Assignment 19: Introduction to Tailwind CSS

Theory Assignment

1. **Explain what Tailwind CSS is and how it differs from traditional CSS frameworks.**
   Tailwind CSS is a utility-first CSS framework that provides a large set of utility classes for styling directly in your HTML. Unlike traditional frameworks like Bootstrap, which use pre-designed components, Tailwind focuses on small, reusable utility classes (e.g., text-center, p-4) that allow you to design custom styles without writing CSS.

2. **Describe the concept of utility-first CSS and its advantages.**
   Utility-first CSS emphasizes using utility classes to style elements instead of writing custom CSS rules.
   Advantages:

   - Faster development since you can directly apply classes without creating new CSS rules.
   - Highly customizable as styles are built directly into HTML.
   - Reduced CSS file size by removing unused styles during production with tools like PurgeCSS.
   - Consistent styling due to reusable utility classes.

3. **List and explain at least five common Tailwind classes.**

   - **text-center**: Centers text horizontally.
   - **p-4**: Adds padding of 1rem to all sides.
   - **bg-blue-500**: Sets the background color to a blue shade.
   - **flex**: Applies flexbox to an element.
   - **rounded-lg**: Adds large rounded corners to an element.

Practical Assignment

1. **Redesign your HTML webpage using Tailwind CSS by:**

   - Applying utility classes for styling and layout adjustments.
     Example: Use classes like bg-gray-100, flex, and text-center to style your webpage.
   - Ensuring it is responsive with Tailwind's responsive utilities.
     Example: Use responsive classes like sm:text-sm, md:text-md, and lg:text-lg for font sizes.

2. **Create a pricing table using Tailwind with three columns for different pricing options, including buttons and card elements.**
   Example Code:

html

Copy code

```
<div class="flex justify-center space-x-4">
```

```
  <div class="bg-white shadow-lg rounded-lg p-6">
    <h3 class="text-xl font-bold">Basic</h3>
    <p class="text-gray-500">$10/month</p>
    <button class="mt-4 bg-blue-500 text-white px-4 py-2 rounded">Choose Plan</button>
  </div>
  <div class="bg-white shadow-lg rounded-lg p-6">
    <h3 class="text-xl font-bold">Standard</h3>
    <p class="text-gray-500">$20/month</p>
    <button class="mt-4 bg-blue-500 text-white px-4 py-2 rounded">Choose Plan</button>
  </div>
  <div class="bg-white shadow-lg rounded-lg p-6">
    <h3 class="text-xl font-bold">Premium</h3>
    <p class="text-gray-500">$30/month</p>
    <button class="mt-4 bg-blue-500 text-white px-4 py-2 rounded">Choose Plan</button>
  </div>
</div>
```

Assignment 20: Advanced Tailwind CSS Components

Theory Assignment

1. **Explain how Tailwind’s configuration file works and its role in customizing Tailwind.**
   Tailwind’s configuration file (tailwind.config.js) allows you to customize the framework, such as adding custom colors, spacing, breakpoints, and plugins. You can define a theme that aligns with your project’s design requirements, making it more flexible.

2. **Describe how to create responsive designs using Tailwind’s breakpoints.**
   Tailwind uses predefined breakpoints (sm, md, lg, xl, 2xl) for responsive design. You can prefix utility classes with breakpoints to apply styles conditionally.
   Example:

html

Copy code

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

3. **Discuss using custom colors and spacing with Tailwind’s configuration.**
   You can add custom colors and spacing in tailwind.config.js under the theme property.
   Example:

js

Copy code

```
module.exports = {
  theme: {
    extend: {
      colors: {
        customBlue: '#1E40AF',
      },
      spacing: {
        '72': '18rem',
      },
    },
  },
};
```

Practical Assignment

1. **Customize Tailwind’s configuration to include a new color scheme and spacing values.**
   Update tailwind.config.js with:

js

Copy code

```
module.exports = {
  theme: {
    extend: {
      colors: {
        customGreen: '#10B981',
      },
      spacing: {
        '84': '21rem',
      },
    },
  },
};
```

2. **Create a custom button and card component using Tailwind and your custom configurations.**
   Example Button:

html

Copy code

```
<button class="bg-customGreen text-white px-6 py-2 rounded-md hover:bg-green-700">
  Custom Button
</button>
```

Example Card:

html

Copy code

```
<div class="bg-white shadow-lg rounded-lg p-6">
  <h3 class="text-xl font-bold">Custom Card</h3>
  <p class="text-gray-500">This is a customizable card.</p>
</div>
```

3. **Design a login page layout with Tailwind, including a form and a styled submit button.**
   Example Code:

html

Copy code

```
<div class="flex items-center justify-center min-h-screen bg-gray-100">
  <form class="bg-white p-6 rounded-lg shadow-md">
    <h2 class="text-2xl font-bold mb-4">Login</h2>
    <input type="email" placeholder="Email" class="block w-full p-2 mb-4 border rounded">
    <input type="password" placeholder="Password" class="block w-full p-2 mb-4 border rounded">
    <button class="bg-blue-500 text-white px-4 py-2 rounded hover:bg-blue-700">
      Submit
    </button>
  </form>
```

</div>