

Tailwind CSS

Tailwind CSS : Content

- What is Tailwind CSS?
- Why Tailwind CSS?
- Pre - Requisites
- Bootstrap VS Tailwind
- Utility-First Approach
- Tools Required
- First Project
- First Project - Node
- Utility Classes
- Sample Code
- Questionnaire

What is Tailwind CSS?

- ❶ Tailwind is an open source CSS framework released on 1st November 2017, In contrast to the traditional OOCSS Framework, like Bootstrap or Materialize etc.,
- ❷ It is so popular and the fastest-growing framework available.
- ❸ Tailwind is a CSS Utility First approach framework that makes our life as a front end and back end developer a lot easier.
- ❹ It is a html element driven CSS, where in we append CSS – utilities to design unique user interfaces

Why the name Tailwind so Given?

- As the wind blowing behind a vehicle makes vehicle moves faster and faster, Similarly as long as the requirement or changes needed we can append CSS to the html element to design unique and customized user interfaces.

Why Tailwind CSS?

Benefits/Advantages:

- It is Highly Customizable
- It Has Common Utility Patterns.
- It Can Be Optimized Using PurgeCSS.
- It Enables Building Complex Responsive Layouts Freely.
- You can easily interact with the community.

Limitations/Disadvantages

- It is Necessary to Build a Step
- Larger HTML Sizes

Pre- Requisites

1 Basic HTML

- Understanding HTML tags and elements.
- Familiarity with HTML attributes and their usage.
- Knowledge of how to structure a webpage using HTML.

2 Basic CSS

- Understanding of CSS syntax and properties.
- Familiarity with CSS selectors, combinators, and specificity.
- Knowledge of how to apply styles to HTML elements.

3 Basic JavaScript (Optional)

- While not mandatory for learning Tailwind CSS itself, basic JavaScript knowledge can be helpful.
- Understanding how to manipulate the DOM with JavaScript.

What is Bootstrap?

- Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development.
- It contains CSS and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.
- It aims to ease the development of dynamic websites and web applications.

Bootstrap VS Tailwind

Bootstrap	Tailwind CSS
It is one of the most well-liked HTML, CSS, and JS frameworks for building apps that are mobile-first and responsive.	This is the most well-liked CSS framework for developing unique user interfaces.
It provides pre-made layouts and themes.	It provides distinctive utility-first courses
Since Bootstrap has built-in website templates, applications made with it tend to be the same	Tailwind CSS allows for the creation of distinctive and adaptable applications and web pages.
An earlier framework noted for its responsiveness and effectiveness is called Bootstrap.	A more recent framework, Tailwind CSS, is continually developing and expanding.
It requires a large file size	It requires a small file size

Example

Bootstrap VS Tailwind

```
<button type="submit" class="btn btn-primary">Primary</button>
```



```
<button type="submit" class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded">Primary</button>
```


Example Continued

Classes for Bootstrap

```
<button type="submit" class="btn btn-primary">Primary</button>
```

```
.btn{  
  display:inline-block;  
  font-weight:400;  
  color:#212529;  
  text-align:center;  
  vertical-align:middle;  
  -webkit-user-select:none;  
  -moz-user-select:none;  
  -ms-user-select:none;  
  user-select:none;  
  background-color:transparent;  
  border:1px solid transparent;  
  padding:.375rem .75rem;  
  font-size:1rem;  
  line-height:1.5;  
  border-radius:.25rem;  
  transition:color .15s ease-in-out;  
}
```

```
.btn-primary {  
  color:#fff;  
  background-color:#007bff;  
  border-color:#007bff;  
}
```

Example Continued

Classes for Tailwind

```
<button type="submit" class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded">Primary</button>
```

```
.bg-blue-500 {  
  --tw-bg-opacity: 1;  
  background-color: rgba(59, 130, 246, var(--tw-bg-opacity));  
}
```

```
.hover\:bg-blue-500:hover {  
  --tw-bg-opacity: 1;  
  background-color: rgba(59, 130, 246, var(--tw-bg-opacity));  
}
```

```
.text-white {  
  --tw-text-opacity: 1;  
  color: rgba(255, 255, 255, var(--tw-text-opacity));  
}
```

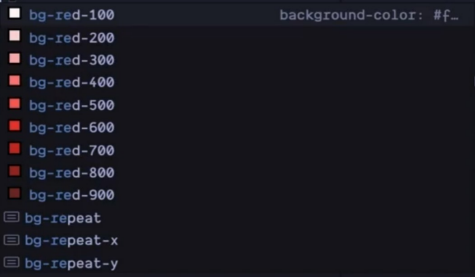
```
.font-bold {  
  font-weight: 700;  
}
```

```
.py-2 {  
  padding-top: 0.5rem;  
  padding-bottom: 0.5rem;  
}
```

```
.rounded {  
  border-radius: 0.25rem;  
}
```

Tailwind CSS Has More Options:

```
1 <div class="md:flex">
2   <div class="md:flex-shrink-0">
3     
4   </div>
5   <div class="mt-4 md:mt-0 md:ml-6 bg-red-100">
6     <div class="uppercase tracking-wide">
7       <a href="#" class="block mt-1 text-l hover:underline">Finding customers f
8     <p class="mt-2 text-gray-600">Gettin
9     are five ideas you can use to find y
10  </div>
11 </div>
```



Utility First Approach

- It Mainly focuses on using small, single-purpose utility classes to build custom designs directly in your HTML

Key Concepts of Utility-First Approach

- 1 **Single-purpose Classes:** Classes are designed to do one thing well, such as `'text-center'` for centering text or `'bg-blue-500'` for applying a specific background color.
- 2 **Predictability:** The classes are named in a way that makes it clear what they do, which improves readability and maintainability.
- 3 **Composability:** By combining multiple utility classes, you can achieve complex designs without writing custom CSS.
- 4 **Speed of Development:** It allows for rapid prototyping and development since you don't need to create custom styles for every element.

Benefits of Utility First Approach

- **Consistency:** Ensures a consistent look and feel across the application by using predefined classes.
- **Reusability:** The Classes can be reused across different components, reducing the need to write new CSS for every element.
- **Maintainability:** Easier to maintain and update since changes are made in a predictable manner.
- **Performance:** Reduces the size of the CSS file by reusing utility classes, which can improve load times.

Tools Required?

① Code Editor(Any)

- Visual Studio Code (VSCode): A popular, feature-rich code editor with numerous extensions.
- Sublime Text: A lightweight and fast editor.
- Atom: An open-source editor with a strong community.

② Node.js and npm (Node Package Manager)

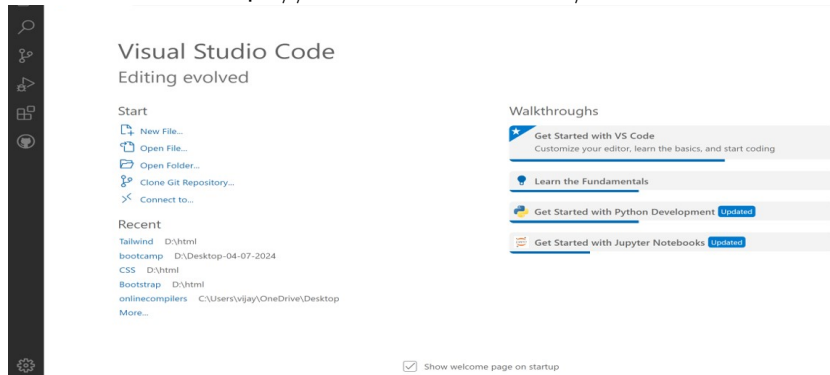
- Node.js: A JavaScript runtime required to use npm.
- npm: The package manager for Node.js, used to install Tailwind CSS and other dependencies.

③ Web Browser(Any)

- Google Chrome: Offers robust developer tools for inspecting and debugging CSS.
- Firefox: Also provides excellent developer tools and is known for its CSS Grid and Flexbox support.
- Internet Ex: Also provides excellent developer tools and is known for its CSS Grid and Flexbox support.

Visual Studio Code

Download Link: <https://code.visualstudio.com/>



Node.js and npm(Node package manager)

Download Link: <https://nodejs.org/en/download/prebuilt-installer>

Download Node.js®

Download Node.js the way you want.

Package Manager **Prebuilt Installer** Prebuilt Binaries Source Code

I want the version of Node.js for running

 **Download Node.js v20.15.1**

Node.js includes [npm \(10.7.0\)](#).

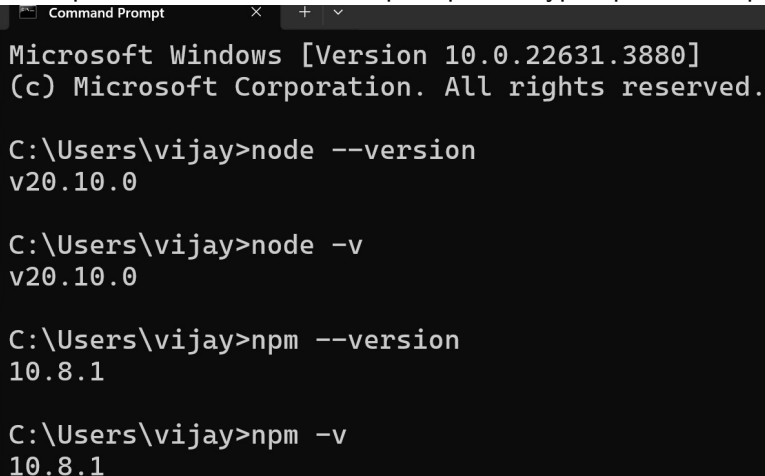
[Read the changelog for this version](#)

[Read the blog post for this version](#)

[Learn how to verify signed SHASUMS](#)

Verify Installation

- Node: Go to the command prompt and type `node -v` or `node --version`
- npm: Go to the command prompt and type `npm -v` or `npm --version`



```
Microsoft Windows [Version 10.0.22631.3880]
(c) Microsoft Corporation. All rights reserved.

C:\Users\vijay>node --version
v20.10.0

C:\Users\vijay>node -v
v20.10.0

C:\Users\vijay>npm --version
10.8.1

C:\Users\vijay>npm -v
10.8.1
```

First Project

Steps to implement Tailwind

- Create a project folder in the machine.
- Open project folder using Visual studio Code.
- create a file index.html
- include the tailwind.css as below:

```
<link href="https://cdn.jsdelivr.net/npm/tailwindcss@2.19/dist/tailwind.min.css" rel="stylesheet">
```

- write the sample code below:

```
<div class="text-6xl font-bold underline">  
welcome to SNIST  
</div>
```

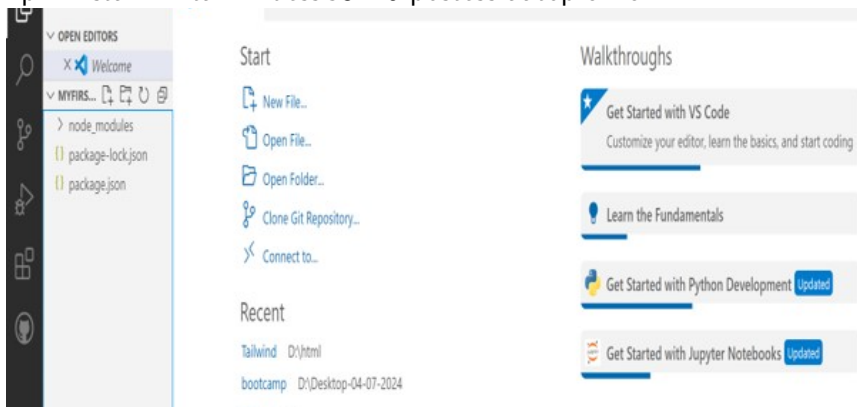
- open the index.html

Output

First Project

Steps to create the project

- Create a project folder in the machine.
- Open project folder using Visual studio Code.
- Go the terminal and type the following command
`npm install -D tailwindcss@3.1.0 postcss autoprefixer`



First Project continued..

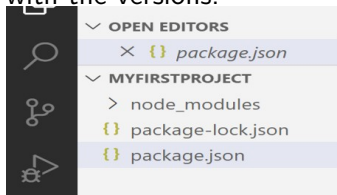
In the above command:

- 'tailwindcss@3.1.0': This specifies that you want to install the latest version of the Tailwind CSS framework. Tailwind CSS is a utility-first CSS framework used for designing web interfaces.
- 'postcss@latest': This specifies that you want to install the latest version of PostCSS. PostCSS is a tool for transforming CSS with JavaScript plugins, and it's commonly used to process CSS with Tailwind CSS.
- 'autoprefixer@latest': This specifies that you want to install the latest version of Autoprefixer. Autoprefixer is a PostCSS plugin that automatically adds vendor prefixes to CSS rules, ensuring compatibility with different browsers.

First Project continued..

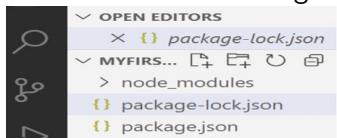
Understanding folder structure

- 1 **nodemodules:** It is a package manager folder for node.js consists all the necessary modules installed.
- 2 **package.json:** It shows all the dependencies installed in the project with the versions.



```
{ } package.json > ...  
1  {  
2    "dependencies": {  
3      "autoprefixer": "^10.4.19",  
4      "postcss": "^8.4.39",  
5      "tailwindcss": "^3.4.4"  
6    }  
7  }
```

- 3 **package-lock.json:** It locks all the dependencies installed and make sure those versions will get installed in various environments.



```
{ } package-lock.json > ...  
1  {  
2    "name": "myfirstproject",  
3    "lockfileVersion": 3,  
4    "requires": true,  
5    "packages": {  
6      "": {
```

First Project continued..

Creating a configuration file

- Go to the terminal and type the command **npx tailwindcss init** will generate **tailwind.config.css** file
- Configure the above file to refer all files(html/js/css) in **src** folder as shown right side.



```
JS tailwind.config.js > ...
1  /** @type {import('tailwindcss').Config} */
2  module.exports = {
3    content: [],
4    theme: {
5      extend: {},
6    },
7    plugins: [],
8  }
9
10
11
12
```

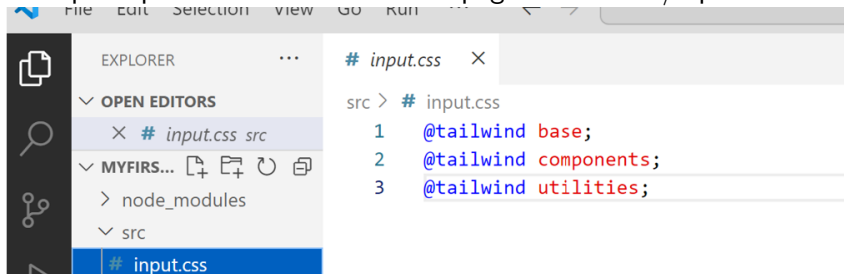
```
JS tailwind.config.js > [unknown]
1  /** @type {import('tailwindcss').Config}
2  module.exports = {
3    content: ["./**/*.html", "./**/*.js"],
4    theme: {
5      extend: {},
6    },
7    plugins: [],
8  }
9
10
11
12
```

First Project continued..

Creating a CSS file: src/input.css and write the below code:

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

- This file is created to pull out the tailwind directives, through which we generate a output css file with these directives to use in html to develop unique and customizable web pages. Path: src/input.css

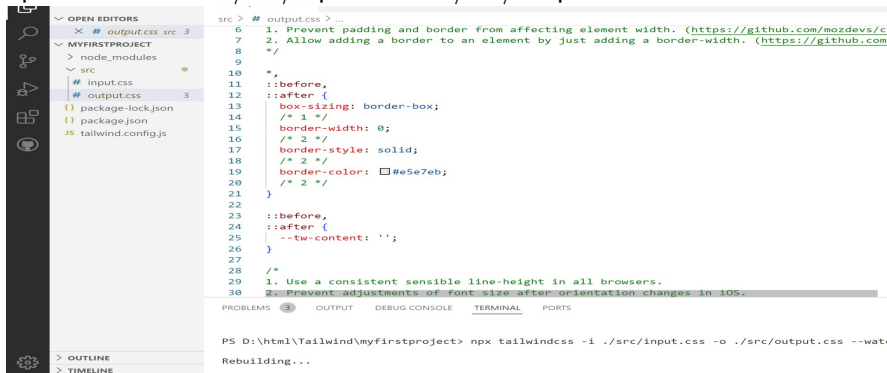


First Project continued..

Process a css file created above and generate new file:

- This can be done in multiple ways, one way is to type the following command in terminal:

```
npx tailwindcss -i ./src/input.css -o ./src/output.css --watch
```



```

6 1. Prevent padding and border from affecting element width. (https://github.com/mozdevs/
7 2. Allow adding a border to an element by just adding a border-width. (https://github.com
8 */
9
10 *,
11 ::before,
12 ::after {
13   box-sizing: border-box;
14   /* 1 */
15   border-width: 0;
16   /* 2 */
17   border-style: solid;
18   /* 2 */
19   border-color: #e5e7eb;
20   /* 2 */
21 }
22
23 ::before,
24 ::after {
25   --tw-content: '';
26 }
27
28 /*
29 1. Use a consistent sensible line-height in all browsers.
30 2. Prevent adjustments of font size after orientation changes in iOS.

```

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

PS D:\html\Tailwind\myfirstproject> npx tailwindcss -i ./src/input.css -o ./src/output.css --watch
Rebuilding...

```

First Project continued..

Create a Webpage

- Create a new file index.html under src folder
- Import the output.css in the above html file and include the following code as a sample:

```
<div class="text-6xl font-bold underline">  
welcome to SNIST  
</div>
```

Output

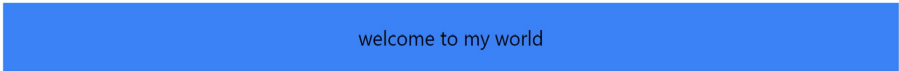
welcome to SNIST

Utility Classes

Common Utility Classes:

- Margin and padding ('m-4', 'p-4')
- Text alignment ('text-center')
- Background color ('bg-blue-500')

```
<div class="m-4 p-4 text-center bg-blue-500">  
  welcome to my world  
</div>
```



welcome to my world

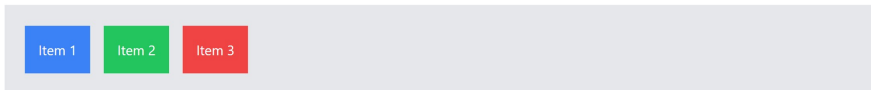
Utility Classes..

Flexbox Utilities:

- Using 'flex', 'flex-row', 'flex-col'
- Aligning items ('items-center', 'justify-between')

Example:

```
<div class="flex flex-row bg-gray-200 p-4 mt-10">  
  <div class="bg-blue-500 text-white p-4 m-2">Item 1</div>  
  <div class="bg-green-500 text-white p-4 m-2">Item 2</div>  
  <div class="bg-red-500 text-white p-4 m-2">Item 3</div>  
</div>
```



Utility Classes..

Grid Utilities:

- Using 'grid', 'grid-cols-3'
- Gap utilities ('gap-4')

Example:

```
<div class="grid grid-cols-3 gap-4" □ bg-gray-200 p-4">  
<div class="□ bg-blue-500 □ text-white p-4">Item 1</div>  
<div class="□ bg-green-500 □ text-white p-4">Item 2</div>  
<div class="□ bg-red-500 □ text-white p-4">Item 3</div>  
<div class="□ bg-yellow-500 □ text-white p-4">Item 4</div>  
<div class="□ bg-purple-500 □ text-white p-4">Item 5</div>  
<div class="□ bg-pink-500 □ text-white p-4">Item 6</div>  
</div>
```



Utility Classes..

Typography:

- Font size ('text-lg', 'text-2xl')
- Font weight ('font-bold', 'font-light')
- Line height ('leading-tight')

```
<div class="text-lg font-bold bg-gray-200 p-4 mb-4">  
  This is text with `text-lg` and `font-bold` classes,  
  making it large and bold.
```

```
</div>
```

```
<div class="text-2xl font-light leading-tight bg-gray-200 p-4 mb-4">  
  This is text with `text-2xl` and `font-light` classes,  
  making it 2 times extra large and light.  
  This is text with `text-2xl` and `font-light` classes,  
  making it 2 times extra large and light.
```

```
</div>
```

This is text with `text-lg` and `font-bold` classes, making it large and bold.

This is text with `text-2xl` and `font-light` classes, making it 2 times extra large and light. This is text with `text-2xl` and `font-light` classes, making it 2 times extra large and light.

Utility Classes..

Color Utilites:

- Text color ('text-red-500')
- Background color ('bg-green-500')
- Border color ('border-blue-500')

Example:

```
<div class="bg-blue-500 text-orange-400 border border-green-400 p-4 mb-4">  
  This is a div with a blue background color.  
</div>
```

This is a div with a blue background color.

Utility Classes..

Spacing Utilities:

- Margin ('m-4', 'mt-2')
- Padding ('p-4', 'pt-2')
- Space between ('space-x-4', 'space-y-4')

Example:

```
<div class="bg-yellow-500 text-white p-6 mt-4 mb-4">
```

This div has padding of 6 (1.5rem or 24px) on all sides, a top margin of 4 (1rem or 16px) and a bottom margin of 4 (1rem or 16px).

```
</div>
```

This div has padding of 6 (1.5rem or 24px) on all sides, a top margin of 4 (1rem or 16px) and a bottom margin of 4 (1rem or 16px).

Utility Classes..

Spacing Utilities: Spacing Example

```
<div class="bg-yellow-500 text-white p-6 mt-4 mb-4 flex space-x-4">
```

```
<div class='w-1/2 bg-gray-400'>
```

This div has padding of 6 (1.5rem or 24px) on all sides, a top margin of 4 (1rem or 16px) and a bottom margin of 4 (1rem or 16px).

```
</div>
```

```
<div class='w-1/2 bg-green-400'>
```

This div has padding of 6 (1.5rem or 24px) on all sides, a top margin of 4 (1rem or 16px) and a bottom margin of 4 (1rem or 16px).

```
</div>
```

```
</div>
```

This div has padding of 6 (1.5rem or 24px) on all sides, a top margin of 4 (1rem or 16px) and a bottom margin of 4 (1rem or 16px).

This div has padding of 6 (1.5rem or 24px) on all sides, a top margin of 4 (1rem or 16px) and a bottom margin of 4 (1rem or 16px).

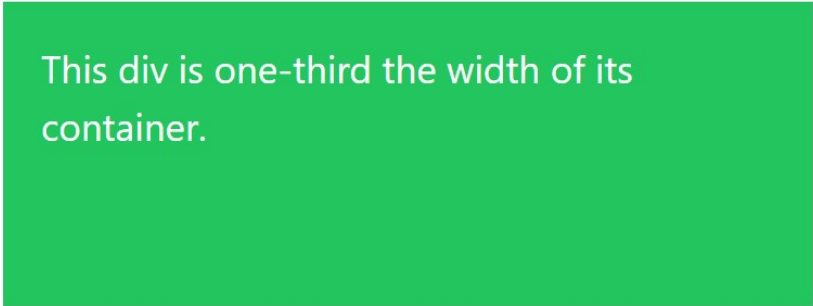
Utility Classes..

Background Utilities:

- Background color ('bg-yellow-300')
- Background image and position

Example:

```
<div class="bg-green-500 text-white w-1/3 h-32 p-4 mb-4">  
  This div is one-third the width of its container.  
</div>
```



This div is one-third the width of its container.

Utility Classes..

Responsive Utilities:

- 'sm': Small screens (min-width: 640px)
- 'md': Medium screens (min-width: 768px)
- 'lg': Large screens (min-width: 1024px)
- 'xl': Extra large screens (min-width: 1280px)
- '2xl': 2x Extra large screens (min-width: 1536px)

Example:

```
<h1 class="text-base sm:text-lg md:text-xl lg:text-2xl">
```

Responsive Heading

```
</h1>
```

Responsive Heading

Utility Classes..

Responsive Utilities: One more example

```
<div class="container mx-auto px-4 py-8">
  <h1 class="text-3xl font-bold mb-8 text-center">Product Gallery</h1>
  <div class="grid grid-cols-1 sm:grid-cols-2 md:grid-cols-3 lg:grid-cols-4 gap-4">
    <div class="bg-white rounded-lg shadow-md p-4">
      <h2 class="text-xl font-semibold mt-4">Product 1</h2>
      <p class="text-gray-600 mt-2">$20.00</p>
    </div>
    <div class="bg-white rounded-lg shadow-md p-4">
      <h2 class="text-xl font-semibold mt-4">Product 2</h2>
      <p class="text-gray-600 mt-2">$25.00</p>
    </div>
    <div class="bg-white rounded-lg shadow-md p-4">
      <h2 class="text-xl font-semibold mt-4">Product 3</h2>
      <p class="text-gray-600 mt-2">$30.00</p>
    </div>
    <div class="bg-white rounded-lg shadow-md p-4">
      <h2 class="text-xl font-semibold mt-4">Product 4</h2>
      <p class="text-gray-600 mt-2">$35.00</p>
    </div>
  </div>
</div>
```

Utility Classes..

Responsive Utilities: Output

Product Gallery

Product 1 \$20.00	Product 2 \$25.00	Product 3 \$30.00	Product 4 \$35.00
-----------------------------	-----------------------------	-----------------------------	-----------------------------

Questionnaire

Short Answer Questions

- 1 What is Tailwind CSS, and what makes it different from other CSS frameworks?
- 2 Why would you choose Tailwind CSS for a project instead of other frameworks?
- 3 What prior knowledge is helpful before starting with Tailwind CSS?
- 4 What does the "utility-first" design philosophy mean in the context of Tailwind CSS?
- 5 What are utility classes in Tailwind CSS?

Long Answer Questions

- 1 What are utility classes in Tailwind CSS, and how do they function within the framework? Describe how these classes, such as text-center or bg-blue-500, are used to apply specific styles directly in the HTML markup and how they contribute to a utility-first design approach.
- 2 What are the essential steps to create and configure a project with Tailwind CSS?
- 3 How can you customize Tailwind CSS classes to fit specific design requirements for your project? Explain in detail?
- 4 What are the main reasons for choosing Tailwind CSS over other CSS frameworks or writing custom CSS? Elaborate the Advantages and Disadvantages
- 5 What are utility classes in Tailwind CSS used to give responsive design, and how are they used to style HTML elements? Explain with an example?

A set of navigation icons typically found in Beamer presentations, including symbols for back, forward, search, and other slide controls.