-blue-600 hover:bg-blue-700
text-white py-3 rounded-lg font-semibold transition flex items-
center justify-center space-x-2">
            <svg class="w-5 h-5" fill="none"
stroke="currentColor" viewBox="0 0 24 24">
              <path stroke-linecap="round" stroke-
linejoin="round" stroke-width="2" d="M3 3h2l.4 2M7 13h10l4-
8H5.4M7 13L5.4 5M7 13l-2.293 2.293c-.63.63-.184 1.707.707
1.707H17m0 0a2 2 0 100 4 2 2 0 000-4zm-8 2a2 2 0 11-4 0 2 2 0 014
0z"></path>
            </svg>
            <span>Add to Cart</span>
          </button>

        </div>

      </div>

      <!-- Product Card 2: With Color Options -->
      <div class="bg-white rounded-xl shadow-md overflow-hidden
group hover:shadow-xl transition-shadow">
```

```
        <div class="relative overflow-hidden bg-gray-100">
          <img
            src="https://images.unsplash.com/photo-1523275335684-
37898b6baf30?w=400&h=400&fit=crop"
            alt="Product"
            class="w-full h-64 object-cover transform group-
hover:scale-110 transition-transform duration-500"
          >

          <span class="absolute top-4 left-4 bg-green-500 text-
white text-xs font-bold px-3 py-1 rounded-full">
            Best Seller
          </span>

          <button class="absolute top-4 right-4 w-10 h-10 bg-
white rounded-full shadow-md flex items-center justify-center
text-gray-400 hover:text-red-500 transition-colors">
            <svg class="w-5 h-5" fill="none"
stroke="currentColor" viewBox="0 0 24 24">
              <path stroke-linecap="round" stroke-
linejoin="round" stroke-width="2" d="M4.318 6.318a4.5 4.5 0 000
6.364L12 20.364l7.682-7.682a4.5 4.5 0 00-6.364-6.364L12 7.636l-
1.318-1.318a4.5 4.5 0 00-6.364 0z"></path>
            </svg>
          </button>
        </div>

        <div class="p-6">
          <div class="text-xs text-gray-500 uppercase tracking-
wider mb-2">
            Accessories
          </div>

          <h3 class="text-lg font-semibold text-gray-900 mb-2">
```

```
            Premium Watch
          </h3>

          <div class="flex items-center mb-3">
            <div class="flex text-yellow-400">
              <svg class="w-4 h-4" fill="currentColor" viewBox="0
0 20 20"><path d="M9.049 2.927c.3-.921 1.603-.921 1.902 0l1.07
3.292a1 1 0 00.95.69h3.462c.969 0 1.371 1.24.588 1.81l-2.8
2.034a1 1 0 00-.364 1.118l1.07 3.292c.3.921-.755 1.688-1.54
1.118l-2.8-2.034a1 1 0 00-1.175 0l-2.8 2.034c-.784.57-1.838-.197-
1.539-1.118l1.07-3.292a1 1 0 00-.364-1.118L2.98
8.72c-.783-.57-.38-1.81.588-1.81h3.461a1 1 0 00.951-.69l1.07-
3.292z"></path></svg>
              <svg class="w-4 h-4" fill="currentColor" viewBox="0
0 20 20"><path d="M9.049 2.927c.3-.921 1.603-.921 1.902 0l1.07
3.292a1 1 0 00.95.69h3.462c.969 0 1.371 1.24.588 1.81l-2.8
2.034a1 1 0 00-.364 1.118l1.07 3.292c.3.921-.755 1.688-1.54
1.118l-2.8-2.034a1 1 0 00-1.175 0l-2.8 2.034c-.784.57-1.838-.197-
1.539-1.118l1.07-3.292a1 1 0 00-.364-1.118L2.98
8.72c-.783-.57-.38-1.81.588-1.81h3.461a1 1 0 00.951-.69l1.07-
3.292z"></path></svg>
              <svg class="w-4 h-4" fill="currentColor" viewBox="0
0 20 20"><path d="M9.049 2.927c.3-.921 1.603-.921 1.902 0l1.07
3.292a1 1 0 00.95.69h3.462c.969 0 1.371 1.24.588 1.81l-2.8
2.034a1 1 0 00-.364 1.118l1.07 3.292c.3.921-.755 1.688-1.54
1.118l-2.8-2.034a1 1 0 00-1.175 0l-2.8 2.034c-.784.57-1.838-.197-
1.539-1.118l1.07-3.292a1 1 0 00-.364-1.118L2.98
8.72c-.783-.57-.38-1.81.588-1.81h3.461a1 1 0 00.951-.69l1.07-
3.292z"></path></svg>
              <svg class="w-4 h-4" fill="currentColor" viewBox="0
0 20 20"><path d="M9.049 2.927c.3-.921 1.603-.921 1.902 0l1.07
3.292a1 1 0 00.95.69h3.462c.969 0 1.371 1.24.588 1.81l-2.8
2.034a1 1 0 00-.364 1.118l1.07 3.292c.3.921-.755 1.688-1.54
1.118l-2.8-2.034a1 1 0 00-1.175 0l-2.8 2.034c-.784.57-1.838-.197-
1.539-1.118l1.07-3.292a1 1 0 00-.364-1.118L2.98
```

```
8.72c-.783-.57-.38-1.81.588-1.81h3.461a1 1 0 00.951-.69l1.07-
3.292z"></path></svg>
              <svg class="w-4 h-4 text-gray-300"
fill="currentColor" viewBox="0 0 20 20"><path d="M9.049
2.927c.3-.921 1.603-.921 1.902 0l1.07 3.292a1 1 0
00.95.69h3.462c.969 0 1.371 1.24.588 1.81l-2.8 2.034a1 1 0
00-.364 1.118l1.07 3.292c.3.921-.755 1.688-1.54 1.118l-2.8-
2.034a1 1 0 00-1.175 0l-2.8 2.034c-.784.57-1.838-.197-1.539-
1.118l1.07-3.292a1 1 0 00-.364-1.118L2.98 8.72c-.783-.57-.38-
1.81.588-1.81h3.461a1 1 0 00.951-.69l1.07-3.292z"></path></svg>
            </div>
            <span class="text-sm text-gray-600 ml-2">(89 reviews)
</span>
          </div>

          <!-- Color Selection -->
          <div class="mb-4">
            <p class="text-sm text-gray-700 font-medium mb-
2">Color:</p>
            <div class="flex space-x-2">
              <button class="w-8 h-8 rounded-full bg-black
border-2 border-blue-600 focus:outline-none"></button>
              <button class="w-8 h-8 rounded-full bg-gray-400
border-2 border-transparent hover:border-gray-600 focus:outline-
none"></button>
              <button class="w-8 h-8 rounded-full bg-blue-500
border-2 border-transparent hover:border-blue-600 focus:outline-
none"></button>
              <button class="w-8 h-8 rounded-full bg-rose-400
border-2 border-transparent hover:border-rose-600 focus:outline-
none"></button>
            </div>
          </div>

          <div class="flex items-baseline space-x-2 mb-4">
```

```
            <span class="text-2xl font-bold text-gray-
900">$299.00</span>
          </div>

          <div class="flex space-x-2">
            <button class="flex-1 bg-blue-600 hover:bg-blue-700
text-white py-3 rounded-lg font-semibold transition">
              Buy Now
            </button>
            <button class="px-4 py-3 border-2 border-blue-600
text-blue-600 hover:bg-blue-50 rounded-lg transition">
              <svg class="w-5 h-5" fill="none"
stroke="currentColor" viewBox="0 0 24 24">
                <path stroke-linecap="round" stroke-
linejoin="round" stroke-width="2" d="M3 3h2l.4 2M7 13h10l4-
8H5.4M7 13L5.4 5M7 13l-2.293 2.293c-.63.63-.184 1.707.707
1.707H17m0 0a2 2 0 100 4 2 2 0 000-4zm-8 2a2 2 0 11-4 0 2 2 0 014
0z"></path>
              </svg>
            </button>
          </div>

        </div>

      </div>

      <!-- Product Card 3: Out of Stock -->
      <div class="bg-white rounded-xl shadow-md overflow-hidden
group hover:shadow-xl transition-shadow">

        <div class="relative overflow-hidden bg-gray-100">
          <img
            src="https://images.unsplash.com/photo-1560343090-
f0409e92791a?w=400&h=400&fit=crop"
            alt="Product"
```

```
            class="w-full h-64 object-cover transform group-
hover:scale-110 transition-transform duration-500 opacity-75"
          >

          <!-- Out of Stock Overlay -->
          <div class="absolute inset-0 bg-black bg-opacity-40
flex items-center justify-center">
            <span class="bg-gray-900 text-white text-lg font-bold
px-6 py-3 rounded-lg">
              Out of Stock
            </span>
          </div>

          <button class="absolute top-4 right-4 w-10 h-10 bg-
white rounded-full shadow-md flex items-center justify-center
text-gray-400 hover:text-red-500 transition-colors">
            <svg class="w-5 h-5" fill="none"
stroke="currentColor" viewBox="0 0 24 24">
              <path stroke-linecap="round" stroke-
linejoin="round" stroke-width="2" d="M4.318 6.318a4.5 4.5 0 000
6.364L12 20.364l7.682-7.682a4.5 4.5 0 00-6.364-6.364L12 7.636l-
1.318-1.318a4.5 4.5 0 00-6.364 0z"></path>
            </svg>
          </button>
        </div>

        <div class="p-6">
          <div class="text-xs text-gray-500 uppercase tracking-
wider mb-2">
            Footwear
          </div>

          <h3 class="text-lg font-semibold text-gray-900 mb-2">
            Running Shoes
          </h3>
```

```
          <div class="flex items-center mb-3">
            <div class="flex text-yellow-400">
              <svg class="w-4 h-4" fill="currentColor" viewBox="0
0 20 20"><path d="M9.049 2.927c.3-.921 1.603-.921 1.902 0l1.07
3.292a1 1 0 00.95.69h3.462c.969 0 1.371 1.24.588 1.81l-2.8
2.034a1 1 0 00-.364 1.118l1.07 3.292c.3.921-.755 1.688-1.54
1.118l-2.8-2.034a1 1 0 00-1.175 0l-2.8 2.034c-.784.57-1.838-.197-
1.539-1.118l1.07-3.292a1 1 0 00-.364-1.118L2.98
8.72c-.783-.57-.38-1.81.588-1.81h3.461a1 1 0 00.951-.69l1.07-
3.292z"></path></svg>
              <svg class="w-4 h-4" fill="currentColor" viewBox="0
0 20 20"><path d="M9.049 2.927c.3-.921 1.603-.921 1.902 0l1.07
3.292a1 1 0 00.95.69h3.462c.969 0 1.371 1.24.588 1.81l-2.8
2.034a1 1 0 00-.364 1.118l1.07 3.292c.3.921-.755 1.688-1.54
1.118l-2.8-2.034a1 1 0 00-1.175 0l-2.8 2.034c-.784.57-1.838-.197-
1.539-1.118l1.07-3.292a1 1 0 00-.364-1.118L2.98
8.72c-.783-.57-.38-1.81.588-1.81h3.461a1 1 0 00.951-.69l1.07-
3.292z"></path></svg>
              <svg class="w-4 h-4" fill="currentColor" viewBox="0
0 20 20"><path d="M9.049 2.927c.3-.921 1.603-.921 1.902 0l1.07
3.292a1 1 0 00.95.69h3.462c.969 0 1.371 1.24.588 1.81l-2.8
2.034a1 1 0 00-.364 1.118l1.07 3.292c.3.921-.755 1.688-1.54
1.118l-2.8-2.034a1 1 0 00-1.175 0l-2.8 2.034c-.784.57-1.838-.197-
1.539-1.118l1.07-3.292a1 1 0 00-.364-1.118L2.98
8.72c-.783-.57-.38-1.81.588-1.81h3.461a1 1 0 00.951-.69l1.07-
3.292z"></path></svg>
              <svg class="w-4 h-4" fill="currentColor" viewBox="0
0 20 20"><path d="M9.049 2.927c.3-.921 1.603-.921 1.902 0l1.07
3.292a1 1 0 00.95.69h3.462c.969 0 1.371 1.24.588 1.81l-2.8
2.034a1 1 0 00-.364 1.118l1.07 3.292c.3.921-.755 1.688-1.54
1.118l-2.8-2.034a1 1 0 00-1.175 0l-2.8 2.034c-.784.57-1.838-.197-
1.539-1.118l1.07-3.292a1 1 0 00-.364-1.118L2.98
8.72c-.783-.57-.38-1.81.588-1.81h3.461a1 1 0 00.951-.69l1.07-
3.292z"></path></svg>
```

```
              <svg class="w-4 h-4" fill="currentColor" viewBox="0
0 20 20"><path d="M9.049 2.927c.3-.921 1.603-.921 1.902 0l1.07
3.292a1 1 0 00.95.69h3.462c.969 0 1.371 1.24.588 1.81l-2.8
2.034a1 1 0 00-.364 1.118l1.07 3.292c.3.921-.755 1.688-1.54
1.118l-2.8-2.034a1 1 0 00-1.175 0l-2.8 2.034c-.784.57-1.838-.197-
1.539-1.118l1.07-3.292a1 1 0 00-.364-1.118L2.98
8.72c-.783-.57-.38-1.81.588-1.81h3.461a1 1 0 00.951-.69l1.07-
3.292z"></path></svg>
            </div>
            <span class="text-sm text-gray-600 ml-2">(245
reviews)</span>
          </div>

          <div class="flex items-baseline space-x-2 mb-4">
            <span class="text-2xl font-bold text-gray-
900">$129.99</span>
          </div>

          <button class="w-full bg-gray-300 text-gray-500 py-3
rounded-lg font-semibold cursor-not-allowed" disabled>
            Currently Unavailable
          </button>

          <button class="w-full mt-2 border-2 border-blue-600
text-blue-600 hover:bg-blue-50 py-3 rounded-lg font-semibold
transition">
            Notify When Available
          </button>

        </div>

      </div>

    </div>
```

```
    <!-- Detailed Product View -->
    <h2 class="text-2xl font-bold text-gray-900 mb-6">Detailed
Product View</h2>

    <div class="bg-white rounded-xl shadow-lg overflow-hidden">
      <div class="grid grid-cols-1 lg:grid-cols-2 gap-8 p-8">

        <!-- Image Gallery -->
        <div>
          <!-- Main Image -->
          <div class="mb-4 rounded-xl overflow-hidden bg-gray-
100">
            <img
              id="main-image"
              src="https://images.unsplash.com/photo-
1505740420928-5e560c06d30e?w=600&h=600&fit=crop"
              alt="Product"
              class="w-full h-96 object-cover"
            >
          </div>

          <!-- Thumbnail Gallery -->
          <div class="grid grid-cols-4 gap-4">
            <button class="rounded-lg overflow-hidden border-2
border-blue-600 focus:outline-none" onclick="changeImage(this)">
              <img src="https://images.unsplash.com/photo-
1505740420928-5e560c06d30e?w=150&h=150&fit=crop" alt="Thumb"
class="w-full h-24 object-cover">
            </button>
            <button class="rounded-lg overflow-hidden border-2
border-transparent hover:border-gray-300 focus:outline-none"
onclick="changeImage(this)">
              <img src="https://images.unsplash.com/photo-
1572635196237-14b3f281503f?w=150&h=150&fit=crop" alt="Thumb"
class="w-full h-24 object-cover">
```

```
            </button>
            <button class="rounded-lg overflow-hidden border-2 border-transparent hover:border-gray-300 focus:outline-none" onclick="changeImage(this)">
              <img src="https://images.unsplash.com/photo-1523275335684-37898b6baf30?w=150&h=150&fit=crop" alt="Thumb" class="w-full h-24 object-cover">
            </button>
            <button class="rounded-lg overflow-hidden border-2 border-transparent hover:border-gray-300 focus:outline-none" onclick="changeImage(this)">
              <img src="https://images.unsplash.com/photo-1560343090-f0409e92791a?w=150&h=150&fit=crop" alt="Thumb" class="w-full h-24 object-cover">
            </button>
          </div>
        </div>

        <!-- Product Details -->
        <div>

          <!-- Brand & Title -->
          <div class="mb-4">
            <p class="text-sm text-blue-600 font-semibold mb-2">Premium Brand</p>
            <h1 class="text-3xl font-bold text-gray-900 mb-2">
              Wireless Noise-Cancelling Headphones
            </h1>
            <p class="text-gray-600">
              Experience premium sound quality with our latest wireless headphones
            </p>
          </div>

          <!-- Rating -->
```

```
          <div class="flex items-center mb-6">
            <div class="flex text-yellow-400 mr-2">
              <svg class="w-5 h-5" fill="currentColor" viewBox="0
0 20 20"><path d="M9.049 2.927c.3-.921 1.603-.921 1.902 0l1.07
3.292a1 1 0 00.95.69h3.462c.969 0 1.371 1.24.588 1.81l-2.8
2.034a1 1 0 00-.364 1.118l1.07 3.292c.3.921-.755 1.688-1.54
1.118l-2.8-2.034a1 1 0 00-1.175 0l-2.8 2.034c-.784.57-1.838-.197-
1.539-1.118l1.07-3.292a1 1 0 00-.364-1.118L2.98
8.72c-.783-.57-.38-1.81.588-1.81h3.461a1 1 0 00.951-.69l1.07-
3.292z"></path></svg>
              <svg class="w-5 h-5" fill="currentColor" viewBox="0
0 20 20"><path d="M9.049 2.927c.3-.921 1.603-.921 1.902 0l1.07
3.292a1 1 0 00.95.69h3.462c.969 0 1.371 1.24.588 1.81l-2.8
2.034a1 1 0 00-.364 1.118l1.07 3.292c.3.921-.755 1.688-1.54
1.118l-2.8-2.034a1 1 0 00-1.175 0l-2.8 2.034c-.784.57-1.838-.197-
1.539-1.118l1.07-3.292a1 1 0 00-.364-1.118L2.98
8.72c-.783-.57-.38-1.81.588-1.81h3.461a1 1 0 00.951-.69l1.07-
3.292z"></path></svg>
              <svg class="w-5 h-5" fill="currentColor" viewBox="0
0 20 20"><path d="M9.049 2.927c.3-.921 1.603-.921 1.902 0l1.07
3.292a1 1 0 00.95.69h3.462c.969 0 1.371 1.24.588 1.81l-2.8
2.034a1 1 0 00-.364 1.118l1.07 3.292c.3.921-.755 1.688-1.54
1.118l-2.8-2.034a1 1 0 00-1.175 0l-2.8 2.034c-.784.57-1.838-.197-
1.539-1.118l1.07-3.292a1 1 0 00-.364-1.118L2.98
8.72c-.783-.57-.38-1.81.588-1.81h3.461a1 1 0 00.951-.69l1.07-
3.292z"></path></svg>
              <svg class="w-5 h-5" fill="currentColor" viewBox="0
0 20 20"><path d="M9.049 2.927c.3-.921 1.603-.921 1.902 0l1.07
3.292a1 1 0 00.95.69h3.462c.969 0 1.371 1.24.588 1.81l-2.8
2.034a1 1 0 00-.364 1.118l1.07 3.292c.3.921-.755 1.688-1.54
1.118l-2.8-2.034a1 1 0 00-1.175 0l-2.8 2.034c-.784.57-1.838-.197-
1.539-1.118l1.07-3.292a1 1 0 00-.364-1.118L2.98
8.72c-.783-.57-.38-1.81.588-1.81h3.461a1 1 0 00.951-.69l1.07-
3.292z"></path></svg>
              <svg class="w-5 h-5" fill="currentColor" viewBox="0
```

```
0 20 20"><path d="M9.049 2.927c.3-.921 1.603-.921 1.902 0l1.07
3.292a1 1 0 00.95.69h3.462c.969 0 1.371 1.24.588 1.81l-2.8
2.034a1 1 0 00-.364 1.118l1.07 3.292c.3.921-.755 1.688-1.54
1.118l-2.8-2.034a1 1 0 00-1.175 0l-2.8 2.034c-.784.57-1.838-.197-
1.539-1.118l1.07-3.292a1 1 0 00-.364-1.118L2.98
8.72c-.783-.57-.38-1.81.588-1.81h3.461a1 1 0 00.951-.69l1.07-
3.292z"></path></svg>
            </div>
            <span class="text-gray-900 font-semibold">4.8</span>
            <span class="text-gray-600 ml-2">(432 reviews)</span>
          </div>

          <!-- Price -->
          <div class="mb-6 pb-6 border-b border-gray-200">
            <div class="flex items-baseline space-x-3">
              <span class="text-4xl font-bold text-gray-
900">$249.99</span>
              <span class="text-xl text-gray-500 line-
through">$349.99</span>
              <span class="bg-red-100 text-red-800 text-sm font-
bold px-3 py-1 rounded-full">
                Save 29%
              </span>
            </div>
            <p class="text-green-600 font-medium mt-2 flex items-
center">
              <svg class="w-5 h-5 mr-1" fill="currentColor"
viewBox="0 0 20 20">
                <path fill-rule="evenodd" d="M10 18a8 8 0 100-16
8 8 0 000 16zm3.707-9.293a1 1 0 00-1.414-1.414L9 10.586 7.707
9.293a1 1 0 00-1.414 1.414l2 2a1 1 0 001.414 0l4-4z" clip-
rule="evenodd"></path>
              </svg>
              In stock - Ships within 24 hours
            </p>
```

```
          </div>

          <!-- Color Selection -->
          <div class="mb-6">
            <label class="block text-sm font-semibold text-gray-
900 mb-3">
              Color: <span class="font-normal text-gray-
600">Black</span>
            </label>
            <div class="flex space-x-3">
              <button class="w-12 h-12 rounded-lg bg-black
border-2 border-blue-600 focus:outline-none relative">
                <svg class="absolute inset-0 m-auto w-6 h-6 text-
white" fill="currentColor" viewBox="0 0 20 20">
                  <path fill-rule="evenodd" d="M16.707 5.293a1 1
0 010 1.414l-8 8a1 1 0 01-1.414 0l-4-4a1 1 0 011.414-1.414L8
12.586l7.293-7.293a1 1 0 011.414 0z" clip-rule="evenodd"></path>
                </svg>
              </button>
              <button class="w-12 h-12 rounded-lg bg-white
border-2 border-gray-300 hover:border-gray-400 focus:outline-
none"></button>
              <button class="w-12 h-12 rounded-lg bg-blue-500
border-2 border-gray-300 hover:border-gray-400 focus:outline-
none"></button>
              <button class="w-12 h-12 rounded-lg bg-red-500
border-2 border-gray-300 hover:border-gray-400 focus:outline-
none"></button>
            </div>
          </div>

          <!-- Quantity Selector -->
          <div class="mb-6">
            <label class="block text-sm font-semibold text-gray-
900 mb-3">
```

```
              Quantity
            </label>
            <div class="flex items-center space-x-4">
              <div class="flex items-center border-2 border-gray-
300 rounded-lg">
                <button class="px-4 py-2 text-gray-600 hover:bg-
gray-100 transition">
                  <svg class="w-4 h-4" fill="none"
stroke="currentColor" viewBox="0 0 24 24">
                    <path stroke-linecap="round" stroke-
linejoin="round" stroke-width="2" d="M20 12H4"></path>
                  </svg>
                </button>
                <input type="text" value="1" class="w-12 text-
center font-semibold focus:outline-none">
                <button class="px-4 py-2 text-gray-600 hover:bg-
gray-100 transition">
                  <svg class="w-4 h-4" fill="none"
stroke="currentColor" viewBox="0 0 24 24">
                    <path stroke-linecap="round" stroke-
linejoin="round" stroke-width="2" d="M12 4v16m8-8H4"></path>
                  </svg>
                </button>
              </div>
              <span class="text-sm text-gray-600">Only 12 items
left</span>
            </div>
          </div>

          <!-- Action Buttons -->
          <div class="flex space-x-4 mb-6">
            <button class="flex-1 bg-blue-600 hover:bg-blue-700
text-white py-4 rounded-lg font-semibold transition flex items-
center justify-center space-x-2">
              <svg class="w-5 h-5" fill="none"
```

```
stroke="currentColor" viewBox="0 0 24 24">
                <path stroke-linecap="round" stroke-
linejoin="round" stroke-width="2" d="M3 3h2l.4 2M7 13h10l4-
8H5.4M7 13L5.4 5M7 13l-2.293 2.293c-.63.63-.184 1.707.707
1.707H17m0 0a2 2 0 100 4 2 2 0 000-4zm-8 2a2 2 0 11-4 0 2 2 0 014
0z"></path>
              </svg>
              <span>Add to Cart</span>
            </button>
            <button class="px-6 py-4 border-2 border-gray-300
hover:border-red-500 text-gray-700 hover:text-red-500 rounded-lg
transition">
              <svg class="w-6 h-6" fill="none"
stroke="currentColor" viewBox="0 0 24 24">
                <path stroke-linecap="round" stroke-
linejoin="round" stroke-width="2" d="M4.318 6.318a4.5 4.5 0 000
6.364L12 20.364l7.682-7.682a4.5 4.5 0 00-6.364-6.364L12 7.636l-
1.318-1.318a4.5 4.5 0 00-6.364 0z"></path>
              </svg>
            </button>
          </div>

          <button class="w-full bg-gray-900 hover:bg-gray-800
text-white py-4 rounded-lg font-semibold transition mb-6">
            Buy Now
          </button>

          <!-- Features -->
          <div class="space-y-3 border-t border-gray-200 pt-6">
            <div class="flex items-center text-sm text-gray-700">
              <svg class="w-5 h-5 text-green-500 mr-3"
fill="currentColor" viewBox="0 0 20 20">
                <path fill-rule="evenodd" d="M10 18a8 8 0 100-16
8 8 0 000 16zm3.707-9.293a1 1 0 00-1.414-1.414L9 10.586 7.707
9.293a1 1 0 00-1.414 1.414l2 2a1 1 0 001.414 0l4-4z" clip-
```

```
rule="evenodd"></path>
              </svg>
              Free shipping on orders over $50
            </div>
            <div class="flex items-center text-sm text-gray-700">
              <svg class="w-5 h-5 text-green-500 mr-3"
fill="currentColor" viewBox="0 0 20 20">
                <path fill-rule="evenodd" d="M10 18a8 8 0 100-16
8 8 0 000 16zm3.707-9.293a1 1 0 00-1.414-1.414L9 10.586 7.707
9.293a1 1 0 00-1.414 1.414l2 2a1 1 0 001.414 0l4-4z" clip-
rule="evenodd"></path>
              </svg>
              30-day money-back guarantee
            </div>
            <div class="flex items-center text-sm text-gray-700">
              <svg class="w-5 h-5 text-green-500 mr-3"
fill="currentColor" viewBox="0 0 20 20">
                <path fill-rule="evenodd" d="M10 18a8 8 0 100-16
8 8 0 000 16zm3.707-9.293a1 1 0 00-1.414-1.414L9 10.586 7.707
9.293a1 1 0 00-1.414 1.414l2 2a1 1 0 001.414 0l4-4z" clip-
rule="evenodd"></path>
              </svg>
              2-year warranty included
            </div>
            <div class="flex items-center text-sm text-gray-700">
              <svg class="w-5 h-5 text-green-500 mr-3"
fill="currentColor" viewBox="0 0 20 20">
                <path fill-rule="evenodd" d="M10 18a8 8 0 100-16
8 8 0 000 16zm3.707-9.293a1 1 0 00-1.414-1.414L9 10.586 7.707
9.293a1 1 0 00-1.414 1.414l2 2a1 1 0 001.414 0l4-4z" clip-
rule="evenodd"></path>
              </svg>
              24/7 customer support
            </div>
          </div>
```

```
        </div>

      </div>

      <!-- Product Description Tabs -->
      <div class="border-t border-gray-200">
        <div class="flex border-b border-gray-200">
          <button class="px-8 py-4 text-sm font-semibold text-
blue-600 border-b-2 border-blue-600">
            Description
          </button>
          <button class="px-8 py-4 text-sm font-semibold text-
gray-600 hover:text-gray-900">
            Specifications
          </button>
          <button class="px-8 py-4 text-sm font-semibold text-
gray-600 hover:text-gray-900">
            Reviews (432)
          </button>
        </div>

        <div class="p-8">
          <h3 class="text-lg font-semibold text-gray-900 mb-
4">Product Description</h3>
          <div class="prose prose-gray max-w-none">
            <p class="text-gray-700 mb-4">
              Experience unparalleled audio quality with our
premium wireless noise-cancelling headphones.
              Engineered for audiophiles and everyday users
alike, these headphones deliver crystal-clear
              sound across all frequencies.
            </p>
            <p class="text-gray-700 mb-4">
              With advanced active noise cancellation technology,
```

```
you can immerse yourself in your music
              without any distractions. The 40-hour battery life
ensures you can enjoy your favorite tracks
              all day long.
            </p>
            <h4 class="text-md font-semibold text-gray-900 mb-
2">Key Features:</h4>
            <ul class="list-disc list-inside text-gray-700 space-
y-2 mb-4">
              <li>Premium 40mm drivers for exceptional sound
quality</li>
              <li>Active noise cancellation (ANC) technology</li>
              <li>40-hour battery life with quick charge
support</li>
              <li>Comfortable memory foam ear cushions</li>
              <li>Bluetooth 5.0 with multi-device pairing</li>
              <li>Built-in microphone for hands-free calls</li>
              <li>Foldable design with premium carrying case</li>
            </ul>
          </div>
        </div>
      </div>

    </div>

  </div>

  <!-- JavaScript -->
  <script>
    function changeImage(button) {
      const mainImage = document.getElementById('main-image');
      const newSrc =
button.querySelector('img').src.replace('w=150&h=150',
'w=600&h=600');
      mainImage.src = newSrc;
```

```
      // Update border styling

document.querySelectorAll('[onclick="changeImage(this)"]').forEach(b
=> {
        btn.classList.remove('border-blue-600');
        btn.classList.add('border-transparent', 'hover:border-
gray-300');
      });
      button.classList.add('border-blue-600');
      button.classList.remove('border-transparent',
'hover:border-gray-300');
    }
  </script>

</body>
</html>
```

This e-commerce example demonstrates:

✅ **Product card variations** (standard, with colors, out of stock)
✅ **Badges and labels** (sale, new, best seller)
✅ **Wishlist functionality** (heart icon)
✅ **Quick view on hover**
✅ **Image gallery with thumbnails**
✅ **Color selection**
✅ **Quantity selector**
✅ **Rating stars**
✅ **Price display** with original/sale price
✅ **Stock status indicators**
✅ **Product tabs** (description, specs, reviews)
✅ **Multiple CTAs** (add to cart, buy now, wishlist)
✅ **Trust badges** (free shipping, warranty, etc.)

Would you like me to continue with **Project 4: Blog Post Layout** next?

1/7/2026, 1:15:14 PM

Project 4: Blog Post Layout

A complete blog layout with article header, content, sidebar, author bio, comments, and related posts.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Blog Post - Tailwind CSS Guide</title>
  <script src="https://cdn.tailwindcss.com"></script>
</head>
<body class="bg-gray-50">

  <!-- Navigation -->
  <nav class="bg-white shadow-sm sticky top-0 z-50">
    <div class="max-w-7xl mx-auto px-4 sm:px-6 lg:px-8">
      <div class="flex justify-between items-center h-16">
        <div class="flex items-center space-x-8">
          <h1 class="text-xl font-bold text-gray-
900">DevBlog</h1>
          <div class="hidden md:flex space-x-6">
            <a href="#" class="text-gray-700 hover:text-blue-600
transition">Home</a>
            <a href="#" class="text-gray-700 hover:text-blue-600
transition">Articles</a>
            <a href="#" class="text-gray-700 hover:text-blue-600
transition">Categories</a>
            <a href="#" class="text-gray-700 hover:text-blue-600
```

```
transition">About</a>
          </div>
        </div>
        <div class="flex items-center space-x-4">
          <button class="hidden sm:block text-gray-700
hover:text-blue-600">
            <svg class="w-5 h-5" fill="none"
stroke="currentColor" viewBox="0 0 24 24">
              <path stroke-linecap="round" stroke-
linejoin="round" stroke-width="2" d="M21 21l-6-6m2-5a7 7 0 11-14
0 7 7 0 0114 0z"></path>
            </svg>
          </button>
          <button class="bg-blue-600 hover:bg-blue-700 text-white
px-4 py-2 rounded-lg text-sm font-medium transition">
            Subscribe
          </button>
        </div>
      </div>
    </div>
  </nav>

  <!-- Hero Section -->
  <div class="bg-gradient-to-br from-blue-50 to-indigo-100 py-
12">
    <div class="max-w-4xl mx-auto px-4 sm:px-6 lg:px-8">

      <!-- Breadcrumb -->
      <nav class="flex items-center space-x-2 text-sm text-gray-
600 mb-6">
        <a href="#" class="hover:text-blue-600">Home</a>
        <svg class="w-4 h-4" fill="currentColor" viewBox="0 0 20
20">
          <path fill-rule="evenodd" d="M7.293 14.707a1 1 0 010-
1.414L10.586 10 7.293 6.707a1 1 0 011.414-1.414l4 4a1 1 0 010
```

```
1.414l-4 4a1 1 0 01-1.414 0z" clip-rule="evenodd"></path>
        </svg>
        <a href="#" class="hover:text-blue-600">Web
Development</a>
        <svg class="w-4 h-4" fill="currentColor" viewBox="0 0 20
20">
          <path fill-rule="evenodd" d="M7.293 14.707a1 1 0 010-
1.414L10.586 10 7.293 6.707a1 1 0 011.414-1.414l4 4a1 1 0 010
1.414l-4 4a1 1 0 01-1.414 0z" clip-rule="evenodd"></path>
        </svg>
        <span class="text-gray-900">Article</span>
      </nav>

      <!-- Category Badge -->
      <div class="mb-4">
        <span class="inline-block px-3 py-1 bg-blue-600 text-
white text-xs font-semibold rounded-full uppercase tracking-
wide">
          Web Development
        </span>
      </div>

      <!-- Title -->
      <h1 class="text-4xl sm:text-5xl font-bold text-gray-900 mb-
4 leading-tight">
        Mastering Tailwind CSS: A Complete Guide to Building
Modern UIs
      </h1>

      <!-- Meta Info -->
      <div class="flex flex-wrap items-center space-x-6 text-
gray-600">
        <div class="flex items-center space-x-2">
          <img src="https://i.pravatar.cc/40?img=12" class="w-10
h-10 rounded-full" alt="Author">
```

```
          <div>
            <p class="text-sm font-medium text-gray-900">Sarah
Johnson</p>
            <p class="text-xs">Senior Developer</p>
          </div>
        </div>
        <div class="flex items-center space-x-1">
          <svg class="w-4 h-4" fill="none" stroke="currentColor"
viewBox="0 0 24 24">
            <path stroke-linecap="round" stroke-linejoin="round"
stroke-width="2" d="M8 7V3m8 4V3m-9 8h10M5 21h14a2 2 0 002-2V7a2
2 0 00-2-2H5a2 2 0 00-2 2v12a2 2 0 002 2z"></path>
          </svg>
          <span class="text-sm">January 7, 2026</span>
        </div>
        <div class="flex items-center space-x-1">
          <svg class="w-4 h-4" fill="none" stroke="currentColor"
viewBox="0 0 24 24">
            <path stroke-linecap="round" stroke-linejoin="round"
stroke-width="2" d="M12 8v4l3 3m6-3a9 9 0 11-18 0 9 9 0 0118 0z">
</path>
          </svg>
          <span class="text-sm">12 min read</span>
        </div>
      </div>

    </div>
  </div>

  <!-- Main Content -->
  <div class="max-w-7xl mx-auto px-4 sm:px-6 lg:px-8 py-12">
    <div class="grid grid-cols-1 lg:grid-cols-3 gap-8">

      <!-- Article Content -->
      <article class="lg:col-span-2">
```

```
        <!-- Featured Image -->
        <div class="mb-8 rounded-xl overflow-hidden shadow-lg">
          <img
            src="https://images.unsplash.com/photo-1633356122544-
f134324a6cee?w=1200&h=600&fit=crop"
            alt="Featured"
            class="w-full h-96 object-cover"
          >
        </div>

        <!-- Social Share (Sticky) -->
        <div class="hidden lg:block fixed left-8 top-1/3">
          <div class="flex flex-col space-y-4">
            <button class="w-10 h-10 bg-blue-600 hover:bg-blue-
700 text-white rounded-full flex items-center justify-center
transition">
              <svg class="w-5 h-5" fill="currentColor" viewBox="0
0 24 24">
                <path d="M24 12.073c0-6.627-5.373-12-12-12s-12
5.373-12 12c0 5.99 4.388 10.954 10.125 11.854v-8.385H7.078v-
3.47h3.047V9.43c0-3.007 1.792-4.669 4.533-4.669 1.312 0 2.686.235
2.686.235v2.953H15.83c-1.491 0-1.956.925-1.956
1.874v2.25h3.328l-.532 3.47h-2.796v8.385C19.612 23.027 24 18.062
24 12.073z"/>
              </svg>
            </button>
            <button class="w-10 h-10 bg-blue-400 hover:bg-blue-
500 text-white rounded-full flex items-center justify-center
transition">
              <svg class="w-5 h-5" fill="currentColor" viewBox="0
0 24 24">
                <path d="M23.953 4.57a10 10 0 01-2.825.775 4.958
4.958 0 002.163-2.723c-.951.555-2.005.959-3.127 1.184a4.92 4.92 0
00-8.384 4.482C7.69 8.095 4.067 6.13 1.64 3.162a4.822 4.822 0
```

```
00-.666 2.475c0 1.71.87 3.213 2.188 4.096a4.904 4.904 0 01-
2.228-.616v.06a4.923 4.923 0 003.946 4.827 4.996 4.996 0 01-
2.212.085 4.936 4.936 0 004.604 3.417 9.867 9.867 0 01-6.102
2.105c-.39 0-.779-.023-1.17-.067a13.995 13.995 0 007.557
2.209c9.053 0 13.998-7.496 13.998-13.985 0-.21
0-.42-.015-.63A9.935 9.935 0 0024 4.59z"/>
              </svg>
            </button>
            <button class="w-10 h-10 bg-blue-700 hover:bg-blue-
800 text-white rounded-full flex items-center justify-center
transition">
              <svg class="w-5 h-5" fill="currentColor" viewBox="0
0 24 24">
                <path d="M20.447 20.452h-3.554v-5.569c0-
1.328-.027-3.037-1.852-3.037-1.853 0-2.136 1.445-2.136
2.939v5.667H9.351V9h3.414v1.561h.046c.477-.9 1.637-1.85 3.37-1.85
3.601 0 4.267 2.37 4.267 5.455v6.286zM5.337 7.433c-1.144 0-
2.063-.926-2.063-2.065 0-1.138.92-2.063 2.063-2.063 1.14 0
2.064.925 2.064 2.063 0 1.139-.925 2.065-2.064 2.065zm1.782
13.019H3.555V9h3.564v11.452zM22.225 0H1.771C.792 0 0 .774 0
1.729v20.542C0 23.227.792 24 1.771 24h20.451C23.2 24 24 23.227 24
22.271V1.729C24 .774 23.2 0 22.222 0h.003z"/>
              </svg>
            </button>
            <button class="w-10 h-10 bg-gray-200 hover:bg-gray-
300 text-gray-700 rounded-full flex items-center justify-center
transition">
              <svg class="w-5 h-5" fill="none"
stroke="currentColor" viewBox="0 0 24 24">
                <path stroke-linecap="round" stroke-
linejoin="round" stroke-width="2" d="M8.684 13.342C8.886 12.938 9
12.482 9 12c0-.482-.114-.938-.316-1.342m0 2.684a3 3 0 110-2.684m0
2.684l6.632 3.316m-6.632-6l6.632-3.316m0 0a3 3 0 105.367-2.684 3
3 0 00-5.367 2.684zm0 9.316a3 3 0 105.368 2.684 3 3 0 00-5.368-
2.684z"></path>
```

```
              </svg>
            </button>
          </div>
        </div>

        <!-- Article Body -->
        <div class="bg-white rounded-xl shadow-lg p-8 lg:p-12">

          <!-- Lead Paragraph -->
          <p class="text-xl text-gray-700 leading-relaxed mb-8
font-serif">
            Tailwind CSS has revolutionized the way developers
approach styling web applications.
            In this comprehensive guide, we'll explore everything
you need to know to master
            this powerful utility-first framework.
          </p>

          <!-- Table of Contents -->
          <div class="bg-blue-50 border-l-4 border-blue-600 p-6
mb-8 rounded-r-lg">
            <h3 class="text-lg font-bold text-gray-900 mb-
4">Table of Contents</h3>
            <ol class="space-y-2 text-blue-600">
              <li><a href="#section1" class="hover:underline">1.
Introduction to Tailwind CSS</a></li>
              <li><a href="#section2" class="hover:underline">2.
Core Concepts</a></li>
              <li><a href="#section3" class="hover:underline">3.
Building Components</a></li>
              <li><a href="#section4" class="hover:underline">4.
Best Practices</a></li>
              <li><a href="#section5" class="hover:underline">5.
Conclusion</a></li>
            </ol>
```

```
          </div>

          <!-- Content Sections -->
          <div class="prose prose-lg max-w-none">

            <h2 id="section1" class="text-3xl font-bold text-
gray-900 mb-4 mt-12">
              Introduction to Tailwind CSS
            </h2>

            <p class="text-gray-700 leading-relaxed mb-6">
              Tailwind CSS is a utility-first CSS framework that
provides low-level utility
              classes to build custom designs without writing
custom CSS. Unlike traditional
              CSS frameworks like Bootstrap, Tailwind doesn't
come with pre-designed components.
              Instead, it gives you the building blocks to create
your own unique designs.
            </p>

            <p class="text-gray-700 leading-relaxed mb-6">
              The framework has gained massive popularity among
developers because it promotes
              a more maintainable and scalable approach to
styling. By using utility classes
              directly in your HTML, you can see exactly what
styles are being applied without
              having to jump between files.
            </p>

            <!-- Quote -->
            <blockquote class="border-l-4 border-blue-600 pl-6
py-4 my-8 bg-gray-50 italic text-gray-700">
              "Tailwind CSS is the most productive way I've found
```

```
to write CSS. It's like
              having a complete design system built into the
framework."
              <footer class="mt-2 text-sm text-gray-600 not-
italic">
                — Adam Wathan, Creator of Tailwind CSS
              </footer>
            </blockquote>

            <h2 id="section2" class="text-3xl font-bold text-
gray-900 mb-4 mt-12">
              Core Concepts
            </h2>

            <p class="text-gray-700 leading-relaxed mb-6">
              Understanding the core concepts of Tailwind is
essential for using it effectively.
              Let's explore the fundamental principles that make
this framework so powerful.
            </p>

            <h3 class="text-2xl font-semibold text-gray-900 mb-3
mt-8">
              Utility-First Approach
            </h3>

            <p class="text-gray-700 leading-relaxed mb-6">
              The utility-first approach means that instead of
writing custom CSS classes with
              semantic names, you compose designs using small,
single-purpose utility classes.
              This might seem verbose at first, but it offers
several advantages:
            </p>
```

```
            <ul class="list-disc list-inside text-gray-700 space-y-2 mb-6 ml-4">
              <li>No naming conventions to worry about</li>
              <li>Changes are localized and predictable</li>
              <li>CSS file size stays small</li>
              <li>Faster development once you learn the utilities</li>
            </ul>

            <!-- Code Example -->
            <div class="bg-gray-900 rounded-lg p-6 my-8 overflow-x-auto">
              <pre class="text-sm text-gray-100"><code>&lt;button class="bg-blue-600 hover:bg-blue-700 text-white font-semibold py-2 px-4 rounded"&gt;
  Click me
&lt;/button&gt;</code></pre>
            </div>

            <h3 class="text-2xl font-semibold text-gray-900 mb-3 mt-8">
              Responsive Design
            </h3>

            <p class="text-gray-700 leading-relaxed mb-6">
              Tailwind makes responsive design incredibly simple with its mobile-first breakpoint
              system. You can apply different styles at different screen sizes using responsive
              prefixes like <code class="bg-gray-100 px-2 py-1 rounded text-sm">md:</code>,
              <code class="bg-gray-100 px-2 py-1 rounded text-sm">lg:</code>, and
              <code class="bg-gray-100 px-2 py-1 rounded text-sm">xl:</code>.
```

```
            </p>

            <!-- Info Box -->
            <div class="bg-yellow-50 border border-yellow-200
rounded-lg p-6 my-8">
              <div class="flex items-start">
                <svg class="w-6 h-6 text-yellow-600 mt-1 mr-3
flex-shrink-0" fill="currentColor" viewBox="0 0 20 20">
                  <path fill-rule="evenodd" d="M18 10a8 8 0 11-16
0 8 8 0 0116 0zm-7-4a1 1 0 11-2 0 1 1 0 012 0zM9 9a1 1 0 000
2v3a1 1 0 001 1h1a1 1 0 100-2v-3a1 1 0 00-1-1H9z" clip-
rule="evenodd"></path>
                </svg>
                <div>
                  <h4 class="font-semibold text-yellow-900 mb-
2">Pro Tip</h4>
                  <p class="text-yellow-800 text-sm">
                    Always start with mobile styles first, then
add responsive prefixes for
                    larger screens. This mobile-first approach
ensures your site works well
                    on all devices.
                  </p>
                </div>
              </div>
            </div>

            <h2 id="section3" class="text-3xl font-bold text-
gray-900 mb-4 mt-12">
              Building Components
            </h2>

            <p class="text-gray-700 leading-relaxed mb-6">
              While Tailwind doesn't provide pre-built
components, it makes building your own
```

```
              components straightforward. You can create reusable
component patterns using
              your framework of choice (React, Vue, etc.) or by
extracting common patterns
              into custom CSS classes.
            </p>

            <!-- Image with Caption -->
            <figure class="my-8">
              <img
                src="https://images.unsplash.com/photo-
1517694712202-14dd9538aa97?w=1200&h=600&fit=crop"
                alt="Code Editor"
                class="rounded-lg shadow-lg w-full"
              >
              <figcaption class="text-center text-sm text-gray-
600 mt-3 italic">
                Example of a modern code editor setup for web
development
              </figcaption>
            </figure>

            <h2 id="section4" class="text-3xl font-bold text-
gray-900 mb-4 mt-12">
              Best Practices
            </h2>

            <p class="text-gray-700 leading-relaxed mb-6">
              To get the most out of Tailwind CSS, follow these
best practices:
            </p>

            <ol class="list-decimal list-inside text-gray-700
space-y-3 mb-6 ml-4">
              <li class="pl-2">
```

```
                <strong>Use the configuration file</strong> to
customize your design system
                with your brand colors, fonts, and spacing.
              </li>
              <li class="pl-2">
                <strong>Extract components wisely</strong> - only
create component classes
                for patterns you use frequently.
              </li>
              <li class="pl-2">
                <strong>Leverage @apply sparingly</strong> - it's
there when you need it,
                but using utilities directly is usually better.
              </li>
              <li class="pl-2">
                <strong>Use the JIT compiler</strong> for faster
builds and access to
                arbitrary values.
              </li>
              <li class="pl-2">
                <strong>Keep accessibility in mind</strong> -
Tailwind doesn't enforce
                accessibility, so remember to add ARIA attributes
and semantic HTML.
              </li>
            </ol>

            <!-- Warning Box -->
            <div class="bg-red-50 border border-red-200 rounded-
lg p-6 my-8">
              <div class="flex items-start">
                <svg class="w-6 h-6 text-red-600 mt-1 mr-3 flex-
shrink-0" fill="currentColor" viewBox="0 0 20 20">
                  <path fill-rule="evenodd" d="M8.257 3.099c.765-
1.36 2.722-1.36 3.486 0l5.58 9.92c.75 1.334-.213 2.98-1.742
```

```
2.98H4.42c-1.53 0-2.493-1.646-1.743-2.98l5.58-9.92zM11 13a1 1 0
11-2 0 1 1 0 012 0zm-1-8a1 1 0 00-1 1v3a1 1 0 002 0V6a1 1 0 00-1-
1z" clip-rule="evenodd"></path>
                </svg>
                <div>
                  <h4 class="font-semibold text-red-900 mb-
2">Common Mistake</h4>
                  <p class="text-red-800 text-sm">
                    Don't try to fight the framework by
recreating traditional CSS patterns.
                    Embrace the utility-first approach and you'll
discover its true power.
                  </p>
                </div>
              </div>
            </div>

            <h2 id="section5" class="text-3xl font-bold text-
gray-900 mb-4 mt-12">
              Conclusion
            </h2>

            <p class="text-gray-700 leading-relaxed mb-6">
              Tailwind CSS represents a paradigm shift in how we
approach styling web applications.
              While it may take some time to adjust to the
utility-first mindset, the benefits
              in terms of productivity, maintainability, and
consistency make it well worth the
              learning curve.
            </p>

            <p class="text-gray-700 leading-relaxed mb-6">
              Whether you're building a simple landing page or a
complex web application,
```

```
                Tailwind provides the tools you need to create
beautiful, responsive designs
                quickly and efficiently. Start with the basics,
experiment with different
                utilities, and soon you'll find yourself building
interfaces faster than ever before.
              </p>

            </div>

            <!-- Tags -->
            <div class="pt-8 mt-8 border-t border-gray-200">
              <h4 class="text-sm font-semibold text-gray-900 mb-
3">Tags:</h4>
              <div class="flex flex-wrap gap-2">
                <span class="px-3 py-1 bg-gray-100 text-gray-700
text-sm rounded-full hover:bg-gray-200 cursor-pointer
transition">
                  Tailwind CSS
                </span>
                <span class="px-3 py-1 bg-gray-100 text-gray-700
text-sm rounded-full hover:bg-gray-200 cursor-pointer
transition">
                  Web Development
                </span>
                <span class="px-3 py-1 bg-gray-100 text-gray-700
text-sm rounded-full hover:bg-gray-200 cursor-pointer
transition">
                  CSS Framework
                </span>
                <span class="px-3 py-1 bg-gray-100 text-gray-700
text-sm rounded-full hover:bg-gray-200 cursor-pointer
transition">
                  Frontend
                </span>
```

```
              <span class="px-3 py-1 bg-gray-100 text-gray-700
text-sm rounded-full hover:bg-gray-200 cursor-pointer
transition">
                UI/UX
              </span>
            </div>
          </div>

          <!-- Share Buttons (Mobile) -->
          <div class="lg:hidden pt-8 mt-8 border-t border-gray-
200">
            <h4 class="text-sm font-semibold text-gray-900 mb-
3">Share this article:</h4>
            <div class="flex space-x-3">
              <button class="flex-1 bg-blue-600 hover:bg-blue-700
text-white py-2 rounded-lg transition">
                Facebook
              </button>
              <button class="flex-1 bg-blue-400 hover:bg-blue-500
text-white py-2 rounded-lg transition">
                Twitter
              </button>
              <button class="flex-1 bg-blue-700 hover:bg-blue-800
text-white py-2 rounded-lg transition">
                LinkedIn
              </button>
            </div>
          </div>

        </div>

        <!-- Author Bio -->
        <div class="bg-gradient-to-r from-blue-50 to-indigo-50
rounded-xl shadow-lg p-8 mt-8">
          <div class="flex flex-col sm:flex-row items-start
```

```
sm:items-center space-y-4 sm:space-y-0 sm:space-x-6">
            <img src="https://i.pravatar.cc/100?img=12" class="w-
20 h-20 rounded-full shadow-lg" alt="Author">
            <div class="flex-1">
              <h3 class="text-xl font-bold text-gray-900 mb-
2">Sarah Johnson</h3>
              <p class="text-gray-600 mb-3">
                Senior Developer and technical writer with 8+
years of experience in web
                development. Passionate about modern CSS
frameworks and building accessible
                user interfaces.
              </p>
              <div class="flex space-x-4">
                <a href="#" class="text-blue-600 hover:text-blue-
800 transition">
                  <svg class="w-5 h-5" fill="currentColor"
viewBox="0 0 24 24">
                    <path d="M23.953 4.57a10 10 0 01-2.825.775
4.958 4.958 0 002.163-2.723c-.951.555-2.005.959-3.127 1.184a4.92
4.92 0 00-8.384 4.482C7.69 8.095 4.067 6.13 1.64 3.162a4.822
4.822 0 00-.666 2.475c0 1.71.87 3.213 2.188 4.096a4.904 4.904 0
01-2.228-.616v.06a4.923 4.923 0 003.946 4.827 4.996 4.996 0 01-
2.212.085 4.936 4.936 0 004.604 3.417 9.867 9.867 0 01-6.102
2.105c-.39 0-.779-.023-1.17-.067a13.995 13.995 0 007.557
2.209c9.053 0 13.998-7.496 13.998-13.985 0-.21
0-.42-.015-.63A9.935 9.935 0 0024 4.59z"/>
                  </svg>
                </a>
                <a href="#" class="text-blue-600 hover:text-blue-
800 transition">
                  <svg class="w-5 h-5" fill="currentColor"
viewBox="0 0 24 24">
                    <path d="M20.447 20.452h-3.554v-5.569c0-
1.328-.027-3.037-1.852-3.037-1.853 0-2.136 1.445-2.136
```

```
2.939v5.667H9.351V9h3.414v1.561h.046c.477-.9 1.637-1.85 3.37-1.85
3.601 0 4.267 2.37 4.267 5.455v6.286zM5.337 7.433c-1.144 0-
2.063-.926-2.063-2.065 0-1.138.92-2.063 2.063-2.063 1.14 0
2.064.925 2.064 2.063 0 1.139-.925 2.065-2.064 2.065zm1.782
13.019H3.555V9h3.564v11.452zM22.225 0H1.771C.792 0 0 .774 0
1.729v20.542C0 23.227.792 24 1.771 24h20.451C23.2 24 24 23.227 24
22.271V1.729C24 .774 23.2 0 22.222 0h.003z"/>
                  </svg>
                </a>
                <a href="#" class="text-blue-600 hover:text-blue-
800 transition">
                  <svg class="w-5 h-5" fill="currentColor"
viewBox="0 0 24 24">
                    <path d="M12 0C5.374 0 0 5.373 0 12c0 5.302
3.438 9.8 8.207 11.387.599.111.793-.261.793-.577v-2.234c-
3.338.726-4.033-1.416-4.033-1.416-.546-1.387-1.333-1.756-1.333-
1.756-1.089-.745.083-.729.083-.729 1.205.084 1.839 1.237 1.839
1.237 1.07 1.834 2.807 1.304 3.492.997.107-.775.418-1.305.762-
1.604-2.665-.305-5.467-1.334-5.467-5.931 0-1.311.469-2.381 1.236-
3.221-.124-.303-.535-1.524.117-3.176 0 0 1.008-.322 3.301
1.23A11.509 11.509 0 0112 5.803c1.02.005 2.047.138 3.006.404
2.291-1.552 3.297-1.23 3.297-1.23.653 1.653.242 2.874.118
3.176.77.84 1.235 1.911 1.235 3.221 0 4.609-2.807 5.624-5.479
5.921.43.372.823 1.102.823 2.222v3.293c0
.319.192.694.801.576C20.566 21.797 24 17.3 24 12c0-6.627-5.373-
12-12-12z"/>
                  </svg>
                </a>
              </div>
            </div>
            <button class="bg-blue-600 hover:bg-blue-700 text-
white px-6 py-2 rounded-lg font-medium transition">
              Follow
            </button>
          </div>
```

```
        </div>

        <!-- Comments Section -->
        <div class="bg-white rounded-xl shadow-lg p-8 mt-8">
          <h3 class="text-2xl font-bold text-gray-900 mb-
6">Comments (12)</h3>

          <!-- Comment Form -->
          <div class="mb-8 pb-8 border-b border-gray-200">
            <textarea
              rows="4"
              class="w-full border border-gray-300 rounded-lg px-
4 py-3 focus:ring-2 focus:ring-blue-500 focus:border-transparent
resize-none"
              placeholder="Join the discussion..."
            ></textarea>
            <div class="flex justify-end mt-3">
              <button class="bg-blue-600 hover:bg-blue-700 text-
white px-6 py-2 rounded-lg font-medium transition">
                Post Comment
              </button>
            </div>
          </div>

          <!-- Comments List -->
          <div class="space-y-6">

            <!-- Comment 1 -->
            <div class="flex space-x-4">
              <img src="https://i.pravatar.cc/48?img=3" class="w-
12 h-12 rounded-full flex-shrink-0" alt="User">
              <div class="flex-1">
                <div class="bg-gray-50 rounded-lg p-4">
                  <div class="flex items-center justify-between
mb-2">
```

```
                    <h4 class="font-semibold text-gray-900">Emma
Wilson</h4>
                    <span class="text-xs text-gray-500">2 hours
ago</span>
                  </div>
                  <p class="text-gray-700 text-sm">
                    Great article! I've been using Tailwind for
about a year now and it's
                    completely changed how I approach CSS. The
productivity gains are real.
                  </p>
                </div>
                <div class="flex items-center space-x-4 mt-2 ml-
4">
                  <button class="text-sm text-gray-600
hover:text-blue-600 flex items-center">
                    <svg class="w-4 h-4 mr-1" fill="none"
stroke="currentColor" viewBox="0 0 24 24">
                      <path stroke-linecap="round" stroke-
linejoin="round" stroke-width="2" d="M14 10h4.764a2 2 0 011.789
2.894l-3.5 7A2 2 0 0115.263 21h-4.017c-.163 0-.326-.02-.485-.06L7
20m7-10V5a2 2 0 00-2-2h-.095c-.5 0-.905.405-.905.905 0 .714-.211
1.412-.608 2.006L7 11v9m7-10h-2M7 20H5a2 2 0 01-2-2v-6a2 2 0 012-
2h2.5"></path>
                    </svg>
                    Like (5)
                  </button>
                  <button class="text-sm text-gray-600
hover:text-blue-600">Reply</button>
                </div>
              </div>
            </div>

            <!-- Comment 2 (with reply) -->
            <div class="flex space-x-4">
```

```
              <img src="https://i.pravatar.cc/48?img=5" class="w-
12 h-12 rounded-full flex-shrink-0" alt="User">
              <div class="flex-1">
                <div class="bg-gray-50 rounded-lg p-4">
                  <div class="flex items-center justify-between
mb-2">
                    <h4 class="font-semibold text-gray-900">Mike
Chen</h4>
                    <span class="text-xs text-gray-500">5 hours
ago</span>
                  </div>
                  <p class="text-gray-700 text-sm">
                    Would love to see a follow-up article on
advanced patterns and component
                    extraction strategies. This was super
helpful!
                  </p>
                </div>
                <div class="flex items-center space-x-4 mt-2 ml-
4">
                  <button class="text-sm text-gray-600
hover:text-blue-600 flex items-center">
                    <svg class="w-4 h-4 mr-1" fill="none"
stroke="currentColor" viewBox="0 0 24 24">
                      <path stroke-linecap="round" stroke-
linejoin="round" stroke-width="2" d="M14 10h4.764a2 2 0 011.789
2.894l-3.5 7A2 2 0 0115.263 21h-4.017c-.163 0-.326-.02-.485-.06L7
20m7-10V5a2 2 0 00-2-2h-.095c-.5 0-.905.405-.905.905 0 .714-.211
1.412-.608 2.006L7 11v9m7-10h-2M7 20H5a2 2 0 01-2-2v-6a2 2 0 012-
2h2.5"></path>
                    </svg>
                    Like (12)
                  </button>
                  <button class="text-sm text-gray-600
hover:text-blue-600">Reply</button>
```

```
                </div>

                <!-- Reply -->
                <div class="flex space-x-4 mt-4 ml-12">
                  <img src="https://i.pravatar.cc/48?img=12"
class="w-10 h-10 rounded-full flex-shrink-0" alt="Author">
                  <div class="flex-1">
                    <div class="bg-blue-50 rounded-lg p-4">
                      <div class="flex items-center justify-
between mb-2">
                        <div class="flex items-center space-x-2">
                          <h4 class="font-semibold text-gray-
900">Sarah Johnson</h4>
                          <span class="bg-blue-600 text-white
text-xs px-2 py-0.5 rounded-full">Author</span>
                        </div>
                        <span class="text-xs text-gray-500">3
hours ago</span>
                      </div>
                      <p class="text-gray-700 text-sm">
                        Thanks Mike! Great suggestion - I'm
actually working on a follow-up
                        article about advanced patterns. Stay
tuned!
                      </p>
                    </div>
                  </div>
                </div>

              </div>
            </div>

            <!-- Load More Comments -->
            <div class="text-center pt-4">
              <button class="text-blue-600 hover:text-blue-800
```

```
font-medium text-sm">
                Load more comments (9 remaining)
              </button>
            </div>

          </div>

        </div>

      </article>

      <!-- Sidebar -->
      <aside class="lg:col-span-1 space-y-8">

        <!-- Author Card (Desktop) -->
        <div class="hidden lg:block bg-white rounded-xl shadow-lg
p-6 sticky top-24">
          <div class="text-center">
            <img src="https://i.pravatar.cc/80?img=12" class="w-
20 h-20 rounded-full mx-auto mb-4 shadow-lg" alt="Author">
            <h3 class="font-bold text-gray-900 mb-1">Sarah
Johnson</h3>
            <p class="text-sm text-gray-600 mb-4">Senior
Developer</p>
            <p class="text-sm text-gray-700 mb-4">
              Writing about web development, CSS, and modern
frontend technologies.
            </p>
            <button class="w-full bg-blue-600 hover:bg-blue-700
text-white py-2 rounded-lg font-medium transition mb-3">
              Follow
            </button>
            <div class="flex justify-center space-x-3">
              <a href="#" class="text-gray-400 hover:text-blue-
600 transition">
```

```
                <svg class="w-5 h-5" fill="currentColor"
viewBox="0 0 24 24">
                  <path d="M23.953 4.57a10 10 0 01-2.825.775
4.958 4.958 0 002.163-2.723c-.951.555-2.005.959-3.127 1.184a4.92
4.92 0 00-8.384 4.482C7.69 8.095 4.067 6.13 1.64 3.162a4.822
4.822 0 00-.666 2.475c0 1.71.87 3.213 2.188 4.096a4.904 4.904 0
01-2.228-.616v.06a4.923 4.923 0 003.946 4.827 4.996 4.996 0 01-
2.212.085 4.936 4.936 0 004.604 3.417 9.867 9.867 0 01-6.102
2.105c-.39 0-.779-.023-1.17-.067a13.995 13.995 0 007.557
2.209c9.053 0 13.998-7.496 13.998-13.985 0-.21
0-.42-.015-.63A9.935 9.935 0 0024 4.59z"/>
                </svg>
              </a>
              <a href="#" class="text-gray-400 hover:text-blue-
600 transition">
                <svg class="w-5 h-5" fill="currentColor"
viewBox="0 0 24 24">
                  <path d="M12 0C5.374 0 0 5.373 0 12c0 5.302
3.438 9.8 8.207 11.387.599.111.793-.261.793-.577v-2.234c-
3.338.726-4.033-1.416-4.033-1.416-.546-1.387-1.333-1.756-1.333-
1.756-1.089-.745.083-.729.083-.729 1.205.084 1.839 1.237 1.839
1.237 1.07 1.834 2.807 1.304 3.492.997.107-.775.418-1.305.762-
1.604-2.665-.305-5.467-1.334-5.467-5.931 0-1.311.469-2.381 1.236-
3.221-.124-.303-.535-1.524.117-3.176 0 0 1.008-.322 3.301
1.23A11.509 11.509 0 0112 5.803c1.02.005 2.047.138 3.006.404
2.291-1.552 3.297-1.23 3.297-1.23.653 1.653.242 2.874.118
3.176.77.84 1.235 1.911 1.235 3.221 0 4.609-2.807 5.624-5.479
5.921.43.372.823 1.102.823 2.222v3.293c0
.319.192.694.801.576C20.566 21.797 24 17.3 24 12c0-6.627-5.373-
12-12-12z"/>
                </svg>
              </a>
            </div>
          </div>
        </div>
```

```
        <!-- Newsletter -->
        <div class="bg-gradient-to-br from-blue-600 to-indigo-700 rounded-xl shadow-lg p-6 text-white">
          <h3 class="text-xl font-bold mb-2">Subscribe to Newsletter</h3>
          <p class="text-blue-100 text-sm mb-4">
            Get the latest articles and tutorials delivered to your inbox.
          </p>
          <input
            type="email"
            placeholder="Your email"
            class="w-full px-4 py-2 rounded-lg text-gray-900 mb-3 focus:outline-none focus:ring-2 focus:ring-white"
          >
          <button class="w-full bg-white hover:bg-gray-100 text-blue-600 py-2 rounded-lg font-semibold transition">
            Subscribe
          </button>
          <p class="text-xs text-blue-100 mt-3">
            Join 10,000+ developers. Unsubscribe anytime.
          </p>
        </div>

        <!-- Popular Posts -->
        <div class="bg-white rounded-xl shadow-lg p-6">
          <h3 class="text-lg font-bold text-gray-900 mb-4">Popular Posts</h3>
          <div class="space-y-4">

            <a href="#" class="flex space-x-3 group">
              <img src="https://images.unsplash.com/photo-1498050108023-c5249f4df085?w=80&h=80&fit=crop" class="w-20 h-20 rounded-lg object-cover flex-shrink-0" alt="Post">
```

```
              <div>
                <h4 class="text-sm font-semibold text-gray-900
group-hover:text-blue-600 transition line-clamp-2">
                  10 Essential JavaScript Tips Every Developer
Should Know
                </h4>
                <p class="text-xs text-gray-500 mt-1">5 min
read</p>
              </div>
            </a>

            <a href="#" class="flex space-x-3 group">
              <img src="https://images.unsplash.com/photo-
1555066931-4365d14bab8c?w=80&h=80&fit=crop" class="w-20 h-20
rounded-lg object-cover flex-shrink-0" alt="Post">
              <div>
                <h4 class="text-sm font-semibold text-gray-900
group-hover:text-blue-600 transition line-clamp-2">
                  Building a Full-Stack App with React and
Node.js
                </h4>
                <p class="text-xs text-gray-500 mt-1">12 min
read</p>
              </div>
            </a>

            <a href="#" class="flex space-x-3 group">
              <img src="https://images.unsplash.com/photo-
1593720213428-28a5b9e94613?w=80&h=80&fit=crop" class="w-20 h-20
rounded-lg object-cover flex-shrink-0" alt="Post">
              <div>
                <h4 class="text-sm font-semibold text-gray-900
group-hover:text-blue-600 transition line-clamp-2">
                  The Future of Web Development in 2026
                </h4>
```

```
                <p class="text-xs text-gray-500 mt-1">8 min
read</p>
              </div>
            </a>

          </div>
        </div>

        <!-- Categories -->
        <div class="bg-white rounded-xl shadow-lg p-6">
          <h3 class="text-lg font-bold text-gray-900 mb-
4">Categories</h3>
          <div class="space-y-2">
            <a href="#" class="flex items-center justify-between
py-2 hover:text-blue-600 transition">
              <span class="text-sm text-gray-700">Web
Development</span>
              <span class="bg-gray-100 text-gray-600 text-xs px-2
py-1 rounded-full">42</span>
            </a>
            <a href="#" class="flex items-center justify-between
py-2 hover:text-blue-600 transition">
              <span class="text-sm text-gray-
700">JavaScript</span>
              <span class="bg-gray-100 text-gray-600 text-xs px-2
py-1 rounded-full">38</span>
            </a>
            <a href="#" class="flex items-center justify-between
py-2 hover:text-blue-600 transition">
              <span class="text-sm text-gray-700">CSS</span>
              <span class="bg-gray-100 text-gray-600 text-xs px-2
py-1 rounded-full">25</span>
            </a>
            <a href="#" class="flex items-center justify-between
py-2 hover:text-blue-600 transition">
```

```
              <span class="text-sm text-gray-700">React</span>
              <span class="bg-gray-100 text-gray-600 text-xs px-2
py-1 rounded-full">31</span>
            </a>
            <a href="#" class="flex items-center justify-between
py-2 hover:text-blue-600 transition">
              <span class="text-sm text-gray-
700">Tutorials</span>
              <span class="bg-gray-100 text-gray-600 text-xs px-2
py-1 rounded-full">19</span>
            </a>
          </div>
        </div>

      </aside>

    </div>

    <!-- Related Posts -->
    <div class="mt-16">
      <h2 class="text-2xl font-bold text-gray-900 mb-8">Related
Articles</h2>
      <div class="grid grid-cols-1 md:grid-cols-3 gap-8">

        <!-- Related Post 1 -->
        <article class="bg-white rounded-xl shadow-lg overflow-
hidden group hover:shadow-xl transition-shadow">
          <div class="relative overflow-hidden">
            <img src="https://images.unsplash.com/photo-
1587620962725-abab7fe55159?w=400&h=250&fit=crop" class="w-full h-
48 object-cover transform group-hover:scale-110 transition-
transform duration-500" alt="Post">
            <span class="absolute top-4 left-4 bg-purple-600
text-white text-xs font-bold px-3 py-1 rounded-full">
              CSS
```

```
            </span>
          </div>
          <div class="p-6">
            <h3 class="text-xl font-bold text-gray-900 mb-2
group-hover:text-blue-600 transition">
              Advanced CSS Grid Techniques
            </h3>
            <p class="text-gray-600 text-sm mb-4">
              Learn how to create complex layouts using CSS
Grid's powerful features.
            </p>
            <div class="flex items-center text-sm text-gray-500">
              <span>Jan 5, 2026</span>
              <span class="mx-2">•</span>
              <span>8 min read</span>
            </div>
          </div>
        </article>

        <!-- Related Post 2 -->
        <article class="bg-white rounded-xl shadow-lg overflow-
hidden group hover:shadow-xl transition-shadow">
          <div class="relative overflow-hidden">
            <img src="https://images.unsplash.com/photo-
1555099962-4199c345e5dd?w=400&h=250&fit=crop" class="w-full h-48
object-cover transform group-hover:scale-110 transition-transform
duration-500" alt="Post">
            <span class="absolute top-4 left-4 bg-green-600 text-
white text-xs font-bold px-3 py-1 rounded-full">
              JavaScript
            </span>
          </div>
          <div class="p-6">
            <h3 class="text-xl font-bold text-gray-900 mb-2
group-hover:text-blue-600 transition">
```

```
              Modern JavaScript Patterns
            </h3>
            <p class="text-gray-600 text-sm mb-4">
              Explore the latest patterns and best practices in
modern JavaScript development.
            </p>
            <div class="flex items-center text-sm text-gray-500">
              <span>Jan 4, 2026</span>
              <span class="mx-2">•</span>
              <span>10 min read</span>
            </div>
          </div>
        </article>

        <!-- Related Post 3 -->
        <article class="bg-white rounded-xl shadow-lg overflow-
hidden group hover:shadow-xl transition-shadow">
          <div class="relative overflow-hidden">
            <img src="https://images.unsplash.com/photo-
1633356122544-f134324a6cee?w=400&h=250&fit=crop" class="w-full h-
48 object-cover transform group-hover:scale-110 transition-
transform duration-500" alt="Post">
            <span class="absolute top-4 left-4 bg-blue-600 text-
white text-xs font-bold px-3 py-1 rounded-full">
              React
            </span>
          </div>
          <div class="p-6">
            <h3 class="text-xl font-bold text-gray-900 mb-2
group-hover:text-blue-600 transition">
              Building with React Hooks
            </h3>
            <p class="text-gray-600 text-sm mb-4">
              Master React Hooks to write cleaner and more
efficient components.
```

```
            </p>
            <div class="flex items-center text-sm text-gray-500">
              <span>Jan 3, 2026</span>
              <span class="mx-2">•</span>
              <span>15 min read</span>
            </div>
          </div>
        </article>

      </div>
    </div>

  </div>

  <!-- Footer -->
  <footer class="bg-gray-900 text-gray-300 mt-20">
    <div class="max-w-7xl mx-auto px-4 sm:px-6 lg:px-8 py-12">
      <div class="grid grid-cols-2 md:grid-cols-4 gap-8">
        <div>
          <h3 class="text-white font-semibold mb-4">About</h3>
          <ul class="space-y-2 text-sm">
            <li><a href="#" class="hover:text-white
transition">Our Story</a></li>
            <li><a href="#" class="hover:text-white
transition">Team</a></li>
            <li><a href="#" class="hover:text-white
transition">Careers</a></li>
          </ul>
        </div>
        <div>
          <h3 class="text-white font-semibold mb-4">Content</h3>
          <ul class="space-y-2 text-sm">
            <li><a href="#" class="hover:text-white
transition">Articles</a></li>
            <li><a href="#" class="hover:text-white
```

```
transition">Tutorials</a></li>
            <li><a href="#" class="hover:text-white
transition">Videos</a></li>
          </ul>
        </div>
        <div>
          <h3 class="text-white font-semibold mb-
4">Community</h3>
          <ul class="space-y-2 text-sm">
            <li><a href="#" class="hover:text-white
transition">Forum</a></li>
            <li><a href="#" class="hover:text-white
transition">Discord</a></li>
            <li><a href="#" class="hover:text-white
transition">Newsletter</a></li>
          </ul>
        </div>
        <div>
          <h3 class="text-white font-semibold mb-4">Legal</h3>
          <ul class="space-y-2 text-sm">
            <li><a href="#" class="hover:text-white
transition">Privacy</a></li>
            <li><a href="#" class="hover:text-white
transition">Terms</a></li>
            <li><a href="#" class="hover:text-white
transition">Contact</a></li>
          </ul>
        </div>
      </div>
      <div class="border-t border-gray-800 mt-8 pt-8 text-center
text-sm">
        <p>© 2026 DevBlog. All rights reserved.</p>
      </div>
    </div>
  </footer>
```

```
</body>
</html>
```

This blog layout demonstrates:

✅ **Complete article layout** with hero, content, and metadata
✅ **Sticky navigation**
✅ **Breadcrumb navigation**
✅ **Table of contents**
✅ **Rich typography** (headings, quotes, code blocks, lists)
✅ **Info/warning boxes**
✅ **Image with captions**
✅ **Social sharing** (sticky on desktop, inline on mobile)
✅ **Author bio** with social links
✅ **Comments section** with nested replies
✅ **Sidebar widgets** (author card, newsletter, popular posts, categories)
✅ **Related articles** grid
✅ **Tag system**
✅ **Responsive design** throughout

Congratulations! 🎉 You've now completed all 12 modules of the comprehensive Tailwind CSS course. You've learned:

1. ✅ Setup and configuration
2. ✅ Core utilities and concepts
3. ✅ Colors and backgrounds
4. ✅ Typography
5. ✅ Layout (Flexbox & Grid)
6. ✅ Responsive design
7. ✅ State variants and interactions

8. ✅ Borders, shadows, and effects
9. ✅ Customization
10. ✅ Components and patterns
11. ✅ Best practices and optimization
12. ✅ Real-world projects

You now have the knowledge and practical examples to build production-ready websites and applications with Tailwind CSS. Keep practicing, experiment with different patterns, and most importantly, build real projects to solidify your understanding!

1/7/2026, 1:24:42 PM

You're absolutely right! I apologize for missing these important properties. Let me create a comprehensive module covering these essential CSS properties that work with Tailwind.

BONUS MODULE: Essential Properties & Utilities

Object Fit & Object Position

These properties control how images and videos are sized and positioned within their containers.

Object Fit

Controls how content is resized to fit its container.

```
<!-- object-contain: Scales content to fit inside container
(maintains aspect ratio, may show empty space) -->
```

```
<div class="w-64 h-64 bg-gray-200">
  <img src="landscape.jpg" class="w-full h-full object-contain"
alt="Landscape">
  <!-- Image will be fully visible, but might not fill the
container -->
</div>

<!-- object-cover: Scales content to fill container (maintains
aspect ratio, may crop) -->
<div class="w-64 h-64 bg-gray-200">
  <img src="portrait.jpg" class="w-full h-full object-cover"
alt="Portrait">
  <!-- Image fills entire container, excess is cropped -->
</div>

<!-- object-fill: Stretches content to fill container (doesn't
maintain aspect ratio) -->
<div class="w-64 h-64 bg-gray-200">
  <img src="photo.jpg" class="w-full h-full object-fill"
alt="Photo">
  <!-- Image is stretched/squished to fill container -->
</div>

<!-- object-none: Content not resized (original size) -->
<div class="w-64 h-64 bg-gray-200 overflow-hidden">
  <img src="icon.jpg" class="w-full h-full object-none"
alt="Icon">
  <!-- Image keeps original size, may overflow or be too small --
>
</div>

<!-- object-scale-down: Same as contain or none, whichever is
smaller -->
<div class="w-64 h-64 bg-gray-200">
  <img src="small-icon.jpg" class="w-full h-full object-scale-
```

```
down" alt="Small Icon">
  <!-- If image is smaller than container, uses original size.
Otherwise, contains it. -->
</div>
```

Common Use Cases

```
<!-- Profile pictures (circular avatars) -->
<div class="w-20 h-20 rounded-full overflow-hidden">
  <img src="avatar.jpg" class="w-full h-full object-cover"
alt="Avatar">
  <!-- object-cover ensures face fills the circle without
distortion -->
</div>

<!-- Product cards with consistent image sizes -->
<div class="w-full h-64 bg-gray-100 rounded-lg overflow-hidden">
  <img src="product.jpg" class="w-full h-full object-cover"
alt="Product">
  <!-- All product images have same dimensions, cropped nicely --
>
</div>

<!-- Logo display (maintain aspect ratio, show full logo) -->
<div class="w-48 h-24 bg-white">
  <img src="logo.svg" class="w-full h-full object-contain"
alt="Logo">
  <!-- Logo is fully visible and not distorted -->
</div>

<!-- Hero background image -->
<div class="relative w-full h-screen">
  <img src="hero.jpg" class="absolute inset-0 w-full h-full
object-cover" alt="Hero">
```

```
  <!-- Image covers entire viewport without distortion -->
</div>
```

Object Position

Controls where the content is positioned within its container.

```
<!-- Position presets -->
<img src="photo.jpg" class="w-64 h-64 object-cover object-center"
alt="Center">
<img src="photo.jpg" class="w-64 h-64 object-cover object-top"
alt="Top">
<img src="photo.jpg" class="w-64 h-64 object-cover object-bottom"
alt="Bottom">
<img src="photo.jpg" class="w-64 h-64 object-cover object-left"
alt="Left">
<img src="photo.jpg" class="w-64 h-64 object-cover object-right"
alt="Right">

<!-- Corner positions -->
<img src="photo.jpg" class="w-64 h-64 object-cover object-left-
top" alt="Top Left">
<img src="photo.jpg" class="w-64 h-64 object-cover object-right-
top" alt="Top Right">
<img src="photo.jpg" class="w-64 h-64 object-cover object-left-
bottom" alt="Bottom Left">
<img src="photo.jpg" class="w-64 h-64 object-cover object-right-
bottom" alt="Bottom Right">
```

Real-world example:

```
<!-- Portrait photos - focus on faces (usually in upper portion)
-->
<div class="w-48 h-64 overflow-hidden rounded-lg">
```

```
  <img src="portrait.jpg" class="w-full h-full object-cover
object-top" alt="Portrait">
  <!-- Keeps face visible even when cropping -->
</div>

<!-- Product showcase - show product details at bottom -->
<div class="w-64 h-80 overflow-hidden rounded-lg">
  <img src="shoes.jpg" class="w-full h-full object-cover object-
bottom" alt="Shoes">
  <!-- Shows the product at bottom of frame -->
</div>
```

Overflow Properties

Controls what happens when content is too large for its container.

Overflow Basics

```
<!-- overflow-auto: Adds scrollbar only when needed -->
<div class="w-64 h-32 overflow-auto bg-gray-100 p-4">
  <p>Long content here that will make the container scrollable
when it exceeds the height...</p>
</div>

<!-- overflow-hidden: Hides overflow content (no scrollbar) -->
<div class="w-64 h-32 overflow-hidden bg-gray-100 p-4">
  <p>Content beyond the container height is hidden and not
accessible...</p>
</div>

<!-- overflow-visible: Content overflows and is visible (default)
-->
```

```
<div class="w-64 h-32 overflow-visible bg-gray-100 p-4">
  <p>Content extends beyond container boundaries...</p>
</div>

<!-- overflow-scroll: Always shows scrollbars (even if not
needed) -->
<div class="w-64 h-32 overflow-scroll bg-gray-100 p-4">
  <p>Short content but scrollbars still appear...</p>
</div>
```

Directional Overflow

```
<!-- Horizontal overflow only -->
<div class="w-64 h-32 overflow-x-auto overflow-y-hidden bg-gray-
100 p-4">
  <div class="w-[500px]">Wide content scrolls horizontally...
</div>
</div>

<!-- Vertical overflow only -->
<div class="w-64 h-32 overflow-y-auto overflow-x-hidden bg-gray-
100 p-4">
  <p>Tall content...</p>
  <p>More content...</p>
  <p>Even more content...</p>
</div>

<!-- Hide horizontal, scroll vertical -->
<div class="overflow-x-hidden overflow-y-auto w-64 h-48">
  <!-- Common for chat interfaces or feeds -->
</div>
```

Common Patterns

```
<!-- Horizontal scrolling cards (mobile product carousel) -->
<div class="overflow-x-auto overflow-y-hidden whitespace-nowrap
pb-4">
  <div class="inline-block w-64 h-80 bg-white rounded-lg shadow-
lg mx-2">Card 1</div>
  <div class="inline-block w-64 h-80 bg-white rounded-lg shadow-
lg mx-2">Card 2</div>
  <div class="inline-block w-64 h-80 bg-white rounded-lg shadow-
lg mx-2">Card 3</div>
  <div class="inline-block w-64 h-80 bg-white rounded-lg shadow-
lg mx-2">Card 4</div>
</div>

<!-- Chat/Feed container with auto-scroll -->
<div class="h-96 overflow-y-auto overflow-x-hidden bg-white
border border-gray-200 rounded-lg p-4">
  <div class="space-y-4">
    <div class="bg-blue-100 rounded-lg p-3">Message 1</div>
    <div class="bg-gray-100 rounded-lg p-3">Message 2</div>
    <div class="bg-blue-100 rounded-lg p-3">Message 3</div>
    <!-- More messages -->
  </div>
</div>

<!-- Code block with horizontal scroll -->
<div class="overflow-x-auto bg-gray-900 text-gray-100 rounded-lg
p-4">
  <pre><code>const veryLongFunctionName = (parameter1,
parameter2, parameter3, parameter4) => {
  return parameter1 + parameter2 + parameter3 + parameter4;
}</code></pre>
</div>

<!-- Table with horizontal scroll on mobile -->
<div class="overflow-x-auto">
```

```
  <table class="min-w-full">
    <thead>
      <tr>
        <th class="px-6 py-3">Column 1</th>
        <th class="px-6 py-3">Column 2</th>
        <th class="px-6 py-3">Column 3</th>
        <th class="px-6 py-3">Column 4</th>
      </tr>
    </thead>
    <tbody>
      <!-- Table rows -->
    </tbody>
  </table>
</div>

<!-- Image gallery with hidden overflow for hover effects -->
<div class="grid grid-cols-3 gap-4">
  <div class="aspect-square overflow-hidden rounded-lg">
    <img src="photo.jpg" class="w-full h-full object-cover
transform hover:scale-110 transition-transform duration-300">
  </div>
</div>

<!-- Modal with scrollable content -->
<div class="fixed inset-0 bg-black bg-opacity-50 flex items-
center justify-center p-4">
  <div class="bg-white rounded-xl max-w-2xl w-full max-h-[90vh]
overflow-hidden flex flex-col">
    <div class="p-6 border-b">
      <h2 class="text-2xl font-bold">Modal Title</h2>
    </div>
    <div class="flex-1 overflow-y-auto p-6">
      <!-- Long scrollable content -->
    </div>
    <div class="p-6 border-t">
```

```
      <button class="bg-blue-600 text-white px-4 py-2
rounded">Confirm</button>
    </div>
  </div>
</div>
```

Overflow with Text

```
<!-- Text ellipsis (single line) -->
<div class="w-48 truncate">
  This is a very long text that will be cut off with an
ellipsis...
</div>

<!-- Multi-line text ellipsis (requires line-clamp) -->
<div class="w-64 line-clamp-2">
  This is a longer text that will be displayed on multiple lines
but will be cut off after two lines with an ellipsis at the
end...
</div>

<!-- Line clamp variations -->
<p class="line-clamp-1">Single line...</p>
<p class="line-clamp-2">Two lines...</p>
<p class="line-clamp-3">Three lines...</p>
<p class="line-clamp-4">Four lines...</p>
<p class="line-clamp-5">Five lines...</p>
<p class="line-clamp-6">Six lines...</p>
<p class="line-clamp-none">No clamp (shows all text)</p>

<!-- Product card with text truncation -->
<div class="bg-white p-4 rounded-lg shadow">
  <img src="product.jpg" class="w-full h-48 object-cover rounded
mb-4">
```

```
  <h3 class="font-bold truncate mb-2">Very Long Product Name That
Gets Cut Off</h3>
  <p class="text-sm text-gray-600 line-clamp-3">
    This is a product description that can span multiple lines
but will be limited
    to exactly three lines before being cut off with an ellipsis
at the end.
  </p>
</div>
```

Screen Resolution & Aspect Ratio

Aspect Ratio

Controls the aspect ratio of elements.

```
<!-- Common aspect ratios -->
<div class="aspect-square bg-blue-500">
  <!-- 1:1 - Perfect square -->
</div>

<div class="aspect-video bg-blue-500">
  <!-- 16:9 - Widescreen video -->
</div>

<!-- Custom aspect ratios -->
<div class="aspect-[4/3] bg-blue-500">
  <!-- 4:3 - Traditional TV/monitor -->
</div>

<div class="aspect-[21/9] bg-blue-500">
  <!-- 21:9 - Ultrawide -->
```

```
</div>

<div class="aspect-[9/16] bg-blue-500">
  <!-- 9:16 - Vertical video (mobile/stories) -->
</div>

<div class="aspect-[3/2] bg-blue-500">
  <!-- 3:2 - Classic photo format -->
</div>

<!-- Reset aspect ratio -->
<div class="aspect-auto">
  <!-- No aspect ratio constraint -->
</div>
```

Real-World Examples

```
<!-- Video embed with proper aspect ratio -->
<div class="aspect-video bg-black">
  <iframe
    src="https://www.youtube.com/embed/..."
    class="w-full h-full"
    allow="accelerometer; autoplay; encrypted-media"
  ></iframe>
</div>

<!-- Instagram-style square images -->
<div class="grid grid-cols-3 gap-2">
  <div class="aspect-square overflow-hidden rounded">
    <img src="photo1.jpg" class="w-full h-full object-cover">
  </div>
  <div class="aspect-square overflow-hidden rounded">
    <img src="photo2.jpg" class="w-full h-full object-cover">
  </div>
```

```
  <div class="aspect-square overflow-hidden rounded">
    <img src="photo3.jpg" class="w-full h-full object-cover">
  </div>
</div>

<!-- Product cards with consistent aspect ratio -->
<div class="bg-white rounded-lg shadow-lg overflow-hidden">
  <div class="aspect-[4/3] bg-gray-100">
    <img src="product.jpg" class="w-full h-full object-cover">
  </div>
  <div class="p-4">
    <h3 class="font-bold">Product Name</h3>
    <p class="text-gray-600">$99.99</p>
  </div>
</div>

<!-- Story/Reel style vertical videos -->
<div class="aspect-[9/16] max-w-sm mx-auto bg-black rounded-lg
overflow-hidden">
  <video class="w-full h-full object-cover" controls>
    <source src="story.mp4" type="video/mp4">
  </video>
</div>

<!-- Hero section with cinematic aspect ratio -->
<div class="aspect-[21/9] relative overflow-hidden">
  <img src="hero.jpg" class="absolute inset-0 w-full h-full
object-cover">
  <div class="absolute inset-0 bg-black bg-opacity-40 flex items-
center justify-center">
    <h1 class="text-white text-5xl font-bold">Hero Title</h1>
  </div>
</div>
```

Responsive Aspect Ratios

```
<!-- Different aspect ratios at different breakpoints -->
<div class="aspect-square md:aspect-video lg:aspect-[21/9]">
  <!-- Square on mobile, 16:9 on tablet, 21:9 on desktop -->
</div>

<!-- Responsive image gallery -->
<div class="grid grid-cols-2 md:grid-cols-3 lg:grid-cols-4 gap-
4">
  <div class="aspect-square overflow-hidden rounded-lg">
    <img src="photo1.jpg" class="w-full h-full object-cover
hover:scale-110 transition-transform duration-300">
  </div>
  <!-- More images -->
</div>
```

Additional Important Properties

Whitespace

Controls text wrapping and whitespace handling.

```
<!-- whitespace-normal: Default text wrapping -->
<p class="whitespace-normal">
  This text wraps normally when it reaches the end of the
container.
</p>

<!-- whitespace-nowrap: Prevents text wrapping -->
<p class="whitespace-nowrap overflow-hidden">
  This text will not wrap and will overflow or be hidden.
```

```
</p>

<!-- whitespace-pre: Preserves whitespace and line breaks -->
<p class="whitespace-pre">
Line 1
  Line 2 with indent
    Line 3 with more indent
</p>

<!-- whitespace-pre-line: Preserves line breaks but collapses
spaces -->
<p class="whitespace-pre-line">
Line 1
Line 2    with    extra    spaces
Line 3
</p>

<!-- whitespace-pre-wrap: Preserves whitespace but wraps text -->
<p class="whitespace-pre-wrap">
  This preserves    spaces    and wraps text at container edge.
</p>

<!-- whitespace-break-spaces: Like pre-wrap but handles line
breaks differently -->
<p class="whitespace-break-spaces">
  Preserves    spaces    and breaks lines at spaces if needed.
</p>
```

Word Break

Controls how words break at line ends.

```
<!-- break-normal: Default word breaking -->
<p class="break-normal w-48">
  ThisIsAVeryLongWordThatWillNotBreak until it absolutely has to.
```

```
</p>

<!-- break-words: Breaks words if needed to prevent overflow -->
<p class="break-words w-48">
  ThisIsAVeryLongWordThatWillBreakIfNeededToFitTheContainer
</p>

<!-- break-all: Breaks at any character (not just word
boundaries) -->
<p class="break-all w-48">
  ThisTextWillBreakAtAnyCharacterNotJustWordBoundaries
</p>

<!-- break-keep: Prevents breaks (mainly for CJK text) -->
<p class="break-keep">
  这段中文文本不会被分割
</p>
```

Common use cases:

```
<!-- URL display -->
<div class="w-64 p-4 bg-gray-100 rounded">
  <p class="text-sm break-all text-blue-600">
    https://www.example.com/very-long-url-path/that-needs-to-
break/properly
  </p>
</div>

<!-- Email addresses -->
<div class="w-48 p-4 bg-gray-100 rounded">
  <p class="text-sm break-all">
    verylongemailaddress@extremelylongdomainname.com
  </p>
</div>
```

```
<!-- Code snippets -->
<div class="w-64 p-4 bg-gray-900 text-gray-100 rounded overflow-
x-auto">
  <code class="whitespace-pre">
const variable = "value";
function example() {
  return true;
}
  </code>
</div>
```

Hyphens

Controls hyphenation of words.

```
<!-- hyphens-none: No hyphenation -->
<p class="hyphens-none w-48">
  Supercalifragilisticexpialidocious is a very long word.
</p>

<!-- hyphens-manual: Only hyphenate at &shy; -->
<p class="hyphens-manual w-48">

Super&shy;cali&shy;fragi&shy;listic&shy;expi&shy;ali&shy;docious
</p>

<!-- hyphens-auto: Automatic hyphenation -->
<p class="hyphens-auto w-48" lang="en">
  Supercalifragilisticexpialidocious will hyphenate
automatically.
</p>
```

Pointer Events

Controls whether element responds to pointer events.

```
<!-- pointer-events-none: Element doesn't respond to pointer events -->
<button class="pointer-events-none bg-gray-300 px-4 py-2 rounded">
  Can't Click Me
</button>

<!-- pointer-events-auto: Normal pointer events (default) -->
<button class="pointer-events-auto bg-blue-600 text-white px-4 py-2 rounded">
  Can Click Me
</button>

<!-- Useful for overlays that shouldn't block clicks -->
<div class="relative">
  <img src="photo.jpg" class="w-full h-64 object-cover">
  <!-- Overlay that doesn't block clicks to image -->
  <div class="absolute inset-0 bg-gradient-to-t from-black to-transparent pointer-events-none"></div>
  <!-- But buttons on overlay should work -->
  <button class="absolute bottom-4 left-4 bg-white px-4 py-2 rounded pointer-events-auto">
    Click Me
  </button>
</div>
```

User Select

Controls whether text can be selected.

```
<!-- select-none: Text cannot be selected -->
<p class="select-none">
```

```
  You cannot select this text with your cursor.
</p>

<!-- select-text: Text can be selected (default) -->
<p class="select-text">
  You can select this text normally.
</p>

<!-- select-all: Clicking selects all text -->
<p class="select-all">
  Click anywhere to select all this text.
</p>

<!-- select-auto: Default browser behavior -->
<p class="select-auto">
  Normal selection behavior.
</p>

<!-- Common use case: preventing selection of UI elements -->
<div class="flex items-center space-x-2 select-none">
  <button class="bg-blue-600 text-white px-4 py-2 rounded">Button
1</button>
  <button class="bg-blue-600 text-white px-4 py-2 rounded">Button
2</button>
  <!-- User can't accidentally select button text while clicking
-->
</div>

<!-- Allow selection of code but not surrounding UI -->
<div class="bg-gray-900 rounded-lg p-4 select-none">
  <div class="flex items-center justify-between mb-2">
    <span class="text-gray-400 text-sm">index.js</span>
    <button class="text-gray-400 hover:text-white text-
sm">Copy</button>
  </div>
```

```
  <pre class="select-text text-gray-100"><code>const example =
"code";</code></pre>
</div>
```

Resize

Controls whether an element can be resized by the user.

```
<!-- resize-none: Cannot be resized (default for most elements) -
->
<textarea class="resize-none border border-gray-300 p-2 rounded
w-full h-24"></textarea>

<!-- resize: Can be resized both directions -->
<textarea class="resize border border-gray-300 p-2 rounded w-full
h-24"></textarea>

<!-- resize-x: Can be resized horizontally -->
<textarea class="resize-x border border-gray-300 p-2 rounded w-
full h-24"></textarea>

<!-- resize-y: Can be resized vertically -->
<textarea class="resize-y border border-gray-300 p-2 rounded w-
full h-24"></textarea>
```

Scroll Behavior

Controls smooth scrolling.

```
<!-- scroll-smooth: Smooth scrolling for anchor links -->
<html class="scroll-smooth">
  <body>
    <a href="#section1">Go to Section 1</a>
```

```
    <div id="section1" class="h-screen">
      Section 1 content
    </div>
  </body>
</html>

<!-- scroll-auto: Instant scrolling (default) -->
<html class="scroll-auto">
  <!-- Jumps instantly to anchor -->
</html>

<!-- Smooth scroll container -->
<div class="h-96 overflow-y-auto scroll-smooth">
  <div class="h-screen">Section 1</div>
  <div class="h-screen">Section 2</div>
  <div class="h-screen">Section 3</div>
</div>
```

Scroll Snap

Creates snap points while scrolling.

```
<!-- Horizontal scroll snap carousel -->
<div class="flex overflow-x-auto snap-x snap-mandatory scrollbar-
hide">
  <div class="flex-shrink-0 w-full h-64 snap-center bg-blue-500
flex items-center justify-center text-white text-4xl">
    Slide 1
  </div>
  <div class="flex-shrink-0 w-full h-64 snap-center bg-green-500
flex items-center justify-center text-white text-4xl">
    Slide 2
  </div>
  <div class="flex-shrink-0 w-full h-64 snap-center bg-red-500
flex items-center justify-center text-white text-4xl">
```

```
    Slide 3
  </div>
</div>

<!-- Vertical scroll snap -->
<div class="h-screen overflow-y-auto snap-y snap-mandatory">
  <div class="h-screen snap-start bg-blue-500"></div>
  <div class="h-screen snap-start bg-green-500"></div>
  <div class="h-screen snap-start bg-red-500"></div>
</div>

<!-- Image gallery with snap -->
<div class="flex overflow-x-auto gap-4 p-4 snap-x snap-
mandatory">
  <img src="photo1.jpg" class="w-80 h-96 object-cover rounded-lg
flex-shrink-0 snap-center">
  <img src="photo2.jpg" class="w-80 h-96 object-cover rounded-lg
flex-shrink-0 snap-center">
  <img src="photo3.jpg" class="w-80 h-96 object-cover rounded-lg
flex-shrink-0 snap-center">
</div>

<!-- Snap alignment options -->
<div class="overflow-x-auto snap-x">
  <div class="snap-start">Snaps to start</div>
  <div class="snap-center">Snaps to center</div>
  <div class="snap-end">Snaps to end</div>
</div>

<!-- Snap strictness -->
<div class="snap-mandatory">
</div>
<div class="snap-proximity">
</div>
<div class="snap-none">
```

Mix Blend Mode

Controls how element's content blends with background.

```
<!-- Blend modes -->
<div class="relative">
  <img src="background.jpg" class="w-full h-64 object-cover">
  <div class="absolute inset-0 bg-blue-500 mix-blend-multiply">
</div>
</div>

<!-- Available modes -->
<div class="mix-blend-normal">Normal</div>
<div class="mix-blend-multiply">Multiply</div>
<div class="mix-blend-screen">Screen</div>
<div class="mix-blend-overlay">Overlay</div>
<div class="mix-blend-darken">Darken</div>
<div class="mix-blend-lighten">Lighten</div>
<div class="mix-blend-color-dodge">Color Dodge</div>
<div class="mix-blend-color-burn">Color Burn</div>
<div class="mix-blend-hard-light">Hard Light</div>
<div class="mix-blend-soft-light">Soft Light</div>
<div class="mix-blend-difference">Difference</div>
<div class="mix-blend-exclusion">Exclusion</div>
<div class="mix-blend-hue">Hue</div>
<div class="mix-blend-saturation">Saturation</div>
<div class="mix-blend-color">Color</div>
<div class="mix-blend-luminosity">Luminosity</div>

<!-- Text over image with blend mode -->
<div class="relative h-96">
  <img src="photo.jpg" class="absolute inset-0 w-full h-full
object-cover">
  <h1 class="absolute inset-0 flex items-center justify-center
text-9xl font-bold text-white mix-blend-difference">
```

```
    BLEND
  </h1>
</div>
```

Background Blend Mode

Controls how background layers blend together.

```
<div class="h-64 bg-blue-500 bg-gradient-to-r from-purple-500 to-pink-500 bg-blend-multiply">
  <!-- Blends gradient with solid color -->
</div>

<!-- Image with color overlay blend -->
<div class="h-64 bg-blue-500 bg-[url('photo.jpg')] bg-cover bg-blend-overlay">
  <!-- Blends image with blue background -->
</div>
```

Complete Real-World Example

Here's a comprehensive example using many of these properties:

```
<!DOCTYPE html>
<html lang="en" class="scroll-smooth">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Property Showcase</title>
  <script src="https://cdn.tailwindcss.com"></script>
```

```
</head>
<body class="bg-gray-50">

  <!-- Hero with aspect ratio and object-fit -->
  <section class="relative aspect-[21/9] overflow-hidden">
    <img
      src="https://images.unsplash.com/photo-1618005182384-
a83a8bd57fbe?w=1920&h=1080&fit=crop"
      class="absolute inset-0 w-full h-full object-cover"
      alt="Hero"
    >
    <div class="absolute inset-0 bg-gradient-to-t from-black/80
to-transparent"></div>
    <div class="absolute inset-0 flex items-center justify-
center">
      <h1 class="text-6xl font-bold text-white select-none">
        Welcome
      </h1>
    </div>
  </section>

  <!-- Horizontal scroll gallery with snap -->
  <section class="py-12">
    <div class="max-w-7xl mx-auto px-4">
      <h2 class="text-3xl font-bold mb-6">Gallery</h2>
      <div class="flex overflow-x-auto gap-4 pb-4 snap-x snap-
mandatory scrollbar-hide">
        <div class="flex-shrink-0 w-80 aspect-[4/3] snap-center">
          <img
            src="https://images.unsplash.com/photo-1682687220742-
aba13b6e50ba?w=400&h=300&fit=crop"
            class="w-full h-full object-cover rounded-lg shadow-
lg"
            alt="Gallery 1"
          >
```

```
        </div>
        <div class="flex-shrink-0 w-80 aspect-[4/3] snap-center">
          <img
            src="https://images.unsplash.com/photo-1682687221038-
404cb8830901?w=400&h=300&fit=crop"
            class="w-full h-full object-cover rounded-lg shadow-
lg"
            alt="Gallery 2"
          >
        </div>
        <div class="flex-shrink-0 w-80 aspect-[4/3] snap-center">
          <img
            src="https://images.unsplash.com/photo-1682687220063-
4742bd7fd538?w=400&h=300&fit=crop"
            class="w-full h-full object-cover rounded-lg shadow-
lg"
            alt="Gallery 3"
          >
        </div>
        <div class="flex-shrink-0 w-80 aspect-[4/3] snap-center">
          <img
            src="https://images.unsplash.com/photo-1682687220208-
22d7a2543e88?w=400&h=300&fit=crop"
            class="w-full h-full object-cover rounded-lg shadow-
lg"
            alt="Gallery 4"
          >
        </div>
      </div>
    </div>
  </section>

  <!-- Product grid with various properties -->
  <section class="py-12">
    <div class="max-w-7xl mx-auto px-4">
```

```
      <h2 class="text-3xl font-bold mb-6">Products</h2>
      <div class="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3
gap-6">

        <!-- Product Card -->
        <div class="bg-white rounded-lg shadow-lg overflow-hidden
group">
          <div class="aspect-square overflow-hidden bg-gray-100">
            <img
              src="https://images.unsplash.com/photo-
1505740420928-5e560c06d30e?w=400&h=400&fit=crop"
              class="w-full h-full object-cover transform group-
hover:scale-110 transition-transform duration-500"
              alt="Product"
            >
          </div>
          <div class="p-4">
            <h3 class="font-bold text-lg truncate mb-2">
              Premium Wireless Headphones
            </h3>
            <p class="text-gray-600 text-sm line-clamp-2 mb-4">
              Experience studio-quality sound with active noise
cancellation and 40-hour battery life
            </p>
            <div class="flex items-center justify-between">
              <span class="text-2xl font-bold">$299</span>
              <button class="bg-blue-600 hover:bg-blue-700 text-
white px-4 py-2 rounded-lg transition select-none">
                Add to Cart
              </button>
            </div>
          </div>
        </div>

        <!-- More products... -->
```

```
      </div>
    </div>
  </section>

  <!-- Chat/Comments with overflow scroll -->
  <section class="py-12 bg-white">
    <div class="max-w-4xl mx-auto px-4">
      <h2 class="text-3xl font-bold mb-6">Comments</h2>
      <div class="h-96 overflow-y-auto overflow-x-hidden bg-gray-
50 rounded-lg p-4 space-y-4">
        <div class="bg-white rounded-lg p-4 shadow">
          <div class="flex items-start space-x-3">
            <img
              src="https://i.pravatar.cc/40?img=1"
              class="w-10 h-10 rounded-full object-cover"
              alt="User"
            >
            <div class="flex-1">
              <div class="flex items-center space-x-2 mb-1">
                <span class="font-semibold">John Doe</span>
                <span class="text-xs text-gray-500">2 hours
ago</span>
              </div>
              <p class="text-gray-700 break-words">
                This is a comment with a very long URL:
https://www.example.com/very-long-path/to/resource
              </p>
            </div>
          </div>
        </div>
        <!-- More comments... -->
      </div>

      <!-- Comment form -->
```

```
      <div class="mt-4">
        <textarea
          class="w-full border border-gray-300 rounded-lg p-3
resize-y focus:ring-2 focus:ring-blue-500 focus:border-
transparent"
          placeholder="Add a comment..."
          rows="3"
        ></textarea>
        <button class="mt-2 bg-blue-600 hover:bg-blue-700 text-
white px-6 py-2 rounded-lg transition">
          Post Comment
        </button>
      </div>
    </div>
  </section>

  <!-- Video section with aspect ratio -->
  <section class="py-12">
    <div class="max-w-4xl mx-auto px-4">
      <h2 class="text-3xl font-bold mb-6">Featured Video</h2>
      <div class="aspect-video bg-black rounded-lg overflow-
hidden shadow-2xl">
        <iframe
          src="https://www.youtube.com/embed/dQw4w9WgXcQ"
          class="w-full h-full"
          allow="accelerometer; autoplay; encrypted-media;
gyroscope; picture-in-picture"
          allowfullscreen
        ></iframe>
      </div>
    </div>
  </section>

</body>
</html>
```

Key Takeaways

✅ `object-cover` - Fills container, crops excess (best for thumbnails, avatars)
✅ `object-contain` - Shows entire image, may have empty space (best for logos)
✅ `object-position` - Controls where image is positioned when cropped
✅ `overflow-auto` - Adds scrollbar only when needed
✅ `overflow-hidden` - Hides overflow (for image zoom effects)
✅ `aspect-ratio` - Maintains consistent proportions (video, images, cards)
✅ `line-clamp-{n}` - Limits text to specific number of lines
✅ `truncate` - Single-line ellipsis
✅ `break-words` - Prevents long words from overflowing
✅ `whitespace-nowrap` - Prevents text wrapping
✅ `select-none` - Prevents text selection (for UI elements)
✅ `pointer-events-none` - Makes element not respond to clicks
✅ `scroll-snap` - Creates smooth snap points (carousels, galleries)

Practice Assignment

Create a complete photo gallery page that uses:

1. `aspect-square` for consistent image sizing
2. `object-cover` for proper image display
3. `overflow-x-auto` with `snap-x` for mobile carousel
4. `line-clamp-2` for image captions
5. `hover:scale-110` with `overflow-hidden` for zoom effect
6. Grid layout that's responsive
7. Modal with `overflow-y-auto` for full-size images

Would you like me to create this practice assignment, or do you have questions about any of these properties?

1/7/2026, 1:31:09 PM