

Advanced Components Using Tailwind CSS



Engage and Think



A startup uses Tailwind CSS to build a product page. During the preview, the team notices UI issues—buttons look inconsistent, form fields are misaligned, and the navbar overlaps the hero section. These problems delay the launch.

How can Tailwind CSS help fix these UI issues and ensure a consistent design across devices?

Learning Objectives

By the end of this lesson, you will be able to:

- Demonstrate styling techniques using Tailwind CSS for building responsive buttons
- Apply design principles using the Tailwind utilities to improve user interaction across devices
- Illustrate form validation states using Tailwind modifiers for clear user feedback
- Organize navigation elements using responsive classes for better accessibility in web applications
- Construct layout components using utility classes for consistent content alignment





Styling Essential UI Elements

Essential UI Elements: Introduction

It refers to the core components in a user interface such as buttons, dropdowns, badges, spinners, and progress bars that enable user interaction, provide feedback, and enhance overall usability.

1

Button: Initiates a function or event upon user interaction, typically via a click

2

Dropdown: Displays a list of options from which the user can select

3

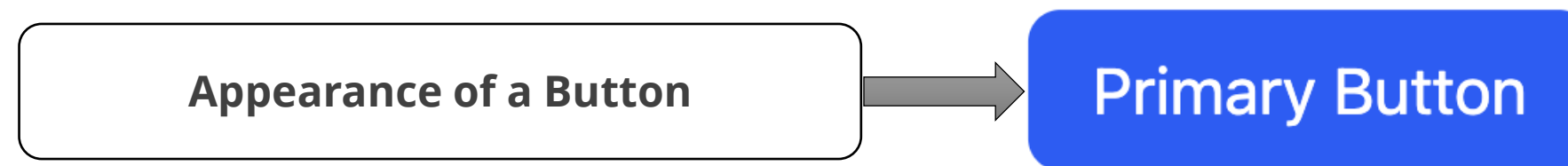
Badge: Serves as a visual marker for alerts, notifications, or counts

4

Spinner / Progress bar: Indicates ongoing processes and indicates loading or completion status

Styling a Button Using Tailwind

A button in web development is a UI element that enables user interaction to trigger an action or event, usually through clicking.



When styled using Tailwind CSS, buttons can adopt consistent design patterns using utility classes that define layout, color, spacing, states (such as hover, active, or disabled), and responsiveness.

Styling a Button Using Tailwind

The following table outlines important styling classes along with their syntax and explanations:

Class Name	Syntax	Explanations
bg-blue-500	<code>class="bg-blue-500"</code>	Sets the background colour to blue with a shade value of 500
text-white	<code>class="text-white"</code>	Sets the text color to white
px-4 py-2	<code>class="px-4 py-2"</code>	Adds horizontal and vertical padding
rounded-lg	<code>class="rounded-lg"</code>	Rounds the corners of the element
hover:bg-blue-600	<code>class="hover:bg-blue-600"</code>	Changes background color when hovered
transition	<code>class="transition"</code>	Enables smooth transitions between states
cursor-pointer	<code>class="cursor-pointer"</code>	Changes the mouse pointer to a hand icon when hovered

Styling a Button Using Tailwind

Tailwind CSS provides utility-first classes to create various button styles. Below are the example button types and their corresponding class combinations:



These buttons use padding (px, py), text color (text-*), background color (bg-*), rounded corners (rounded), and hover effects (hover:bg-*) to reflect interaction and feedback.

Quick Check

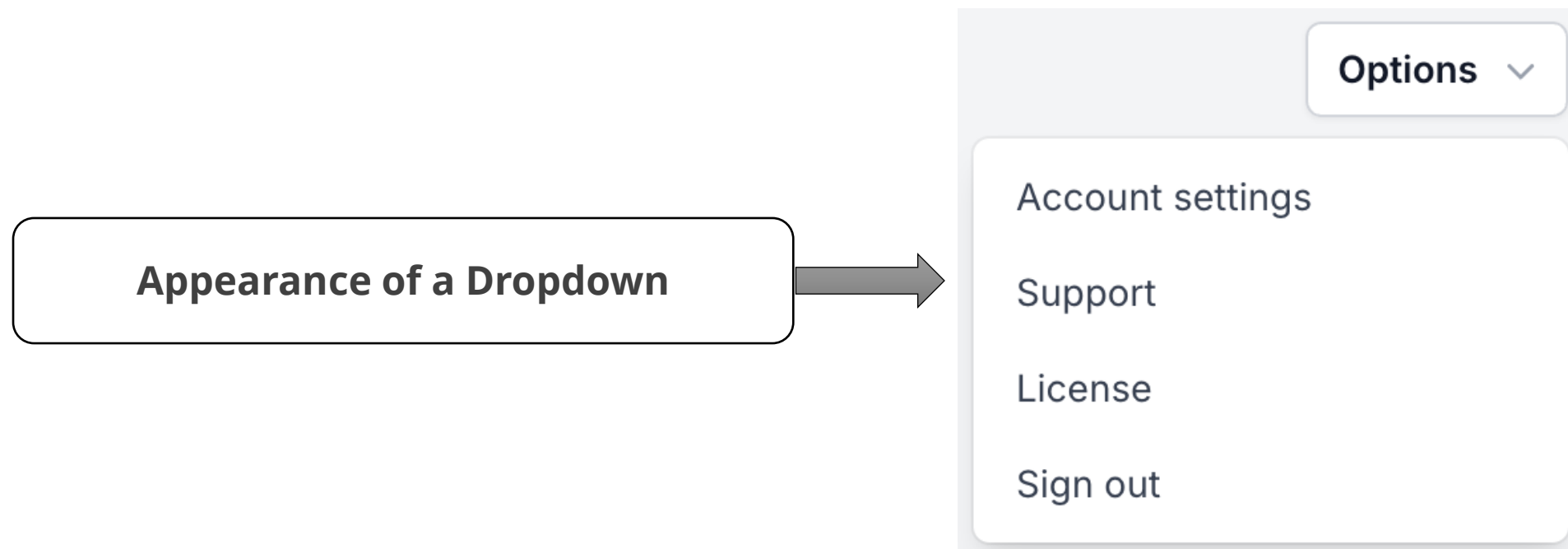


You are styling a *Submit* button using Tailwind CSS. It should have a blue background, white text, padding, rounded corners, a darker hover effect, a pointer cursor on hover, and smooth transitions. Which of the following class combinations best meet all these requirements?

- A. `bg-blue-500 text-black px-4 py-2 rounded-sm`
- B. `bg-blue-500 text-white px-4 py-2 rounded-lg hover:bg-blue-600 cursor-pointer transition`
- C. `bg-white text-blue-500 px-2 py-1 rounded hover:text-blue-600`
- D. `bg-blue-300 text-white hover:bg-blue-500`

Styling a Dropdown Using Tailwind

A dropdown is a user interface element that displays a list of options when clicked, allowing users to select one from the list.



It can be styled using Tailwind CSS to improve usability, accessibility, and visual appeal.

Styling a Dropdown Using Tailwind

Below is a list of important styling classes along with their syntax and explanations:

Class Name	Syntax	Explanations
details	<code><details class="relative w-32"></code>	Sets up the dropdown container with relative positioning and a fixed width of 8rem (128px)
summary	<code><summary class="bg-blue-500 text-white px-4 py-2 rounded-lg cursor-pointer hover:bg-blue-600 transition"></code>	Acts as the toggle button to show/hide the dropdown
bg-blue-500	<code><summary class="bg-blue-500"></code>	Applies a medium blue background
text-white	<code><summary class="text-white"></code>	Sets the font color to white
px-4 py-2	<code><summary class="px-4 py-2"></code>	Adds padding: 1rem left/right, 0.5rem top/bottom
rounded-lg	<code><summary class="rounded-lg"></code>	Applies large-rounded corners (typically 0.5rem radius)

Styling a Dropdown Using Tailwind

Below is a list of important styling classes along with their syntax and explanations:

Class Name	Syntax	Explanations
cursor-pointer	<code><summary class="cursor-pointer"></code>	Changes the mouse cursor to a pointer (hand) on hover
hover:bg-blue-600	<code><summary class="hover:bg-blue-600"></code>	Applies a darker blue background when hovered
transition	<code><summary class="transition"></code>	Enables smooth transitions (For example: hover effects)
absolute	<code><div class="absolute"></code>	Positions the dropdown absolutely within the relative container
left-0	<code><div class="left-0"></code>	Aligns the dropdown to the left edge of its containing block
mt-2	<code><div class="mt-2"></code>	Adds a margin-top of 0.5rem (8px), usually to create spacing between the button and dropdown list

Styling a Dropdown Using Tailwind

Below is a list of important styling classes along with their syntax and explanations:

Class Name	Syntax	Explanations
w-full	<code><div class="w-full"></code>	Makes the dropdown take up 100% width of its parent container
shadow-lg	<code><div class="shadow-lg"></code>	Applies a large shadow, enhancing visual depth
rounded-lg	<code><div class="rounded-lg"></code>	Rounds the corners of the dropdown container (usually 0.5rem)
block	<code></code>	Forces the element to behave as a block element, making it take full width. Useful for clickable dropdown items
px-4 py-2	<code></code>	Adds padding (1rem on the x-axis and 0.5rem on the y-axis)
hover:bg-gray-100	<code></code>	Applies a light gray background on hover for visual feedback

Quick Check

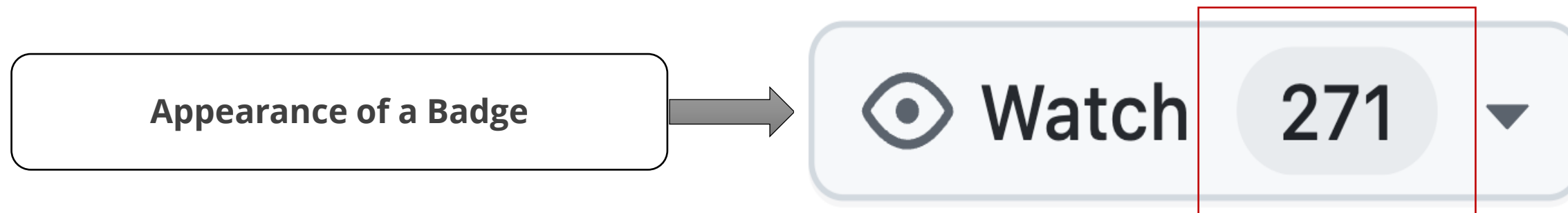
You are building a navigation menu for a web application and want to use Tailwind CSS to implement a dropdown that is both accessible and visually appealing. Which of the following approaches best aligns with Tailwind CSS principles for styling dropdown components?

- A. Use JavaScript for all styling
- B. Use custom CSS to override styles
- C. Avoid Tailwind for dropdowns
- D. Use Tailwind utility classes for layout and appearance



Styling a Badge Using Tailwind

A badge is a compact visual indicator used to display counts, alerts, or status updates within a user interface.



It is often attached to icons or buttons to draw attention to important information.

Styling a Badge Using Tailwind

Below is a list of important styling classes along with their syntax and explanations:

Class Name	Syntax	Explanations
bg-blue-100, bg-red-100, bg-yellow-100, bg-green-100, bg-gray-800, bg-gray-100	<code>bg-{color}-{shade}</code>	Sets the background color of the badge using predefined Tailwind color utilities
text-blue-700, text-red-700, text-yellow-700, text-green-700, text-gray-800, text-white	<code>text-{color}-{shade}</code>	Sets the text color of the badge
px-3, py-1	<code>px-{value}, py-{value}</code>	Adjusts horizontal and vertical padding. Affects size and spacing of content inside the badge
rounded-full	<code>rounded-full</code>	Applies fully rounded corners to create a circular or pill-like shape
text-sm	<code>text-sm</code>	Sets the font size to small, making the badge compact

Styling a Badge Using Tailwind

Badges in Tailwind CSS are used to display short labels or status indicators with distinct background and text colors, helping users quickly identify the state or category of an item.

Active

Inactive

Pending

Success

Dark

Light

These classes help create clear, compact labels for statuses such as Active, Inactive, Pending, Success, Dark, and Light.

Quick Check

You are designing a dashboard that needs to display the status of multiple user accounts. You want to use Tailwind CSS to visually differentiate between statuses like Active, Inactive, Pending, and Success.

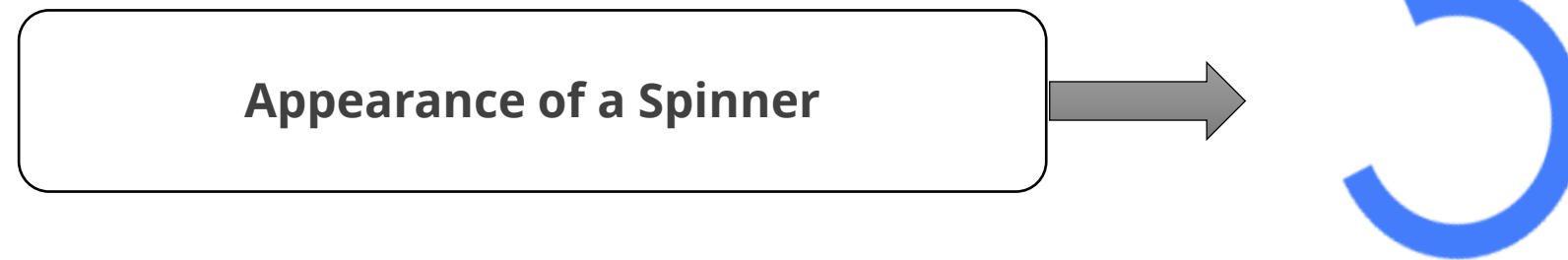
Which of the following Tailwind CSS features will help you create compact and easily distinguishable status labels?

- A. Applying `display: none` to hide status text
- B. Using responsive layout classes like flex and grid
- C. Combining background and text color utility classes for styling badges
- D. Adding hover and focus states to enhance interactivity



Styling a Spinner Using Tailwind

A spinner is a user interface element that visually indicates an ongoing process or operation, such as loading data or waiting for a response.



It is commonly used to signal to users that their request is being processed, helping to manage expectations and improve overall user experience during periods of inactivity.

Styling a Spinner Using Tailwind

Below is a list of important styling classes along with their syntax and explanations:

Class Name	Syntax	Explanations
w-8	<code>class="w-8"</code>	Sets the width to 2rem (8 × 0.25rem)
h-8	<code>class="h-8"</code>	Sets the height to 2rem (8 × 0.25rem)
border-4	<code>class="border-4"</code>	Applies a 4px border width
border-blue-500	<code>class="border-blue-500"</code>	Applies a blue border with the 500 shade
border-t-transparent	<code>class="border-t-transparent"</code>	Makes the top border transparent, which is essential for the spinning effect
rounded-full	<code>class="rounded-full"</code>	Makes the shape fully circular by setting border-radius: 9999px. While the expected value is 50%, Tailwind applies a high value to guarantee a perfect circle. You may revise that for technical accuracy.
animate-spin	<code>class="animate-spin"</code>	Applies continuous spin animation using Tailwind's keyframe utility

Quick Check



You are building a user interface using Tailwind CSS. To show users that the system is processing their request, which visual element would be the most appropriate?

- A. An icon that appears only after the task is completed
- B. A spinner that rotates to indicate ongoing activity
- C. A progress bar that stays at a fixed percentage
- D. A button that changes color when clicked

Styling a Progress Bar Using Tailwind

A progress bar is a user interface component that visually communicates the progression of a task or operation by displaying the percentage of completion along a horizontal or vertical bar.



Progress bars help users track the status of an ongoing task, such as file uploads or form submissions, by indicating how much of the process has been completed and how much remains.

Styling a Progress Bar Using Tailwind

Below is a list of important styling classes along with their syntax and explanations:

Class Name	Syntax	Explanations
w-full	<code><div class="w-full"></code>	Sets the width of an element to 100% of its parent container
max-w-md	<code><div class="max-w-md"></code>	Restricts the maximum width to a predefined medium value (typically 28rem or 448px).
bg-gray-200	<code><div class="bg-gray-200"></code>	Applies a light gray background color
rounded-full	<code><div class="rounded-full"></code>	Applies a fully rounded border radius to make corners completely circular
h-4	<code><div class="h-4"></code>	Sets the height to 1rem (16px)
overflow-hidden	<code><div class="overflow-hidden"></code>	Ensures content beyond the element's bounds is not visible

Styling a Progress Bar Using Tailwind

Below is a list of important styling classes along with their syntax and explanations:

Class Name	Syntax	Explanations
bg-blue-500	<code><div class="bg-blue-500"></code>	Applies a medium blue background color to an element (For example: progress fill)
text-sm	<code></code>	Sets the font size to small (typically 0.875rem or 14px)
mt-1	<code></code>	Adds a small top margin (0.25rem or 4px) to create spacing
block	<code></code>	Displays the element as a block-level element, which takes the full width
text-gray-700	<code></code>	Applies a dark gray text color for improved readability

Quick Check



You are building a form that allows users to upload documents. You want to visually show how much of the upload process is complete using a progress bar. Which feature is most important for this progress bar to include?

- A. Icons to represent each file type
- B. Numeric display showing percentage completed
- C. Cancel button to stop the upload
- D. Thumbnail preview of the uploaded files

Assisted Practice



Styling Essential Elements Using Tailwind

Duration: 10 Min.

Problem statement:

You have been tasked to create an interactive web page using Tailwind CSS to style essential UI elements, including buttons, dropdowns, badges, spinner, and progress bar.

Outcome:

By the end of this demo, you will be able to style and configure buttons, dropdowns, badges, spinners, and progress bars using Tailwind utility classes. Ensure responsive design and smooth user interaction.

Note: Refer to the demo document for detailed steps:
01_Styling_Essential_Elements_using_Tailwind

Assisted Practice: Guidelines



Steps to be followed:

1. Set up the project folder and base HTML structure
2. Create and style the button components
3. Create a dropdown menu with three selectable options
4. Design badge elements
5. Add a loading spinner
6. Build a progress bar
7. Run the implementation in a web browser



Styling Presentation UI Elements

Presentation Elements in Tailwind

Tailwind CSS provides structured components that assist in presenting content effectively to users. These components include hero sections, tables, and lists, each serving a distinct purpose in organizing and displaying information on a webpage.

01

Hero section: Designed to capture user attention at the top of a webpage and introduce the site or emphasize key content

02

Tables: Structured to present data systematically in rows and columns with customizable styling options

03

Lists: Organized to display multiple items using various list types and utility classes for enhanced readability

Styling a Hero Section Using Tailwind

A Hero Section is a prominent and visually impactful area located at the top of a webpage, designed to introduce the brand, product, or core message to users.

Appearance of a Progress Bar



Welcome to Tailwind CSS.

Learn how to create modern UI components with Tailwind CSS.

Get Started

It helps in establishing a strong first impression and guides users toward key content or calls to action.

Styling a Hero Section Using Tailwind

Below is a list of important styling classes along with their syntax and explanations for background and text colors:

Class Name	Syntax	Explanations
bg-blue-600	<code>class="bg-blue-600"</code>	Sets the background color to a shade of blue (600 intensity)
text-white	<code>class="text-white"</code>	Sets the text color to white
bg-white	<code>class="bg-white"</code>	Sets the background color to white
text-blue-600	<code>class="text-blue-600"</code>	Sets the text color to a medium-dark shade of blue (intensity 600)
hover:bg-gray-100	<code>class="hover:bg-gray-100"</code>	Changes the background color to light gray when the user hovers over the element

Styling a Hero Section Using Tailwind

Below is a list of important styling classes along with their syntax and explanations for padding and margin:

Class Name	Syntax	Explanations
py-20	<code>class="py-20"</code>	Adds vertical padding of 20 units (5 rem)
px-5	<code>class="px-5"</code>	Adds horizontal padding of 5 units (1.25 rem)
px-6	<code>class="px-6"</code>	Adds horizontal padding of 6 units (1.5 rem)
py-3	<code>class="py-3"</code>	Adds vertical padding of 3 units (0.75 rem)
mb-4	<code>class="mb-4"</code>	Adds a bottom margin of 4 units (1 rem)
mb-6	<code>class="mb-6"</code>	Adds a bottom margin of 6 units (1.5 rem)
mx-auto	<code>class="mx-auto"</code>	Centers the element horizontally by setting automatic left and right margins

Styling a Hero Section Using Tailwind

Below is a list of important styling classes along with their syntax and explanations for text styling:

Class Name	Syntax	Explanations
text-4xl	<code>class="text-4xl"</code>	Sets the text size to 4xl (2.25 rem)
text-lg	<code>class="text-lg"</code>	Sets the text size to large (lg, 1.125 rem)
font-bold	<code>class="font-bold"</code>	Makes the text bold
text-center	<code>class="text-center"</code>	Centers the text horizontally

Styling a Hero Section Using Tailwind

Below is a list of important styling classes along with their syntax and explanations for layout sizing and transition effects:

Class Name	Syntax	Explanations
max-w-3xl	<code>class="max-w-3xl"</code>	Sets the maximum width of the content to 48rem (equivalent to the 3xl breakpoint)
rounded-lg	<code>class="rounded-lg"</code>	Applies a large border radius to round the corners of the element
shadow-md	<code>class="shadow-md"</code>	Applies a medium-sized box shadow to the element
transition	<code>class="transition"</code>	Enables smooth transition effects for interactive state changes (e.g., hover, focus)
duration-300	<code>class="duration-300"</code>	Sets the duration of the transition effect to 300 milliseconds

Quick Check



You are designing a responsive webpage and want to add vertical padding of 5 rem to a section. Which Tailwind class should you use?

- A. py-20
- B. px-20
- C. mb-6
- D. mx-auto

Styling a Table Using Tailwind

Tables are HTML elements used to present data in rows and columns. They help display structured information clearly and efficiently, making it easier for users to scan and understand.

Appearance of a Progress Bar

Name	Email	Role
John Doe	john@example.com	Admin
Jane Smith	jane@example.com	User
Michael Johnson	michael@example.com	Moderator

Use Tailwind CSS to add borders, padding, background colors, and hover effects to enhance table readability and user experience.

Styling a Table Using Tailwind

Below is a list of important styling classes along with their syntax and explanations for container and layout:

Class Name	Syntax	Explanations
overflow-x-auto	<code>class="overflow-x-auto"</code>	Enables horizontal scrolling if the table overflows its container
my-10	<code>class="my-10"</code>	Adds vertical margin of 10 units (2.5 rem) above and below the container
min-w-full	<code>class="min-w-full"</code>	Sets the minimum width of the table to 100% of the parent container
shadow-md	<code>class="shadow-md"</code>	Applies a medium box shadow to the table for a raised effect
rounded-lg	<code>class="rounded-lg"</code>	Applies a large border-radius to round the corners of the table

Styling a Table Using Tailwind

Below is a list of important styling classes along with their syntax and explanations for border background colors:

Class Name	Syntax	Explanations
border	<code>class="border"</code>	Adds a border around the table
border-gray-300	<code>class="border-gray-300"</code>	Sets the border color to a light gray (300 intensity)
bg-white	<code>class="bg-white"</code>	Sets the table background color to white
bg-gray-200	<code>class="bg-gray-200"</code>	Sets the background color of the table header to light gray (200 intensity)
hover:bg-gray-100	<code>class="hover:bg-gray-100"</code>	Changes the row background color to light gray (100 intensity) on hover

Styling a Table Using Tailwind

Below is a list of important styling classes along with their syntax and explanations for text alignment and padding:

Class Name	Syntax	Explanations
text-left	<code>class="text-left"</code>	Aligns text to the left within table header cells
py-3	<code>class="py-3"</code>	Adds vertical padding of 3 units (0.75 rem) to header cells
px-6	<code>class="px-6"</code>	Adds horizontal padding of 6 units (1.5 rem) to header cells
py-4	<code>class="py-4"</code>	Adds vertical padding of 4 units (1 rem) to body cells

Quick Check

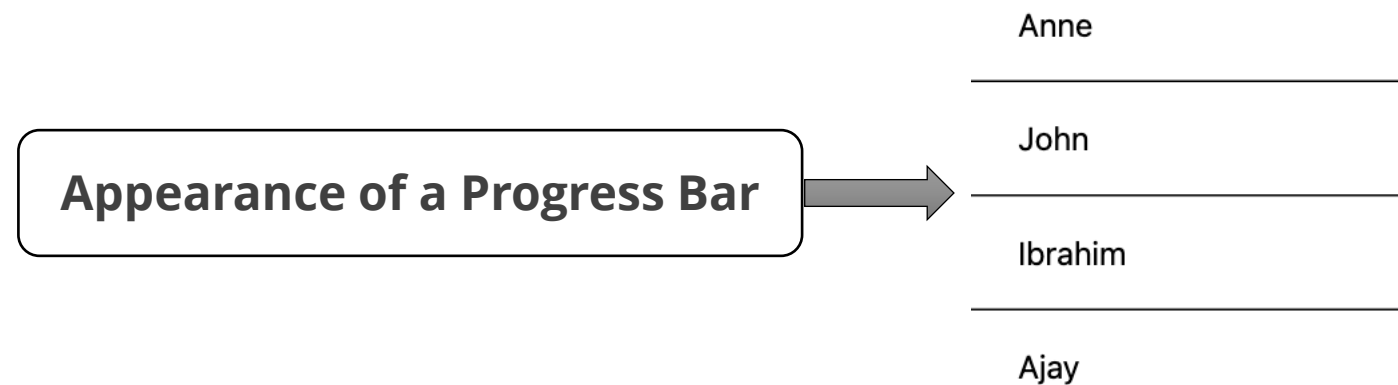


You are styling a table using Tailwind CSS. You want to make the table background white, the header a light gray, and highlight each row on hover. Which combination of classes will achieve this?

- A. `bg-gray-200 border-gray-300 hover:bg-white`
- B. `bg-gray-100 border-white hover:bg-gray-200`
- C. `bg-white border-gray-300 hover:bg-gray-100`
- D. `bg-white border-gray-100 hover:bg-gray-300`

Styling a List Using Tailwind

A list is a structured format used to display related items in a readable and organized sequence.



Use Tailwind utility classes to enhance the visual layout of lists, such as spacing, borders, and hover effects, for better user experience

Styling a List Using Tailwind

Below is a list of important styling classes along with their syntax and explanations for container and layout:

Class Name	Syntax	Explanations
max-w-md	<code>class="max-w-md"</code>	Sets the maximum width of the container to 28rem (medium size)
my-10	<code>class="my-10"</code>	Adds vertical margin of 10 units (2.5 rem) above and below the container
bg-white	<code>class="bg-white"</code>	Sets the background color of the container to white
rounded-lg	<code>class="rounded-lg"</code>	Applies a large border-radius to round the corners of the container
overflow-hidden	<code>class="overflow-hidden"</code>	Prevents overflow content from being visible outside the container's boundaries

Styling a List Using Tailwind

Below is a list of important styling classes along with their syntax and explanations for padding, borders and hover effects:

Class Name	Syntax	Explanations
px-6	<code>class="px-6"</code>	Adds horizontal padding of 6 units (1.5 rem) to each list item
py-4	<code>class="py-4"</code>	Adds vertical padding of 4 units (1 rem) to each list item
border-b	<code>class="border-b"</code>	Adds a bottom border to each list item (except the last one)
hover:bg-gray-100	<code>class="hover:bg-gray-100"</code>	Changes the background color to light gray (100 intensity) when the item is hovered over

Quick Check



You are styling a list using Tailwind CSS. Each item should have padding, a bottom border (except the last one), and change to light gray when hovered. Which combination of classes should be applied to each list item?

- A. `px-6 border-b py-4 hover:bg-gray-100`
- B. `py-4 px-4 border-b hover:border-gray-100`
- C. `px-6 py-2 border hover:bg-gray-300`
- D. `py-4 border px-6 hover:bg-gray-100`

Assisted Practice



Styling Presentation Elements using Tailwind

Duration: 40 Min.

Problem statement:

You have been tasked to create a responsive hero section using Tailwind CSS to welcome users with a headline, subtext, and a call-to-action button.

Outcome:

By the end of this demo, you will be able to style and configure a hero section using Tailwind utility classes, ensuring proper alignment, spacing, and responsive behavior.

Note: Refer to the demo document for detailed steps:
02_Styling_Presentation_Elements_using_Tailwind

Assisted Practice: Guidelines



Steps to be followed:

1. Open Visual Studio Code and create index.html
2. Add and style presentation UI components



Styling Form Input UI Elements

Styling a Form Using Tailwind

Forms are interactive components of a web application that are used to gather, validate, and submit user-provided data to a server for processing.

Appearance of a Progress Bar



Registration Form

Full Name

Email

Password

☐ Accept Terms and Conditions

Gender

☐ Male ☐ Female

Country

Select Country ▾

Upload Profile Picture

Choose file No file chosen

Register

They play a critical role in enabling communication between users and the system, supporting functions such as registration, login, feedback, and data submission.

Styling a Form Using Tailwind

Below is a list of important styling classes along with their syntax and explanations for container and layout:

Class Name	Syntax	Explanations
space-y-4	<code>class="space-y-4"</code>	Adds vertical spacing of 1rem (4 units) between vertically stacked elements
flex	<code>class="flex"</code>	Applies a flex container to arrange child elements in a row or column
flex-col	<code>class="flex-col"</code>	Stacks flex items vertically (in a column layout)
items-center	<code>class="items-center"</code>	Aligns flex items to the center along the cross-axis (vertical in a row layout)
block	<code>class="block"</code>	Displays the element as a block, making it span the full width of the parent
space-x-4	<code>class="space-x-4"</code>	Adds horizontal spacing of 1rem (4 units) between elements in a horizontal row

Styling a Form Using Tailwind

Below is a list of important styling classes along with their syntax and explanations for typography:

Class Name	Syntax	Explanations
text-lg	<code>class="text-lg"</code>	Sets the font size to large (1.125rem)
font-medium	<code>class="font-medium"</code>	Applies medium font weight
text-gray-600	<code>class="text-gray-600"</code>	Sets the text color to gray with 600 intensity
text-gray-700	<code>class="text-gray-700"</code>	Sets the text color to gray with 700 intensity
text-red-500	<code>class="text-red-500"</code>	Sets the text color to red with 500 intensity
text-xs	<code>class="text-xs"</code>	Sets the font size to extra small (0.75rem)

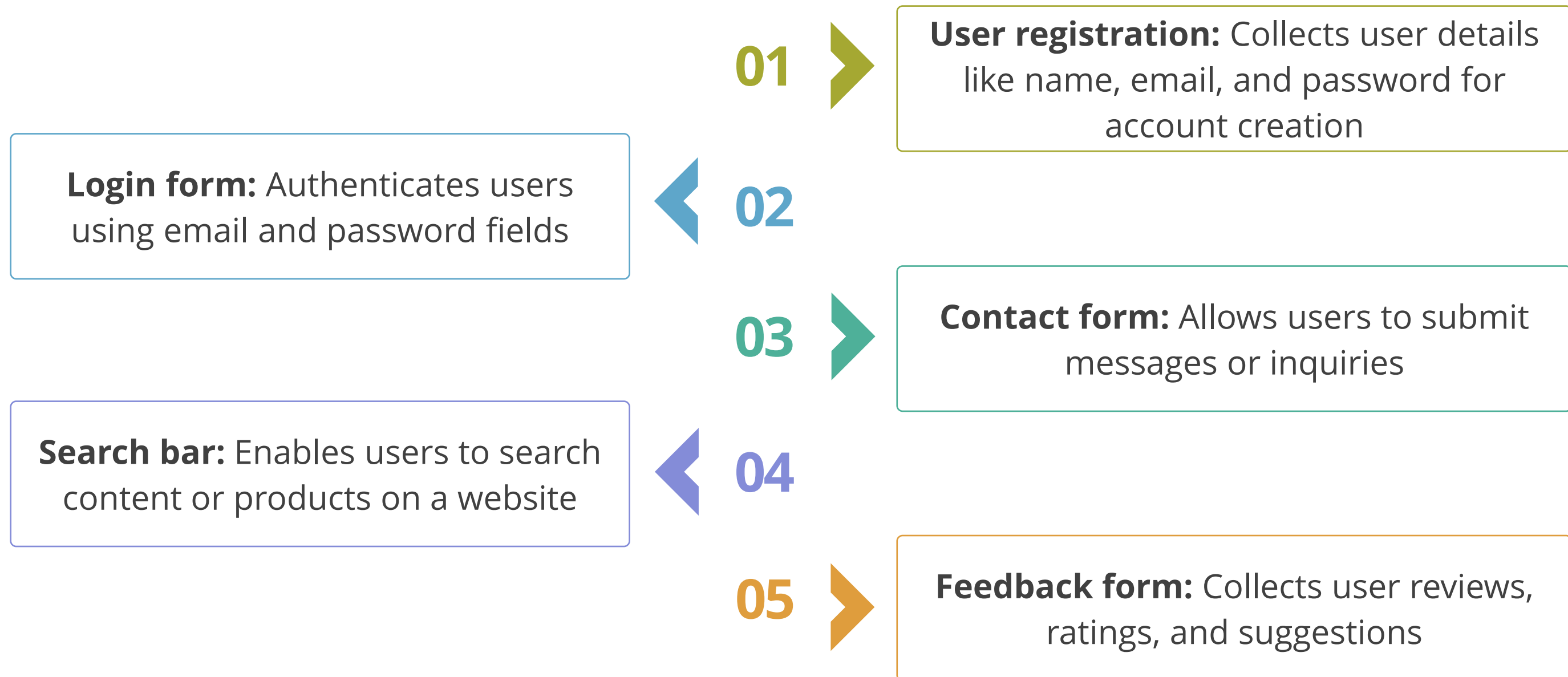
Styling a Form Using Tailwind

Below is a list of important styling classes along with their syntax and explanations for form and inputs:

Class Name	Syntax	Explanations
border-gray-300	class="border-gray-300"	Sets the border color to gray with 300 intensity
rounded-md	class="rounded-md"	Applies a medium border radius to round the corners of the input element
focus:outline-none	class="focus:outline-none"	Removes the default outline when the input is focused
focus:ring-2	class="focus:ring-2"	Applies a ring (outline) of 2 pixels when the input is focused
focus:ring-blue-400	class="focus:ring-blue-400"	Sets the color of the focus ring to blue with 400 intensity

Form: Use Cases

Common use cases of form includes:



Quick Check



You are designing a web page for a new service-based company. One of your goals is to help users contact customer support by submitting their issues or questions directly through the website. Which form is most suitable for allowing users to send messages or inquiries to the support team?

- A. Login form
- B. Feedback form
- C. Search bar
- D. Contact form

Assisted Practice



Styling Input Elements Using Tailwind CSS

Duration: 40 Min.

Problem statement:

You have been tasked to create a responsive form using Tailwind CSS to collect user information, including text inputs, radio buttons, checkboxes, and a submit button.

Outcome:

By the end of this demo, you will be able to style and configure form inputs using Tailwind utility classes, ensuring proper alignment, spacing, focus states, and responsive behavior to create a user-friendly and professional form interface.

Note: Refer to the demo document for detailed steps:
03_Styling_Input_Elements_Using_Tailwind

Assisted Practice: Guidelines



Steps to be followed:

1. Open Visual Studio Code and create index.html
2. Implement the styled form layout



Styling Navigation UI Elements

Navigation UI Elements: Introduction

These are the interface components that help users locate, access, and move between different sections or pages within a digital product, such as a website or application.

Defines the layout and guides users across web or mobile interfaces

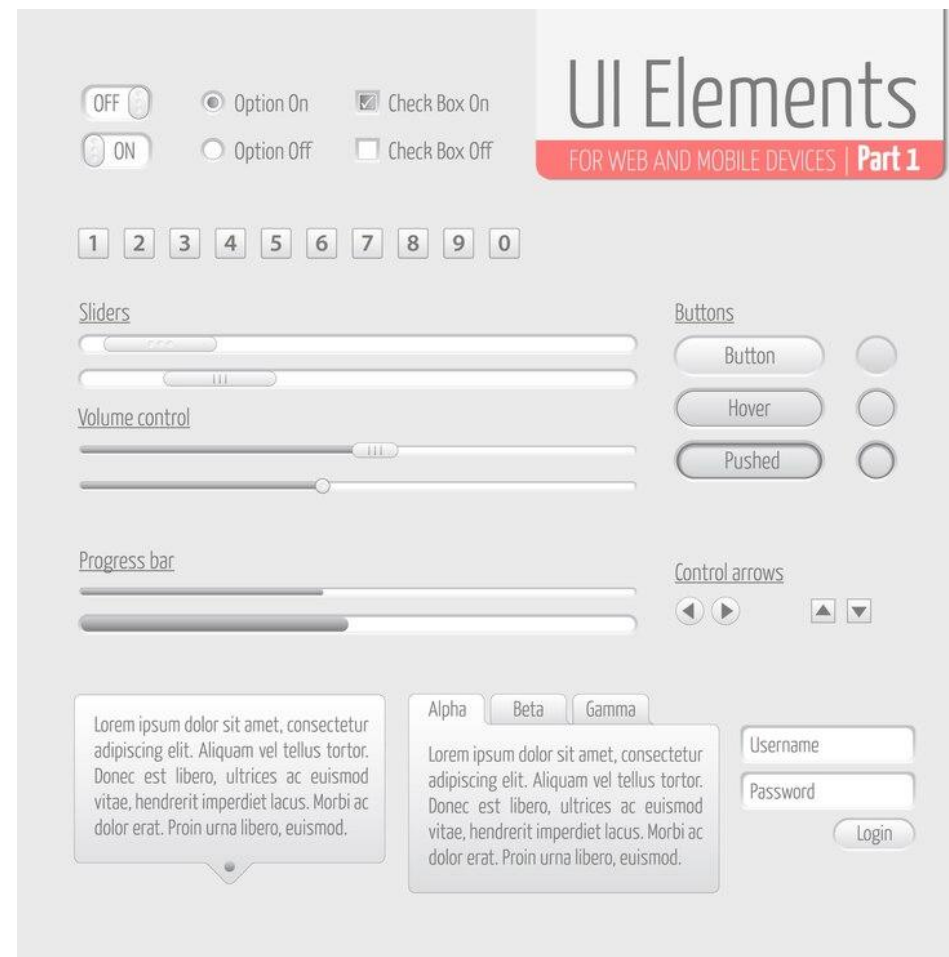
Includes components such as navbars and breadcrumbs

Key points

Enables responsive and customizable styling using Tailwind CSS

Navigation UI Elements: Navbar and Breadcrumbs

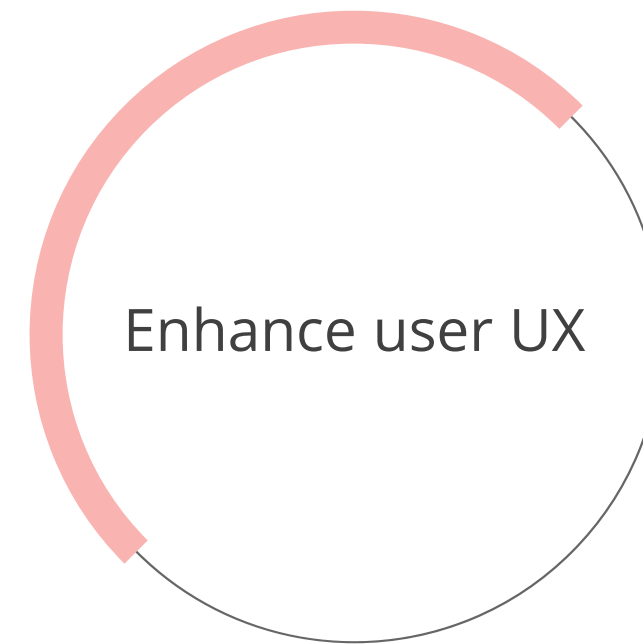
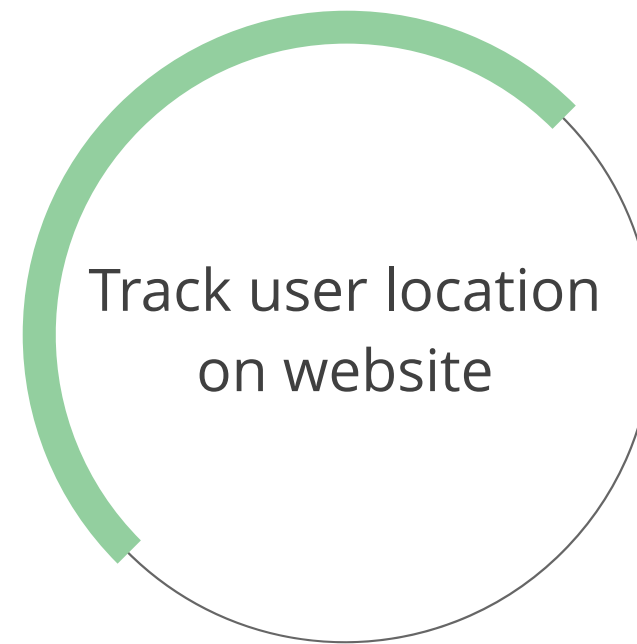
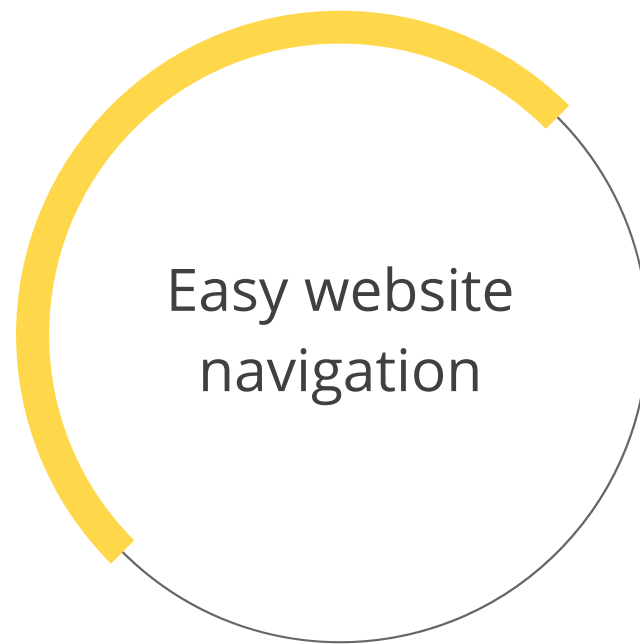
A navbar is a horizontal or vertical user interface element that displays links to the main sections of a website or application, enabling users to quickly access key pages or features.



Breadcrumbs are a secondary navigation aid that displays the user's current location within a hierarchical structure, allowing easy navigation back to previous levels or categories.

Navbar and Breadcrumbs: Applications

Key applications of navbar and breadcrumbs include:



Navbars and breadcrumbs enhance user experience by providing clear navigation paths and helping users track their location within a site.

Styling a Navbar Using Tailwind

A navbar is a menu that helps users move between main sections of a website or app, usually placed at the top or side of the screen.

Appearance of a Navbar



```
graph TD; A[Appearance of a Navbar] --> B[Navbar Example];
```

App Name

Home

About

Services

Contact

A navbar organizes links to important pages, providing users with a consistent and structured navigation experience.

Styling a Navbar Using Tailwind

Below is a list of important styling classes along with their syntax and explanations for flex and layout:

Class Name	Syntax	Explanations
flex	<code>class="flex"</code>	Applies a flex container, enabling horizontal or vertical alignment of child elements
justify-around	<code>class="justify-around"</code>	Distributes items evenly with equal space around them
items-center	<code>class="items-center"</code>	Vertically centers items along the cross axis (perpendicular to main axis)
flex-shrink-0	<code>class="flex-shrink-0"</code>	Prevents the item from shrinking when the flex container is resized
hidden	<code>class="hidden"</code>	Hides the element completely from view
md:flex	<code>class="md:flex"</code>	Applies flex layout starting from medium screen sizes and above
md:hidden	<code>class="md:hidden"</code>	Hides the element on medium screens and larger

Styling a Navbar Using Tailwind

Below is a list of important styling classes along with their syntax and explanations for background and text color:

Class Name	Syntax	Explanations
bg-blue-600	<code>class="bg-blue-600"</code>	Sets the background color to blue with 600 intensity
text-white	<code>class="text-white"</code>	Sets the text color to white

Styling a Navbar Using Tailwind

Below is a list of important styling classes along with their syntax and explanations for spacing and sizing:

Class Name	Syntax	Explanations
space-x-8	<code>class="space-x-8"</code>	Adds horizontal spacing of 2rem (8 units) between flex items
max-w-7xl	<code>class="max-w-7xl"</code>	Sets the maximum width of the container to 80rem (7xl size)
px-4	<code>class="px-4"</code>	Adds horizontal padding of 1rem (4 units)
sm:px-6	<code>class="sm:px-6"</code>	Adds horizontal padding of 1.5rem (6 units) on small screens and larger
lg:px-8	<code>class="lg:px-8"</code>	Adds horizontal padding of 2rem (8 units) on large screens and larger
space-y-2	<code>class="space-y-2"</code>	Adds vertical spacing of 0.5rem (2 units) between child elements
pb-4	<code>class="pb-4"</code>	Adds padding at the bottom of the element

Styling a Navbar Using Tailwind

Below is a list of important styling classes along with their syntax and explanations for borders and focus effects:

Class Name	Syntax	Explanations
rounded-md	<code>class="rounded-md"</code>	Rounds the corners of the element with medium radius
hover:bg-blue-500	<code>class="hover:bg-blue-500"</code>	Changes the background color to blue with 500 intensity on hover
focus:outline-none	<code>class="focus:outline-none"</code>	Removes the default focus outline when the element is focused

Quick Check



You are building a website that needs to help users move quickly between different pages and show them exactly where they are within the site structure. You want to ensure both orientation and easy access to key pages. Which two navigation UI elements should you include to meet both goals?

- A. Sidebar and button group
- B. Navbar and breadcrumb
- C. Modal and alert box
- D. Card layout and image carousel

Styling a Breadcrumb Using Tailwind

A breadcrumb is a user interface element that displays a navigational path, showing the user's current location within a website or application's hierarchy and allowing them to return to previous pages.

Appearance of a Breadcrumb



```
graph TD; A[Appearance of a Breadcrumb] --> B[Home / Services / Service1];
```

[Home](#) / [Services](#) / Service1

A breadcrumb improves navigation by helping users trace and backtrack their journey within the site structure.

Styling a Breadcrumb Using Tailwind

Below is a list of important styling classes along with their syntax and explanations for flex and layout:

Class Name	Syntax	Explanations
flex	<code>flex</code>	Applies display: flex to make the element a flex container
inline-flex	<code>inline-flex</code>	Applies display: inline-flex, making the element behave like an inline-level flex container
items-center	<code>items-center</code>	Aligns flex items vertically along the center of the container
space-x-1	<code>space-x-1</code>	Adds horizontal spacing of 0.25rem (4px) between flex items

Styling a Breadcrumb Using Tailwind

Below is a list of important styling classes along with their syntax and explanations for typography:

Class Name	Syntax	Explanations
text-gray-500	<code>text-gray-500</code>	Sets the text color to gray at a 500 intensity level
text-blue-600	<code>text-blue-600</code>	Sets the text color to blue at a 600 intensity level
hover:text-blue-400	<code>hover:text-blue-400</code>	Changes the text color to blue at a 400 intensity level when the element is hovered
mx-2	<code>mx-2</code>	Applies horizontal margin of 0.5rem (8px) to the element

Assisted Practice



Styling Navigation Elements Using Tailwind CSS

Duration: 40 Min.

Problem statement:

You have been tasked to implement responsive navigation UI components using Tailwind CSS. This includes building navigation bars, dropdowns for smaller screens, and breadcrumbs, enabling enhanced user experience and smooth navigation flow.

Outcome:

By the end of this demo, you will be able to create a responsive navigation bar, mobile menu toggle, and breadcrumb navigation using Tailwind CSS to enhance user experience and layout responsiveness.

Note: Refer to the demo document for detailed steps:
[04_Styling_Navigation_Elements_Using_Tailwind_CSS](#)

Assisted Practice: Guidelines



Steps to be followed:

1. Open Visual Studio Code and create index.html
2. Add navigation elements code

Quick Check



You are navigating a product catalog on an e-commerce website. You first clicked on Home, then Electronics, and finally Mobile Phones. You now want to go back to the Electronics section without restarting your search. Which navigation UI element helps you return to the previous category quickly?

- A. Navbar
- B. Sidebar
- C. Breadcrumb
- D. Footer links

Key Takeaways

- 🕒 A button in web development is a user interface element designed to initiate an action or event when interacted with, typically by clicking.
- 🕒 A spinner is a user interface element that visually indicates an ongoing process or operation, such as loading data or waiting for a response.
- 🕒 A list is a structured format used to display related items in a readable and organized sequence.
- 🕒 The Tailwind grid system enables the creation of complex, table-like layouts by defining rows, columns, and spacing using utility classes.
- 🕒 The align-items property in CSS Flexbox controls the vertical alignment of flex items along the cross axis within a flex container.



Developing a Dashboard for an E-Commerce Web Application Using Tailwind CSS



Project Agenda: To develop a Tailwind CSS-based E-commerce Dashboard that displays key store metrics, product listings, and order details using only HTML and Tailwind CSS. The focus will be on maximum utilization of Tailwind utility classes without any JavaScript or backend integration.

Description: You are tasked with designing a fully responsive dashboard for an e-commerce site using Tailwind CSS. The dashboard will include a sidebar for navigation, a header with user information, and main content areas displaying product lists, recent orders, and sales summaries using tables and card layouts. This project emphasizes using Tailwind's utility-first approach for styling, spacing, typography, and responsiveness.

Developing a Dashboard for an E-Commerce Web Application Using Tailwind CSS



Steps to be performed:

1. Set up a new index.html file
2. Add the dashboard UI related HTML code
3. Run the code

Expected deliverables: A fully responsive and visually appealing e-commerce dashboard built using Tailwind CSS. The dashboard should include:

- A fixed navbar for navigation with dummy links like Profile and Logout
- A product list styled using Tailwind's grid or flex utilities
- A clean and professional UI, ensuring responsiveness across different screen sizes using Tailwind's responsive utility classes



Thank You