

Module 1: Introduction to Tailwind CSS (Day 1–2)

Goal: Understand what Tailwind is and how it differs from normal CSS.

Topics:

- What is Tailwind CSS?
 - Utility-first CSS concept
 - Advantages of Tailwind vs traditional CSS
 - CDN setup (quick start in HTML)
 - Folder structure and config overview
 - Writing your first Tailwind classes
 - Practice:
 - Create a simple landing page using CDN
-

Module 2: Installation and Setup (Day 3–4)

Goal: Install Tailwind properly for projects.

Topics:

- Installing Tailwind via **npm**
 - Initializing Tailwind config (`npx tailwindcss init`)
 - Integrating Tailwind with:
 - HTML project
 - React (Vite or CRA) project
 - Understanding `tailwind.config.js`
 - Customizing colors, fonts, and breakpoints
 - Practice:
 - Create a new React app and add Tailwind
-

Module 3: Utility Classes Essentials (Week 1)

Goal: Learn all core Tailwind building blocks.

Topics:

- Spacing (margin, padding, gap)

- Colors (text-color, bg-color, border-color)
 - Typography (font-size, weight, alignment)
 - Borders, shadows, radius
 - Width, height, min/max sizing
 - Flexbox and grid utilities
 - Practice:
 - Design a profile card layout using Tailwind utilities only
-

Module 4: Responsive Design (Week 2)

Goal: Build mobile-friendly layouts easily.

Topics:

- Mobile-first design principle
 - Breakpoints: sm, md, lg, xl, 2xl
 - Responsive classes syntax (md:text-lg, etc.)
 - Hiding & showing elements (hidden, block, md:flex)
 - Container widths & fluid layouts
 - Practice:
 - Responsive portfolio homepage
 - Navbar that stacks on mobile
-

Module 5: Advanced Styling (Week 3)

Goal: Explore Tailwind's deep customization.

Topics:

- Custom fonts and color palettes
 - Dark mode setup
 - Gradients and background images
 - Hover, focus, and active states
 - Pseudo-classes and pseudo-elements
 - Practice:
 - Create a dark-mode toggle landing page
-

Module 6: Layout Techniques (Week 4)

Goal: Build full web layouts using Tailwind.

Topics:

- Flexbox utilities (justify, items, grow, shrink)
 - Grid layouts (cols, rows, gaps)
 - Positioning (absolute, relative, z-index)
 - Responsive navigation bar
 - Cards and sections
 - Practice:
 - Build a 3-section homepage with navbar and footer
-

Module 7: Tailwind Components & Plugins (Week 5)

Goal: Speed up development using pre-built components.

Topics:

- Using Tailwind UI / Flowbite / DaisyUI
 - Installing plugins (forms, typography, aspect-ratio)
 - Creating reusable custom classes (@apply)
 - Practice:
 - Build a contact form with Tailwind Forms plugin
 - Style a blog layout using typography plugin
-

Module 8: Animations & Transitions (Week 6)

Goal: Add life to your UI.

Topics:

- Transition classes (transition, duration, ease)
- Animation utilities (animate-bounce, animate-pulse)
- Creating custom animations in config file
- Framer Motion + Tailwind (in React projects)
- Practice:
 - Animated button hover
 - Fade-in sections on scroll

Module 9: Working with React + Tailwind (Week 7–8)

Goal: Combine React's logic with Tailwind's design.

Topics:

- Setting up Tailwind with Vite + React
 - Conditional class names (`className={}`)
 - Responsive components (cards, modals, dashboards)
 - Theme switching (dark/light)
 - Building reusable component folders
 - Practice:
 - Build a React dashboard with Tailwind styling
 - Deploy to Vercel
-

Module 10: Optimization & Deployment (Week 9)

Goal: Prepare your Tailwind projects for production.

Topics:

- Purging unused CSS (content config)
 - Minification and performance tuning
 - Deploying Tailwind projects to Netlify / Vercel
 - Best practices & folder structuring
 - Practice:
 - Optimize and deploy one portfolio project
-

Module 11: Final Projects

Goal: Build real-world portfolio projects.

Project Ideas:

1. Personal Portfolio Website
2. E-commerce Landing Page
3. Blog Homepage Layout
4. Dashboard UI
5. Animated Login/Register Form

6. Responsive Pricing Table